LilyPond

The music typesetter

Fragmentos de código

The LilyPond development team

This document shows a selected set of LilyPond snippets from the LilyPond Wiki (`https://wiki.lilypond.community`). It is in the public domain.

For more information about how this manual fits with the other documentation, or to read this manual in other formats, see Sección “Manuals” en *General Information*.

If you are missing any manuals, the complete documentation can be found at `https://lilypond.org/`.

This document has been placed in the public domain.

For LilyPond version 2.27.4

Índice General

Preface **1**

Musical notation

1 Pitches **5**

Añadir un ámbito por voz 5
Añadir una indicación de octava alta a una sola voz 5
Cabezas de nota tipo Aiken, variante delgada 6
Alterar la longitud de las plicas unidas por una barra 6
Indicaciones de tesitura 7
Indicación de tesitura después de la armadura 7
Ámbitos con varias voces 8
Aplicar estilos de cabeza según la nota de la escala 8
Cambiar la dirección de la plica de las notas de la tercera línea automáticamente, basado en la melodía 9
Cambiar el texto de las indicaciones de octava alta y baja 10
Modificación de la separación en las indicaciones de tesitura 10
Cambiar el intervalo de las líneas de la pauta 11
Las claves se pueden transponer en intervalos arbitrarios 12
Colorear las notas según su altura 12
Crear una secuencia de notas a distintas alturas 13
Creación de armaduras de clave personalizadas 14
Direction of merged 'fa' shape note heads 14
Forzar un becuadro de cancelación antes de una alteración accidental 15
Forzar la impresión de la clave 15
Generación de notas aleatorias 15
Ocultar las alteraciones sobre notas ligadas al principio de un sistema nuevo 16
Mantener el tamaño del símbolo en los cambios de clave 16
Ejemplo de *makam* 17
Modificación de la inclinación de la línea de octava 17
Armaduras de tonalidad no tradicionales 18
Números como notas de notación fácil 18
Plantilla de orquesta, coro y piano 20
Evitar que se añadan becuadros adicionales automáticamente 25
Evitar que se impriman becuadros cuando cambia la armadura 25
Citar otra voz con transposición 25
Separar las cancelaciones de tonalidad de los cambios de armadura 26
Transposing pitches with minimum accidentals ("smart" transpose) 27
Ejemplo de *makam* en la música turca 28
Trucaje de las propiedades de clave 29
Using \autoChange with more than one voice 31

2 Rhythms **32**

Adding beams, slurs, ties, etc., when using tuplet and non-tuplet rhythms 32
Escritura de partes de percusión 32
Ajuste del espaciado de las notas de adorno 33

Alineación de los números de compás 33
Formas alternativas de la figura breve 34
Apoyatura o nota de adorno antes de la línea divisoria 35
Subdivisiones de barra automáticas 35
Modificar duraciones automáticamente 36
Finales de barra en el contexto Score 37
Barras recortadas 38
Barras que atraviesan saltos de línea 39
Cambiar el salto de las barras en ángulo 40
Cambiar la forma de los silencios multicompás 40
Modificar el número de puntillos de una nota 40
Cambiar el tempo sin indicación metronómica 41
Cambiar el número del grupo especial 41
Modificar el compás dentro de una sección polimétrica utilizando `\scaleDurations` 41
Chant or psalm notation 43
Complex time signatures 43
Símbolos de dirección y símbolos de agrupación de compás 44
Controlar la visibilidad de los corchetes de grupo especial 44
Ejemplo de cencerro y campana 45
Crear indicaciones metronómicas en modo de marcado 46
Grabado manual de las ligaduras 46
Trémolos con barras flotantes 47
Escribir varios tresillos con una sola instrucción `\tuplet` 48
Forzar que las marcas de ensayo comiencen en una letra o número dados 48
Generar corchetes personalizados 49
Ritmos rasgueados de guitarra 50
Indicaciones de compás polimétricas profundamente personalizadas 51
Ejemplo de cajas chinas grave y aguda 52
Making an object invisible using \hide 52
Hacer ligaduras de expresión con estructura compleja de discontinuidad 53
Controlar manualmente las posiciones de las barras 54
Combinar los silencios multicompás en una parte polifónica 54
Modificar la longitud del corchete de grupo especial 54
Desplazar las notas con puntillo en polifonía 55
Control de la anchura de un silencio multicompás 56
Marcado de silencios multicompás 56
Números de agrupación especial distintos a los predeterminados 57
Numerar silencios seguidos de un compás 57
`Partcombine` y `\autoBeamOff` 58
Ejemplo de percusión 58
Permitir saltos de línea dentro de grupos especiales con barra 60
Colocar las barras de las notas de adorno a la misma altura que las barras de notas normales 61
Posicionamiento de las notas de adorno con espacio flotante 61
Posicionar los silencios multicompás 62
Colocar calderones opuestos sobre la barra de compás 63
Evitar que una marca de ensayo final suprima el corchete de un grupo de valoración especial 64
Imprimir números de compás a intervalos regulares 64
Impresión de números de compás para compases divididos 65
Imprimir números de compás dentro de rectángulos o circunferencias 65
Impresión de los números de compás usando `modulo-bar-number-visible` 66

Impresión de los números de compás con intervalos regulares variables 67
Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama 67
Imprimir música que tenga compases distintos en cada pentagrama 68
Imprimir el número de compás en el primer compás 70
Imrpesión de corchetes de tresillo en el lado de la cabeza de la nota 71
Redefinición de los valores globales predeterminados para notas de adorno 71
Suprimir los números de compás de toda la partitura 72
Estilos de silencios 72
Alteración de los finales de barra predeterminados 74
Barras rítmicas 74
Silencios de separación en modo de letra 74
Silencios de separación en modo de letra (2) 75
Stemlets (plicas de tamaño reducido) 75
Barras que se atienen al pulso estrictamente 76
Subdivisión de barras 76
Ejemplo de música para tam-tam 78
Ejemlo de pandereta 78
Rectángulo de tres segmentos visibles 78
Indicación de compás entre corchetes 79
Compás entre paréntesis 80
Indicación de compás imprimiendo solo el numerador (en lugar de la fracción) 80
Trucar la disposición de las notas de adorno dentro de la música 80
Indicaciones de compás personalizadas 81
Uso de estilos alternativos para los corchetes 81
Utilizar la barra que tacha las notas de adorno con notas normales 83
Uso de ligaduras en los arpegios 83

3 Expressive marks 84

Encerrar entre paréntesis una marca expresiva o una nota de un acorde 84
Añadir marcas de tiempo a glissandos largos 84
Ajustar las posiciones de las ligaduras verticalmente 85
Ajustar la forma de las subidas y caídas de tono 85
Alinear los extremos de los reguladores con las direcciones de `NoteColumn` 86
Formas alternativas de la figura breve 86
Ligaduras asimétricas 87
Marcas de respiración 87
Regulador interrumpido 88
Caesura ("railtracks") with fermata 89
Centrar texto debajo de un regulador 89
Cambiar el texto y los estilos de objeto de extensión para las indicaciones dinámicas textuales 91
Modificar el aspecto continuo de una ligadura de expresión a punteado o intermitente 91
Cambiar el símbolo de la marca de respiración 91
Modificar el número de puntillos de una nota 92
Combinar indicaciones dinámicas con marcados textuales 92
Combining dynamics with markup texts (2) 92
Glissando contemporáneo 93
Controlar la visibilidad de los objetos de extensión después de un salto de línea 93
Controlar la colocación de las digitaciones de acordes 94
Controlar la ordenación vertical de las inscripciones 95
Crear un grupeto de anticipación 96

Crear arpegios entre notas de voces distintas 96
Crear arpegios que se cruzan entre pentagramas dentro de un sistema de piano 96
Creación de arpegios que se cruzan entre pentagramas dentro de otros contextos 97
Crear digitaciones de dos cifras 98
Hacer ligaduras entre voces distintas 98
Crear elementos de extensión textuales 99
Dynamics spanner with custom text 100
Glissando por debajo de un objeto gráfico 100
Reguladores con distintos estilos de línea 100
Ocultar la línea de extensión de las expresiones textuales de dinámica 101
Horizontally aligning custom dynamics like “più f” 101
Insertar una cesura 104
Ligaduras *laissez vibrer* 105
Puntas de flecha para las líneas 105
Hacer ligaduras de expresión con estructura compleja de discontinuidad 106
Modificar los valores predeterminados para la notación abreviada de las articulaciones . 106
Desplazar ligaduras de expresión verticalmente 107
Desplazar los extremos de un regulador 108
Posicionar símbolos de arpegio 108
Situar los elementos de marcado de texto por dentro de las ligaduras 108
Imprimir reguladores en diversos estilos 109
Impresión de reguladores utilizando la notación «al niente» 109
Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama 110
Establecer el comportamiento de los reguladores en las barras de compás 110
Ajustar la longitud mínima de los reguladores 111
Imprimir la misma articulación encima y debajo de la misma nota o acorde 111
Slanted parentheses around dynamics 112
Snap pizzicato (“Bartok” pizzicato) 112
Using \arpeggioBracket to make divisi more visible 113
Using a bracket to clarify divisi 113
Usar una raya corta como símbolo de respiración 114
Utilizar ligaduras dobles para acordes legato 114
Uso de la propiedad `whiteout` 115
Línea vertical como una articulación barroca 115
Alinear verticalmente expresiones dinámicas que abarcan varias notas 116

4 Repeats 117

Cambiar las líneas divisorias predeterminadas 117
Controlar el aspecto de las barras de trémolo 117
Trémolos de pentagrama cruzado 119
Trémolos con barras flotantes 119
Símbolos de porcentaje sueltos 120
Measure counters 120
Visibilidad del contador de repeticiones de tipo porcentaje 121
Contador de repeticiones de tipo porcentaje 122
Posicionar los símbolos de repetición segno y coda (con saltos de línea) 122
Establecer el tipo de repetición doble predeterminado para la primera y segunda vez ... 123
Acortar los corchetes de primera y segunda vez 123
Unfolding tremolo repeats 124
Corchetes de primera y segunda vez debajo de los acordes 124
Casillas de primera y segunda vez en varios pentagramas 125

Elementos de marcado para el texto de las casillas de repetición usando `repeatCommands` . 126

5 Simultaneous notes . . . 127

Voces adicionales para evitar colisiones . . . 127
Cambiar los textos de `\partCombine` . . . 127
Changing a single note's size in a chord . . . 128
Clusters («racimos») . . . 128
Combinar dos partes sobre el mismo pentagrama . . . 129
Impresión de acordes complejos . . . 130
Forzar el desplazamiento horizontal de las notas . . . 130
Making an object invisible using \hide . . . 131
Desplazar las notas con puntillo en polifonía . . . 132
Evitar las advertencias sobre columnas de notas que chocan . . . 132
Dos pares de `\partCombine` en un solo pentagrama . . . 133

6 Staff notation . . . 135

Añadir un ámbito por voz . . . 135
Añadir un pentagrama nuevo . . . 135
Añadir un pentagrama adicional en un salto de línea . . . 136
Añadir indicadores a los pentagramas que se dividen después de un salto de línea . . . 137
Añadir notas guía orquestales a una partitura vocal . . . 141
Numeración de compases alternativa . . . 143
Indicación de tesitura después de la armadura . . . 144
Cambiar las líneas divisorias predeterminadas . . . 145
Cambiar el número de líneas de una pauta . . . 145
Modificar el tamaño de la pauta . . . 146
Crear pentagramas en blanco . . . 146
Creación de armaduras de clave personalizadas . . . 148
Plicas de pentagrama cruzado . . . 149
Mostrar corchete o llave en grupos de un solo pentagrama . . . 150
Mostrar un sistema GrandStaff completo si sigue con vida uno solo de sus pentagramas . 150
Extending a trill spanner . . . 152
Extender glissandos sobre repeticiones . . . 153
Ligaduras planas . . . 154
Forcing measure width to adapt to a metronome mark's width . . . 157
Glissando por debajo de un objeto gráfico . . . 157
Harmonizing bar line thickness for staves with different sizes . . . 157
Incipits . . . 160
Insertar fragmentos de partituras sobre el pentagrama, como elementos de marcado . . . 164
Hacer que `TabStaff` imprima la cuerda superior en la parte de abajo . . . 165
Formateado de tablaturas con letras . . . 166
Hacer que los glissandos se puedan dividir en el salto de línea . . . 166
Hacer unas líneas del pentagrama más gruesas que las otras . . . 168
Measure counters . . . 168
Disposición *Mensurstriche* (líneas divisorias entre pentagramas) . . . 169
Modificación de la inclinación de la línea de octava . . . 169
Anidado de grupos de pentagramas . . . 170
Armaduras de tonalidad no tradicionales . . . 171
Plantilla de orquesta, coro y piano . . . 172
Print chord names with same root and different bass as slash and bass note . . . 177
Putting lyrics inside the staff . . . 179

Citar otra voz 179
Citar otra voz con transposición 181
Quitar la llave en el primer sistema de una partitura de piano 181
Quitar las barras de compás entre los pentagramas de un StaffGroup, PianoStaff o GrandStaff 182
Quitar la primera línea vacía 183
Fijar un separador entre los sistemas 184
Dar forma a ligaduras de unión individuales entre acordes 185
Barras de compás en forma de raya corta 188
Indicación de compás entre corchetes 188
Compás entre paréntesis 188
Trucaje de las propiedades de clave 189
Dos pares de `\partCombine` en un solo pentagrama 190
Uso del corchete recto al comienzo de un grupo de pentagramas 192
Using \autoChange with more than one voice 192
Using mark lines in a Frenched score 193
Grupos de pentagramas alineados verticalmente sin barra inicial de conexión 196
Corchetes de primera y segunda vez debajo de los acordes 202
Casillas de primera y segunda vez en varios pentagramas 203

7 Editorial annotations 205

Añadir digitaciones a la partitura 205
Añadir enlaces a los objetos 205
Insertar elementos de marcado en una tablatura 207
Permitir que las digitaciones se impriman dentro del pentagrama 207
Numeración de compases alternativa 208
Corchetes de análisis encima del pentagrama 209
Corchetes de análisis con etiquetas 210
Aplicar estilos de cabeza según la nota de la escala 211
Poner en blanco las líneas del pentagrama utilizando la instrucción `\whiteout` 212
Changing a single note's size in a chord 212
Modificar el aspecto continuo de una ligadura de expresión a punteado o intermitente 212
Colorear las notas según su altura 213
Controlar la colocación de las digitaciones de acordes 214
Crear un grupeto de anticipación 215
Crear pentagramas en blanco 215
Crear digitaciones de dos cifras 217
Dirección predeterminada de las plicas sobre la tercera línea del pentagrama 217
Diferentes tamaños de fuente para `instrumentName` y `shortInstrumentName` 218
Rodear los objetos gráficos con rectángulos 219
Trazar circunferencias alrededor de la cabeza de las notas 220
Rodear diversos objetos con una circunferencia 220
Empotrar PostScript nativo dentro de un bloque \markup 221
Generate special note head shapes 221
Líneas de rejilla: modificar su aspecto 222
Líneas de rejilla: destacar ritmos y la relación temporal entre notas 223
Ligados ascendentes y descendentes (*hammer-on* y *pull-off*) 224
Indicaciones de ligado ascendente y descendente (*hammer-on* y *pull-off*) usando acordes 224
Indicaciones de *hammer-on* y *pull-off* usando voces 225
Hacer unas líneas del pentagrama más gruesas que las otras 225
Marcar notas de las partes habladas con unas aspas en la plica 225

Measure counters ... 226
Corchete que abarca varios compases ... 227
Posicionamiento preciso de indicaciones de digitación ... 228
Situar los elementos de marcado de texto por dentro de las ligaduras ... 229
Impresión de texto de derecha a izquierda ... 229
Líneas de extensión para números de cuerda ... 229
Uso de la propiedad `whiteout` ... 230

8 Text ... 231

Insertar elementos de marcado en una tablatura ... 231
Añadir la fecha actual a una partitura ... 231
Adjusting vertical spacing of lyrics ... 232
Alinear y centrar los nombres de instrumento ... 232
Alineación de sílabas con melisma ... 234
Aligning text marks to notes ... 234
Poner en blanco las líneas del pentagrama utilizando la instrucción `\whiteout` ... 235
Centrar texto debajo de un regulador ... 235
Cambiar el texto de las indicaciones de octava alta y baja ... 237
Cambiar la familia de tipografía predeterminada para el texto ... 237
Combinar indicaciones dinámicas con marcados textuales ... 238
Combining dynamics with markup texts (2) ... 239
Combinar dos partes sobre el mismo pentagrama ... 239
Crear elementos de extensión textuales ... 240
Demonstrating all \header fields ... 241
Empotrar PostScript nativo dentro de un bloque \markup ... 243
Dar formato a sílabas de la letra ... 243
Cómo poner ligaduras entre sílabas de la letra ... 243
Alineación de la letra ... 244
Objetos de lista de marcados ... 244
Marcado de silencios multicompás ... 245
Acerca de la ubicuidad de los objetos de marcado ... 246
Imprimir el número de versión ... 247
Plantilla de piano con letra centrada ... 247
Imprimir marcas de ensayo en cualquier pentagrama ... 248
Impresión de texto de derecha a izquierda ... 249
Putting lyrics inside the staff ... 249
Slanted parentheses around dynamics ... 250
Elemento de marcado de texto independiente en dos columnas ... 250
Líneas de extensión para números de cuerda ... 251
Rectángulo de tres segmentos visibles ... 251
UTF-8 ... 252
Plantilla para conjunto vocal con letras alineadas encima y debajo de los pentagramas ... 254
Elementos de marcado para el texto de las casillas de repetición usando `repeatCommands` ... 255

Specialist notation

9 Vocal music 259

Añadir un ámbito por voz 259
Añadir indicadores a los pentagramas que se dividen después de un salto de línea 259
Añadir notas guía orquestales a una partitura vocal 263
Adjusting vertical spacing of lyrics 265
Alineación de sílabas con melisma 265
Indicaciones de tesitura 266
Indicación de tesitura después de la armadura 267
Ámbitos con varias voces 267
Ancient notation template – modern transcription of Gregorian music 268
Plantilla de salmo del canto anglicano 268
Disposición de distintas letras en la misma línea 271
Modificar la tipografía para cada estrofa 272
Chant or psalm notation 273
Forzar la visibilidad de los guiones separadores de sílabas 273
Dar formato a sílabas de la letra 274
Cómo poner ligaduras entre sílabas de la letra 274
Plantilla para himnos 274
Alineación de la letra 276
Marcar notas de las partes habladas con unas aspas en la plica 277
Plantilla de orquesta, coro y piano 278
Plantilla de piano con melodía y letra 283
Putting lyrics inside the staff 284
Plantilla de coro SATB, a cuatro pentagramas 284
Plantilla de pentagrama único con música, letra y acordes 286
Single-staff template with notes, lyrics, chords, and frets 287
Plantilla de pentagrama único don notas y letra 288
Silencios de separación en modo de letra 288
Silencios de separación en modo de letra (2) 289
Using \arpeggioBracket to make divisi more visible 289
Using a bracket to clarify divisi 290
Uso de etiquetas para producir música mensural y moderna a partir del mismo código fuente 290
Alineación vertical de la letra y los compases de ossia 293
Alineación vertical de los números de estrofa de diferentes pentagramas 294
Letra común centrada verticalmente 295
Plantilla de conjunto vocal 296
Plantilla de conjunto vocal con reducción de piano automática 298
Plantilla para conjunto vocal con letras alineadas encima y debajo de los pentagramas 301
Estrofa para solista y estribillo a dos voces 302

10 Keyboard and other multi-staff instruments 305

Símbolos de registración de acordeón 305
Modificar el texto de las indicaciones de pedal 306
Clusters («racimos») 306
Controlar la colocación de las digitaciones de acordes 307
Hacer ligaduras entre voces distintas 307
Solución de problemas de acordes y barras que cruzan el pentagrama 308
Trémolos de pentagrama cruzado 309

Ajuste fino de las líneas de pedal ... 309
Indicating cross-staff chords with a bracket ... 310
Plantilla para combo de jazz ... 311
Ligaduras *laissez vibrer* ... 316
Plantilla de piano (sencilla) ... 317
Plantilla de piano con letra centrada ... 317
Plantilla de piano con melodía y letra ... 318
Quitar la llave en el primer sistema de una partitura de piano ... 319
Using \autoChange with more than one voice ... 320
Plantilla de conjunto vocal con reducción de piano automática ... 321

11 Unfretted string instruments ... 324

Hacer ligaduras entre voces distintas ... 324
Armónicos con puntillo ... 324
Snap pizzicato ("Bartok" pizzicato) ... 325
Plantilla de cuarteto de cuerda (sencilla) ... 325
Plantilla de cuarteto de cuerda con particellas independientes ... 326

12 Fretted string instruments ... 330

Añadir digitaciones a la partitura ... 330
Añadir digitaciones a las tablaturas ... 330
Insertar elementos de marcado en una tablatura ... 330
Permitir que las digitaciones se impriman dentro del pentagrama ... 331
Automatic fretboards barré ... 332
Modificar la orientación de los trastes ... 332
Chord changes for fretboards ... 333
Glissando de acordes en tablatura ... 334
Acordes con digitaciones ampliadas para diagramas de trastes y tablaturas ... 335
Controlar la colocación de las digitaciones de acordes ... 335
Personalizar los diagramas de posiciones ... 336
Personalizar diagramas de posiciones de marcado ... 337
Digitaciones, indicación del número de cuerda y digitaciones de mano derecha ... 339
Notación del flamenco ... 340
Diagramas de posiciones de acorde, explicados y desarrollados ... 343
Tablas alternativas de diagramas de posiciones ... 350
Armónicos sobre cuerdas pisadas en tablatura ... 352
Ligaduras de guitarra ... 353
Ritmos rasgueados de guitarra ... 354
Ligados ascendentes y descendentes (*hammer-on* y *pull-off*) ... 355
Indicaciones de ligado ascendente y descendente (*hammer-on* y *pull-off*) usando acordes ... 356
Indicaciones de *hammer-on* y *pull-off* usando voces ... 356
Cómo cambiar la posición de un diagrama de posiciones ... 356
Plantilla para combo de jazz ... 357
Ligaduras *laissez vibrer* ... 363
Hacer que `TabStaff` imprima la cuerda superior en la parte de abajo ... 364
Formateado de tablaturas con letras ... 365
Referencia para armónicos sobre cuerdas al aire ... 365
Posicionamiento de digitaciones de mano derecha ... 367
Polifonía en tablaturas ... 368
Setting up predefined fretboards for other instruments ... 369
Deslizamientos en tablatura ... 371

Comportamiento de las plicas y las barras de corchea en tablaturas ... 372
Líneas de extensión para números de cuerda ... 372

13 Percussion ... 374

Escritura de partes de percusión ... 374
Ejemplo de cencerro y campana ... 375
Indicaciones de compás polimétricas profundamente personalizadas ... 375
Ejemplo de cajas chinas grave y aguda ... 376
Plantilla para combo de jazz ... 377
Símbolos de golpes de percusión ... 382
Ejemplo de percusión ... 385
Imprimir música que tenga compases distintos en cada pentagrama ... 387
Ejemplo de música para tam-tam ... 389
Ejemlo de pandereta ... 389

14 Wind instruments ... 391

Modificar el tamaño de los diagramas de viento madera ... 391
Símbolos de digitación para instrumentos de viento ... 391
Notación de los golpes de lengua de la flauta ... 392
Diagramas para viento madera gráficos y textuales ... 392
Diagramas de digitación para la flauta dulce ... 393
Listas de llaves para los diagramas de viento madera ... 394
Listado de los diagramas para viento madera ... 395

15 Chord notation ... 397

Añadir un bajo cifrado encima o debajo de las notas ... 397
Añadir barras de compás al contexto de nombres de acorde (ChordNames) ... 397
Ajustar los glifos de las alteraciones en el bajo cifrado ... 398
Changing a single note's size in a chord ... 398
Modificación del separador de acordes ... 398
Cambiar las posiciones de las alteraciones del bajo cifrado ... 399
Excepciones para los nombres de acorde ... 399
Nombre de acorde maj7 ... 400
Nombres alternativos de acordes ... 400
Acordes con digitaciones ampliadas para diagramas de trastes y tablaturas ... 410
Clusters («racimos») ... 411
Controlar la colocación de las digitaciones de acordes ... 411
Solución de problemas de acordes y barras que cruzan el pentagrama ... 412
Personalización del estilo de las rejillas de acordes ... 413
Personalización del símbolo de Sin Acorde ... 414
Impresión de acordes complejos ... 414
Dividir manualmente las líneas de extensión del bajo cifrado para ciertos números solamente ... 415
Print chord names with same root and different bass as slash and bass note ... 415
Imprimir los acordes cuando se produce un cambio ... 418
Hoja guía de acordes o *lead sheet* sencilla ... 418
Plantilla de pentagrama único con música, letra y acordes ... 419
Single-staff template with notes, lyrics, chords, and frets ... 419
Plantilla de pentagrama único con música y acordes ... 420
Centrado vertical de las líneas de bajo cifrado emparejadas ... 421
Corchetes de primera y segunda vez debajo de los acordes ... 422

16 Contemporary music 423

Barras recortadas 423
Regulador interrumpido 424
Modificar el compás dentro de una sección polimétrica utilizando `\scaleDurations` 424
Clusters («racimos») 426
Glissando contemporáneo 426
Ligaduras planas 427
Notación de los golpes de lengua de la flauta 429
Indicaciones de compás polimétricas profundamente personalizadas 430
Marcar notas de las partes habladas con unas aspas en la plica 431
Armaduras de tonalidad no tradicionales 431
Imprimir música que tenga compases distintos en cada pentagrama 432
Screech and boink 434
Stemlets (plicas de tamaño reducido) 436

17 Ancient notation 438

Añadir un bajo cifrado encima o debajo de las notas 438
Tipografía de música antigua 438
Ancient notation template – modern transcription of Gregorian music 441
Indicaciones de compás antiguas 442
Chant or psalm notation 442
Custos 442
Incipits 443
Disposición *Mensurstriche* (líneas divisorias entre pentagramas) 448
Estilos de silencios 448
Uso de etiquetas para producir música mensural y moderna a partir del mismo código fuente 450
Línea vertical como una articulación barroca 452

18 World music 453

Improvisación de música árabe 453
Ejemplo de *makam* 453
Impresión de texto de derecha a izquierda 453
Ejemplo de *makam* en la música turca 454

Other collections

19 Automatic notation 459

Subdivisiones de barra automáticas 459
Forzar que las marcas de ensayo comiencen en una letra o número dados 459
Generación de partituras completas y partes de libro en Scheme sin usar el analizador sintáctico 460
Evitar que se añadan becuadros adicionales automáticamente 462
Evitar que se impriman becuadros cuando cambia la armadura 462
Plantilla de conjunto vocal con reducción de piano automática 463

20 Breaks ... 466
Añadir un pentagrama adicional en un salto de línea ... 466
Posicionar los símbolos de repetición segno y coda (con saltos de línea) ... 467
Quitar la primera línea vacía ... 467

21 Connecting notes ... 470
Adding beams, slurs, ties, etc., when using tuplet and non-tuplet rhythms ... 470
Subdivisiones de barra automáticas ... 470
Modificar el aspecto continuo de una ligadura de expresión a punteado o intermitente ... 471
Controlar la visibilidad de los corchetes de grupo especial ... 471
Hacer ligaduras entre voces distintas ... 472
Ligaduras *laissez vibrer* ... 473
Controlar manualmente las posiciones de las barras ... 473

22 Contexts and engravers ... 474
Añadir un ámbito por voz ... 474
Añadir un pentagrama nuevo ... 474
Añadir un pentagrama adicional en un salto de línea ... 475
Añadir barras de compás al contexto de nombres de acorde (ChordNames) ... 476
Indicación de tesitura después de la armadura ... 476
Corchetes de análisis con etiquetas ... 477
Cambiar la dirección de la plica de las notas de la tercera línea automáticamente, basado en la melodía ... 477
Modificar la salida MIDI para que tenga un canal por cada voz ... 478
Modificar el compás dentro de una sección polimétrica utilizando `\scaleDurations` ... 479
Crear arpegios entre notas de voces distintas ... 480
Crear pentagramas en blanco ... 481
Creación de arpegios que se cruzan entre pentagramas dentro de otros contextos ... 483
Creación de armaduras de clave personalizadas ... 483
Plicas de pentagrama cruzado ... 484
Definir un grabador en Scheme: grabador de ámbito o tesitura ... 485
Mostrar un sistema GrandStaff completo si sigue con vida uno solo de sus pentagramas ... 492
Los grabadores, uno por uno ... 494
Líneas de rejilla: modificar su aspecto ... 497
Líneas de rejilla: destacar ritmos y la relación temporal entre notas ... 498
Measure counters ... 499
Corchete que abarca varios compases ... 500
Disposición *Mensurstriche* (líneas divisorias entre pentagramas) ... 501
Anidado de grupos de pentagramas ... 501
Permitir saltos de línea dentro de grupos especiales con barra ... 502
Print chord names with same root and different bass as slash and bass note ... 503
Imprimir marcas de ensayo en cualquier pentagrama ... 506
Imprimir música que tenga compases distintos en cada pentagrama ... 506
Suprimir los números de compás de toda la partitura ... 509
Uso del corchete recto al comienzo de un grupo de pentagramas ... 509
Using mark lines in a Frenched score ... 510
Uso de etiquetas para producir música mensural y moderna a partir del mismo código fuente ... 513
Estrofa para solista y estribillo a dos voces ... 515
Corchetes de primera y segunda vez debajo de los acordes ... 517
Casillas de primera y segunda vez en varios pentagramas ... 518

23 Education ... 519
Líneas de rejilla: destacar ritmos y la relación temporal entre notas ... 519
Hacer unas líneas del pentagrama más gruesas que las otras ... 520

24 Headword ... 521
Encabezamiento de música antigua ... 521
Encabezamiento de acordes ... 524
Encabezamiento de indicaciones editoriales ... 526
Encabezamiento de indicaciones expresivas ... 527
Encabezamiento de bajo cifrado ... 528
Encabezamiento de instrumentos con trastes ... 530
Encabezamiento de instrumentos de teclado ... 533
Encabezamiento de alturas ... 537
Encabezamiento de repeticiones ... 538
Encabezamiento de Duraciones ... 540
Encabezamiento de notas simultáneas ... 542
Encabezamiento de pentagramas ... 544
Encabezamiento de Texto ... 546
Encabezamiento de Instrumentos sin trastes ... 548
Encabezamiento de música vocal ... 551
Encabezamiento de instrumentos de viento ... 553

25 MIDI ... 555
Modificar la salida MIDI para que tenga un canal por cada voz ... 555
Cambiar el tempo sin indicación metronómica ... 556
Creacaión de matices dinámicos personalizados en el MIDI ... 556
Demo of MIDI instruments ... 557
Reemplazo de la ecualización predeterminada de instrumentos MIDI ... 560

26 Non-music ... 562
Alinear y centrar los nombres de instrumento ... 562
Demonstrating all \header fields ... 563
Listas de llaves para los diagramas de viento madera ... 564

27 Paper and layout ... 566
Alinear y centrar los nombres de instrumento ... 566
Disposición de distintas letras en la misma línea ... 567
Partes de libro ... 568
Modificar el tamaño de la pauta ... 572
Recorte de sistemas ... 573
Crear pentagramas en blanco ... 575
Demonstrating all \header fields ... 577
Mostrar un sistema GrandStaff completo si sigue con vida uno solo de sus pentagramas ... 578
Fijar un separador entre los sistemas ... 580
Índice general (tabla de contenidos) ... 581
Grupos de pentagramas alineados verticalmente sin barra inicial de conexión ... 583

28 Preparing parts ... 590
Forzar que las marcas de ensayo comiencen en una letra o número dados ... 590
Numerar silencios seguidos de un compás ... 590
Plantilla de cuarteto de cuerda con particellas independientes ... 590

29 Real music ... 594
Modificar la salida MIDI para que tenga un canal por cada voz ... 594
Crear una secuencia de notas a distintas alturas ... 595
Hacer ligaduras entre voces distintas ... 595
Trémolos de pentagrama cruzado ... 596
Demo of MIDI instruments ... 596
Armónicos con puntillo ... 600
Indicaciones de compás polimétricas profundamente personalizadas ... 600
Indicating cross-staff chords with a bracket ... 601
Insertar fragmentos de partituras sobre el pentagrama, como elementos de marcado ... 602
Ejemplo de percusión ... 603
Imprimir música que tenga compases distintos en cada pentagrama ... 604

30 Really cool ... 607
Añadir la fecha actual a una partitura ... 607
Poner en blanco las líneas del pentagrama utilizando la instrucción `\whiteout` ... 607
Centrar texto debajo de un regulador ... 608
Modificación de propiedades para objetos gráficos individuales ... 609
Clusters («racimos») ... 610
Colorear las notas según su altura ... 610
Crear una secuencia de notas a distintas alturas ... 611
Generación de notas aleatorias ... 612
Generación de partituras completas y partes de libro en Scheme sin usar el analizador sintáctico ... 612
Hacer unas líneas del pentagrama más gruesas que las otras ... 614
Armaduras de tonalidad no tradicionales ... 615
Imprimir música que tenga compases distintos en cada pentagrama ... 615

31 Really simple ... 619
Añadir un pentagrama nuevo ... 619
Escritura de partes de percusión ... 619
Añadir digitaciones a la partitura ... 620
Aligning text marks to notes ... 620
Corchetes de análisis encima del pentagrama ... 621
Changing a single note's size in a chord ... 621
Modificar la tipografía para cada estrofa ... 622
Modificar el aspecto continuo de una ligadura de expresión a punteado o intermitente ... 622
Combinar indicaciones dinámicas con marcados textuales ... 623
Combining dynamics with markup texts (2) ... 623
Forzar que las marcas de ensayo comiencen en una letra o número dados ... 624
Alineación de la letra ... 624
Combinar los silencios multicompás en una parte polifónica ... 625
Modificar la longitud del corchete de grupo especial ... 625
Imprimir el número de versión ... 626
Plantilla de piano (sencilla) ... 626

Plantilla de piano con letra centrada ... 626
Plantilla de piano con melodía y letra ... 627
Plantilla de pentagrama único con música, letra y acordes ... 628
Plantilla de pentagrama único con música y acordes ... 629
Plantilla de pentagrama único don notas y letra ... 630
Plantilla de un solo pentagrama, con notas únicamente ... 630
Silencios de separación en modo de letra ... 631
Silencios de separación en modo de letra (2) ... 631
Plantilla de cuarteto de cuerda (sencilla) ... 631
Uso de la instrucción \tweak para retocar objetos gráficos individuales ... 633
Plantilla de conjunto vocal ... 633
Casillas de primera y segunda vez en varios pentagramas ... 635

32 Scheme ... 637
Incluir digitaciones adicionales con Scheme ... 637
Añadir indicadores a los pentagramas que se dividen después de un salto de línea ... 637
Añadir enlaces a los objetos ... 641
Añadir notas guía orquestales a una partitura vocal ... 643
Añadir la fecha actual a una partitura ... 645
Ajustar las posiciones de las ligaduras verticalmente ... 646
Centrar texto debajo de un regulador ... 646
Modificación de propiedades para objetos gráficos individuales ... 648
Excepciones para los nombres de acorde ... 649
Colorear las notas según su altura ... 650
Crear una secuencia de notas a distintas alturas ... 651
Creacaión de matices dinámicos personalizados en el MIDI ... 651
Personalizar la posición y el número de puntos en las líneas divisorias de repetición ... 652
Definir un grabador en Scheme: grabador de ámbito o tesitura ... 653
Diferentes tamaños de fuente para `instrumentName` y `shortInstrumentName` ... 660
Imprimir el árbol genealógico de un grob ... 661
Trazar circunferencias alrededor de la cabeza de las notas ... 663
Rodear diversos objetos con una circunferencia ... 663
Extender glissandos sobre repeticiones ... 664
Ligaduras planas ... 665
Notación de los golpes de lengua de la flauta ... 668
Tablas alternativas de diagramas de posiciones ... 668
Generate special note head shapes ... 670
Generar corchetes personalizados ... 671
Generación de notas aleatorias ... 672
Generación de partituras completas y partes de libro en Scheme sin usar el analizador sintáctico ... 672
Símbolos de porcentaje sueltos ... 675
Números como notas de notación fácil ... 675
Modificación de articulaciones por tipo ... 676
Posicionamiento de las notas de adorno con espacio flotante ... 678
Print chord names with same root and different bass as slash and bass note ... 679
Reemplazo de la ecualización predeterminada de instrumentos MIDI ... 681
Separar las cancelaciones de tonalidad de los cambios de armadura ... 682
Slanted parentheses around dynamics ... 683
Líneas de extensión para números de cuerda ... 683
Rectángulo de tres segmentos visibles ... 684

Transposing pitches with minimum accidentals ("smart" transpose) 685
Dos pares de \partCombine en un solo pentagrama 686
Indicaciones de compás personalizadas 688
Utilizar ly:grob-object para acceder a los grobs con \tweak 688
Línea vertical como una articulación barroca 690

33 Spacing 691

Adjusting vertical spacing of lyrics 691
Permitir que las digitaciones se impriman dentro del pentagrama 691
Breaking horizontal alignment of dynamics and textscripts 692
Desalinear indicaciones dinámicas y textuales verticalmente 693
Harmonizing bar line thickness for staves with different sizes 693
Etiqueta de página 695
Espaciado de las notas estrictamente proporcional 696
Indicaciones dinámicas y textuales alineadas verticalmente 697
Alineación vertical de la letra y los compases de ossia 697

34 Specific notation 699

Símbolos de registración de acordeón 699
Añadir barras de compás al contexto de nombres de acorde (ChordNames) 700
Escritura de partes de percusión 700
Añadir digitaciones a las tablaturas 701
Cabezas de nota tipo Aiken, variante delgada 701
Permitir que las digitaciones se impriman dentro del pentagrama 702
Cambiar el número de líneas de una pauta 702
Chant or psalm notation 703
Excepciones para los nombres de acorde 703
Nombre de acorde maj7 704
Acordes con digitaciones ampliadas para diagramas de trastes y tablaturas 705
Clusters («racimos») 705
Glissando contemporáneo 706
Controlar la colocación de las digitaciones de acordes 706
Ejemplo de cencerro y campana 707
Crear pentagramas en blanco 708
Custos 709
Demo of MIDI instruments 710
Direction of merged 'fa' shape note heads 714
Empotrar PostScript nativo dentro de un bloque \markup 714
Los grabadores, uno por uno 715
Notación del flamenco 718
Ejemplo de cajas chinas grave y aguda 721
Cómo cambiar la posición de un diagrama de posiciones 722
Cómo poner ligaduras entre sílabas de la letra 723
Ligaduras *laissez vibrer* 723
Ejemplo de percusión 723
Ejemplo de música para tam-tam 725
Ejemlo de pandereta 725
Indicación de compás entre corchetes 726
Compás entre paréntesis 726
Uso de una voz adicional para los saltos de línea y de página 726
Listado de los diagramas para viento madera 727

35 Symbols and glyphs ... 729
Símbolos de registración de acordeón ... 729
Añadir indicadores a los pentagramas que se dividen después de un salto de línea ... 730
Tipografía de música antigua ... 734
Marcas de respiración ... 737
Regulador interrumpido ... 738
Caesura ("railtracks") with fermata ... 738
Custos ... 739
Personalizar la posición y el número de puntos en las líneas divisorias de repetición ... 740
Símbolos de digitación para instrumentos de viento ... 741
Cómo poner ligaduras entre sílabas de la letra ... 742
Posicionar los símbolos de repetición segno y coda (con saltos de línea) ... 742
Estilos de silencios ... 743
Elementos de marcado para el texto de las casillas de repetición usando `repeatCommands` ... 744

36 Templates ... 745
Ancient notation template – modern transcription of Gregorian music ... 745
Plantilla de salmo del canto anglicano ... 745
Plantilla para himnos ... 748
Plantilla para combo de jazz ... 750
Plantilla de orquesta, coro y piano ... 756
Plantilla de piano (sencilla) ... 761
Plantilla de piano con letra centrada ... 761
Plantilla de piano con melodía y letra ... 762
Plantilla de coro SATB, a cuatro pentagramas ... 763
Plantilla de pentagrama único con música, letra y acordes ... 765
Single-staff template with notes, lyrics, chords, and frets ... 766
Plantilla de pentagrama único con música y acordes ... 767
Plantilla de pentagrama único don notas y letra ... 767
Plantilla de un solo pentagrama, con notas únicamente ... 768
Plantilla de cuarteto de cuerda (sencilla) ... 769
Plantilla de cuarteto de cuerda con particellas independientes ... 770
Plantilla de conjunto vocal ... 772
Plantilla de conjunto vocal con reducción de piano automática ... 774
Plantilla para conjunto vocal con letras alineadas encima y debajo de los pentagramas ... 776
Estrofa para solista y estribillo a dos voces ... 778

37 Titles ... 781
Añadir la fecha actual a una partitura ... 781
Alinear y centrar los nombres de instrumento ... 781
Demonstrating all \header fields ... 783
Imprimir el número de versión ... 784

38 Tweaks and overrides ... 785
Añadir una indicación de octava alta a una sola voz ... 785
Añadir enlaces a los objetos ... 785
Insertar elementos de marcado en una tablatura ... 787
Añadir marcas de tiempo a glissandos largos ... 788
Ajuste del espaciado de las notas de adorno ... 788
Ajustar las posiciones de las ligaduras verticalmente ... 789

Adjusting vertical spacing of lyrics 790
Aligning text marks to notes 790
Alterar la longitud de las plicas unidas por una barra 791
Numeración de compases alternativa 791
Corchetes de análisis encima del pentagrama 793
Corchetes de análisis con etiquetas 793
Ligaduras asimétricas 794
Breaking horizontal alignment of dynamics and textscripts 794
Desalinear indicaciones dinámicas y textuales verticalmente 795
Caesura ("railtracks") with fermata 796
Changing a single note's size in a chord 796
Cambiar el grosor y el espaciado de las barras 797
Cambiar la forma de los silencios multicompás 797
Modificación de propiedades para objetos gráficos individuales 798
Cambiar el texto y los estilos de objeto de extensión para las indicaciones dinámicas textuales 798
Cambiar la familia de tipografía predeterminada para el texto 799
Modificar el tamaño de la pauta 800
Cambiar el tempo sin indicación metronómica 800
Modificar el texto de las indicaciones de pedal 801
Controlar la visibilidad de los objetos de extensión después de un salto de línea 801
Controlar el aspecto de las barras de trémolo 802
Controlar la ordenación vertical de las inscripciones 803
Controlar la visibilidad de los corchetes de grupo especial 804
Crear un grupeto de anticipación 805
Creación de armaduras de clave personalizadas 805
Crear elementos de extensión textuales 806
Solución de problemas de acordes y barras que cruzan el pentagrama 807
Plicas de pentagrama cruzado 808
Custos 808
Personalizar los diagramas de posiciones 809
Personalizar diagramas de posiciones de marcado 811
Mostrar corchete o llave en grupos de un solo pentagrama 812
Imprimir el árbol genealógico de un grob 813
Armónicos con puntillo 815
Rodear los objetos gráficos con rectángulos 815
Rodear diversos objetos con una circunferencia 816
Dynamics spanner with custom text 816
Extending a trill spanner 817
Extender glissandos sobre repeticiones 817
Ajuste fino de las líneas de pedal 819
Ligaduras planas 819
Forzar un becuadro de cancelación antes de una alteración accidental 822
Forzar el desplazamiento horizontal de las notas 822
Diagramas de posiciones de acorde, explicados y desarrollados 822
Generate special note head shapes 829
Generar corchetes personalizados 830
Glissando por debajo de un objeto gráfico 831
Reguladores con distintos estilos de línea 832
Horizontally aligning custom dynamics like "più f" 832
Cómo cambiar la posición de un diagrama de posiciones 835
Insertar una cesura 836

Mantener el tamaño del símbolo en los cambios de clave ... 837
Puntas de flecha para las líneas ... 837
Making an object invisible using \hide ... 838
Hacer que los glissandos se puedan dividir en el salto de línea ... 839
Controlar manualmente las posiciones de las barras ... 840
Número del compás centrado entre las barras ... 840
Disposición *Mensurstriche* (líneas divisorias entre pentagramas) ... 841
Modificación de la inclinación de la línea de octava ... 842
Desplazar las notas con puntillo en polifonía ... 842
Desplazar ligaduras de expresión verticalmente ... 843
Anidado de grupos de pentagramas ... 844
Modificación de articulaciones por tipo ... 845
Visibilidad del contador de repeticiones de tipo porcentaje ... 846
Posicionar símbolos de arpegio ... 847
Posicionamiento preciso de indicaciones de digitación ... 847
Posicionar los silencios multicompás ... 848
Situar los elementos de marcado de texto por dentro de las ligaduras ... 849
Imprimir números de compás dentro de rectángulos o circunferencias ... 849
Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama ... 850
Impresión de los nombres de las notas con o sin indicación de la octava ... 850
Imrpesión de corchetes de tresillo en el lado de la cabeza de la nota ... 851
Espaciado de las notas estrictamente proporcional ... 852
Quitar la llave en el primer sistema de una partitura de piano ... 853
Quitar las barras de compás entre los pentagramas de un StaffGroup, PianoStaff o GrandStaff ... 854
Quitar la primera línea vacía ... 854
Estilos de silencios ... 855
Barras rítmicas ... 856
Separar las cancelaciones de tonalidad de los cambios de armadura ... 857
Establecer el comportamiento de los reguladores en las barras de compás ... 858
Fijar un separador entre los sistemas ... 858
Dar forma a ligaduras de unión individuales entre acordes ... 859
Imprimir la misma articulación encima y debajo de la misma nota o acorde ... 862
Líneas de extensión para números de cuerda ... 863
Evitar las advertencias sobre columnas de notas que chocan ... 863
Indicación de compás entre corchetes ... 863
Compás entre paréntesis ... 864
Indicación de compás imprimiendo solo el numerador (en lugar de la fracción) ... 864
Corchete de tresillo y cambio de pentagrama ... 865
Trucaje de las propiedades de clave ... 865
Trucar la disposición de las notas de adorno dentro de la música ... 867
Uso de estilos alternativos para los corchetes ... 867
Utilizar ly:grob-object para acceder a los grobs con \tweak ... 869
Uso de la instrucción \tweak para retocar objetos gráficos individuales ... 870
Indicaciones dinámicas y textuales alineadas verticalmente ... 870
Alineación vertical de la letra y los compases de ossia ... 871
Alineación vertical de los números de estrofa de diferentes pentagramas ... 872
Centrado vertical de las líneas de bajo cifrado emparejadas ... 873

39 Workaround ... 875

Añadir un pentagrama adicional en un salto de línea ... 875

Apoyatura o nota de adorno antes de la línea divisoria 876
Breaking horizontal alignment of dynamics and textscripts 876
Desalinear indicaciones dinámicas y textuales verticalmente 877
Modificar el compás dentro de una sección polimétrica utilizando `\scaleDurations` 878
Solución de problemas de acordes y barras que cruzan el pentagrama 879
Impresión de acordes complejos 880
Extender glissandos sobre repeticiones 881
Forcing measure width to adapt to a metronome mark's width 882
Hacer unas líneas del pentagrama más gruesas que las otras 882
Marcar notas de las partes habladas con unas aspas en la plica 883
Posicionamiento de las notas de adorno con espacio flotante 883
Posicionar los símbolos de repetición segno y coda (con saltos de línea) 884
Evitar que una marca de ensayo final suprima el corchete de un grupo de valoración especial 885
Impresión de texto de derecha a izquierda 886
Slanted parentheses around dynamics 886
Ejemlo de pandereta 887
Transposing pitches with minimum accidentals ("smart" transpose) 887
Unfolding tremolo repeats 889
Uso de una voz adicional para los saltos de línea y de página 889
Indicaciones dinámicas y textuales alineadas verticalmente 890

Preface

This document shows a selected set of LilyPond snippets from the LilyPond Wiki (`https://wiki.lilypond.community`), the successor of the LilyPond Snippet Repository (LSR).

We would like to address many thanks to Sebastiano Vigna for maintaining the LSR and Jean Abou Samra for maintaining the LilyPond Wiki!

Please note that this document is not an exact subset of the LilyPond Wiki: some snippets come from the `Documentation/snippets/new/` LilyPond sources directory, and snippets from the Wiki are converted by `convert-ly`, as the LilyPond Wiki is (mainly) based on a stable LilyPond version, and this document is for version 2.27.4.

Snippets are grouped by LilyPond Wiki categories (`https://wiki.lilypond.community/wiki/Special:Categories`), using the same chapter order as the Notation Reference (if possible). Snippets may be tagged with several categories, which means that they may appear multiple times in this document. Not all Wiki categories may be present here, though.

For all snippets, the indentation of staves is set to zero.

In the HTML version of this document, you can click on the file name or figure for each example to see the corresponding input file.

Musical notation
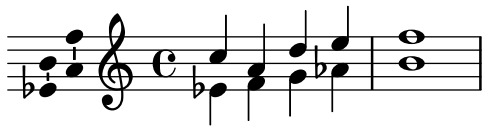
1 Pitches

See also Sección “Pitches” en *Notation Reference*.

Añadir un ámbito por voz

Se puede añadir un ámbito por cada voz. En este caso, el ámbito se debe desplazar manualmente para evitar colisiones.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



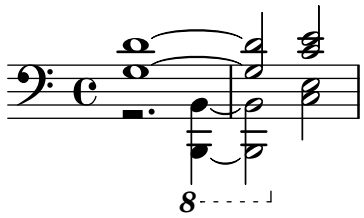
Añadir una indicación de octava alta a una sola voz

Si tiene más de una voz en el mismo pentagrama, el cambio de octavación de una voz transportará la posición de las notas en todas las voces mientras dure el corchete de octava. Si la octavación se quiere aplicar a una voz solamente, hay que trasladar el grabador `Ottava_spanner_engraver` al contexto `Voice`.

```
\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}

{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
    {
```

```
      r2.
      \ottava -1
      <b,,, b,,>4 ~ |
      q2
      \ottava 0
      <c e>2
    }
  >>
}
```
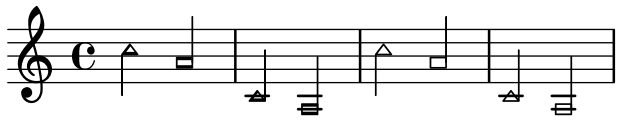


Cabezas de nota tipo Aiken, variante delgada

Las notas blancas de tipo Aiken se vuelven algo difíciles de leer con tamaños de pentagrama pequeños, especialmente con líneas adicionales. Al perder el espacio vacío del interior pueden parecer más bien figuras negras.

```
\score {
  {
    \aikenHeads
    c''2 a' c' a

    % Switch to thin-variant noteheads
    \set shapeNoteStyles = ##(doThin reThin miThin
                              faThin sol laThin tiThin)
    c'' a' c' a
  }
}
```

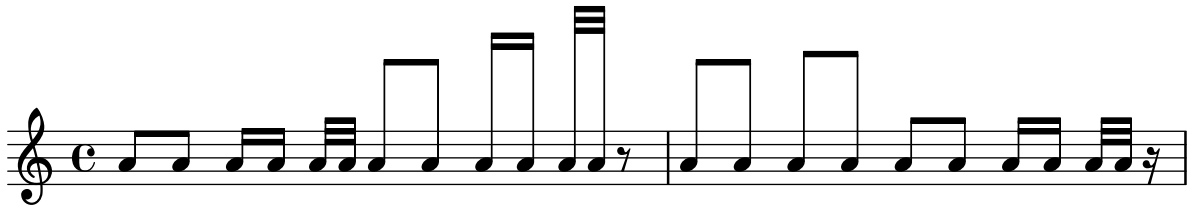


Alterar la longitud de las plicas unidas por una barra

Se puede variar la longitud de las plicas de las figuras unidas por una barra mediante la sobreescritura de la propiedad `beamed-lengths` de los detalles (`details`) del objeto `Stem`. Si se utiliza un solo valor como argumento, la longitud se aplica a todas las plicas. Si se usan varios argumentos, el primero se aplica a las corcheas, el sgundo a las semicorcheas y así sucesivamente. El último argumento también se aplica a todas las figuras que son mmás cortas que la longitud de la figura del último argumento. También se pueden usar argumentos no enteros.

```
\relative c'' {
  \override Stem.details.beamed-lengths = #'(2)
    a8[ a] a16[ a] a32[ a]
    \override Stem.details.beamed-lengths = #'(8 10 12)
    a8[ a] a16[ a] a32[ a] r8 |
  \override Stem.details.beamed-lengths = #'(8)
    a8[ a]
```

```
    \override Stem.details.beamed-lengths = #'(8.5)
    a8[ a]
    \revert Stem.details.beamed-lengths
    a8[ a] a16[ a] a32[ a] r16 |
}
```
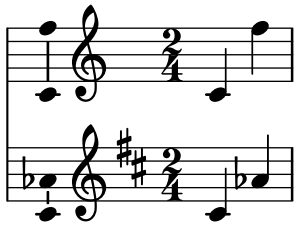


Indicaciones de tesitura

Las indicaciones de ámbito o tesitura indican rangos de alturas para las voces.

Las alteraciones accidentales solo se muestran si no forman parte de la armadura de tonalidad. Los objetos gráficos `AmbitusNoteHead` también tienen líneas adicionales.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

<<
  \new Staff {
    \relative c' {
      \time 2/4
      c4 f'
    }
  }
  \new Staff {
    \relative c' {
      \time  2/4
      \key d \major
      cis4 as'
    }
  }
>>
```
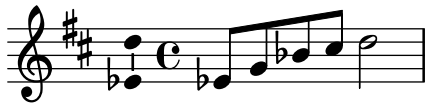


Indicación de tesitura después de la armadura

De manera predeterminada, las indicaciones de ámbito o tesitura se sitúan a la izquierda de la clave. La función `\ambitusAfter` permite cambiar esta colocación. La sintaxis es `\ambitusAfter` *grob-interface* (consulte Interfaces de los objetos gráficos (`https://lilypond.org/doc/v2.`

24/Documentation/internals/graphical-object-interfaces) para ver una lista de los valores posibles para *grob-interface*.)

Un caso de utilización bastante común es imprimir la indicación de texitura entre la armadura de la clave y la indicación de compás.
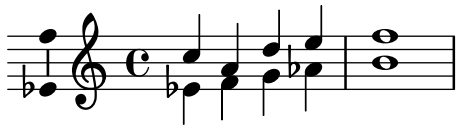
```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



Ámbitos con varias voces

La adición del grabador Ambitus_engraver al contexto de Staff crea un solo ámbito por pentagrama, incluso en el caso de pentagramas con varias voces.

```
\new Staff \with {
  \consists "Ambitus_engraver"
  }
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Aplicar estilos de cabeza según la nota de la escala

La propiedad shapeNoteStyles se puede usar para definir varios estilos de cabezas de nota para cada grado de la escala (según esté establecido por la armadura o por la propiedad tonic).

Esta propiedad requiere un conjunto de símbolos, que pueden ser puramente arbitrarios (se permiten expresiones geométricas como triangle, triángulo, cross, aspas, y xcircle, círculo con aspas) o basados en una antigua tradición americana de grabado (ciertos nombres de nota latinos trambién se permiten).

Dicho esto, para imitar antiguos cancioneros americanos, existen varios estilos predefinidos de cabezas de nota disponibles a través de instrucciones de abreviatura como \aikenHeads o \sacredHarpHeads.

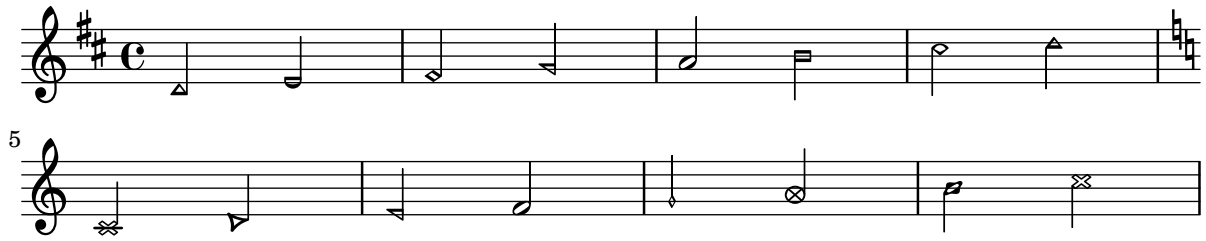
Este ejemplo muestra distintas formas de obtener cabezas de notas con forma, y muestra la capacidad de transportar una melodía sin perder la correspondencia entre las funciones armónicas y los estilos de cabezas de nota.

```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                              #f la ti)
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                              mensural xcircle diamond)
    \fragment
  }
}
```
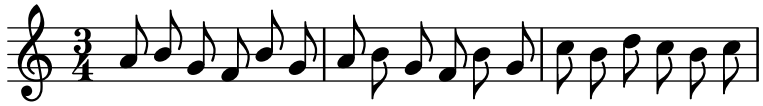


Cambiar la dirección de la plica de las notas de la tercera línea automáticamente, basado en la melodía

LilyPond puede alterar la dirección de la plica de las notas que van en la tercera línea de un pentagrama de forma que siga la melodía, mediante la adición del grabador `Melody_engraver` al contexto Voice.

Se puede usar la propiedad de contexto `suspendMelodyDecisions` para desactivar este comportamiento localmente.

```
\relative c'' {
  \time 3/4
  a8 b g f b g |
  \set suspendMelodyDecisions = ##t
  a  b g f b g |
  \unset suspendMelodyDecisions
  c  b d c b c |
}
```

```
\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \autoBeamOff
  }
}
```

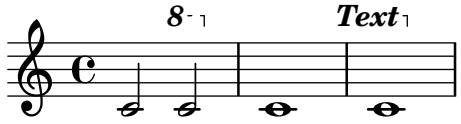


Cambiar el texto de las indicaciones de octava alta y baja

Internamente, `\ottava` fija las propiedades `ottavation` (por ejemplo para `8va` o `8vb`) y `middleCPosition`. Para sobreescribir el texto del corchete, fije `ottavation` después de invocar a `\ottava`.

Un texto breve es especialmente útil cuando se usa una indicación corta de octava.

```
{
  c'2
  \ottava 1
  \set Staff.ottavation = "8"
  c''2
  \ottava 0
  c'1
  \ottava 1
  \set Staff.ottavation = "Text"
  c''1
}
```



Modificación de la separación en las indicaciones de tesitura

Es posible ajustar la separación predeterminada entre las notas de la indicación de tesitura y la línea que las une.
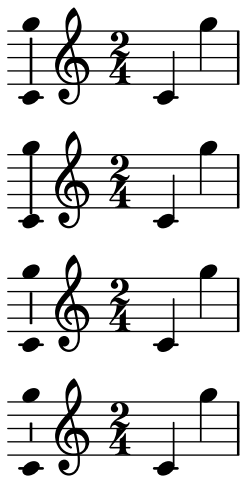
```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}
```

```
\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 0
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 1
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 1.5
  c'4 g''
}
```



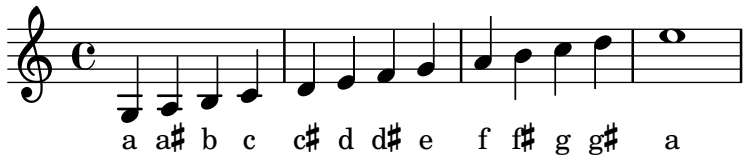
Cambiar el intervalo de las líneas de la pauta

Se usa staffLineLayoutFunction para cambiar la posición de las notas. Este fragmento de código muestra cómo ajustar su valor al de ly:pitch-semitones para producir una escala cromática con la distancia entre cada espacio y línea del pentagrama igual a un semitono.

```
scale = \relative c' {
  a4 ais b c
  cis4 d dis e
  f4 fis g gis
  a1
}

\new Staff \with {
  \remove "Accidental_engraver"
  staffLineLayoutFunction = #ly:pitch-semitones
}
{
  <<
    \scale
    \context NoteNames {
```
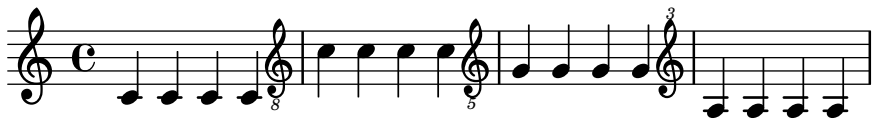
```
      \set printOctaveNames = ##f
      \scale
    }
  >>
}
```



Las claves se pueden transponer en intervalos arbitrarios

Se pueden transponer las claves en intervalos arbitrarios, no solo octavas.

```
\relative c' {
  \clef treble
  c4 c c c
  \clef "treble_8"
  c4 c c c
  \clef "treble_5"
  c4 c c c
  \clef "treble^3"
  c4 c c c
}
```



Colorear las notas según su altura

Es posible colorear la cabeza de las notas dependiendo de su altura y/o de su nombre: la función que se usa en este ejemplo hace posible incluso distinguir los armónicos.
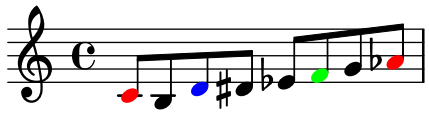
```
% Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 2 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 3 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 3 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 4 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 5 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 5 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 6 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 1 NATURAL) (x11-color 'blue))
    (cons (ly:make-pitch 0 3 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 4 FLAT) (x11-color 'blue))
    (cons (ly:make-pitch 0 5 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 6 FLAT) (x11-color 'blue))))
```

```
% Compare pitch and alteration (not octave).
#(define (pitch-equals? p1 p2)
   (and
    (= (ly:pitch-alteration p1) (ly:pitch-alteration p2))
    (= (ly:pitch-notename p1) (ly:pitch-notename p2))))

#(define (pitch-to-color pitch)
   (let ((color (assoc pitch color-mapping pitch-equals?)))
     (if color
         (cdr color))))

#(define (color-notehead grob)
   (pitch-to-color
    (ly:event-property (event-cause grob) 'pitch)))

\score {
  \new Staff \relative c' {
    \override NoteHead.color = #color-notehead
    c8 b d dis ees f g aes
  }
}
```
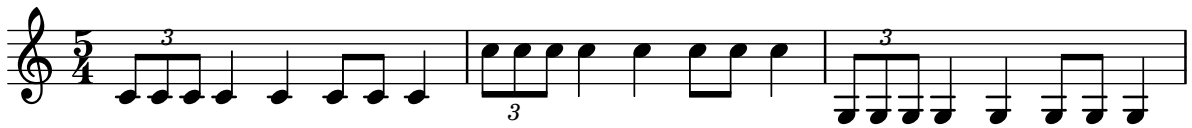


Crear una secuencia de notas a distintas alturas

En una música que tenga muchas apariciones de la misma secuencia de notas a distintas alturas, podría ser de utilidad la siguiente función musical. Admite una nota, de la que solo se utiliza su altura. Las funciones de apoyo en Scheme se han tomado prestadas del documento de "Consejos y trucos"de la versión 2.10 del manual.

Este ejemplo crea las duraciones rítmicas que se usan a todo lo largo de *Marte*, de *Los Planetas* de Gustav Holst.

```
rhythm =
#(define-music-function (p) (ly:pitch?)
   "Make the rhythm in Mars (the Planets) at the given pitch"
  #{ \tuplet 3/2 { $p 8 8 8 } 4 4 8 8 4 #})

\new Staff {
  \time 5/4
  \rhythm c'
  \rhythm c''
  \rhythm g
}
```
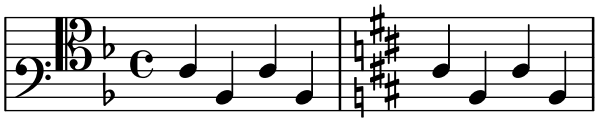


Creación de armaduras de clave personalizadas

LilyPond contempla armaduras de tonalidad personalizadas. En este ejemplo se muestra una armadura de Re menor y Re mayor con un rango ampliado de bemoles mostrados.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil =
    #(lambda (grob)
        (grob-interpret-markup grob
          #{ \markup\combine
               \musicglyph "clefs.C"
               \translate #'(-3 . -2)
                 \musicglyph "clefs.F"
          #}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}

{
  \key d\minor f bes, f bes, |
  \key d\major fis b, fis b, |
}
```
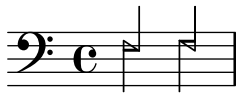


Direction of merged ‘fa’ shape note heads

Using property `NoteCollision.fa-merge-direction`, the direction of “fa” shape note heads (“fa”, “faThin”, etc.) can be controlled independently of the stem direction if two voices with the same pitch and different stem directions are merged. If this property is not set, the “down” glyph variant is used.

```
{
  \clef bass
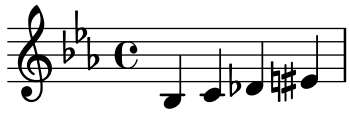
  << { \aikenHeads
       f2
       \override Staff.NoteCollision.fa-merge-direction = #UP
       f2 }
  \\ { \aikenHeads
       f2
       f2 }
  >>
}
```



Forzar un becuadro de cancelación antes de una alteración accidental

El ejemplo siguiente muestra cómo forzar un becuadro antes de una alteración accidental.
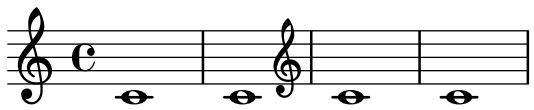
```
\relative c' {
  \key es \major
  bes c des
  \tweak Accidental.restore-first ##t
  eis
}
```



Forzar la impresión de la clave

Cuando ya se ha impreso la clave y aún no se ha cambiado a una distinta, LilyPond ignorará la repetición de la instrucción `\clef`, pues no constituye un cambio de clave. Es posible forzar la reimpresión de la clave usando la instrucción `\set Staff.forceClef = ##t`.

```
\relative c' {
  \clef treble
  c1
  \clef treble
  c1
  \set Staff.forceClef = ##t
  c1
  \clef treble
  c1
}
```



Generación de notas aleatorias

Este fragmento de código basado en Scheme genera notas aleatorias. Úselo como
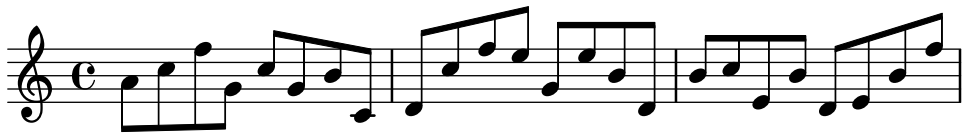
`\randomNotes` *n desde hasta duración*

para generar *n* notas aleatorias entre las alturas *desde* y *hasta*, con una longitud de *duración*.

```
randomNotes =
#(define-music-function (n from to dur)
   (integer? ly:pitch? ly:pitch? ly:duration?)
   (let ((from-step (ly:pitch-steps from))
         (to-step (ly:pitch-steps to)))
     (make-sequential-music
      (map (lambda (_)
             (let* ((step (+ from-step
                             (random (- to-step from-step))))
```

```
                (pitch (ly:make-pitch 0 step 0)))
            #{ $pitch $dur #}))
        (iota n)))))

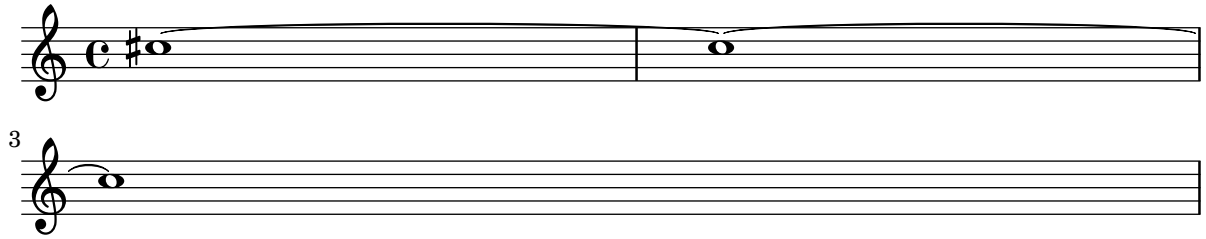
\randomNotes 24 c' g'' 8
```



Ocultar las alteraciones sobre notas ligadas al principio de un sistema nuevo

Aquí se muestra la manera de ocultar las alteraciones de las notas ligadas al comienzo de un sistema nuevo.

```
\relative c'' {
  \override Accidental.hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```
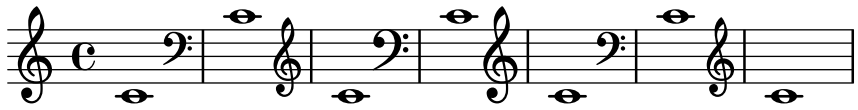


Mantener el tamaño del símbolo en los cambios de clave

Cuando se produce un cambio de clave, el símbolo de clave se imprime a un tamaño menor que la clave inicial. Esto se puede ajustar estableciendo el valor de la propiedad de contexto `full-size-change` a `#t`.

```
\relative c' {
  \clef "treble"
  c1
  \clef "bass"
  c1
  \clef "treble"
  c1
  \override Staff.Clef.full-size-change = ##t
  \clef "bass"
  c1
  \clef "treble"
  c1
  \revert Staff.Clef.full-size-change
  \clef "bass"
  c1
  \clef "treble"
  c1
```
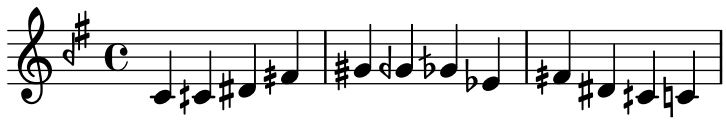
```
}
```



Ejemplo de *makam*

El «Makam» es un tipo de melodía de Turquía que utiliza alteraciones microtonales de 1/9 de tono.

Consulte el archivo de inicio `ly/makam.ly` para ver detalles de los nombres de las notas y las alteraciones.

```
\include "makam.ly"

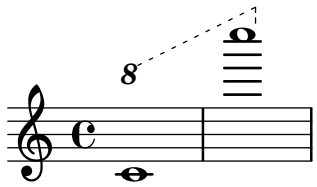
\relative c' {
  \set Staff.keyAlterations = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Modificación de la inclinación de la línea de octava

Es posible cambiar la inclinación de la línea de octava alta o baja.

```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0)
                (attach-dir . ,LEFT)
                (padding . 0)
                (stencil-align-dir-y . ,CENTER)))
       (right . ((Y . 5.0) ; Change the number here
                 (padding . 0)
                 (attach-dir . ,RIGHT)
                 (text . ,(make-draw-dashed-line-markup
                           (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
     #ly:horizontal-line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
     #ly:horizontal-line-spanner::calc-right-bound-info
  \ottava 1
  c1
  c'''1
}
```



Armaduras de tonalidad no tradicionales

La muy utilizada instrucción \key establece la propiedad keyAlterations dentro del contexto Staff.

Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

```
\set Staff.keyAlterations =
  #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)
```
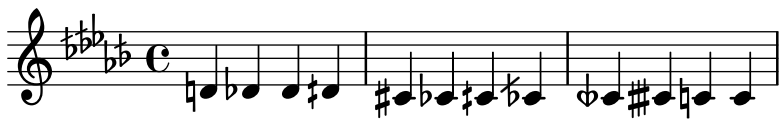
donde, para cada elemento dentro de la lista, *octava* especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), *paso* especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y *alteración* es SHARP, FLAT o DOUBLE-SHARP, etc., precedido de una coma.

De forma alternativa, podemos usar el formato más conciso (*paso . alteración*) para cada elemento de la lista si se usa la misma alteración en todas las octavas.

Para escalas microtonales en las que un "sostenido" no son 100 cents, *alteración* se refiere a la proporción de un tono entero de 200 cents.

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
                                 (6 . ,SEMI-FLAT))
  % \set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Números como notas de notación fácil

Las cabezas de nota de notación fácil utilizan la propiedad note-names del objeto NoteHead para determinar lo que aparece dentro de la cabeza. Mediante la sobreescritura de esta propiedad, es posible imprimir números que representen el grado de la escala.

Se puede crear un grabador simple que haga esto para la cabeza de cada nota que ve.

```
#(define Ez_numbers_engraver
   (make-engraver
    (acknowledgers
     ((note-head-interface engraver grob source-engraver)
      (let* ((context (ly:translator-context engraver))
             (tonic-pitch (ly:context-property context 'tonic))
             (tonic-name (ly:pitch-notename tonic-pitch))
             (grob-pitch
              (ly:event-property (event-cause grob) 'pitch))
             (grob-name (ly:pitch-notename grob-pitch))
             (delta (modulo (- grob-name tonic-name) 7))
             (note-names
              (make-vector 7 (number->string (1+ delta)))))
        (ly:grob-set-property! grob 'note-names note-names))))))

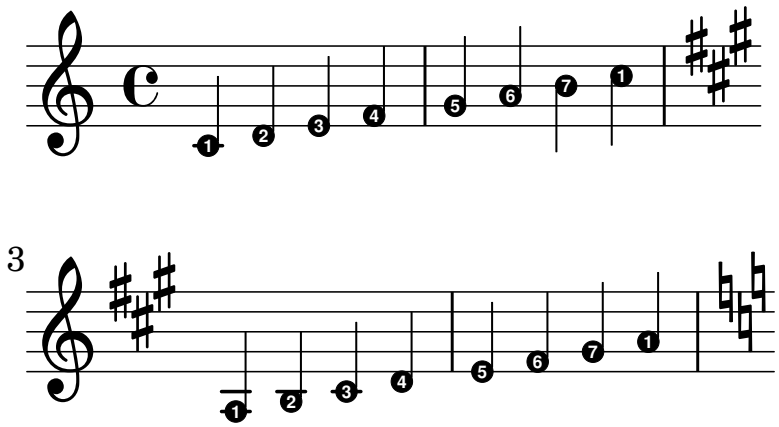
#(set-global-staff-size 30)

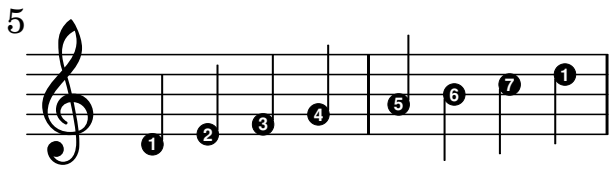
\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

  \key a \major
  a,4 b cis d
  e4 fis gis a \break

  \key d \dorian
  d,4 e f g
  a4 b c d
}
```





Plantilla de orquesta, coro y piano

Esta plantilla muestra el uso de contextos `StaffGroup` y `GrandStaff` anidados para sub-agrupar instrumentos del mismo tipo, y una forma de usar `\transpose` de manera que unas variables contengan la música para instrumentos transpositores en afinación de concierto.

```
#(set-global-staff-size 17)

\paper {
  indent = 3.0\cm  % add space for instrumentName
  short-indent = 1.5\cm  % add less space for shortInstrumentName
}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.
clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns
hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }
sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }
altoILyrics = \sopranoLyrics
altoIIMusic = \relative c' { \key g \major g'1 b }
altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }
tenorLyrics = \sopranoLyrics

pianoRHMusic = \relative c { \key g \major g''1 b }
pianoLHMusic = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }
violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }
```

```
celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\book {
  \score {
    <<
      \new StaffGroup = "StaffGroup_woodwinds" <<
        \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
          \fluteMusic

        \new Staff = "Staff_clarinet" \with {
          instrumentName = \markup { \concat { "Clarinet in B" \flat } }
        }
          % Declare that written Middle C in the music
          % to follow sounds a concert B flat, for
          % output using sounded pitches such as MIDI.
          %\transposition bes

          % Print music for a B-flat clarinet
          \transpose bes c' \clarinetMusic
      >>

      \new StaffGroup = "StaffGroup_brass" <<
        \new Staff = "Staff_hornI" \with {
          instrumentName = "Horn in F"
        }
          % \transposition f
          \transpose f c' \hornMusic

        \new Staff = "Staff_trumpet" \with {
          instrumentName = "Trumpet in  C"
        }
          \trumpetMusic
      >>

      \new RhythmicStaff = "RhythmicStaff_percussion" \with {
        instrumentName = "Percussion"
      }
        \percussionMusic

      \new PianoStaff \with {
        instrumentName = "Piano"
      } <<
        \new Staff { \pianoRHMusic }
        \new Staff { \pianoLHMusic }
      >>

      \new ChoirStaff = "ChoirStaff_choir" <<
        \new Staff = "Staff_soprano" \with {
          instrumentName = "Soprano"
        }
```

```
    \new Voice = "soprano" \sopranoMusic
    \new Lyrics \lyricsto "soprano" { \sopranoLyrics }

  \new GrandStaff = "GrandStaff_altos" \with {
    \accepts Lyrics
  } <<
    \new Staff = "Staff_altoI" \with {
      instrumentName = "Alto I"
    }
      \new Voice = "altoI"
      \altoIMusic
      \new Lyrics \lyricsto "altoI" { \altoILyrics }
    \new Staff = "Staff_altoII" \with {
      instrumentName = "Alto II"
    }
      \new Voice = "altoII"
      \altoIIMusic
      \new Lyrics \lyricsto "altoII" { \altoIILyrics }
  >>

  \new Staff = "Staff_tenor" \with {
    instrumentName = "Tenor"
  }
    \new Voice = "tenor" \tenorMusic
    \new Lyrics \lyricsto "tenor" { \tenorLyrics }
>>

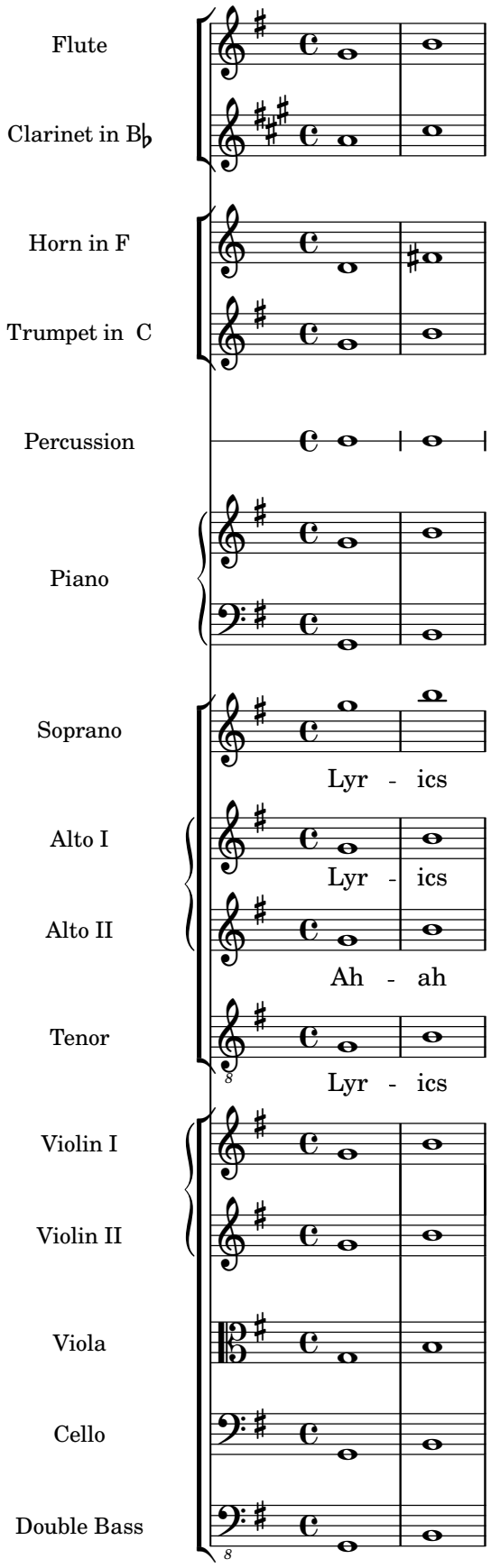
\new StaffGroup = "StaffGroup_strings" <<
  \new GrandStaff = "GrandStaff_violins" <<
    \new Staff = "Staff_violinI" \with {
      instrumentName = "Violin I"
    }
      \violinIMusic
    \new Staff = "Staff_violinII" \with {
      instrumentName = "Violin II"
    }
      \violinIIMusic
  >>

  \new Staff = "Staff_viola" \with {
    instrumentName = "Viola"
  }
    \violaMusic

  \new Staff = "Staff_cello" \with {
    instrumentName = "Cello"
  }
    \celloMusic

  \new Staff = "Staff_bass" \with {
    instrumentName = "Double Bass"
  }
```
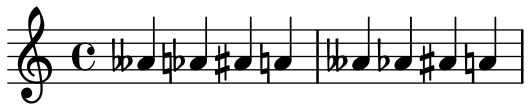
```
            \bassMusic
        >>
      >>
  }
}
```


Flute
Clarinet in B♭
Horn in F
Trumpet in C
Percussion
Piano
Soprano
Lyr - ics
Alto I
Lyr - ics
Alto II
Ah - ah
Tenor
Lyr - ics
Violin I
Violin II
Viola
Cello
Double Bass

Evitar que se añadan becuadros adicionales automáticamente

Según las reglas tradicionales de composición tipográfica, se imprime un becuadro antes de un sostenido o un bemol cuando se tiene que cancelar un doble sostenido o un doble bemol anterior en la misma nota. Para modificar este comportamiento a la práctica actual, establezca el valor de la propiedad `extraNatural` a `#f` (falso) dentro del contexto de `Staff`.
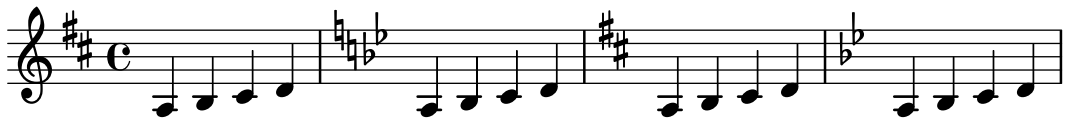
```
\relative c'' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Evitar que se impriman becuadros cuando cambia la armadura

Cuando cambia la armadura de la tonalidad, se imprimen becuadros automáticamente para cancelar las alteraciones de las armaduras anteriores. Esto se puede evitar estableciendo la propiedad `printKeyCancellation` al valor "#f" dentro del contexto `Staff`.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Citar otra voz con transposición
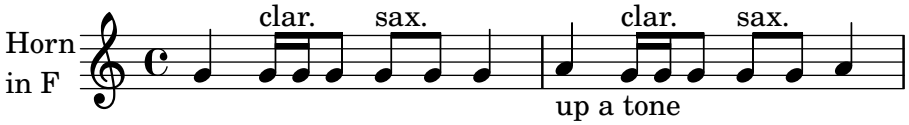
Los pasajes citados tienen en cuenta la transposición de la fuente tanto como la del destino. En este ejemplo, todos los instrumentos interpretan una nota con el sonido del Do central; el destino de un instrumento transpositor en Fa. La parte de destino se puede transportar utilizando `\transpose`. En este caso las notas citadas permanecen sin cambios.

```
\addQuote clarinet {
  \transposition bes
  \*8 { d'16 d' d'8 }
}

\addQuote sax {
  \transposition es'
  \*16 a8
}
```

```
quoteTest = {
  % french horn
  \transposition f
  g'4
  << \quoteDuring "clarinet" { \skip 4 } s4^"clar." >>
  << \quoteDuring "sax" { \skip 4 } s4^"sax." >>
  g'4
}

{
  \new Staff \with {
    instrumentName = \markup { \column { Horn "in F" } }
  }
  \quoteTest
  \transpose c' d' << \quoteTest s4_"up a tone" >>
}
```



Separar las cancelaciones de tonalidad de los cambios de armadura

De forma predeterminada, las alteraciones accidentales que se usan para las cancelaciones en las armaduras se colocan adyacentes a las que se usan para los cambios de tonalidad. Este comportamiento se puede cambiar sobreescribiendo la propiedad `break-align-orders` del objeto gráfico `BreakAlignment`.

Si busca la definición de `break-alignment-interface` en la Referencia de funcionamiento interno (`https://lilypond.org/doc/v2.24/Documentation/internals/break_002dalignment_002dinterface`) de LilyPond, obtendrá la siguiente lista de los símbolos de break-align disponibles:

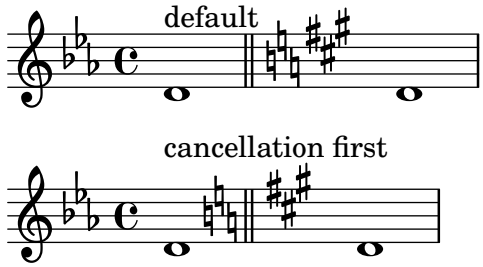
```
ambitus
breathing-sign
clef
cue-clef
cue-end-clef
custos
key-cancellation
key-signature
left-edge
signum-repetitionis
staff-bar
staff-ellipsis
time-signature
```

A partir de esta lista, vemos que lo que queremos hacer es trasladar la `key-cancellation` de forma que esté antes del `staff-bar`. Esto se hace con la función `\breakAlignInsert`.

```
music = { \key es \major d'1 \bar "||"
          \key a \major d'1 }

{ <>^\markup "default"
  \music }

{ <>^\markup "cancellation first"
  \breakAlignInsert key-cancellation before staff-bar
  \music }
```



Transposing pitches with minimum accidentals ("smart" transpose)

This example uses some Scheme code to enforce enharmonic modifications for notes in order to have the minimum number of accidentals. In this case, the following rules apply:

- double accidentals should be removed
- b sharp → c
- e sharp → f
- c flat → b
- f flat → e

In this manner, the most natural enharmonic notes are chosen.

```
#(define (naturalize-pitch p)
   (let ((o (ly:pitch-octave p))
         ;; `ly:pitch-alteration` returns quarter tone steps.
         (a (* 4 (ly:pitch-alteration p)))
         (n (ly:pitch-notename p)))
     (cond
      ((and (> a 1)
            (or (eqv? n 6) (eqv? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1)
            (or (eqv? n 0) (eqv? n 3)))
       (set! a (+ a 2))
       (set! n (- n 1))))
     (cond
      ((> a 2)
       (set! a (- a 4))
       (set! n (+ n 1)))
      ((< a -2)
       (set! a (+ a 4))
       (set! n (- n 1))))
```
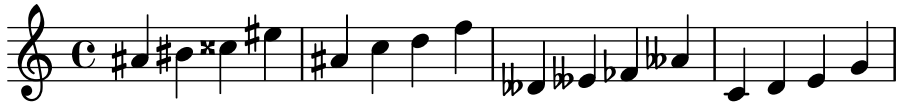
```
      (when (< n 0)
        (set! o (- o 1))
        (set! n (+ n 7)))
      (when (> n 6)
        (set! o (+ o 1))
        (set! n (- n 7)))
      (ly:make-pitch o n (/ a 4))))

#(define (naturalize music)
   (let ((es (ly:music-property music 'elements))
         (e (ly:music-property music 'element))
         (p (ly:music-property music 'pitch)))
     (when (pair? es)
       (ly:music-set-property! music 'elements
                               (map naturalize es)))
     (when (ly:music? e)
       (ly:music-set-property! music 'element
                               (naturalize e)))
     (when (ly:pitch? p)
       (set! p (naturalize-pitch p))
       (ly:music-set-property! music 'pitch p))
     music))

naturalizeMusic =
#(define-music-function (m) (ly:music?)
   (naturalize m))

music = \relative c' { c4 d e g }

\new Staff {
  \transpose c ais { \music }
  \naturalizeMusic \transpose c ais { \music }
  \transpose c deses { \music }
  \naturalizeMusic \transpose c deses { \music }
}
```



Ejemplo de *makam* en la música turca

Esta plantilla usa el comienzo de un *Saz Semai* turco bien conocido que es familiar dentro del repertorio, para ilustrar algunos de los elementos de la notación musical de Turquía.

```
#(set-default-paper-size "a6" 'landscape)

\include "turkish-makam.ly"

\header {
  title = "Hüseyni Saz Semaisi"
  composer = "Lavtacı Andon"
  tagline = ##f
```

```
}

\relative {
  \set Staff.extraNatural = ##f
  \set Staff.autoBeaming = ##f

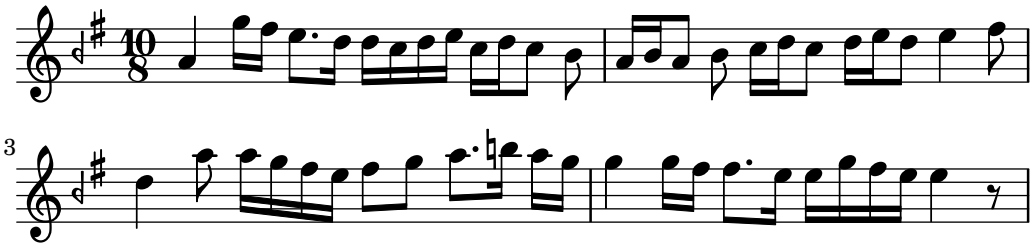
  \key a \huseyni
  \time 10/8

  a'4 g'16[ fb] e8.[ d16] d[ c d e] c[ d c8] bfc |
  a16[ bfc a8] bfc c16[ d c8] d16[ e d8] e4 fb8 |
  d4 a'8 a16[ g fb e] fb8[ g] a8.[ b16] a16[ g] |
  g4 g16[ fb] fb8.[ e16] e[ g fb e] e4 r8 |
}

\layout {
  indent = 0
}
```

Hüseyni Saz Semaisi

Lavtacı Andon



Trucaje de las propiedades de clave

La modificación del glifo de la clave, su posición o su octavación, no cambian 'per se' la posición de las siguientes notas del pentagrama. Para conseguir armaduras de tonalidad sobre las líneas del pentagrama adecuadas, también debe especificarse `middleCClefPosition`, con valores positivos o negativos que mueven el "Do central" hacia arriba o hacia abajo, respectivamente, en relación con la línea central del pentagrama (usualmente la tercera).

Por ejemplo, la instrucción `\clef "treble_8"` equivale a un ajuste de las propiedades de contexto `clefGlyph`, `clefPosition` (que controla la posición vertical de la clave sobre el pentagrama), `middleCPosition` y `clefTransposition`. Se imprime una clave cada vez que se modifica cualquiera de las propiedades excepto `middleCPosition`.

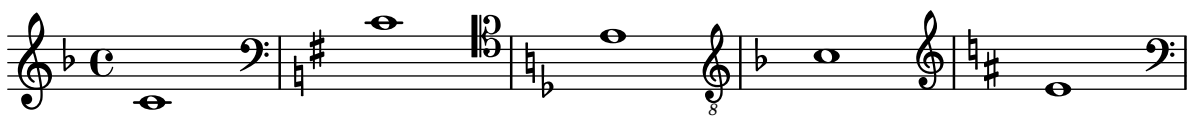
Los siguientes ejemplos muestran las posibilidades cuando se ajustan estas propiedades manualmente. En la primera línea, los cambios manuales preservan el posicionamiento relativo estándar de las claves y las notas, pero no lo hacen en la segunda línea.
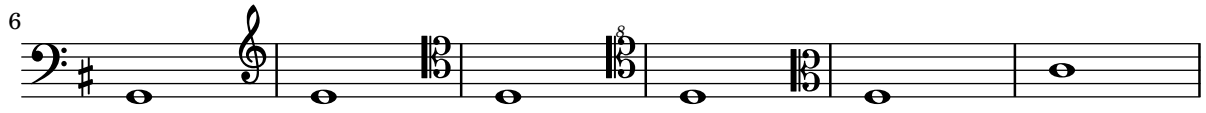
```
{
  % The default treble clef.
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  \set Staff.middleCPosition = 6
```

```
  \set Staff.middleCClefPosition = 6
  \key g \major
  c'1
  % The baritone clef.
  \set Staff.clefGlyph = "clefs.C"
  \set Staff.clefPosition = 4
  \set Staff.middleCPosition = 4
  \set Staff.middleCClefPosition = 4
  \key f \major
  c'1
  % The standard choral tenor clef.
  \set Staff.clefGlyph = "clefs.G"
  \set Staff.clefPosition = -2
  \set Staff.clefTransposition = -7
  \set Staff.middleCPosition = 1
  \set Staff.middleCClefPosition = 1
  \key f \major
  c'1
  % A non-standard clef.
  \set Staff.clefPosition = 0
  \set Staff.clefTransposition = 0
  \set Staff.middleCPosition = -4
  \set Staff.middleCClefPosition = -4
  \key g \major
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
  % and clefs.
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  c'1
  \set Staff.clefGlyph = "clefs.G"
  c'1
  \set Staff.clefGlyph = "clefs.C"
  c'1
  \set Staff.clefTransposition = 7
  c'1
  \set Staff.clefTransposition = 0
  \set Staff.clefPosition = 0
  c'1

  % Return to the normal clef.
  \set Staff.middleCPosition = 0
  c'1
}
```





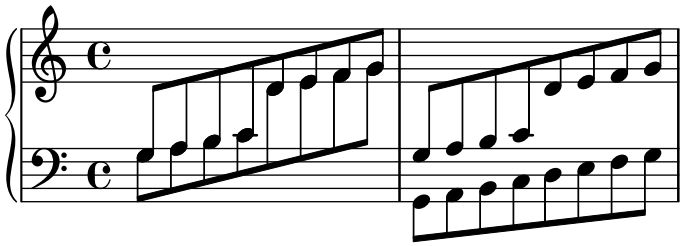
Using \autoChange with more than one voice

Here is a demonstration of how to use \autoChange with more than one voice.

```
\score {
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,8 a b c d e f g
          }
        }

        \new Voice {
          \voiceTwo
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,,8 a b c d e f g
          }
        }
      >>
    }

    \new Staff = "down" {
      \clef bass
    }
  >>
}
```



2 Rhythms

See also Sección “Rhythms” en *Notation Reference*.
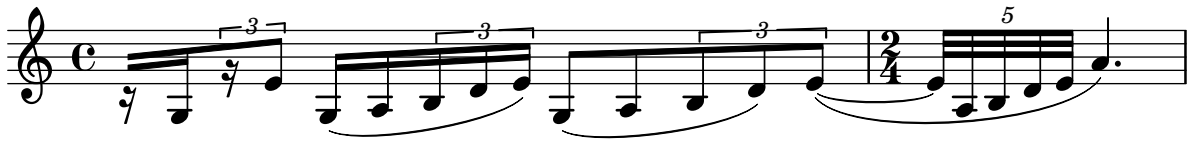
Adding beams, slurs, ties, etc., when using tuplet and non-tuplet rhythms

LilyPond primarily uses postfix syntax for inputting parentheses, brackets, etc., which might feel unintuitive for novices.

For example, when entering a manual beam, the left square bracket has to be placed *after* the starting note and its duration, not before. Similarly, the right square bracket should directly follow the note which is to be at the end of the requested beaming, even if this note happens to be inside a tuplet section.

This snippet demonstrates how to combine manual beaming, manual slurs, ties, and phrasing slurs with tuplet sections (enclosed within curly braces).

```
{
  r16[ g16 \tuplet 3/2 { r16 e'8] }
  g16( a \tuplet 3/2 { b d' e') }
  g8[( a \tuplet 3/2 { b d') e']\( ~ }
  \time 2/4
  \tuplet 5/4 { e'32 a b d' e' } a'4.\)
}
```



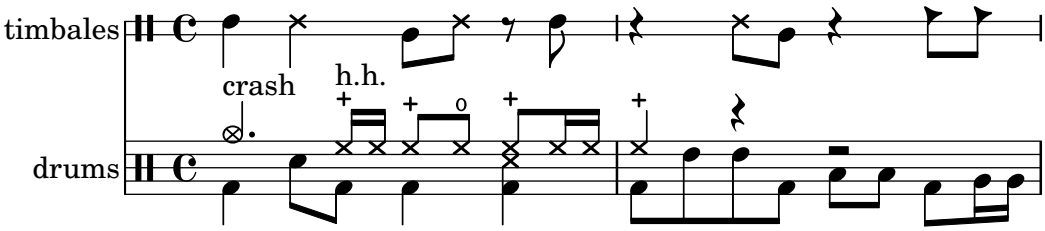
Escritura de partes de percusión

Mediante la utilización de las potentes herramientas preconfiguradas como la función `\drummode` y el contexto `DrumStaff`, la introducción de partes para percusión es muy fácil: las percusiones se sitúan en sus propias posiciones de pentagrama (con una clave especial) y tienen las cabezas correspondientes al instrumento. Es posible añadir un símbolo adicional a la percusión o reducir el número de líneas.

```
drh = \drummode {
        cymc4.^"crash" hhc16^"h.h." hh hhc8 hho hhc8 hh16 hh
        hhc4 r4 r2
      }
drl = \drummode {
        bd4 sn8 bd bd4 << bd ss >>
        bd8 tommh tommh bd toml toml bd tomfh16 tomfh
      }
timb = \drummode {
         timh4 ssh timl8 ssh r timh r4
         ssh8 timl r4 cb8 cb
       }

\score {
  <<
    \new DrumStaff \with {
      instrumentName = "timbales"
```

```
      drumStyleTable = #timbales-style
      \override StaffSymbol.line-count = #2
      \override BarLine.bar-extent = #'(-1 . 1)
    }
    <<
      \timb
    >>
    \new DrumStaff \with { instrumentName = "drums" }
    <<
      \new DrumVoice { \stemUp \drh }
      \new DrumVoice { \stemDown \drl }
    >>
  >>
  \layout { }
  \midi { \tempo 4 = 120 }
}
```
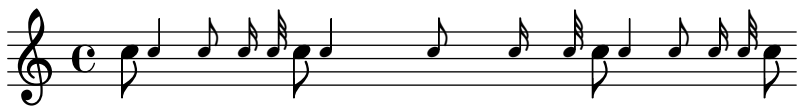


Ajuste del espaciado de las notas de adorno

Se puede ajustar la separación entre las notas de adorno utilizando la propiedad `spacing-increment` de `Score.GraceSpacing`.

```
graceNotes = {
  \grace { c4 c8 c16 c32 }
  c8
}

\relative c'' {
  c8
  \graceNotes
  \override Score.GraceSpacing.spacing-increment = #2.0
  \graceNotes
  \revert Score.GraceSpacing.spacing-increment
  \graceNotes
}
```



Alineación de los números de compás

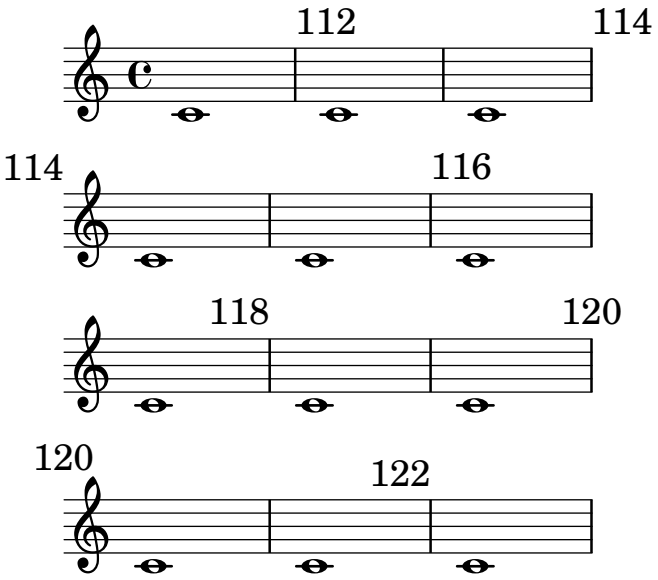
La alineación predeterminada de los números de compás depende de su posición: al comienzo de un pentagrama, los números se alinean por la derecha; en cualquier otra posición, se alinean por la izquierda. Usando la función de Scheme `break-alignment-list`, se puede cambiar esto; los tres argumentos de la función son la alineación para las posiciones al final de la línea, en la parte media de la línea y al principio de la línea (en este orden).

```
\relative c' {
  \set Score.currentBarNumber = 111
  \override Score.BarNumber.break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = 2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)

  c1 | c1 | c1 | \break
  c1 | c1 | c1 | \break

  \override Score.BarNumber.self-alignment-X =
    #(break-alignment-list CENTER RIGHT CENTER)
  c1 | c1 | c1 | \break
  c1 | c1 | c1 |
}
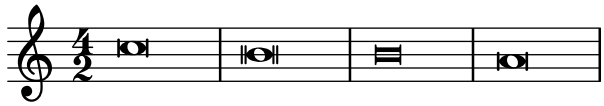
\paper {
  line-width = 70\mm
}
```



Formas alternativas de la figura breve

Las figuras de breve también están disponibles con dos líneas verticales a los lados de la cabeza en lugar de una sola línea y la forma en estilo barroco.

```
\relative c'' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}
```



Apoyatura o nota de adorno antes de la línea divisoria
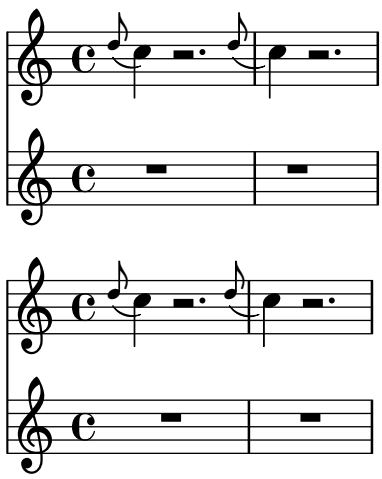
De manera predeterminada, las apoyaturas y las notas de adorno que están en el primer pulso de un compás se imprimen después de la línea divisoria. En pentagramas sueltos, una posible solución para que se impriman antes de la línea divisoria es añadir una línea invisible y después la línea visible.

Sin embargo, en sistemas de varios pentagramas, añadir una barra de compás invisible distorsiona el posicionamiento de los silencios de compás completo en las otras pautas; ya no estarían centrados sino desplazados ligeramente hacia la izquierda. Una solución mejor para tales situaciones es usar la instrucción `\afterGrace` command junto a la configuración apropiada de `afterGraceFraction`.

```
<<
  {
    \appoggiatura d''8 c''4 r2. |
    \appoggiatura { \bar "" d''8 \bar "|" } |
    c''4 r2.
  }
  { R1 | R1 }
>>

afterGraceFraction = 15/16

<<
  {
    \appoggiatura d''8 c''4 \afterGrace r2. d''8( |
    c''4) r2.
  }
  { R1 | R1 }
>>
```

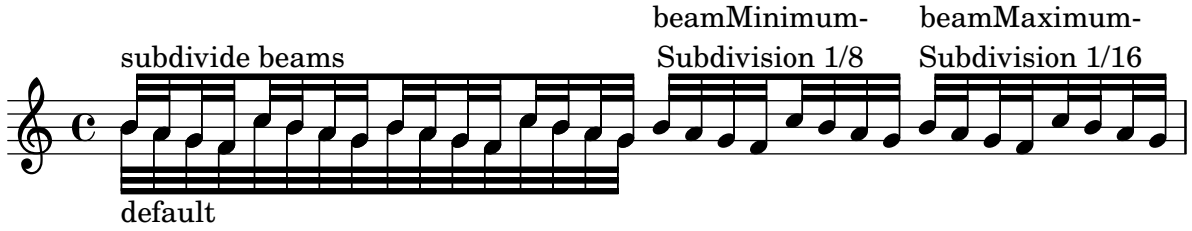


Subdivisiones de barra automáticas

Se pueden subdividir las barras automáticamente. Estableciendo la propiedad `subdivideBeams`, las barras se subdividen en siempre que sea posible. Se pueden limitar los intervalos

y la profundidad de la subdivisión con las propiedades `beamMinimumSubdivision` y `beamMaximumSubdivision`, respectivamente.

```
\new Staff {
  \relative c'' {
    <<
      {
        \voiceOne
        \set subdivideBeams = ##t
        b32[^"subdivide beams" a g f c' b a g
        b32 a g f c' b a g]
      }
      \new Voice {
        \voiceTwo
        b32_"default"[ a g f c' b a g
        b32 a g f c' b a g]
      }
    >>
    \oneVoice
    \once \set beamMinimumSubdivision = #1/8
    b32^\markup \center-column { "beamMinimum-"
                                 "Subdivision 1/8" } [ a g f c' b a g]
    \once \set beamMaximumSubdivision = #1/16
    b32^\markup \center-column { "beamMaximum-"
                                 "Subdivision 1/16" } [ a g f c' b a g]
  }
}
```



Modificar duraciones automáticamente
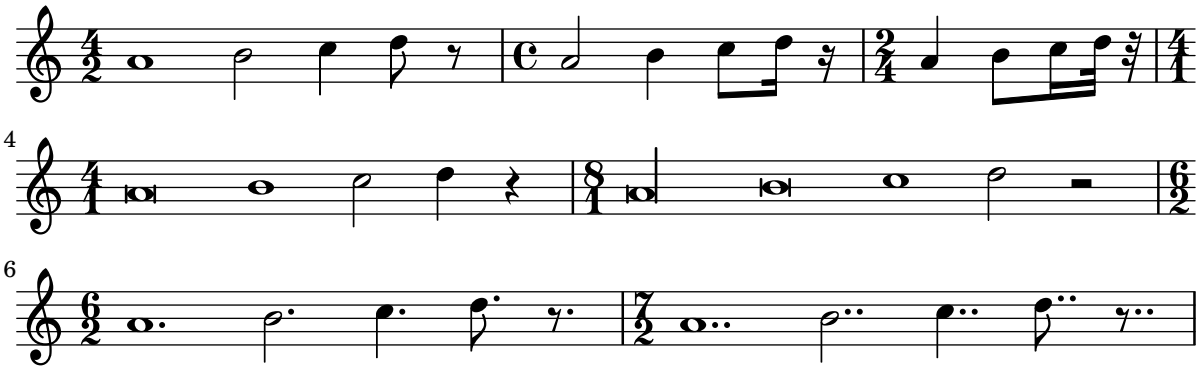
Se puede usar `shiftDurations` para cambiar la longitud de las notas de una pieza musical.

Toma dos argumentos: el factor de escalado como una potencia de dos, y el número de puntillos que añadir como un entero positivo.

```
music = \relative c'' { a1 b2 c4 d8 r }

{
  \time 4/2
  \music
  \time 4/4
  \shiftDurations 1 0 \music
  \time 2/4
  \shiftDurations 2 0 \music
  \time 4/1
  \shiftDurations -1 0 \music
  \time 8/1
  \shiftDurations -2 0 \music
```
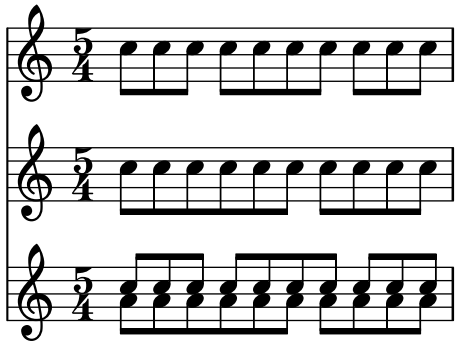
```
  \time 6/2
  \shiftDurations 0 1 \music
  \time 7/2
  \shiftDurations 0 2 \music
}
```



Finales de barra en el contexto Score

Las reglas de final de barra especificadas en el contexto `Score` se aplican a todos los pentagramas, pero se pueden modificar tanto en los niveles de `Staff` como de `Voice`:

```
\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.beatBase = #1/8
  \set Score.beatStructure = 3,4,3
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = 6,4
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
        % Modify beaming for this voice only
        \new Voice {
          \voiceTwo
          \set Voice.beatStructure = 6,4
          a8 a a a a a a a a
        }
      >>
    }
  >>
}
```



Barras recortadas

Las barras recortadas al principio y al final de las barras, junto con las barras unidas a notas sueltas que parecen corchetes planos, son posibles con una combinación de `stemLeftBeamCount`, `stemRightBeamCount` e indicadores de barra `[]` pareados.

Para imitar los corchetes planos hacia la derecha sobre notas sueltas, indicadores de barra `[]` pareados y fije el valor de `stemLeftBeamCount` a cero. En cambio, para imitar corchetes planos hacia la izquierda sobre notas sueltas, fije el valor de `stemRightBeamCount` a cero (línea uno).

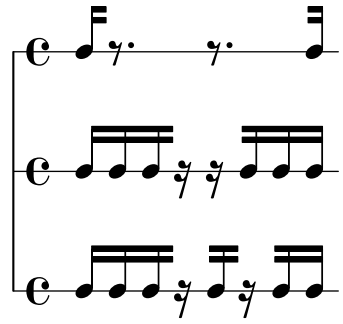
Para las barras recortadas hacia la derecha al final de una serie de noras barradas, fije el valor de `stemRightBeamCount` a un valor positivo. Por su parte, para barras recortadas hacia la derecha al comienzo de una serie de notas barradas, fije `stemLeftBeamCount` (línea dos).

A veces tiene sentido que una nota suelta entre dos silencios lleve dos barras recortadas, hacia la derecha y hacia la izquierda. Esto se hace escribiendo solamente indicadores de barrado `[]` pareados (línea tres).

Observe que `\set stemLeftBeamCount` siempre equivale a `\once \set`. En otras palabras, los ajustes para el número de barras no son "persistentes", o sea que la pareja de barras recortadas unidas a la semicorchea aislada del último ejemplo no tiene nada que ver con la instrucción `\set` de la barra anterior.

```
\score {
  <<
    \new RhythmicStaff {
      \set stemLeftBeamCount = 0
      c16[] r8.
      r8.
      \set stemRightBeamCount = 0
      16[]
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r r
      \set stemLeftBeamCount = 2
      16 16 16
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r16
      16[] r16
      \set stemLeftBeamCount = 2
```

```
      16 16
    }
  >>
}
```



Barras que atraviesan saltos de línea

Normalmente LilyPond rehúsa insertar un salto de línea automático si las barra atraviesa la línea divisoria. Se puede cambiar este comportamiento por medio del establecimiento de la propiedad `Beam.breakable` al valor `#t`.

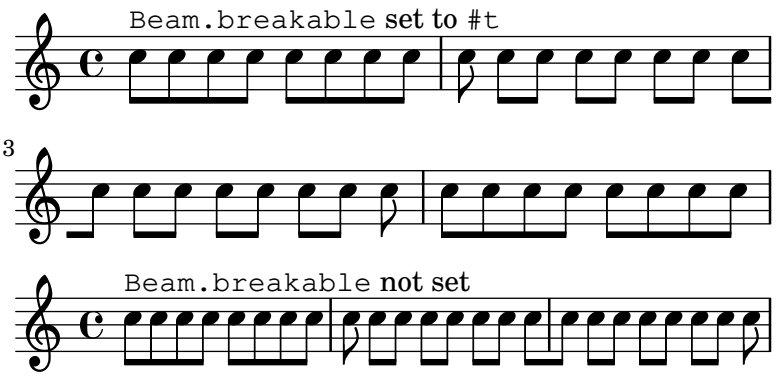
Esta propiedad no afecta a los saltos manuales insertados con instrucciones como `\break`.
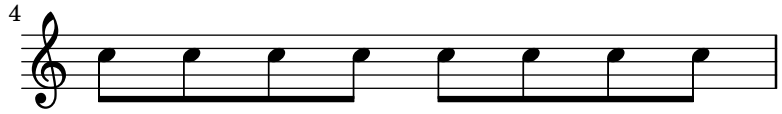
```
music = {
  \*8 c8
  c8 \*7 { c[ c] }  c
  \*8 c8
}

\relative c'' {
  <>^\markup { \typewriter Beam.breakable set to \typewriter "#t" }
  \override Beam.breakable = ##t
  \music
}

\relative c'' {
  <>^\markup { \typewriter Beam.breakable not set }
  \music
}

\paper {
  line-width = 100\mm
}
```
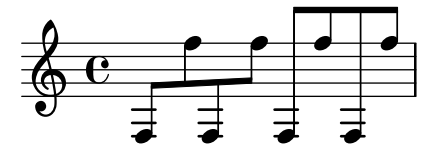




Cambiar el salto de las barras en ángulo

Se insertan automáticamente barras en ángulo cuando se detecta un intervalo muy grande entre las notas. Se puede hacer un ajuste fino de este comportamiento a través de la propiedad `auto-knee-gap`. Se traza una barra doblada si el salto es mayor que el valor de `auto-knee-gap` más el ancho del objeto barra (que depende de la duración de las notas y de la inclinación de la barra). De forma predeterminada `auto-knee-gap` está establecido a 5.5 espacios de pentagrama.

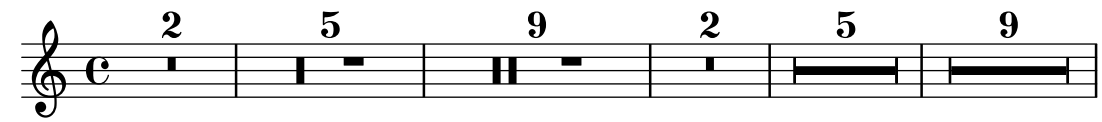
```
{
  f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



Cambiar la forma de los silencios multicompás

Si hay diez compases de silencio o menos, se imprime en el pentagrama una serie de silencios de breve y longa (conocidos en alemán como "Kirchenpausen", «silencios eclesiásticos»); en caso contrario se muestra una barra gruesa y larga. Este valor predeterminado de diez se puede cambiar sobreescribiendo la propiedad `expand-limit`:
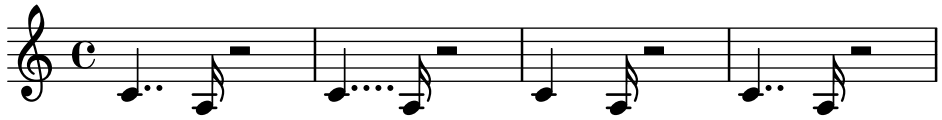
```
\relative c'' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = 3
    R1*2 | R1*5 | R1*9
  }
}
```



Modificar el número de puntillos de una nota

La cantidad de puntillos de una nota se puede sobreescribir mediante el establecimiento de la propiedad `dot-count` del objeto gráfico `Dots`.
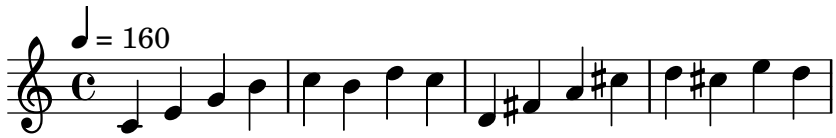
```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = 4
  c4.. a16 r2 |
  \override Dots.dot-count = 0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```



Cambiar el tempo sin indicación metronómica

Para cambiar el tempo en la salida MIDI sin imprimir nada, hacemos invisible la indicación metronómica:

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```

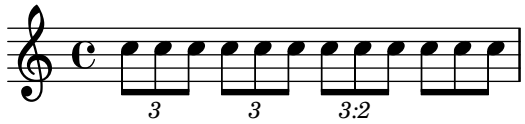


Cambiar el número del grupo especial

De forma predeterminada solo se imprime sobre el corchete de grupo el numerador del grupo especial, o sea, el numerador del argumento de la instrucción `\tuplet`.

De forma alternativa, se puede imprimr un quebrado en la forma *numerador*:*denominador* del número del grupo, o eliminar el número.

```
\relative c'' {
  \tuplet 3/2 { c8 c c }
  \tuplet 3/2 { c8 c c }
  \override TupletNumber.text = #tuplet-number::calc-fraction-text
  \tuplet 3/2 { c8 c c }
  \omit TupletNumber
  \tuplet 3/2 { c8 c c }
}
```



Modificar el compás dentro de una sección polimétrica utilizando `\scaleDurations`

Polirritmia flexible con compases no alineados

Para poder crear explícitamente contextos con compases independientes, suprima el traductor `Timing_translator` del contexto `Score` y defina un contexto `TimingStaffGroup` que contenga

el traductor Timing_translator. Esto hace que Timing sea un alias de TimingStaffGroup, con la mira puesta en las instrucciones \time hacia el TimingStaffGroup contenedor.

A diferencia de la instrucción incorporada \enablePerStaffTiming de LilyPond, este enfoque requiere la creación explícita de contextos TimingStaffGroup; a cambio, permite crear varios contextos Staff que de forma conjunta siguen al compás definido en su TimingStaffGroup contenedor.

Indicaciones de compás escalados localmente

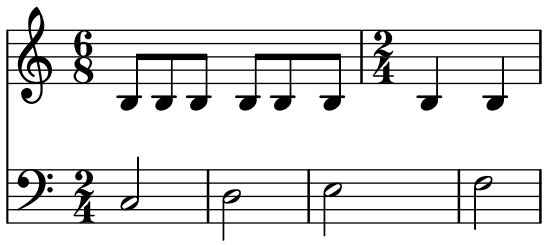
Use la instrucción no escalable \time para establecer un compás de la longitud deseada en Timing, más conocido como TimingStaffGroup. En este fragmento de código, todos los pentagramas que están por debajo de TimingStaffGroup usan una indicación de compás escalada, de forma que cualquier compás que establezcamos con la longitud deseada es tan bueno como cualquier otro. Si hubiese un contexto contenedor que no usara una indicación de compás escalada, la elección del compás que se establezca en Timing tendría importancia en dicho contexto.

Use la instrucción \polymetric \time para establecer propiedades métricas escalablesen los contextos que están por debajo de Timing, y use la instrucción \scaleDurations para escalar tanto el compás local como las notas que han de llenar el compás.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \accepts TimingStaffGroup
  }
  \context {
    \StaffGroup
    \name TimingStaffGroup
    \alias StaffGroup
    \consists "Timing_translator"
  }
}

<<
  \new TimingStaffGroup <<
    \new Staff {
      \scaleDurations 8/5 {
        \time 6/5 % to set measure length in Timing
        \context Staff \polymetric \time 6/8
        b8 b b b b b
        \time 4/5 % to set measure length in Timing
        \context Staff \polymetric \time 2/4
        b4 b
      }
    }
  >>
  \new TimingStaffGroup <<
    \new Staff {
      \clef bass
      \time 2/4
      c2 d e f
    }
  >>
```

>>

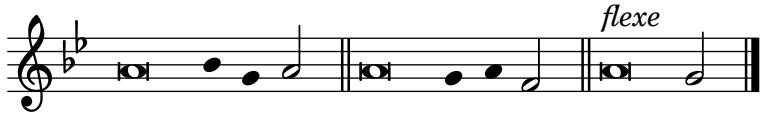


Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}
```
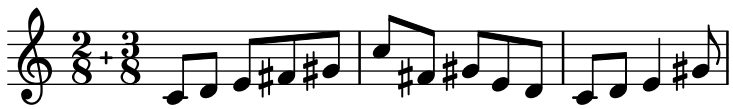


Complex time signatures

Odd time signatures (such as “5/8”) can often be played as complex time signatures (e.g. “3/8 + 2/8”), which combine two or more inequal metrics.

LilyPond can make such music quite easy to read and play, by explicitly printing the time signatures and adapting the automatic beaming behavior.

```
\relative c' {
  \time #'((2 . 8) (3 . 8))
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```



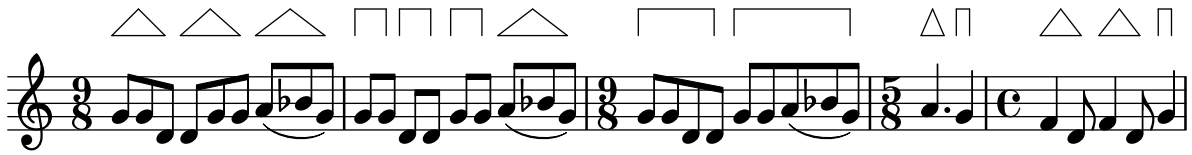
Símbolos de dirección y símbolos de agrupación de compás

Las propiedades de contexto controlan el agrupamiento de los pulsos dentro de un compás: `beatStructure` lista la longitud de cada pulso en unidades de `beatBase`. Hay establecidos valores de `beatStructure` para muchos tipos de compases en `scm/time-signature-settings.scm`. Estas propiedades se pueden cambiar o establecer con `\set`.

Como alternativa, `\time` acepta opcionalmente una estructura de pulsos para usarla de forma predeterminada. `\time` se aplica al contexto `Timing`, por lo que no restablece los valores de `beatStructure` ni de `beatBase` que se establezcan en otros contextos de nivel inferior, como `Voice`.

Si el grabador `Measure_grouping_engraver` está incluido en uno de los contextos de presentación, se imprimirán signos de agrupación de pulsos. Estos símbolos facilitan la lectura de música moderna rítmicamente compleja. En este ejemplo, el compás de 9/8 se agrupa según dos patrones distintos utilizando los dos métodos, mientras que el compás de 5/8 se agrupa de acuerdo con el ajuste predeterminado que está en `scm/time-signature-settings.scm`. Para el compás de 4/4 hay que fijar explícitamente `beatBase` a corcheas para que se muestre el patrón irregular del compás.

```
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = 2,2,2,3
    g8 g d d g g a( bes g) |
    \time 4,5 9/8
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
    \time 3,3,2 4/4
    \set Timing.beatBase = #1/8
    f4 d8 f4 d8 g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



Controlar la visibilidad de los corchetes de grupo especial

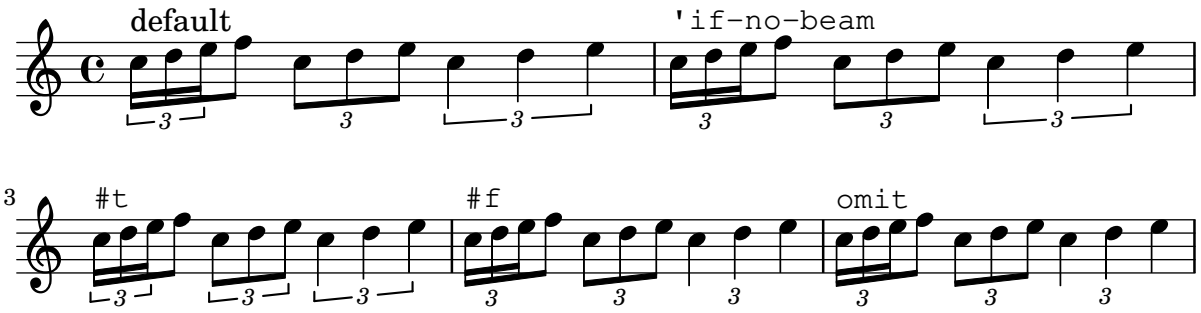
El comportamiento predeterminado de la visibilidad de los corchetes de grupo de valoración especial es imprimir el corchete a no ser que haya una barra de la misma longitud que el grupo especial.

Para controlar la visibilidad de los corchetes de grupo, establezca la propiedad `bracket-visibility` a `#t` (imprimir el corchete siempre), `#if-no-beam` (imprimir el corchete

solamente si no hay barra), o `#f` (no imprimir nunca el corchete). Este último equivale de hecho a omitir el objeto `TupletBracket` de la salida impresa.

```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e } f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

\new Voice {
  \relative c' {
    \override Score.TextMark.non-musical = ##f
    \textMark "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    \textMark \markup \typewriter "'if-no-beam" \music
    \override TupletBracket.bracket-visibility = ##t
    \textMark \markup \typewriter "#t" \music
    \override TupletBracket.bracket-visibility = ##f
    \textMark \markup \typewriter "#f" \music
    \omit TupletBracket
    \textMark \markup \typewriter "omit" \music
  }
}
```



Ejemplo de cencerro y campana

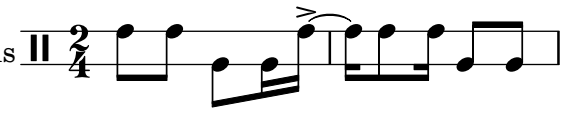
Dos campanas diferentes, escritas con ‘cb’ (cencerro) y ‘rb’ (campana, ridebell).

```
#(define mydrums '((ridebell default #f  3)
                   (cowbell  default #f -2)))

\new DrumStaff \with { instrumentName = #"Different Bells" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \set DrumStaff.clefPosition = 0.5
  \override DrumStaff.StaffSymbol.line-positions = #'(-2 3)
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)
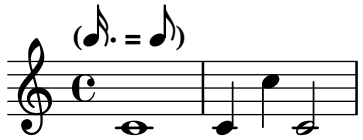
  \time 2/4
  rb8 8 cb8 16 rb16-> ~ |
  16 8 16 cb8 8 |
}
```

Different Bells 

Crear indicaciones metronómicas en modo de marcado

Se pueden crear indicaciones metronómicas nuevas en modo de marcado, pero no cambian el tempo en la salida MIDI.

```
\relative c' {
  \tempo \markup {
    \concat {
      (
      \smaller \general-align #Y #DOWN \note { 16. } #UP
      " = "
      \smaller \general-align #Y #DOWN \note { 8 } #UP
      )
    }
  }
  c1
  c4 c' c,2
}
```



Grabado manual de las ligaduras

Se puede grabar a mano una ligadura modificando la propiedad `staff-position` (un desplazamiento) del objeto gráfico `Tie`; si hay más de una ligadura en el mismo momento musical, se pueden ajustar manualmente cambiando la propiedad `tie-configuration` del objeto `TieColumn`.

El desplazamiento indica la distancia a partir de la tercera línea del pentagrama en espacios de pentagrama, y la dirección puede ser 1 (hacia arriba) o -1 (hacia abajo).

Observe que LilyPond distingue entre valores exactos e inexactos para el primer número. Si se está usando un valor exacto (p.ej., un entero o una fracción como `(/ 4 5)`), el valor sirve como una posición vertical aproximada que después se ajusta por parte de LilyPond para que la ligadura evite las líneas del pentagrama. Si se usa un valor inexacto, como un número de coma flotante, se toma como la posición vertical precisa sin más ajustes posteriores.

```
\relative c' {
  <>^"default"
  g'1 ~ g

  <>^"0"
  \once \override Tie.staff-position = 0
  g1 ~ g

  <>^"0.0"
  \once \override Tie.staff-position = 0.0
  g1 ~ g

  <>^"reset"
```

```
  \revert Tie.staff-position
  g1 ^~ g
}

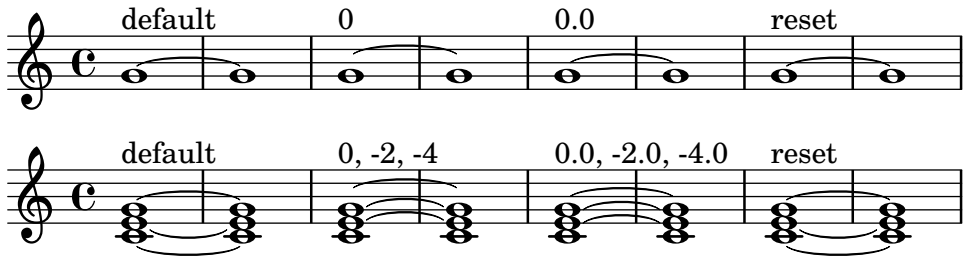
\relative c' {
  \override TextScript.outside-staff-priority = ##f
  \override TextScript.padding = 0

  <>^"default"
  <c e g>1~ <c e g>

  <>^"0, -2, -4"
  \override TieColumn.tie-configuration =
    #'((0 . 1) (-2 . 1) (-4 . 1))
  <c e g>1~ <c e g>

  <>^"0.0, -2.0, -4.0"
  \override TieColumn.tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g>1~ <c e g>

  <>^"reset"
  \override TieColumn.tie-configuration = ##f
  <c e g>1~ <c e g>
}
```
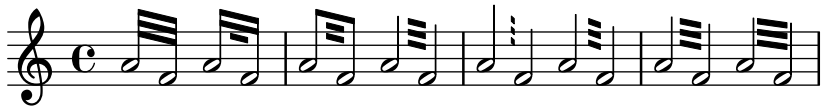


Trémolos con barras flotantes

Si la duración total de un trémolo es menor de una negra, o exactamente de una blanca, normalmente se compone tipográficamente con todas las barras de corchea tocando las plicas. Ciertos estilos de notación de partituras componen algunas de estas barras como barras flotantes centradas que no tocan las plicas de las figuras. El número de barras flotantes en este tipo de trémolo está controlado por la propiedad `gap-count` del objeto `Beam` y la separación entre las barras y las plicas se fija con la propiedad `gap`.

```
\relative c'' {
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #2
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #3
  \repeat tremolo 8 { a32 f }

  \override Beam.gap-count = #3
```

```
  \override Beam.gap = #1.33
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.67
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.33
  \repeat tremolo 8 { a32 f }
}
```

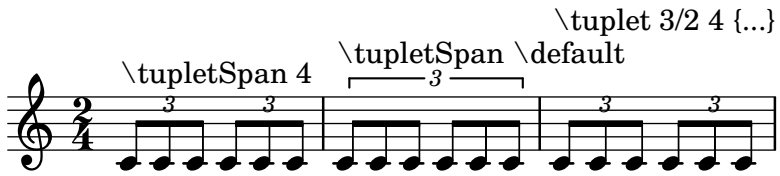


Escribir varios tresillos con una sola instrucción \tuplet

La propiedad tupletSpannerDuration establece cuánto tiempo debe durar cada uno de los tresillos contenidos dentro de los corchetes después de la instrucción \tuplet. Muchos tresillos consecutivos se pueden entonces escribir dentro de una sola instrucción \tuplet, ahorrando algo de tecleo.

Hay formas de fijar tupletSpannerDuration aparte de usar una instrucción \set. La instrucción \tupletSpan lo fija a una duración dada, o la borra cuando se especifica en su lugar una duración \default. Otra forma es usar un argumento opcional con \tuplet.

```
\relative c' {
  \time 2/4
  \tupletSpan 4
  \tuplet 3/2 { c8^"\\tupletSpan 4" c c c c c }
  \tupletSpan \default
  \tuplet 3/2 { c8^"\\tupletSpan \\default" c c c c c }
  \tuplet 3/2 4 { c8^"\\tuplet 3/2 4 {...}" c c c c c }
}
```



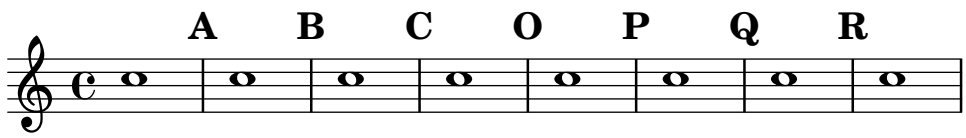
Forzar que las marcas de ensayo comiencen en una letra o número dados

Este fragmento de código muestra cómo obtener marcas de ensayo ordenadas automáticamente, pero partiendo de la letra o número deseados.

```
\relative c'' {
  \override Score.RehearsalMark.Y-offset = #3.5

  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
```

```
  c1 \mark \default
  c1
}
```



Generar corchetes personalizados

La propiedad `stencil` del grob `Flag` (el objeto gráfico corchete) se puede fijar a una función de Scheme personalizada que genere el glifo del corchete.

```
#(define-public (weight-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (log (- (ly:grob-property stem-grob 'duration-log) 2))
          (is-up? (eqv? (ly:grob-property stem-grob 'direction) UP))
          (yext (if is-up? (cons (* log -0.8) 0) (cons 0 (* log 0.8))))
          (flag-stencil (make-filled-box-stencil '(-0.4 . 0.4) yext))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stroke-stencil (if (equal? stroke-style "grace")
                              (make-line-stencil 0.2 -0.9 -0.4 0.9 -0.4)
                              empty-stencil)))
     (ly:stencil-add flag-stencil stroke-stencil)))


% Create a flag stencil by looking up the glyph from the font
#(define (inverted-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (dir (if (eqv? (ly:grob-property stem-grob 'direction) UP) "d" "u"))
          (flag (retrieve-glyph-flag "" dir "" grob))
          (line-thickness (ly:staff-symbol-line-thickness grob))
          (stem-thickness (ly:grob-property stem-grob 'thickness))
          (stem-width (* line-thickness stem-thickness))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stencil (if (null? stroke-style)
                       flag
                       (add-stroke-glyph flag stem-grob dir stroke-style "")))
          (rotated-flag (ly:stencil-rotate-absolute stencil 180 0 0)))
     (ly:stencil-translate rotated-flag (cons (- (/ stem-width 2)) 0))))

snippetexamplenotes =
{
  \autoBeamOff c'8 d'16 c'32 d'64 \acciaccatura {c'8} d'64
}

{
  \time 1/4
  <>^"Normal flags"
  \snippetexamplenotes

  <>_"Custom flag: inverted"
  \override Flag.stencil = #inverted-flag
```

```
  \snippetexamplenotes

  <>^"Custom flag: weight"
  \override Flag.stencil = #weight-flag
  \snippetexamplenotes

  <>_"Revert to normal"
  \revert Flag.stencil
  \snippetexamplenotes
}
```

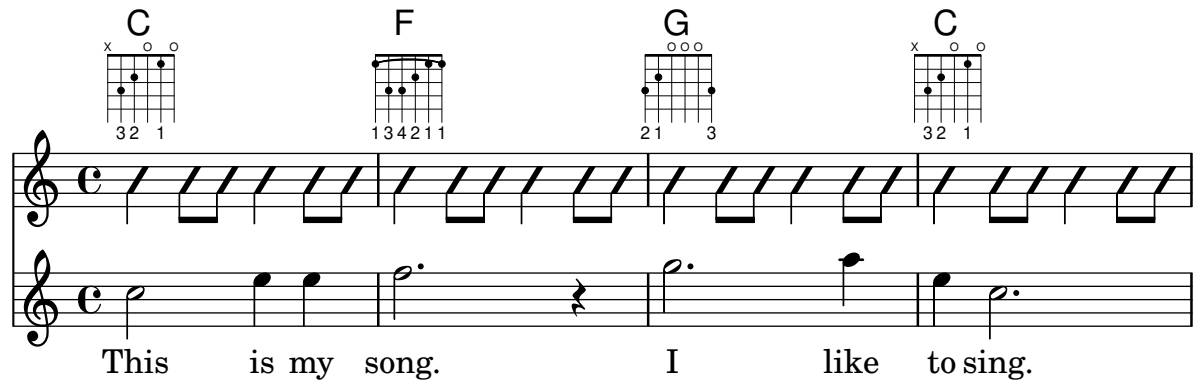


Ritmos rasgueados de guitarra

Para la música de guitarra es posible mostrar los ritmos de rasgueo, además de las notas de la melodía, acordes y diagramas de posiciones.

```
\include "predefined-guitar-fretboards.ly"

<<
  \new ChordNames \chordmode {
    c1 | f | g | c
  }
  \new FretBoards \chordmode {
    c1 | f | g | c
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
  \new Voice = "melody" \relative c'' {
    c2 e4 e4
    f2. r4
    g2. a4
    e4 c2.
  }
  \new Lyrics \lyricsto "melody" {
    This is my song.
    I like to sing.
  }
>>
```



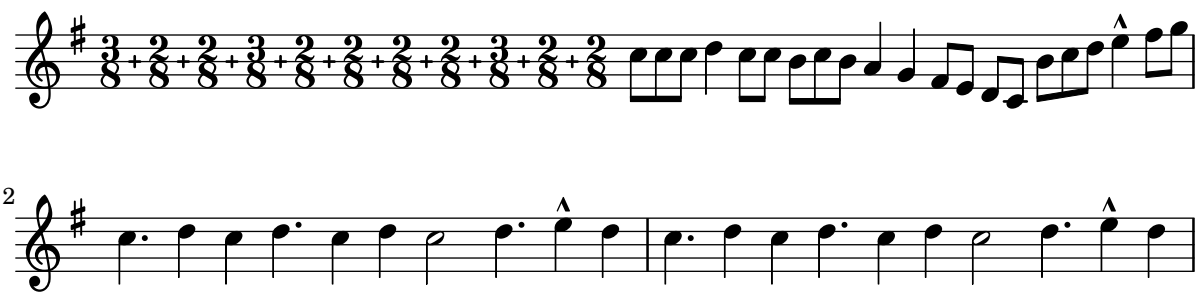
Indicaciones de compás polimétricas profundamente personalizadas

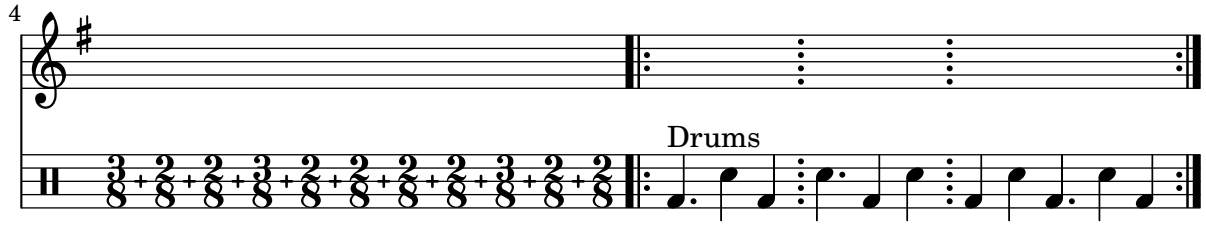
Aunque aquí el elemento más esencial no era la indicación de compás polimétrica que se muestra, se ha incluido para mostrar el pulso de la pieza, que por cierto es la plantilla de una canción de los Balcanes real.

```
melody = \relative c'' {
  \key g \major
  \time #'((3 . 8) (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8)
           (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8))
  \set Timing.beamExceptions = #'()
  \set Timing.beatStructure = 3,2,2,3,2,2,2,2,3,2,2
  c8 c c d4 c8 c b c b a4 g fis8 e d c b' c d e4-^ fis8 g \break
  c,4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
}

drum = \new DrumStaff \drummode {
  \repeat volta 2 {
    bd4.^\markup { Drums } sn4 bd \bar ";"
    sn4. bd4 sn \bar ";"
    bd sn bd4. sn4 bd
  }
}

\new Staff {
  \melody
  \drum
}
```





Ejemplo de cajas chinas grave y aguda

Dos cajas chinas, escritas como 'wbh' (*high woodblock*) y 'wbl' (*low woodblock*). La longitud de la línea divisoria se ha alterado con una instrucción `\override` para que no sea demasiado corta. Las posiciones de las dos líneas de la pauta se tienen que definir explícitamente también.

```
% These lines define the position of the woodblocks in the stave;
% if you like, you can change it or you can use special note heads
% for the woodblocks.
#(define mydrums '((hiwoodblock default #f  3)
                  (lowoodblock default #f -2)))

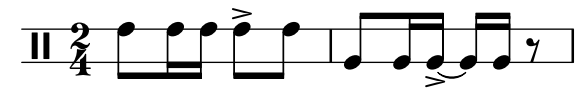
woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

  % This is necessary; if not entered,
  % the barline would be too short!
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)
  % small correction for the clef:
  \set DrumStaff.clefPosition = 0.5
}

\new DrumStaff {
  % with this you load your new drum style table
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \woodstaff

  \drummode {
    \time 2/4
    wbh8 16 16 8-> 8 |
    wbl8 16 16-> ~ 16 16 r8 |
  }
}
```



Making an object invisible using \hide

Applying `\hide` to a grob causes objects of this type to be printed with "invisible ink". They are not printed, but all of their other behavior is retained:
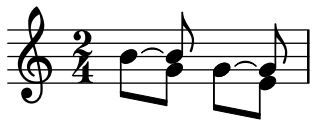
- the objects still take up space,
- they take part in collision resolution, and

- slurs, ties, and beams can be attached to them as usual.

This snippet demonstrates how to connect different voices using ties. Normally, ties only connect two notes in the same voice. By introducing a tie in a different voice, and blanking the first up-stem in that voice, the tie appears to cross voices.

```
\relative {
  \time 2/4
  <<
    {
      \once \hide Stem
      \once \override Stem.length = #8
      b'8 ~ 8\noBeam
      \once \hide Stem
      \once \override Stem.length = #8
      g8 ~ 8\noBeam
    }
    \\
    {
      b8 g g e
    }
  >>
}

\paper {
  line-width = 40\mm
  ragged-right = ##f
}
```
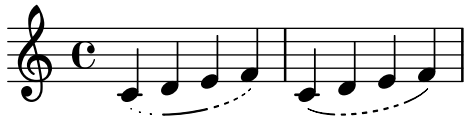


Hacer ligaduras de expresión con estructura compleja de discontinuidad

Las ligaduras de expresión se pueden construir con patrones de discontinuidad complejos mediante la definición de la propiedad `dash-definition`, que es una lista de segmentos de la ligadura, que a su vez son listas de parámetros que definen el comportamiento de discontinuidad del segmento dado.

Los segmentos de la ligadura se definen en términos del parámetro de bezier *t*, cuyo rango va de 0 a la izquierda de la ligadura hasta 1 en el extremo derecho de la ligadura. Cada segmento tiene la forma (*t-inicio t-final fracción-discontinuidad período-discontinuidad*). En el segmento que va desde *t-inicio* hasta *t-final*, el patrón se define por los valores de *fracción-discontinuidad* y *período-discontinuidad*. La *fracción-discontinuidad* especifica el período en que el trazo es continuo y de color negro. Si se establece al valor de 1 se obtiene un segmento de ligadura continuo. Las unidades de *período-discontinuidad* son espacios de pentagrama.

```
\relative c' {
  \once \override
    Slur.dash-definition = #'((  0  0.3  0.1 0.75)
                              (0.3  0.6  1   1   )
                              (0.65 1.0  0.4 0.75))
  c4( d e f)
```
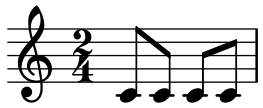
```
  \once \override
    Slur.dash-definition = #'((0    0.25  1   1   )
                              (0.3  0.7   0.4 0.75)
                              (0.75 1.0   1   1   ))
  c4( d e f)
}
```



Controlar manualmente las posiciones de las barras

Se pueden controlar manualmente las posiciones de las barras de corchea sobreescribiendo el valor del parámetro `positions` del objeto gráfico `Beam`.

```
\relative c' {
  \time 2/4
  % from upper staff-line (position 2) to center (position 0)
  \override Beam.positions = #'(2 . 0)
  c8 c
  % from center to one above center (position 1)
  \override Beam.positions = #'(0 . 1)
  c8 c
}
```

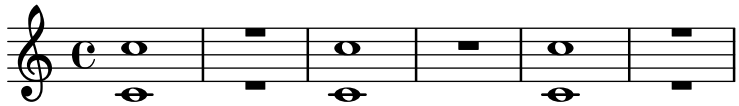


Combinar los silencios multicompás en una parte polifónica

Los silencios multicompás en un pentagrama polifónico se sitúan de forma diferente dependiendo de a qué voz pertenecen. Se pueden imprimir sobre la misma línea del pentagrama, usando el ajuste que aparece a continuación. Si omitimos la palabra clave `\once`, el cambio afecta a todos los silencios que siguen, dentro de la voz dada.

```
normalPos = \once \revert MultiMeasureRest.direction

<<
  { c''1 R c'' \normalPos R c'' R } \\
  { c'1 R c' \normalPos R c' R }
>>
```



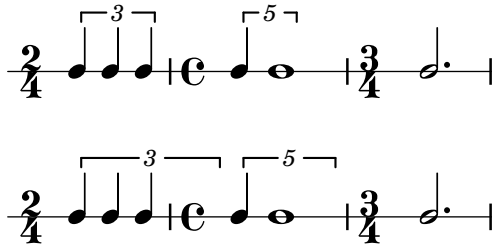
Modificar la longitud del corchete de grupo especial

Se puede hacer que los corchetes de grupo de valoración especial, como los tresillos, se extiendan horizontalmente hasta abarcar música anterior al propio grupo o incluyan la nota siguiente. Los corchetes predeterminados finalizan en el extremo derecho de la última nota del grupo; el corchete de un grupo de duración completa se exiende más a la derecha, bien para que cubra toda la notación no rítmica hasta la nota siguiente, o bien para que cubra solamente el espacio

vacía que hay antes del elemento siguiente de notación, sea éste una clave, una indicación de compás, u otra nota. El ejemplo muestra cómo activar el modo de duración completa de los grupos de valoración especial y cómo cambiar la parte de música que cubren.

```
\new RhythmicStaff {
  % Defaults.
  \time 2/4 \tuplet 3/2 { c4 4 4 }
  \time 4/4 \tuplet 5/4 { 4 1 }
  \time 3/4 2.
}
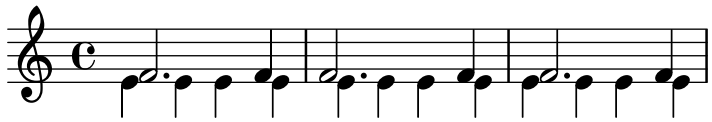
\new RhythmicStaff {
  % Set tuplets to be extendable...
  \set tupletFullLength = ##t
  % ...to cover all items up to the next note
  \set tupletFullLengthNote = ##t
  \time 2/4 \tuplet 3/2 { c4 4 4 }
  % ...or to cover just whitespace.
  \set tupletFullLengthNote = ##f
  \time 4/4 \tuplet 5/4 { 4 1 }
  \time 3/4 2.
}
```



Desplazar las notas con puntillo en polifonía

Cuando se puede una nota en la voz superior para evitar la colisión con una nota de otra voz, el comportamiento predeterminado es desplazar la nota superior a la derecha. Se puede cambiar modificando la propiedad `prefer-dotted-right` del objeto gráfico `NoteCollision`.

```
\new Staff \relative c' <<
  {
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##f
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##t
    f2. f4
  }
  \\
  { e4 e e e e e e e e e e }
>>
```



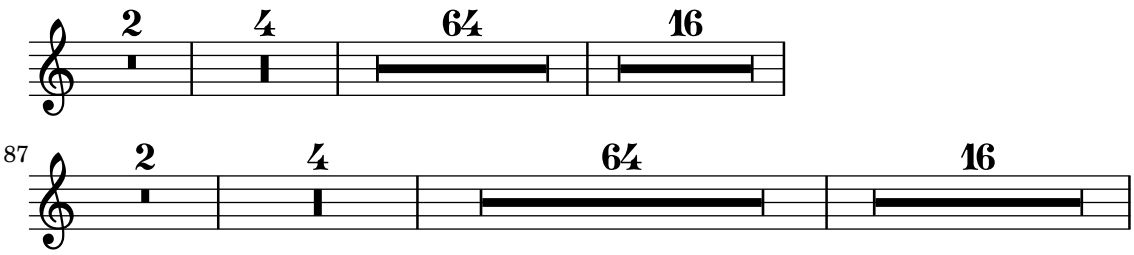
Control de la anchura de un silencio multicompás

Los silencios de varios compases tienen una longitud que concuerda con su duración total, que está bajo el control de la propiedad `space-increment` del grob `MultiMeasureRest`; su valor predeterminado es 2.

```
\relative c' {
  \omit Staff.TimeSignature
  \compressEmptyMeasures

  R1*2 R1*4 R1*64 R1*16 \break
  \override MultiMeasureRest.space-increment = 4
  R1*2 R1*4 R1*64 R1*16
}

\layout {
  ragged-right = ##t
}
```

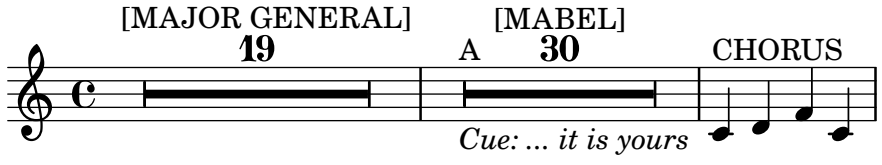


Marcado de silencios multicompás

Los elementos de marcado aplicados a un silencio multicompás se centran encima o debajo de éste. Los elementos de marcado extensos que se adjuntan a silencios multicompás no producen la expansión del compás. Para expandir un silencio multicompás de forma que quepa todo el marcado, utilice un acorde vacío con un marcado aplicado antes del silencio multicompás.

El texto aplicado a un siencio sparador de esta forma se alinea por la izquierda a la posición en que la nota estaría situada dentro del compás, pero si la longitud del compás está determinada por la longitud del texto, éste aparecerá centrado.
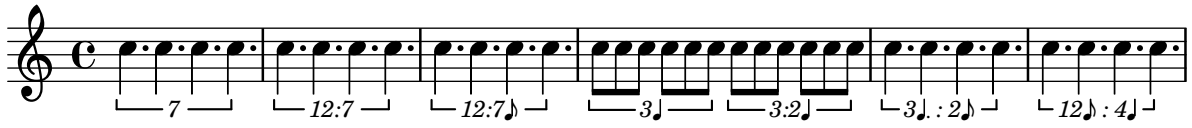
```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```



Números de agrupación especial distintos a los predeterminados

LilyPond también proporciona funciones de formato para imprimir números de grupo especial diferentes a la propia fracción, así como para añadir una figura al número o a la fracción de la agrupación.

```
\relative c'' {
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      (tuplet-number::non-default-tuplet-fraction-text 12 7)
      (ly:make-duration 3 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-denominator-text
      (ly:make-duration 2 0))
  \tuplet 3/2  { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-fraction-text
      (ly:make-duration 2 0))
  \tuplet 3/2  { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::fraction-with-notes
      (ly:make-duration 2 1) (ly:make-duration 3 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-fraction-with-notes 12
      (ly:make-duration 3 0) 4 (ly:make-duration 2 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
}
```

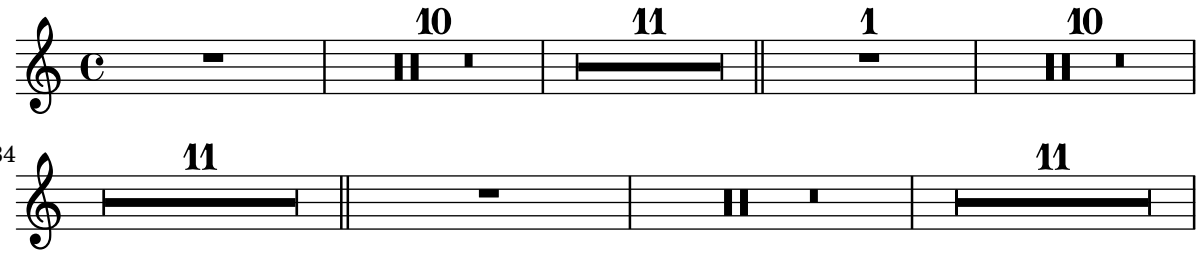


Numerar silencios seguidos de un compás

Los silencios de varios compases muestran su duración con un número excepto los que tienen un solo compás. Esto se puede cambiar fijando el valor de `restNumberThreshold`.

```
{
  \compressEmptyMeasures
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 0
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 10
  R1 R1*10 R1*11
```

```
}
```



Partcombine y \autoBeamOff

Puede ser difícil comprender la función de `\autoBeamOff` cuando se usa con `\partCombine`. Puede ser preferible usar
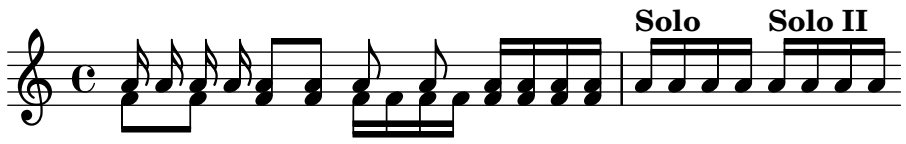
```
\set Staff.autoBeaming = ##f
```

en su lugar, para asegurarnos de que el barrado automático se desactiva para todo el pentagrama. Úselo en un lugar de la partitura en el que no esté aún activo ninguna barra generada por el algoritmo de barrado automático.

Internamente, `\partCombine` funciona con cuatro voces: plica arriba cuando hay una sola voz, plica abajo con la otra, combinadas, y una sola voz. Para poder usar `\autoBeamOff` con el objeto de detener todo el barrado automático cuando se usa con `\partCombine`, es necesario hacer *cuatro* llamadas a `\autoBeamOff`.

```
{
  % \set Staff.autoBeaming = ##f % turns off all auto-beaming

  \partCombine {
    \autoBeamOff   % applies to split up-stems
    \*4 a'16
    % \autoBeamOff % applies to combined stems
    \*4 a'8
    \*4 a'16
    % \autoBeamOff % applies to solo
    \*4 a'16
    r4
  } {
    % \autoBeamOff % applies to split down-stems
    \*4 f'8
    \*8 f'16 |
    r4
    \*4 a'16
  }
}
```



Ejemplo de percusión

Un ejemplo breve extraído de la *Historia de un soldado* de Strawinsky.

```
#(define mydrums '((bassdrum   default #f  4)
                   (snare      default #f -4)
                   (tambourine default #f  0)))

U = \stemUp
D = \stemDown

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}

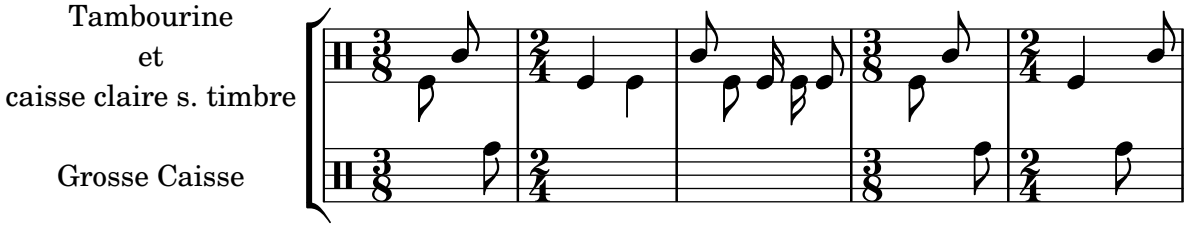
drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
      sn4 \D sn4 |
      \U tamb8 \D sn \U sn16 \D sn \U sn8 |
      \D sn8 \U tamb s |
      \U sn4 s8 \U tamb
    }
  >>
}

drumsB = \drummode {
  s4 bd8 s2*2 s4 bd8 s4 bd8 s
}

\layout {
  indent = 40\mm
  \context {
    \DrumStaff
    drumStyleTable = #(alist->hash-table mydrums)
  }
}

\score {
  \new StaffGroup <<
    \new DrumStaff \with {
      instrumentName = \markup \center-column {
        "Tambourine"
        "et"
        "caisse claire s. timbre" }
    } \drumsA
    \new DrumStaff \with {
      instrumentName = "Grosse Caisse"
    }\drumsB
  >>
```

```
}
```



Permitir saltos de línea dentro de grupos especiales con barra

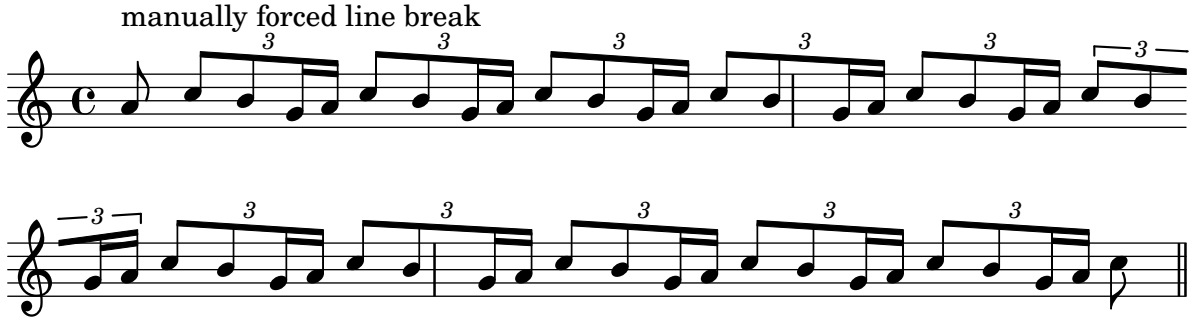
Estos ejemplos artificiales muestran cómo se pueden permitir tanto los saltos de línea manuales como los automáticos dentro de un grupo de valoración especial unido por una barra que no se puede dividir en una duración de manera exacta.

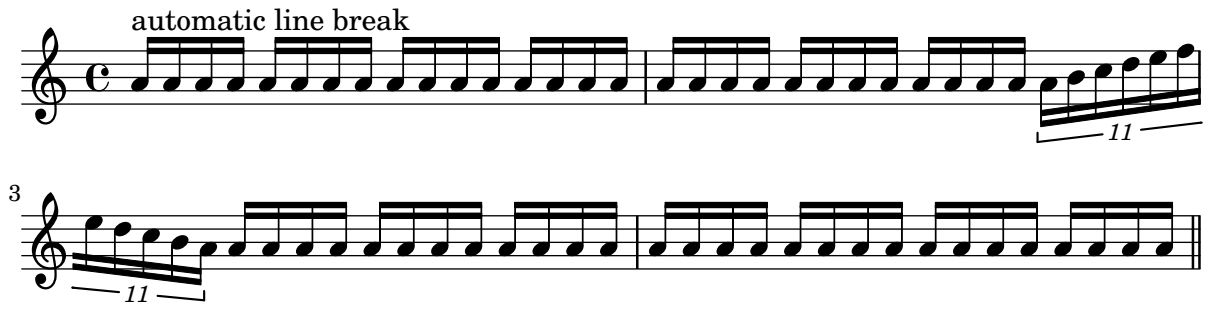
Esta funcionalidad solo funciona con grupos de valoración especial barrados manualmente.

```
\layout {
  \context {
    \Voice
    % Permit automatic line breaks within tuplets.
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks.
    \override Beam.breakable = ##t
  }
}

\relative c'' {
  <>^"manually forced line break"
  a8
  \*5 { \tuplet 3/2 { c8[ b g16 a] } }
  \tuplet 3/2 { c8[ b \break g16 a] }
  \*5 { \tuplet 3/2 { c8[ b g16 a] } }
  c8 \bar "||"
}

\relative c'' {
  <>^"automatic line break"
  \*28 a16
  \tuplet 11/8 { a16[ b c d e f e d c b a] }
  \*28 a16 \bar "||"
}
```
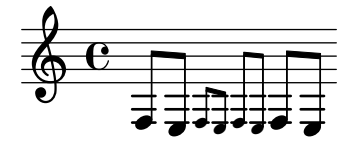




Colocar las barras de las notas de adorno a la misma altura que las barras de notas normales

Cuando se colocan figuras sobre líneas adicionales, sus barras se sitúan generalmente en medio del pentagrama. La barra de las notas de adorno es más corta y las notas de adorno sobre líneas adicionales podrían tener la barra fuera del pentagrama. Podemos corregir este barrado para las notas de adorno.

```
\relative c {
  f8[ e]
  \grace {
    f8[ e]
    \override Stem.no-stem-extend = ##f
    f8[ e]
    \revert Stem.no-stem-extend
  }
  f8[ e]
}
```



Posicionamiento de las notas de adorno con espacio flotante

Al establecer la propiedad `strict-grace-spacing` hacemos que las columnas musicales para las notas de adorno sean 'flotantes', es decir, desacopladas de las notas que no son de adorno: primero se aplica el espaciado de las notas normales, y luego se ponen las columnas musicales de las notas de adorno a la izquierda de las columnas musicales de las notas principales.

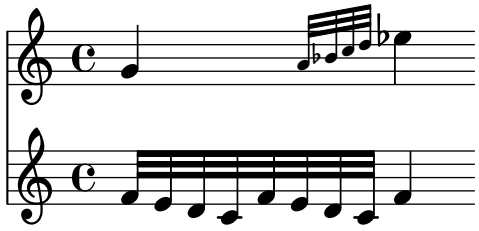
Sin embargo, a causa del bug número 6876 (`https://gitlab.com/lilypond/lilypond/-/issues/6876`), las alteraciones accidentales se ignoran si esta propiedad está establecida. Este fragmento de código ofrece una solución para este problema.

Otro desafortunado efecto colateral de esta propiedad es que LilyPond no comprueba si hay espacio horizontal suficiente para las notas de adorno (este problema recibe seguimiento como bug número 2630 (`https://gitlab.com/lilypond/lilypond/-/issues/2630`)). Tenemos que asegurarnos de que ha espacio suficiente, por ejemplo, por medio del uso de `\newSpacingSection` junto con un valor adecuado para la duración `base-shortest-duration` del objeto gráfico `SpacingSpanner`.

```
shiftedGrace =
#(define-music-function (offset music) (number? ly:music?)
   #{
     \override NoteHead.X-offset = #(- offset 0.85)
     \override Stem.X-offset = #offset
     \grace { $music }
     \revert NoteHead.X-offset
     \revert Stem.X-offset
   #})

\relative c'' <<
  { g4 \shiftedGrace #-1.3 a32 \shiftedGrace #-0.5 { bes c d } es4 }
  { f,32 e d c f e d c f4 }
>>

\layout {
  \context {
    \Score
    \override SpacingSpanner.strict-grace-spacing = ##t
  }
}
```



Posicionar los silencios multicompás

A diferencia de los silencios normales, no existe una instrucción predefinida para modificar la posición predeterminada de un símbolo de silencio multicompás sobre el pentagrama, adjuntándolo a una nota, independientemente de cuál sea su forma. Sin embargo, en la música polifónica los silencios multicompás de las voces de numeración par e impar están separados verticalmente.

Este fragmento de código muestra la forma de controlar la colocación de los silencios multicompás.

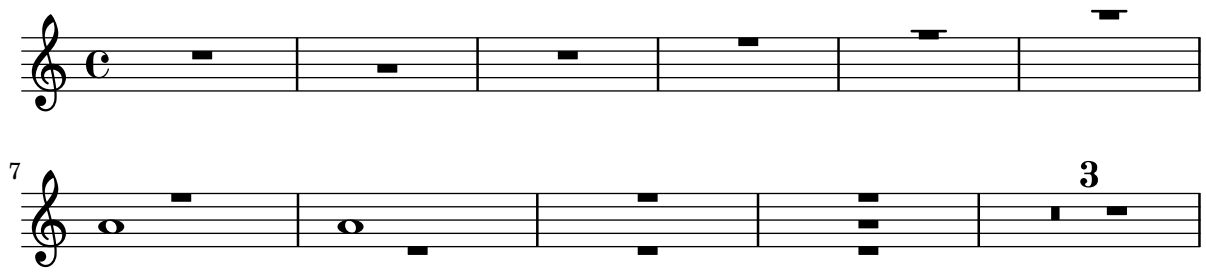
```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line.
  R1
  % They can be moved using an override or tweak.
  \tweak staff-position -2 R1
  \tweak staff-position 0 R1
  \tweak staff-position 2 R1
  \override MultiMeasureRest.staff-position = 3 R1
  \override MultiMeasureRest.staff-position = 6 R1
  \revert MultiMeasureRest.staff-position
  \break

  % Odd-numbered voices are under the top line.
  << { R1 } \\ { a1 } >>
  % Even-numbered voices are under the bottom line.
```

```
  << { a1 } \\ { R1 } >>
  % Multi-measure rests in both voices remain separate.
  << { R1 } \\ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override or tweak.
  << { R1 } \\ { R1 } \\ { \tweak staff-position -2 R1 } >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed.
  \compressMMRests
  <<
    \revert MultiMeasureRest.direction
    { R1*3 } \\
    \revert MultiMeasureRest.direction
    { R1*3 }
  >>
}
```



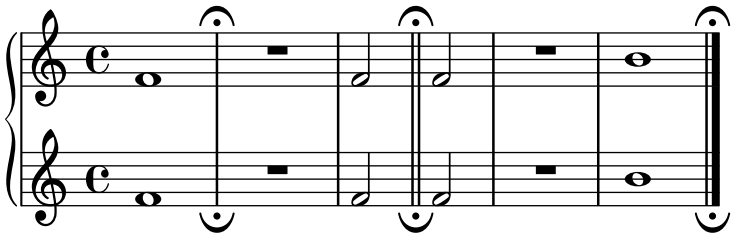
Colocar calderones opuestos sobre la barra de compás

Este fragmento es una demostración de una instrucción que imprime calderones encima y debajo de la línea divisoria al mismo tiempo. Si no hay ninguna barra de compás, añade una doble barra. Semánticamente, la instrucción codifica una cesura más larga de lo normal, lo que se puede considerar un mal uso dependiendo de la situación.

```
twoWayFermata = {
  \once \set Staff.caesuraType = #'((underlying-bar-line . "||"))
  \once \set Staff.caesuraTypeTransform = ##f
  \caesura ^\fermata _\fermata
}

music = {
  f'1 \twoWayFermata
  R1
  f'2 \twoWayFermata f'2
  R1
  b'1 \twoWayFermata \fine
}

\new GrandStaff <<
  \new Staff \music
  \new Staff \music
>>
```



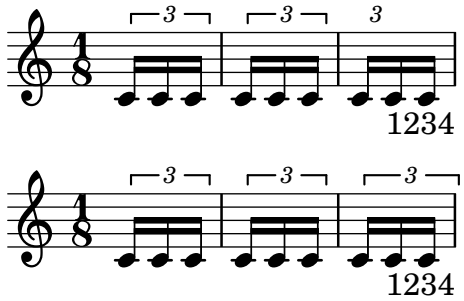
Evitar que una marca de ensayo final suprima el corchete de un grupo de valoración especial

A causa del bug número 2362 (`https://gitlab.com/lilypond/lilypond/-/issues/2362`), al añadir una marca de ensayo `mark` al final puede ocurrir que se pierda una indicación de grupo especial colocado al final. Se puede solventar fijando `TupletBracket.full-length-to-extent` a `#f`.

```
\new Staff {
  \set tupletFullLength = ##t
  \time 1/8
  \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' c' }
  \tweak direction #DOWN \textEndMark "1234"
}

\new Staff {
  \set tupletFullLength = ##t
  \override TupletBracket.full-length-to-extent = ##f

  \time 1/8
  \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' c' }
  \tweak direction #DOWN \textEndMark "1234"
}
```

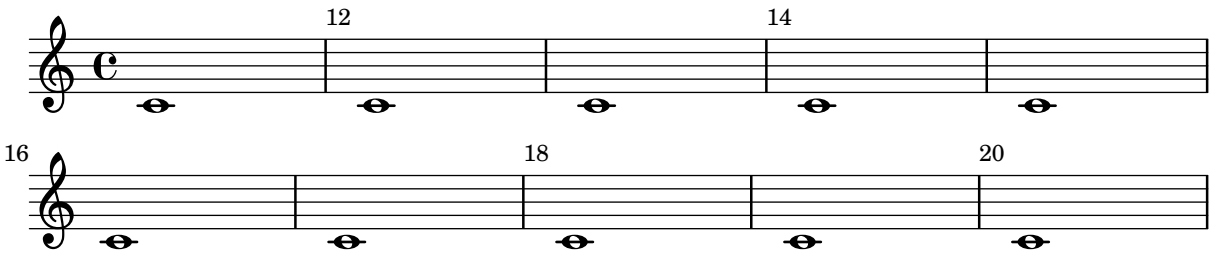


Imprimir números de compás a intervalos regulares

Se pueden imprimir los números de compás a intervalos regulares mediante el establecimiento de la propiedad `barNumberVisibility`. Aquí los números de compás se imprimen a cada dos compases excepto al final de la línea.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = 11
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c
  \break
```

```
  c1 | c | c | c | c
}
```

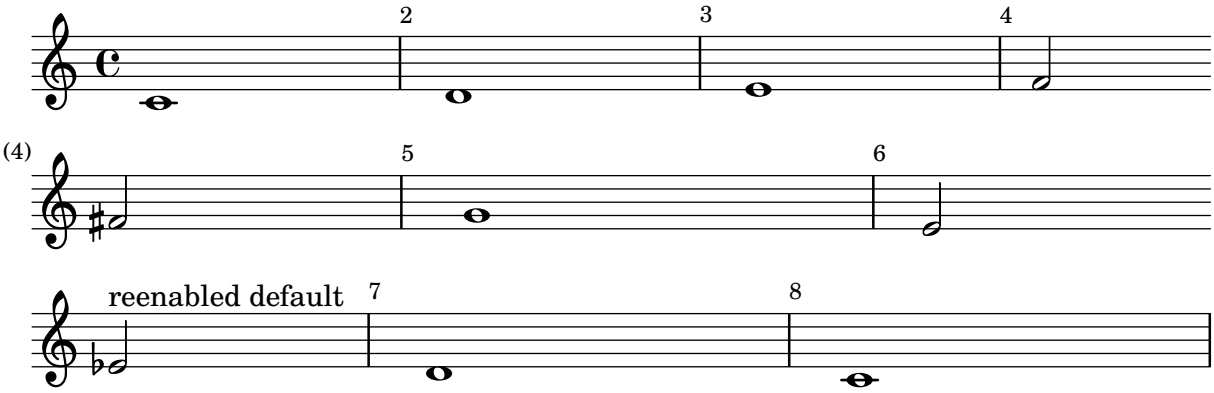


Impresión de números de compás para compases divididos

Por omisión, el número de un compás que está dividido no se repite al comienzo de la línea siguiente. Use `first-bar-number-invisible-save-broken-bars` para `barNumberVisibility` para tener un `BarNumber` encerrado entre paréntesis en ese caso.

```
\layout {
  \context {
    \Score
    barNumberVisibility = #first-bar-number-invisible-save-broken-bars
    \override BarNumber.break-visibility = ##(#f #t #t)
  }
}

\relative c' {
  c1 | d | e | f2 \break
  fis2 | g1 | e2 \break
  <>^"reenabled default"
  % back to default -
  % \unset Score.barNumberVisibility would do so as well
  \set Score.barNumberVisibility =
    #first-bar-number-invisible-and-no-parenthesized-bar-numbers
  es2 | d1 | c
}
```



Imprimir números de compás dentro de rectángulos o circunferencias

Los números de compás también se pueden imprimir dentro de rectángulos o de circunferencias.

```
\relative c' {
  % Center bar numbers except at the beginning of a staff.
  \override Score.BarNumber.self-alignment-X =
    #(break-alignment-list CENTER CENTER 0.3)

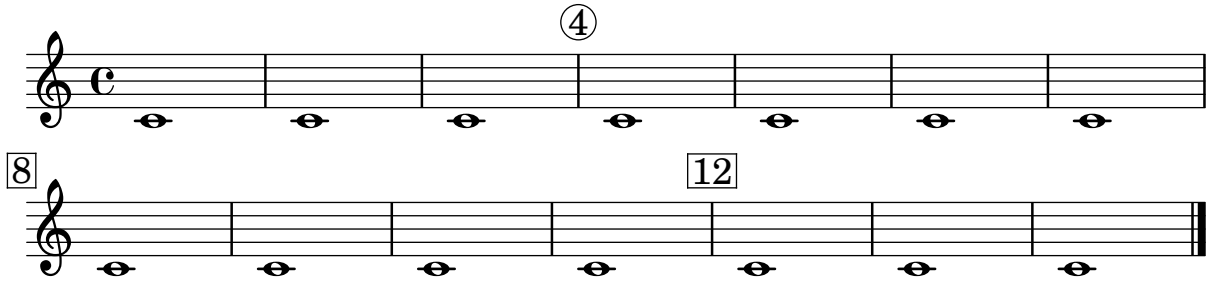
  % Prevent bar numbers at the end of a line and permit them elsewhere.
  \override Score.BarNumber.break-visibility = #end-of-line-invisible

  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2.
  \override Score.BarNumber.font-size = 2

  % Draw a circle round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \*7 c1 \break

  % Draw a box round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \*7 c1 \bar "|."
}
```



Impresión de los números de compás usando `modulo-bar-number-visible`

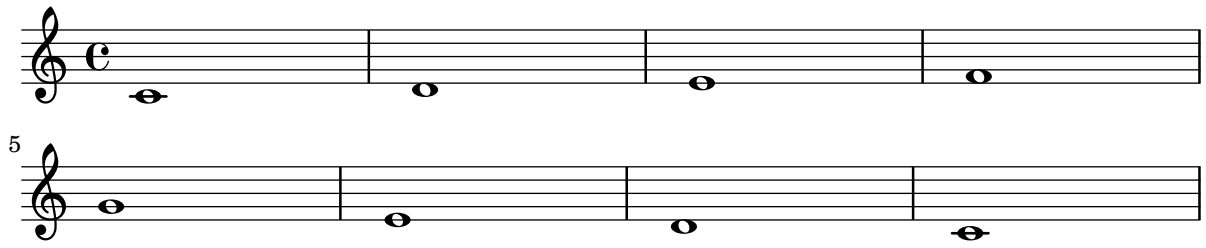
Si el resto de dividir el número del compás en curso entre el primer argumento del `modulo-bar-number-visible` es igual al segundo argumento, entonces imprimir el número del compás.

Es útil para imprimir el número de compás a ciertas distancias. Algunos ejemplos:

- `(modulo-bar-number-visible 3 2)` → imprime 2, 5, 8, ...
- `(modulo-bar-number-visible 4 2)` → imprime 2, 6, 10, ...
- `(modulo-bar-number-visible 2 1)` → imprime 3, 5, 7, ...
- `(modulo-bar-number-visible 5 0)` → imprime 5, 10, 15, ...

```
\layout {
  \context {
    \Score
    \override BarNumber.break-visibility = ##(#f #t #t)
    barNumberVisibility = #(modulo-bar-number-visible 5 0)
  }
}
```

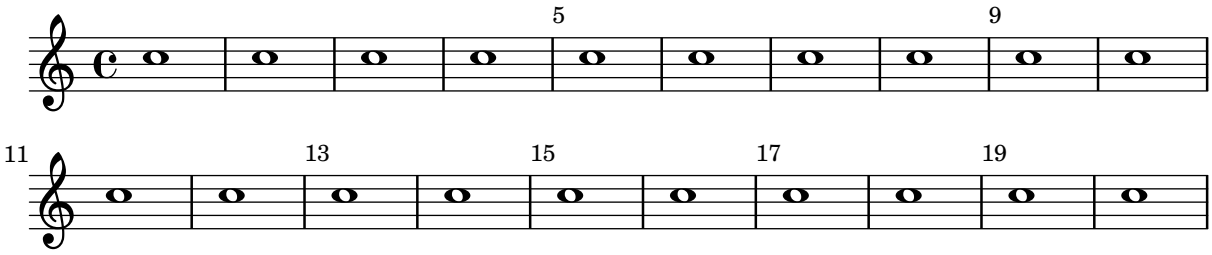
```
\relative c' {
  c1 | d | e | f \break
  g1 | e | d | c
}
```



Impresión de los números de compás con intervalos regulares variables

Usando la función de contexto `set-bar-number-visibility`, se pueden cambiar los intervalos de numeración de los compases.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \context Score \applyContext #(set-bar-number-visibility 4)
  \*10 c'1
  \context Score \applyContext #(set-bar-number-visibility 2)
  \*10 c
}
```



Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama

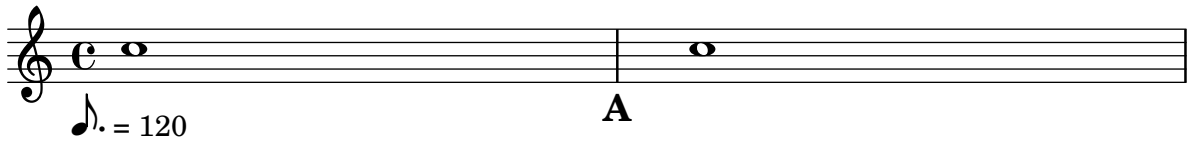
De forma predeterminada, las indicaciones metronómicas y las letras de ensayo se imprimen encima del pentagrama. Para colocarlas debajo del pentagrama, simplemente ajustamos adecuadamente la propiedad `direction` de `MetronomeMark` o de `RehearsalMark`.

```
\layout {
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
```

```
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Imprimir música que tenga compases distintos en cada pentagrama

En el siguiente fragmento de código, dos partes distintas tienen un compás completamente diferente y sin embargo se mantienen sincronizadas. Las barras de compás ya no se pueden imprimir en el nivel de `Score`; para permitir barras de compás independientes en cada parte se trasladan los grabadores `Default_barline_engraver` y `Timing_translator` desde el contexto de partitura `Score` hasta el contexto de pentagrama `Staff`.

Si son necesarios números de compás, el grabador de números de compás `Bar_number_engraver` también debe trasladarse, ya que descansa en propiedades fijadas por el `Timing_translator`; se puede usar un bloque `\with` para añadir números de compás al pentagrama apropiado.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}
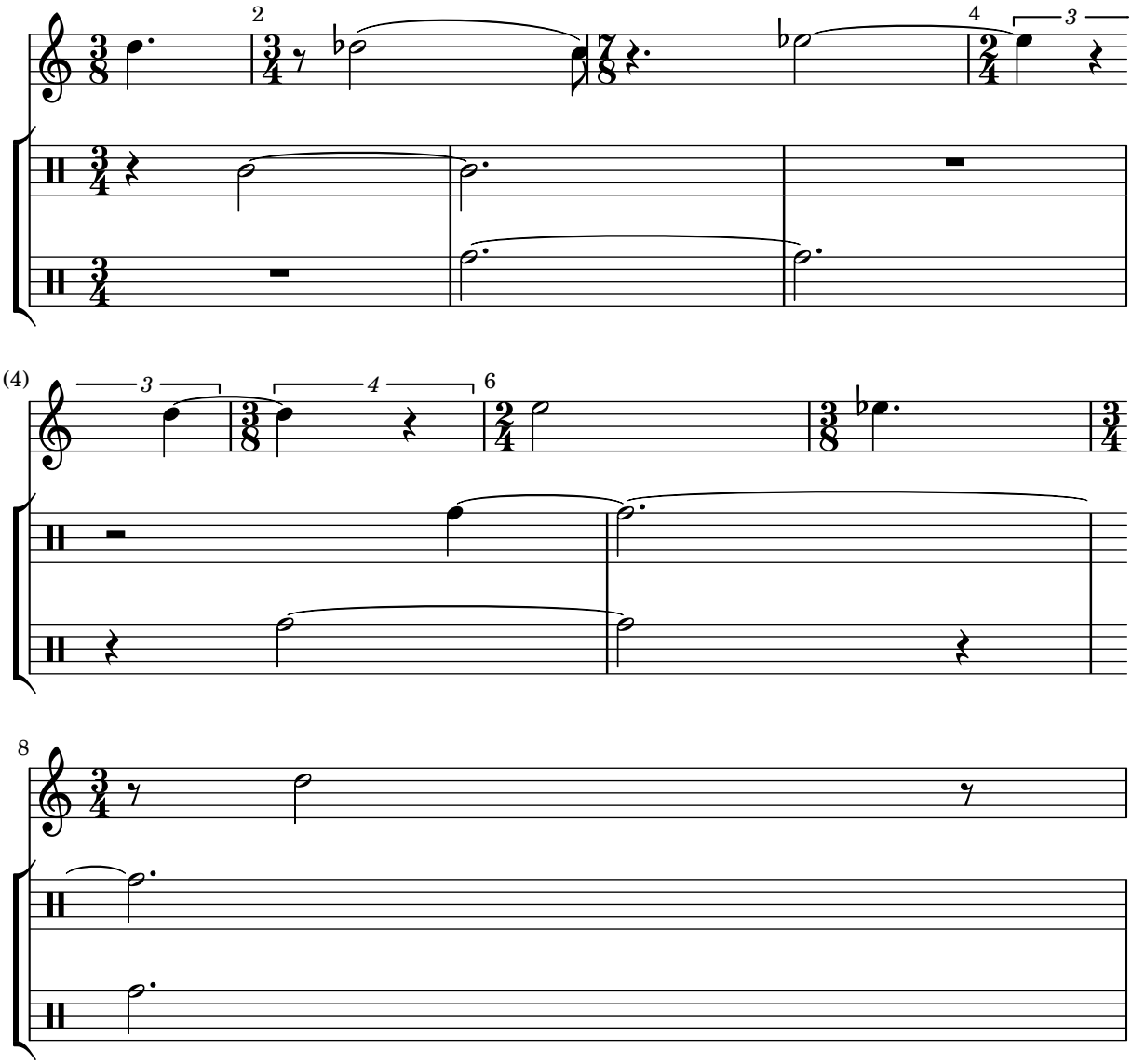
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
```

```
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. ees''2 ~ |
    \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
    \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
    \time 2/4 e''2 |
    \time 3/8 es''4. |
    \time 3/4 r8 d''2 r8 |
  }
>>

Perkussion = \new StaffGroup <<
  \new Staff <<
    \global
    {
      \clef percussion
      \time 3/4 r4 c'2 ~ |
      c'2. |
      R2. |
      r2 g'4 ~ |
      g'2. ~ |
      g'2. |
    }
  >>
  \new Staff <<
    \global {
      \clef percussion
      \time 3/4 R2. |
      g'2. ~ |
      g'2. |
      r4 g'2 ~ |
      g'2 r4 |
      g'2. |
    }
  >>
>>

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}
```

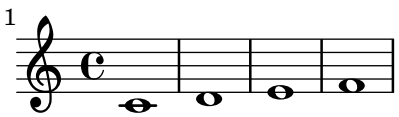


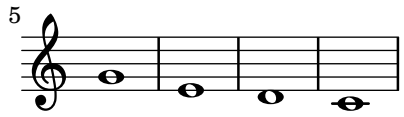
Imprimir el número de compás en el primer compás

De forma predeterminada se suprime el número del primer compás de una partitura si es menor o igual a to 1. Esto se puede cambiar fijando la propiedad de contexto `barNumberVisibility` con el valor `all-bar-numbers-visible`

```
\paper {
  line-width = 50\mm
}

\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  c1 | d | e | f \break
  g1 | e | d | c
}
```





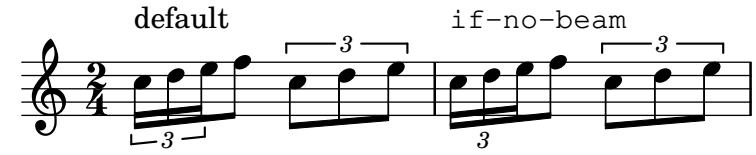
Imrpesión de corchetes de tresillo en el lado de la cabeza de la nota

Cualquiera que sea la opción elegida para el control de la visibilidad del corchete de los tresillos, mostrará u ocultará el corchete independientemenre de su ubicación (en el lado de la plica o en el lado de la cabeza). Sin embargo, cuando se coloca el corchete del lado de la cabeza, algunos autores recomiendan imprimir siempre el corchete de tresillo. Se puede usar la opción `visible-over-note-heads` para obtener esto.

```
music = \relative c'' {
  \tupletNeutral \tuplet 3/2 { c16[ d e } f8]
  \tupletUp \tuplet 3/2 { c8 d e }
}

\new Voice {
  \relative c' {
    \override TextScript.staff-padding = #2.5

    \time 2/4
    \override TupletBracket.visible-over-note-heads = ##t
    \override Score.TextMark.non-musical = ##f
    <>^\markup "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    <>^\markup \typewriter "if-no-beam" \music
  }
}
```



Redefinición de los valores globales predeterminados para notas de adorno

Los valores predeterminados para las notas de adorno están almacenados en los identificadores siguientes.
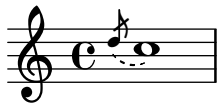
```
startGraceMusic
stopGraceMusic
startAcciaccaturaMusic
stopAcciaccaturaMusic
startAppoggiaturaMusic
stopAppoggiaturaMusic
```

Todos ellos están definidos en el archivo `ly/grace-init.ly`. Redefiniéndolos se pueden obtener otros efectos.

```
startAcciaccaturaMusic = {
  <>(
  \override Flag.stroke-style = "grace"
  \slurDashed
}

stopAcciaccaturaMusic = {
  \revert Flag.stroke-style
  \slurSolid
  <>)
}

\relative c'' {
  \acciaccatura d8 c1
}
```

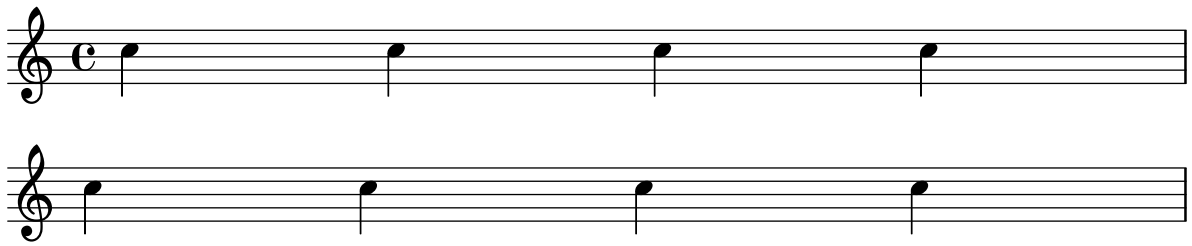


Suprimir los números de compás de toda la partitura

Se pueden eliminar completamente los números de compás quitando el grabador `Bar_number_engraver` del contexto de `Score`.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    % \remove "Bar_number_engraver"
  }
}

\relative c'' {
  c4 c c c \break
  c4 c c c
}
```



Estilos de silencios

Los silencios se pueden imprimir en distintos estilos.

```
restsA = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
  s64 s128 s256 s512 s1024 s1024
}
restsB = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
  r64 r128 r256 r512 r1024 s1024
}

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

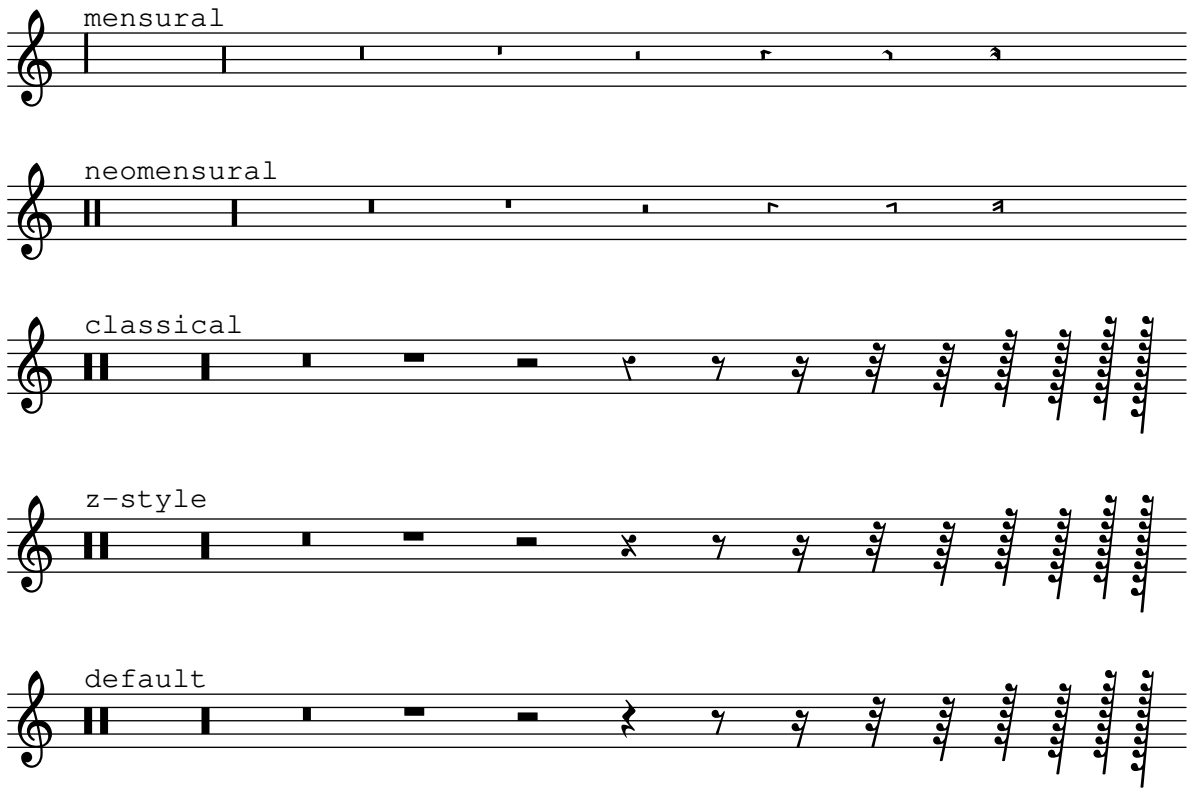
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>^\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>^\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>^\markup \typewriter { default } \restsB \bar "" \break
}
```
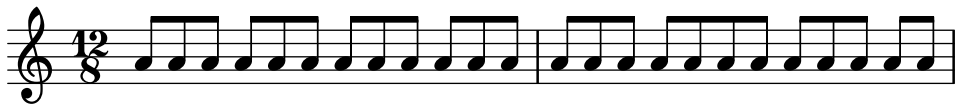


Alteración de los finales de barra predeterminados

Para tipografiar las barras agrupadas en la forma 3-4-3-2 solo es necesario modificar la estructura de pulsos:

```
\relative c'' {
  \time 12/8

  % Default beaming
  a8 a a a a a a a a a a a

  % Set new values for beam endings
  \set Score.beatStructure = 3,4,3,2
  a8 a a a a a a a a a a a
}
```



Barras rítmicas

En las hojas guía de acordes o *lead-sheets* "sencillas", a veces no se imprime realmente ninguna nota. En su lugar se hace una notación que tiene solamente "patrones rítmicos" y acordes encima de los compases para representar la estructura de la canción. Tal funcionalidad puede ser útil al crear o transcribir la estructura de una canción, o si se quieren compartir las hojas guía con guitarristas o músicos de jazz.

```
startPat = {
  \improvisationOn
  \omit Stem
}
stopPat = {
  \improvisationOff
  \undo \omit Stem
}

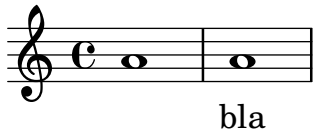
\new Voice \with {
  \consists Pitch_squash_engraver
} {
  c'4 d' e' f' |
  \startPat
  4 4 4 4 |
  \stopPat
  f'4 e' d' c'
}
```



Silencios de separación en modo de letra

La sintaxis 's' para los silencios de desplazamiento solamente está disponible en los modos de nota y de acorde. En otras situaciones, por ejemplo al escribir la letra de las canciones, se recomienda usar la instrucción `\skip`.
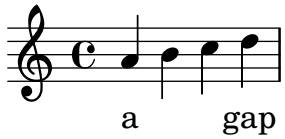
```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip1 bla1 }
>>
```



Silencios de separación en modo de letra (2)

Aunque no se pueden usar los silencios de separación 's' dentro de `\lyricmode` (se toman como una "s", literal, no como un espacio), sí están disponibles las comillas dobles (`""`) y el carácter subrayado (`_`).

```
<<
  \relative c'' { a4 b c d }
  \new Lyrics \lyricmode { a4 "" _ gap }
>>
```



Stemlets (plicas de tamaño reducido)

En ciertas convenciones notacionales se permite que las barras se extiendan por encima de los silencios. Dependiendo de nuestras preferencias, estas barras pueden presentar pequeñas plicas o 'stemlets' para que el ojo aprecie más fácilmente el ritmo, y en algunos casos de música moderna el propio silencio se omite y permanecen solamente las pequeñas plicas sin cabeza.

Este fragmento de código muestra una progresión a partir de la notación tradicional, pasando por barras sobre silencios y plicas sobre los silencios, hasta las plicas solamente. Las pequeñas plicas o Stemlets se generan sobreescribiendo la propiedad `stemlet-length` del objeto `Stem`, y los silencios se ocultan usando `\hide`.
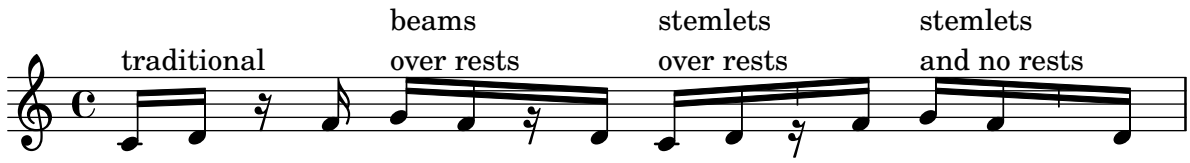
Algunos elementos de `\markup` se incluyen en la fuente para imprimir títulos sobre cada estilo de notación.

```
\paper {
  ragged-right = ##f
}

{
  c'16^\markup { traditional } d' r f'
  g'16[^\markup \column { "beams" "over rests" } f' r d']

  % N.B. use Score.Stem to set for the whole score.
  \override Staff.Stem.stemlet-length = #0.75
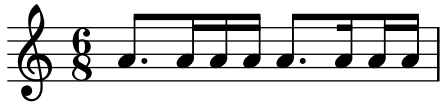
  c'16[^\markup \column { "stemlets" "over rests" } d' r f']
  g'16[^\markup \column { "stemlets" "and no rests" } f'
  \once \hide Rest
  r16 d']
}
```



Barras que se atienen al pulso estrictamente

Se puede hacer que las barras secundarias apunten en la dirección del pulso o fracción a que pertenecen. La primera barra evita los corchetes sueltos (que es el comportamiento predeterminado); la segunda barra sigue el pulso o fracción estrictamente.

```
\relative c'' {
  \time 6/8
  a8. a16 a a
  \set strictBeatBeaming = ##t
  a8. a16 a a
}
```



Subdivisión de barras

Las barras de semicorchea, o de figuras más breves, no se subdividen de forma predeterminada. Esto es, las barras de más de dos plicas se amplían sobre el grupo completo de notas, sin dividirse. Este comportamiento se puede modificar para subdividir las barras en subgrupos mediante el establecimiento de la propiedad `subdivideBeams` al valor `#t`. Cuando está activada, las barras se subdividen a intervalos definidos por el valor actual de la subivisión.

Usando las propiedades `beamMinimumSubdivision` y `beamMaximumSubdivision` se pueden configurar los límites de la subdivisión automática de las barras, concretamente las duraciones mínima y máxima en que las barras secundarios se eliminan. Los valores por omisión son `0` para la primera y `+inf.0` para la segunda, haciendo que LilyPond subdivida las barras tanto como sea posible.

Existen dos casos especiales que tener en cuenta.

- Si el numerador de `beamMaximumSubdivision` no es una potencia de 2, las duraciones candidatas a la subdivisión son `beamMaximumSubdivision` dividido por las potencias de 2 que sean mayores o iguales que `beamMinimumSubdivision`.
- Si `beamMaximumSubdivision` es menor que `beamMinimumSubdivision`, la profundidad de las subdivisiones de las barras está limitada por `beamMaximumSubdivision`, pero no la frecuencia y los intervalos rítmicos, desviándose por tanto, posiblemente, de las longitudes correctas o esperadas.

Si `respectIncompleteBeams` se fija al valor `#t`, las subdivisiones incompletas con más de dos plicas se tratan como una 'extensión' del grupo de subdivisión previo, es decir, la duración del grupo de subdivisión anterior se extiende para que comprenda también la subdivisión incompleta. Si en vez de ello se fija a `#f` (que es lo predeterminado), se inicia un grupo de subdivisión nuevo.

```
\relative c'' {
  \time 1/4

  <>^"default"
  c32 c c c c c c c

  <>^"with subdivision"
  \set subdivideBeams = ##t
  c32 c c c c c c c

  <>^"min 1/8"
  \once \set beamMinimumSubdivision = #1/8
  c32 c c c c c c c

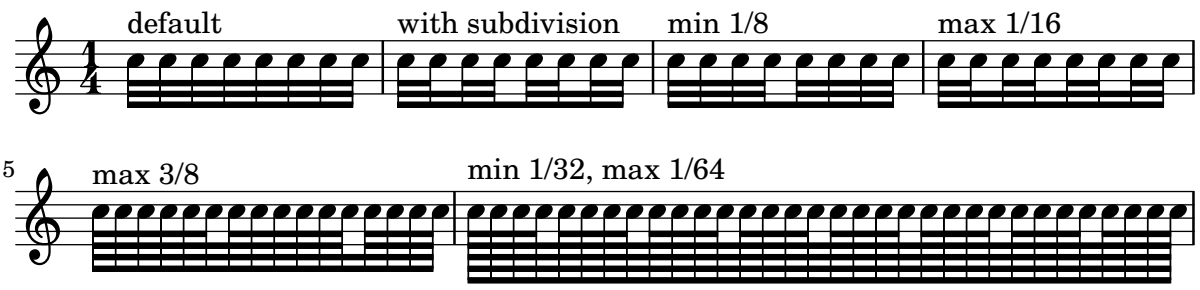
  <>^"max 1/16"
  \once \set beamMaximumSubdivision = #1/16
  c32 c c c c c c c

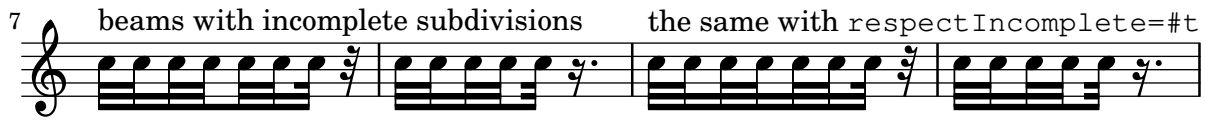
  <>^"max 3/8"
  \once \set beamMaximumSubdivision = #3/8
  \*16 c64

  <>^"min 1/32, max 1/64"
  % Set maximum beam subdivision interval to 1/64 to limit
  % subdivision depth, despite not being metrically correct.
  \once \set beamMinimumSubdivision = #1/32
  \once \set beamMaximumSubdivision = #1/64
  \*32 c128
  \break

  <>^"beams with incomplete subdivisions"
  c32 c c c c c c r32
  c32 c c c c r16.

  <>^\markup { "the same with"
               \typewriter { "respectIncomplete=#t" } }
  \set respectIncompleteBeams = ##t
  % The incomplete subgroup extends the completed subgroup.
  c32 c c c c c c r32
  % No visual change since we have only two stems in the
  % incomplete subgroup.
  c32 c c c c r16.
}
```





Ejemplo de música para tam-tam

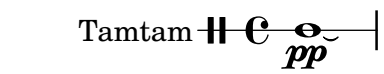
Un ejemplo de tam-tam, escrito como 'tt'.

```
#(define mydrums '((tamtam default #f 0)))

\new DrumStaff \with { instrumentName = #"Tamtam" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  tt 1 \pp \laissezVibrer
}
```



Ejemlo de pandereta

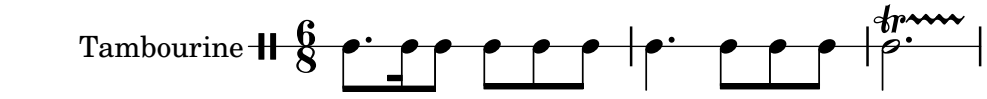
A tambourine example, entered with 'tamb'.

```
#(define mydrums '((tambourine default #f 0)))

\new DrumStaff \with { instrumentName = #"Tambourine" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  \time 6/8
  tamb8. 16 8 8 8 8 |
  tamb4. 8 8 8 |
  % The trick with the scaled duration and the shorter rest
  % is neccessary for the correct ending of the trill-span!
  tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
}
```



Rectángulo de tres segmentos visibles

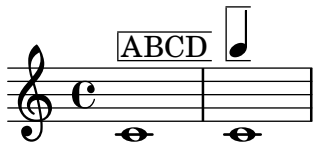
Este ejemplo muestra cómo añadir ina instrucción de marcado para obtener un rectángulo de tres segmentos visibles rodeando un elemento de texto (u otro elemento de marcado).

```
% New command to add a three-sided box, with sides north, west, and south.
% Based on the `box-stencil` command defined in `scm/stencil.scm`.
% Note that ";;" is used to comment a line in Scheme.
#(define-public (NWS-box-stencil stencil thickness padding)
   "Add a box around STENCIL, producing a new stencil."
   (let* ((x-ext (interval-widen (ly:stencil-extent stencil X) padding))
          (y-ext (interval-widen (ly:stencil-extent stencil Y) padding))
          (y-rule (make-filled-box-stencil (cons 0 thickness) y-ext))
          (x-rule (make-filled-box-stencil
                   (interval-widen x-ext thickness) (cons 0 thickness))))
     ;; (set! stencil (ly:stencil-combine-at-edge stencil X 1 y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil X LEFT y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil Y UP x-rule 0.0))
     (set! stencil (ly:stencil-combine-at-edge stencil Y DOWN x-rule 0.0))
     stencil))

% The corresponding markup command, based on the `\box` command defined
% in `scm/define-markup-commands.scm`.
#(define-markup-command (NWS-box layout props arg) (markup?)
   #:properties ((thickness 0.1) (font-size 0) (box-padding 0.2))
   "Draw a box round ARG.

Look at THICKNESS, BOX-PADDING, and FONT-SIZE properties to determine
line thickness and padding around the markup."
   (let ((pad (* (magstep font-size) box-padding))
         (m (interpret-markup layout props arg)))
     (NWS-box-stencil m thickness pad)))
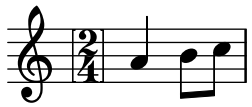

\relative c' {
  c1^\markup { \NWS-box ABCD }
  c1^\markup { \NWS-box \note {4} #1.0 }
}
```



Indicación de compás entre corchetes

La indicación de compás se puede encerrar entre corchetes.
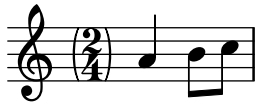
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Compás entre paréntesis

Se puede encerrar la indicación de compás entre paréntesis.
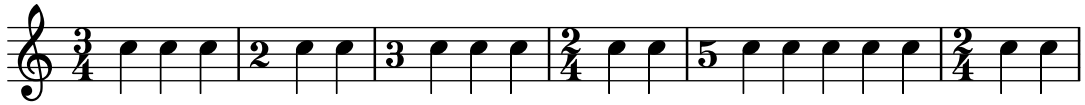
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Indicación de compás imprimiendo solo el numerador (en lugar de la fracción)

A veces, la indicación de compás no debe imprimir la fracción completa (p.ej. 7/4), sino solo el numerador (el dígito 7 en este caso). Esto se puede hacer fácilmente utilizando `\override Staff.TimeSignature.style = #'single-number` para cambiar el estilo permanentemente. Usando `\revert Staff.TimeSignature.style`, se puede revertir el cambio. Para aplicar el estilo de un número único a una sola indicación de compás, use `\tweak`.

```
\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-number
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-number style only for the next time signature
  \tweak style #'single-number \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```
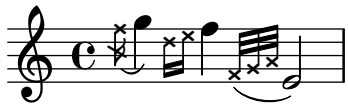


Trucar la disposición de las notas de adorno dentro de la música

La disposición de las expresiones de adorno se puede cambiar usando las funciones `add-grace-property` y `remove-grace-property`.

El ejemplo siguiente borra la definición de la propiedad `direction` (dirección) de los objetos gráficos `Stem` (plicas) para esta nota de adorno, de manera que las plicas no siemmpre apuntan hacia arriba, y cambia la forma predeterminada de las cabezas a aspas.

```
\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
       \acciaccatura { f16 } g4
       \grace { d16 e } f4
       \appoggiatura { f,32 g a } e2
    }
  }
}
```
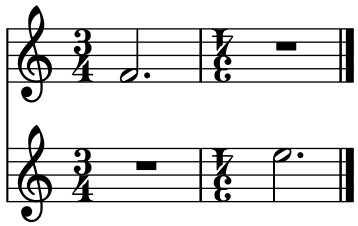


Indicaciones de compás personalizadas

Se pueden definir indicaciones de compás nuevas. La indicación de compás del segundo compás se imprime en posición invertida en los dos pentagramas.

```
#(add-simple-time-signature-style 'topsy-turvy
   (lambda (fraction)
     (make-rotate-markup 180 (make-compound-meter-markup fraction))))

<<
  \new Staff {
    \time 3/4 f'2.
    \override Score.TimeSignature.style = #'topsy-turvy
    \time 3/4 R2. \bar "|."
  }
  \new Staff {
    R2. e''
  }
>>
```



Uso de estilos alternativos para los corchetes

Se pueden imprimir formas alternativos del corchete o gancho de las corcheas y figuras menores, mediante la sobreescritura de la propiedad `stencil` del objeto `Flag`. LilyPond porporciona las siguientes funciones: `modern-straight-flag`, `old-straight-flag` y `flat-flag`. Utilice `\revert` para recuperar la forma predeterminada.

Para obtener corchetes más compactos o apilados verticalmente, llame a la instrucción `\flagStyleStacked`, que se puede anular después con `\flagStyleDefault`.

La sobreescritura de la forma de Flag no modifica la manera en que los corchetes se posicionan verticalmente. Esto se nota especialmente en los corchetes rectos: LilyPond no ajusta dinámicamente los huecos verticales entre los corchetes individuales de la misma forma que con las barras de corchea. Una posible solución para armonizar la apariencia es sustituir los corchetes planos por medias barras, como se ve en el segundo pentagrama; sin embargo, esto no se puede hacer automáticamente. En el código de este fragmento, las medias barras se escriben con el prefijo @, por ejemplo @c8.

Tenga en cuenta que las medias barras *no* son objetos gráficos del tipo Flag. Esto significa, particularmente, que la modificación de las propiedades de Flag no van a tener ningún efecto sobre ellos (tendríamos que usar en su lugar las propiedades del objeto Beam), y las propiedades para su objeto Stem asociado se comportarán también como barras.

```
"@" =
#(define-music-function (music) (ly:music?)
  #{ \set stemLeftBeamCount = 0 $music [] #})

testnotes = {
  \autoBeamOff
  c8 d16 e''32 f64 \acciaccatura { g,,,8 } a128 b
}

\relative c' {
  \override TextScript.staff-padding = 6
  \time 1/4
    <>^"default" \testnotes
  \override Flag.stencil = #modern-straight-flag
    <>_"modern straight" \testnotes
  \override Flag.stencil = #old-straight-flag
    <>^"old straight" \testnotes
  \override Flag.stencil = #flat-flag
    <>_"flat" \testnotes
  \revert Flag.stencil

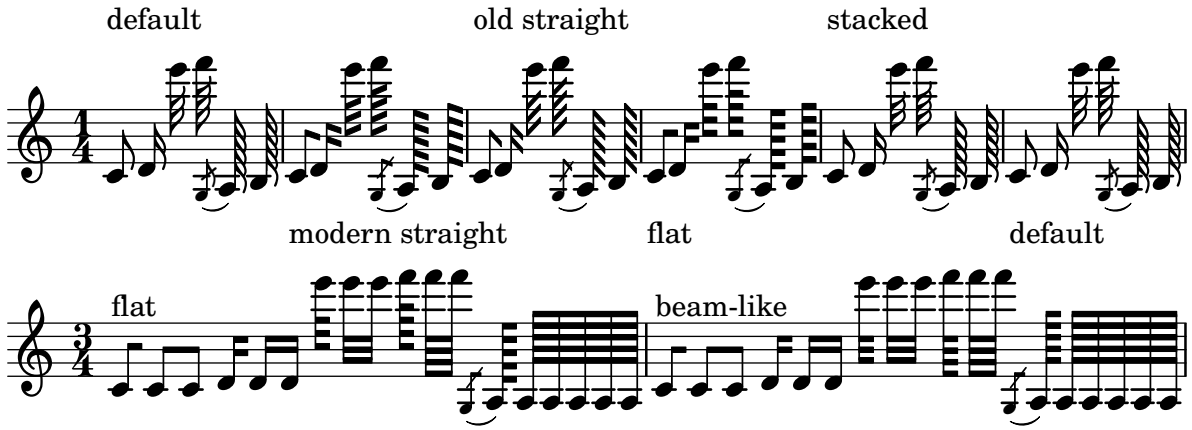
  \flagStyleStacked
    <>^"stacked" \testnotes
  \flagStyleDefault
    <>_"default" \testnotes
}

\relative c' {
  \time 3/4
  \override Flag.stencil = #flat-flag

  <>^"flat" c8 c[ c] d16 d[ d] e''32 e[ e] f64 f[ f]
    \acciaccatura { g,,,8 } a128 a[ a a a a]
  <>^"beam-like" @c8 c[ c] @d16 d[ d] @e''32 e[ e] @f64 f[ f]
    \acciaccatura { g,,,8 } @a128 a[ a a a a]
}

\layout {
  indent = 0
  \context {
```
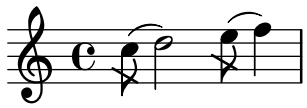
```
    \Score
    \override NonMusicalPaperColumn.line-break-permission = ##f
  }
}
```



Utilizar la barra que tacha las notas de adorno con notas normales

Es posible aplicar la barrita que cruza la barra de las acciaccaturas, en otras situaciones.

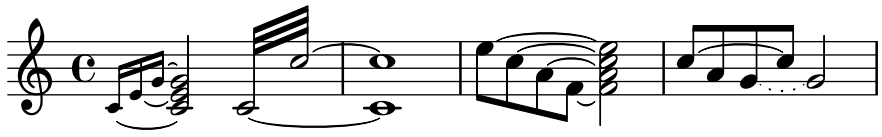
```
\relative c'' {
  \override Flag.stroke-style = "grace"
  c8( d2) e8( f4)
}
```



Uso de ligaduras en los arpegios

En ocasiones se usan ligaduras de unión para escribir los arpegios. En este caso, las dos notas ligadas no tienen que ser consecutivas. Esto se puede conseguir estableciendo la propiedad `tieWaitForNote` al valor `#t`. La misma funcionalidad es de utilidad, por ejemplo, para ligar un trémolo a un acorde, pero en principio también se puede usar para notas normales consecutivas.

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```
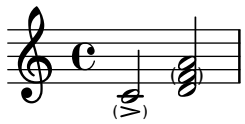


3 Expressive marks

See also Sección “Expressive marks” en *Notation Reference*.

Encerrar entre paréntesis una marca expresiva o una nota de un acorde

La función \parenthesize es un truco especial que encierra objetos entre paréntesis. El grob asociado es Parentheses.

```
\relative c' {
  c2-\parenthesize ->
  \override Parentheses.padding = #0.1
  \override Parentheses.font-size = #-4
  <d \parenthesize f a>2
}
```



Añadir marcas de tiempo a glissandos largos
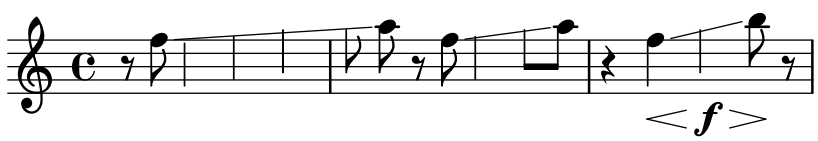
Los pulsos que se saltan en glissandos muy largos se indican a veces mediante marcas de tiempo que consisten en figuras sin cabeza. Estas plicas se pueden usar también para albergar indicaciones expresivas intermedias.

Si las plicas no quedan bien alineadas con el glissando, podría ser necesario recolocarlas ligeramente.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c'' {
  r8 f8\glissando \glissandoSkipOn f4 g a |
  a8\noBeam \glissandoSkipOff a8
    r8 f8\glissando \glissandoSkipOn g4 a8 \glissandoSkipOff a8 |
  r4 f\glissando\< \glissandoSkipOn a4\f\> \glissandoSkipOff b8\! r |
}
```



Ajustar las posiciones de las ligaduras verticalmente
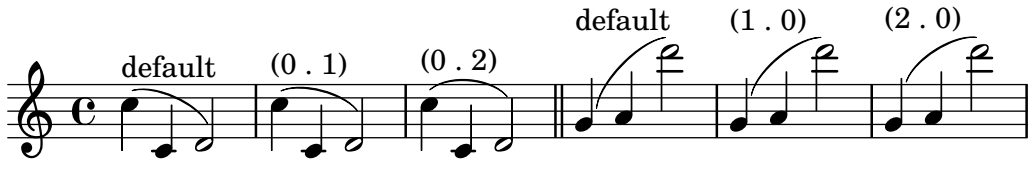
Usando `\override Slur.positions` es posible fijar la posición vertical de los puntos de inicio y de fin de una ligadura de expresión a valores absolutos (o mejor dicho, forzar al algoritmo de ligaduras de LilyPond para que tenga en cuenta estos valores según se desee). En muchos casos, esto implica bastante ensayo y error hasta que se encuentran unos valores aceptables. Porblablemente habrá probado la instrucción `\offset` solo para comprobar que no funciona con las ligaduras de expresión, sino que emite una advertencia en su lugar.

El código de este fragmento nos permite trucar las posiciones de comienzo y final especificando cambios *relativos*, de manera parecida a `\offset`.

La sintaxis es: `\offsetPositions #'(`*dy1 . dy2*`)`

```
offsetPositions =
#(define-music-function (offsets) (number-pair?)
  #{
     \once \override Slur.control-points =
       #(lambda (grob)
          (match-let ((((_ . y1) _ _ (_ . y2))
                       (ly:slur::calc-control-points grob))
                      ((off1 . off2) offsets))
            (set! (ly:grob-property grob 'positions)
                  (cons (+ y1 off1) (+ y2 off2)))
            (ly:slur::calc-control-points grob)))
  #})

\relative c'' {
  c4(^"default" c, d2)
  \offsetPositions #'(0 . 1)
  c'4(^"(0 . 1)" c, d2)
  \offsetPositions #'(0 . 2)
  c'4(^"(0 . 2)" c, d2)
  \bar "||"
  g4(^"default" a d'2)
  \offsetPositions #'(1 . 0)
  g,,4(^"(1 . 0)" a d'2)
  \offsetPositions #'(2 . 0)
  g,,4(^"(2 . 0)" a d'2)
}
```



Ajustar la forma de las subidas y caídas de tono

Puede ser necesario trucar la propiedad `shortest-duration-space` para poder ajustar el tamaño de las caídas y subidas de tono («falls» y «doits»).

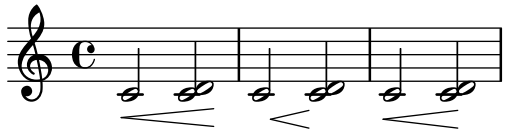
```
\relative c'' {
  \override Score.SpacingSpanner.shortest-duration-space = 4.0
  c2-\bendAfter 5
  c2-\bendAfter -4.75
  c2-\bendAfter 8.5
  c2-\bendAfter -6
}
```



Alinear los extremos de los reguladores con las direcciones de NoteColumn

Se pueden alinear los extremos de los reguladores con los puntos izquierdo, central o derecho de los objetos gráficos NoteColumn por medio de la sobreescritura de la propiedad endpoint-alignments, que es una pareja de números que representan los extremos izquierdo y derecho de un regulador. Se espera que los endpoint-alignments sean direcciones (-1, 0 o 1). Cualquier otro valor se transforma acompañado de una advertencia. La derecha de un regulador que termina en un silencio no resulta afectada, terminando siempre en el borde izquierdo del silencio.

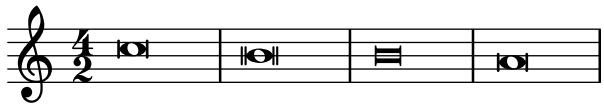
```
{
  c'2\< <c' d'>\! |
  \override Hairpin.endpoint-alignments = #'(1 . -1)
  c'2\< <c' d'>\! |
  \override Hairpin.endpoint-alignments = #`(,LEFT . ,CENTER)
  c'2\< <c' d'>\! |
}
```



Formas alternativas de la figura breve

Las figuras de breve también están disponibles con dos líneas verticales a los lados de la cabeza en lugar de una sola línea y la forma en estilo barroco.

```
\relative c'' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}
```

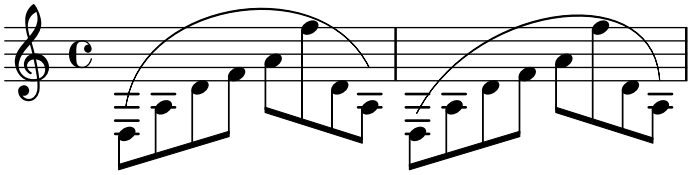


Ligaduras asimétricas

Se puede hacer que una ligadura de expresión sea asimétrica para adaptarse mejor a un patrón asimétrico de notas.

```
slurNotes = { d,8( a' d f a f' d, a) }

\relative c' {
  \stemDown
  \slurUp
  \slurNotes
  \once \override Slur.eccentricity = #3.0
  \slurNotes
}
```



Marcas de respiración

Las marcas de respiración están disponibles para varios gustos: coma (la predeterminada), raya corta, uve y "vías del tren" (cesura).
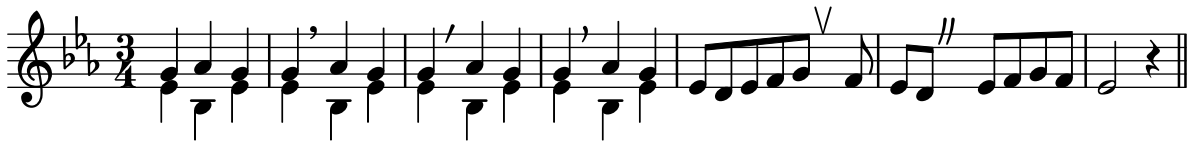
```
\new Staff \relative c'' {
  \key es \major
  \time 3/4
  % this bar contains no \breathe
  << { g4 as g } \\ { es4 bes es } >> |
  % Modern notation:
  % by default, \breathe uses the rcomma, just as if saying:
  % \override BreathingSign.text =
  %   #(make-musicglyph-markup "scripts.rcomma")
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % rvarcomma and lvarcomma are variations of the default rcomma
  % and lcomma
  % N.B.: must use Staff context here, since we start a Voice below
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % raltcomma and laltcomma are alternative variations of the
  % default rcomma and lcomma
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.raltcomma" }
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % vee
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.uupbow" }
  es8[ d es f g] \breathe f |
```

```
  % caesura
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.caesura.curved" }
  es8[ d] \breathe es[ f g f] |
  es2 r4 \bar "||"
}
```
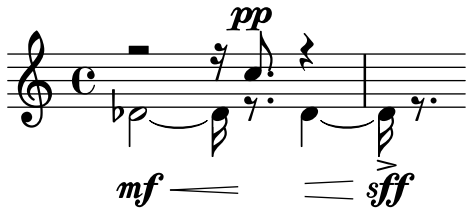


Regulador interrumpido

Para hacer invisibles partes de un regulador de crescendo, se usa el método de dibujar un rectángulo blanco encima de la parte respectiva del regulador, tapándola. El rectángulo se define como un elemento de marcado de texto.

La instrucción de marcado `with-dimensions` indica a LilyPond que considere solamente el extremo inferior del rectángulo cuando realice el espaciado de éste frente al regulador. La propiedad `staff-padding` evita que el rectángulo quepa entre el regulador y el pentagrama.

Asegúrse de que el regulador está en una capa más baja que el elemento de marcado de texto para trazar el rectángulo encima del regulador.

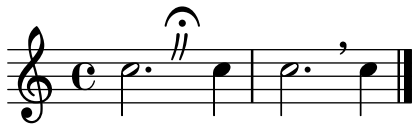
```
\relative c' {
  <<
    {
      \dynamicUp
      r2 r16 c'8.\pp r4
    }
    \\
    {
      \override DynamicLineSpanner.layer = #0
      des,2\mf\< ~
      \override TextScript.layer = #2
      \once\override TextScript.staff-padding = #6
      \once\override TextScript.vertical-skylines = #'()
      des16_\markup \with-dimensions #'(2 . 7) #'(0 . 0)
                    \with-color #white
                    \filled-box #'(2 . 7) #'(0 . 2) #0
      r8. des4 ~ des16->\sff r8.
    }
  >>
}
```



Caesura ("railtracks") with fermata

A caesura is sometimes denoted by a double "railtracks" breath mark with a fermata sign positioned above. This snippet shows an optically pleasing combination of railtracks and fermata.

```
\relative c'' {
  c2.
  % construct the symbol
  \override BreathingSign.text = \markup {
    \override #'(direction . 1)
    \override #'(baseline-skip . 1.8)
    \dir-column {
      \translate #'(0.155 . 0)
        \center-align \musicglyph "scripts.caesura.curved"
      \center-align \musicglyph "scripts.ufermata"
    }
  }
  \breathe c4
  % set the breath mark back to normal
  \revert BreathingSign.text
  c2. \breathe c4
  \bar "|."
}
```



Centrar texto debajo de un regulador

Este ejemplo proporciona una función para tipografiar un regulador con texto por debajo, como "molto" o "poco". El texto añadido cambia la dirección de acuerdo con la del regulador. El objeto Hairpin está alineado con un objeto gráfico DynamicText.

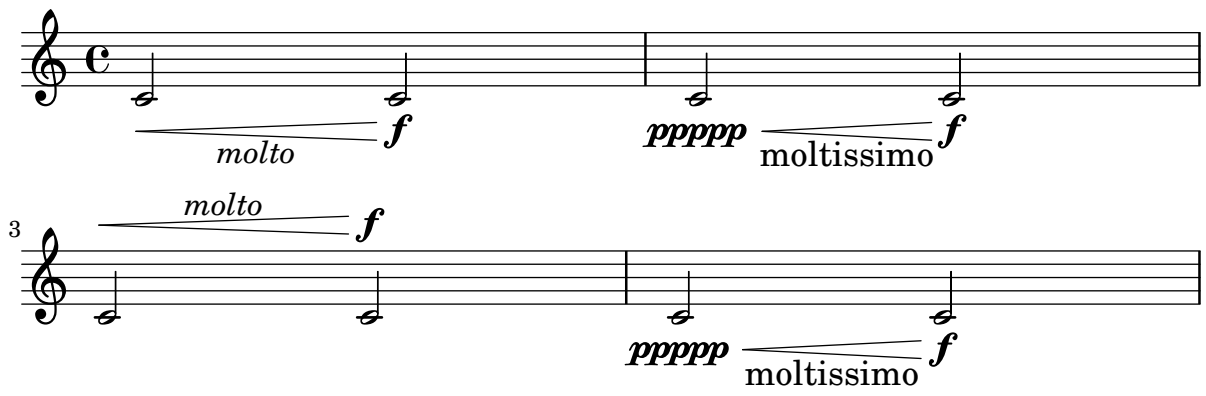
El ejemplo ilustra también cómo modificar la manera en que se imprime normalmente un objeto, utilizando código de Scheme.

```
hairpinWithCenteredText =
#(define-music-function (text) (markup?)
  #{
    \once \override Voice.Hairpin.after-line-breaking =
      #(lambda (grob)
         (let* ((stencil (ly:hairpin::print grob))
                (par-y (ly:grob-parent grob Y))
                (dir (ly:grob-property par-y 'direction))
                (staff-line-thickness
                 (ly:output-def-lookup (ly:grob-layout grob)
                                       'line-thickness))
                (new-stencil
                 (ly:stencil-aligned-to
                  (ly:stencil-combine-at-edge
                   (ly:stencil-aligned-to stencil X CENTER)
                   Y dir
                   (ly:stencil-aligned-to
                    (grob-interpret-markup
```

```
                    grob
                    (make-fontsize-markup
                     (magnification->font-size
                      (+ (ly:staff-symbol-staff-space grob)
                         (/ staff-line-thickness 2)))
                     text))
                   X CENTER))
                 X LEFT))
               (staff-space (ly:output-def-lookup
                (ly:grob-layout grob) 'staff-space))
               (par-x (ly:grob-parent grob X))
               (dyn-text (grob::has-interface par-x
                                              'dynamic-text-interface))
               (dyn-text-stencil-x-length
                (if dyn-text
                    (interval-length
                     (ly:stencil-extent
                      (ly:grob-property par-x 'stencil) X))
                    0))
               (x-shift
                (if dyn-text (- (+ staff-space dyn-text-stencil-x-length)
                                (* 0.5 staff-line-thickness))
                    0)))
          (ly:grob-set-property! grob 'Y-offset 0)
          (ly:grob-set-property! grob
                                 'stencil (ly:stencil-translate-axis
                                           new-stencil
                                           x-shift X))))
  #})

hairpinMolto = \hairpinWithCenteredText \markup { \italic molto }
hairpinMore = \hairpinWithCenteredText \markup { \larger moltissimo }

\relative c' {
  \hairpinMolto c2\< c\f
  \hairpinMore c2\ppppp\< c\f
  \break
  \hairpinMolto c2^\< c\f
  \hairpinMore c2\ppppp\< c\f
}
```
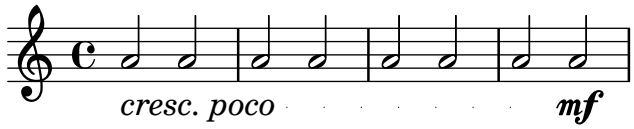


Cambiar el texto y los estilos de objeto de extensión para las indicaciones dinámicas textuales

Se puede modificar el texto empleado para los *crescendos* y *decrescendos* modificando las propiedades de contexto `crescendoText` y `decrescendoText`.

El estilo de la línea de extensión se puede cambiar modificando la propiedad `'style` de `DynamicTextSpanner`. El valor predeterminado es `dashed-line` (línea discontinua), y entre otros valores posibles se encuentran `line` (línea), `dotted-line` (línea de puntos) y `none` (nada):
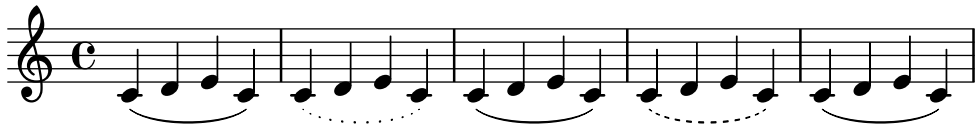
```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Modificar el aspecto continuo de una ligadura de expresión a punteado o intermitente

Se puede cambiar el aspecto de las ligaduras de expresión de continuas a punteadas o intermitentes.
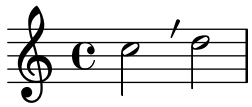
```
\relative c' {
  c4( d e c)
  \slurDotted
  c4( d e c)
  \slurSolid
  c4( d e c)
  \slurDashed
  c4( d e c)
  \slurSolid
  c4( d e c)
}
```



Cambiar el símbolo de la marca de respiración

El glifo de la marca de respiración se puede ajustar sobreescribiendo la propiedad de texto del objeto de presentación `BreathingSign`, con cualquier otro texto de marcado.
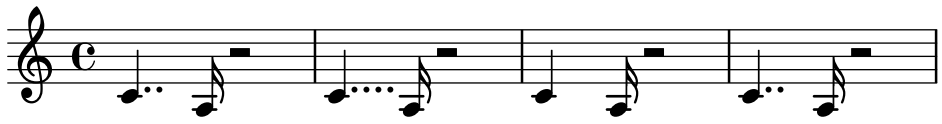
```
\relative c'' {
  c2
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  \breathe
  d2
}
```



Modificar el número de puntillos de una nota

La cantidad de puntillos de una nota se puede sobreescribir mediante el establecimiento de la propiedad `dot-count` del objeto gráfico `Dots`.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = 4
  c4.. a16 r2 |
  \override Dots.dot-count = 0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```
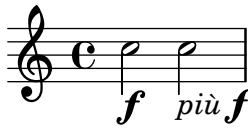


Combinar indicaciones dinámicas con marcados textuales

Ciertas indicaciones dinámicas pueden llevar textos (como "più *f*" o "*p* subito"). Se pueden producir usando un bloque `\markup`; el objeto resultante se comporta como un objeto gráfico `TextScript`.

See also "Combining dynamics with markup texts (2)".

```
piuF = \markup { \italic più \dynamic f }

\score {
  \relative c'' {
    c2\f c-\piuF
  }
}
```
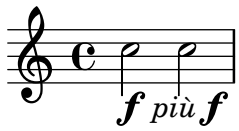


Combining dynamics with markup texts (2)

Some dynamics may involve text indications (such as "più *f*" or "*p* subito"). These can be produced using the `make-dynamic-script` Scheme function; the resulting object behaves like a `DynamicText` grob.

See also “Combining dynamics with markup texts”.

```
piuF = #(make-dynamic-script
          #{ \markup { \normal-text \italic più \dynamic f } #})
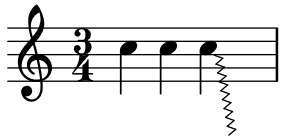
\score {
  \relative c'' {
    c2\f c\piuF
  }
}
```



Glissando contemporáneo

Se puede tipografiar un gissando contemporáneo sin nota final utilizando una nota oculta y temporalización de cadenza.

```
\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}
```



Controlar la visibilidad de los objetos de extensión después de un salto de línea

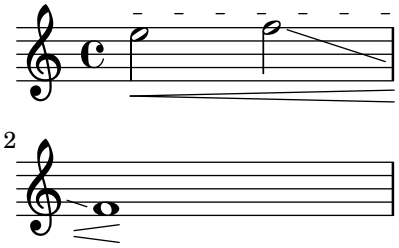
La visibilidad de los objetos de extensión que acaban en la primera nota después de un salto de línea está controlada por la función de callback de `after-line-breaking` `ly:spanner::kill-zero-spanned-time`.

Para los objetos como los glissandos y los reguladores, el comportamiento predeterminado es ocultar el objeto de extensión después del salto; la inhabilitación de la función de callback hace que el objeto de extensión roto por la izquierda pueda mostrarse.

De forma inversa, los objetos de extensión que son visibles normalmente, como los objetos de extensión de texto, se pueden ocultar habilitando la función de callback.

```
\paper {
  line-width = 50\mm
}

\relative c'' {
  \override Hairpin.to-barline = ##f
  \override Glissando.breakable = ##t
  % show hairpin
  \override Hairpin.after-line-breaking = ##t
  % hide text span
  \override TextSpanner.after-line-breaking =
    #ly:spanner::kill-zero-spanned-time
  e2\<\startTextSpan
  % show glissando
  \override Glissando.after-line-breaking = ##t
  f2\glissando
  \break
  f,1\!\stopTextSpan
}
```



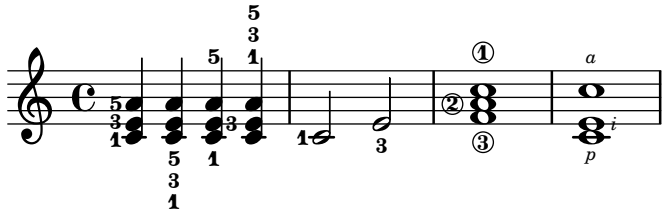
Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación usando la propiedad `fingeringOrientation`. Para que se tenga en cuenta la orientación de las digitaciones, la instrucción de digitación se debe utilizar dentro de una construcción de acorde (`<...>`) aunque sea una sola nota. Se puede controlar la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar usando las propiedades `stringNumberOrientation` y `strokeFingerOrientation`, respectivamente.

Estas propiedades se pueden establecer a una lista de entre uno y tres valores. Controlan si las digitaciones se pueden situar por encima (si aparece `up` en la lista), por debajo (si aparece `down`), a la izquierda (si aparece `left`), o a la derecha (si aparece `right`). Por contra, si una localización no aparece en la lista, no se coloca ninguna digitadión en ese lugar. LilyPond admite esas restricciones y trata de encontrar la mejor ubicación para las digitaciones de las notas de los acordes siguientes. Observe que `left` y `right` se excluyen mutuamente (las digitaciones se pueden colocar en un lado o en el otro, no en los dos).

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
```

```
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Controlar la ordenación vertical de las inscripciones

El orden vertical que ocupan las inscripciones gráficas está controlado con la propiedad `script-priority`. Cuanto más bajo es este número, más cerca de la nota se colocará. En este ejemplo, el `TextScript` (el símbolo del *sostenido*) tiene primero la prioridad más baja, por lo que se sitúa en la posición más baja en el primer ejemplo. En el segundo, el `Script` (el símbolo del *semitrino*) es el que la tiene más baja, por lo que se sitúa en la parte interior. Cuando dos objetos tienen la misma prioridad, el orden en que se introducen determina cuál será el que aparece en primer lugar.

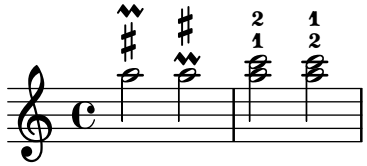
Observe que para los objetos gráficos `Fingering`, `StringNumber` y `StrokeFinger`, si se usan dentro de un acorde, el orden vertical también viene determinado por la posición vertical de la cabeza de la nota asociada, que se añade a (o, dependiendo de la dirección, se sustrae de) el valor `script-priority` del objeto gráfico. Esto nos asegura que para las digitaciones que están por encima de un acorde, la nota inferior está asociada con la digitación inferior (y viceversa para la otra dirección); no importa si escribimos las notas del acorde de arriba a abajo o de abajo a arriba.

De manera predeterminada, las inscripciones de carácter menos técnico se sitúan más cerca de la cabeza de la nota; el orden es más o menos: articulación, armónico, digitación, digitación de la mano derecha, número de cuerda, calderón, indicaciones de arco, e inscripción de texto.

```
\relative c''' {
  \once \override TextScript.script-priority = -100
  a2^\prall^\markup { \sharp }

  \once \override Script.script-priority = -100
  a2^\prall^\markup { \sharp }

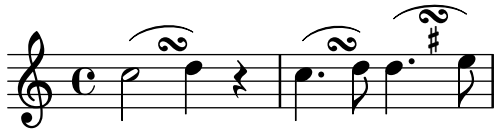
  \set fingeringOrientations = #'(up)
  <c-2 a-1>2
  <a-1 c\tweak script-priority -100 -2>2
}
```



Crear un grupeto de anticipación

La creación de un grupeto circular de anticipación entre dos notas, donde la nota inferior del grupeto utiliza una alteración, requiere varias sobreescrituras de propiedades. La propiedad `outside-staff-priority` se debe establecer al valor `#f`, pues en caso contrario tendría prioridad sobre la propiedad `avoid-slur`. Cambiando el primer argumento de `\after` (que es una duración) puede ajustarse la posición horizontal.
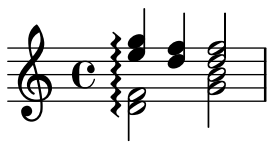
```
\relative c'' {
  \after 2*2/3 \turn c2( d4) r |
  \after 4 \turn c4.( d8)
  \after 4
  {
    \once \set suggestAccidentals = ##t
    \once \override AccidentalSuggestion.outside-staff-priority = ##f
    \once \override AccidentalSuggestion.avoid-slur = #'inside
    \once \override AccidentalSuggestion.font-size = -3
    \once \override AccidentalSuggestion.script-priority = -1
    \once \hideNotes
    cis8\turn \noBeam
  }
  d4.( e8)
}
```



Crear arpegios entre notas de voces distintas

Se puede trazar un símbolo de *arpeggio* entre notas de distintas voces que están sobre el mismo pentagrama si el grabador `Span_arpeggio_engraver` se traslada al contexto de `Staff`.

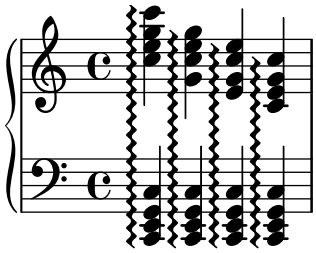
```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Crear arpegios que se cruzan entre pentagramas dentro de un sistema de piano

Dentro de un `PianoStaff`, es posible hacer que un arpegio cruce entre los pentagramas ajustando la propiedad `PianoStaff.connectArpeggios`.

```
\new PianoStaff \relative c'' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
  }
  \new Staff {
    \clef bass
    \*4 <c,, e g c>4\arpeggio
  }
>>
```



Creación de arpegios que se cruzan entre pentagramas dentro de otros contextos

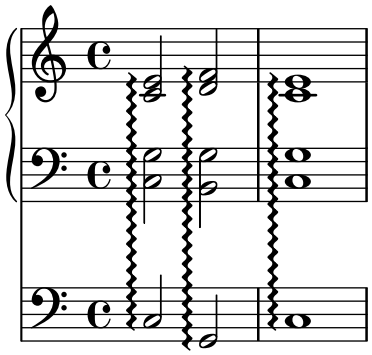
Se pueden crear arpegios que se cruzan entre pentagramas dentro de contextos distintos de `GrandStaff` y sus parientes derivados `PianoStaff`, `ChoirStaff` y `StaffGroup` si se incluye el grabador `Span_arpeggio_engraver` en el contexto de `Score`.

```
<<
  \new PianoStaff <<
    \new Voice \relative c' {
      <c e>2\arpeggio <d f>2\arpeggio
      <c e>1\arpeggio
    }
    \new Voice \relative c {
      \clef bass
      <c g'>2\arpeggio <b g'>2\arpeggio
      <c g'>1\arpeggio
    }
  >>

  \new Staff \relative c {
    \set Score.connectArpeggios = ##t
    \clef bass
    c2\arpeggio g\arpeggio
    c1\arpeggio
  }
>>

\layout {
  \context {
    \Score
```

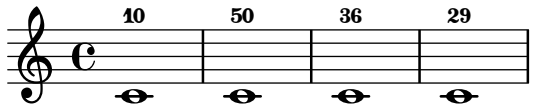
```
    \consists "Span_arpeggio_engraver"
  }
}
```



Crear digitaciones de dos cifras

Es posible crear digitaciones con un número mayor de 5.

```
\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}
```



Hacer ligaduras entre voces distintas
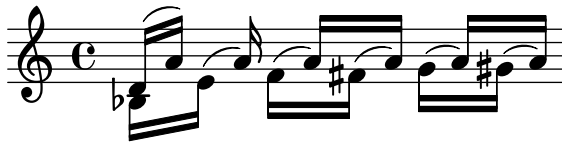
En determinadas situaciones es necesario crear ligaduras de expresión entre notas que están en voces distintas.

La solución es añadir notas invisibles a una de las voces utilizando `\hideNotes`.

Este ejemplo es el compás 235 de la Chacona de la segunda Partita para violín solo, BWV 1004, de Bach.

```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
```

```
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```



Crear elementos de extensión textuales

Las instrucciones \startTextSpan y \stopTextSpan permiten la creación de elementos de extensión textuales tan fácilmente como indicaciones de pedal u octavaciones. Sobreescribimos ciertas propiedades del objeto TextSpanner para modificar su salida.

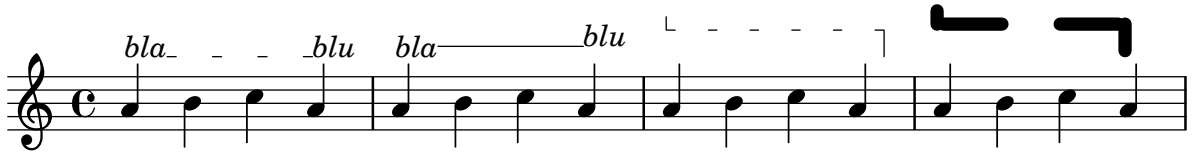
```
\paper { ragged-right = ##f }

\relative c'' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.bound-details.left.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . -2) }
  \once \override TextSpanner.bound-details.right.padding = #-2
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.dash-period = #10
  \override TextSpanner.dash-fraction = #0.5
  \override TextSpanner.thickness = #10
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}
```
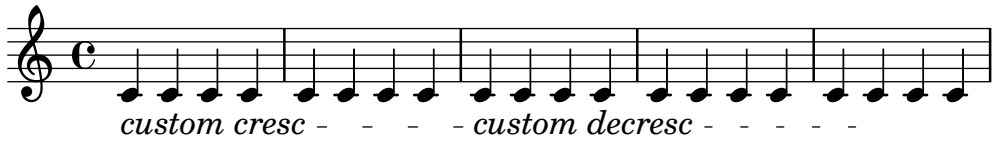


Dynamics spanner with custom text

Postfix functions for custom crescendo text spanners. The spanners should start on the first note of the measure. One has to use -\mycresc, otherwise the spanner start will rather be assigned to the next note.

```
% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.
mycresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'CrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))
mydecresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'DecrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4 c4\! c4 c4
}
```



Glissando por debajo de un objeto gráfico

Los objetos gráficos de columna de nota (los grobs NoteColumn) pueden ser sobrepasados por los glissandos.

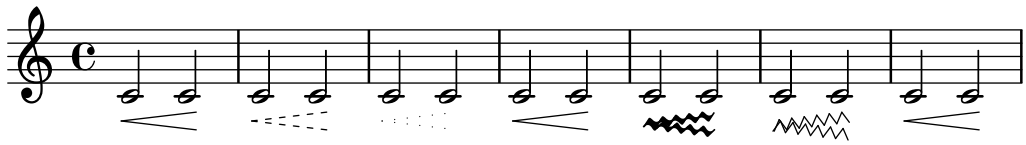
```
\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}
```



Reguladores con distintos estilos de línea

Los reguladores pueden imprimirse en uno cualquiera de los estilos de line-interface: dashed-line (discontinuo), dotted-line (punteado), line (línea continua), trill (trino) o zigzag (zig-zag).

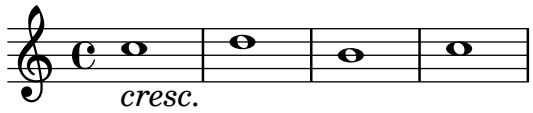
```
\relative c' {
  c2\< c\!
  \override Hairpin.style = #'dashed-line
  c2\< c\!
  \override Hairpin.style = #'dotted-line
  c2\< c\!
  \override Hairpin.style = #'line
  c2\< c\!
  \override Hairpin.style = #'trill
  c2\< c\!
  \override Hairpin.style = #'zigzag
  c2\< c\!
  \revert Hairpin.style
  c2\< c\!
}
```



Ocultar la línea de extensión de las expresiones textuales de dinámica

Los cambios de dinámica con estilo de texto (como *cresc.* y *dim.*) se imprimen con una línea intermitente que muestra su alcance. Esta línea se puede suprimir de la siguiente manera:

```
\relative c'' {
  \override DynamicTextSpanner.style = #'none
  \crescTextCresc
  c1\< | d | b | c\!
}
```



Horizontally aligning custom dynamics like “più f”

Some dynamic expressions involve additional text, like “sempre **pp**”. Since dynamics are usually centered under the note, the \pp would be displayed way after the note it applies to.

To correctly align the “sempre **pp**” horizontally so that it is aligned as if it were only the \pp, there are several approaches:

- Simply use `\once \override DynamicText.X-offset = #-9.2` before the note with the dynamics to manually shift it to the correct position. Drawback: This has to be done manually each time you use that dynamic markup...
- Add some padding (`#:hspace 7.1`) into the definition of your custom dynamic mark so that after LilyPond center-aligns it, it is already correctly aligned. Drawback: The padding really takes up that space and does not allow any other markup or dynamics to be shown in that position.
- Shift the dynamic script `\once \override ... .X-offset = ....` Drawback: `\once \override` is needed for every invocation!

- Set the dimensions of the additional text to 0 (using `#:with-dimensions '(0 . 0) '(0 . 0)`). Drawback: For LilyPond, "sempre" has no extent now. This means it might put other stuff there, causing collisions (which are not detected by LilyPond's collision detection algorithm!). There also seems to be some spacing, so it is not exactly the same alignment as without the additional text.
- Add an explicit shift directly inside the scheme function for the dynamic script.
- Set an explicit alignment inside the dynamic script. By default, this won't have any effect, only if one sets `X-offset`! Drawback: One needs to set `DynamicText.X-offset`, which will apply to all dynamic texts! Also, it is aligned at the right edge of the additional text, not at the center of \pp.

```
\paper {
  ragged-right = ##f
  indent = 5\cm
}

% Solution 1: Using a simple markup with a particular halign value
% Drawback: It's a markup, not a dynamic command, so \dynamicDown
%           etc. will have no effect
semppMarkup = \markup { \halign #1.4 \italic "sempre" \dynamic "pp" }

% Solution 2: Using a dynamic script & shifting with
%             \once \override ...X-offset = ..
% Drawback: \once \override needed for every invocation
semppK =
#(make-dynamic-script
  (markup #:line
          (#:normal-text
           #:italic "sempre"
           #:dynamic "pp")))

% Solution 3: Padding the dynamic script so the center-alignment
%             puts it at the correct position
% Drawback: the padding really reserves the space, nothing else can be there
semppT =
#(make-dynamic-script
  (markup #:line
          (#:normal-text
           #:italic "sempre"
           #:dynamic "pp"
           #:hspace 7.1)))

% Solution 4: Dynamic, setting the dimensions of the additional text to 0
% Drawback: To lilypond "sempre" has no extent, so it might put
%           other stuff there => collisions
% Drawback: Also, there seems to be some spacing, so it's not exactly the
%           same alignment as without the additional text
semppM =
#(make-dynamic-script
  (markup #:line
          (#:with-dimensions '(0 . 0) '(0 . 0)
                             #:right-align
```

```
                              #:normal-text
                              #:italic "sempre"
                              #:dynamic "pp")))

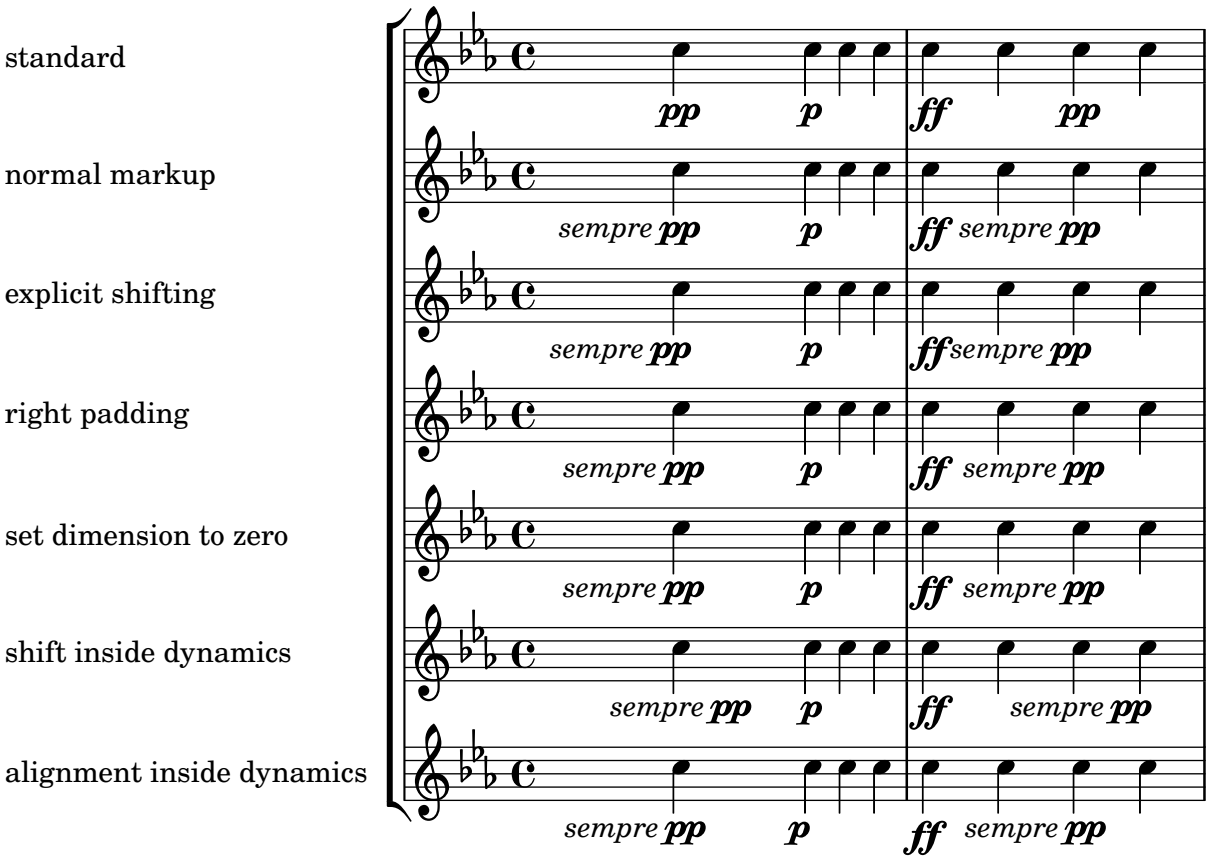
% Solution 5: Dynamic with explicit shifting inside the scheme function
semppG =
#(make-dynamic-script
  (markup #:hspace 0
          #:translate '(-18.85 . 0)
          #:line (#:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

% Solution 6: Dynamic with explicit alignment. This has only effect
%             if one sets X-offset!
% Drawback: One needs to set DynamicText.X-offset!
% Drawback: Aligned at the right edge of the additional text,
%           not at the center of pp
semppMII =
#(make-dynamic-script
  (markup #:line (#:right-align
                  #:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

\new StaffGroup <<
  \new Staff \with { instrumentName = "standard" }
    \relative c'' {
      \key es \major
      c4\pp c\p c c | c\ff c c\pp c
    }
  \new Staff \with {instrumentName = "normal markup" }
    \relative c'' {
      \key es \major
      c4-\semppMarkup c\p c c | c\ff c c-\semppMarkup c
    }
  \new Staff \with { instrumentName = "explicit shifting" }
    \relative c'' {
      \key es \major
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c\p c c
      c4\ff c
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c
    }
  \new Staff \with { instrumentName = "right padding" }
    \relative c'' {
      \key es \major
      c4\semppT c\p c c | c\ff c c\semppT c
    }
  \new Staff \with { instrumentName = "set dimension to zero" }
    \relative c'' {
```

```
      \key es \major
      c4\semppM c\p c c | c\ff c c\semppM c
    }
  \new Staff \with { instrumentName = "shift inside dynamics" }
    \relative c'' {
      \key es \major
      c4\semppG c\p c c | c\ff c c\semppG c
    }
  \new Staff \with { instrumentName = "alignment inside dynamics" }
    \relative c'' {
      \key es \major
      \override DynamicText.X-offset = #-1
      c4\semppMII c\p c c | c\ff c c\semppMII c
    }
>>

\layout { \override Staff.InstrumentName.self-alignment-X = #LEFT }
```
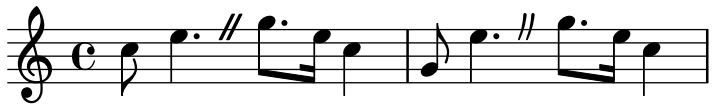


Insertar una cesura

Las marcas de cesura se pueden crear sobreescribiendo la propiedad `text` del objeto `BreathingSign`.

También está disponible una marca de cesura curva.

```
\relative c'' {
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```
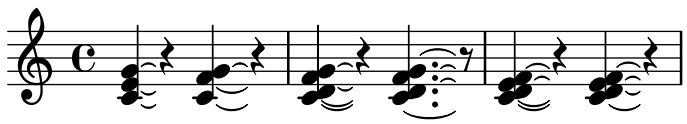


Ligaduras *laissez vibrer*

Las ligaduras "laissez vibrer" (dejar vibrar) tienen un tamaño fijo. Se puede ajustar su posición usando la propiedad `tie-configuration`.

Véase también el fragmento de código "Ligaduras laissez vibrer largas".

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
     = #`((-7 . ,DOWN)
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



Puntas de flecha para las líneas

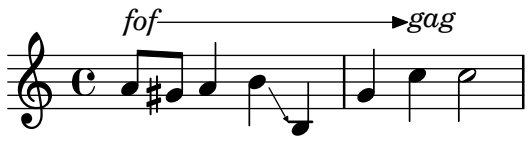
Se pueden aplicar puntas de flecha a los elementos de extensión de texto y de línea (como los glissandos).

```
\relative c'' {
  \override TextSpanner.bound-padding = #1.0
  \override TextSpanner.style = #'line
  \override TextSpanner.bound-details.right.arrow = ##t
  \override TextSpanner.bound-details.left.text = #"fof"
  \override TextSpanner.bound-details.right.text = #"gag"
  \override TextSpanner.bound-details.right.padding = #0.6

  \override TextSpanner.bound-details.right.stencil-align-dir-y = #CENTER
  \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
```

```
  \override Glissando.bound-details.right.arrow = ##t
  \override Glissando.arrow-length = #0.5
  \override Glissando.arrow-width = #0.25

  a8\startTextSpan gis a4 b\glissando b,
  g'4 c\stopTextSpan c2
}
```
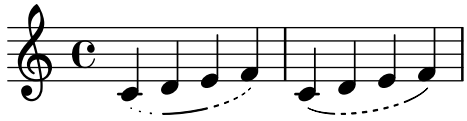


Hacer ligaduras de expresión con estructura compleja de discontinuidad

Las ligaduras de expresión se pueden construir con patrones de discontinuidad complejos mediante la definición de la propiedad `dash-definition`, que es una lista de segmentos de la ligadura, que a su vez son listas de parámetros que definen el comportamiento de discontinuidad del segmento dado.

Los segmentos de la ligadura se definen en términos del parámetro de bezier *t*, cuyo rango va de 0 a la izquierda de la ligadura hasta 1 en el extremo derecho de la ligadura. Cada segmento tiene la forma (*t-inicio t-final fracción-discontinuidad período-discontinuidad*). En el segmento que va desde *t-inicio* hasta *t-final*, el patrón se define por los valores de *fracción-discontinuidad* y *período-discontinuidad*. La *fracción-discontinuidad* especifica el período en que el trazo es continuo y de color negro. Si se establece al valor de 1 se obtiene un segmento de ligadura continuo. Las unidades de *período-discontinuidad* son espacios de pentagrama.

```
\relative c' {
  \once \override
    Slur.dash-definition = #'((  0  0.3  0.1 0.75)
                              (0.3  0.6  1   1   )
                              (0.65 1.0  0.4 0.75))
  c4( d e f)
  \once \override
    Slur.dash-definition = #'((0    0.25  1   1   )
                              (0.3  0.7   0.4 0.75)
                              (0.75 1.0   1   1   ))
  c4( d e f)
}
```



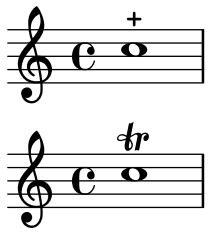
Modificar los valores predeterminados para la notación abreviada de las articulaciones

Las abreviaturas se encuentran definidas dentro del archivo `ly/script-init.ly`, donde las variables `dashHat`, `dashPlus`, `dashDash`, `dashBang`, `dashLarger`, `dashDot` y `dashUnderscore` reciben valores predeterminados. Se pueden modificar estos valores predeterminados para las abreviaturas. Por ejemplo, para asociar la abreviatura `-+` (`dashPlus`) con el símbolo del *semitrino* en lugar del símbolo predeterminado "+", asigne el valor `\trill` a la variable `dashPlus`:

```
\relative c'' { c1-+ }

dashPlus = \trill

\relative c'' { c1-+ }
```
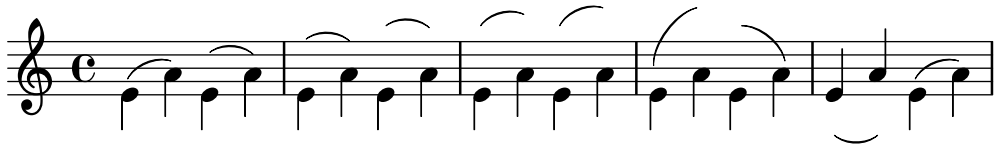


Desplazar ligaduras de expresión verticalmente

Se puede ajustar la posición vertical de una ligadura de expresión utilizando la propiedad `positions` del objeto `Slur`. La propiedad tiene dos parámetros, refiriéndose el primero al extremo izquierdo de la ligadura y el segundo al derecho. Los valores de los parámetros no se utilizan por parte de LilyPond para producir un desplazamiento exacto de la ligadura: más bien selecciona la colocación que mejor aspecto tiene, teniendo en cuenta los valores de los parámetros. Los valores positivos desplazan la ligadura hacia arriba, y son adecuados a notas que tienen las plicas hacia abajo. Los valores negativos desplazan las ligaduras bajas aún más hacia abajo.

Véase también el fragmento de código “Ajustar las posiciones de las ligaduras verticalmente”.

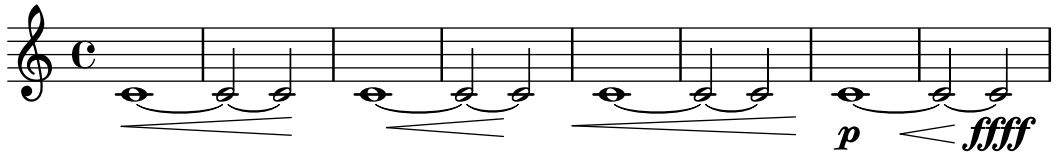
```
\relative c' {
  \stemDown
  e4( a)
  \override Slur.positions = #'(1 . 1)
  e4( a)
  \override Slur.positions = #'(2 . 2)
  e4( a)
  \override Slur.positions = #'(3 . 3)
  e4( a)
  \override Slur.positions = #'(4 . 4)
  e4( a)
  \override Slur.positions = #'(5 . 5)
  e4( a)
  \override Slur.positions = #'(0 . 5)
  e4( a)
  \override Slur.positions = #'(5 . 0)
  e4( a)
  \stemUp
  \override Slur.positions = #'(-5 . -5)
  e4( a)
  \stemDown
  \revert Slur.positions
  e4( a)
}
```



Desplazar los extremos de un regulador

Se pueden desplazar los extremos de los reguladores fijando la propiedad `shorten-pair` del objeto `Hairpin`. Los valores positivos mueven los extremos hacia la derecha, los negativos hacia la izquierda. A diferencia de la propiedad `minimum-length`, esta propiedad afecta solo a la apariencia del regulador; no ajusta el espaciado horizontal (incluida la posición de los matices circundantes). Este método es así adecuado para el ajuste fino de un regulador dentro del espacio de que dispone.

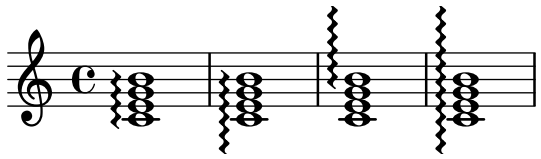
```
{
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(2 . 2)
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(-2 . -2)
  c'1~\<
  c'2~ c'\!
  c'1~\p-\tweak shorten-pair #'(2 . 0)\<
  c'2~ c'\ffff
}
```



Posicionar símbolos de arpegio

Si necesitamos alargar o acortar un símbolo de arpegio, podemos modificar independientemente los extremos superior e inferior.
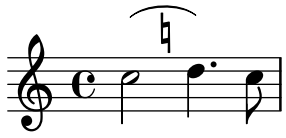
```
\relative c' {
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 0)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(0 . 5)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 5)
  <c e g b>1\arpeggio
}
```



Situar los elementos de marcado de texto por dentro de las ligaduras

Los elementos de marcado de texto deben tener la propiedad `outside-staff-priority` establecida a `#f` para que se impriman por dentro de las ligaduras de expresión.

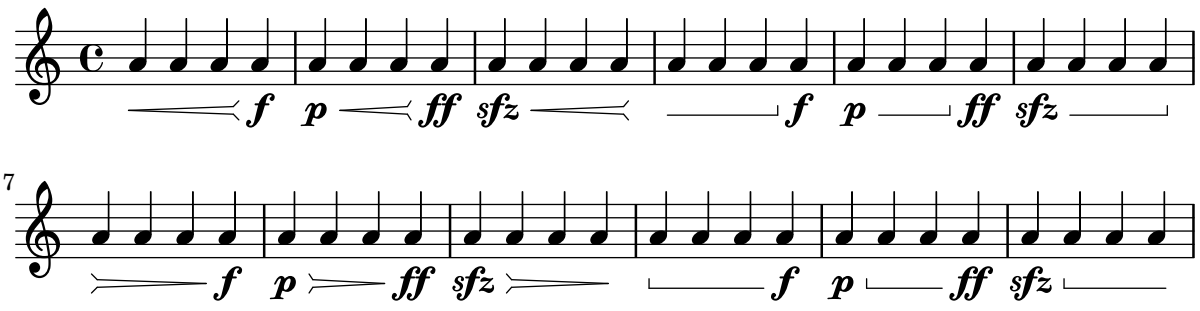
```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```



Imprimir reguladores en diversos estilos

Los reguladores se pueden crear en una amplia variedad de estilos.
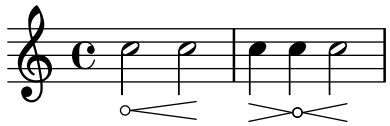
```
\relative c'' {
  \override Hairpin.stencil = #flared-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #flared-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
}
```



Impresión de reguladores utilizando la notación «al niente»

Se pueden imprimir reguladores con un círculo en la punta (notación «al niente») estableciendo la propiedad `circled-tip` del objeto `Hairpin` al valor `#t`.

```
\relative c'' {
  \override Hairpin.circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```



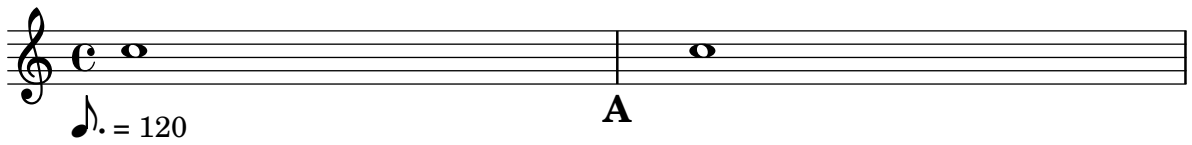
Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama

De forma predeterminada, las indicaciones metronómicas y las letras de ensayo se imprimen encima del pentagrama. Para colocarlas debajo del pentagrama, simplemente ajustamos adecuadamente la propiedad `direction` de `MetronomeMark` o de `RehearsalMark`.

```
\layout {
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

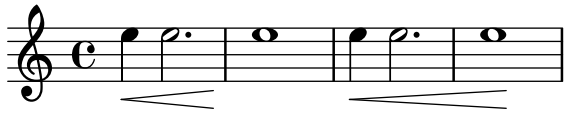
  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Establecer el comportamiento de los reguladores en las barras de compás

Si la nota que da fin a un regulador cae sobre la primera parte de un compás, el regulador se detiene en la línea divisoria inmediatamente precedente. Se puede controlar este comportamiento sobreescribiendo la propiedad `to-barline`.

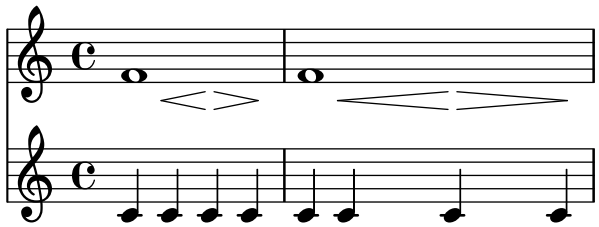
```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



Ajustar la longitud mínima de los reguladores

Si los reguladores son demasiado cortos, se pueden alargar modificando la propiedad `minimum-length` del objeto `Hairpin`.

```
<<
  {
    \after 4 \< \after 2 \> \after 2. \! f'1
    \override Hairpin.minimum-length = 8
    \after 4 \< \after 2 \> \after 2. \! f'1
  }
  {
    \*8 c'4
  }
>>
```



Imprimir la misma articulación encima y debajo de la misma nota o acorde

De forma predeterminada, LilyPond no permite poner la misma articulación (un acento, un calderón, un círculo de armónico, etc.) encima y debajo de la nota. Por ejemplo, `c4_\fermata^\fermata` imprime solamente el calderón inferior. El calderón superior sencillamente se ignora.

Sin embargo, se pueden adosar inscripciones (de igual forma que las digitaciones) dentro de un acorde, lo que significa que es posible tener tantas articulaciones como se desee. Este enfoque tiene la ventaja de que ignora la plica y posiciona la articulación de forma relativa a la cabeza de la nota. Puede verse esto en el caso de los flageolets (indicaciones de armónico) que aparecen en el fragmento de código. Para recrear el comportamiento de las inscripciones fuera del acorde, se requiere `add-stem-support`.

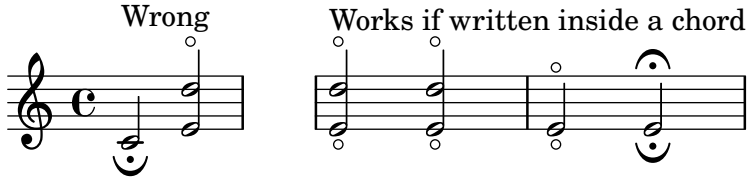
La solución consiste en escribir la nota como un acorde y añadir las articulaciones dentro de los paréntesis en ángulo `<...>`, usando los modificadores de dirección `^` y `_` según corresponda.

```
\relative c' {
  <>^"Wrong"
  c2_\fermata^\fermata % The second fermata is ignored!
  <e d'>2^\flageolet_\flageolet

  \stopStaff s1 \startStaff

  <>^"Works if written inside a chord"
  <e_\flageolet d'^\flageolet>2
  <e_\flageolet d'^\flageolet>2
```

```
  <e_\flageolet^\flageolet>2
  <e_\fermata^\fermata>2
}
```
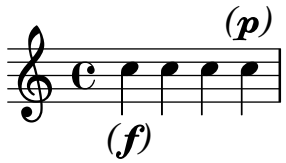


Slanted parentheses around dynamics

The easiest way to parenthesize dynamics is to use the normal `\parenthesize` command and adjust the font size, e.g., `{ c'4\tweak Parentheses.font-size 2 \parenthesize \f }`. However, this results in upright parentheses, while dynamic marks are normally slanted. This snippet gives a method to make the parentheses slanted as well.

```
paren =
#(define-event-function (dyn) (ly:event?)
   (make-dynamic-script
    #{ \markup \concat {
         \normal-text \italic \fontsize #2 (
         \pad-x #0.2 #(ly:music-property dyn 'text)
         \normal-text \italic \fontsize #2 )
       }
    #}))

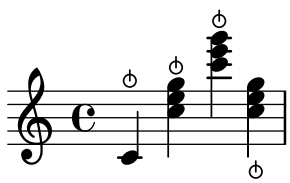
\relative c'' {
  c4\paren\f c c \dynamicUp c\paren\p
}
```



Snap pizzicato (“Bartok” pizzicato)

A snap pizzicato (also known as “Bartok pizzicato”) is a “strong pizzicato where the string is plucked vertically by snapping and rebounds off the fingerboard of the instrument” (Wikipedia). It is denoted by a circle with a vertical line going from the center upwards outside the circle.

```
\relative c' {
  c4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}
```

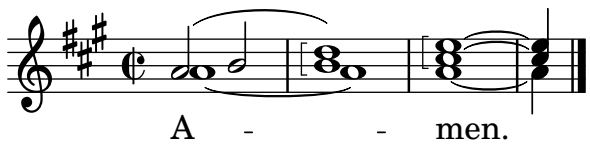


Using \arpeggioBracket to make divisi more visible

The \arpeggioBracket command can be used to indicate the division of voices where there are no stems to provide the information. This is often seen in choral music.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper"
      <<
        { \voiceOne \arpeggioBracket
          a2( b2
          <b d>1\arpeggio)
          <cs e>\arpeggio ~
          <cs e>4
        }
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower"
      { \voiceTwo
        a1 ~
        a
        a ~
        a4 \bar "|."
      }
    >>
  }
}
```



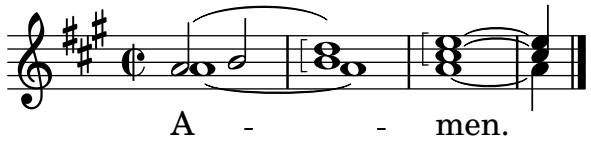
Using a bracket to clarify divisi

The \nonArpeggiato command can be used to indicate the division of voices where there are no stems to provide the information. This is often seen in choral music.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper" <<
        {
          \voiceOne
          a2( b2
          <b d>1\nonArpeggiato)
```
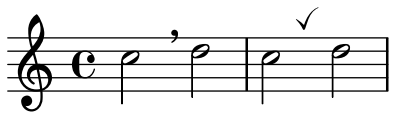
```
          <cs e>\nonArpeggiato ~
          <cs e>4
          \fine
        }
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower" {
        \voiceTwo
        a1 ~
        a
        a ~
        a4
        \fine
      }
    >>
  }
}
```



Usar una raya corta como símbolo de respiración

La música vocal y de viento usa con frecuencia una raya corta como signo de respiración. Esto indica una repiración que quita un poco de tiempo a la nota anterior en lugar de producir una corta pausa, lo que se indica con la marca de respiración en forma de coma. La marca se puede mover ligeramente hacia arriba para alejarla del pentagrama.

```
\relative c'' {
  c2
  \breathe
  d2
  \override BreathingSign.Y-offset = #2.6
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.tickmark" }
  c2
  \breathe
  d2
}
```



Utilizar ligaduras dobles para acordes legato

Algunos compositores escriben dos *ligaduras* cuando quieren acordes legato. Esto se puede conseguir estableciendo la propiedad de contexto `doubleSlurs`.

```
\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}
```
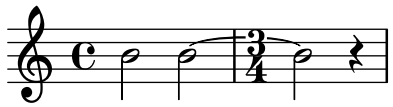


Uso de la propiedad `whiteout`

Se puede imprimir cualquier objeto gráfico sobre un fondo blanco para enmascarar parte de los objetos que están por debajo. Esto puede ser útil para mejorar el aspecto de las colisiones en situaciones complejas cuando no es práctico reposicionar los objetos. Es necesario establecer explícitamente la propiedad de capa (`layer`) para controlar qué objetos resultan enmascarados por el fondo blanco.

En este ejemplo, la colisión de la ligadura de unión con la indicación de compás resulta mejorada enmascarando la parte de la ligadura que cruza a la indicación de compás estableciendo la propiedad `whiteout` de `TimeSignature`. Para hacer esto, se mueve `TimeSignature` a una capa por encima de `Tie`, que se deja en la capa predeterminada 1, y `StaffSymbol` se mueve a una capa por encima de `TimeSignature` de manera que no resulte enmascarada.

```
{
  \override Score.StaffSymbol.layer = 4
  \override Staff.TimeSignature.layer = 3
  b'2 b'~
  \once \override Staff.TimeSignature.whiteout = ##t
  \time 3/4
  b' r4
}
```
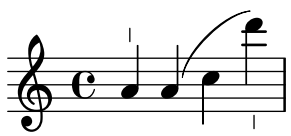


Línea vertical como una articulación barroca

Esta línea corta vertical situada encima de la nota es de uso común en música barroca. Su significado varía, pero en general indica notas que se deben tocar con más “peso”. El ejemplo siguiente muestra cómo conseguir dicha notación.

```
upline =
\tweak stencil
  #(lambda (grob)
    (grob-interpret-markup grob #{ \markup \draw-line #'(0 . 1) #}))
  \stopped
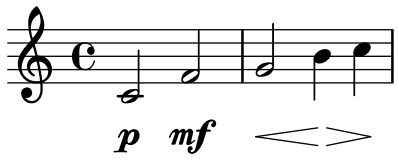
\relative c' {
  a'4^\upline a( c d')_\upline
}
```



Alinear verticalmente expresiones dinámicas que abarcan varias notas

Las expresiones dinámicas que se comienzan, terminan o se producen en la misma nota se alinean verticalmente. Para asegurar que las expresiones dinámicas se alinean cuando no se producen sobre la misma nota, incremente la propiedad `staff-padding` del objeto `DynamicLineSpanner`.

```
\relative c' {
  \override DynamicLineSpanner.staff-padding = #4
  c2\p f\mf
  g2\< b4\> c\!
}
```



4 Repeats

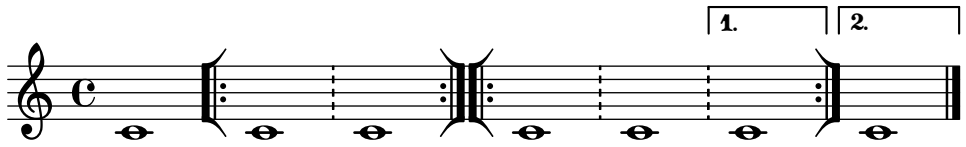
See also Sección “Repeats” en *Notation Reference*.

Cambiar las líneas divisorias predeterminadas

Se pueden cambiar las líneas divisorias predeterminadas cuando se redefinen en un contexto `Score`.

```
\layout {
  \context {
    \Score
    % Changing the defaults from engraver-init.ly
    measureBarType = "!"
    startRepeatBarType = "[|:"
    endRepeatBarType = ":|]"
    doubleRepeatBarType = ":|][|:"
  }
}

{
  c'1
  \repeat volta 2 { c' c' }
  \repeat volta 2 { c' c' \alternative { \volta 1 { c' }
                                        \volta 2 { c' } } }
  \bar "|."
}
```



Controlar el aspecto de las barras de trémolo

Usando diversas propiedades del objeto gráfico `StemTremolo` es posible controlar la apariencia de las barras de trémolo.

- La propiedad `slope` fija la inclinación de las barras de trémolo.
- La propiedad `shape` determina si las barras de trémolo tienen forma de rectángulo (valor `rectangle`) o de barras de corchea pequeñas (valor `beam-like`).
- La propiedad `style` fija la inclinación y la forma dependiendo de si la nota tiene corchete, barra o solo plica. Esto existe en contraste con las dos propiedades anteriores, que cambian la inclinación y la forma incondicionalmente. Hay dos estilos definidos.
  - `default`: las barras para los corchetes de plica hacia abajo son más largos y más inclinados que las barras para los corchetes de plica hacia arriba; las barras de trémolo sobre notas barradas tienen una forma rectangular y son paralelas a la barra de corchea.
  - `constant`: todas las barras de trémolo tienen forma de barra de corchea y tienen la misma inclinación excepto para los corchetes de plica hacia abajo.

```
music = {
  a''4:32 a':
  e''8: \noBeam e':
  a'':[ a':]
  f':[ g':]
  d':[ d':]
}

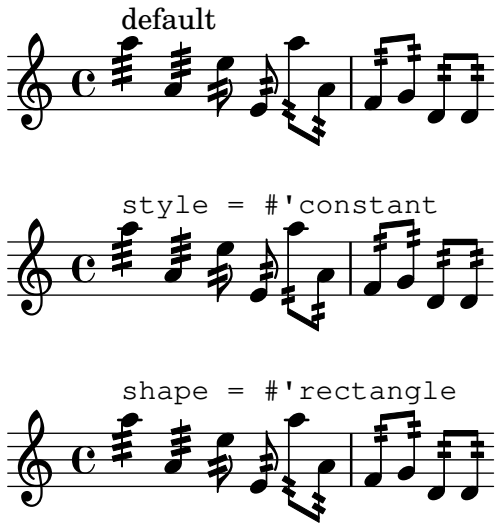
\new Staff {
  <>^\markup "default"
  \music
}

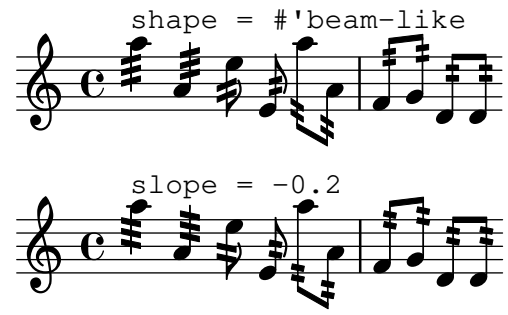
\new Staff {
  <>^\markup \typewriter "style = #'constant"
  \override StemTremolo.style = #'constant
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'rectangle"
  \override StemTremolo.shape = #'rectangle
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'beam-like"
  \override StemTremolo.shape = #'beam-like
  \music
}

\new Staff {
  <>^\markup \typewriter "slope = -0.2"
  \override StemTremolo.slope = -0.2
  \music
}
```

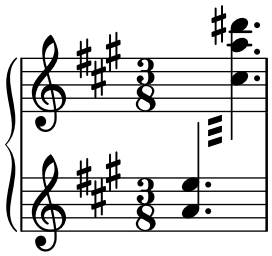




Trémolos de pentagrama cruzado

Dado que `\repeat tremolo` espera exactamente dos argumentos musicales para los trémolos de acorde, la nota o acorde que cambia de pentagrama en un trémolo que cruza el pentagrama se debe colocar dentro de llaves curvas junto a su instrucción `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```
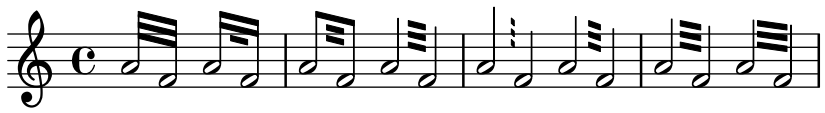


Trémolos con barras flotantes

Si la duración total de un trémolo es menor de una negra, o exactamente de una blanca, normalmente se compone tipográficamente con todas las barras de corchea tocando las plicas. Ciertos estilos de notación de partituras componen algunas de estas barras como barras flotantes centradas que no tocan las plicas de las figuras. El número de barras flotantes en este tipo de trémolo está controlado por la propiedad `gap-count` del objeto `Beam` y la separación entre las barras y las plicas se fija con la propiedad `gap`.

```
\relative c'' {
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #2
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #3
  \repeat tremolo 8 { a32 f }

  \override Beam.gap-count = #3
  \override Beam.gap = #1.33
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.67
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.33
  \repeat tremolo 8 { a32 f }
}
```
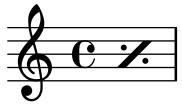


Símbolos de porcentaje sueltos

También se pueden imprimir símbolos de porcentaje sueltos.

```
makePercent =
#(define-music-function (note) (ly:music?)
   "Make a percent repeat the same length as NOTE."
   (make-music 'PercentEvent
               'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```



Measure counters

This snippet demonstrates the use of the `Measure_counter_engraver` to number groups of successive measures. Any stretch of measures may be numbered, whether consisting of repetitions or not.

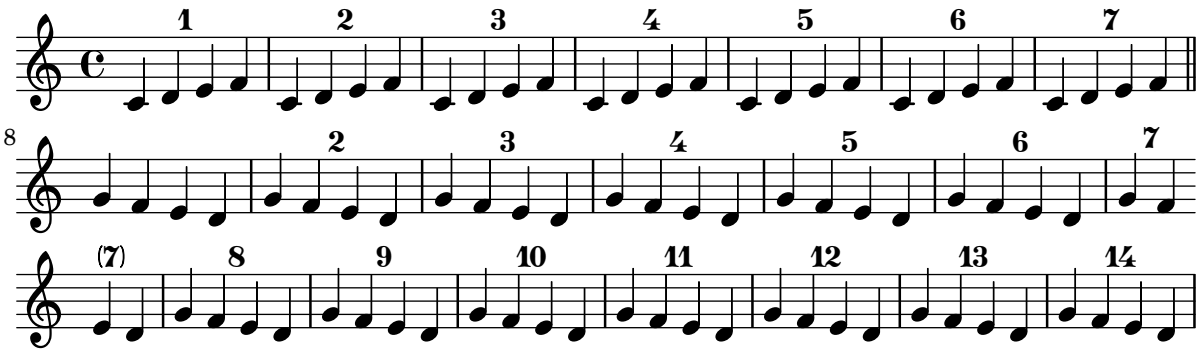
The engraver must be added to the appropriate context. Here, a `Staff` context is used; another possibility is a `Dynamics` context.

The counter is begun with `\startMeasureCount` and ended with `\stopMeasureCount`. Numbering will start by default with 1, but this behavior may be modified by overriding the `count-from` property.

When a measure extends across a line break, the number will appear twice, the second time in parentheses.

```
\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

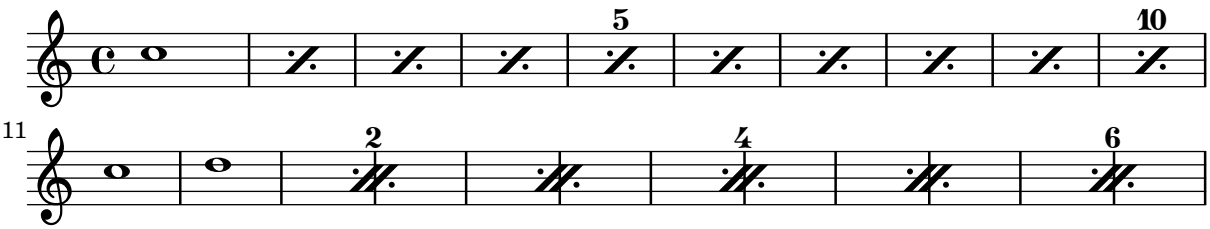
\new Staff {
  \startMeasureCount
  \*7 { c'4 d' e' f' }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \*5 { g'4 f' e' d' }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \*7 { g'4 f' e' d' }
  \stopMeasureCount
}
```



Visibilidad del contador de repeticiones de tipo porcentaje

Se pueden mostrar los contadores de las repeticiones del tipo porcentaje a intervalos regulares mediante el establecimiento de la propiedad de contexto `repeatCountVisibility`.

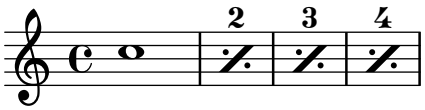
```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \%10 c1 \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \%6 { c1 d1 }
}
```



Contador de repeticiones de tipo porcentaje

Las repeticiones de compases completos de más de dos repeticiones pueden llevar impreso un contador si se ajusta el valor de la propiedad de contexto `countPercentRepeats`.

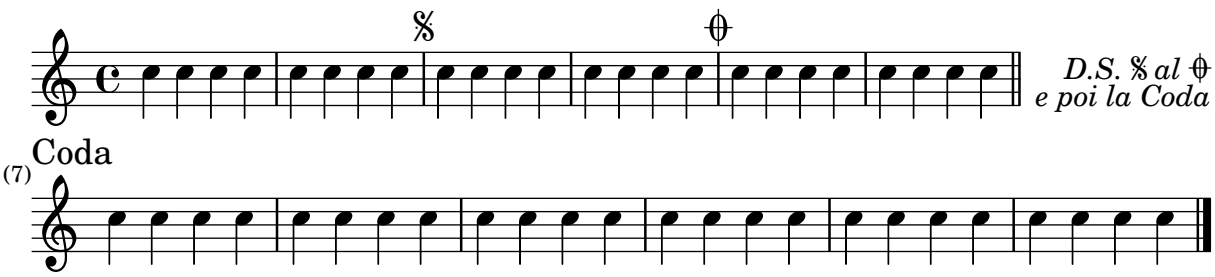
```
\relative c'' {
  \set countPercentRepeats = ##t
  \%4 c1
}
```



Posicionar los símbolos de repetición segno y coda (con saltos de línea)

Si queremos colocar un símbolo de segno de salida y añadir texto como “D.S. al Coda” junto a él donde normalmente están las líneas del pentagrama, cpodemos usar este código. La coda continúa en una línea nueva. Hay una variante documentada en el fragmento de código, donde la coda se mantiene en la misma línea.
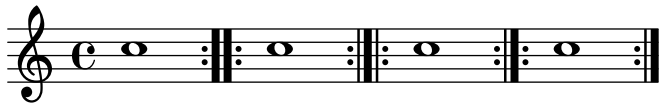
```
\relative c'' {
  c4 c c c | c c c c |
  \repeat segno 2 {
    c4 c c c | c c c c |
    \alternative {
      \volta 1 {
        c4 c c c | c c c c |
        % If you don't use \break at Coda, use \noBreak here
        % and after \bar "" below.
        \noBreak
        \section % double bar line
        \cadenzaOn % pause bar count
        \stopStaff % remove staff lines
        % Increasing the unfold counter will expand the staff-free space
        \repeat unfold 4 {
          s1
          \bar ""
        }
        % Place JumpScript where the staff would normally be.
        \once \override Score.JumpScript.outside-staff-priority = ##f
        \once \override Score.JumpScript.Y-offset = 0
        \startStaff % resume bar count
        \cadenzaOff % show staff lines again
      }
    }
  }
  \sectionLabel "Coda"
  % Show Coda on a new line
  \break
  \*6 { c4 c c c }
  \fine
}
```



Establecer el tipo de repetición doble predeterminado para la primera y segunda vez

Existen distintos estilos de repeticiones dobles para la primera y segunda vez, que se pueden seleccionar utilizando la propiedad de contexto `doubleRepeatBarType`.
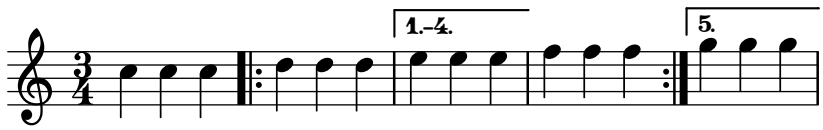
```
\relative c'' {
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":..:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":|.|:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":|.:"
  \repeat volta 2 { c1 }
}
```



Acortar los corchetes de primera y segunda vez

De forma predeterminada, los corchetes de primera y segunda vez se trazan encima de los finales alternativos completos, pero es posible acortartlos sobreescribiendo el valor de `VoltaBracket.musical-length`. En el ejemplo siguiente, el corchete solo abarca un compás, que corresponde a una duración de 3/4.

```
\fixed c'' {
  \time 3/4
  c4 c c
  \repeat volta 5 {
    d4 d d
    \alternative {
      \volta 1,2,3,4 {
        \once \override Score.VoltaBracket.musical-length =
          \musicLength 2.
        e4 e e
        f4 f f
      }
      \volta 5 {
        g4 g g } } }
}
```



Unfolding tremolo repeats
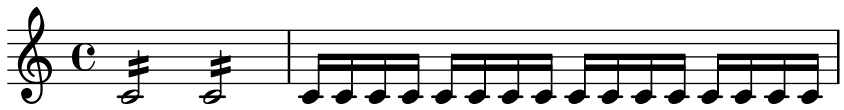
Currently, *note*:*duration*, which is more or less a shortcut for \repeat tremolo, is not unfolded by \unfoldRepeats (this is tracked in Issue #6145 (https://gitlab.com/lilypond/lilypond/-/issues/6145)). The function given in this snippet provides a workaround.

```
fixTremolos =
#(define-music-function (music) (ly:music?)
   (music-map
    (lambda (m)
      (let ((event (any (lambda (a)
                          (and (music-is-of-type? a 'tremolo-event)
                               a))
                        (ly:music-property m 'articulations))))
        (if event
            (let* ((total-tremolo-duration (ly:music-property m
                                                              'duration))
                   (tremolo-type (ly:music-property event
                                                    'tremolo-type))
                   (one-tremolo-note-duration
                    (ly:make-duration (ly:intlog2 tremolo-type)))
                   (tremolo-note-count
                    (/ tremolo-type (expt 2 (ly:duration-log
                                             total-tremolo-duration)))))
              (set! (ly:music-property m 'duration)
                    one-tremolo-note-duration)
              (set! (ly:music-property m 'articulations)
                    (delete! event (ly:music-property m 'articulations)))
              (make-music 'TremoloRepeatedMusic
                          'repeat-count tremolo-note-count
                          'element m))
            m)))
    music))

unfoldRepeats = \unfoldRepeats #'() \fixTremolos \etc

music = { \repeat tremolo 8 c'16 c'2:16 }

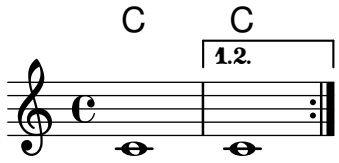
{
  \music
  \unfoldRepeats \music
}
```



Corchetes de primera y segunda vez debajo de los acordes

Mediante la adición del grabador Volta_engraver al pentagrama pertinente, se pueden poner los corchetes de primera y segunda vez debajo de los acordes.

```
\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



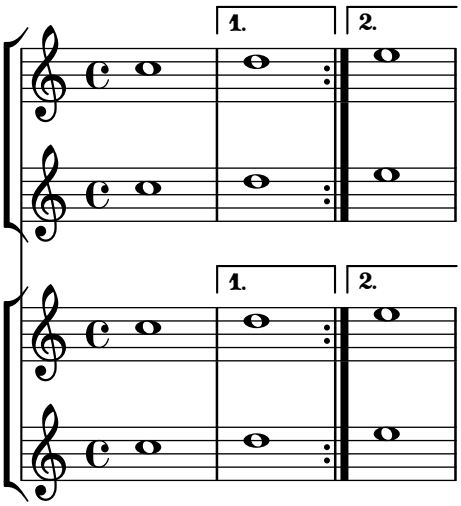
Casillas de primera y segunda vez en varios pentagramas

Añadiendo el grabador `Volta_engraver` al pentagrama deseado, las casillas de primera y segunda vez se pueden poner sobre pautas distintas a la del extremo superior de la partitura.

Las instrucciones `\repeat` y otras relacionadas deben estar presentes en todos los pentagramas.

```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}

<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```
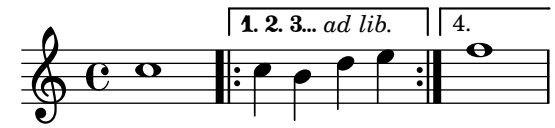


Elementos de marcado para el texto de las casillas de repetición usando `repeatCommands`

Aunque las casillas de repetición se especifican de forma óptima usando `\repeat volta`, debe usarse la propiedad de contexto `repeatCommands` en caso de que el texto de la casilla requiera un formato más avanzado con `\markup`.

Puesto que `repeatCommands` admite una lista, el método más sencillo de incluir elementos de marcado es usar un identificador para el texto e insertarlo en la lista de instrucciones usando la sintaxis de Scheme `#`((volta ,`*textIdentifier*`) ...)` (observe el uso del apóstrofo invertido después de # y la coma antes de *textIdentifier*). Las instrucciones de principio y fin de repetición se pueden añadir como elementos de lista independientes:

```
voltaAdLib = \markup { \volta-number { 1. 2. 3... } \italic { ad lib. } }

\relative c'' {
  c1
  \set Score.repeatCommands = #`((volta ,voltaAdLib) start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```



5 Simultaneous notes
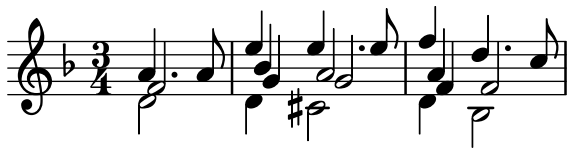
See also Sección “Simultaneous notes” en *Notation Reference*.

Voces adicionales para evitar colisiones

En ciertos casos de polifonía compleja, se necesitan voces adicionales para evitar colisiones entre las notas. Si se necesitan más de cuatro voces paralelas, las voces adicionales se añaden definiendo una variable que utiliza la función de Scheme `context-spec-music`.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)
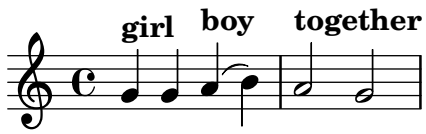
\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    \new Voice  {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \new Voice {
      \voiceTwo
      d,2
      d4 cis2
      d4 bes2
    }
    \new Voice {
      \voiceThree
      f'2
      bes4 a2
      a4 s2
    }
    \new Voice {
      \voiceFive
      s2
      g4 g2
      f4 f2
    }
  >>
}
```



Cambiar los textos de `\partCombine`

Al utilizar la posibilidad de combinación automática de partes, se puede modificar el texto que se imprime para las secciones de solo y de unísono:

```
\new Staff <<
  \set Staff.soloText = "girl"
  \set Staff.soloIIText = "boy"
  \set Staff.aDueText = "together"
  \partCombine
    \relative c'' {
      g4 g r r
      a2 g
    }
    \relative c'' {
      r4 r a( b)
      a2 g
    }
>>
```

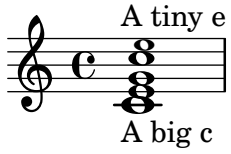


Changing a single note's size in a chord

Individual note heads in a chord can be modified with the `\tweak` command inside a chord, by altering the `font-size` property.

Inside the chord (within the brackets < >), before the note to be altered, place the `\tweak` command, followed by `font-size` and define the proper size like `#-2` (a tiny note head).

```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```

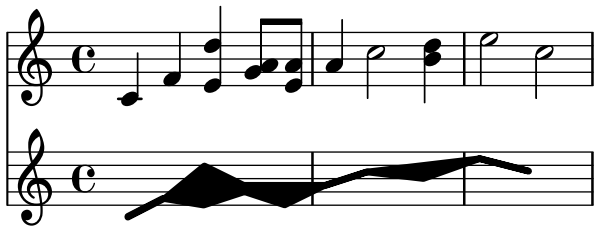


Clusters («racimos»)

Los «clusters» o racimos son un mecanismo para indicar la interpretación de un ámbito de notas al mismo tiempo.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Combinar dos partes sobre el mismo pentagrama

La herramienta de combinación de partes (o sea, la instrucción \partCombine) permite la combinación de varias partes diferentes sobre el mismo pentagrama. Las indicaciones textuales tales como "solo" o "a2" se añaden de forma predeterminada; para quitarlas, sencillamente establezca la propiedad printPartCombineTexts al valor #f. Para partituras vocales (como himnos), no hay necesidad de añadir los textos "solo" o "a2", por lo que se deben desactivar. Sin embargo, podría ser mejor no usarlos si hay solos, porque éstos no se indicarán. En tales casos podría ser preferible la notación polifónica estándar.

Este fragmento de código presenta las tres formas en que se pueden imprimir dos partes sobre un solo pentagrama: polifonía estándar, \partCombine sin textos, y \partCombine con textos.

```
musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}
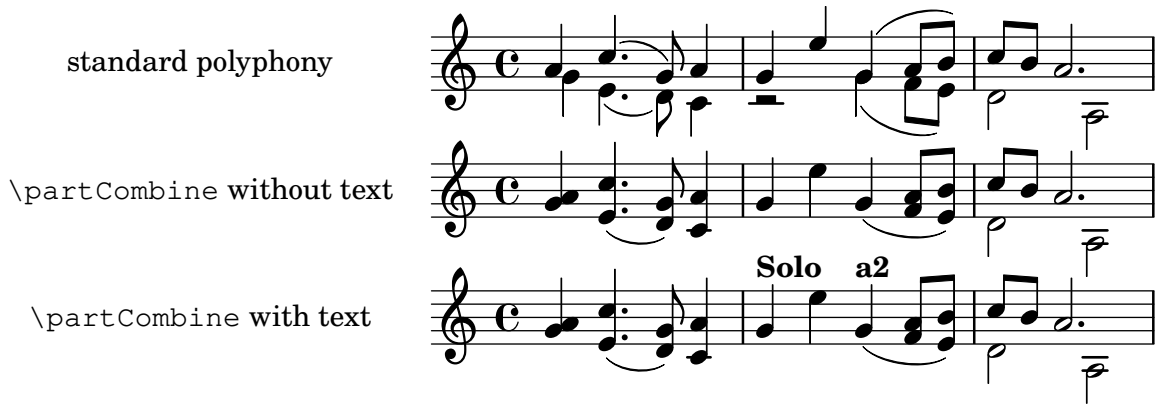
\score {
  <<
    \new Staff \with {
      instrumentName = "standard polyphony"
    } << \musicUp \\ \musicDown >>

    \new Staff \with {
      instrumentName =
        \markup { \typewriter "\\partCombine" without text}
      printPartCombineTexts = ##f
    } \partCombine \musicUp \musicDown

    \new Staff \with {
      instrumentName =
        \markup { \typewriter "\\partCombine" with text}
    } \partCombine \musicUp \musicDown
  >>

  \layout {
    indent = 6.0\cm
```

```
    \context {
      \Score
      % Setting this to a large value avoids a bar line at the
      % beginning that would connect the three staves otherwise.
      \override SystemStartBar.collapse-height = 30
    }
  }
}
```
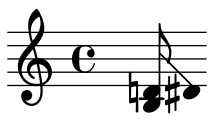


Impresión de acordes complejos

He aquí una forma de imprimir un acorde en el que suena la misma nota dos veces con distintas alteraciones.

```
fixA = {
  \once \override Stem.length = #12
}

fixB = {
  \once \override NoteHead.X-offset = #1.7
  \once \override Stem.length = #7
  \once \override Stem.rotation = #'(45 0 0)
  \once \override Stem.extra-offset = #'(-0.1 . -0.2)
  \once \override Flag.style = #'no-flag
  \once \override Accidental.extra-offset = #'(4 . -.1)
}

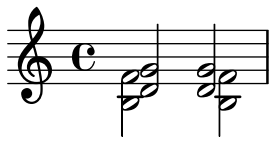

\relative c' {
  << { \fixA <b d!>8 } \\ { \voiceThree \fixB dis } >> s
}
```



Forzar el desplazamiento horizontal de las notas

Cuando el motor de tipografiado no es capaz de todo, se puede usar la sintaxis siguiente para sobreescribir las decisiones de tipografía. Las unidades de medida que se usan aquí son espacios de pentagrama.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
    <b f'>2
    \once \override NoteColumn.force-hshift = 1.7
    <b f'>2
  }
>>
```



Making an object invisible using \hide
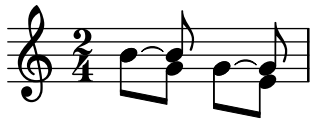
Applying \hide to a grob causes objects of this type to be printed with "invisible ink". They are not printed, but all of their other behavior is retained:

- the objects still take up space,
- they take part in collision resolution, and
- slurs, ties, and beams can be attached to them as usual.

This snippet demonstrates how to connect different voices using ties. Normally, ties only connect two notes in the same voice. By introducing a tie in a different voice, and blanking the first up-stem in that voice, the tie appears to cross voices.

```
\relative {
  \time 2/4
  <<
    {
      \once \hide Stem
      \once \override Stem.length = #8
      b'8 ~ 8\noBeam
      \once \hide Stem
      \once \override Stem.length = #8
      g8 ~ 8\noBeam
    }
    \\
    {
      b8 g g e
    }
  >>
}
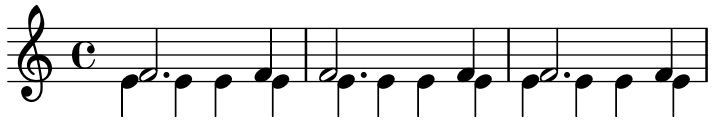
\paper {
  line-width = 40\mm
  ragged-right = ##f
}
```



Desplazar las notas con puntillo en polifonía

Cuando se puede una nota en la voz superior para evitar la colisión con una nota de otra voz, el comportamiento predeterminado es desplazar la nota superior a la derecha. Se puede cambiar modificando la propiedad `prefer-dotted-right` del objeto gráfico `NoteCollision`.

```
\new Staff \relative c' <<
  {
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##f
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##t
    f2. f4
  }
  \\
  { e4 e e e e e e e e e e }
>>
```
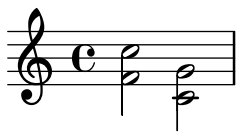


Evitar las advertencias sobre columnas de notas que chocan

Si se colocan sobre la misma posición notas de dos voces distintas con las plicas en la misma dirección, y ninguna de las voces tiene un desplazamiento o ambas tienen el mismo desplazamiento, aparece el mensaje de error "advertencia: demasiadas columnas de notas en colisión; se ignora" aparece al compilar el archivo de LilyPond. Este mensaje se puede evitar fijando la propiedad `ignore-collision` del objeto `NoteColumn` al valor `#t`. Observe que esto no elimina solamente las advertencias, sino que hace que LilyPond deje de intentar resolver las colisiones en absoluto, por lo que pueden obtenerse resultados distintos de los esperados si no se usa con prudencia.

```
ignore = \override NoteColumn.ignore-collision = ##t

\relative c' {
  \new Staff <<
    \new Voice { \ignore \stemDown f2 g }
    \new Voice { c2 \stemDown c, }
  >>
}
```



Dos pares de \partCombine en un solo pentagrama

La función \partCombine toma dos expresiones musicales, cada una de las cuales contiene una parte o voz, y las distribuye entre cuatro contextos de Voice llamados “one”, “two”, “solo” y “shared” dependiendo de cuándo y cómo las partes o voces se combinan en una voz común.

Las voces que devuelve \partCombine pueden ver ajustadas sus propiedades de presentación de la manera usual. Aquí definimos extensiones de \partCombine para facilitar la inclusión de cuatro voces dentro del mismo pentagrama.

\partCombineUp y \partCombineDown son variantes de \partCombine pensadas para producir combinaciones de dos voces con la plica hacia arriba y hacia abajo, respectivamente. Sin embargo, combinarlas para comprimir cuatro partes en un solo pentagrama requiere ciertos ajustes especiales, los que este fragmento de código define adecuadamente.

```
customPartCombineUp =
#(define-music-function (part1 part2) (ly:music? ly:music?)
  "Make an up-stem `VoiceBox` context that combines PART1 and PART2.

The context is called 'Up'; internally, the function calls
`\\partCombineUp`."
  #{
    \new VoiceBox = "Up" <<
      \context Voice = "one" { \voiceOne }
      \context Voice = "two" { \voiceThree }
      \context Voice = "shared" { \voiceOne }
      \context Voice = "solo" { \voiceOne }
      \context NullVoice = "null" {}
      \partCombine #part1 #part2
    >>
  #})

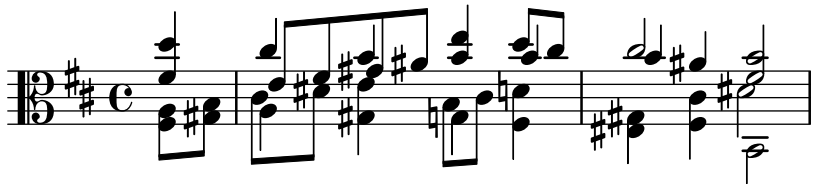
customPartCombineDown =
#(define-music-function (part3 part4) (ly:music? ly:music?)
  "Make a down-stem `VoiceBox` context that combines PART3 and PART4.

The context is called 'Down'; internally, the function calls
`\\partCombineDown`."
  #{
    \new VoiceBox = "Down" <<
      \set VoiceBox.soloText = #"Solo III"
      \set VoiceBox.soloIIText = #"Solo IV"
      \context Voice ="one" { \voiceFour }
      \context Voice ="two" { \voiceTwo }
      \context Voice ="shared" { \voiceFour }
      \context Voice ="solo" { \voiceFour }
      \context NullVoice = "null" {}
      \partCombine #part3 #part4
    >>
  #})

soprano = { d'4 | cis'  b  e'  d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }
```

```
\new Staff <<
  \key b\minor
  \clef alto
  \partial 4
  \transpose b b' \customPartCombineUp \soprano \alto
  \customPartCombineDown \tenor \bass
>>

\layout {
  \context {
    \Staff
    \accepts "VoiceBox"
  }
  \context {
    \name "VoiceBox"
    \type "Engraver_group"
    \defaultchild "Voice"
    \accepts "Voice"
    \accepts "NullVoice"
  }
}
```
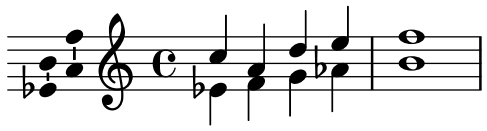


6 Staff notation

See also Sección “Staff notation” en *Notation Reference*.

Añadir un ámbito por voz

Se puede añadir un ámbito por cada voz. En este caso, el ámbito se debe desplazar manualmente para evitar colisiones.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```

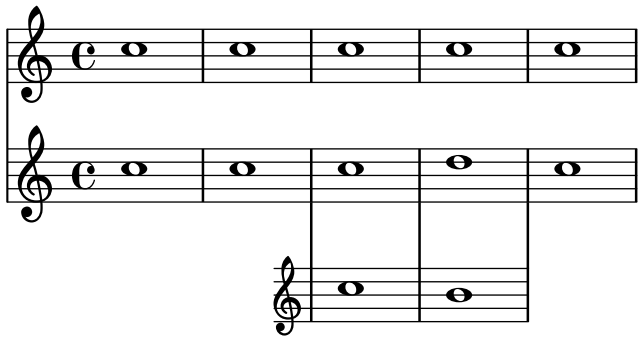


Añadir un pentagrama nuevo

Se puede añadir (posiblemente de forma temporal) un pentagrama nuevo una vez que la pieza ha comenzado.

```
\score {
  <<
    \new Staff \relative c'' {
      c1 | c | c | c | c
    }
    \new StaffGroup \relative c'' {
      \new Staff {
        c1 | c
        <<
          { c1 | d }
          \new Staff {
            \once \omit Staff.TimeSignature
            c1 | b
          }
        >>
        c1
      }
    }
```

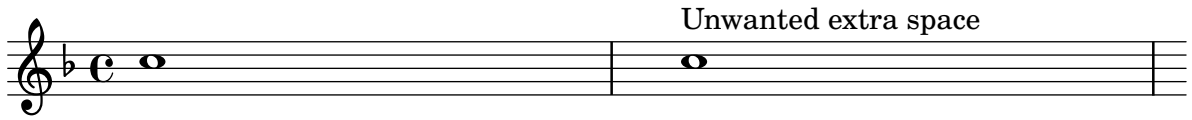
```
  >>
}
```

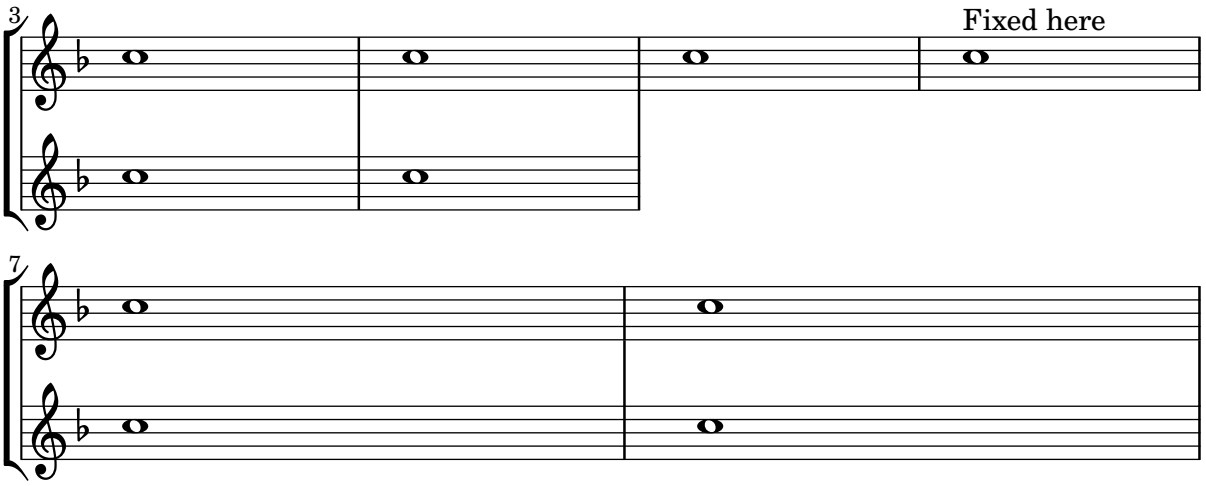


Añadir un pentagrama adicional en un salto de línea

Al añadir un pentagrama nuevo en un salto de línea, por desgracia se añade un espacio adicional al final de la línea antes del salto (reservado para hacer sitio a un cambio de armadura que de todas formas nunca se imprime). La solución alternativa es establecer la propiedad `Staff.explicitKeySignatureVisibility` del objeto gráfico `Staff` como se muestra en el ejemplo.

```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
       \new Staff {
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
    c1 | c^"Fixed here" \break
    << { c1 | c }
       \new Staff {
         \once \set Staff.explicitKeySignatureVisibility =
           #end-of-line-invisible
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
  }
}
```





Añadir indicadores a los pentagramas que se dividen después de un salto de línea

Este fragmento de código define las instrucciones \splitStaffBarLine, \convUpStaffBarLine y \convDownStaffBarLine, que añaden flechas en las sobre una línea divisoria para denotar que cada una de las voces que comparten un pentagrama continúan en su propio pentagrama en el siguiente sistema, o que las voces así divididas se recombinan.

Observe que la implementación de este fragmento de código traza unas flechas sin dimensiones en el margen derecho. Para la impresión formal, esto no produce ningún problema. Sin embargo, es necesario aumentar las dimensiones de la caja circundante en el sentido horizontal si se procesa el código como imagen para evitar el recorte, como se muestra abajo.

```
#(define-markup-command (arrow-at-angle layout props angle-deg length fill)
   (number? number? boolean?)
   (let* ((PI-OVER-180 (/ (atan 1 1) 34))
          (degrees->radians (lambda (degrees) (* degrees PI-OVER-180)))
          (angle-rad (degrees->radians angle-deg))
          (target-x (* length (cos angle-rad)))
          (target-y (* length (sin angle-rad))))
     (interpret-markup layout props
       (markup
        #:translate (cons (/ target-x 2) (/ target-y 2))
        #:rotate angle-deg
        #:translate (cons (/ length -2) 0)
        #:concat (#:draw-line (cons length 0)
                  #:arrow-head X RIGHT fill)))))

splitStaffBarLineMarkup = \markup \with-dimensions #'(0 . 0) #'(0 . 0) {
  \combine
  \arrow-at-angle #45 #(sqrt 8) ##t
  \arrow-at-angle #-45 #(sqrt 8) ##t
}

splitStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
```

```
      X RIGHT
      (grob-interpret-markup grob splitStaffBarLineMarkup)
      0))
  \break
}

convDownStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . -.13)\arrow-at-angle #-45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

convUpStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . .14)\arrow-at-angle #45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

\paper {
  indent = 10\mm
  short-indent = 10\mm
  line-width = 8\cm
}

separateSopranos = {
  \set Staff.instrumentName = "AI AII"
  \set Staff.shortInstrumentName = "AI AII"
  \splitStaffBarLine
  \change Staff = "up"
}
convSopranos = {
  \convDownStaffBarLine
  \change Staff = "shared"
  \set Staff.instrumentName = "S A"
  \set Staff.shortInstrumentName = "S A"
}
```

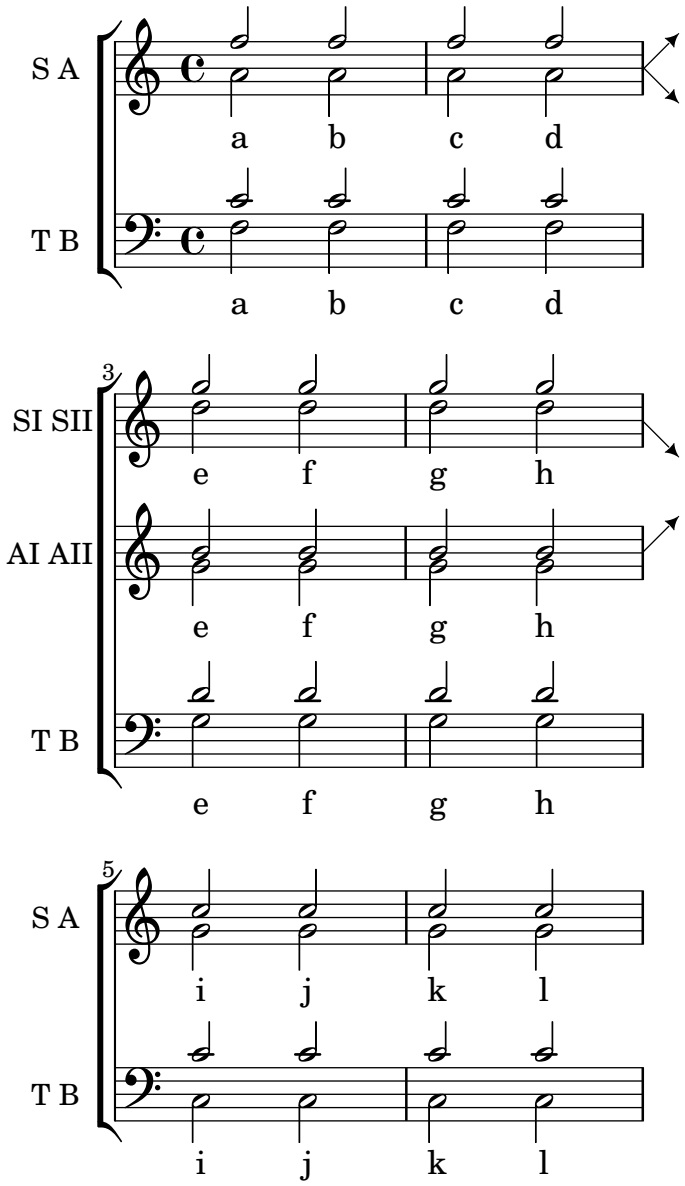
```
sI = {
  \voiceOne
  \*4 f''2
  \separateSopranos
  \*4 g''2
  \convSopranos
  \*4 c''2
}
sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \*4 d''2
}
aI = {
  \voiceTwo
  \*4 a'2
  \voiceOne
  \*4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \*4 g'2
}
aII = {
  s1*2
  \voiceTwo
  \*4 g'2
}
ten = {
  \voiceOne
  \*4 c'2
  \*4 d'2
  \*4 c'2
}
bas = {
  \voiceTwo
  \*4 f2
  \*4 g2
  \*4 c2
}

\markup \pad-x #3  % avoid cropping
  \score {
    <<
      \new ChoirStaff <<
        \new Staff = up \with {
          instrumentName = "SI SII"
          shortInstrumentName = "SI SII"
        } {
          s1*4
        }
```

```
        \new Staff = shared \with {
          instrumentName = "S A"
          shortInstrumentName = "S A"
        } <<
          \new Voice = sopI \sI
          \new Voice = sopII \sII
          \new Voice = altI \aI
          \new Voice = altII \aII
        >>
        \new Lyrics \with {
          alignBelowContext = up
        }
        \lyricsto sopII { e f g h }
        \new Lyrics \lyricsto altI { a b c d e f g h i j k l }

        \new Staff = men \with {
          instrumentName = "T B"
          shortInstrumentName = "T B"
        } <<
          \clef F
          \new Voice = ten \ten
          \new Voice = bas \bas
        >>
        \new Lyrics \lyricsto bas { a b c d e f g h i j k l }
      >>
    >>

    \layout {
      \context {
        \Staff \RemoveEmptyStaves
        \override VerticalAxisGroup.remove-first = ##t
      }
    }
}
```



Añadir notas guía orquestales a una partitura vocal

Este ejemplo muestra una forma de simplificar la adición de muchas notas guía orquestales a la reducción de piano en una partitura vocal. La función musical `\cueWhile` toma cuatro argumentos: la música de la que se toma la cita, como viene definida por `\addQuote`, el nombre que insertar antes de las notas guía, y después `UP` o `DOWN` para especificar `\voiceOne` con el nombre encima del pentagrama o bien `\voiceTwo` con el nombre debajo del pentagrama, y finalmente la música de piano con la que las notas guía deben aparecer en paralelo. El nombre del instrumento citado se posiciona a la izquierda de las notas guía. Se pueden citar muchos pasajes como guía, pero no se pueden superponer en el tiempo entre ellos.

```
cueWhile =
#(define-music-function
   (instrument name dir music)
   (string? string? ly:dir? ly:music?)
   #{
     \cueDuring $instrument #dir {
       \once \override TextScript.self-alignment-X = #RIGHT
```

```
      \once \override TextScript.direction = $dir
      <>-\markup { \tiny #name }
      $music
    }
  #})
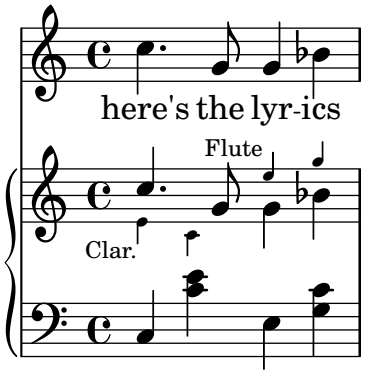
flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```



Numeración de compases alternativa

Fijando la propiedad de contexto `alternativeNumberingStyle`, se dispone de dos métodos alternativos adicionales para la numeración de compases dentro de las repeticiones.
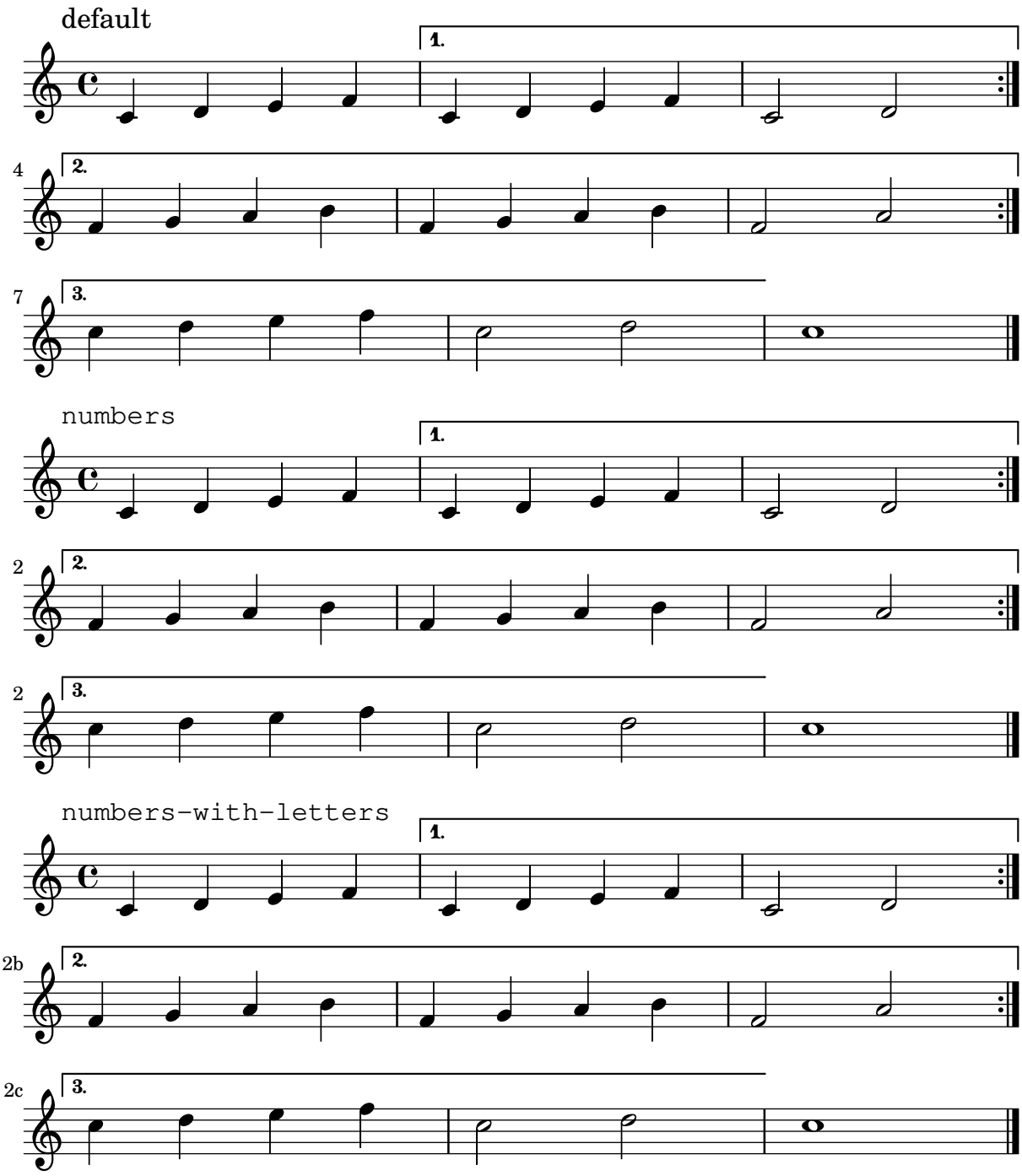
```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|."
}

{
  \textMark \markup \large "default"
  \music
}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
  \music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}
```
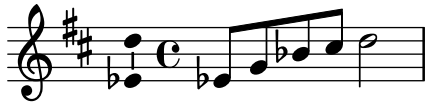


Indicación de tesitura después de la armadura

De manera predeterminada, las indicaciones de ámbito o tesitura se sitúan a la izquierda de la clave. La función `\ambitusAfter` permite cambiar esta colocación. La sintaxis es `\ambitusAfter` *`grob-interface`* (consulte Interfaces de los objetos gráficos (`https://lilypond.org/doc/v2.24/Documentation/internals/graphical-object-interfaces`) para ver una lista de los valores posibles para *grob-interface*.)

Un caso de utilización bastante común es imprimir la indicación de texitura entre la armadura de la clave y la indicación de compás.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```

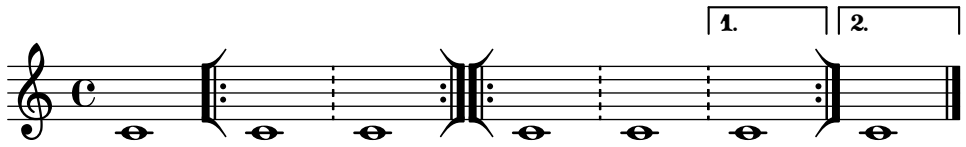


Cambiar las líneas divisorias predeterminadas

Se pueden cambiar las líneas divisorias predeterminadas cuando se redefinen en un contexto Score.

```
\layout {
  \context {
    \Score
    % Changing the defaults from engraver-init.ly
    measureBarType = "!"
    startRepeatBarType = "[|:"
    endRepeatBarType = ":|]"
    doubleRepeatBarType = ":|][|:"
  }
}

{
  c'1
  \repeat volta 2 { c' c' }
  \repeat volta 2 { c' c' \alternative { \volta 1 { c' }
                                        \volta 2 { c' } } }
  \bar "|."
}
```



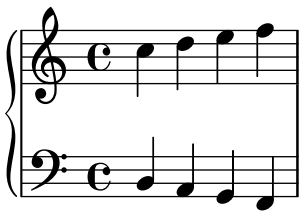
Cambiar el número de líneas de una pauta

El número de líneas de una pauta se puede modificar sobreescribiendo la propiedad `line-count` del objeto `StaffSymbol`.

```
upper = \relative c'' {
  c4 d e f
}

lower = \relative c {
  \clef bass
  c4 b a g
}
```

```
\score {
  \context PianoStaff <<
    \new Staff {
      \upper
    }
    \new Staff {
      \override Staff.StaffSymbol.line-count = #4
      \lower
    }
  >>
}
```



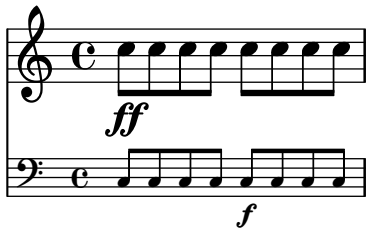
Modificar el tamaño de la pauta

La manera más sencilla de redimensionar los pentagramas es usar

```
#(set-global-staff-size tamaño)
```

El tamaño de una pauta individual se puede cambiar usando las propiedades `staff-space` y de `fontSize`.

```
<<
  \new Staff \relative c'' {
    \dynamicDown c8\ff c c c c c c c
  }
  \new Staff \with {
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
  } \relative c {
    \clef bass c8 c c c c\f c c c
  }
>>
```



Crear pentagramas en blanco

Para crear pentagramas en blanco, genere compases vacíos y después elimine el grabador de números de compás `Bar_number_engraver` del contexto `Score`, y los grabadores de la indicación de compás `Time_signature_engraver`, de la clave `Clef_engraver` y de los compases `Bar_engraver` del contexto de `Staff`.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

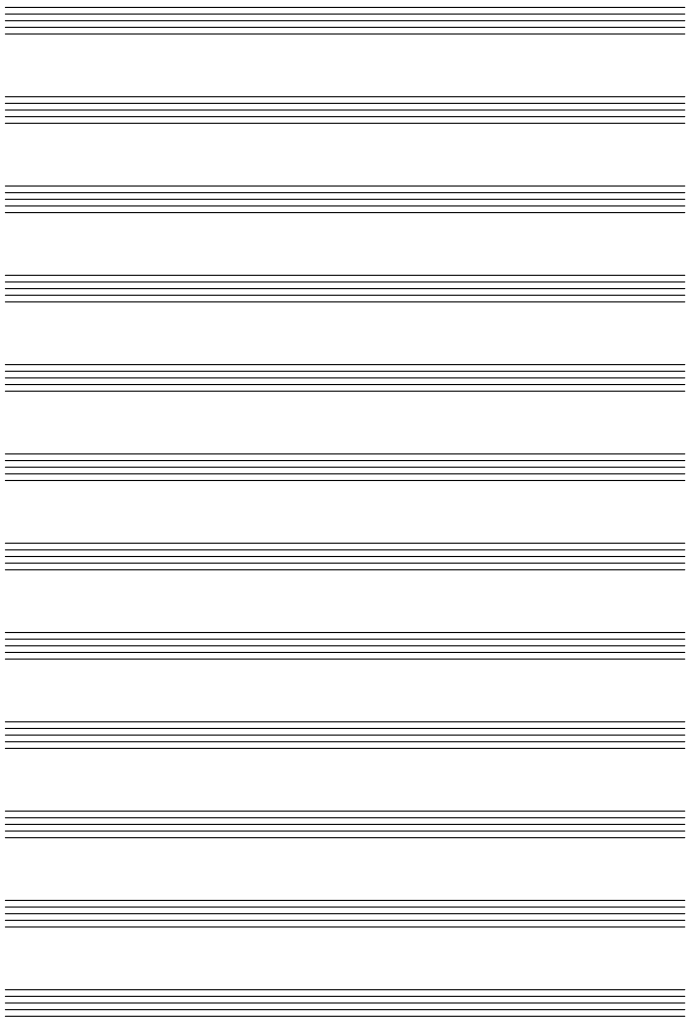
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % for the documentation
  \paper {
    #(set-paper-size "a6")
    ragged-last-bottom = ##f
    line-width = 90\mm
    left-margin = 7.5\mm
    bottom-margin = 5\mm
    top-margin = 5\mm
    tagline = ##f
  }

  % uncomment these lines for "letter" size
  %{
  \paper {
    #(set-paper-size "letter")
    ragged-last-bottom = ##f
    line-width = 7.5\in
    left-margin = 0.5\in
    bottom-margin = 0.25\in
    top-margin = 0.25\in
    tagline = ##f
   }
  %}

  % uncomment these lines for "A4" size
  %{
  \paper {
    #(set-paper-size "a4")
    ragged-last-bottom = ##f
    line-width = 180\mm
```

```
    left-margin = 15\mm
    bottom-margin = 10\mm
    top-margin = 10\mm
    tagline = ##f
  }
  %}
}
```



Creación de armaduras de clave personalizadas
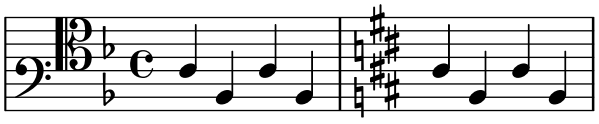
LilyPond contempla armaduras de tonalidad personalizadas. En este ejemplo se muestra una armadura de Re menor y Re mayor con un rango ampliado de bemoles mostrados.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil =
    #(lambda (grob)
       (grob-interpret-markup grob
```

```
          #{ \markup\combine
               \musicglyph "clefs.C"
               \translate #'(-3 . -2)
                 \musicglyph "clefs.F"
          #}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}

{
  \key d\minor f bes, f bes, |
  \key d\major fis b, fis b, |
}
```



Plicas de pentagrama cruzado

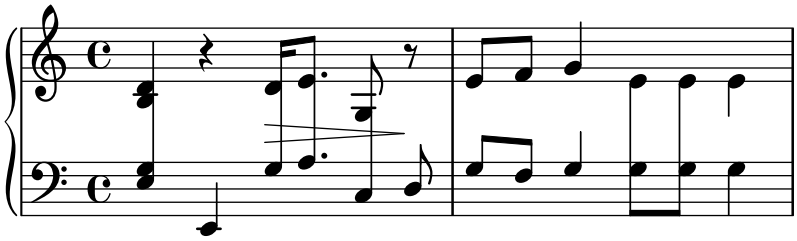
Este fragmento de código muestra el uso del grabador Span_stem_engraver y de \crossStaff para conectar automáticamente plicas de un pentagrama a otro.

No es necesario especificar las longitudes de las plicas porque la distancia variable entre las cabezas de las notas y los pentagramas se calcula automáticamente. Sin embargo, es importante aplicar \crossStaff a la voz o pentagrama correctos (esto es, en el lado opuesto de donde está o estaría posicionada la barra de corcheas) para conseguir el efecto deseado

```
\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

\new PianoStaff <<
  \new Staff {
    <b d'>4 r d'16\> e'8. g8 r\! |
    e'8 f' g'4
      \voiceTwo
      % Down to lower staff
      \crossStaff { e'8 e'8 } e'4 |
  }

  \new Staff {
    \clef bass
    \voiceOne
    % Up to upper staff
    \crossStaff { <e g>4 e, g16 a8. c8 } d |
    g8 f g4 \voiceTwo g8 g g4 |
  }
>>
```
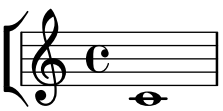


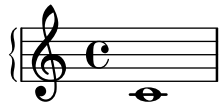
Mostrar corchete o llave en grupos de un solo pentagrama

Si hay un solo pentagrama en un contexto `ChoirStaff` o `StaffGroup`, el comportamiento predeterminado es que no se imprima el corchete en la barra inicial. Esto se puede cambiar estableciendo la propiedad `collapse-height` a un valor que sea menor que el número de líneas en la pauta.

Observe que en contextos como `PianoStaff` y `GrandStaff` en que los sistemas empiezan con una llave en lugar de un corchete, se debe establecer el valor de una propiedad distinta, como se ve en el segundo sistema del ejemplo.

```
\score {
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
```





Mostrar un sistema GrandStaff completo si sigue con vida uno solo de sus pentagramas

En muchas partituras orquestales es costumbre no mostrar los pentagramas de aquellos instrumentos que están en silencio durante bastante tiempo; esto se llama 'partitura francesa'. LilyPond aporta esta funcionalidad a través de la instrucción `\RemoveEmptyStaves`.

Cuando vuelven a sonar, con frecuencia se prefiere mostrar *todos los instrumentos del grupo*. Esto se puede hacer añadiendo el grabador `Keep_alive_together_engraver` al contexto agrupador (p.ej.: un `GrandStaff` o un `StaffGroup`).

En este ejemplo, los violines están en silencio durante el segundo sistema. Solo el violín primero suena en el último compás del tercer sistema, pero se muestra también el pentagrama del violín segundo.

```
\score {
  <<
    \new Staff = "Staff_flute" \with {
      instrumentName = "Flute"
      shortInstrumentName = "Fl"
    } \relative c' {
      \*3 { c'4 c c c | c c c c | c c c c | \break }
    }
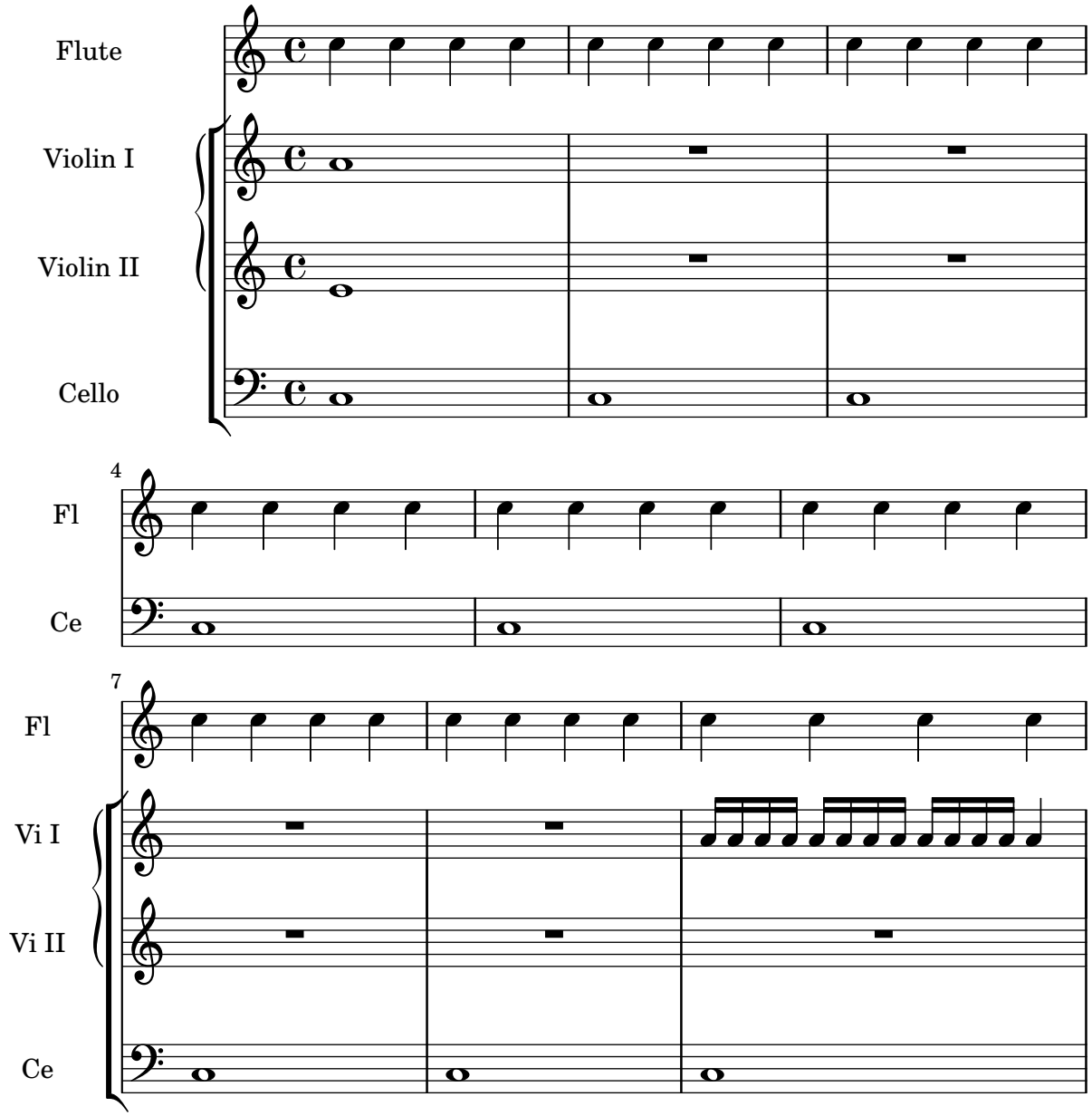
    \new StaffGroup = "StaffGroup_Strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
          shortInstrumentName = "Vi I"
        } \relative c'' {
          a1 | R1*7 | \*12 a16 a4 |
        }
        \new Staff = "StaffViolinII" \with {
          instrumentName = "Violin II"
          shortInstrumentName = "Vi II"
        } \relative c' {
          e1 | R1*8 |
        }
      >>

      \new Staff = "Staff_cello" \with {
        instrumentName = "Cello"
        shortInstrumentName = "Ce"
      } \relative c {
        \clef bass \*9 c1 |
      }
    >>
  >>
}

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm

  \context {
    \GrandStaff
    \consists Keep_alive_together_engraver
  }
  \context {
    \Staff
```

```
    \RemoveEmptyStaves
  }
}
```



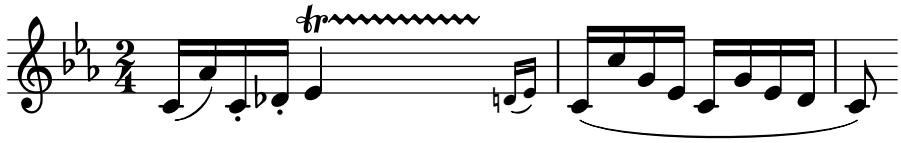
Extending a trill spanner

For `TrillSpanner` grobs, the `minimum-length` property becomes effective only if the `set-spacing-rods` procedure is called explicitly.

To do this, the `springs-and-rods` property should be set to `ly:spanner::set-spacing-rods`.

```
\relative c' {
  \key c\minor
  \time 2/4
  c16( as') c,-. des-.
  \once\override TrillSpanner.minimum-length = #15
  \once\override TrillSpanner.springs-and-rods = #ly:spanner::set-spacing-rods
```

```
  \afterGrace es4\startTrillSpan { d16[(\stopTrillSpan es)] }
  c( c' g es c g' es d
  c8)
}
```



Extender glissandos sobre repeticiones

Se puede simular un glissando que se extiende hasta el interior de varios bloques `\alternative` de primera y segunda vez mediante la adición de una nota de adorno oculta con un glissando al comienzo de cada bloque `\alternative`. La nota de adorno debe estar a la misma altura que la nota que da inicio al primer glissando. Esto se implementa aquí con una función musical que toma como argumento la altura de la nota de adorno.

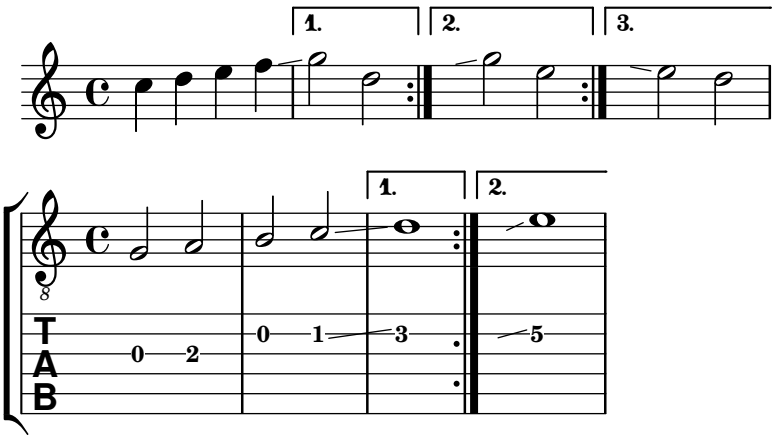
Observe que en música polifónica la nota de adorno debe coincidir con las notas de adorno correspondientes en todas las otras voces.

```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music =  \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}
```

```
\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff  <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
}
```



Ligaduras planas

Este fragmento de código aporta una función `flared-tie` para trazar una ligadura formada por líneas rectas. Su objetivo es sustituir a la función predeterminada de trazado de la ligadura (esto es, servir como un argumento sustitutorio para la propiedad `stencil` del objeto gráfico `Tie`).

El argumento de `flared-tie` es una lista de pare de coordenadas que especifican puntos adicionales entre el primero y el último para abarcar todas las líneas de la ligadura. Los puntos primero y último son idénticos a los puntos de inicio y final de la ligadura, respectivamente. Los valores de las coordenadas X e Y son múltiplos de la longitud y la altura del rectángulo circundante (teniendo también en cuenta la dirección de la ligadura); por tanto, el primer punto tiene las coordenadas (0,0), y el último (1,0).

La función `flare-tie` define una forma abreviada de especificar una ligadura plana. Es posible efectuar un trucaje adicional de la forma mediante la sobreescritura de `Tie.details.height-limit` o con la función `\shape`.

También es posible cambiar la definición personalizada sobre la marcha.

```
#(define ((flared-tie coords) grob)
   (define (pair-to-list pair)
     (list (car pair) (cdr pair)))

   (define (normalize-coords goods x y dir)
     (map
      (lambda (coord)
        (cons (* x (car coord)) (* y dir (cdr coord))))
      goods))

   (define (my-c-p-s points thick)
```

```
    (make-connected-path-stencil points thick 1.0 1.0 #f #f))

  ;; Calling `ly:tie::print` and assigning its return value to a
  ;; variable in this outer `let` triggers LilyPond to position the
  ;; tie, allowing us to extract its extents.  We only proceed,
  ;; however, if the tie doesn't get discarded (for whatever reason).
  (let ((sten (ly:tie::print grob)))
    (if (grob::is-live? grob)
        (let* ((layout (ly:grob-layout grob))
               (line-thickness (ly:output-def-lookup layout
                                                     'line-thickness))
               (thickness (ly:grob-property grob 'thickness 0.1))
               (used-thick (* line-thickness thickness))
               (dir (ly:grob-property grob 'direction))
               (xex (ly:stencil-extent sten X))
               (yex (ly:stencil-extent sten Y))
               (lenx (interval-length xex))
               (leny (interval-length yex))
               (xtrans (car xex))
               (ytrans (if (> dir 0)(car yex) (cdr yex)))
               ;; Add last point.
               (coord-list (append coords '((1.0 . 0.0))))
               (uplist
                (map pair-to-list
                     (normalize-coords coord-list lenx (* leny 2) dir))))
          (ly:stencil-translate
           (my-c-p-s uplist used-thick)
           (cons xtrans ytrans)))
        '())))

% Define a default tie shape consisting of three straight lines.
#(define flare-tie
   (flared-tie '((0.1 . 0.3) (0.9 . 0.3))))

\relative c' {
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <a c e a c e a c e>~ q
  \once \override Tie.stencil = #flare-tie
  q~ q\break

  <>^\markup \small \typewriter "height-limit = 14"
  \override Tie.details.height-limit = 14
  a'4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

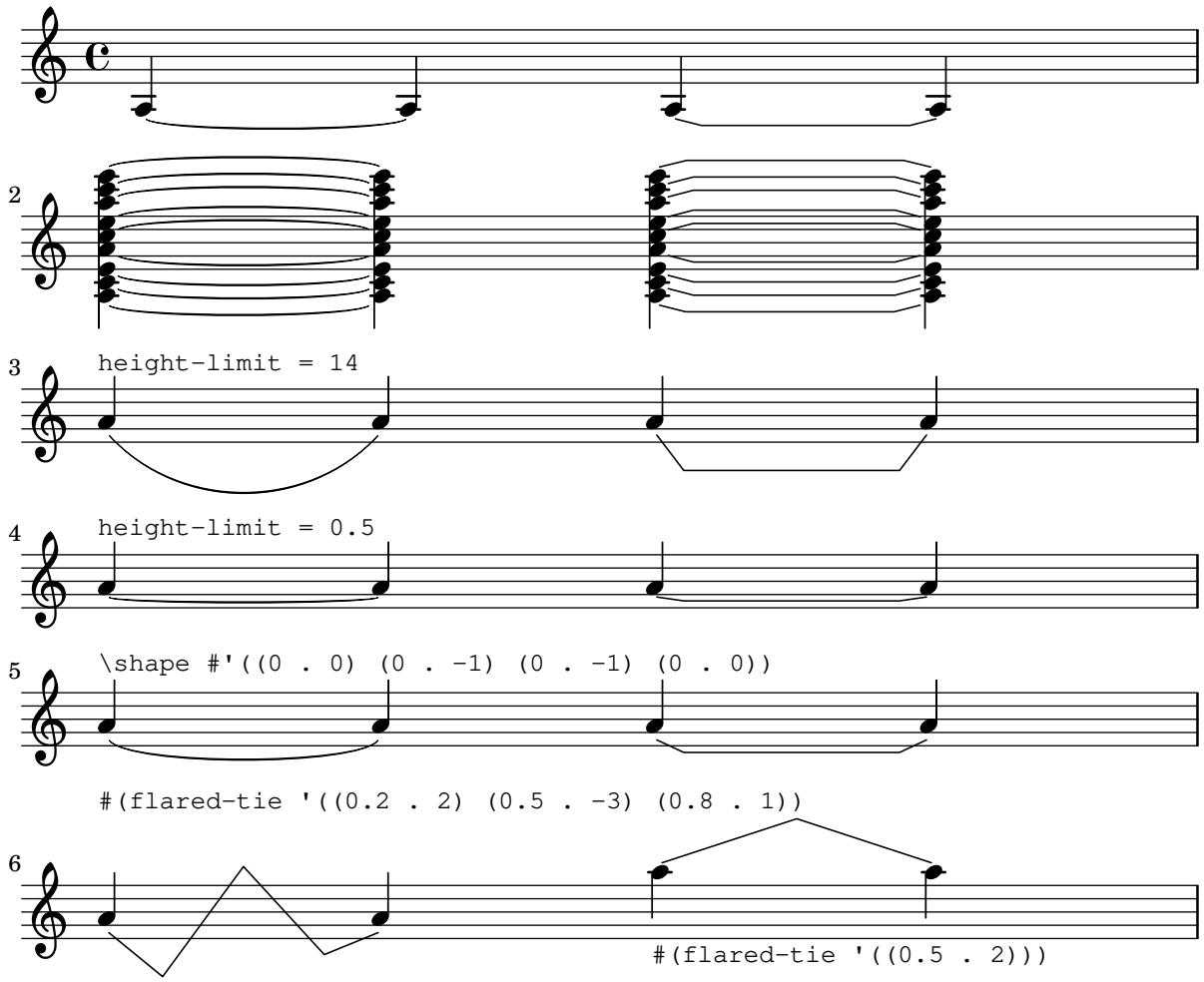
  <>^\markup \small \typewriter "height-limit = 0.5"
  \override Tie.details.height-limit = 0.5
  a4~ a
```

```
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  \revert Tie.details.height-limit

  <>^\markup \small \typewriter
            "\\shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0))"
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a
  \once \override Tie.stencil = #flare-tie
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a \break

  <>^\markup \small \typewriter
            "#(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1))"
  \once \override Tie.stencil =
        #(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))
  a4~ a
  <>_\markup \small \typewriter
            "#(flared-tie '((0.5 . 2)))"
  \once \override Tie.stencil = #(flared-tie '((0.5 . 2)))
  a'4~ a
}
```



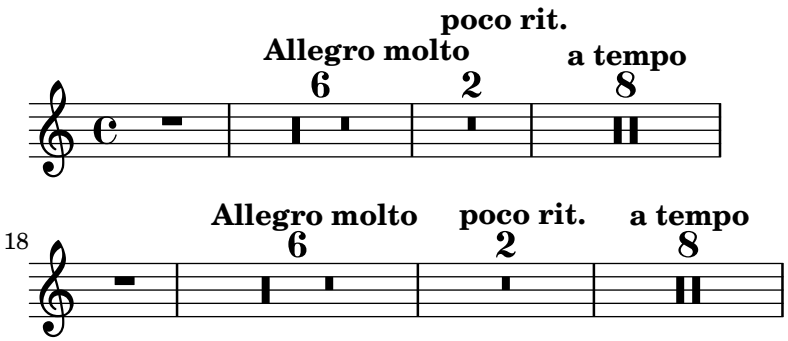
Forcing measure width to adapt to a metronome mark's width

By default, metronome marks do not influence horizontal spacing. This can be solved through a simple override, as shown in the second half of the example.

```
example = {
  R1
  \tempo "Allegro molto" R1*6
  \tempo "poco rit." R1*2
  \tempo "a tempo" R1*8 \break
}

{
  \compressMMRests {
    \example
    \override Score.MetronomeMark.extra-spacing-width = #'(-3 . 0)
    \example
  }
}
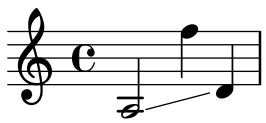
\layout {
  ragged-right = ##t
}
```



Glissando por debajo de un objeto gráfico

Los objetos gráficos de columna de nota (los grobs `NoteColumn`) pueden ser sobrepasados por los glissandos.

```
\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}
```



Harmonizing bar line thickness for staves with different sizes

When using `\magnifyStaff` only for some staves in a `StaffGroup`, `BarLine` grobs do not align any more due to its changed properties `thick-thickness`, `hair-thickness`, and `kern`.

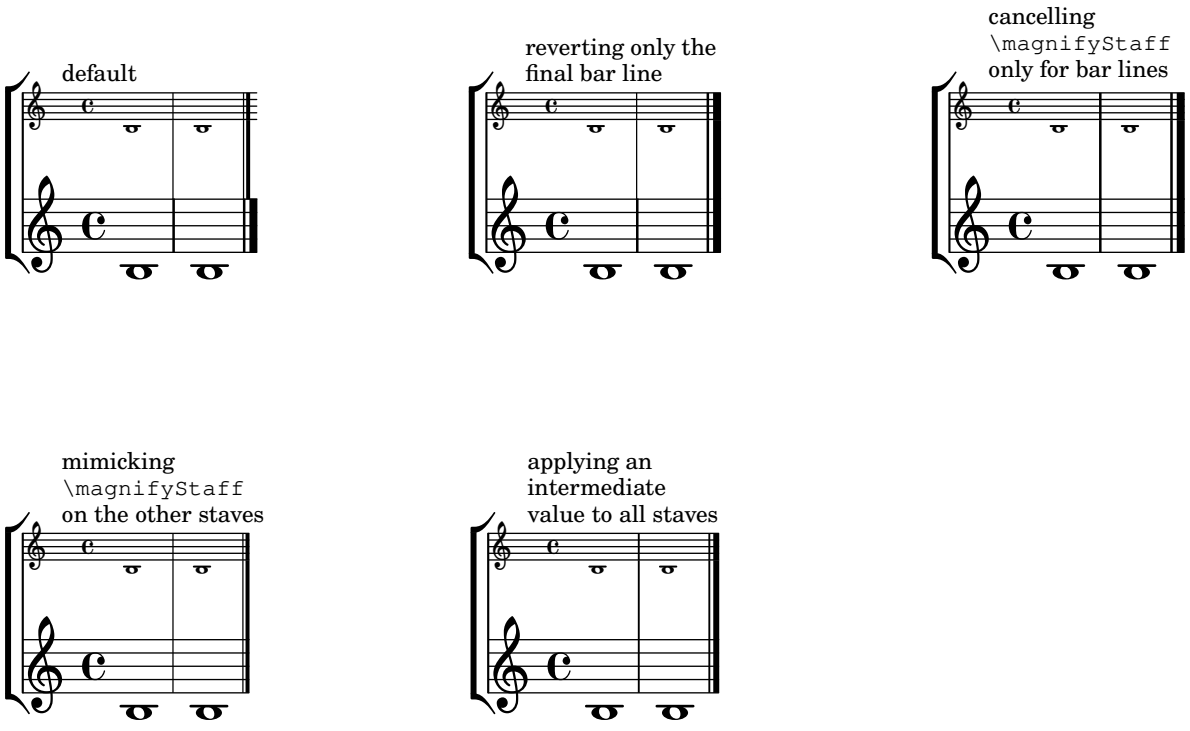
To fix this, multiple workarounds are available, as demonstrated below.

```
\markuplist {
  % First row.
  \fill-line {
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny "default"
          b1 b \bar "|."
        }
        \new Staff { b b }
      >>
    }
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny \column { "reverting only the"
                                            "final bar line" }
          b1 b
          \revert Staff.BarLine.thick-thickness
          \revert Staff.BarLine.hair-thickness
          \revert Staff.BarLine.kern
          \bar "|."
        }
        \new Staff { b b }
      >>
    }
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2
                           #(revert-props 'magnifyStaff 0
                                          '((BarLine thick-thickness)
                                            (BarLine hair-thickness)
                                            (BarLine kern))) } {
          \textMark \markup \tiny \column { "cancelling"
                                            \typewriter "\\magnifyStaff"
                                            "only for bar lines" }
          b1 b \bar "|."
        }
        \new Staff { b b }
      >>
    }
  }

  \vspace #2

  % Second row.
  \fill-line {
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny \column { "mimicking"
                                            \typewriter "\\magnifyStaff"
```

```
                                          "on the other staves" }
          b1 b \bar "|." }
        \new Staff \with { #(scale-props 'magnifyStaff 1/2 #t
                                         '((BarLine thick-thickness)
                                           (BarLine hair-thickness)
                                           (BarLine kern))) } {
          b b }
      >>
    }
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2
                           #(scale-props 'magnifyStaff 3/2 #t
                                         '((BarLine thick-thickness)
                                           (BarLine hair-thickness)
                                           (BarLine kern))) } {
          \textMark \markup \tiny \column { "applying an"
                                            "intermediate"
                                            "value to all staves" }
          b1 b \bar "|." }
        \new Staff \with { #(scale-props 'magnifyStaff 3/4 #t
                                         '((BarLine thick-thickness)
                                           (BarLine hair-thickness)
                                           (BarLine kern))) } {
           b b }
      >>
    }
    ""
  }
}
```



Incipits

Al transcribir música mensural, es útil poner un incipit al comienzo de la pieza para indicar la tonalidad y el compás originales. Aunque en la actualidad los músicos están acostumbrados a las líneas divisorias para reconocer más rápidamente los patrones rítmicos, en el período de la música mensural aún no se habían inventado las barras de compás; de hecho, el metro cambiaba con frecuencia cada pocas notas. Como compromiso, las barras de compás se imprimían muchas veces entre las pautas en lugar de dibujarse atravesando las líneas del pentagrama.

```
% A short excerpt from the Jubilate Deo by Orlande de Lassus

global = {
  \set Score.skipBars = ##t
  \key g \major
  \time 4/4

  % the actual music
  \skip 1*8

  % let finis bar go through all staves
  \override Staff.BarLine.transparent = ##f

  % finis bar
  \bar "|."
}

discantusIncipit = \new PetrucciStaff {
  \clef "petrucci-c1"
  \key f \major
  \time 2/2
  c''1.
}

discantusNotes = {
  \transpose c' c'' {
    \clef "treble"
    d'2. d'4 |
    b e' d'2 |
    c'4 e'4.( d'8 c' b |
    a4) b a2 |
    b4.( c'8 d'4) c'4 |
    \once \hide NoteHead
    c'1 |
    b\breve |
  }
}

discantusLyrics = \lyricmode {
  Ju -- bi -- la -- te De -- o,
  om -- nis ter -- ra, __ om-
  "..."
  -us.
}
```

```
altusIncipit = \new PetrucciStaff {
  \clef "petrucci-c3"
  \key f \major
  \time 2/2
  e'1\rest f'1.
}

altusNotes = {
  \transpose c' c'' {
    \clef "treble"
    r2 g2. e4 fis g |
    a2 g4 e |
    fis g4.( fis16 e fis4) |
    g1 |
    \once \hide NoteHead
    g1 |
    g\breve |
  }
}

altusLyrics = \lyricmode {
  Ju -- bi -- la -- te
  De -- o, om -- nis ter -- ra,
  "..."
  -us.
}

tenorIncipit = \new PetrucciStaff {
  \clef "petrucci-c4"
  \key f \major
  \time 2/2
  r\longa
  r\breve
  r1 c'1.
}

tenorNotes = {
  \transpose c' c' {
    \clef "treble_8"
    R1 |
    R1 |
    R1 |
    % two measures
    r2 d'2. d'4 b e' |
    \once \hide NoteHead
    e'1 |
    d'\breve |
  }
}

tenorLyrics = \lyricmode {
```

```
  Ju -- bi -- la -- te
  "..."
  -us.
}
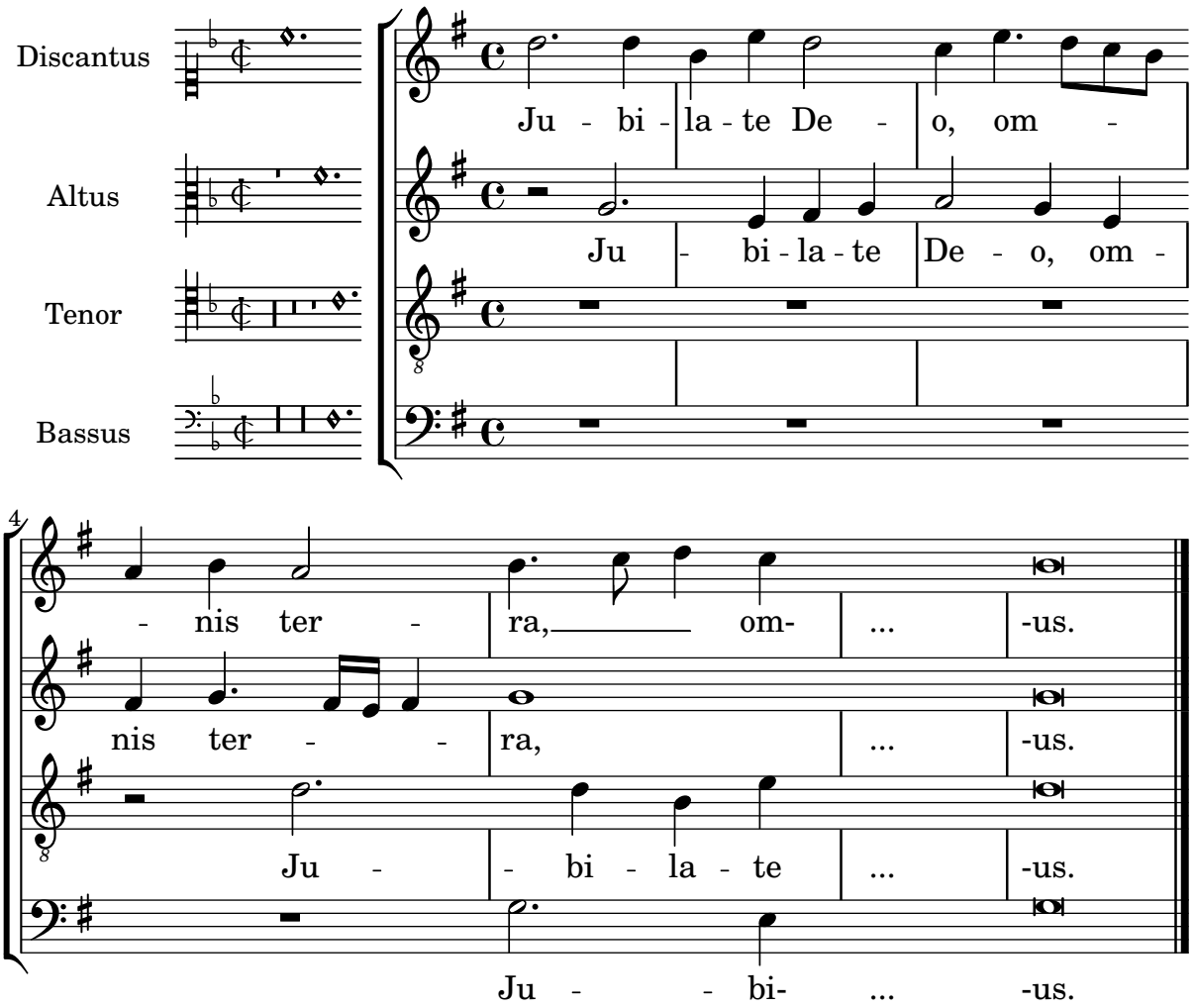
bassusIncipit = \new PetrucciStaff {
  % The original print shows the b flat
  % for the f major key signature twice.
  \override Staff.KeySignature.flat-positions = #'((-7 . 6))
  \clef "mensural-f"
  \key f\major
  \time 2/2
  \tweak Y-offset #1 r\longa \tweak Y-offset #1 r\longa
  f1.
}

bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \hide NoteHead
    e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  Ju -- bi-
  "..."
  -us.
}

\score {
  <<
    \new StaffGroup = choirStaff <<
      \new Voice = "discantusNotes" <<
        \set Staff.instrumentName = "Discantus"
        \incipit #1 \discantusIncipit
        \global
        \discantusNotes
      >>
      \new Lyrics \lyricsto discantusNotes { \discantusLyrics }
      \new Voice = "altusNotes" <<
        \set Staff.instrumentName = "Altus"
        \global
        \incipit #1 \altusIncipit
        \altusNotes
      >>
```

```
      \new Lyrics \lyricsto altusNotes { \altusLyrics }
      \new Voice = "tenorNotes" <<
        \set Staff.instrumentName = "Tenor"
        \global
        \incipit #1 \tenorIncipit
        \tenorNotes
      >>
      \new Lyrics \lyricsto tenorNotes { \tenorLyrics }
      \new Voice = "bassusNotes" <<
        \set Staff.instrumentName = "Bassus"
        \global
        \incipit #1 \bassusIncipit
        \bassusNotes
      >>
      \new Lyrics \lyricsto bassusNotes { \bassusLyrics }
    >>
  >>
  \layout {
    \context {
      \Score
      %% no bar lines in staves or lyrics
      \hide BarLine
    }
    %% the next two instructions keep the lyrics between the bar lines
    \context {
      \Lyrics
      \consists "Bar_engraver"
      \consists "Separating_line_group_engraver"
    }
    \context {
      \Voice
      %% no slurs
      \hide Slur
      %% Comment in the below "\remove" command to allow line
      %% breaking also at those bar lines where a note overlaps
      %% into the next measure.  The command is commented out in this
      %% short example score, but especially for large scores, you
      %% will typically yield better line breaking and thus improve
      %% overall spacing if you comment in the following command.
      %%\remove "Forbid_line_break_engraver"
    }
    indent = 5\cm
    incipit-width = 2.5\cm
  }
}
```



Insertar fragmentos de partituras sobre el pentagrama, como elementos de marcado

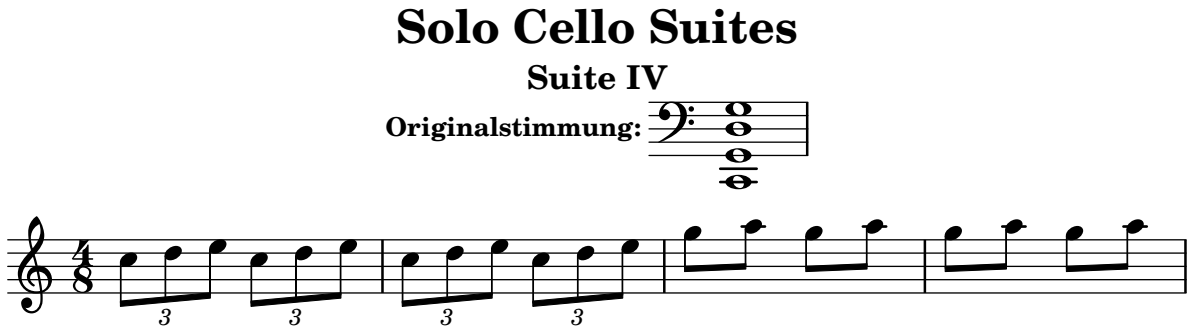
La instrucción \markup es muy versátil. En este fragmento de código, contiene un bloque \score con una partitura completa en lugar de textos u otras marcas.

```
tuning = \markup \score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \clef bass
    <c, g, d g>1
  }
  \layout {
    indent = 0\cm
  }
}

\header {
  title = "Solo Cello Suites"
  subtitle = "Suite IV"
  subsubtitle = \markup { Originalstimmung: \raise #0.5 \tuning }
  tagline = ##f
}
```

```
\layout {
  ragged-right = ##f
}

\relative c'' {
  \time 4/8
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { c d e }
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { c d e }
  g8 a g a
  g8 a g a
}
```



Hacer que TabStaff imprima la cuerda superior en la parte de abajo

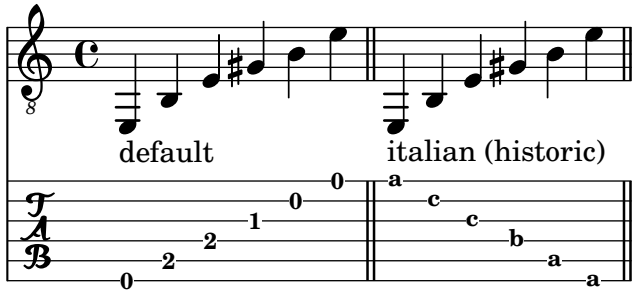
En las tablaturas, la primera cuerda se imprime normalmente en la parte de arriba. Si queremos que esté en la parte de abajo, hay que cambiar la propiedad de contexto `stringOneTopmost` al valor `#f`. Para un ajuste al nivel del contexto, se puede hacer también esta asignacion dentro del bloque `\layout`.

```
%\layout {
%  \context {
%    \Score
%    stringOneTopmost = ##f
%  }
%  \context {
%    \TabStaff
%    tablatureFormat = #fret-letter-tablature-format
%  }
%}

m = {
  \cadenzaOn
  e, b, e gis! b e'
  \bar "||"
}

<<
  \new Staff {
    \clef "G_8"
    <>_"default" \m
    <>_"italian (historic)"\m
```

```
  }
  \new TabStaff
  {
    \m
    \set Score.stringOneTopmost = ##f
    \set TabStaff.tablatureFormat = #fret-letter-tablature-format
    \m
  }
>>
```

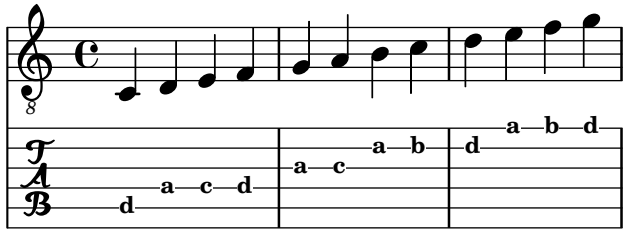


Formateado de tablaturas con letras

La tablatura se puede formatear utilizando letras en lugar de números.

```
music = \relative c {
  c4 d e f
  g4 a b c
  d4 e f g
}

<<
  \new Staff {
    \clef "G_8"
    \music
  }
  \new TabStaff \with {
    tablatureFormat = #fret-letter-tablature-format
  } {
    \music
  }
>>
```



Hacer que los glissandos se puedan dividir en el salto de línea

Normalmente, LilyPond rehúsa insertar un salto de línea automático en el punto en que un glissando cruza la barra de compás. Este comportamiento se puede cambiar estableciendo el valor de la propiedad `Glissando.breakable` a `#t`. También, estableciendo el valor de la propiedad `after-line-breaking` a `#t` hace que la línea del glissando continúe después del salto.

La propiedad `breakable` no afecta a los saltos manuales insertados con instrucciones como `\break`.

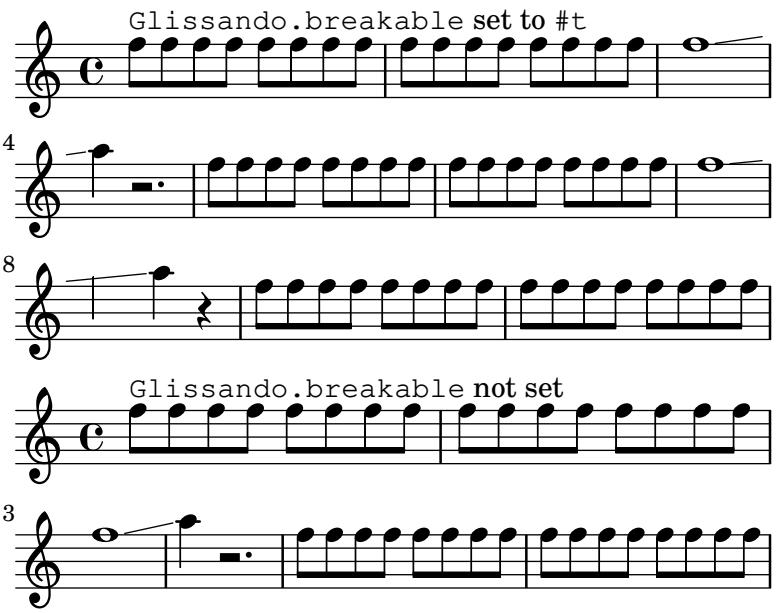
```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
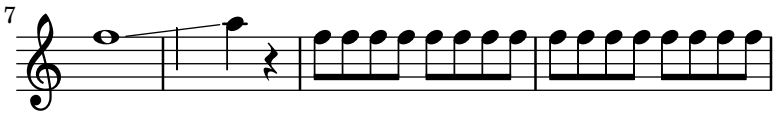
music = {
  \*16 f8 |
  f1\glissando |
  a4 r2. |
  \*16 f8 |
  f1\glissando \once\glissandoSkipOn |
  a2 a4 r4 |
  \*16 f8
}

\relative c'' {
  <>^\markup { \typewriter Glissando.breakable
               set to \typewriter "#t" }
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  \music
}

\relative c'' {
  <>^\markup { \typewriter Glissando.breakable not set }
  \music
}

\paper {
  line-width = 100\mm
}
```
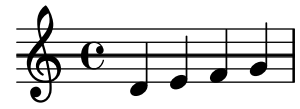




Hacer unas líneas del pentagrama más gruesas que las otras

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Measure counters

This snippet demonstrates the use of the `Measure_counter_engraver` to number groups of successive measures. Any stretch of measures may be numbered, whether consisting of repetitions or not.

The engraver must be added to the appropriate context. Here, a `Staff` context is used; another possibility is a `Dynamics` context.

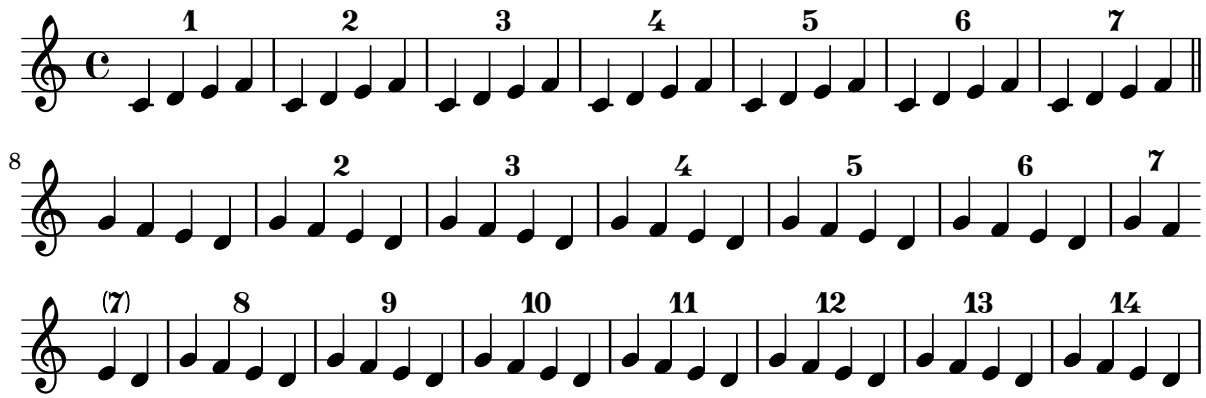
The counter is begun with `\startMeasureCount` and ended with `\stopMeasureCount`. Numbering will start by default with 1, but this behavior may be modified by overriding the `count-from` property.

When a measure extends across a line break, the number will appear twice, the second time in parentheses.

```
\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

\new Staff {
  \startMeasureCount
  \*7 { c'4 d' e' f' }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \*5 { g'4 f' e' d' }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \*7 { g'4 f' e' d' }
```

```
  \stopMeasureCount
}
```



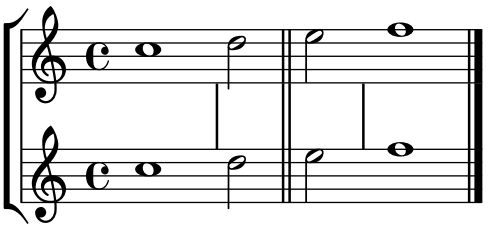
Disposición *Mensurstriche* (líneas divisorias entre pentagramas)

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir fijando `measureBarType` al valor `"-span|"` y usando un contexto de agrupamiento que admita barras extendidas, como `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

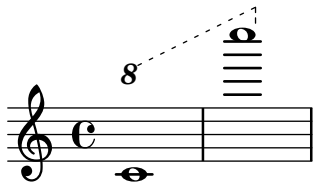
\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



Modificación de la inclinación de la línea de octava

Es posible cambiar la inclinación de la línea de octava alta o baja.

```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0)
                (attach-dir . ,LEFT)
                (padding . 0)
                (stencil-align-dir-y . ,CENTER)))
       (right . ((Y . 5.0) ; Change the number here
                 (padding . 0)
                 (attach-dir . ,RIGHT)
                 (text . ,(make-draw-dashed-line-markup
                           (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
     #ly:horizontal-line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
     #ly:horizontal-line-spanner::calc-right-bound-info
  \ottava 1
  c1
  c'''1
}
```



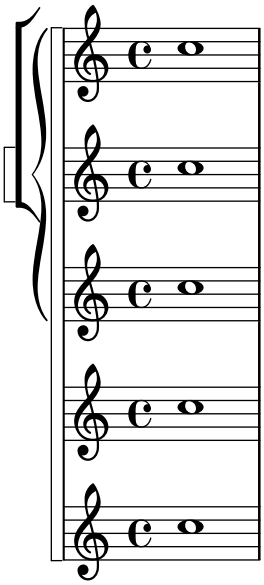
Anidado de grupos de pentagramas

Se puede utilizar la propiedad `systemStartDelimiterHierarchy` para crear grupos de pentagramas anidados de forma más compleja. La propiedad `systemStartDelimiterHierarchy` del contexto `StaffGroup` toma una lista alfabética del número de pentagramas producidos. Se puede proporcionar antes de cada pentagrama un delimitador de comienzo de sistema. Se debe encerrar entre corchetes y admite tantos pentagramas como encierren las llaves. Se pueden omitir los elementos de la lista, pero el primer corchete siempre abarca todos los pentagramas. Las posibilidades son `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` y `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = 4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare
          (SystemStartBrace
            (SystemStartBracket a
              (SystemStartSquare b))
            c)
          d)

  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
```

>>



Armaduras de tonalidad no tradicionales

La muy utilizada instrucción `\key` establece la propiedad `keyAlterations` dentro del contexto `Staff`.

Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

```
\set Staff.keyAlterations =
  #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)
```
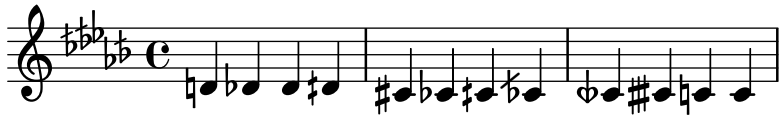
donde, para cada elemento dentro de la lista, *octava* especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), *paso* especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y *alteración* es `SHARP`, `FLAT` o `DOUBLE-SHARP`, etc., precedido de una coma.

De forma alternativa, podemos usar el formato más conciso (*paso . alteración*) para cada elemento de la lista si se usa la misma alteración en todas las octavas.

Para escalas microtonales en las que un "sostenido" no son 100 cents, *alteración* se refiere a la proporción de un tono entero de 200 cents.

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
                                 (6 . ,SEMI-FLAT))
  % \set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Plantilla de orquesta, coro y piano

Esta plantilla muestra el uso de contextos `StaffGroup` y `GrandStaff` anidados para sub-agrupar instrumentos del mismo tipo, y una forma de usar `\transpose` de manera que unas variables contengan la música para instrumentos transpositores en afinación de concierto.

```
#(set-global-staff-size 17)

\paper {
  indent = 3.0\cm  % add space for instrumentName
  short-indent = 1.5\cm  % add less space for shortInstrumentName
}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.
clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns
hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }
sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }
altoILyrics = \sopranoLyrics
altoIIMusic = \relative c' { \key g \major g'1 b }
altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }
tenorLyrics = \sopranoLyrics

pianoRHMusic = \relative c { \key g \major g''1 b }
pianoLHMusic = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }
violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }
```

```
bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\book {
  \score {
    <<
      \new StaffGroup = "StaffGroup_woodwinds" <<
        \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
          \fluteMusic

        \new Staff = "Staff_clarinet" \with {
          instrumentName = \markup { \concat { "Clarinet in B" \flat } }
        }
          % Declare that written Middle C in the music
          % to follow sounds a concert B flat, for
          % output using sounded pitches such as MIDI.
          %\transposition bes

          % Print music for a B-flat clarinet
          \transpose bes c' \clarinetMusic
      >>

      \new StaffGroup = "StaffGroup_brass" <<
        \new Staff = "Staff_hornI" \with {
          instrumentName = "Horn in F"
        }
          % \transposition f
          \transpose f c' \hornMusic

        \new Staff = "Staff_trumpet" \with {
          instrumentName = "Trumpet in  C"
        }
          \trumpetMusic
      >>

      \new RhythmicStaff = "RhythmicStaff_percussion" \with {
        instrumentName = "Percussion"
      }
        \percussionMusic

      \new PianoStaff \with {
        instrumentName = "Piano"
      } <<
        \new Staff { \pianoRHMusic }
        \new Staff { \pianoLHMusic }
      >>

      \new ChoirStaff = "ChoirStaff_choir" <<
        \new Staff = "Staff_soprano" \with {
          instrumentName = "Soprano"
        }
          \new Voice = "soprano" \sopranoMusic
          \new Lyrics \lyricsto "soprano" { \sopranoLyrics }
```

```
    \new GrandStaff = "GrandStaff_altos" \with {
      \accepts Lyrics
    } <<
      \new Staff = "Staff_altoI" \with {
        instrumentName = "Alto I"
      }
        \new Voice = "altoI"
        \altoIMusic
        \new Lyrics \lyricsto "altoI" { \altoILyrics }
      \new Staff = "Staff_altoII" \with {
        instrumentName = "Alto II"
      }
        \new Voice = "altoII"
        \altoIIMusic
        \new Lyrics \lyricsto "altoII" { \altoIILyrics }
    >>

    \new Staff = "Staff_tenor" \with {
      instrumentName = "Tenor"
    }
      \new Voice = "tenor" \tenorMusic
      \new Lyrics \lyricsto "tenor" { \tenorLyrics }
  >>

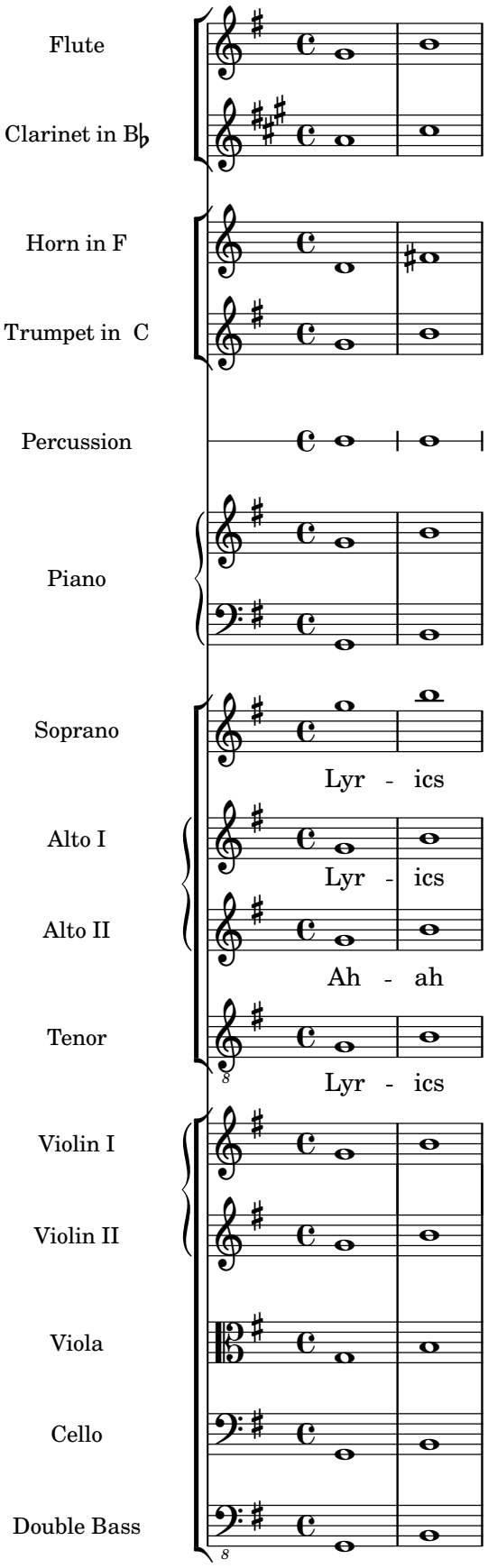
  \new StaffGroup = "StaffGroup_strings" <<
    \new GrandStaff = "GrandStaff_violins" <<
      \new Staff = "Staff_violinI" \with {
        instrumentName = "Violin I"
      }
        \violinIMusic
      \new Staff = "Staff_violinII" \with {
        instrumentName = "Violin II"
      }
        \violinIIMusic
    >>

    \new Staff = "Staff_viola" \with {
      instrumentName = "Viola"
    }
      \violaMusic

    \new Staff = "Staff_cello" \with {
      instrumentName = "Cello"
    }
      \celloMusic

    \new Staff = "Staff_bass" \with {
      instrumentName = "Double Bass"
    }
      \bassMusic
  >>
```

```
    >>
  }
}
```


Flute
Clarinet in B♭
Horn in F
Trumpet in C
Percussion
Piano
Soprano
Lyr - ics
Alto I
Lyr - ics
Alto II
Ah - ah
Tenor
Lyr - ics
Violin I
Violin II
Viola
Cello
Double Bass

Print chord names with same root and different bass as slash and bass note

To print subsequent chord names only differing in their bass note as slash and bass note without a root symbol, use the Scheme engraver defined in this snippet. The behaviour is controlled by the chordChanges context property.

```
#(define Bass_changes_equal_root_engraver
   (lambda (ctx)
     "Drop root for follow-up chord names that differ only in bass.

In other words, the chord name sequence 'D D/C D/B' gets actually
printed as 'D /C /B'.

Set the `chordChanges` context property to `#t` to activate this
feature."
     (let ((chord-pitches '())
           (last-chord-pitches '())
           (bass-pitch #f))
       (make-engraver
        ((initialize this-engraver)
         (let ((chord-note-namer (ly:context-property ctx
                                                      'chordNoteNamer)))
           ;; Set `chordNoteNamer`, respecting user setting if already
           ;; done.
           (ly:context-set-property! ctx 'chordNoteNamer
                                     (if (procedure? chord-note-namer)
                                         chord-note-namer
                                         (chord-name:markup #f)))))

        (listeners
         ((note-event this-engraver event)
          (let* ((pitch (ly:event-property event 'pitch))
                 (pitch-name (ly:pitch-notename pitch))
                 (pitch-alt (ly:pitch-alteration pitch))
                 (bass (ly:event-property event 'bass #f))
                 (inversion (ly:event-property event 'inversion #f)))
            ;; We look at the `bass` and `inversion` event properties
            ;; to decide how to handle the current note event.  If
            ;; `inversion` is set we add the bass note to the chord as
            ;; an ordinary member so that we can compare inversed
            ;; chords; if `bass` is set the bass note is not added to
            ;; the chord.
            ;;
            ;; In the `chord-pitches` list we actually collect only
            ;; the notes' pitch names (which are integers) and pitch
            ;; alterations as pairs, ignoring the octave.
            (cond (bass (set! bass-pitch pitch))
                  (inversion
                   (set! bass-pitch pitch)
                   (set! chord-pitches
                         (cons (cons pitch-name pitch-alt)
                               chord-pitches)))
                  (else
```

```
                (set! chord-pitches
                      (cons (cons pitch-name pitch-alt)
                            chord-pitches)))))))

        (acknowledgers
         ((chord-name-interface this-engraver grob source-engraver)
          (let ((chord-changes (ly:context-property ctx
                                                    'chordChanges #f)))
            ;; If subsequent chords are equal apart from their bass,
            ;; we change the `text` property to print only the slash
            ;; and the bass note (via the formatter stored in the
            ;; `chordNoteNamer` context property).
            ;;
            ;; Equality is tested by comparing the sorted lists of
            ;; this chord's elements and the previous chord.  Sorting
            ;; is needed because inverted chords may have a different
            ;; order of pitches.  Note that we only do a simplified
            ;; sorting using the pitch name, ignoring the alteration.
            (if (and bass-pitch
                     chord-changes
                     (equal? (sort chord-pitches car<)
                             (sort last-chord-pitches car<)))
                (ly:grob-set-property!
                 grob 'text
                 (make-line-markup
                  (list
                   (ly:context-property ctx 'slashChordSeparator)
                   ((ly:context-property ctx 'chordNoteNamer)
                    bass-pitch
                    (ly:context-property ctx
                                         'chordNameLowercaseMinor))))))
            (set! last-chord-pitches chord-pitches)
            (set! chord-pitches '())
            (set! bass-pitch #f))))

        ((finalize this-engraver)
         (set! last-chord-pitches '()))))


myChords = \chordmode {
  % \germanChords

  \set chordChanges = ##t
  d2:m d:m/cis

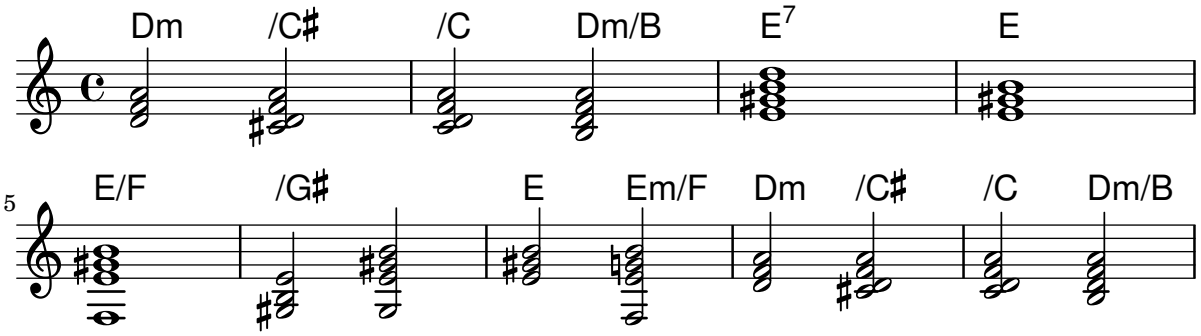
  d:m/c
  \set chordChanges = ##f
  d:m/b

  e1:7
  \set chordChanges = ##t
  e
```

```
  \break

  \once \set chordChanges = ##f
  e1/f
  e2/gis e/+gis e e:m/f d:m d:m/cis d:m/c
  \set chordChanges = ##f
  d:m/b
}
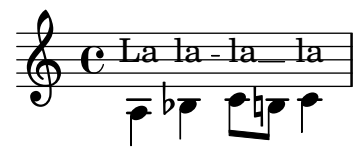
<<
  \new ChordNames
    \with { \consists #Bass_changes_equal_root_engraver }
    \myChords
  \new Staff \myChords
>>
```



Putting lyrics inside the staff

Se pueden mover las líneas de letra verticalmente para imprimirlas dentro del pentagrama. Las letras se mueven con `\override LyricText.extra-offset = #'(0 . dy)`, y hay instrucciones similares para mover las líneas extensoras y los guiones. El mejor valor para el desplazamiento *dy* se debe buscar mediante un proceso de ensayo y error.

```
<<
  \new Staff <<
    \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
  >>
  \new Lyrics \with {
    \override LyricText.extra-offset = #'(0 . 8.6)
    \override LyricExtender.extra-offset = #'(0 . 8.6)
    \override LyricHyphen.extra-offset = #'(0 . 8.6)
  } \lyricsto "voc" { La la -- la __ _ la }
>>
```



Citar otra voz

La propiedad de contexto `quotedEventTypes` determina los tipos de eventos musicales que resultan citados. El valor predeterminado es `(note-event rest-event tie-event beam-event`

tuplet-span-event), que significa que solo aparecen en la expresión \quoteDuring las notas, silencios, ligaduras, barras y grupos especiales.

En el ejemplo siguiente, el silencio de semicorchea no aparece en el fragmento citado porque rest-event no aparece dentro del valor redefinido de quotedEventTypes.

Para ver una lista de los tipos de evento, consulte la sección “Music classes” de la Referencia de funcionamiento interno.
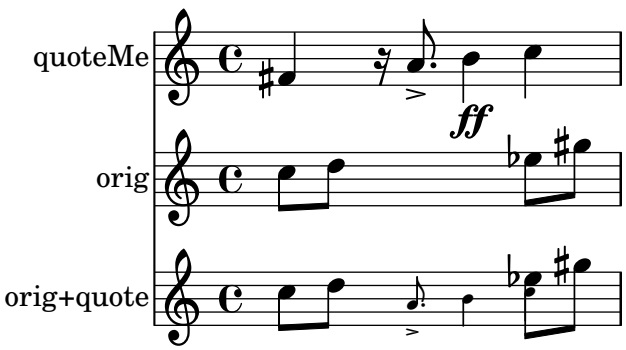
```
quoteMe = \relative c' {
  fis4 r16 a8.-> b4\ff c
}
\addQuote quoteMe \quoteMe

original = \relative c'' {
  c8 d s2
  \once \override NoteColumn.ignore-collision = ##t
  es8 gis8
}

<<
  \new Staff \with { instrumentName = "quoteMe" }
  \quoteMe

  \new Staff \with { instrumentName = "orig" }
  \original

  \new Staff \with {
    instrumentName = "orig+quote"
    quotedEventTypes = #'(note-event articulation-event)
  }
  \relative c''
  <<
    \original
    \new Voice {
      s4
      \set fontSize = #-4
      \override Stem.length-fraction = #(magstep -4)
      \quoteDuring "quoteMe" { \skip 2. }
    }
  >>
>>
```



Citar otra voz con transposición
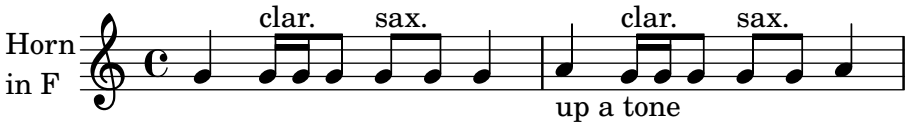
Los pasajes citados tienen en cuenta la transposición de la fuente tanto como la del destino. En este ejemplo, todos los instrumentos interpretan una nota con el sonido del Do central; el destino de un instrumento transpositor en Fa. La parte de destino se puede transportar utilizando `\transpose`. En este caso las notas citadas permanecen sin cambios.

```
\addQuote clarinet {
  \transposition bes
  \*8 { d'16 d' d'8 }
}

\addQuote sax {
  \transposition es'
  \*16 a8
}

quoteTest = {
  % french horn
  \transposition f
  g'4
  << \quoteDuring "clarinet" { \skip 4 } s4^"clar." >>
  << \quoteDuring "sax" { \skip 4 } s4^"sax." >>
  g'4
}

{
  \new Staff \with {
    instrumentName = \markup { \column { Horn "in F" } }
  }
  \quoteTest
  \transpose c' d' << \quoteTest s4_"up a tone" >>
}
```



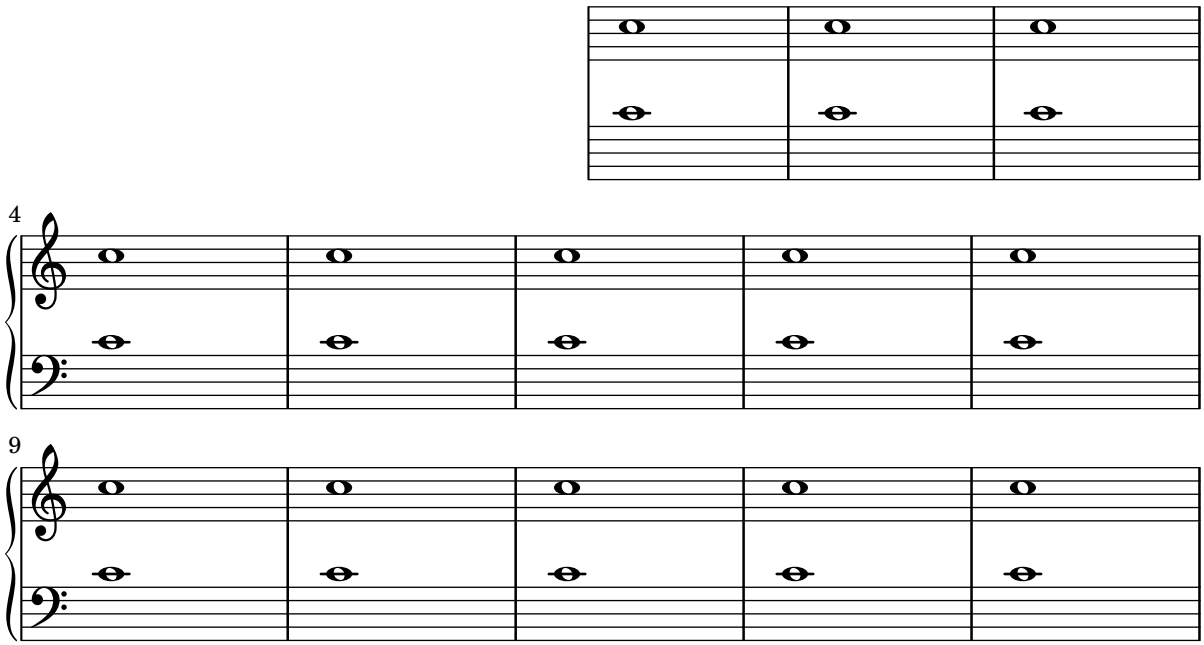
Quitar la llave en el primer sistema de una partitura de piano

Este fragmento elimina la primera llave de un `PianoStaff` o un `GrandStaff`, junto con las claves. Puede ser útil cuando se está cortando y pegando la imagen de la partitura editada dentro de otra música existente.

El código emplea `\alterBroken` para ocultar el delimitador en forma de llave que aparece al principio.

```
someMusic =  {
  \once \omit Staff.Clef
  \once \omit Staff.TimeSignature
  \*3 c1 \break
  \*5 c1 \break
  \*5 c1
}
```

```
\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c'' \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
    indent=75\mm
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}
```

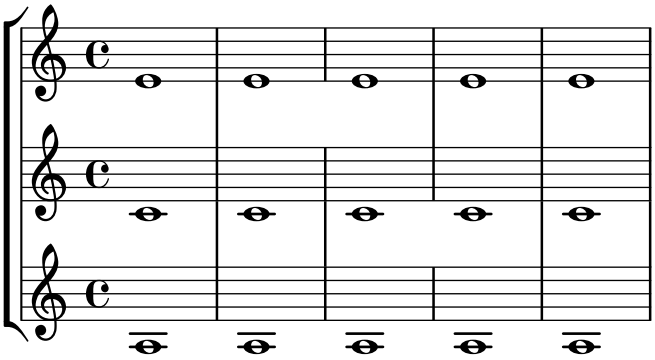


Quitar las barras de compás entre los pentagramas de un StaffGroup, PianoStaff o GrandStaff

De forma predeterminada, las líneas divisorias en los contextos `StaffGroup`, `PianoStaff` o `GrandStaff` se conectan entre los pentagramas mediante una línea. Se puede alterar este comportamiento pentagrama a pentagrama.

```
\relative c' {
  \new StaffGroup <<
    \new Staff {
      e1 | e
      \once \override Staff.BarLine.allow-span-bar = ##f
      e1 | e | e
    }
    \new Staff {
      c1 | c | c
      \once \override Staff.BarLine.allow-span-bar = ##f
      c1 | c
    }
```

```
    \new Staff {
      a1 | a | a | a | a
    }
  >>
}
```



Quitar la primera línea vacía

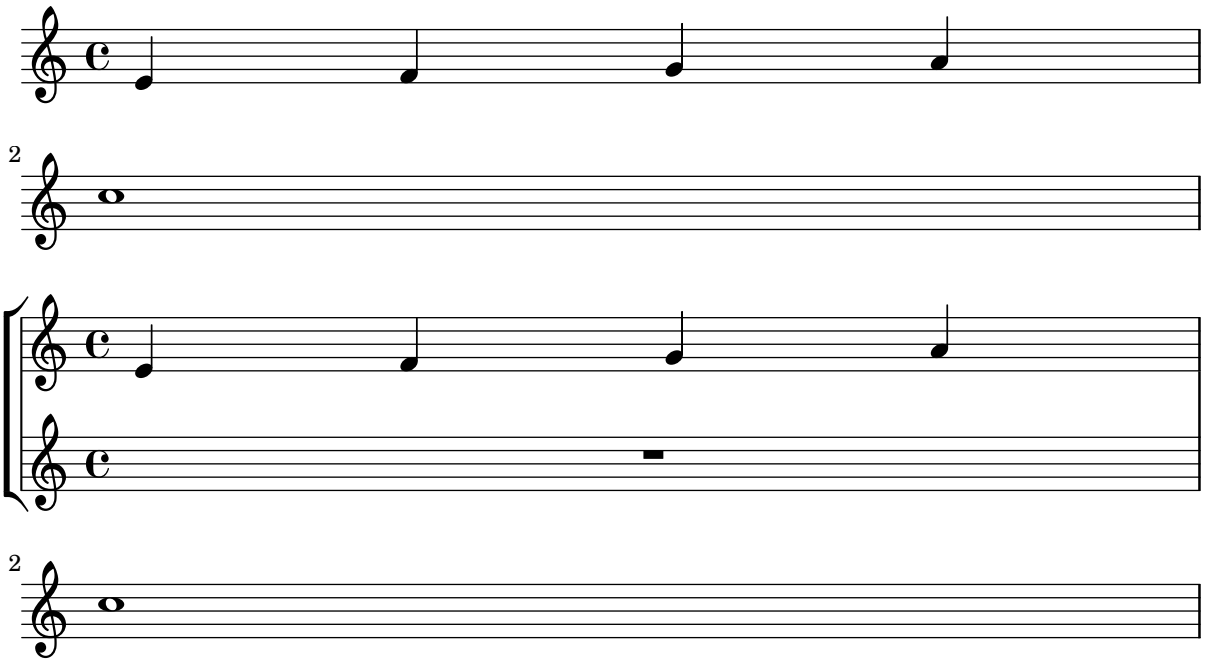
Para eliminar el primer pentagrama vacío de una aprtitura, establecemos la propiedad `remove-first` del objeto gráfico `VerticalAxisGroup` al valor `#t`. Esto se puede hacer globalmente dentro del bloque `\layout`, o localmente dentro del pentagrama concreto que se quiere suprimir. En este último caso, tenemos que especificar el contexto (`Staff` se aplica solo al pentagrama actual) delante de la propiedad.

El pentagrama inferior del segundo grupo no se elimina, porque el ajuste solo se aplica al pentagrama concreto dentro del que se escribe.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup.remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>

\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
```

```
  \new Staff {
    R1 \break
    R
  }
>>
```



Fijar un separador entre los sistemas

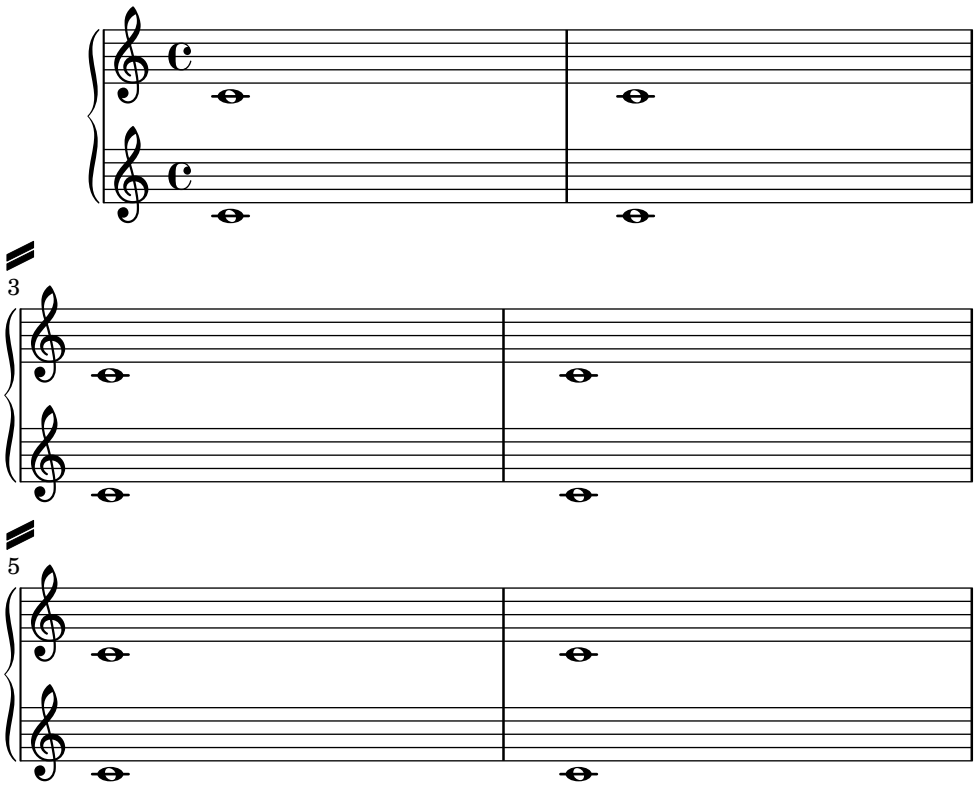
Se pueden insertar separadores sistema entre los sistemas de una página. Se puede usar cualquier elemento de marcado, pero `\slashSeparator` está disponible como una elección predeterminada adecuada.

```
#(set-default-paper-size "a5")

\paper {
  system-separator-markup = \slashSeparator
  tagline = ##f
}

notes = \relative c' {
  c1 | c \break
  c1 | c \break
  c1 | c
}

\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}
```



Dar forma a ligaduras de unión individuales entre acordes

Para dar forma a las ligaduras de unión individuales entre acordes use el método que se muestra más abajo.

```
{
  \textMark "Chords can be tied note by note."
  <c'~ e'~ g'~ c''~>2 q
}

{
  \textMark \markup \override #'(baseline-skip . 3) \wordwrap {
    Modifying those ties with \typewriter "\\shape" does not succeed,
    because \typewriter TieColumn positions them on its own behalf,
    ignoring \typewriter "\\shape" input more or less.  You may
    circumvent this by setting \typewriter positioning-done to
    \typewriter "#t" -- alas, \typewriter positioning-done is an
    internal property, and setting it to \typewriter "#t" means: all
    positioning is done, don't do anything further.  The next example
    demonstrates a case where the positioning is not finished: all tie
    directions are down, and the thickness is not accurate.
  }
  <c'~ e'~ g'~ c''~>2
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark "To fix that, enter ties with explicit direction modifiers."
  <c'_~ e'_~ g'_~ c''^~>2
```

```
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark \markup {
    Now you can use \typewriter "\\shape" for each tie as usual. }
  <c'-\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
   e'-\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
   g'-\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
   c''-\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
  >2
  \once \override TieColumn.positioning-done = ##t
  q
}

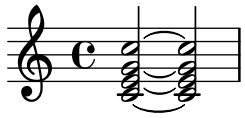
{
  \textMark "This also works at line breaks."
  <c'-\shape #'(((0 . 0) (0 . -10) (0 . -10) (0 . 0))
                ((0 . 0) (0 . -10) (0 . -10) (0 . 0))) _~
   e'-\shape #'(((0 . 0) (0 . -5) (0 . -5) (0 . 0))
                ((0 . 0) (0 . -5) (0 . -5) (0 . 0))) _~
   g'-\shape #'(((0 . 0) (0 . -2) (0 . -2) (0 . 0))
                ((0 . 0) (0 . -2) (0 . -2) (0 . 0))) _~
   c''-\shape #'(((0 . 0) (0 . 5) (0 . 5) (0 . 0))
                 ((0 . 0) (0 . 5) (0 . 5) (0 . 0))) ^~
  >2
  \break
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark \markup {
    It also works with the \typewriter tieWaitForNote property. }
  \set tieWaitForNote = ##t
  c'4-\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
  e'-\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
  g'-\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
  c''-\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
  \once \override TieColumn.positioning-done = ##t
  <c' e' g' c''>1
}

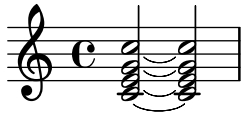
\layout {
  indent = 0
  \context {
    \Score
    \override TextMark.padding = #4
    \override TextMark.break-align-symbols = #'(left-edge)
  }
}
```

```
\paper {
  score-system-spacing.padding = 3
}
```

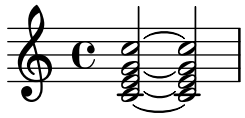
Chords can be tied note by note.



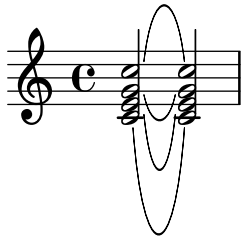
Modifying those ties with `\shape` does not succeed, because `TieColumn` positions them on its own behalf, ignoring `\shape` input more or less. You may circumvent this by setting `positioning-done` to `#t` – alas, `positioning-done` is an internal property, and setting it to `#t` means: all positioning is done, don't do anything further. The next example demonstrates a case where the positioning is not finished: all tie directions are down, and the thickness is not accurate.



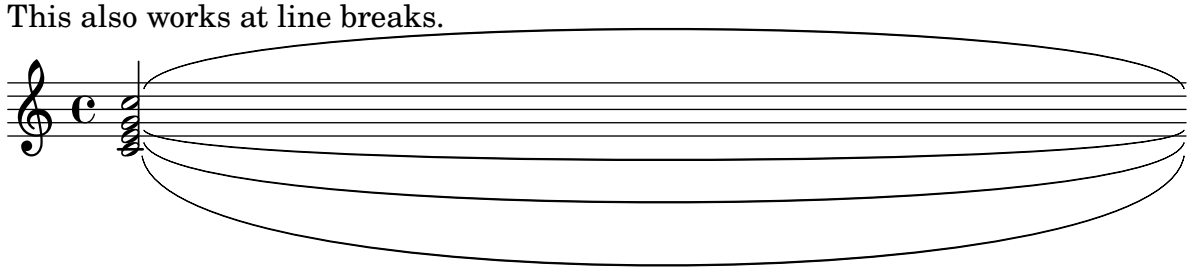
To fix that, enter ties with explicit direction modifiers.

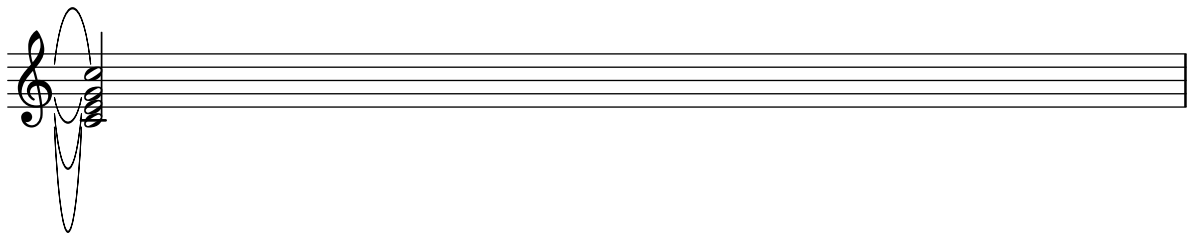


Now you can use `\shape` for each tie as usual.

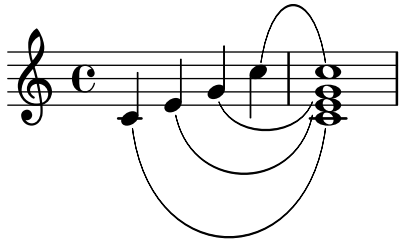


This also works at line breaks.
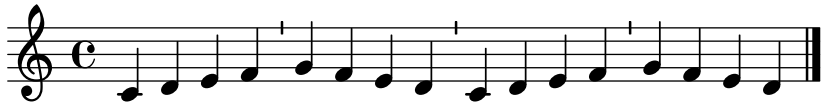




It also works with the `tieWaitForNote` property.



Barras de compás en forma de raya corta

Con frecuencia se usan barras de compás en forma de 'Tick' o raya corta donde la línea divisoria se emplea solamente a efectos de coordinación y no se quiere implicar con ella ningún énfasis rítmico.
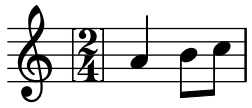
```
\relative c' {
  \set Score.measureBarType = #"'"
  c4 d e f
  g4 f e d
  c4 d e f
  g4 f e d
  \bar "|."
}
```



Indicación de compás entre corchetes

La indicación de compás se puede encerrar entre corchetes.
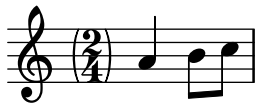
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Compás entre paréntesis

Se puede encerrar la indicación de compás entre paréntesis.

```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Trucaje de las propiedades de clave

La modificación del glifo de la clave, su posición o su octavación, no cambian 'per se' la posición de las siguientes notas del pentagrama. Para conseguir armaduras de tonalidad sobre las líneas del pentagrama adecuadas, también debe especificarse `middleCClefPosition`, con valores positivos o negativos que mueven el "Do central" hacia arriba o hacia abajo, respectivamente, en relación con la línea central del pentagrama (usualmente la tercera).

Por ejemplo, la instrucción `\clef "treble_8"` equivale a un ajuste de las propiedades de contexto `clefGlyph`, `clefPosition` (que controla la posición vertical de la clave sobre el pentagrama), `middleCPosition` y `clefTransposition`. Se imprime una clave cada vez que se modifica cualquiera de las propiedades excepto `middleCPosition`.

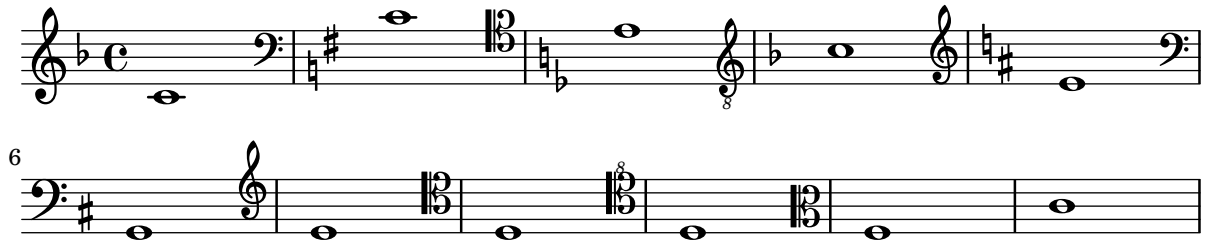
Los siguientes ejemplos muestran las posibilidades cuando se ajustan estas propiedades manualmente. En la primera línea, los cambios manuales preservan el posicionamiento relativo estándar de las claves y las notas, pero no lo hacen en la segunda línea.

```
{
  % The default treble clef.
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  \set Staff.middleCPosition = 6
  \set Staff.middleCClefPosition = 6
  \key g \major
  c'1
  % The baritone clef.
  \set Staff.clefGlyph = "clefs.C"
  \set Staff.clefPosition = 4
  \set Staff.middleCPosition = 4
  \set Staff.middleCClefPosition = 4
  \key f \major
  c'1
  % The standard choral tenor clef.
  \set Staff.clefGlyph = "clefs.G"
  \set Staff.clefPosition = -2
  \set Staff.clefTransposition = -7
  \set Staff.middleCPosition = 1
  \set Staff.middleCClefPosition = 1
  \key f \major
  c'1
  % A non-standard clef.
  \set Staff.clefPosition = 0
  \set Staff.clefTransposition = 0
  \set Staff.middleCPosition = -4
  \set Staff.middleCClefPosition = -4
  \key g \major
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
  % and clefs.
  \set Staff.clefGlyph = "clefs.F"
```

```
  \set Staff.clefPosition = 2
  c'1
  \set Staff.clefGlyph = "clefs.G"
  c'1
  \set Staff.clefGlyph = "clefs.C"
  c'1
  \set Staff.clefTransposition = 7
  c'1
  \set Staff.clefTransposition = 0
  \set Staff.clefPosition = 0
  c'1

  % Return to the normal clef.
  \set Staff.middleCPosition = 0
  c'1
}
```



Dos pares de \partCombine en un solo pentagrama

La función \partCombine toma dos expresiones musicales, cada una de las cuales contiene una parte o voz, y las distribuye entre cuatro contextos de Voice llamados “one”, “two”, “solo” y “shared” dependiendo de cuándo y cómo las partes o voces se combinan en una voz común.

Las voces que devuelve \partCombine pueden ver ajustadas sus propiedades de presentación de la manera usual. Aquí definimos extensiones de \partCombine para facilitar la inclusión de cuatro voces dentro del mismo pentagrama.

\partCombineUp y \partCombineDown son variantes de \partCombine pensadas para producir combinaciones de dos voces con la plica hacia arriba y hacia abajo, respectivamente. Sin embargo, combinarlas para comprimir cuatro partes en un solo pentagrama requiere ciertos ajustes especiales, los que este fragmento de código define adecuadamente.

```
customPartCombineUp =
#(define-music-function (part1 part2) (ly:music? ly:music?)
  "Make an up-stem `VoiceBox` context that combines PART1 and PART2.

The context is called 'Up'; internally, the function calls
`\\partCombineUp`."
  #{
    \new VoiceBox = "Up" <<
      \context Voice = "one" { \voiceOne }
      \context Voice = "two" { \voiceThree }
      \context Voice = "shared" { \voiceOne }
      \context Voice = "solo" { \voiceOne }
      \context NullVoice = "null" {}
      \partCombine #part1 #part2
    >>
```

```
  #})

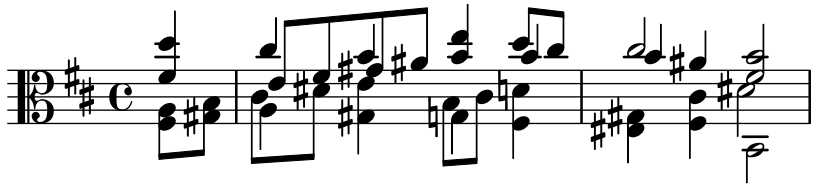
customPartCombineDown =
#(define-music-function (part3 part4) (ly:music? ly:music?)
  "Make a down-stem `VoiceBox` context that combines PART3 and PART4.

The context is called 'Down'; internally, the function calls
`\\partCombineDown`."
  #{
    \new VoiceBox = "Down" <<
      \set VoiceBox.soloText = #"Solo III"
      \set VoiceBox.soloIIText = #"Solo IV"
      \context Voice ="one" { \voiceFour }
      \context Voice ="two" { \voiceTwo }
      \context Voice ="shared" { \voiceFour }
      \context Voice ="solo" { \voiceFour }
      \context NullVoice = "null" {}
      \partCombine #part3 #part4
    >>
  #})

soprano = { d'4 | cis'  b  e'  d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }

\new Staff <<
  \key b\minor
  \clef alto
  \partial 4
  \transpose b b' \customPartCombineUp \soprano \alto
  \customPartCombineDown \tenor \bass
>>

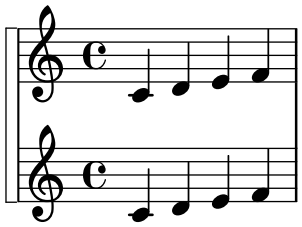
\layout {
  \context {
    \Staff
    \accepts "VoiceBox"
  }
  \context {
    \name "VoiceBox"
    \type "Engraver_group"
    \defaultchild "Voice"
    \accepts "Voice"
    \accepts "NullVoice"
  }
}
```



Uso del corchete recto al comienzo de un grupo de pentagramas

Se puede usar el delimitador de comienzo de un sistema `SystemStartSquare` estableciéndolo explícitamente dentro de un contexto `StaffGroup` o `ChoirStaffGroup`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



Using \autoChange with more than one voice

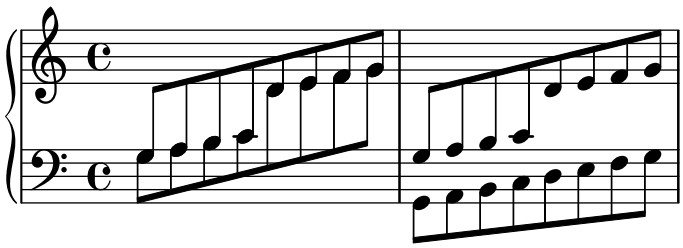
Here is a demonstration of how to use `\autoChange` with more than one voice.

```
\score {
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,8 a b c d e f g
          }
        }

        \new Voice {
          \voiceTwo
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,,8 a b c d e f g
          }
```

```
        }
      >>
    }

    \new Staff = "down" {
      \clef bass
    }
  >>
}
```



Using mark lines in a Frenched score

Using `MarkLine` contexts (such as in "Placing rehearsal marks other than above the top staff") in a Frenched score can be problematic if all the staves between two `MarkLines` are removed in one system. The `Keep_alive_together_engraver` can be used within each `StaffGroup` to keep the `MarkLine` alive only as long as the other staves in the group stay alive.

```
bars = {
  \tempo Allegro 4=120
  s1*2
  \*5 { \mark \default s1*2 }
  \bar "||"
  \tempo Adagio 4=40
  s1*2
  \*8 { \mark \default s1*2 }
  \bar "|."
}
winds = \*120 c''4
trumpet = {
  \*8 g'2
  R1*16
  \*4 g'2
  R1*8
}
trombone = {
  \*4 c'1
  R1*8
  d'1
  R1*17
}
strings = \*240 c''8
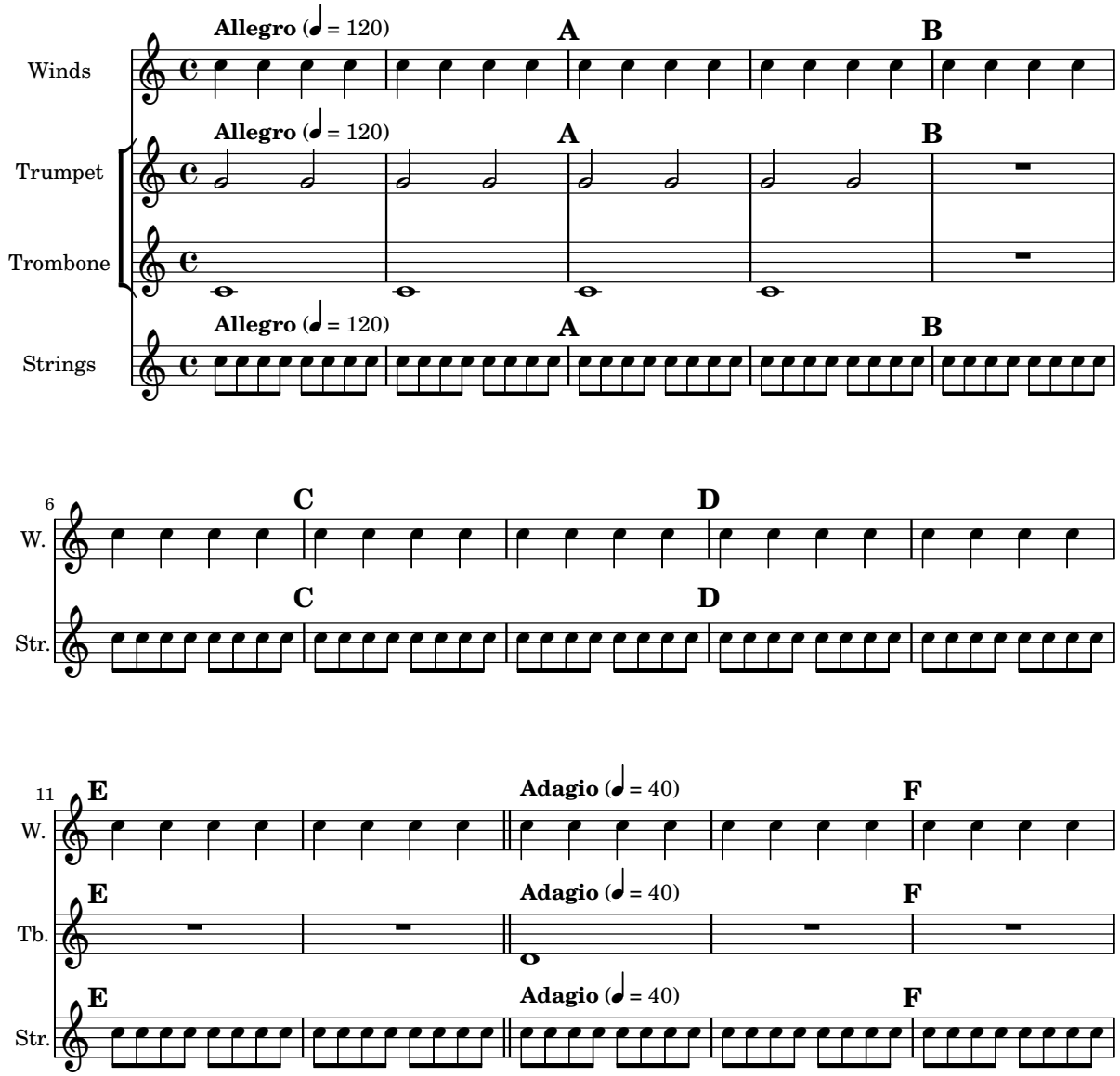
#(set-global-staff-size 16)
\paper {
  systems-per-page = 5
```
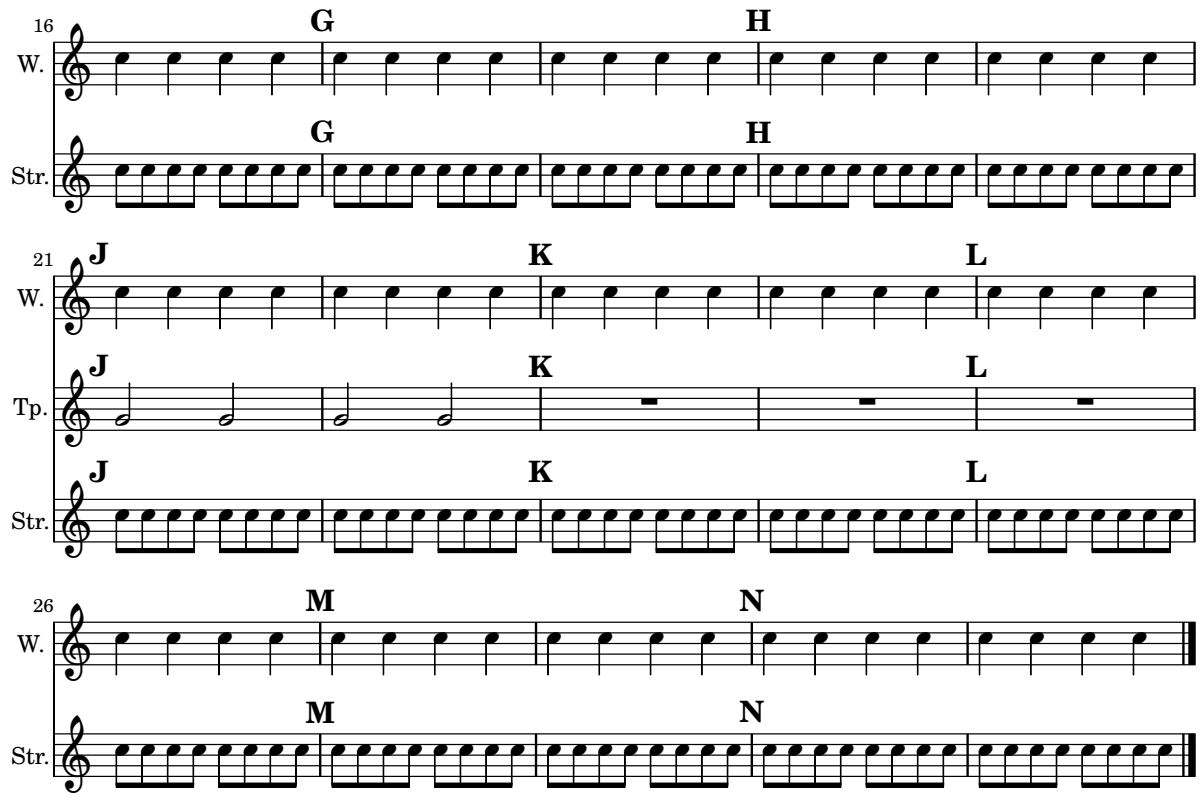
```
  ragged-last-bottom = ##f
  tagline = ##f
}

\layout {
  indent = 16\mm
  short-indent = 5\mm
  \context {
    \name MarkLine
    \type Engraver_group
    \consists Apply_output_engraver
    \consists Axis_group_engraver
    \consists Mark_engraver
    \consists Metronome_mark_engraver
    \consists Staff_collecting_engraver
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = #'any
    \override VerticalAxisGroup.staff-affinity = #DOWN
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing.padding = 1
    keepAliveInterfaces = #'()
  }
  \context {
    \Staff
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = ##f
  }
  \context {
    \StaffGroup
    \accepts MarkLine
    \consists Keep_alive_together_engraver
  }
  \context {
    \Score
    \remove Mark_engraver
    \remove Metronome_mark_engraver
    \remove Staff_collecting_engraver
    \override BarNumber.Y-offset = #3
  }
}

\score {
  <<
    \new StaffGroup = "winds" \with {
      instrumentName = "Winds"
      shortInstrumentName = "W."
    } <<
      \new MarkLine \bars
      \new Staff \winds
    >>
    \new StaffGroup = "brass" <<
      \new MarkLine \bars
      \new Staff = "trumpet" \with {
```

```
        instrumentName = "Trumpet"
        shortInstrumentName = "Tp."
      } \trumpet
      \new Staff = "trombone" \with {
        instrumentName = "Trombone"
        shortInstrumentName = "Tb."
      } \trombone
    >>
    \new StaffGroup = "strings" \with {
      instrumentName = "Strings"
      shortInstrumentName = "Str."
    } <<
      \new MarkLine \bars
      \new Staff = "strings" { \strings }
    >>
  >>
}
```





Grupos de pentagramas alineados verticalmente sin barra inicial de conexión

Este fragmento muestra cómo conseguir StaffGroups alineados verticalmente con un SystemStartBar por cada StaffGroup, pero sin conectarlos entre sí.

Observe que esto solo funciona correctamente para música que quepa completamente dentro de un único sistema.

```
#(set-global-staff-size 15)

\paper {
  ragged-right = ##f
  print-all-headers = ##t
  tagline = ##f
}

\layout {
  indent = 0

  \context {
    \StaffGroup
    \consists Text_mark_engraver
    \consists Staff_collecting_engraver
    systemStartDelimiterHierarchy =
      #'(SystemStartBrace (SystemStartBracket a b))
  }

  \context {
    \Score
```

```
    \remove Text_mark_engraver
    \remove Staff_collecting_engraver
    \override SystemStartBrace.style = #'bar-line
    \omit SystemStartBar
    \override SystemStartBrace.padding = #-0.1
    \override SystemStartBrace.thickness = #1.6
    \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #15
  }
}

%%%% EXAMPLE

txt =
\lyricmode {
  Wer4 nur den lie -- ben Gott läßt wal2 -- ten4
  und4 hof -- fet auf ihn al -- le Zeit2.
}

% First StaffGroup "exercise"

eI = \relative c' {
  \textMark \markup {
    \bold Teacher:
    This is a simple setting of the choral. Please improve it. }
  \key a \minor
  \time 4/4
  \voiceOne

  \partial 4 e4
  a b c b
  a b gis2
  e4\fermata g! g f
  e a a gis
  a2.\fermata
  \bar ":|."
}

eII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo
  \partial 4 c4
  e e e gis
  a f e2
  b4 b d d
  c c d d
  c2.
  \bar ":|."
}

eIII = \relative c' {
  \key a \minor
```

```
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a4
  c b a b
  c d b2
  gis4 g g b
  c a f e
  e2.
}

eIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a,4
  a' gis a e
  a, d e2
  e,4\fermata e' b g
  c f d e
  a,2.\fermata
  \bar ":|."
}

exercise = \new StaffGroup = "exercise" <<
  \new Staff <<
    \new Voice \eI
    \new Voice \eII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \eIII
    \new Voice \eIV
  >>
>>


% Second StaffGroup "simple Bach"

sbI = \relative c' {
  \textMark \markup { \bold" Pupil:" Here's my version! }
  \key a \minor
  \time 4/4
  \voiceOne

  \partial 4 e4
  a b c b
```

```
  a b gis2
  e4\fermata g! g f
  e a a gis
  a2.\fermata
  \bar ":|."
}

sbII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo
  \partial 4 c8 d
  e4 e e8 f g4
  f f e2
  b4 b8 c d4 d
  e8 d c4 b8 c d4
  c2.
  \bar ":|."
}

sbIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a8 b
  c4 b a b8 c
  d4 d8 c b2
  gis4 g g8 a b4
  b a8 g f4 e
  e2.
}

sbIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a,4
  a' gis a e
  f8 e d4 e2
  e,4\fermata e' b a8 g
  c4 f8 e d4 e
  a,2.\fermata
  \bar ":|."
}

simpleBach = \new StaffGroup = "simple Bach" <<
  \new Staff <<
    \new Voice \sbI
```

```
    \new Voice \sbII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \sbIII
    \new Voice \sbIV
  >>
>>


% Third StaffGroup "chromatic Bach"

cbI = \relative c' {
  \textMark \markup {
    \bold "Teacher:"
    \column {
      "Well, you simply copied and transposed a version of J.S.Bach."
      "Do you know this one?"
    }
  }
  \key a \minor
  \time 4/4
  \voiceOne

  \partial 4 e4
  a b c b
  a b gis4. fis8
  e4\fermata g! g f
  e a a8 b gis4
  a2.\fermata
  \bar ":|."
}

cbII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo

  \partial 4 c8 d
  e4 e e8 fis gis4
  a8 g! f!4 e2
  b4 e e d
  d8[ cis] d dis e fis e4
  e2.
  \bar ":|."
}

cbIII = \relative c' {
  \key a \minor
  \time 4/4
```

```
  \clef bass
  \voiceOne

  \partial 4 a8 b
  c[ b] a gis8 a4 d,
  e8[ e'] d c b4. a8
  gis4 b c d8 c
  b[ a] a b c b b c16 d
  c2.
}

cbIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a4
  c, e a, b
  c d e2
  e4\fermata e a b8 c
  gis[ g] fis f e dis e4
  a,2.\fermata
  \bar ":|."
}

chromaticBach = \new StaffGroup = "chromatic Bach" <<
  \new Staff <<
    \new Voice \cbI
    \new Voice \cbII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \cbIII
    \new Voice \cbIV
  >>
>>
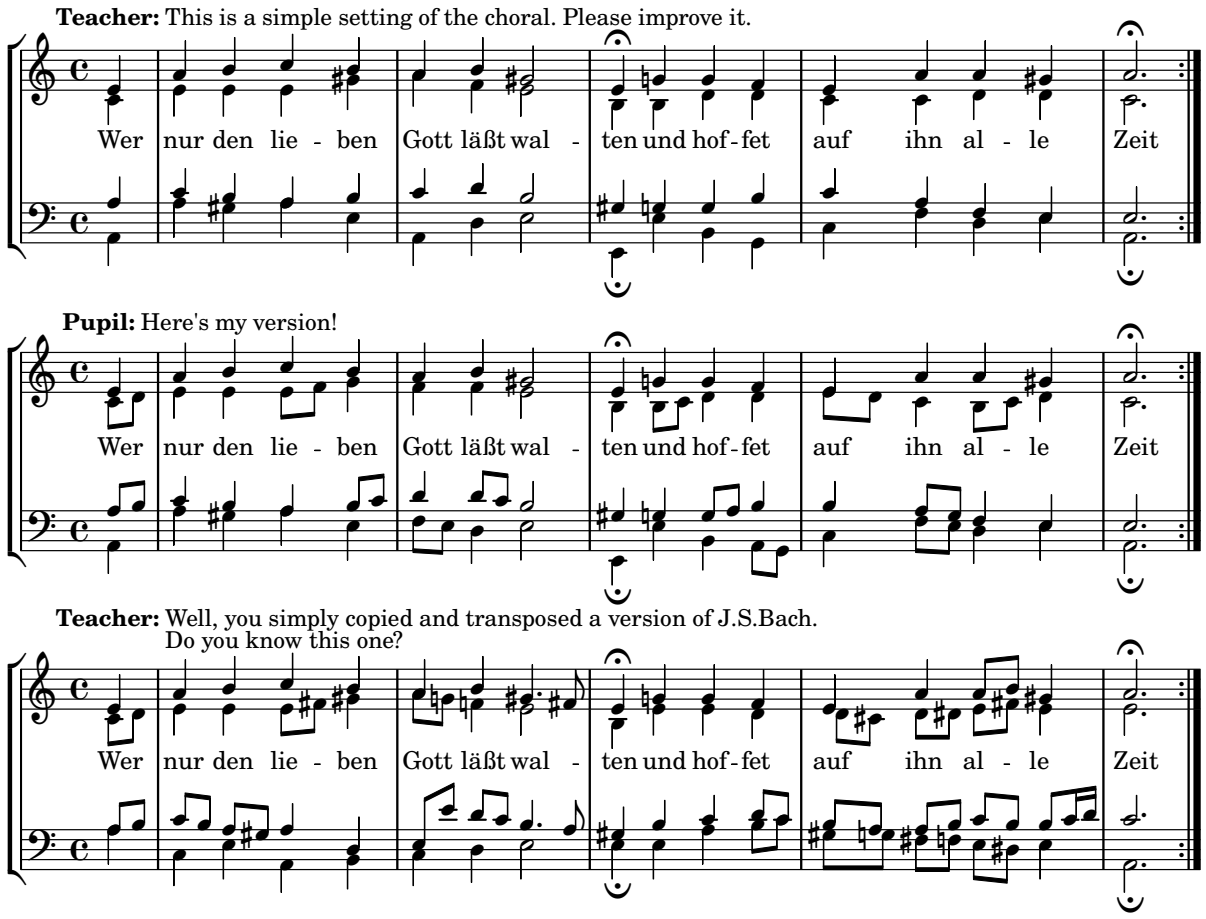

% Score

\score {
  <<
    \exercise
    \simpleBach
    \chromaticBach
  >>

  \header {
    title = \markup \column {
```

```
                \combine \null \vspace #1
                "Exercise: Improve the given choral"
                " "
              }
  }

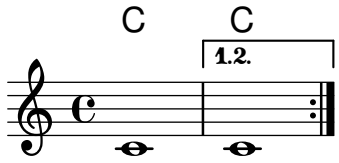
  \layout {
    \context {
      \Lyrics
      \override LyricText.X-offset = #-1
    }
  }
}
```



Corchetes de primera y segunda vez debajo de los acordes

Mediante la adición del grabador `Volta_engraver` al pentagrama pertinente, se pueden poner los corchetes de primera y segunda vez debajo de los acordes.

```
\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



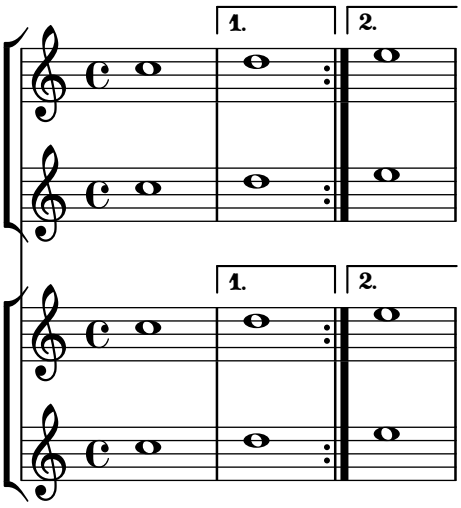
Casillas de primera y segunda vez en varios pentagramas

Añadiendo el grabador `Volta_engraver` al pentagrama deseado, las casillas de primera y segunda vez se pueden poner sobre pautas distintas a la del extremo superior de la partitura.

Las instrucciones `\repeat` y otras relacionadas deben estar presentes en todos los pentagramas.

```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}

<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```


1.
2.
1.
2.
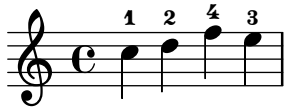
7 Editorial annotations

See also Sección “Editorial annotations” en *Notation Reference*.

Añadir digitaciones a la partitura

Se pueden escribir instrucciones de digitación usando una sintaxis muy sencilla.

```
\relative c'' {
  c4-1 d-2 f-4 e-3
}
```



Añadir enlaces a los objetos

Para añadir un enlace al sello de un objeto gráfico, podemos usar `add-link` tal y como se define aquí. Funciona con `\override` y con `\tweak`.

Inconveniente: `point-and-click` (apuntar y pulsar) quedará obstacuilzado por los objetos gráficos enlazados.

Limitación: funciona solamente para PDF.

Los objetos enlazados se colorean con una instrucción aparte.

```
#(define (add-link url-strg)
  (lambda (grob)
    (let* ((stil (ly:grob-property grob 'stencil)))
      (if (ly:stencil? stil)
          (let* ((x-ext (ly:stencil-extent stil X))
                 (y-ext (ly:stencil-extent stil Y))
                 (url-expr `(url-link ,url-strg ,x-ext ,y-ext))
                 (new-stil
                   (ly:stencil-add
                     (ly:make-stencil url-expr x-ext y-ext)
                     stil)))
          (ly:grob-set-property! grob 'stencil new-stil))))))

%%%% test

%% For easier maintenance of this snippet the URL is formatted to use the
%% actually used LilyPond version.
%% Of course a literal URL would work as well.

#(define major.minor-version
  (string-join (take (string-split (lilypond-version) #\.) 2) "."))

urlI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

urlII =
#(format #f
```

```
  "http://lilypond.org/doc/v~a/Documentation/notation/rhythms"
  major.minor-version)

urlIII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-heads"
  major.minor-version)

urlIV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/beams"
  major.minor-version)

urlV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-head-styles"
  major.minor-version)

urlVI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

\relative c' {
   \key cis \minor
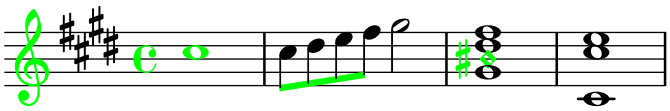
   \once \override Staff.Clef.color = #green
   \once \override Staff.Clef.after-line-breaking =
     #(add-link urlI)

   \once \override Staff.TimeSignature.color = #green
   \once \override Staff.TimeSignature.after-line-breaking =
     #(add-link urlII)

   \once \override NoteHead.color = #green
   \once \override NoteHead.after-line-breaking =
     #(add-link urlIII)

   cis'1
   \once \override Beam.color = #green
   \once \override Beam.after-line-breaking =
     #(add-link urlIV)
   cis8 dis e fis gis2
   <gis,
    \tweak Accidental.color #green
    \tweak Accidental.after-line-breaking #(add-link urlVI)
    \tweak color #green
    \tweak after-line-breaking #(add-link urlV)
    \tweak style #'harmonic
    bis
    dis
    fis
```

```
   >1
   <cis, cis' e>
}
```



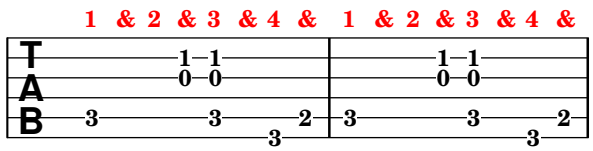
Insertar elementos de marcado en una tablatura

De forma predeterminada, los elementos de marcado no aparecen en la tablatura.

Para hacer que aparezcan, revierta la propiedad `stencil` del objeto gráfico `TextScript` dentro del contexto `TabStaff`.

```
high  = { r4 r8 <g c'> q r8 r4 }
low = { c4 r4 c8 r8 g,8 b, }
pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }
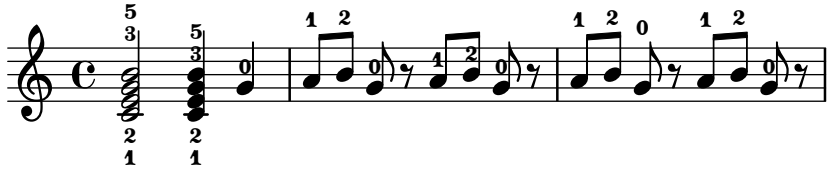
\score {
 \new TabStaff {
   \*2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Hay que prestar atención en situaciones en las que las digitaciones y las plicas están en la misma dirección: por defecto, las digitaciones solo evitan la colisión con plicas unidas por una barra. Se puede cambiar este ajuste para no evitar ninguna plica o evitarlas todas; el ejemplo siguiente muestra las dos copciones, así como la manera de volver al comportamiento predeteminado.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```



Numeración de compases alternativa

Fijando la propiedad de contexto `alternativeNumberingStyle`, se dispone de dos métodos alternativos adicionales para la numeración de compases dentro de las repeticiones.

```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|."
}
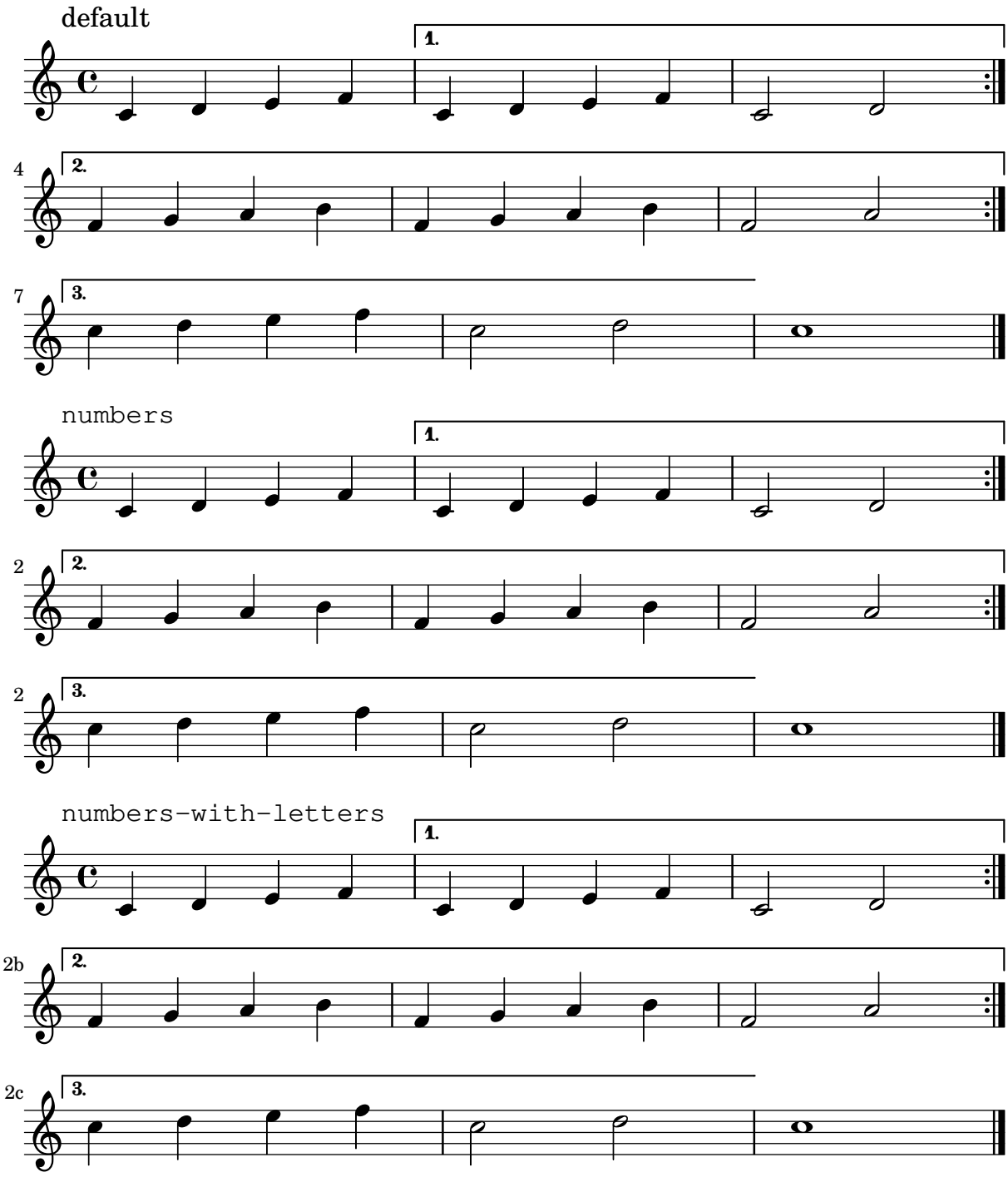
{
  \textMark \markup \large "default"
  \music
}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
  \music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
```

```
    \override TextMark.Y-offset = #5
  }
}
```



Corchetes de análisis encima del pentagrama

De forma predeterminada se añaden corchetes de análisis sencillos debajo del pentagrama. El ejemplo siguiente muestra una manera de colocarlos por encima.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```



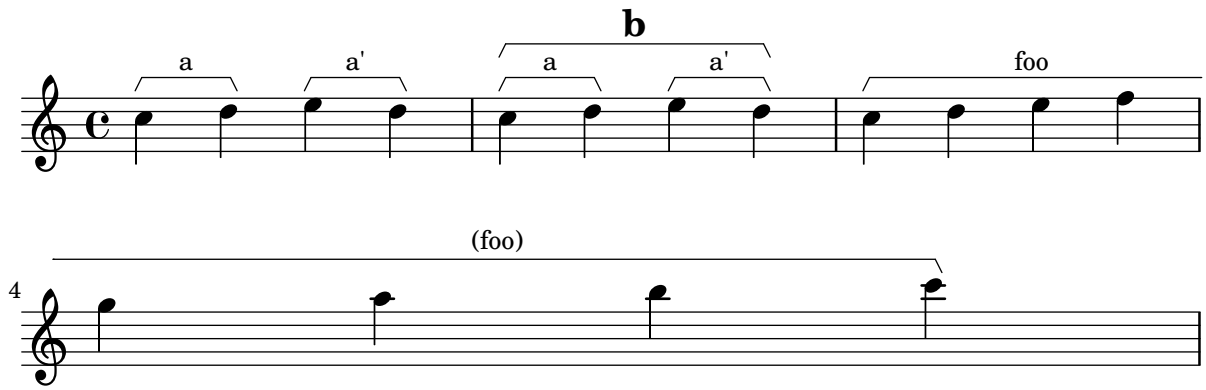
Corchetes de análisis con etiquetas

Se pueden añadir indicaciones de marcado de texto a los corchetes de análisis por medio de la propiedad `text` del objeto gráfico `HorizontalBracketText`. Si se quieren añadir textos diferentes al corchetes que comienzan en el mismo tiempo, es necesario usar la instrucción `\tweak`.

El texto del corchete se encierra entre paréntesis después de un salto de línea. El orden vertical de los corchetes anidados se puede controlar con la propiedad `outside-staff-priority`.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
    c''\startGroup d''\stopGroup
    \once\override HorizontalBracketText.text = "a'"
    e''\startGroup d''\stopGroup |
  c''-\tweak outside-staff-priority #801
      \tweak HorizontalBracketText.text
        \markup \bold \huge "b" \startGroup
     -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```



Aplicar estilos de cabeza según la nota de la escala

La propiedad `shapeNoteStyles` se puede usar para definir varios estilos de cabezas de nota para cada grado de la escala (según esté establecido por la armadura o por la propiedad `tonic`).

Esta propiedad requiere un conjunto de símbolos, que pueden ser puramente arbitrarios (se permiten expresiones geométricas como `triangle`, triángulo, `cross`, aspas, y `xcircle`, círculo con aspas) o basados en una antigua tradición americana de grabado (ciertos nombres de nota latinos trambién se permiten).

Dicho esto, para imitar antiguos cancioneros americanos, existen varios estilos predefinidos de cabezas de nota disponibles a través de instrucciones de abreviatura como `\aikenHeads` o `\sacredHarpHeads`.

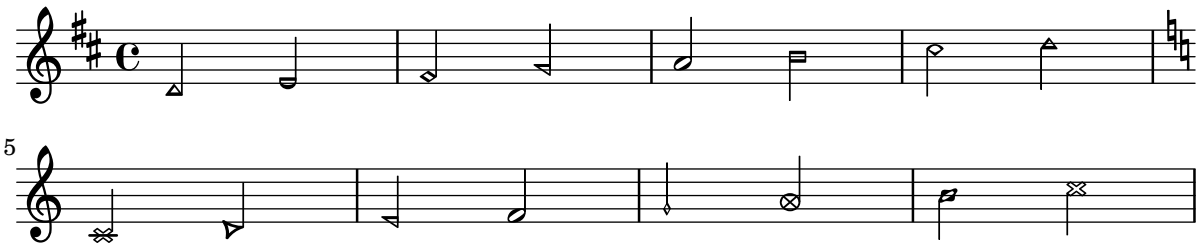
Este ejemplo muestra distintas formas de obtener cabezas de notas con forma, y muestra la capacidad de transportar una melodía sin perder la correspondencia entre las funciones armónicas y los estilos de cabezas de nota.

```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                              #f la ti)
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                              mensural xcircle diamond)
    \fragment
  }
}
```

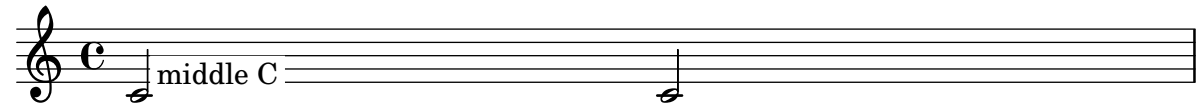


Poner en blanco las líneas del pentagrama utilizando la instrucción `\whiteout`

La instrucción `\whiteout` intercala un rectángulo blanco debajo de un elemento de marcado. Este rectángulo blanco no tapa a ningún otro objeto gráfico, puesto que las líneas del pentagrama están en una capa inferior a la de la mayor parte de los otros objetos.

```
\layout {
  ragged-right = ##f
}

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```

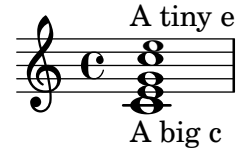


Changing a single note's size in a chord

Individual note heads in a chord can be modified with the `\tweak` command inside a chord, by altering the `font-size` property.

Inside the chord (within the brackets < >), before the note to be altered, place the `\tweak` command, followed by `font-size` and define the proper size like `#-2` (a tiny note head).

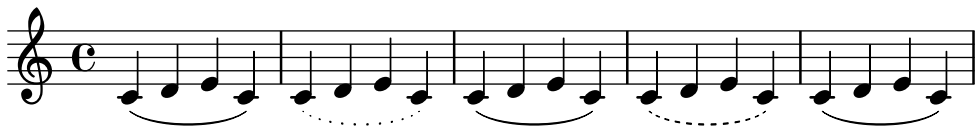
```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Modificar el aspecto continuo de una ligadura de expresión a punteado o intermitente

Se puede cambiar el aspecto de las ligaduras de expresión de continuas a punteadas o intermitentes.

```
\relative c' {
  c4( d e c)
  \slurDotted
  c4( d e c)
  \slurSolid
  c4( d e c)
  \slurDashed
  c4( d e c)
  \slurSolid
  c4( d e c)
}
```



Colorear las notas según su altura

Es posible colorear la cabeza de las notas dependiendo de su altura y/o de su nombre: la función que se usa en este ejemplo hace posible incluso distinguir los armónicos.

```
% Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 2 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 3 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 3 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 4 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 5 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 5 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 6 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 1 NATURAL) (x11-color 'blue))
    (cons (ly:make-pitch 0 3 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 4 FLAT) (x11-color 'blue))
    (cons (ly:make-pitch 0 5 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 6 FLAT) (x11-color 'blue))))
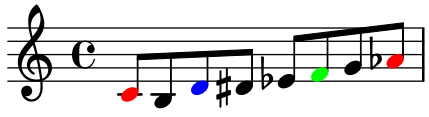
% Compare pitch and alteration (not octave).
#(define (pitch-equals? p1 p2)
   (and
    (= (ly:pitch-alteration p1) (ly:pitch-alteration p2))
    (= (ly:pitch-notename p1) (ly:pitch-notename p2))))

#(define (pitch-to-color pitch)
   (let ((color (assoc pitch color-mapping pitch-equals?)))
     (if color
         (cdr color))))

#(define (color-notehead grob)
```

```
   (pitch-to-color
    (ly:event-property (event-cause grob) 'pitch)))

\score {
  \new Staff \relative c' {
    \override NoteHead.color = #color-notehead
    c8 b d dis ees f g aes
  }
}
```

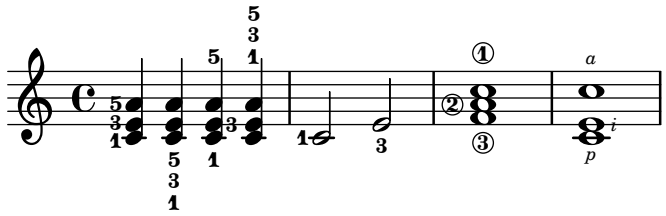


Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación usando la propiedad `fingeringOrientation`. Para que se tenga en cuenta la orientación de las digitaciones, la instrucción de digitación se debe utilizar dentro de una construcción de acorde (`<...>`) aunque sea una sola nota. Se puede controlar la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar usando las propiedades `stringNumberOrientation` y `strokeFingerOrientation`, respectivamente.

Estas propiedades se pueden establecer a una lista de entre uno y tres valores. Controlan si las digitaciones se pueden situar por encima (si aparece `up` en la lista), por debajo (si aparece `down`), a la izquierda (si aparece `left`), o a la derecha (si aparece `right`). Por contra, si una localización no aparece en la lista, no se coloca ninguna digitadión en ese lugar. LilyPond admite esas restricciones y trata de encontrar la mejor ubicación para las digitaciones de las notas de los acordes siguientes. Observe que `left` y `right` se excluyen mutuamente (las digitaciones se pueden colocar en un lado o en el otro, no en los dos).

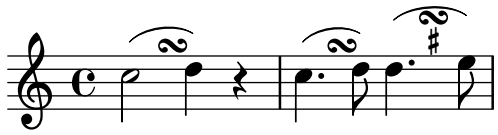
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Crear un grupeto de anticipación

La creación de un grupeto circular de anticipación entre dos notas, donde la nota inferior del grupeto utiliza una alteración, requiere varias sobreescrituras de propiedades. La propiedad `outside-staff-priority` se debe establecer al valor `#f`, pues en caso contrario tendría prioridad sobre la propiedad `avoid-slur`. Cambiando el primer argumento de `\after` (que es una duración) puede ajustarse la posición horizontal.

```
\relative c'' {
  \after 2*2/3 \turn c2( d4) r |
  \after 4 \turn c4.( d8)
  \after 4
  {
    \once \set suggestAccidentals = ##t
    \once \override AccidentalSuggestion.outside-staff-priority = ##f
    \once \override AccidentalSuggestion.avoid-slur = #'inside
    \once \override AccidentalSuggestion.font-size = -3
    \once \override AccidentalSuggestion.script-priority = -1
    \once \hideNotes
    cis8\turn \noBeam
  }
  d4.( e8)
}
```



Crear pentagramas en blanco

Para crear pentagramas en blanco, genere compases vacíos y después elimine el grabador de números de compás `Bar_number_engraver` del contexto `Score`, y los grabadores de la indicación de compás `Time_signature_engraver`, de la clave `Clef_engraver` y de los compases `Bar_engraver` del contexto de `Staff`.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

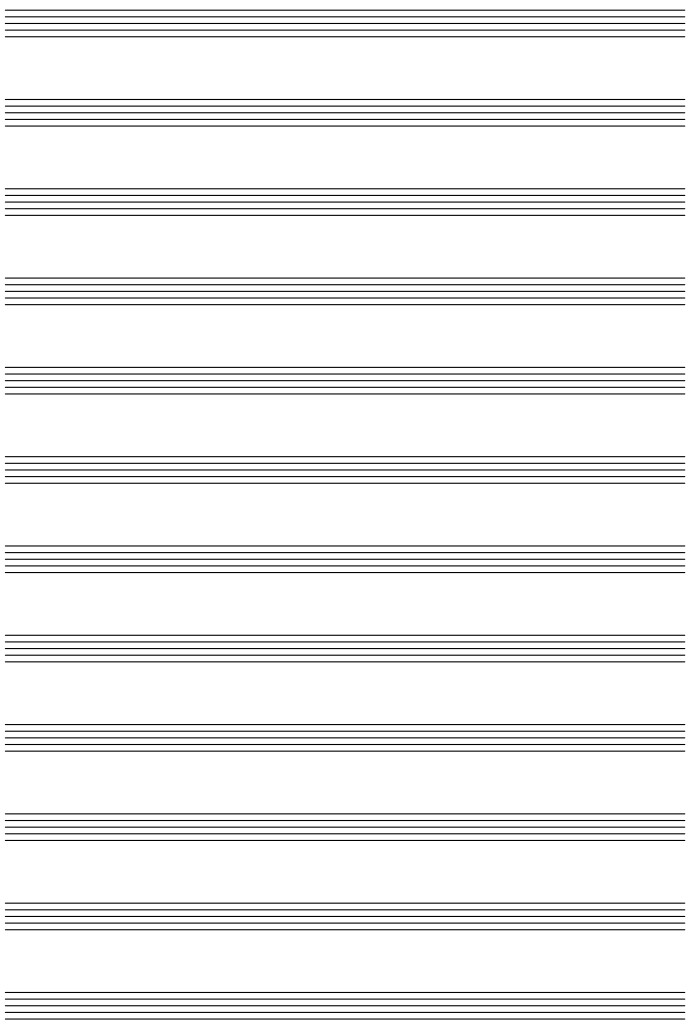
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
```

```
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % for the documentation
  \paper {
    #(set-paper-size "a6")
    ragged-last-bottom = ##f
    line-width = 90\mm
    left-margin = 7.5\mm
    bottom-margin = 5\mm
    top-margin = 5\mm
    tagline = ##f
  }

  % uncomment these lines for "letter" size
  %{
  \paper {
    #(set-paper-size "letter")
    ragged-last-bottom = ##f
    line-width = 7.5\in
    left-margin = 0.5\in
    bottom-margin = 0.25\in
    top-margin = 0.25\in
    tagline = ##f
   }
  %}

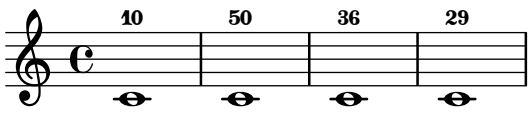
  % uncomment these lines for "A4" size
  %{
  \paper {
    #(set-paper-size "a4")
    ragged-last-bottom = ##f
    line-width = 180\mm
    left-margin = 15\mm
    bottom-margin = 10\mm
    top-margin = 10\mm
    tagline = ##f
  }
  %}
}
```



Crear digitaciones de dos cifras

Es posible crear digitaciones con un número mayor de 5.
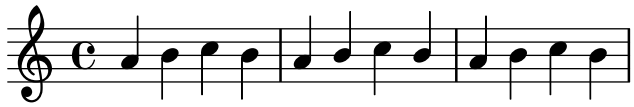
```
\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}
```



Dirección predeterminada de las plicas sobre la tercera línea del pentagrama

La dirección predeterminada de las plicas sobre la tercera línea del pentagrama está determinada por la propiedad `neutral-direction` del objeto `Stem`.

```
\relative c'' {
  a4 b c b
  \override Stem.neutral-direction = #up
  a4 b c b
  \override Stem.neutral-direction = #down
  a4 b c b
}
```



Diferentes tamaños de fuente para instrumentName y shortInstrumentName

Elegir distintos tamaños de fuente para instrumentName y shortInstrumentName como una sobreescritura de contexto.

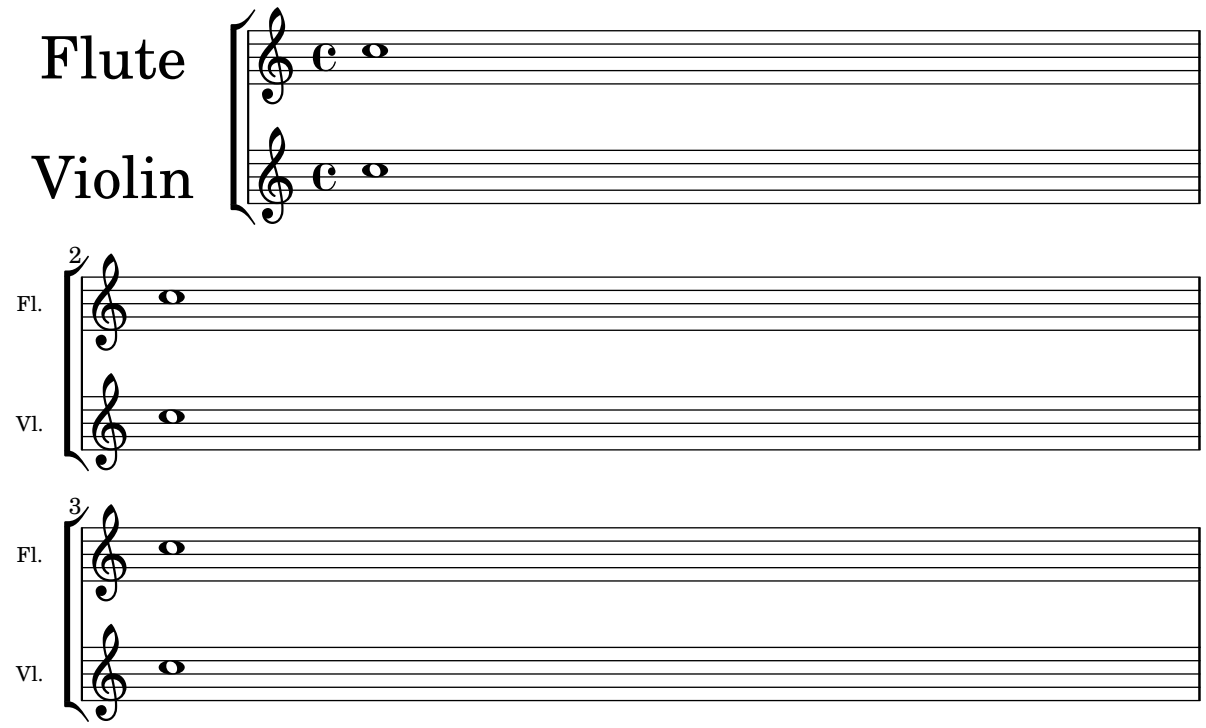
```
InstrumentNameFontSize =
#(define-music-function (font-size-pair) (pair?)
   "Set the font size of `InstrumentName` grobs.

The first value of FONT-SIZE-PAIR sets the font size of the initial
`instrumentName` property, the second value sets the font size of
`shortInstrumentName`."
   ;; This code could be changed or extended to set different values
   ;; for each occurrence of `shortInstrumentName'.
   #{
     \override InstrumentName.after-line-breaking =
       #(lambda (grob)
          (let* ((orig (ly:grob-original grob))
                 (siblings (if (ly:grob? orig)
                               (ly:spanner-broken-into orig)
                               '())))
            (when (pair? siblings)
              (ly:grob-set-property! (car siblings)
                                     'font-size (car font-size-pair))
              (for-each
               (lambda (g)
                 (ly:grob-set-property! g
                                        'font-size (cdr font-size-pair)))
               (cdr siblings)))))
   #})

\layout {
  indent = 3\cm
  short-indent = 0.8\cm

  \context {
    \Staff
    \InstrumentNameFontSize #'(6 . -3)
  }
}
```

```
\new StaffGroup <<
  \new Staff \with {
       instrumentName = "Flute"
       shortInstrumentName = "Fl." } {
    c''1 \break c'' \break c'' }
  \new Staff \with {
       instrumentName = "Violin"
       shortInstrumentName = "Vl." } {
    c''1 \break c'' \break c'' }
>>
```



Rodear los objetos gráficos con rectángulos

La propiedad `stencil` se puede sobreescribir para trazar un rectángulo que encierre objetos gráficos arbitrarios, bien sea usando `\override` o bien `\tweak`.

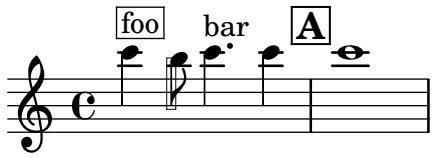
```
\relative c'' {
  \once \override TextScript.stencil =
    #(make-stencil-boxer 0.1 0.3 ly:text-interface::print)
  c'4^"foo"

  \tweak Stem.stencil
    #(make-stencil-boxer 0.05 0.25 ly:stem::print)
  b8

  c4.^"bar" c4

  \override Score.RehearsalMark.stencil =
    #(make-stencil-boxer 0.15 0.3 ly:text-interface::print)
  \mark \default
  c1
```
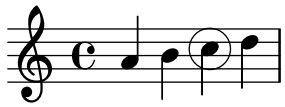
```
}
```



Trazar circunferencias alrededor de la cabeza de las notas

Se puede trazar una circunferencia alrededor de la cabeza de una nota aportando una función de Scheme personalizada para sobreescribir temporalmente la propiedad `stencil`.

```
circle = \tweak NoteHead.stencil
  #(lambda (grob)
    (let* ((note (ly:note-head::print grob))
           (combo-stencil (ly:stencil-add
               note
               (circle-stencil note 0.1 0.8))))
          (ly:make-stencil (ly:stencil-expr combo-stencil)
            (ly:stencil-extent note X)
            (ly:stencil-extent note Y))))
  \etc

{ a' b' \circle c'' d'' }
```

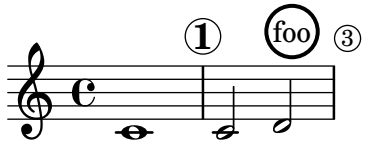


Rodear diversos objetos con una circunferencia

La instrucción `\circle` traza circunferencias alrededor de los objetos de marcado creados con `\markup`. Para otros objetos pueden requerirse trucos específicos, como se muestra aquí para el caso de las letras de ensayo y los números de compás.

```
\relative c' {
  c1
  \set Score.rehearsalMarkFormatter =
    #(lambda (mark context)
             (make-circle-markup (format-mark-numbers mark context)))
  \mark \default

  c2 d^\markup {
    \override #'(thickness . 3) {
      \circle foo
    }
  }
  \override Score.BarNumber.break-visibility = #all-visible
  \override Score.BarNumber.stencil =
    #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
}
```



Empotrar PostScript nativo dentro de un bloque \markup

Se puede insertar código PostScript directamente dentro de un bloque `\markup`.

En general se recomienda usar en su lugar las instrucciones de marcado nativas de LilyPond tales como `\polygon`, que se pueden usar con todos los *backends* de LilyPond.

```
\relative c'' {
  a2-\markup \postscript "0 3 moveto
                          5 2 rlineto
                          stroke"
    -\markup \postscript "[1 1] 0 setdash
                          0 0 moveto
                          5 2 rlineto
                          stroke"
  b2-\markup \postscript "1 1 moveto
                          0 0 1 2 8 4 10 2 rcurveto
                          stroke"
  a'1
}
```



Generate special note head shapes

When a note head with a special shape cannot easily be generated with graphic markup, a drawing specification for `ly:make-stencil` can be used to generate the shape. This snippet gives an example for a parallelogram-shaped note head.

Unfortunately, the available commands in a drawing specification are currently not documented (this is tracked in Issue #6874 (`https://gitlab.com/lilypond/lilypond/-/issues/6874`)); in any case, the used `path` sub-command has the following signature, quite similar to the `make-path-stencil` Scheme function.
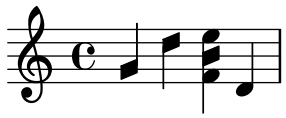
```
(path thickness command-list line-cap-style line-join-style fill)
```

The commands in *command-list* resemble PostScript drawing commands but with arguments after the command name.

```
parallelogram =
  #(ly:make-stencil
    '(path 0.1
           (rmoveto 0 0.25
            lineto 1.2 0.75
            lineto 1.2 -0.25
            lineto 0 -0.75
            lineto 0 0.25)
           round
           round
           #t)
    (cons -0.05 1.25)
    (cons -.75 .75))

myNoteHeads = \override NoteHead.stencil = \parallelogram
normalNoteHeads = \revert NoteHead.stencil
```

```
\relative c'' {
  \myNoteHeads
  g4 d'
  \normalNoteHeads
  <f, \tweak stencil \parallelogram b e>4 d
}
```



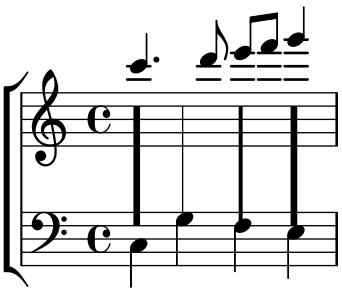
Líneas de rejilla: modificar su aspecto

Se puede cambiar el aspecto de las líneas de rejilla sobreescribiendo algunas de sus propiedades.

```
\new ChoirStaff <<
  \new Staff {
    \relative c'' {
      \stemUp
      c'4. d8 e8 f g4
    }
  }
  \new Staff {
    \relative c {
      % this moves them up one staff space from the default position
      \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
      \stemDown
      \clef bass
      \once \override Score.GridLine.thickness = 5.0
      c4
      \once \override Score.GridLine.thickness = 1.0
      g'4
      \once \override Score.GridLine.thickness = 3.0
      f4
      \once \override Score.GridLine.thickness = 5.0
      e4
    }
  }
>>

\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #1/4
    }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % this moves them to the right half a staff space
    \override NoteColumn.X-offset = -0.5
```

```
  }
}
```



Líneas de rejilla: destacar ritmos y la relación temporal entre notas

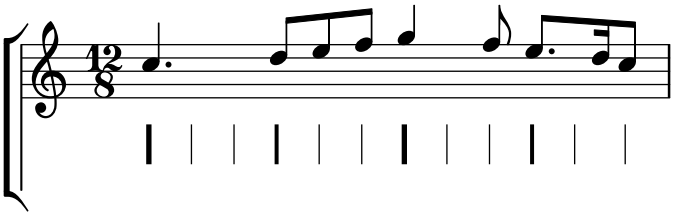
Se pueden trazar líneas verticales normales entre pentagramas para mostrar la relación entre notas; sin embargo, en caso de música monofónica, podemos hacer invisible el segundo pentagrama, y que las líneas sean más cortas, como en este fragmento de código.

```
\new ChoirStaff {
  \relative c'' <<
    \new Staff {
      \time 12/8
      \stemUp
      c4. d8 e8 f g4 f8 e8. d16 c8
    }
    \new Staff {
      % hides staff and notes so that only the grid lines are visible
      \hideNotes
      \hide Staff.BarLine
      \override Staff.StaffSymbol.line-count = #0
      \hide Staff.TimeSignature
      \hide Staff.Clef

      % dummy notes to force regular note spacing
      \once  \override Score.GridLine.thickness = #4.0
      c8 c c
      \once  \override Score.GridLine.thickness = #3.0
      c8 c c
      \once  \override Score.GridLine.thickness = #4.0
      c8 c c
      \once  \override Score.GridLine.thickness = #3.0
      c8 c c
    }
  >>
}

\layout {
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % center grid lines horizontally below note heads
    \override NoteColumn.X-offset = #-0.5
```

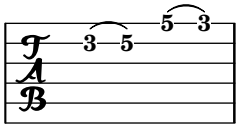
```
  }
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #1/8
    % set line length and positioning:
    % two staff spaces above center line on hidden staff
    % to four spaces below center line on visible staff
    \override GridPoint.Y-extent = #'(2 . -4)
  }
}
```



Ligados ascendentes y descendentes (*hammer-on* y *pull-off*)

Se pueden obtener ligados ascendentes o de entrada y ligados descendentes o de salida (hammer-on y pull-off) usando ligaduras de expresión.

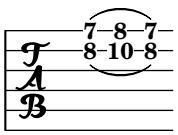
```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Indicaciones de ligado ascendente y descendente (*hammer-on* y *pull-off*) usando acordes

Cuando se usan ligados ascendentes o de entrada (*hammer-on* en inglés) o tirones de salida (*pull-off* en inglés) con notas en acorde, se dibuja un solo arco. Sin embargo son posibles "dobles arcos" si se fija la propiedad `doubleSlurs` a `#t`.

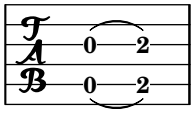
```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



Indicaciones de *hammer-on* y *pull-off* usando voces

El arco del ligado ascendente o de entrada (hammer-on) y descendente o de salida (pull-off) apunta hacia arriba en las voces uno y tres y hacia abajo en las voces dos y cuatro:
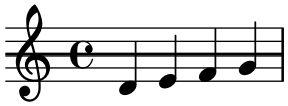
```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Hacer unas líneas del pentagrama más gruesas que las otras

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```
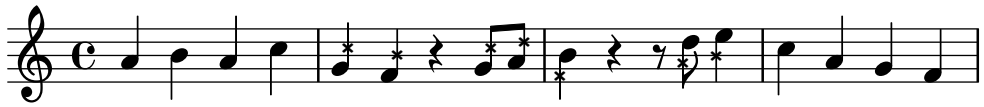


Marcar notas de las partes habladas con unas aspas en la plica

Este ejemplo muestra cómo añadir aspas sobre las plicas. Marcamos el comienzo de una sección hablada con la instrucción `\speakOn`, y el final con `\speakOff`.

```
speakOn = \override Stem.stencil =
  #(lambda (grob)
     (let* ((x-parent (ly:grob-parent grob X))
            (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
       (if is-rest?
           empty-stencil
           (ly:stencil-combine-at-edge
            (ly:stem::print grob)
            Y
            (- (ly:grob-property grob 'direction))
            (grob-interpret-markup
     grob
             (markup #:center-align #:fontsize -4
                     #:musicglyph "noteheads.s2cross"))
            -1.7))))

speakOff = \revert Stem.stencil
```

```
\new Staff {
  \relative c'' {
    a4 b a c
    \speakOn
    g4 f r g8 a
    b4 r r8 d e4
    \speakOff
    c4 a g f
  }
}
```



Measure counters

This snippet demonstrates the use of the `Measure_counter_engraver` to number groups of successive measures. Any stretch of measures may be numbered, whether consisting of repetitions or not.

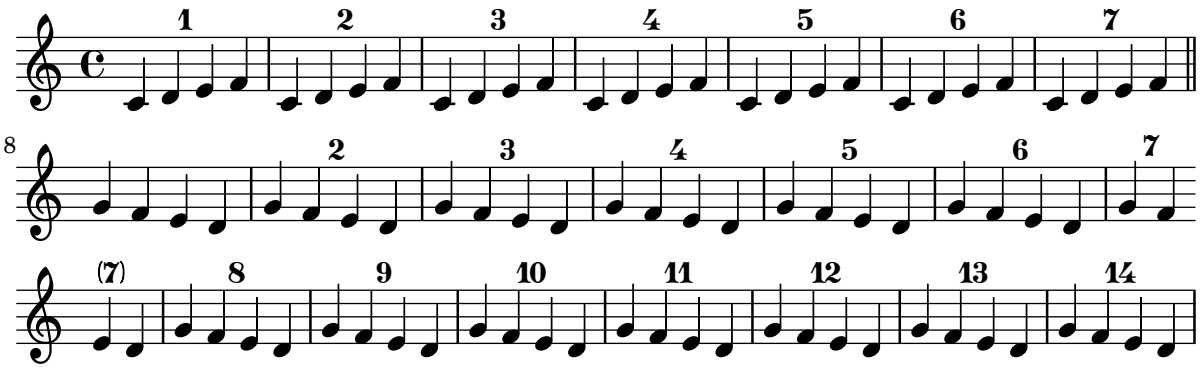
The engraver must be added to the appropriate context. Here, a `Staff` context is used; another possibility is a `Dynamics` context.

The counter is begun with `\startMeasureCount` and ended with `\stopMeasureCount`. Numbering will start by default with 1, but this behavior may be modified by overriding the `count-from` property.

When a measure extends across a line break, the number will appear twice, the second time in parentheses.

```
\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

\new Staff {
  \startMeasureCount
  \*7 { c'4 d' e' f' }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \*5 { g'4 f' e' d' }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \*7 { g'4 f' e' d' }
  \stopMeasureCount
}
```



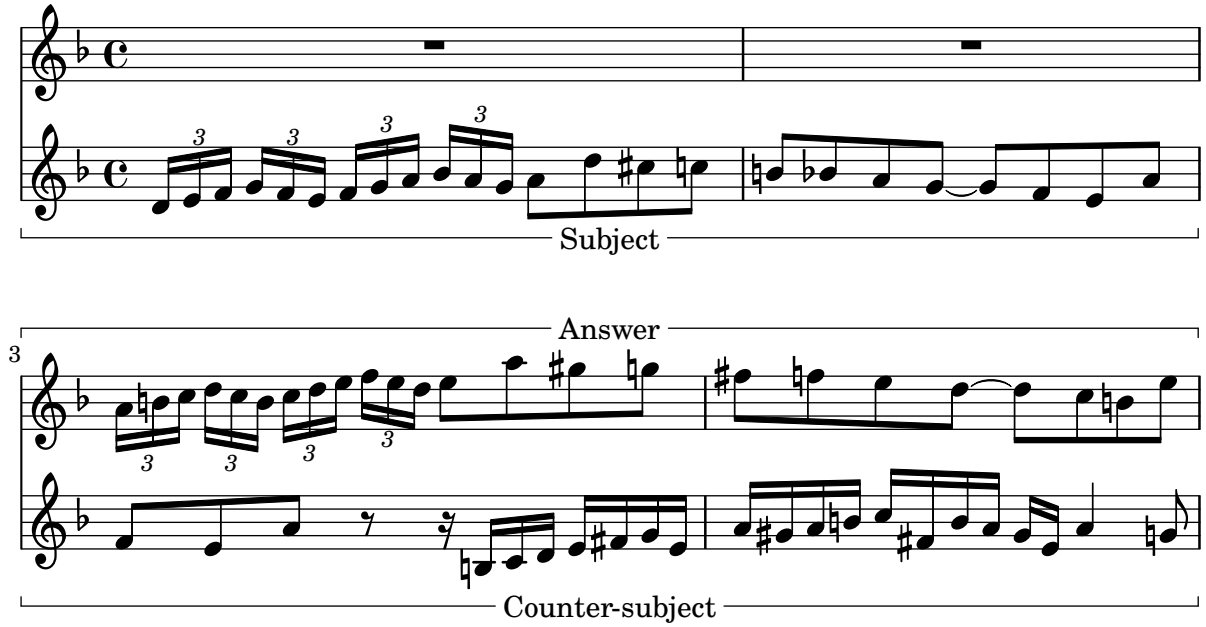
Corchete que abarca varios compases

Los elementos de extensión de compases son una forma alternativa de imprimir corchetes anotados. A diferencia de los corchetes horizontales, estos se extienden entre dos líneas divisorias en lugar de hacerlo entre dos notas. El texto se imprime en el centro del corchete.

```
\layout {
  \context {
    \Staff
    \consists Measure_spanner_engraver
  }
}

<<
  \new Staff \relative c'' {
    \key d \minor
    R1*2
    \tweak text "Answer"
    \startMeasureSpanner
    \tuplet 3/2 8 {
      a16[ b c] d[ c b]  c[ d e] f[ e d]
    }
    e8 a gis g
    fis f e d~ d c b e
    \stopMeasureSpanner
  }
  \new Staff \relative c' {
    \key d \minor
    \tweak text "Subject"
      \tweak direction #DOWN
      \startMeasureSpanner
    \tuplet 3/2 8 {
      d16[ e f] g[ f e] f[ g a] bes[ a g]
    }
    a8 d cis c
    b bes a g~ g f e a
    \stopMeasureSpanner
    \tweak text "Counter-subject"
      \tweak direction #DOWN
      \startMeasureSpanner
    f8 e a r r16 b, c d e fis g e
    a gis a b c fis, b a gis e a4 g8
```

```
    \stopMeasureSpanner
  }
>>
```



Posicionamiento preciso de indicaciones de digitación

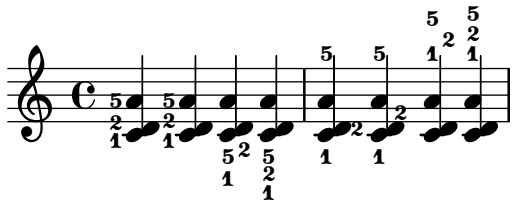
El posicionado semiautomático de las digitaciones dentro de un acorde funciona bien en casi todas las situaciones. Si una de las indicaciones precisa colocarse de forma más exacta, pueden usarse los trucos siguientes. Esto es especialmente útil para corregir el posicionado cuando están presentes intervalos de segunda.

```
\score {
  \relative c' {
    \set fingeringOrientations = #'(left)
    <c-1 d-2 a'-5>4
    <c-1 d-\tweak extra-offset #'(0 . 0.2)-2 a'-5>

    \set fingeringOrientations = #'(down)
    <c-1 d-2 a'-5>
    <c-\tweak extra-offset #'(0 . -1.1)-1
     d-\tweak extra-offset #'(-1.2 . -1.8)-2 a'-5> |

    \set fingeringOrientations = #'(down right up)
    <c-1 d-\tweak extra-offset #'(-0.3 . 0)-2 a'-5>4
    <c-1 d-\tweak extra-offset #'(-1 . 1.2)-2 a'-5>
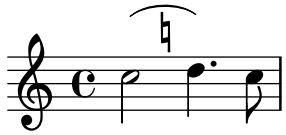
    \set fingeringOrientations = #'(up)
    <c-1 d-\tweak extra-offset #'(0 . 1.1)-2
     a'-\tweak extra-offset #'(0 . 1)-5>
    <c-1 d-\tweak extra-offset #'(-1.2 . 1.5)-2
     a'-\tweak extra-offset #'(0 . 1.4)-5> |
  }
}
```



Situar los elementos de marcado de texto por dentro de las ligaduras

Los elementos de marcado de texto deben tener la propiedad `outside-staff-priority` establecida a `#f` para que se impriman por dentro de las ligaduras de expresión.

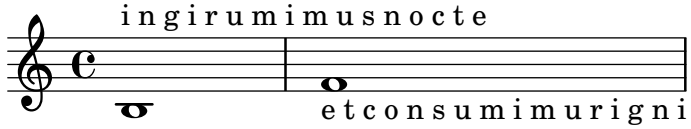
```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```



Impresión de texto de derecha a izquierda

Es posible imprimir texto de derecha a izquierda en un elemento de marcado, como se muestra aquí.

```
{
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```
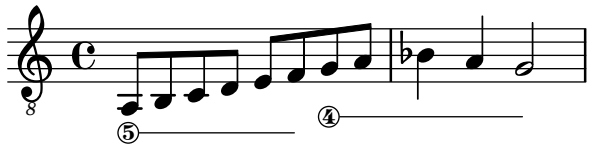


Líneas de extensión para números de cuerda

Hacemos una línea extensora para las indicaciones de número de cuerda, motrando que una serie de notas se han de tocar sobre la misma cuerda.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
```

```
    #})

\relative c {
  \clef "treble_8"
  \textSpannerDown
  \stringNumberSpanner "5" a8\startTextSpan b c d
    e f\stopTextSpan \stringNumberSpanner "4" g\startTextSpan a |
  bes4 a g2\stopTextSpan
}
```
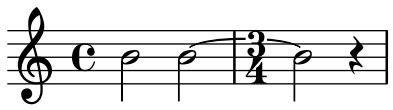


Uso de la propiedad `whiteout`

Se puede imprimir cualquier objeto gráfico sobre un fondo blanco para enmascarar parte de los objetos que están por debajo. Esto puede ser útil para mejorar el aspecto de las colisiones en situaciones complejas cuando no es práctico reposicionar los objetos. Es necesario establecer explícitamente la propiedad de capa (`layer`) para controlar qué objetos resultan enmascarados por el fondo blanco.

En este ejemplo, la colisión de la ligadura de unión con la indicación de compás resulta mejorada enmascarando la parte de la ligadura que cruza a la indicación de compás estableciendo la propiedad `whiteout` de `TimeSignature`. Para hacer esto, se mueve `TimeSignature` a una capa por encima de `Tie`, que se deja en la capa predeterminada 1, y `StaffSymbol` se mueve a una capa por encima de `TimeSignature` de manera que no resulte enmascarada.

```
{
  \override Score.StaffSymbol.layer = 4
  \override Staff.TimeSignature.layer = 3
  b'2 b'~
  \once \override Staff.TimeSignature.whiteout = ##t
  \time 3/4
  b' r4
}
```



8 Text

See also Sección “Text” en *Notation Reference*.

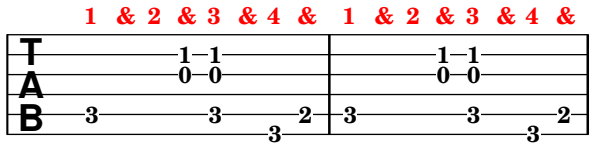
Insertar elementos de marcado en una tablatura

De forma predeterminada, los elementos de marcado no aparecen en la tablatura.

Para hacer que aparezcan, revierta la propiedad `stencil` del objeto gráfico `TextScript` dentro del contexto `TabStaff`.

```
high  = { r4 r8 <g c'> q r8 r4 }
low = { c4 r4 c8 r8 g,8 b, }
pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }

\score {
 \new TabStaff {
   \*2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```



Añadir la fecha actual a una partitura

Con algo de código de Scheme, se puede añadir fácilmente la fecha actual a una partitura.

```
\paper { tagline = ##f }
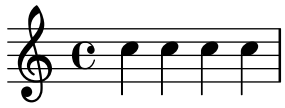
% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))

% use it in the title block:
\header {
  title = "Including the date!"
  subtitle = \date
}
```

```
\score {
  \relative c'' {
    c4 c c c
  }
}
% and use it in a \markup block:
\markup {
  \date
}
```

Including the date!

20-10-2026



20-10-2026
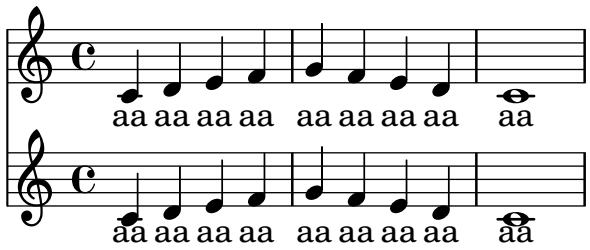
Adjusting vertical spacing of lyrics

This snippet shows how to bring the lyrics line closer to the staff.

```
music = \relative c' { c4 d e f | g4 f e d | c1 }
text = \lyricmode { aa aa aa aa aa aa aa aa aa }

<<
  \new Staff \new Voice = melody \music
  % Default layout:
  \new Lyrics \lyricsto melody \text

  \new Staff \new Voice = melody \music
  % Reducing the minimum space below the staff and above the lyrics.
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  } \lyricsto melody \text
>>
```



Alinear y centrar los nombres de instrumento

La alineación horizontal de los nombres de instrumento se puede trucar modificando la propiedad `self-alignment-X` del objeto gráfico `InstrumentName` (normalmente dentro del contexto `Staff`). Las variables de `\layout` `indent` y `short-indent` definen el espacio en que se alinean los nombres de instrumento antes del primer sistema y de los siguientes, respectivamente.

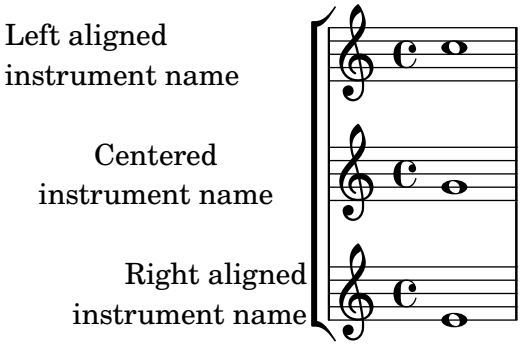
```
\paper {
  left-margin = 3\cm
}

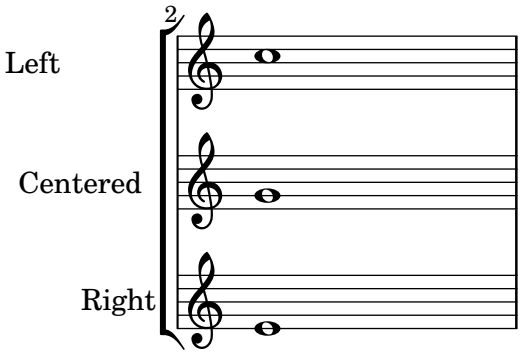
\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                            "instrument name" }
    shortInstrumentName = "Left"
  } {
    c''1 \break c''1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #CENTER
    instrumentName = \markup \center-column { Centered
                                              "instrument name" }
    shortInstrumentName = "Centered"
  } {
    g'1 g'1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #RIGHT
    instrumentName = \markup \right-column { "Right aligned"
                                             "instrument name" }
    shortInstrumentName = "Right"
  } {
    e'1 e'1
  }
>>

\layout {
  indent = 4\cm
  short-indent = 2\cm
  line-width = 6.5\cm
}
```
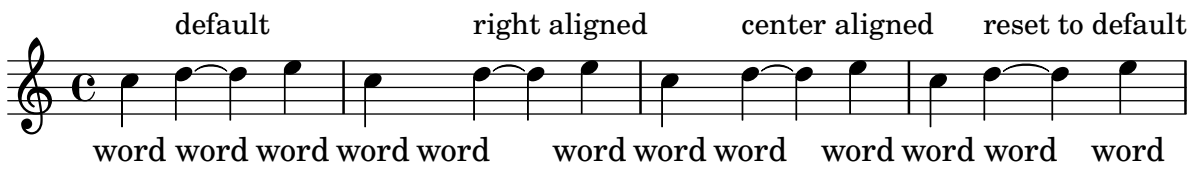




Alineación de sílabas con melisma

De forma predeterminada, las sílabas de la letra que comienzan un melisma se alinean a la izquierda sobre su nota correspondiente. Se puede alterar la alineación usando la propiedad `lyricMelismaAlignment`.

```
<<
  \new Staff {
    \new Voice = "vocal" \relative c'' {
      \override TextScript.staff-padding = #2
      c d~^\markup default  d e
      c d~^\markup "right aligned" d e
      c d~^\markup "center aligned" d e
      c d~^\markup "reset to default" d e
    }
  }
  \new Lyrics \lyricsto "vocal" {
    word word word
    \set lyricMelismaAlignment = #RIGHT
    word word word
    \set lyricMelismaAlignment = #CENTER
    word word word
    \unset lyricMelismaAlignment
    word word word
  }
>>

\layout {
  ragged-right = ##f
}
```

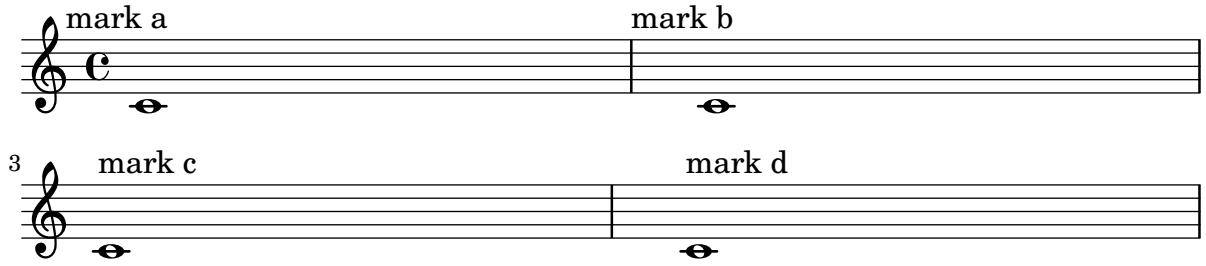


Aligning text marks to notes

By default, `TextMark` objects are aligned to so-called `NonMusicalPaperColumn` grobs, like the left edge of the staff or a bar line. They can be aligned to a note instead by setting the `non-musical` property to `#f`.

```
\layout {
  line-length = 80\mm
}

{
  \textMark "mark a" c'1 |
  \textMark "mark b" c'1 |
  \break
  \override Score.TextMark.non-musical = ##f
  \textMark "mark c" c'1 |
  \textMark "mark d" c'1 |
}
```



Poner en blanco las líneas del pentagrama utilizando la instrucción \whiteout

La instrucción \whiteout intercala un rectángulo blanco debajo de un elemento de marcado. Este rectángulo blanco no tapa a ningún otro objeto gráfico, puesto que las líneas del pentagrama están en una capa inferior a la de la mayor parte de los otros objetos.

```
\layout {
  ragged-right = ##f
}

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```



Centrar texto debajo de un regulador

Este ejemplo proporciona una función para tipografiar un regulador con texto por debajo, como "molto" o "poco". El texto añadido cambia la dirección de acuerdo con la del regulador. El objeto Hairpin está alineado con un objeto gráfico DynamicText.

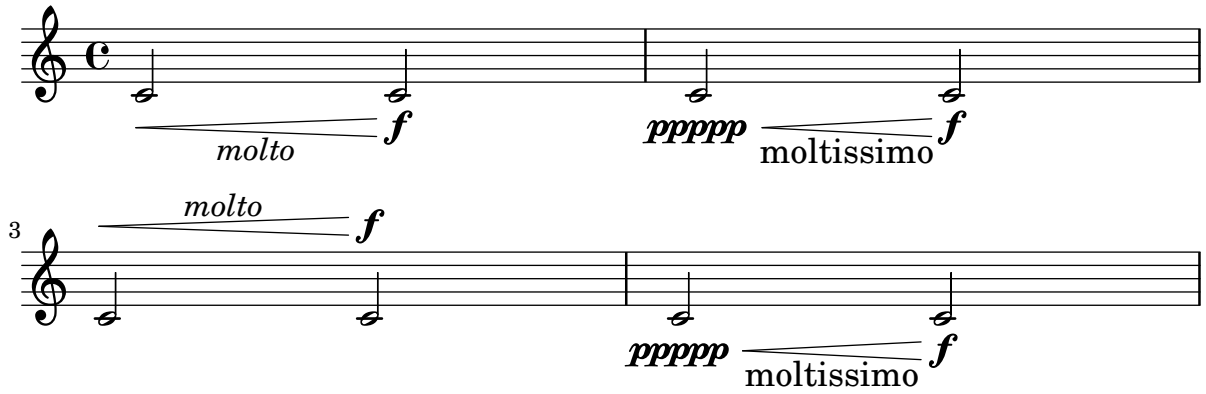
El ejemplo ilustra también cómo modificar la manera en que se imprime normalmente un objeto, utilizando código de Scheme.

```
hairpinWithCenteredText =
#(define-music-function (text) (markup?)
  #{
    \once \override Voice.Hairpin.after-line-breaking =
      #(lambda (grob)
         (let* ((stencil (ly:hairpin::print grob))
                (par-y (ly:grob-parent grob Y))
                (dir (ly:grob-property par-y 'direction))
                (staff-line-thickness
                 (ly:output-def-lookup (ly:grob-layout grob)
                                       'line-thickness))
                (new-stencil
                 (ly:stencil-aligned-to
                  (ly:stencil-combine-at-edge
                   (ly:stencil-aligned-to stencil X CENTER)
                   Y dir
                   (ly:stencil-aligned-to
                    (grob-interpret-markup
                     grob
                     (make-fontsize-markup
                      (magnification->font-size
                       (+ (ly:staff-symbol-staff-space grob)
                          (/ staff-line-thickness 2)))
                      text))
                    X CENTER))
                  X LEFT))
                (staff-space (ly:output-def-lookup
                 (ly:grob-layout grob) 'staff-space))
                (par-x (ly:grob-parent grob X))
                (dyn-text (grob::has-interface par-x
                                               'dynamic-text-interface))
                (dyn-text-stencil-x-length
                 (if dyn-text
                     (interval-length
                      (ly:stencil-extent
                       (ly:grob-property par-x 'stencil) X))
                     0))
                (x-shift
                 (if dyn-text (- (+ staff-space dyn-text-stencil-x-length)
                                 (* 0.5 staff-line-thickness))
                     0)))
           (ly:grob-set-property! grob 'Y-offset 0)
           (ly:grob-set-property! grob
                                  'stencil (ly:stencil-translate-axis
                                            new-stencil
                                            x-shift X))))
  #})

hairpinMolto = \hairpinWithCenteredText \markup { \italic molto }
hairpinMore = \hairpinWithCenteredText \markup { \larger moltissimo }

\relative c' {
```

```
  \hairpinMolto c2\< c\f
  \hairpinMore c2\ppppp\< c\f
  \break
  \hairpinMolto c2^\< c\f
  \hairpinMore c2\ppppp\< c\f
}
```

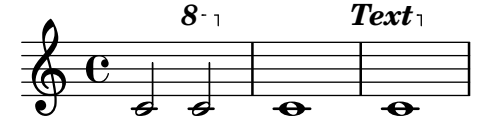


Cambiar el texto de las indicaciones de octava alta y baja

Internamente, `\ottava` fija las propiedades `ottavation` (por ejemplo para `8va` o `8vb`) y `middleCPosition`. Para sobreescribir el texto del corchete, fije `ottavation` después de invocar a `\ottava`.

Un texto breve es especialmente útil cuando se usa una indicación corta de octava.

```
{
  c'2
  \ottava 1
  \set Staff.ottavation = "8"
  c''2
  \ottava 0
  c'1
  \ottava 1
  \set Staff.ottavation = "Text"
  c''1
}
```



Cambiar la familia de tipografía predeterminada para el texto

Las familias de fuente tipográfica para el texto se pueden sobreescribir.

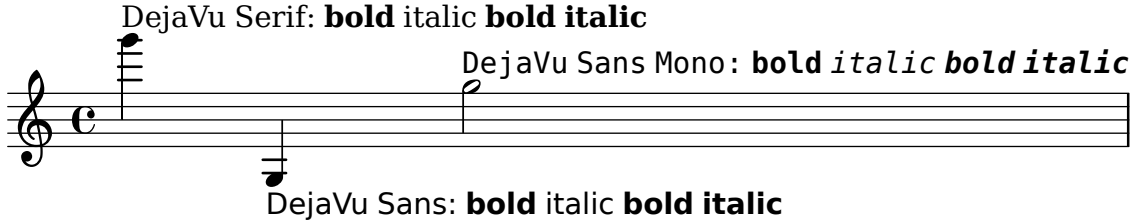
```
%{
You may have to install additional fonts.

Red Hat Fedora: dejavu-fonts-all

Debian GNU/Linux, Ubuntu: fonts-dejavu-core
                          fonts-dejavu-extra
%}
```

```
\paper {
  %{
     run
         lilypond -dshow-available-fonts
     to show all fonts available in the process log.
  %}
  property-defaults.fonts.serif = "DejaVu Serif"
  property-defaults.fonts.sans = "DejaVu Sans"
  property-defaults.fonts.typewriter = "DejaVu Sans Mono"
}

{
  g'''4^\markup {
    DejaVu Serif: \bold bold
                  \italic italic
                  \italic \bold { bold italic }
  }
  g4_\markup {
    \override #'(font-family . sans) {
      DejaVu Sans: \bold bold
                   \italic italic
                   \italic \bold { bold italic }
    }
  }
  g''2^\markup {
    \override #'(font-family . typewriter) {
      DejaVu Sans Mono: \bold bold
                        \italic italic
                        \italic \bold { bold italic }
    }
  }
}
```
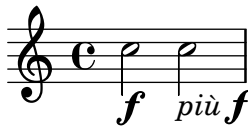


Combinar indicaciones dinámicas con marcados textuales

Ciertas indicaciones dinámicas pueden llevar textos (como "più *f*" o "*p* subito"). Se pueden producir usando un bloque `\markup`; el objeto resultante se comporta como un objeto gráfico `TextScript`.

See also "Combining dynamics with markup texts (2)".

```
piuF = \markup { \italic più \dynamic f }

\score {
  \relative c'' {
    c2\f c-\piuF
  }
}
```
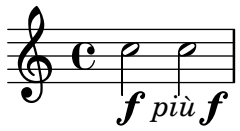


Combining dynamics with markup texts (2)

Some dynamics may involve text indications (such as "più *f*" or "*p* subito"). These can be produced using the `make-dynamic-script` Scheme function; the resulting object behaves like a `DynamicText` grob.

See also "Combining dynamics with markup texts".

```
piuF = #(make-dynamic-script
          #{ \markup { \normal-text \italic più \dynamic f } #})

\score {
  \relative c'' {
    c2\f c\piuF
  }
}
```



Combinar dos partes sobre el mismo pentagrama

La herramienta de combinación de partes (o sea, la instrucción `\partCombine`) permite la combinación de varias partes diferentes sobre el mismo pentagrama. Las indicaciones textuales tales como "solo" o "a2" se añaden de forma predeterminada; para quitarlas, sencillamente establezca la propiedad `printPartCombineTexts` al valor `#f`. Para partituras vocales (como himnos), no hay necesidad de añadir los textos "solo" o "a2", por lo que se deben desactivar. Sin embargo, podría ser mejor no usarlos si hay solos, porque éstos no se indicarán. En tales casos podría ser preferible la notación polifónica estándar.

Este fragmento de código presenta las tres formas en que se pueden imprimir dos partes sobre un solo pentagrama: polifonía estándar, `\partCombine` sin textos, y `\partCombine` con textos.

```
musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
```
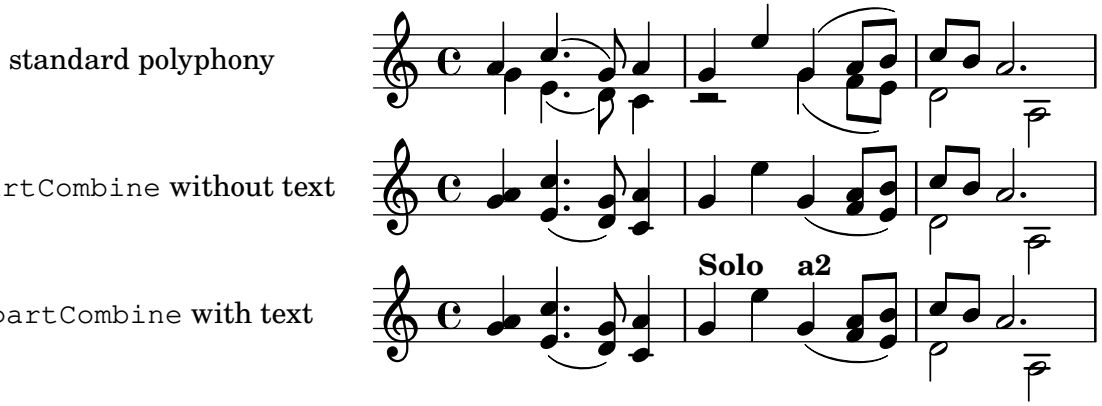
```
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
  <<
    \new Staff \with {
      instrumentName = "standard polyphony"
    } << \musicUp \\ \musicDown >>

    \new Staff \with {
      instrumentName =
        \markup { \typewriter "\\partCombine" without text}
      printPartCombineTexts = ##f
    } \partCombine \musicUp \musicDown

    \new Staff \with {
      instrumentName =
        \markup { \typewriter "\\partCombine" with text}
    } \partCombine \musicUp \musicDown
  >>

  \layout {
    indent = 6.0\cm
    \context {
      \Score
      % Setting this to a large value avoids a bar line at the
      % beginning that would connect the three staves otherwise.
      \override SystemStartBar.collapse-height = 30
    }
  }
}
```



Crear elementos de extensión textuales

Las instrucciones \startTextSpan y \stopTextSpan permiten la creación de elementos de extensión textuales tan fácilmente como indicaciones de pedal u octavaciones. Sobreescribimos ciertas propiedades del objeto TextSpanner para modificar su salida.

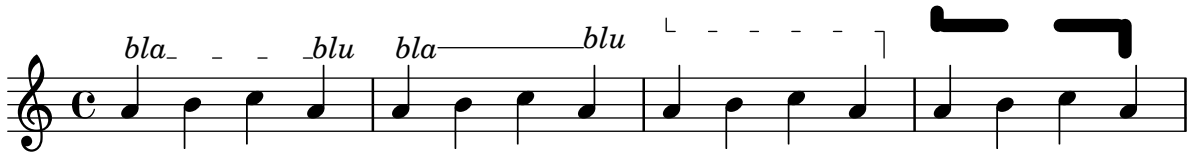
```
\paper { ragged-right = ##f }

\relative c'' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.bound-details.left.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . -2) }
  \once \override TextSpanner.bound-details.right.padding = #-2
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.dash-period = #10
  \override TextSpanner.dash-fraction = #0.5
  \override TextSpanner.thickness = #10
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}
```



Demonstrating all \header fields

A demonstration of all header fields that LilyPond defines by default. Thanks to setting `print-all-headers` to `#t`, much more fields as usual are displayed, indicating the hierarchy of `\header` blocks.

```
\paper {
  #(set-paper-size "a6" 'landscape)
  print-all-headers = ##t
}

\book {
  \header {
    title = "title"
    subtitle = "subtitle"
```

```
    composer = "composer"
    arranger = "arranger"
    instrument = "instrument"
    meter = "meter"
    opus = "opus"
    piece = "piece"
    poet = "poet"
    copyright = "copyright"
    tagline = "tagline"
  }

  \bookpart {
    \score {
      \relative c'' { c1 | c | c | c }

      \header {
        title = "localtitle"
        subtitle = "localsubtitle"
        composer = "localcomposer"
        arranger = "localarranger"
        instrument = "localinstrument"
        meter = "localmeter"
        opus = "localopus"
        piece = "localpiece"
        poet = "localpoet"
        copyright = "localcopyright"
        tagline = "localtagline"
      }
    }
  }
}
```

title

subtitle

poet **instrument** composer

meter arranger
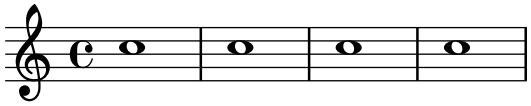
localtitle

localsubtitle

localpoet **localinstrument** localcomposer

localmeter localarranger

localpiece localopus



copyright

tagline

Empotrar PostScript nativo dentro de un bloque \markup

Se puede insertar código PostScript directamente dentro de un bloque \markup.

En general se recomienda usar en su lugar las instrucciones de marcado nativas de LilyPond tales como \polygon, que se pueden usar con todos los *backends* de LilyPond.

```
\relative c'' {
  a2-\markup \postscript "0 3 moveto
                          5 2 rlineto
                          stroke"
    -\markup \postscript "[1 1] 0 setdash
                          0 0 moveto
                          5 2 rlineto
                          stroke"
  b2-\markup \postscript "1 1 moveto
                          0 0 1 2 8 4 10 2 rcurveto
                          stroke"
  a'1
}
```

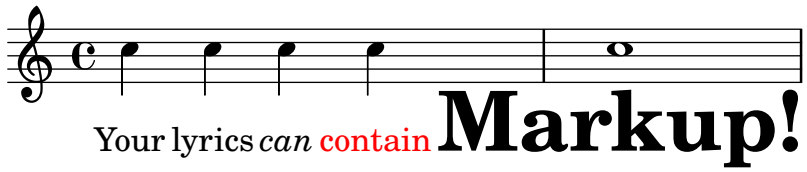


Dar formato a sílabas de la letra

Es posible usar la instrucción \markup dentro de un bloque \lyricmode para dar formato a las sílabas individuales dentro de la letra.

```
mel = \relative c'' { c4 c c c c1 }
lyr = \lyricmode {
  Your lyrics \markup { \italic can }
  \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

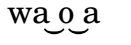
<<
  \new Voice = "melody" \mel
  \new Lyrics \lyricsto "melody" \lyr
>>
```



Cómo poner ligaduras entre sílabas de la letra

Se puede hacer separando las sílabas mediante símbolos de tilde curva.

```
\lyrics {
  wa~o~a
}
```



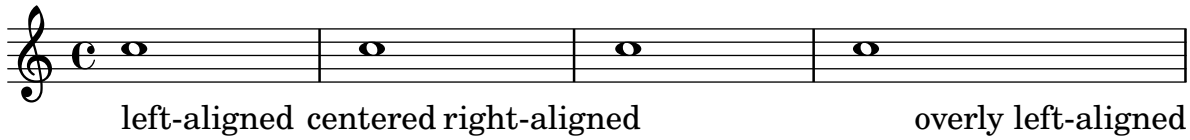
Alineación de la letra

La alineación horizontal de la letra se puede ajustar sobreescribiendo la propiedad `self-alignment-X` del objeto `LyricText`. El valor -1 significa alineado por la izquierda, 0 es centrado y 1 es derecha; de forma alternativa también podemos usar los valores de Scheme `#LEFT`, `#CENTER` y `#RIGHT` en lugar de números. También son posibles otros valores numéricos. Importante: no olvide añadir el prfijo de Scheme '`#`' para los números negativos.

```
\layout {
  ragged-right = ##f
}

\relative c'' {
  c1 c c c
}

\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "centered"
  \once \override LyricText.self-alignment-X = 1
  "right-aligned"
  \once \override LyricText.self-alignment-X = #-1.5
  "overly left-aligned"
}
```



Objetos de lista de marcados

El texto que se extiende sobra más de una página se introduce con la instrucción `\markuplist`. La instrucción de marcado `\paragraph` definida dentro del fragmento de código aplica un sangrado a su argumento antes de llamar a la instrucción `\justified-lines`.

```
#(set-default-paper-size "a6" 'landscape)

\paper {
  line-width = 11\cm
  tagline = ##f
}

#(define-markup-list-command (paragraph layout props args) (markup-list?)
  (interpret-markup-list layout props
   (make-justified-lines-markup-list (cons (make-hspace-markup 2) args))))

\book { % for correct rendering in the PDF documentation
  % Candide, Voltaire
  \markuplist {
    \override-lines #'(baseline-skip . 2.5) {
      \paragraph {
        Il y avait en Westphalie, dans le château de M. le baron de
```

```
        Thunder-ten-tronckh, un jeune garçon à qui la nature avait donné
        les mœurs les plus douces.  Sa physionomie annonçait son âme.
        Il avait le jugement assez droit, avec l'esprit le plus
        \concat { simple \hspace #.3 ; }
        c'est, je crois, pour cette raison qu'on le nommait Candide.  Les
        anciens domestiques de la maison soupçonnaient qu'il était fils
        de la sœur de monsieur le baron et d'un bon et honnête
        gentilhomme du voisinage, que cette demoiselle ne voulut jamais
        épouser parce qu'il n'avait pu prouver que soixante et onze
        quartiers, et que le reste de son arbre généalogique avait été
        perdu par l'injure du temps.
      }
      \vspace #.3
      \paragraph {
        Monsieur le baron était un des plus puissants seigneurs de la
        Westphalie, car son château avait une porte et des fenêtres.  Sa
        grande salle même était ornée d'une tapisserie.  Tous les chiens
        de ses basses-cours composaient une meute dans le
        \concat { besoin \hspace #.3 ; }
        ses palefreniers étaient ses
        \concat { piqueurs \hspace #.3 ; }
        le vicaire du village était
        son grand-aumônier.  Ils l'appelaient tous monseigneur, et ils
        riaient quand il faisait des contes.
      }
    }
  }
}
```

> Il y avait en Westphalie, dans le château de M. le baron de Thunder-ten-tronckh, un jeune garçon à qui la nature avait donné les mœurs les plus douces. Sa physionomie annonçait son âme. Il avait le jugement assez droit, avec l'esprit le plus simple ; c'est, je crois, pour cette raison qu'on le nommait Candide. Les anciens domestiques de la maison soupçonnaient qu'il était fils de la sœur de monsieur le baron et d'un bon et honnête gentilhomme du voisinage, que cette demoiselle ne voulut jamais épouser parce qu'il n'avait pu prouver que soixante et onze quartiers, et que le reste de son arbre généalogique avait été perdu par l'injure du temps.
>
> Monsieur le baron était un des plus puissants seigneurs de la Westphalie, car son château avait une porte et des fenêtres. Sa grande salle même était ornée d'une tapisserie. Tous les chiens de ses basses-cours composaient une meute dans le besoin ; ses palefreniers étaient ses piqueurs ; le vicaire du village était son grand-aumônier. Ils l'appelaient tous monseigneur, et ils riaient quand il faisait des contes.

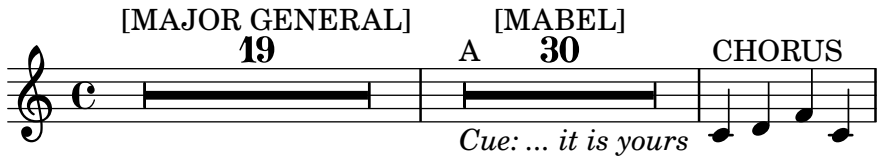
Marcado de silencios multicompás

Los elementos de marcado aplicados a un silencio multicompás se centran encima o debajo de éste. Los elementos de marcado extensos que se adjuntan a silencios multicompás no producen

la expansión del compás. Para expandir un silencio multicompás de forma que quepa todo el marcado, utilice un acorde vacío con un marcado aplicado antes del silencio multicompás.

El texto aplicado a un siencio sparador de esta forma se alinea por la izquierda a la posición en que la nota estaría situada dentro del compás, pero si la longitud del compás está determinada por la longitud del texto, éste aparecerá centrado.

```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```



Acerca de la ubicuidad de los objetos de marcado

Los objetos de texto se escriben como simples cadenas de texto entrecomilladas, o como bloques `\markup` que pueden aceptar una amplia variedad de enriquecimientos de formato de texto y gráficos.

Los bloques de marcado de texto se pueden usar:

- dentro de cualquier objeto `TextScript` (adjunto a notas con `-`, `^` o `_`),
- dentro de cualquier marca `TextMark` escrita con la palabra clave `\textMark`, o con la instrucción `\textEndMark`, u otros objetos similares como `MetronomeMark` escrito con la instrucción `\tempo`,
- como bloques de marcado autosuficientes, introducidos en el nivel superior jerárquico fuera de cualquier bloque `\score`,
- dentro de cualquier definición dentro del bloque `\header` (p.ej. título, subtítulo, autor) o dentro de variables definidas dentro del bloque `\paper` tales como `evenHeaderMarkup` para los números de página.

`\markup` se puede además usar para la letra de las canciones, en los nombres de acorde, y como matices de dinámica. De hecho, es posible usar `\markup` para personalizar la apariencia de prácticamente cualquier objeto, como se muestra en este ejemplo usando varios métodos.

```
\paper {
  paper-width = 8\cm
  paper-height = 8\cm
}

\header {
  title = \markup "Title"
  tagline = \markup "(tagline)"
}
```

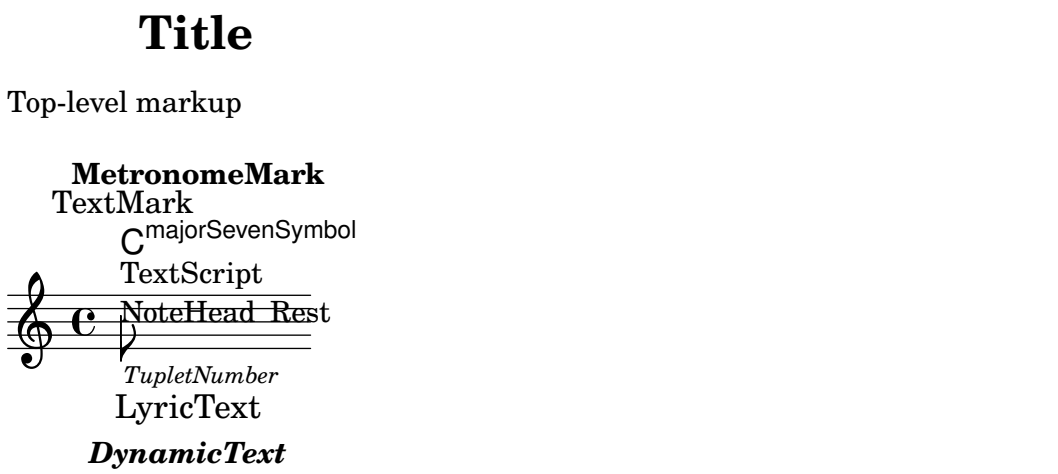
```
\markup "Top-level markup"

dyn = #(make-dynamic-script #{ \markup \serif "DynamicText" #})

<<
  \new ChordNames \with {
    majorSevenSymbol = \markup "majorSevenSymbol"
  } \chordmode { c1:maj7 }
  \new Staff {
    \tempo \markup "MetronomeMark"
    \textMark \markup "TextMark"

    \once \override TupletNumber.text = \markup "TupletNumber"
      \tuplet 3/2 {
        \once \override NoteHead.stencil = #ly:text-interface::print
        \once \override NoteHead.text = \markup \lower #0.5 "NoteHead"
        c''8^\markup "TextScript"

        \once \override Rest.stencil = #(lambda (grob)
          (grob-interpret-markup grob #{ \markup "Rest" #}))
        r4
      }
  }
  \new Lyrics \lyricmode { \markup "LyricText" 1 }
  \new Dynamics { s1\dyn }
>>
```



Imprimir el número de versión

Es posible imprimir el número de versión de LilyPond dentro de un elemento de marcado.

```
\markup { Processed with LilyPond version #(lilypond-version) }
```

Processed with LilyPond version 2.27.4

Plantilla de piano con letra centrada

En lugar de tener un pentagrama dedicado a la melodía y la letra, ésta se puede centrar entre los pentagramas de un sistema de piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

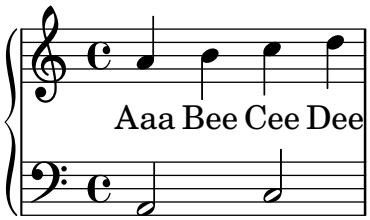
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}
```

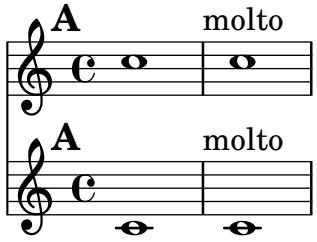


Imprimir marcas de ensayo en cualquier pentagrama

Aunque normalmente las marcas de ensayo y textuales solo se imprimen sobre el pentagrama superior, también se pueden imprimir en otro pentagrama cualquiera.

```
\score {
  <<
    \new Staff { \mark \default c''1 \textMark "molto" c'' }
    \new Staff { \mark \default c'1 \textMark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove Mark_engraver
      \remove Text_mark_engraver
      \remove Staff_collecting_engraver
```

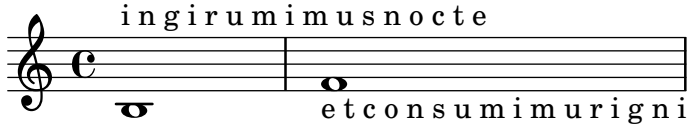
```
    }
    \context {
      \Staff
      \consists Mark_engraver
      \consists Text_mark_engraver
      \consists Staff_collecting_engraver
    }
  }
}
```



Impresión de texto de derecha a izquierda

Es posible imprimir texto de derecha a izquierda en un elemento de marcado, como se muestra aquí.
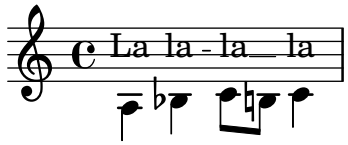
```
{
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```



Putting lyrics inside the staff

Se pueden mover las líneas de letra verticalmente para imprimirlas dentro del pentagrama. Las letras se mueven con `\override LyricText.extra-offset = #'(0 . dy)`, y hay instrucciones similares para mover las líneas extensoras y los guiones. El mejor valor para el desplazamiento *dy* se debe buscar mediante un proceso de ensayo y error.

```
<<
  \new Staff <<
    \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
  >>
  \new Lyrics \with {
    \override LyricText.extra-offset = #'(0 . 8.6)
    \override LyricExtender.extra-offset = #'(0 . 8.6)
    \override LyricHyphen.extra-offset = #'(0 . 8.6)
  } \lyricsto "voc" { La la -- la __ _ la }
>>
```
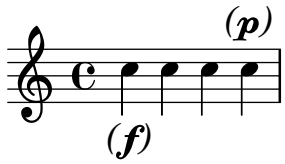


Slanted parentheses around dynamics

The easiest way to parenthesize dynamics is to use the normal \parenthesize command and adjust the font size, e.g., { c'4\tweak Parentheses.font-size 2 \parenthesize \f }. However, this results in upright parentheses, while dynamic marks are normally slanted. This snippet gives a method to make the parentheses slanted as well.

```
paren =
#(define-event-function (dyn) (ly:event?)
   (make-dynamic-script
    #{ \markup \concat {
         \normal-text \italic \fontsize #2 (
         \pad-x #0.2 #(ly:music-property dyn 'text)
         \normal-text \italic \fontsize #2 )
       }
    #}))

\relative c'' {
  c4\paren\f c c \dynamicUp c\paren\p
}
```



Elemento de marcado de texto independiente en dos columnas

Los textos independientes se pueden disponer en varias columnas utilizando instrucciones \markup

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { O sacrum convivium }
      \line { \bold \italic in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
    \hspace #2
    \column \italic {
      \line { O sacred feast }
      \line { \bold \italic in which Christ is received, }
      \line { the memory of His Passion is renewed, }
      \line { the mind is \italic filled with grace, }
```

```
      \line { and a pledge of future glory is given to us. }
      \line { Amen. }
    }
    \hspace #1
  }
}
```

O sacrum convivium
in quo Christus sumitur,
recolitur memoria passionis ejus,
mens impletur gratia,
futurae gloriae nobis pignus datur.
Amen.
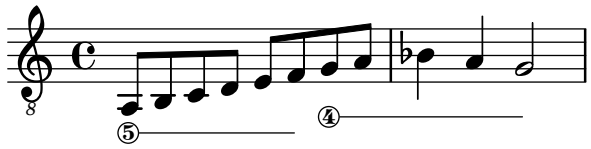
O sacred feast
in which Christ is received,
the memory of His Passion is renewed,
the mind is filled with grace,
and a pledge of future glory is given to us.
Amen.

Líneas de extensión para números de cuerda

Hacemos una línea extensora para las indicaciones de número de cuerda, motrando que una serie de notas se han de tocar sobre la misma cuerda.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})

\relative c {
  \clef "treble_8"
  \textSpannerDown
  \stringNumberSpanner "5" a8\startTextSpan b c d
    e f\stopTextSpan \stringNumberSpanner "4" g\startTextSpan a |
  bes4 a g2\stopTextSpan
}
```



Rectángulo de tres segmentos visibles

Este ejemplo muestra cómo añadir ina instrucción de marcado para obtener un rectángulo de tres segmentos visibles rodeando un elemento de texto (u otro elemento de marcado).

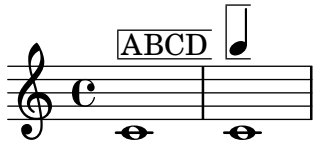
```
% New command to add a three-sided box, with sides north, west, and south.
% Based on the `box-stencil` command defined in `scm/stencil.scm`.
% Note that ";;" is used to comment a line in Scheme.
#(define-public (NWS-box-stencil stencil thickness padding)
   "Add a box around STENCIL, producing a new stencil."
   (let* ((x-ext (interval-widen (ly:stencil-extent stencil X) padding))
          (y-ext (interval-widen (ly:stencil-extent stencil Y) padding))
```

```
          (y-rule (make-filled-box-stencil (cons 0 thickness) y-ext))
          (x-rule (make-filled-box-stencil
                   (interval-widen x-ext thickness) (cons 0 thickness))))
     ;; (set! stencil (ly:stencil-combine-at-edge stencil X 1 y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil X LEFT y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil Y UP x-rule 0.0))
     (set! stencil (ly:stencil-combine-at-edge stencil Y DOWN x-rule 0.0))
     stencil))

% The corresponding markup command, based on the `\box` command defined
% in `scm/define-markup-commands.scm`.
#(define-markup-command (NWS-box layout props arg) (markup?)
   #:properties ((thickness 0.1) (font-size 0) (box-padding 0.2))
   "Draw a box round ARG.

Look at THICKNESS, BOX-PADDING, and FONT-SIZE properties to determine
line thickness and padding around the markup."
   (let ((pad (* (magstep font-size) box-padding))
         (m (interpret-markup layout props arg)))
     (NWS-box-stencil m thickness pad)))


\relative c' {
  c1^\markup { \NWS-box ABCD }
  c1^\markup { \NWS-box \note {4} #1.0 }
}
```



UTF-8

Se pueden usar varios sistemas de escritura para textos (como títulos o la letra de las canciones) si se introducen en la codificación UTF-8, y utilizando un motor basado en Pango. Dependiendo de las tipografías instaladas, este fragmento compone textos en Búlgaro (cirílico), hebreo, japonés y portugués.

```
%{
You may have to install additional fonts.

Red Hat Fedora: linux-libertine-fonts (Latin, Cyrillic, Hebrew)
                google-noto-serif-jp-fonts (Japanese)

Debian GNU/Linux, Ubuntu: fonts-linuxlibertine (Latin, Cyrillic, Hebrew)
                          fonts-noto-cjk (Japanese)
%}

% 'Linux Libertine' fonts also contain Cyrillic and Hebrew glyphs.
\paper {
  property-defaults.fonts.serif =
    "Linux Libertine O, Noto Serif CJK JP, Noto Serif JP"
```

```
}

bulgarian = \lyricmode {
  Жълтата дюля беше щастлива, че пухът, който цъфна, замръзна като гьон.
}

hebrew = \lyricmode {
           .
}

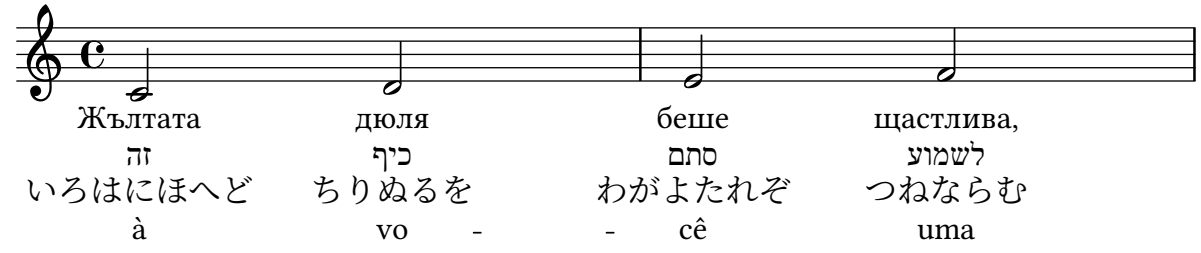
japanese = \lyricmode {
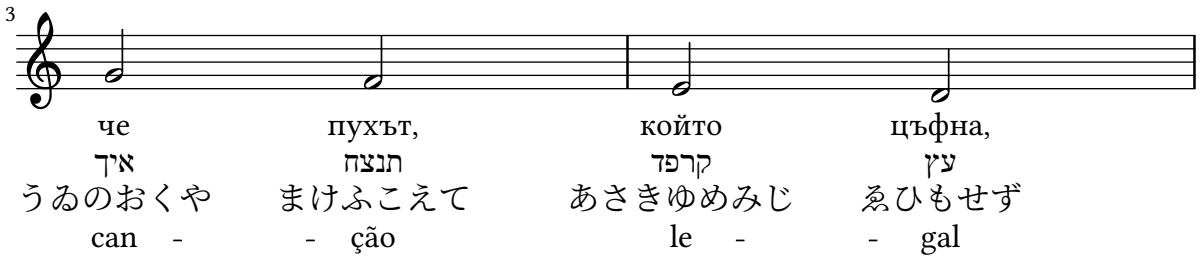



}

% "a nice song for you"
portuguese = \lyricmode {
  à vo -- cê uma can -- ção le -- gal
}

\relative c' {
  c2 d
  e2 f
  g2 f
  e2 d
}
\addlyrics { \bulgarian }
\addlyrics { \hebrew }
\addlyrics { \japanese }
\addlyrics { \portuguese }
```





Plantilla para conjunto vocal con letras alineadas encima y debajo de los pentagramas

Esta plantilla es, básicamente, la misma que la sencilla "Plantilla para conjunto vocal", excepto que aquí todas las líneas de letra se colocan utilizando `alignAboveContext` y `alignBelowContext`.

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
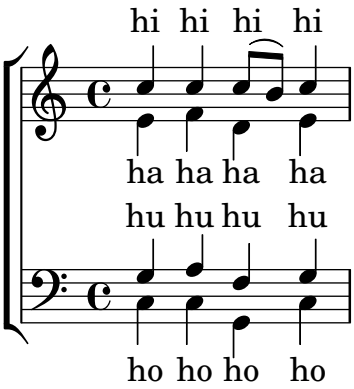
altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative c {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = "women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords
```

```
    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = "men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
    % \new Lyrics \lyricsto "basses" \bassWords
  >>
}
```
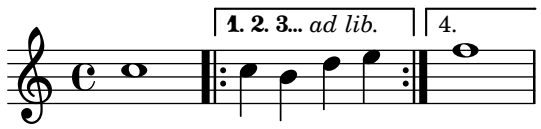


Elementos de marcado para el texto de las casillas de repetición usando `repeatCommands`

Aunque las casillas de repetición se especifican de forma óptima usando `\repeat volta`, debe usarse la propiedad de contexto `repeatCommands` en caso de que el texto de la casilla requiera un formato más avanzado con `\markup`.

Puesto que `repeatCommands` admite una lista, el método más sencillo de incluir elementos de marcado es usar un identificador para el texto e insertarlo en la lista de instrucciones usando la sintaxis de Scheme `#`((volta ,`*textIdentifier*`) ...)` (observe el uso del apóstrofo invertido después de # y la coma antes de *textIdentifier*). Las instrucciones de principio y fin de repetición se pueden añadir como elementos de lista independientes:

```
voltaAdLib = \markup { \volta-number { 1. 2. 3... } \italic { ad lib. } }

\relative c'' {
  c1
  \set Score.repeatCommands = #`((volta ,voltaAdLib) start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```



Specialist notation
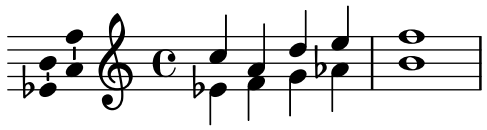
9 Vocal music

See also Sección “Vocal music” en *Notation Reference*.

Añadir un ámbito por voz

Se puede añadir un ámbito por cada voz. En este caso, el ámbito se debe desplazar manualmente para evitar colisiones.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Añadir indicadores a los pentagramas que se dividen después de un salto de línea

Este fragmento de código define las instrucciones `\splitStaffBarLine`, `\convUpStaffBarLine` y `\convDownStaffBarLine`, que añaden flechas en las sobre una línea divisoria para denotar que cada una de las voces que comparten un pentagrama continúan en su propio pentagrama en el siguiente sistema, o que las voces así divididas se recombinan.

Observe que la implementación de este fragmento de código traza unas flechas sin dimensiones en el margen derecho. Para la impresión formal, esto no produce ningún problema. Sin embargo, es necesario aumentar las dimensiones de la caja circundante en el sentido horizontal si se procesa el código como imagen para evitar el recorte, como se muestra abajo.

```
#(define-markup-command (arrow-at-angle layout props angle-deg length fill)
   (number? number? boolean?)
   (let* ((PI-OVER-180 (/ (atan 1 1) 34))
          (degrees->radians (lambda (degrees) (* degrees PI-OVER-180)))
          (angle-rad (degrees->radians angle-deg))
          (target-x (* length (cos angle-rad)))
          (target-y (* length (sin angle-rad))))
     (interpret-markup layout props
       (markup
        #:translate (cons (/ target-x 2) (/ target-y 2))
```

```
        #:rotate angle-deg
        #:translate (cons (/ length -2) 0)
        #:concat (#:draw-line (cons length 0)
                   #:arrow-head X RIGHT fill)))))

splitStaffBarLineMarkup = \markup \with-dimensions #'(0 . 0) #'(0 . 0) {
  \combine
  \arrow-at-angle #45 #(sqrt 8) ##t
  \arrow-at-angle #-45 #(sqrt 8) ##t
}

splitStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob splitStaffBarLineMarkup)
      0))
  \break
}

convDownStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . -.13)\arrow-at-angle #-45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

convUpStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . .14)\arrow-at-angle #45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

\paper {
  indent = 10\mm
```

```
  short-indent = 10\mm
  line-width = 8\cm
}

separateSopranos = {
  \set Staff.instrumentName = "AI AII"
  \set Staff.shortInstrumentName = "AI AII"
  \splitStaffBarLine
  \change Staff = "up"
}
convSopranos = {
  \convDownStaffBarLine
  \change Staff = "shared"
  \set Staff.instrumentName = "S A"
  \set Staff.shortInstrumentName = "S A"
}

sI = {
  \voiceOne
  \*4 f''2
  \separateSopranos
  \*4 g''2
  \convSopranos
  \*4 c''2
}
sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \*4 d''2
}
aI = {
  \voiceTwo
  \*4 a'2
  \voiceOne
  \*4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \*4 g'2
}
aII = {
  s1*2
  \voiceTwo
  \*4 g'2
}
ten = {
  \voiceOne
  \*4 c'2
  \*4 d'2
  \*4 c'2
}
bas = {
```

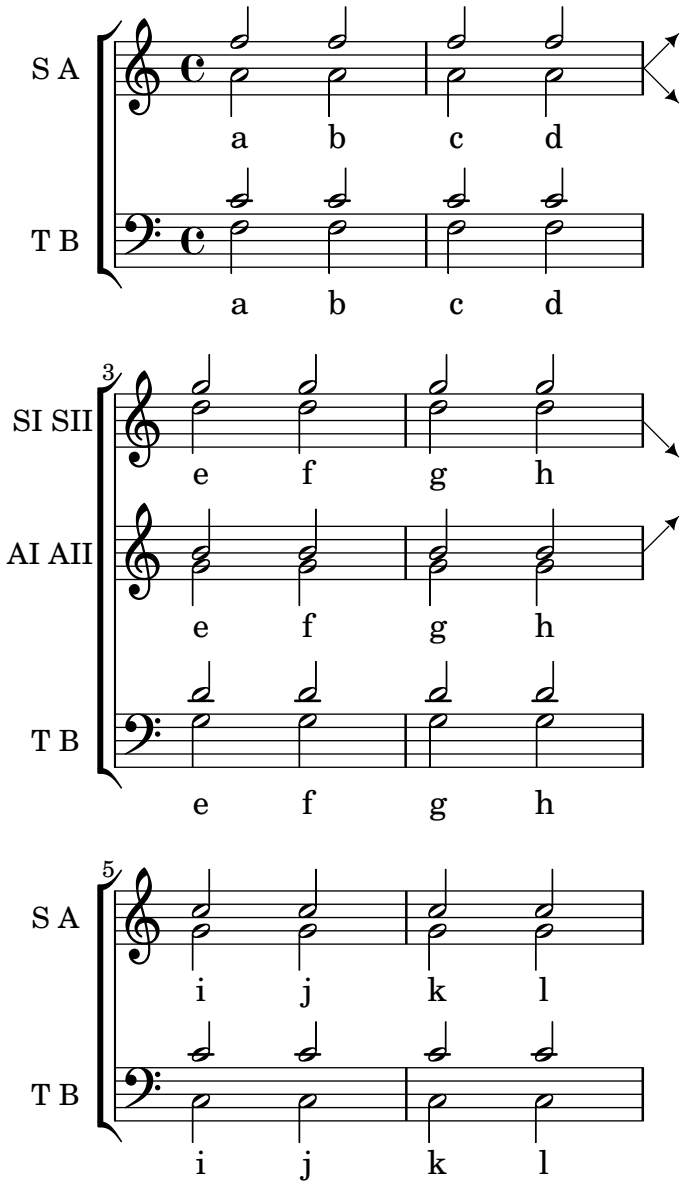
```
  \voiceTwo
  \*4 f2
  \*4 g2
  \*4 c2
}

\markup \pad-x #3  % avoid cropping
  \score {
    <<
      \new ChoirStaff <<
        \new Staff = up \with {
          instrumentName = "SI SII"
          shortInstrumentName = "SI SII"
        } {
          s1*4
        }

        \new Staff = shared \with {
          instrumentName = "S A"
          shortInstrumentName = "S A"
        } <<
          \new Voice = sopI \sI
          \new Voice = sopII \sII
          \new Voice = altI \aI
          \new Voice = altII \aII
        >>
        \new Lyrics \with {
          alignBelowContext = up
        }
        \lyricsto sopII { e f g h }
        \new Lyrics \lyricsto altI { a b c d e f g h i j k l }

        \new Staff = men \with {
          instrumentName = "T B"
          shortInstrumentName = "T B"
        } <<
          \clef F
          \new Voice = ten \ten
          \new Voice = bas \bas
        >>
        \new Lyrics \lyricsto bas { a b c d e f g h i j k l }
      >>
    >>

    \layout {
      \context {
        \Staff \RemoveEmptyStaves
        \override VerticalAxisGroup.remove-first = ##t
      }
    }
}
```



Añadir notas guía orquestales a una partitura vocal

Este ejemplo muestra una forma de simplificar la adición de muchas notas guía orquestales a la reducción de piano en una partitura vocal. La función musical `\cueWhile` toma cuatro argumentos: la música de la que se toma la cita, como viene definida por `\addQuote`, el nombre que insertar antes de las notas guía, y después `UP` o `DOWN` para especificar `\voiceOne` con el nombre encima del pentagrama o bien `\voiceTwo` con el nombre debajo del pentagrama, y finalmente la música de piano con la que las notas guía deben aparecer en paralelo. El nombre del instrumento citado se posiciona a la izquierda de las notas guía. Se pueden citar muchos pasajes como guía, pero no se pueden superponer en el tiempo entre ellos.

```
cueWhile =
#(define-music-function
   (instrument name dir music)
   (string? string? ly:dir? ly:music?)
   #{
     \cueDuring $instrument #dir {
       \once \override TextScript.self-alignment-X = #RIGHT
```

```
      \once \override TextScript.direction = $dir
      <>-\markup { \tiny #name }
      $music
    }
  #})
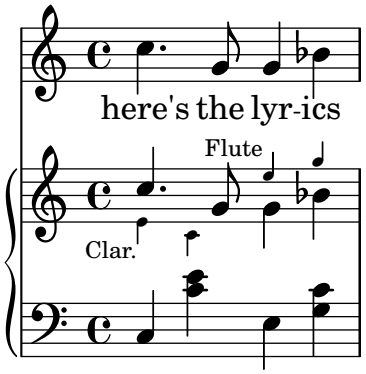
flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```
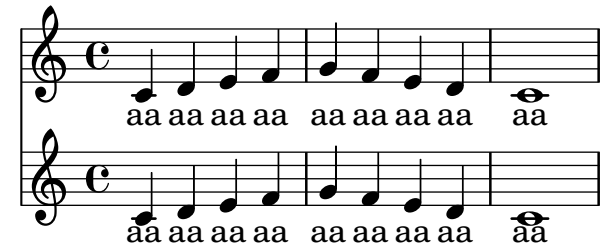


Adjusting vertical spacing of lyrics

This snippet shows how to bring the lyrics line closer to the staff.

```
music = \relative c' { c4 d e f | g4 f e d | c1 }
text = \lyricmode { aa aa aa aa aa aa aa aa aa }

<<
  \new Staff \new Voice = melody \music
  % Default layout:
  \new Lyrics \lyricsto melody \text

  \new Staff \new Voice = melody \music
  % Reducing the minimum space below the staff and above the lyrics.
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  } \lyricsto melody \text
>>
```
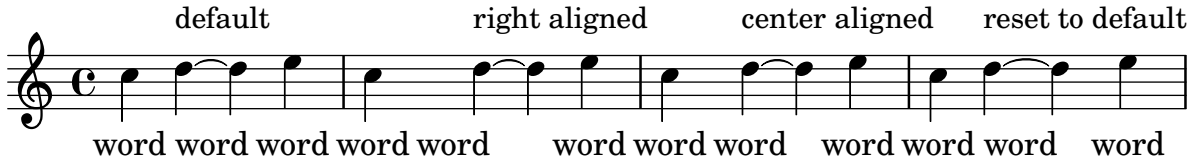


Alineación de sílabas con melisma

De forma predeterminada, las sílabas de la letra que comienzan un melisma se alinean a la izquierda sobre su nota correspondiente. Se puede alterar la alineación usando la propiedad `lyricMelismaAlignment`.

```
<<
  \new Staff {
    \new Voice = "vocal" \relative c'' {
      \override TextScript.staff-padding = #2
      c d~^\markup default  d e
      c d~^\markup "right aligned" d e
      c d~^\markup "center aligned" d e
      c d~^\markup "reset to default" d e
    }
```

```
  }
  \new Lyrics \lyricsto "vocal" {
    word word word
    \set lyricMelismaAlignment = #RIGHT
    word word word
    \set lyricMelismaAlignment = #CENTER
    word word word
    \unset lyricMelismaAlignment
    word word word
  }
>>

\layout {
  ragged-right = ##f
}
```
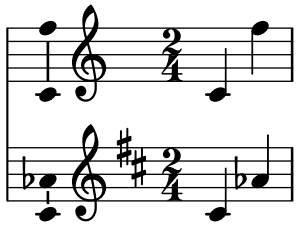


Indicaciones de tesitura

Las indicaciones de ámbito o tesitura indican rangos de alturas para las voces.

Las alteraciones accidentales solo se muestran si no forman parte de la armadura de tonalidad. Los objetos gráficos `AmbitusNoteHead` también tienen líneas adicionales.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

<<
  \new Staff {
    \relative c' {
      \time 2/4
      c4 f'
    }
  }
  \new Staff {
    \relative c' {
      \time  2/4
      \key d \major
      cis4 as'
    }
  }
>>
```
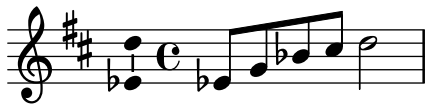


Indicación de tesitura después de la armadura

De manera predeterminada, las indicaciones de ámbito o tesitura se sitúan a la izquierda de la clave. La función `\ambitusAfter` permite cambiar esta colocación. La sintaxis es `\ambitusAfter` *`grob-interface`* (consulte Interfaces de los objetos gráficos (`https://lilypond.org/doc/v2.24/Documentation/internals/graphical-object-interfaces`) para ver una lista de los valores posibles para *grob-interface*.)

Un caso de utilización bastante común es imprimir la indicación de texitura entre la armadura de la clave y la indicación de compás.
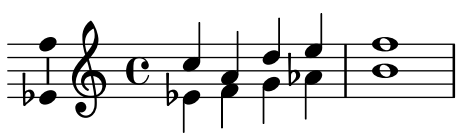
```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



Ámbitos con varias voces

La adición del grabador `Ambitus_engraver` al contexto de `Staff` crea un solo ámbito por pentagrama, incluso en el caso de pentagramas con varias voces.

```
\new Staff \with {
  \consists "Ambitus_engraver"
  }
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



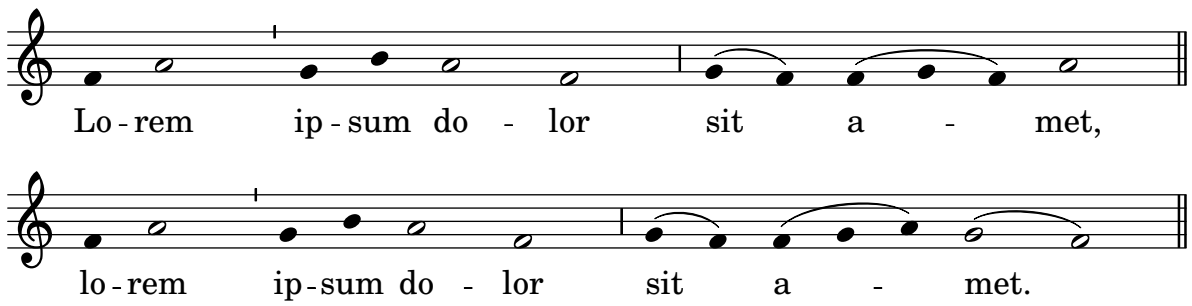
Ancient notation template – modern transcription of Gregorian music

This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter note heads, and special marks, indicating rests of different length.

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g f) a2 \finalis \break
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g a) g2( f) \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met,
  lo -- rem ip -- sum do -- lor sit a -- met.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" \chant
    \new GregorianTranscriptionLyrics = "one" \lyricsto melody \verba
  >>
}
```



Plantilla de salmo del canto anglicano

Esta plantilla muestra una forma de preparar un cántico salmódico anglicano. También muestra cómo se pueden añadir estrofas adicionales como texto independiente por debajo de la música. Las dos estrofas se codifican en estilos diferentes para ilustrar más posibilidades.

```
SopranoMusic = \relative g' {
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

TenorMusic = \relative a {
```

```
  c1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

BassMusic =  \relative c {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}

tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}

% Use markup to center the chant on the page
\markup \fill-line {
  \score { % centered
    \new ChoirStaff <<
      \new Staff <<
        \global
        \clef "treble"
        \new Voice = "Soprano" <<
          \voiceOne
          \SopranoMusic
        >>
        \new Voice = "Alto" <<
          \voiceTwo
          \AltoMusic
        >>
      >>

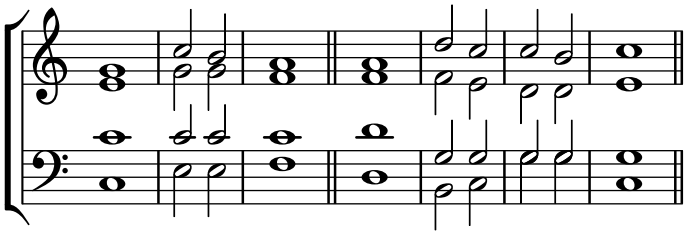
      \new Staff <<
        \clef "bass"
        \global
        \new Voice = "Tenor" <<
          \voiceOne
          \TenorMusic
        >>
        \new Voice = "Bass" <<
          \voiceTwo
          \BassMusic
        >>
      >>
    >>
```

```
    \layout {
      \context {
        \Score
        \override SpacingSpanner.base-shortest-duration =
                    \musicLength 2
      }
      \context {
        \Staff
        \remove "Time_signature_engraver"
      }
    }
  }  % End score
}  % End markup

\markup \fill-line {
  \left-column {
    \null \null \null
    \line { \fontsize #5 O
            \fontsize #3 come
            let us \bold sing | unto \dot the | Lord : let }
    \line { us heartily \concat { re \bold joice }
            \italic \bold in the | strength of | our }
    \line { sal | vation. }

    \null

    \line { \hspace #2.5 8. Today if ye will hear his voice * }
    \line { \concat { \bold hard en }
            \tick not your \tick hearts : as \italic \bold in the pro- }
    \line { vocation * and as \italic \bold in the \bold day of tempt- \tick }
    \line { -ation \tick \italic \bold in the \tick wilderness. }
  }
}
```



O come let us **sing** | unto • the | Lord : let us heartily re**joice** in the | strength of | our sal | vation.

8. Today if ye will hear his voice * **hard**en ' not your ' hearts : as in the provocation * and as in the **day** of tempt- ' -ation ' in the ' wilderness.

Disposición de distintas letras en la misma línea

A veces queremos poner la letra que corresponde a distintos personajes sobre una sola línea, por ejemplo allí donde el texto alterna rápidamente. Este fragmento muestra cómo puede hacerse ajustando la propiedad `nonstaff-nonstaff-spacing` del objeto gráfico `VerticalAxisGroup`.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
              .nonstaff-nonstaff-spacing
              .minimum-distance = ##f
  }
}

aliceSings = \markup { \smallCaps "Alice" }
eveSings = \markup { \smallCaps "Eve" }

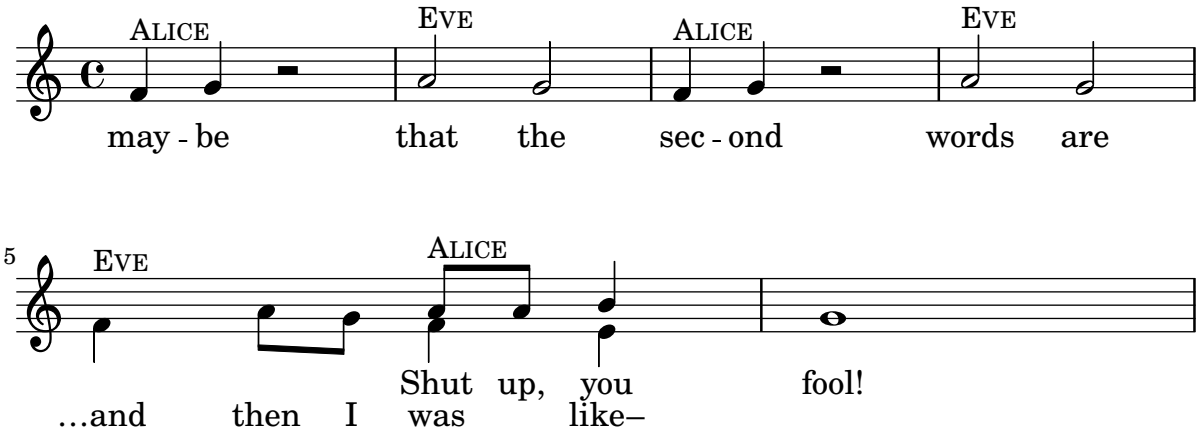
<<
  \new Staff <<
    \new Voice = "alice" {
      f'4^\aliceSings g' r2 |
      s1 |
      f'4^\aliceSings g' r2 |
      s1 | \break
      % ...

      \voiceOne
      s2 a'8^\aliceSings a' b'4 |
      \oneVoice
      g'1
    }
    \new Voice = "eve" {
      s1 |
      a'2^\eveSings g' |
      s1 |
      a'2^\eveSings g'
      % ...

      \voiceTwo
      f'4^\eveSings a'8 g' f'4 e' |
      \oneVoice
      s1
    }
  >>

  \new Lyrics \lyricsto "alice" {
    may -- be
    sec -- ond
    % ...
    Shut up, you fool!
  }
```

```
  \new Lyrics \lyricsto "eve" {
    that the
    words are
    % ...
    ...and then I was like--
  }
>>
```



Modificar la tipografía para cada estrofa

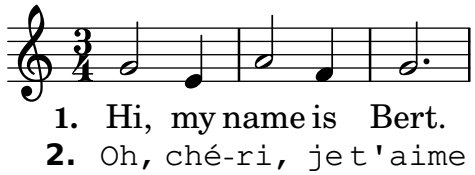
Se pueden cambiar las fuentes tipográficas de forma independiente para cada estrofa, incuida la fuente que se usa para imprimir el número de estrofa.

```
%{
You may have to install additional fonts.

Red Hat Fedora: dejavu-fonts-all

Debian GNU/Linux, Ubuntu: fonts-dejavu-core
                          fonts-dejavu-extra
%}

\relative c'' {
  \time 3/4
  g2 e4
  a2 f4
  g2.
}
\addlyrics {
  \stanza "1. "
  Hi, my name is Bert.
}
\addlyrics {
  \override StanzaNumber.fonts.serif = "DejaVu Sans"
  \stanza "2. "
  \override LyricText.font-family = #'typewriter
  Oh, ché -- ri, je t'aime
}
```

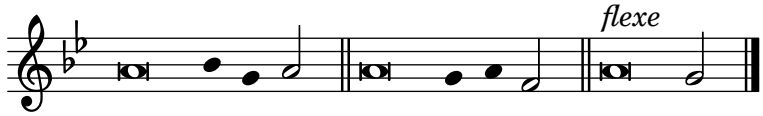


Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

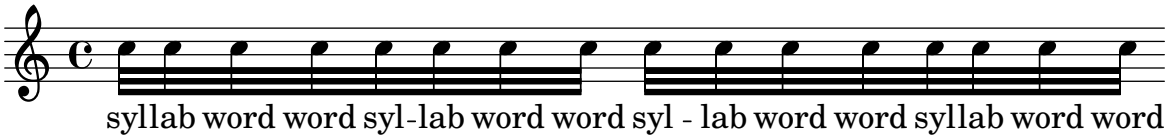
\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}
```



Forzar la visibilidad de los guiones separadores de sílabas

Si LilyPond no cree que haya sitio suficiente para un guión separador de sílabas, lo omitirá. Se puede sobreescribir este comportamiento con la propiedad `minimum-distance` de `LyricHyphen`.

```
\relative c'' {
  c32 c c c
  c32 c c c
  c32 c c c
  c32 c c c
}
\addlyrics {
  syl -- lab word word
  \override LyricHyphen.minimum-distance = #1.0
  syl -- lab word word
  \override LyricHyphen.minimum-distance = #2.0
  syl -- lab word word
  \revert LyricHyphen.minimum-distance
  syl -- lab word word
}
```

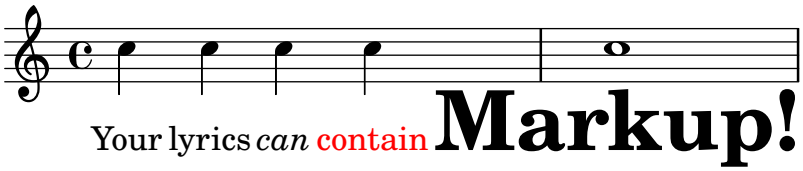


Dar formato a sílabas de la letra

Es posible usar la instrucción \markup dentro de un bloque \lyricmode para dar formato a las sílabas individuales dentro de la letra.

```
mel = \relative c'' { c4 c c c c1 }
lyr = \lyricmode {
  Your lyrics \markup { \italic can }
  \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

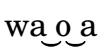
<<
  \new Voice = "melody" \mel
  \new Lyrics \lyricsto "melody" \lyr
>>
```



Cómo poner ligaduras entre sílabas de la letra

Se puede hacer separando las sílabas mediante símbolos de tilde curva.

```
\lyrics {
  wa~o~a
}
```



Plantilla para himnos

Este fragmento de código muestra una forma de preparar un himno en el que cada línea comienza con un compás parcial. También muestra cómo añadir los versos como texto independiente debajo de la música.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \caesura \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \fine
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}
```

```
AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
 \key g \major
}

\score {  % Start score
  \new PianoStaff <<  % Start pianostaff
    \new Staff <<  % Start Staff = RH
      \global
      \clef "treble"
      \new Voice = "Soprano" <<  % Start Voice = "Soprano"
        \Timeline
        \voiceOne
        \SopranoMusic
      >>  % End Voice = "Soprano"
      \new Voice = "Alto" <<  % Start Voice = "Alto"
        \Timeline
        \voiceTwo
        \AltoMusic
      >>  % End Voice = "Alto"
    >>  % End Staff = RH

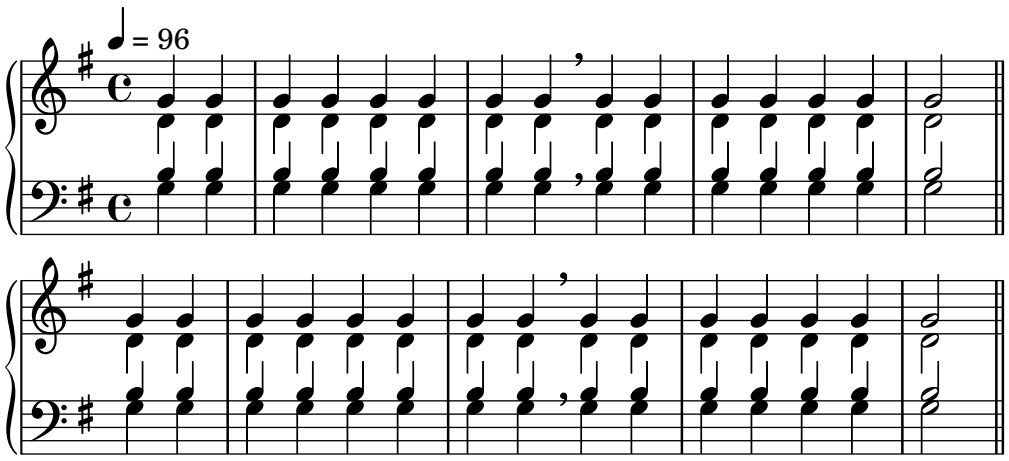
    \new Staff <<  % Start Staff = LH
      \global
      \clef "bass"
      \new Voice = "Tenor" <<  % Start Voice = "Tenor"
        \Timeline
        \voiceOne
        \TenorMusic
      >>  % End Voice = "Tenor"
      \new Voice = "Bass" <<  % Start Voice = "Bass"
        \Timeline
        \voiceTwo
        \BassMusic
      >>  % End Voice = "Bass"
    >>  % End Staff = LH
  >>  % End pianostaff
```

```
}  % End score

\markup \fill-line {
  \left-column {
    "This is line one of the first verse"
    "This is line two of the same"
    \null
    "And here's line one of the second verse"
    "And the next line of the same"
  }
}

\layout {
  \context {
    \Score
    caesuraType = #'((bar-line . "||"))
    fineBarType = "||"
  }
}

\paper {  % Start paper block
  indent = 0          % don't indent first system
  line-width = 130    % shorten line length to suit music
  tagline = ##f       % Don't print tag line, can be removed
}  % End paper block
```



This is line one of the first verse
This is line two of the same

And here's line one of the second verse
And the next line of the same
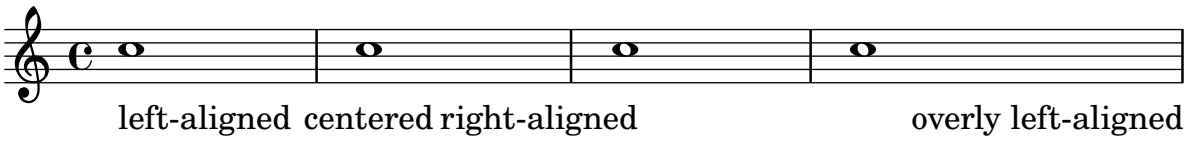
Alineación de la letra

La alineación horizontal de la letra se puede ajustar sobreescribiendo la propiedad `self-alignment-X` del objeto `LyricText`. El valor -1 significa alineado por la izquierda, 0 es centrado y 1 es derecha; de forma alternativa también podemos usar los valores de Scheme `#LEFT`, `#CENTER` y `#RIGHT` en lugar de números. También son posibles otros valores numéricos. Importante: no olvide añadir el prfijo de Scheme '#' para los números negativos.

```
\layout {
  ragged-right = ##f
}

\relative c'' {
  c1 c c c
}

\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "centered"
  \once \override LyricText.self-alignment-X = 1
  "right-aligned"
  \once \override LyricText.self-alignment-X = #-1.5
  "overly left-aligned"
}
```



Marcar notas de las partes habladas con unas aspas en la plica
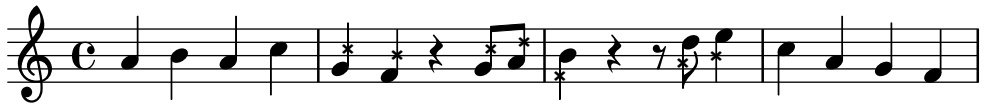
Este ejemplo muestra cómo añadir aspas sobre las plicas. Marcamos el comienzo de una sección hablada con la instrucción \speakOn, y el final con \speakOff.

```
speakOn = \override Stem.stencil =
  #(lambda (grob)
     (let* ((x-parent (ly:grob-parent grob X))
            (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
       (if is-rest?
           empty-stencil
           (ly:stencil-combine-at-edge
            (ly:stem::print grob)
            Y
            (- (ly:grob-property grob 'direction))
            (grob-interpret-markup
     grob
             (markup #:center-align #:fontsize -4
                     #:musicglyph "noteheads.s2cross"))
            -1.7))))

speakOff = \revert Stem.stencil

\new Staff {
  \relative c'' {
    a4 b a c
    \speakOn
    g4 f r g8 a
    b4 r r8 d e4
```

```
    \speakOff
    c4 a g f
  }
}
```



Plantilla de orquesta, coro y piano

Esta plantilla muestra el uso de contextos `StaffGroup` y `GrandStaff` anidados para sub-agrupar instrumentos del mismo tipo, y una forma de usar `\transpose` de manera que unas variables contengan la música para instrumentos transpositores en afinación de concierto.

```
#(set-global-staff-size 17)

\paper {
  indent = 3.0\cm  % add space for instrumentName
  short-indent = 1.5\cm  % add less space for shortInstrumentName
}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.
clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns
hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }
sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }
altoILyrics = \sopranoLyrics
altoIIMusic = \relative c' { \key g \major g'1 b }
altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }
tenorLyrics = \sopranoLyrics

pianoRHMusic = \relative c { \key g \major g''1 b }
pianoLHMusic = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }
violinIIMusic = \relative c' { \key g \major g'1 b }
```

```
violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\book {
  \score {
    <<
      \new StaffGroup = "StaffGroup_woodwinds" <<
        \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
          \fluteMusic

        \new Staff = "Staff_clarinet" \with {
          instrumentName = \markup { \concat { "Clarinet in B" \flat } }
        }
          % Declare that written Middle C in the music
          % to follow sounds a concert B flat, for
          % output using sounded pitches such as MIDI.
          %\transposition bes

          % Print music for a B-flat clarinet
          \transpose bes c' \clarinetMusic
      >>

      \new StaffGroup = "StaffGroup_brass" <<
        \new Staff = "Staff_hornI" \with {
          instrumentName = "Horn in F"
        }
          % \transposition f
          \transpose f c' \hornMusic

        \new Staff = "Staff_trumpet" \with {
          instrumentName = "Trumpet in  C"
        }
          \trumpetMusic
      >>

      \new RhythmicStaff = "RhythmicStaff_percussion" \with {
        instrumentName = "Percussion"
      }
        \percussionMusic

      \new PianoStaff \with {
        instrumentName = "Piano"
      } <<
        \new Staff { \pianoRHMusic }
        \new Staff { \pianoLHMusic }
      >>

      \new ChoirStaff = "ChoirStaff_choir" <<
```

```
  \new Staff = "Staff_soprano" \with {
    instrumentName = "Soprano"
  }
    \new Voice = "soprano" \sopranoMusic
    \new Lyrics \lyricsto "soprano" { \sopranoLyrics }

  \new GrandStaff = "GrandStaff_altos" \with {
    \accepts Lyrics
  } <<
    \new Staff = "Staff_altoI" \with {
      instrumentName = "Alto I"
    }
      \new Voice = "altoI"
      \altoIMusic
      \new Lyrics \lyricsto "altoI" { \altoILyrics }
    \new Staff = "Staff_altoII" \with {
      instrumentName = "Alto II"
    }
      \new Voice = "altoII"
      \altoIIMusic
      \new Lyrics \lyricsto "altoII" { \altoIILyrics }
  >>

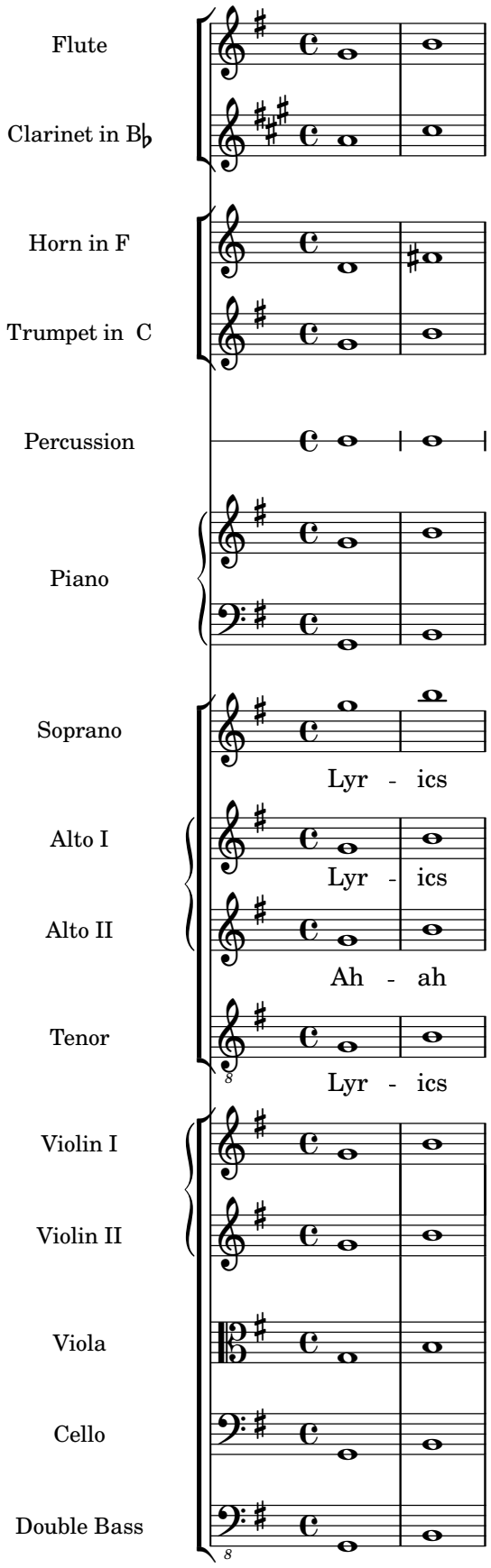
  \new Staff = "Staff_tenor" \with {
    instrumentName = "Tenor"
  }
    \new Voice = "tenor" \tenorMusic
    \new Lyrics \lyricsto "tenor" { \tenorLyrics }
>>

\new StaffGroup = "StaffGroup_strings" <<
  \new GrandStaff = "GrandStaff_violins" <<
    \new Staff = "Staff_violinI" \with {
      instrumentName = "Violin I"
    }
      \violinIMusic
    \new Staff = "Staff_violinII" \with {
      instrumentName = "Violin II"
    }
      \violinIIMusic
  >>

  \new Staff = "Staff_viola" \with {
    instrumentName = "Viola"
  }
    \violaMusic

  \new Staff = "Staff_cello" \with {
    instrumentName = "Cello"
  }
    \celloMusic
```

```
        \new Staff = "Staff_bass" \with {
          instrumentName = "Double Bass"
        }
          \bassMusic
      >>
    >>
  }
}
```


Flute
Clarinet in B♭
Horn in F
Trumpet in C
Percussion
Piano
Soprano
Lyr - ics
Alto I
Lyr - ics
Alto II
Ah - ah
Tenor
Lyr - ics
Violin I
Violin II
Viola
Cello
Double Bass

Plantilla de piano con melodía y letra

He aquí el típico formato dde una canción: un pentagrama con la melodía y la letra, y el acompañamiento de piano por debajo.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

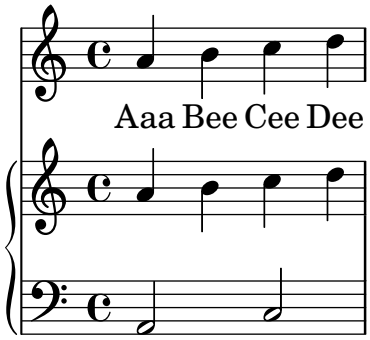
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}
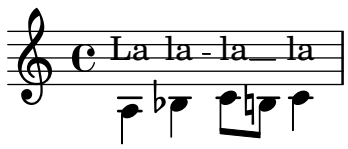
\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```



Putting lyrics inside the staff

Se pueden mover las líneas de letra verticalmente para imprimirlas dentro del pentagrama. Las letras se mueven con `\override LyricText.extra-offset = #'(0 . dy)`, y hay instrucciones similares para mover las líneas extensoras y los guiones. El mejor valor para el desplazamiento *dy* se debe buscar mediante un proceso de ensayo y error.

```
<<
  \new Staff <<
    \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
  >>
  \new Lyrics \with {
    \override LyricText.extra-offset = #'(0 . 8.6)
    \override LyricExtender.extra-offset = #'(0 . 8.6)
    \override LyricHyphen.extra-offset = #'(0 . 8.6)
  } \lyricsto "voc" { La la -- la __ _ la }
>>
```
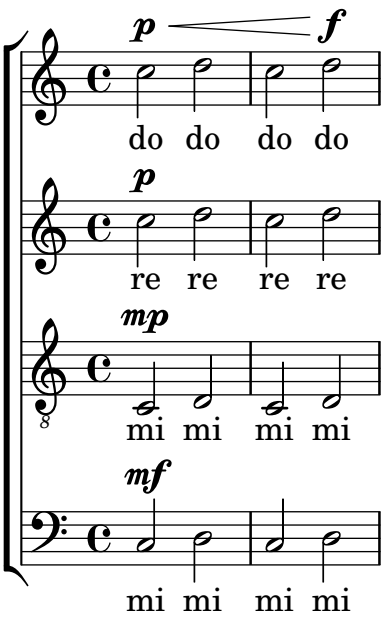


Plantilla de coro SATB, a cuatro pentagramas

He aquí una plantilla de coro SATB en cuatro pentagramas.

```
global = {
  \key c \major
  \time 4/4
  \dynamicUp
}
sopranonotes = \relative c'' {
  c2 \p \< d c d \f
}
sopranowords = \lyricmode { do do do do }
altonotes = \relative c'' {
  c2\p d c d
}
altowords = \lyricmode { re re re re }
tenornotes = {
  \clef "G_8"
  c2\mp d c d
```

```
}
tenorwords = \lyricmode { mi mi mi mi }
bassnotes = {
  \clef bass
  c2\mf d c d
}
basswords = \lyricmode { mi mi mi mi }

\score {
  \new ChoirStaff <<
    \new Staff <<
      \new Voice = "soprano" <<
        \global
        \sopranonotes
      >>
      \new Lyrics \lyricsto "soprano" \sopranowords
    >>
    \new Staff <<
      \new Voice = "alto" <<
        \global
        \altonotes
      >>
      \new Lyrics \lyricsto "alto" \altowords
    >>
    \new Staff <<
      \new Voice = "tenor" <<
        \global
        \tenornotes
      >>
      \new Lyrics \lyricsto "tenor" \tenorwords
    >>
    \new Staff <<
      \new Voice = "bass" <<
        \global
        \bassnotes
      >>
      \new Lyrics \lyricsto "bass" \basswords
    >>
  >>
}
```



Plantilla de pentagrama único con música, letra y acordes

Esta plantilla facilita la preparación de una canción con melodía, letra y acordes.

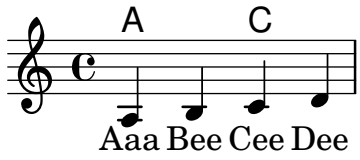
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Single-staff template with notes, lyrics, chords, and frets

Here is a simple lead sheet template with melody, lyrics, chords, and fret diagrams.

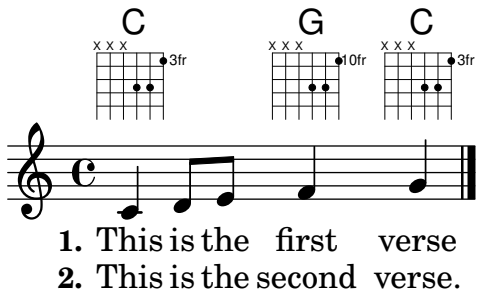
```
verseI = \lyricmode {
  \stanza "1."
  This is the first verse
}

verseII = \lyricmode {
  \stanza "2."
  This is the second verse.
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
  c2 g4 c
}

staffMelody = \relative c' {
   \key c \major
   \clef treble
   % Type notes for melody here
   c4 d8 e f4 g
   \bar "|."
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}
```



Plantilla de pentagrama único don notas y letra
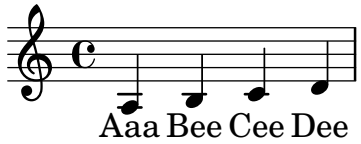
Esta pequeña plantilla muestra una melodía sencilla con letra. Córtela y péguela, escriba las notas y luego la letra. Este ejemplo desactiva el barrado automático, que es lo más frecuente en las partes vocales antiguas. Para usar el barrado automático modifique o marque como un comentario la línea correspondiente.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

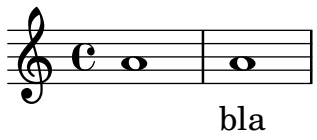
\score{
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Silencios de separación en modo de letra

La sintaxis ‘s’ para los silencios de desplazamiento solamente está disponible en los modos de nota y de acorde. En otras situaciones, por ejemplo al escribir la letra de las canciones, se recomienda usar la instrucción `\skip`.

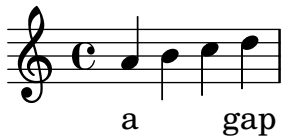
```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip1 bla1 }
>>
```



Silencios de separación en modo de letra (2)

Aunque no se pueden usar los silencios de separación 's' dentro de `\lyricmode` (se toman como una "s", literal, no como un espacio), sí están disponibles las comillas dobles (`""`) y el carácter subrayado (`_`).

```
<<
  \relative c'' { a4 b c d }
  \new Lyrics \lyricmode { a4 "" _ gap }
>>
```



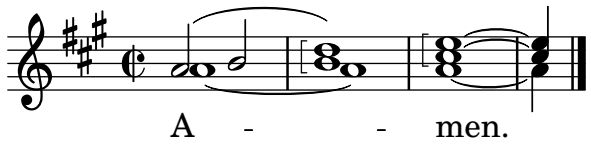
Using \arpeggioBracket to make divisi more visible

The `\arpeggioBracket` command can be used to indicate the division of voices where there are no stems to provide the information. This is often seen in choral music.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper"
      <<
        { \voiceOne \arpeggioBracket
          a2( b2
          <b d>1\arpeggio)
          <cs e>\arpeggio ~
          <cs e>4
        }
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower"
      { \voiceTwo
        a1 ~
        a
        a ~
        a4 \bar "|."
      }
```

```
    >>
  }
}
```

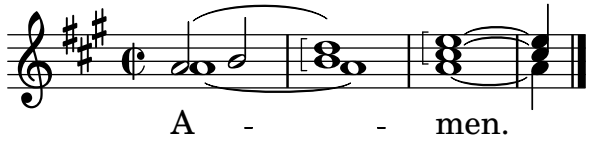


Using a bracket to clarify divisi

The \nonArpeggiato command can be used to indicate the division of voices where there are no stems to provide the information. This is often seen in choral music.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper" <<
        {
          \voiceOne
          a2( b2
          <b d>1\nonArpeggiato)
          <cs e>\nonArpeggiato ~
          <cs e>4
          \fine
        }
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower" {
        \voiceTwo
        a1 ~
        a
        a ~
        a4
        \fine
      }
    >>
  }
}
```



Uso de etiquetas para producir música mensural y moderna a partir del mismo código fuente

Usando etiquetas es posible producir las dos notaciones mensural y moderna a partir del mismo código fuente de la música. En este fragmento se introduce la función \menrest, que permite

que los silencios mensurales estén a la altura de notas como en el original, pero con silencios modetnos en la posicion del pentagrama estándar.

También se pueden usar etiquetas allí donde se necesiten otras diferencias: por ejemplo usando "silencios de compás completo" (`R1`, `R\breve`, etc.) en música moderna, pero silencios normales (`r1`, `r\breve`, etc.) en la versión mensural. La conversión de música mensural a su equivalente moderno se conoce normalmente como *transcription*.

La llamara `c4.\Be c8 c\Am` es lo mismo que `c4.[ c8 c]`. Sin embargo, suprime las advertencias sin se inicia en una nota que no puede llevar barra pero la necesita en todo caso debido al uso del grabador `Completion_heads_engraver`.

[La longitud ligeramente acortada de la línea en la pauta mensural evita el recorte del glifo de los custos cuando LilyPond genera imágenes recortadas muy estrechamente. El uso de `\with-true-dimensions`, tal y como aparece a continuación, evita esto.]

```
\layout {
  line-width = 150\mm
}

menrest = #(define-music-function (note) (ly:music?)
              #{
                \tag #'mens $(make-music 'RestEvent note)
                \tag #'mod $(make-music 'RestEvent note 'pitch '())
              #})
Be = \tag #'mod
       #(begin
          (ly:expect-warning (G_ "stem does not fit in beam"))
          (ly:expect-warning (G_ "beam was started here"))
          (make-span-event 'BeamEvent START))
Am = \tag #'mod ]

MenStyle = {
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
  \omit Slur
  \omit Beam
}

finalis = \section

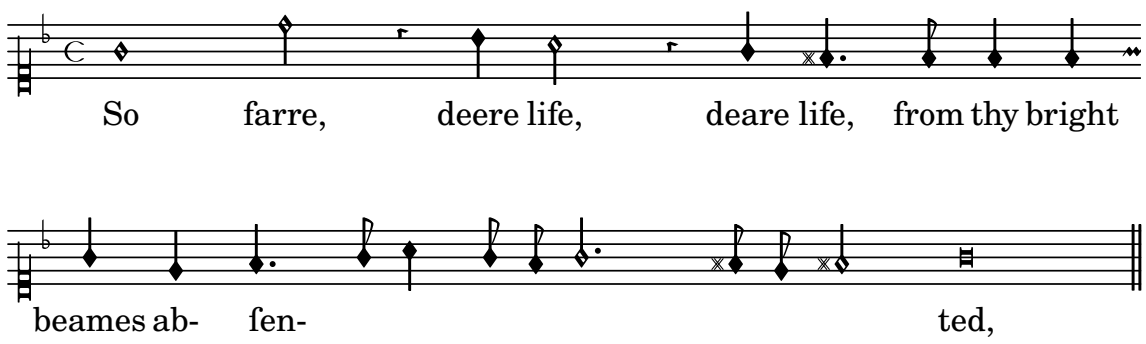
Music = \relative c'' {
  \key f \major
  g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
  g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}
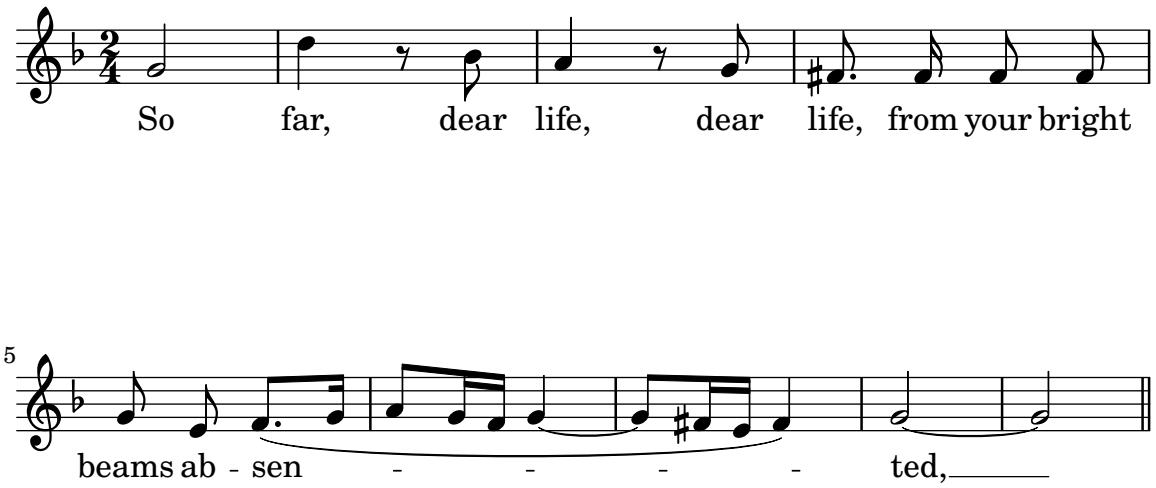
MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}
```

```
\score {
  \keepWithTag #'mens {
    <<
      \new PetrucciStaff {
        \new PetrucciVoice = "Cantus" {
          \clef "petrucci-c1" \time 4/4 \MenStyle \Music
        }
      }
      \new Lyrics \lyricsto "Cantus" \MenLyr
    >>
  }
  \layout {
    \context {
      \PetrucciVoice
      % No longer necessary starting with version 2.25.23.
      \override Flag.style = #'mensural
    }
  }
}

\markup\vspace #1

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff {
        \new Voice = "Sop" \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver"
        } \shiftDurations 1 0 { \time 2/4 \autoBeamOff \Music }
      }
      \new Lyrics \lyricsto "Sop" \ModLyr
    >>
  }
}
```



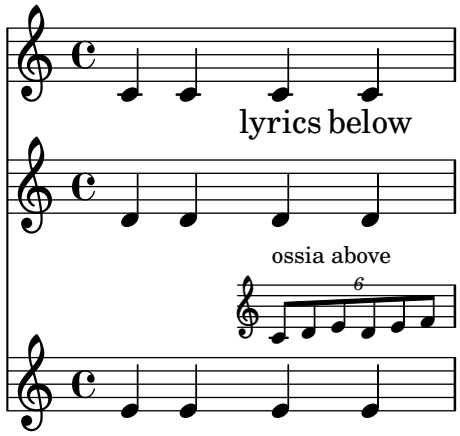


Alineación vertical de la letra y los compases de ossia

Este fragmento de código muestra el uso de las propiedades de contexto `alignBelowContext` y `alignAboveContext` para controlar la posición de la letra y los compases de ossia.

```
\relative c' <<
  \new Staff = "1" { c4 c c c }
  \new Staff = "2" { d4 d d d }
  \new Staff = "3" { e4 e e e }

  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = "1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = "3"
        fontSize = -2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
        \override VerticalAxisGroup.staff-staff-spacing =
           #'((minimum-distance . 0)
              (basic-distance . 0)
              (padding . 1))
      } {
        \tuplet 6/4 {
          \override TextScript.padding = 2
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



Alineación vertical de los números de estrofa de diferentes pentagramas

Puede ocurrir que los números de estrofa no estén alineados verticalmente si los versos están adjuntos a distinto pentagrama. Para corregirlo, seobreescriba la propiedad `self-alignment-X` del objeto gráfico `LyricText`.

```
\markup { default behavior }

<<
  \new Staff { b b b b }
  \lyrics {
    \stanza "3."
    a a a a
  }

  \new Staff { b b b b }
  \lyrics {
    \stanza "1."
    aaaaaaaaaa a a a
  }
  \lyrics {
    \stanza "2."
    a a a a
  }
>>

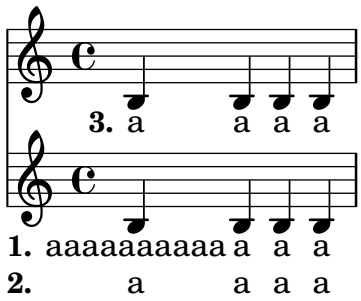
\markup \vspace #1
\markup {
  using \typewriter "self-alignment-X = #LEFT" }

<<
  \new Staff { b b b b }
  \new Lyrics \lyricmode {
    \stanza "3."
    a a a a
  }

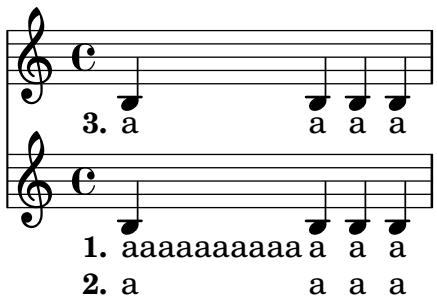
  \new Staff { b b b b }
```

```
  \new Lyrics \lyricmode {
    \stanza "1."
    \once \override LyricText.self-alignment-X = #LEFT
    aaaaaaaaaa a a a
  }
  \new Lyrics \lyricmode {
    \stanza "2."
    a a a a
  }
>>
```

default behavior



using `self-alignment-X = #LEFT`



Letra común centrada verticalmente

En una pieza vocal en la que hay varias líneas de letra (dos, cuatro o más) y hay letras comunes a todas las voces en algún punto, se puede hacer que las letras comunes se centren verticalmente como se muestra en el siguiente ejemplo:

```
dropLyrics = {
  \override LyricText.extra-offset = #'(0 . -4.5)
  \override LyricHyphen.extra-offset = #'(0 . -4.5)
  \override LyricExtender.extra-offset = #'(0 . -4.5)
  \override StanzaNumber.extra-offset = #'(0 . -4.5)
}

raiseLyrics = {
  \revert LyricText.extra-offset
  \revert LyricHyphen.extra-offset
  \revert LyricExtender.extra-offset
  \revert StanzaNumber.extra-offset
```

```
}

skipFour = \*4 { \skip 8 }

lyricsA = \lyricmode {
  The first verse has
  \dropLyrics
  \stanza "   All:"
  the com -- mon __ words
  \raiseLyrics
  used in all four.
}
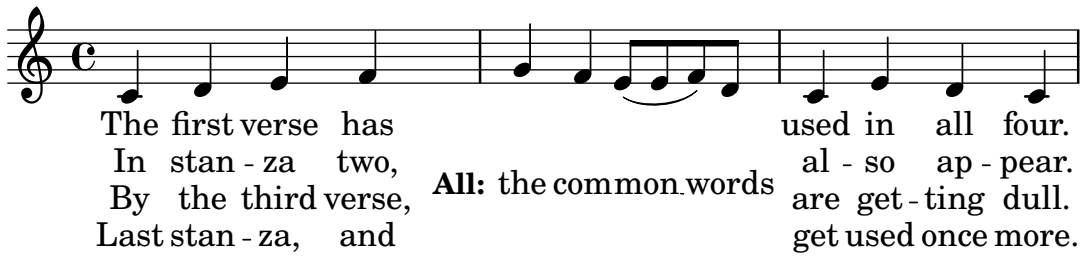
lyricsB = \lyricmode { In stan -- za two,   \skipFour al -- so ap -- pear. }

lyricsC = \lyricmode { By the third verse,  \skipFour are get -- ting dull. }

lyricsD = \lyricmode { Last stan -- za, and \skipFour get used once more. }

melody = \relative c' {
  c4 d e f |
  g f e8( e f) d |
  c4 e d c |
}

\score {
  <<
    \new Voice = m \melody
    \new Lyrics \lyricsto m \lyricsA
    \new Lyrics \lyricsto m \lyricsB
    \new Lyrics \lyricsto m \lyricsC
    \new Lyrics \lyricsto m \lyricsD
  >>
}
```



Plantilla de conjunto vocal

He aquí una partitura vocal estándar para cuatro voces SATB. Con grupos mayores, suele ser útil incluir una sección que aparezca en todas las partes. Por ejemplo, el compás y la armadura casi siempre son los mismos para todas. Como en la "Plantilla para himnos", las cuatro voces se reagrupan en solo dos pentagramas.

```
\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
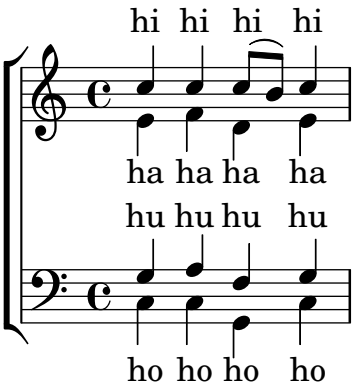
altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Lyrics = "sopranos" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "women" <<
      \new Voice = "sopranos" {
        \voiceOne
        << \global \sopMusic >>
      }
      \new Voice = "altos" {
```

```
        \voiceTwo
        << \global \altoMusic >>
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        << \global \tenorMusic >>
      }
      \new Voice = "basses" {
        \voiceTwo << \global \bassMusic >>
      }
    >>
    \new Lyrics = "basses"
    \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
    \context Lyrics = "altos" \lyricsto "altos" \altoWords
    \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
    \context Lyrics = "basses" \lyricsto "basses" \bassWords
  >>
}
```



Plantilla de conjunto vocal con reducción de piano automática

Esta plantilla añade una reducción de piano automática a la partitura vocal SATB estándar que se mostró en el fragmento de código “Plantilla de conjunto vocal”. Presenta uno de los puntos fuertes de LilyPond: podemos usar una definición de música más de una vez. Si se hace cualquier cambio en las notas de la parte vocal (digamos `tenorMusic`), entonces los cambios se aplican también a la reducción de piano.

```
\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}
```

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords =\lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

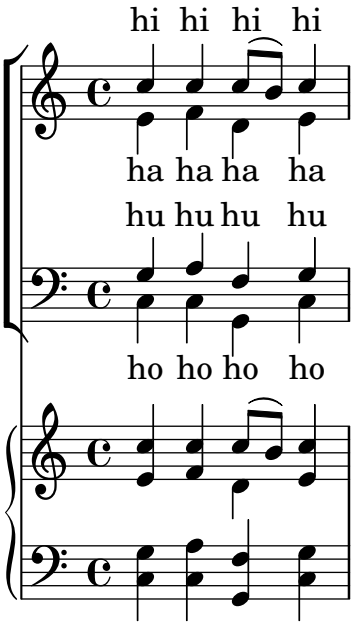
\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"

      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "men" <<
        \clef bass
```

```
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"

      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>

    \new PianoStaff <<
      \new Staff <<
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \sopMusic >>
        << \global \altoMusic >>
      >>
      \new Staff <<
        \clef bass
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \tenorMusic >>
        << \global \bassMusic >>
      >>
    >>
  >>
}
```



Plantilla para conjunto vocal con letras alineadas encima y debajo de los pentagramas

Esta plantilla es, básicamente, la misma que la sencilla "Plantilla para conjunto vocal", excepto que aquí todas las líneas de letra se colocan utilizando `alignAboveContext` y `alignBelowContext`.

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
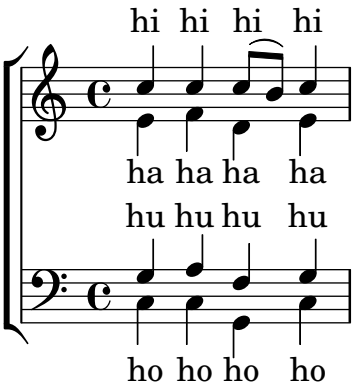
altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative c {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = "women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords
```

```
    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = "men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
    % \new Lyrics \lyricsto "basses" \bassWords
  >>
}
```



Estrofa para solista y estribillo a dos voces

Esta plantilla crea una partitura que comienza con una estrofa para solista y continúa con un estribillo a dos voces. también muestra el uso de silencios de separación dentro de la variable \global para definir cambios de compás (y otros elementos que son comunes a todas las partes) a lo largo de toda la partitura.

```
global = {
  \key g \major

  % verse
  \time 3/4
  s2.*2
  \break

  % refrain
  \time 2/4
  s2*2
  \bar "|."
}

SoloNotes = \relative g' {
  \clef "treble"

  % verse
  g4 g g |
  b4 b b |
```

```
  % refrain
  R2*2 |
}

SoloLyrics = \lyricmode {
  One two three |
  four five six |
}

SopranoNotes = \relative c'' {
  \clef "treble"

  % verse
  R2.*2 |

  % refrain
  c4 c |
  g4 g |
}

SopranoLyrics = \lyricmode {
  la la |
  la la |
}

BassNotes = \relative c {
  \clef "bass"

  % verse
  R2.*2 |

  % refrain
  c4 e |
  d4 d |
}

BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}
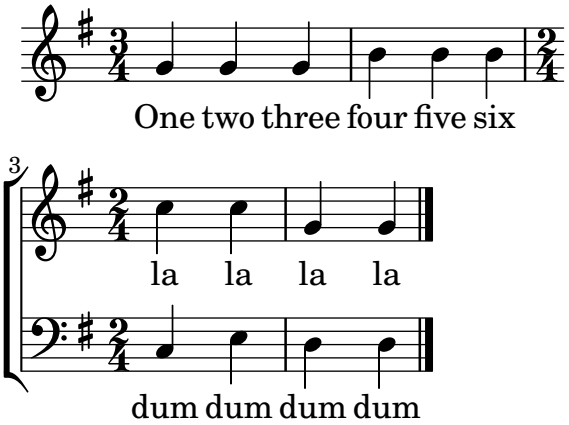
\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

      \new Voice = "BassVoice" << \global \BassNotes >>
      \new Lyrics \lyricsto "BassVoice" \BassLyrics
```

```
    >>
  >>

  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}
```



10 Keyboard and other multi-staff instruments

See also Sección “Keyboard and other multi-staff instruments” en *Notation Reference*.

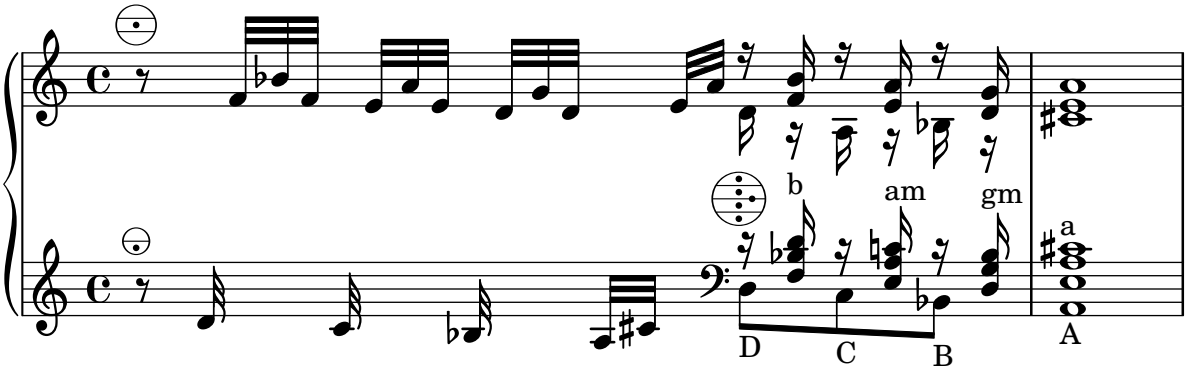
Símbolos de registración de acordeón

Los símbolos de registración de acordeón están disponibles como elementos de `\markup` y como eventos musicales autocontenidos (pues los cambios de registro tienden a ocurrir entre eventos de música). Los registros del bajo no están demasiado estandarizados. Las instrucciones existentes están disponibles en el apartado ‘Símbolos de acordeón’ en la Referencia de la notación (`https://lilypond.org/doc/v2.24/Documentation/notation/accordion#discant-symbols`).

```
#(use-modules (lily accreg))

\new PianoStaff <<
  \new Staff \relative {
    \clef treble
    \discant "10"
    r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
    <<
      { r16 <f bes> r <e a> r <d g> }
      \\
      { d r a r bes r }
    >> |
    <cis e a>1
  }

  \new Staff \relative {
    \clef treble
    \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef bass \stdBass "Master"
    <<
      { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
      <e a cis>1^"a" }
      \\
      { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```
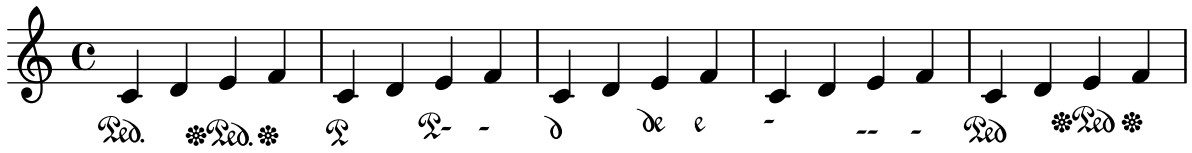


Modificar el texto de las indicaciones de pedal

Se puede usar la propiedad de contexto `Staff.pedalSustainStrings` para fijar el texto de las indicaciones de pisar pedal y levantar pedal. Observe que las únicas cadenas válidas son las que están en la lista de glifos de pedal: los valores que aparecen en este fragmento de código son una relación exhaustiva.

```
sustainNotes = { c4\sustainOn d e\sustainOff\sustainOn f\sustainOff }

\relative c' {
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("P" "P-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("d" "de" "e")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("M" "M-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("Ped" "*Ped" "*")
  \sustainNotes
}

\layout {
  ragged-right = ##f
}
```

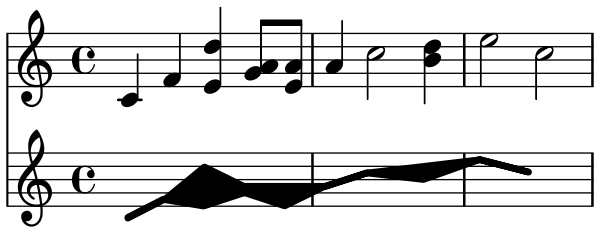


Clusters («racimos»)

Los «clusters» o racimos son un mecanismo para indicar la interpretación de un ámbito de notas al mismo tiempo.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```

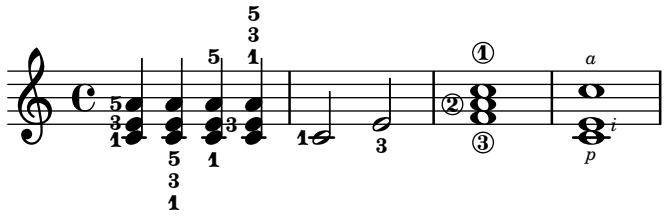


Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación usando la propiedad `fingeringOrientation`. Para que se tenga en cuenta la orientación de las digitaciones, la instrucción de digitación se debe utilizar dentro de una construcción de acorde (`<...>`) aunque sea una sola nota. Se puede controlar la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar usando las propiedades `stringNumberOrientation` y `strokeFingerOrientation`, respectivamente.

Estas propiedades se pueden establecer a una lista de entre uno y tres valores. Controlan si las digitaciones se pueden situar por encima (si aparece `up` en la lista), por debajo (si aparece `down`), a la izquierda (si aparece `left`), o a la derecha (si aparece `right`). Por contra, si una localización no aparece en la lista, no se coloca ninguna digitadión en ese lugar. LilyPond admite esas restricciones y trata de encontrar la mejor ubicación para las digitaciones de las notas de los acordes siguientes. Observe que `left` y `right` se excluyen mutuamente (las digitaciones se pueden colocar en un lado o en el otro, no en los dos).

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```
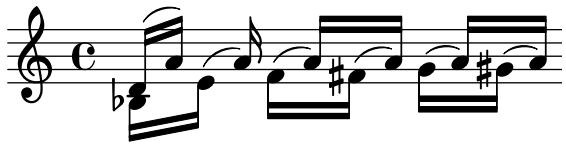


Hacer ligaduras entre voces distintas

En determinadas situaciones es necesario crear ligaduras de expresión entre notas que están en voces distintas.

La solución es añadir notas invisibles a una de las voces utilizando `\hideNotes`.

Este ejemplo es el compás 235 de la Chacona de la segunda Partita para violín solo, BWV 1004, de Bach.

```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```



Solución de problemas de acordes y barras que cruzan el pentagrama

A veces es mejor usar plicas que parten del 'otro' pentagrama para crear acordes de pentagrama cruzado y así engañar al detector de colisiones de plicas de LilyPond. En el siguiente fragmento de código, si se hubieran usado las plicas que provienen del pentagrama inferior, habría sido necesario usar explícitamente
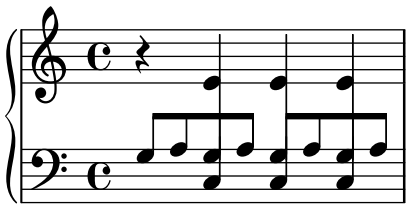
```
\override Staff.Beam.collision-voice-only = ##t
```

de forma que LilyPond no mueva las plicas.

```
\new PianoStaff <<
  \new Staff = up \relative c' <<
    { r4
      \override Stem.cross-staff = ##t
      \override Stem.length = #19 % this is in half-spaces,
          % so it makes stems 9.5 staffspaces long
      \override Stem.Y-offset = #-6 % stems are normally lengthened
          % upwards, so here we must lower the stem by the amount
          % equal to the lengthening - in this case (19 - 7) / 2
          % (7 is default stem length)
      e e e }
    { s4
      \change Staff = "bottom"
      \override NoteColumn.ignore-collision = ##t
      c, c c
    }
```

```
  >>

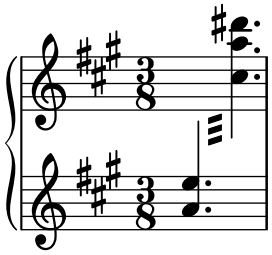
  \new Staff = bottom \relative c' {
    \clef bass
    \voiceOne
    g8 a g a g a g a
  }
>>
```



Trémolos de pentagrama cruzado

Dado que `\repeat tremolo` espera exactamente dos argumentos musicales para los trémolos de acorde, la nota o acorde que cambia de pentagrama en un trémolo que cruza el pentagrama se debe colocar dentro de llaves curvas junto a su instrucción `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```
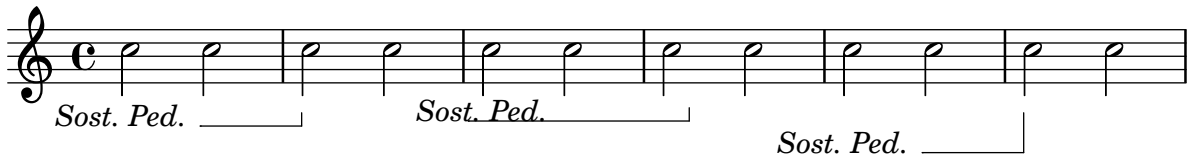


Ajuste fino de las líneas de pedal

Se puede alterar el aspecto de las líneas de pedal de varias formas.

```
\paper {
  ragged-right = ##f
}

\relative c'' {
  c2\sostenutoOn c
  c2\sostenutoOff c
  c2\tweak shorten-pair #'(-7 . -2) \sostenutoOn c
  c2\sostenutoOff c
  c2\tweak edge-height #'(0 . 3) \sostenutoOn c
  c2\sostenutoOff c
}
```



Indicating cross-staff chords with a bracket

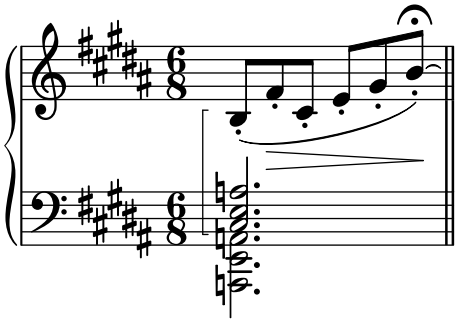
A non-arpeggiato bracket can indicate that notes on two different staves are to be played with the same hand. In order to do this, the `PianoStaff` must be set to accept cross-staff brackets.

The following example typesets measure 65 of Debussy's prelude *Les collines d'Anacapri*.

```
\new PianoStaff <<
  \once \set PianoStaff.connectChordBrackets = ##t

  \new Staff \relative c' {
    \key b \major
    \time 6/8
    b8-.(\nonArpeggiato fis'-.\> cis-.
      e-. gis-. b-.)\!\fermata^\laissezVibrer
    \section
  }

  \new Staff \relative c' {
    \clef bass
    \key b \major
    << { <a e cis>2.\nonArpeggiato } \\
       { <a, e a,>2. } >>
    \section
  }
>>
```



Plantilla para combo de jazz

Ésta es una plantilla bastante avanzada, para un conjunto de jazz. Observe que la notación de todos los instrumentos usa `\key c \major` (Do mayor). Esto se refiere al tono de concierto; la armadura se transporta automáticamente si la música está dentro de una sección `\transpose`.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup \column {
              "LilyPond example file by Amelie Zapf,"
              "Berlin 07/07/2003" }
}

% To make the example display properly in the documentation.
\paper {
  paper-width = 130\mm
  paper-height = 205\mm
}

% #(set-global-staff-size 16)

\include "english.ly"


%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%%%%

sl = { \override NoteHead.style = #'slash
       \hide Stem }
nsl = { \revert NoteHead.style
        \undo \hide Stem }
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

% Insert chord name style stuff here.

jazzChords = { }


%%%%%%%%%%% Keys'n'thangs %%%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ############ Horns ############

% ------ Trumpet ------
trpt = \transpose c d \relative c'' {
  \Key
```

```
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  \trpt
}

% ------ Alto Saxophone ------
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble
  \alto
}

% ------ Baritone Saxophone ------
bari = \transpose c a' \relative c {
  \Key
  c1 | c1 |
  \sl d4^"Solo" d d d \nsl |
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords
  s1 | s |
  d2:maj e:m7 |
}
bariSax = {
  \global
  \clef treble
  \bari
}

% ------ Trombone ------
tbone = \relative c {
  \Key
  c1 | c | c |
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
```

```
  \clef bass
  \tbone
}

% ############ Rhythm Section #############

% ------ Guitar ------
gtr = \relative c'' {
  \Key
  c1 |
  \sl b4 b b b \nsl |
  c1 |
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 | c2:min7+ d2:maj9 | s1 |
}
guitar = {
  \global
  \clef treble
  \gtr
}

%% ------ Piano ------
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c |
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e |
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g |
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c |
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
```

```
  >>
}
PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = <<
  \new Staff = "upper" \PianoRH
  \new Staff = "lower" \PianoLH
>>


% ------ Bass Guitar ------
Bass = \relative c {
  \Key
  c1 | c | c |
}
bass = {
  \global
  \clef bass
  \Bass
}

% ------ Drums ------
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
}
down = \drummode {
  \voiceTwo
  bd4 s bd s |
  bd4 s bd s |
  bd4 s bd s |
}

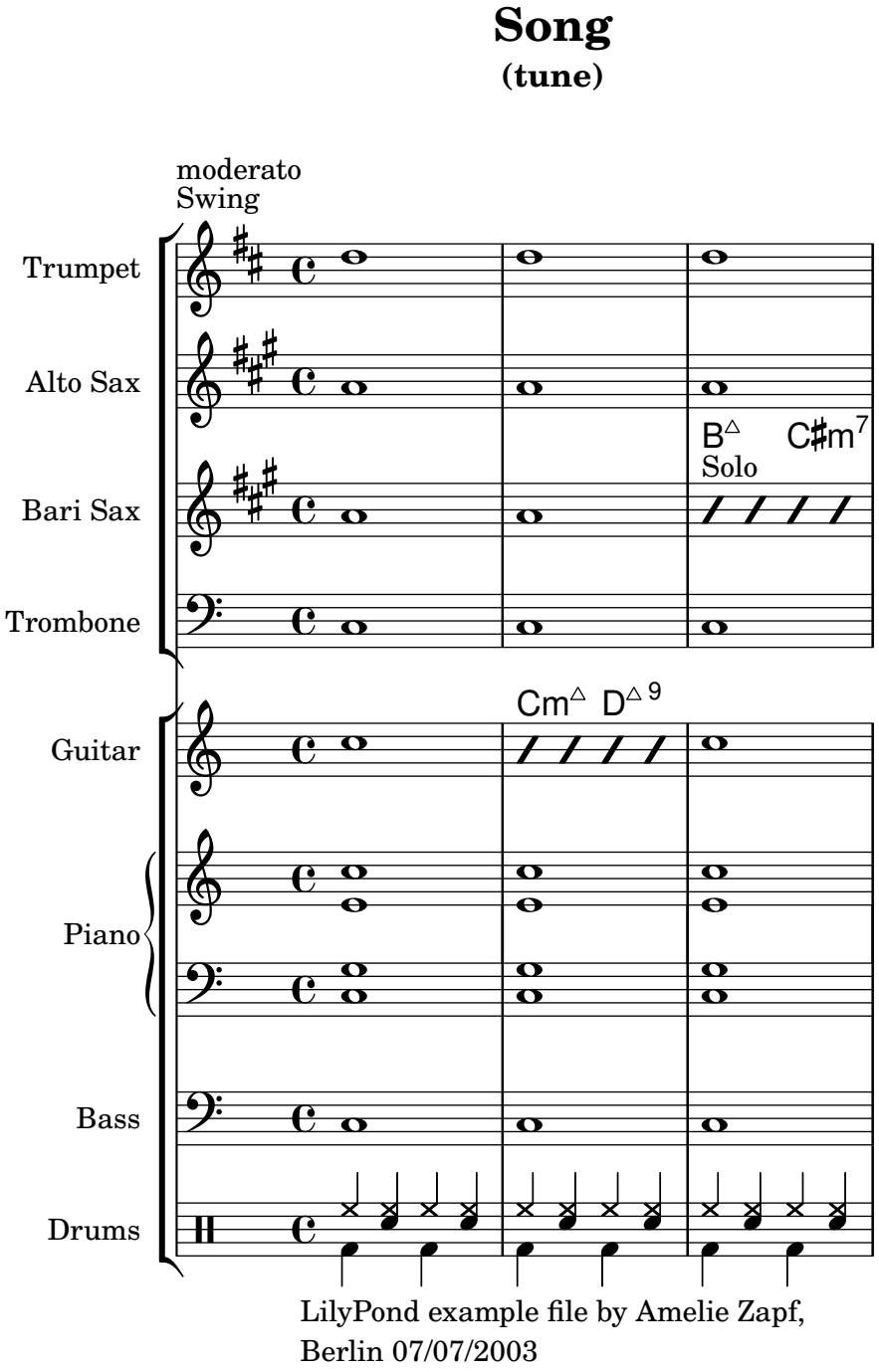
drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%%%
```

```
\book { % For the LilyPond documentation.
  \score {
    <<
      \new StaffGroup = "horns" <<
        \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
          \trumpet
        \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
          \altoSax
        \new ChordNames = "barichords" \with { instrumentName = "Bari Sax" }
          \bariHarmony
        \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
          \bariSax
        \new Staff = "trombone" \with { instrumentName = "Trombone" }
          \trombone
      >>

      \new StaffGroup = "rhythm" <<
        \new ChordNames = "chords" \with { instrumentName = "Guitar" }
          \gtrHarmony
        \new Staff = "guitar" \with { instrumentName = "Guitar" }
          \guitar
        \new PianoStaff = "piano" \with {
          instrumentName = "Piano"
          midiInstrument = "acoustic grand"
        } \piano
        \new Staff = "bass" \with { instrumentName = "Bass" }
          \bass
        \new DrumStaff \with { instrumentName = "Drums" }
          \drumContents
      >>
    >>

    \layout {
      \context {
        \Staff
        \RemoveEmptyStaves
      }
      \context {
        \Score
        \override BarNumber.padding = 3
        \override RehearsalMark.padding = 2
        skipBars = ##t
      }
    }
    \midi { }
  }
}
```



Ligaduras *laissez vibrer*
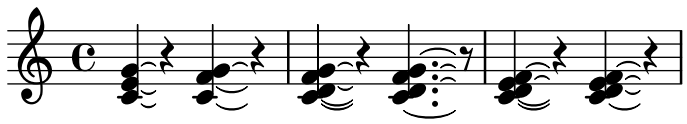
Las ligaduras "laissez vibrer" (dejar vibrar) tienen un tamaño fijo. Se puede ajustar su posición usando la propiedad `tie-configuration`.

Véase también el fragmento de código "Ligaduras laissez vibrer largas".

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
    = #`((-7 . ,DOWN)
```

```
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



Plantilla de piano (sencilla)

Presentamos a continuación una plantilla de piano sencilla con algunas notas.

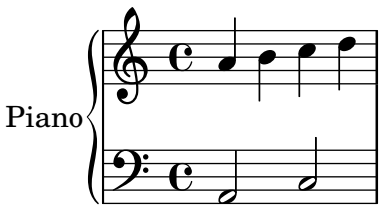
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  \new PianoStaff \with { instrumentName = "Piano" }
  <<
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}
```



Plantilla de piano con letra centrada

En lugar de tener un pentagrama dedicado a la melodía y la letra, ésta se puede centrar entre los pentagramas de un sistema de piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

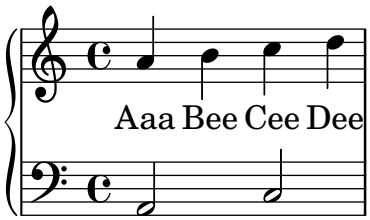
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}
```



Plantilla de piano con melodía y letra

He aquí el típico formato dde una canción: un pentagrama con la melodía y la letra, y el acompañamiento de piano por debajo.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}
```

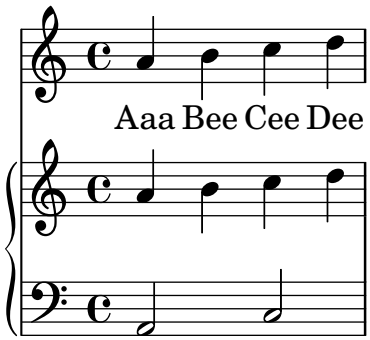
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```



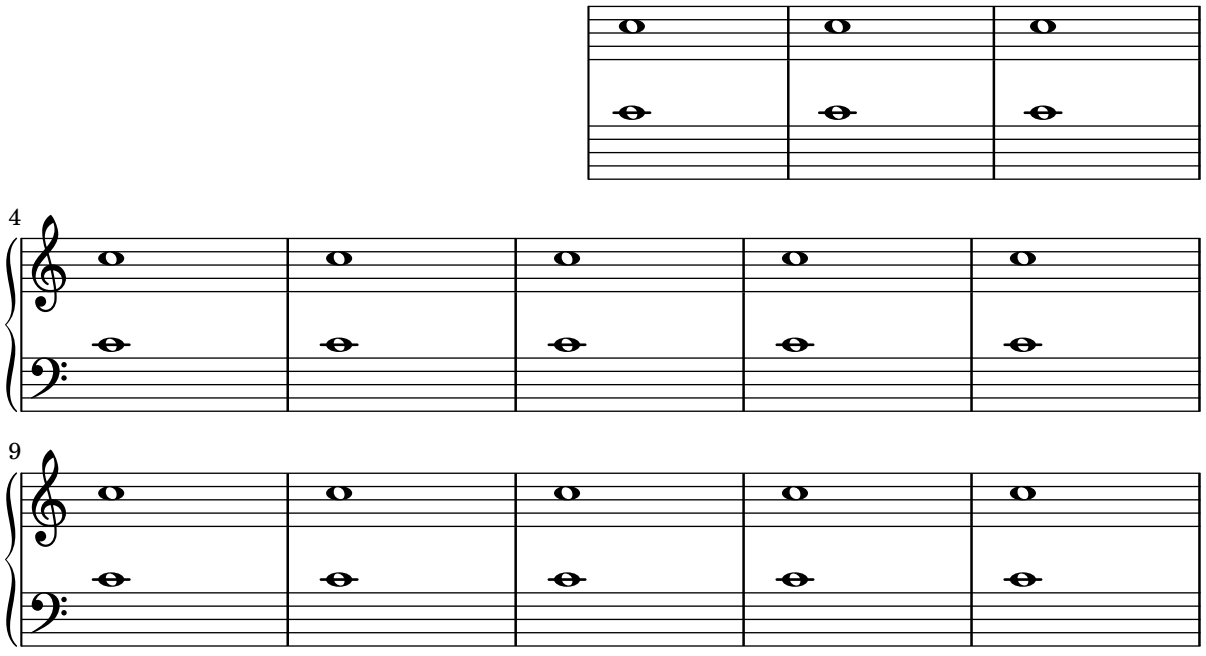
Quitar la llave en el primer sistema de una partitura de piano

Este fragmento elimina la primera llave de un `PianoStaff` o un `GrandStaff`, junto con las claves. Puede ser útil cuando se está cortando y pegando la imagen de la partitura editada dentro de otra música existente.

El código emplea `\alterBroken` para ocultar el delimitador en forma de llave que aparece al principio.

```
someMusic =  {
  \once \omit Staff.Clef
  \once \omit Staff.TimeSignature
  \*3 c1 \break
  \*5 c1 \break
  \*5 c1
}

\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c'' \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
    indent=75\mm
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}
```



Using \autoChange with more than one voice

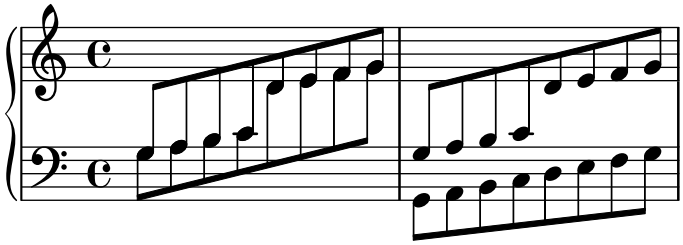
Here is a demonstration of how to use \autoChange with more than one voice.

```
\score {
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
```

```
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,8 a b c d e f g
          }
        }

        \new Voice {
          \voiceTwo
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,,8 a b c d e f g
          }
        }
      >>
    }

    \new Staff = "down" {
      \clef bass
    }
  >>
}
```



Plantilla de conjunto vocal con reducción de piano automática

Esta plantilla añade una reducción de piano automática a la partitura vocal SATB estándar que se mostró en el fragmento de código “Plantilla de conjunto vocal”. Presenta uno de los puntos fuertes de LilyPond: podemos usar una definición de música más de una vez. Si se hace cualquier cambio en las notas de la parte vocal (digamos `tenorMusic`), entonces los cambios se aplican también a la reducción de piano.

```
\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}

global = {
  \key c \major
  \time 4/4
}
```

```
sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords =\lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

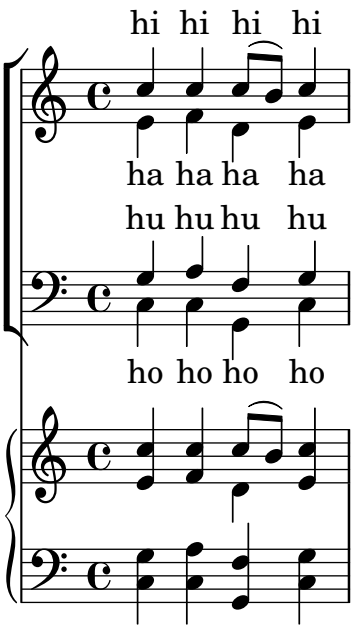
bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"

      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"
```

```
      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>

    \new PianoStaff <<
      \new Staff <<
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \sopMusic >>
        << \global \altoMusic >>
      >>
      \new Staff <<
        \clef bass
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \tenorMusic >>
        << \global \bassMusic >>
      >>
    >>
  >>
}
```



11 Unfretted string instruments

See also Sección “Unfretted string instruments” en *Notation Reference*.
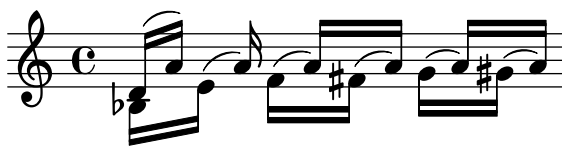
Hacer ligaduras entre voces distintas

En determinadas situaciones es necesario crear ligaduras de expresión entre notas que están en voces distintas.

La solución es añadir notas invisibles a una de las voces utilizando `\hideNotes`.

Este ejemplo es el compás 235 de la Chacona de la segunda Partita para violín solo, BWV 1004, de Bach.
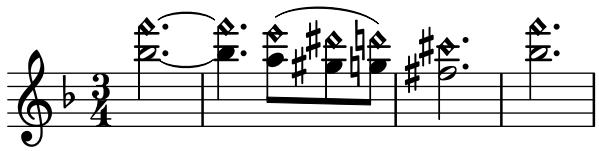
```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```



Armónicos con puntillo

Los armónicos artificiales que usan la instrucción `\harmonic` no tienen puntillo. Para sobreescribir este comportamiento, fije la propiedad de contexto `harmonicDots`.

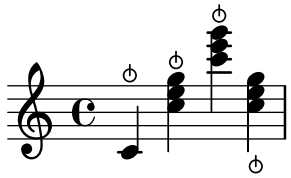
```
\relative c''' {
  \time 3/4
  \key f \major
  \set harmonicDots = ##t
  <bes f'\harmonic>2. ~
  <bes f'\harmonic>4. <a e'\harmonic>8( <gis dis'\harmonic> <g d'\harmonic>)
  <fis cis'\harmonic>2.
  <bes f'\harmonic>2.
}
```



Snap pizzicato ("Bartok" pizzicato)

A snap pizzicato (also known as "Bartok pizzicato") is a "strong pizzicato where the string is plucked vertically by snapping and rebounds off the fingerboard of the instrument" (Wikipedia). It is denoted by a circle with a vertical line going from the center upwards outside the circle.

```
\relative c' {
  c4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}
```



Plantilla de cuarteto de cuerda (sencilla)

Esta plantilla muestra un cuarteto de cuerda normal. También utiliza una sección `\global` para el compás y la armadura.

Véase también el fragmento de código "Plantilla de cuarteto de cuerda con particellas independientes".

```
global= {
  \time 4/4
  \key c \major
}

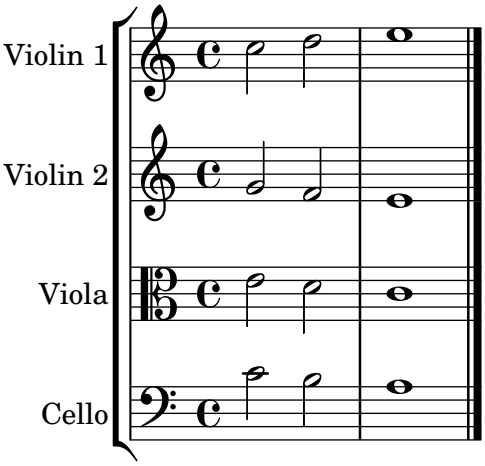
violinOne = \new Voice \relative c'' {
  c2 d
  e1
  \bar "|."
}

violinTwo = \new Voice \relative c'' {
  g2 f
  e1
  \bar "|."
}

viola = \new Voice \relative c' {
  \clef alto
  e2 d
  c1
  \bar "|."
}
```

```
cello = \new Voice \relative c' {
  \clef bass
  c2 b
  a1
  \bar "|."
}

\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Violin 1" }
      << \global \violinOne >>
    \new Staff \with { instrumentName = "Violin 2" }
      << \global \violinTwo >>
    \new Staff \with { instrumentName = "Viola" }
      << \global \viola >>
    \new Staff \with { instrumentName = "Cello" }
      << \global \cello >>
  >>
  \layout { }
  \midi { }
}
```



Plantilla de cuarteto de cuerda con particellas independientes

El fragmento de código “Plantilla de cuarteto de cuerda” produce un resultado satisfactorio para el cuarteto, pero ¿y si tenemos que imprimir las particellas? Esta nueva plantilla muestra cómo usar la funcionalidad `\tag` (etiqueta) para dividir fácilmente una pieza en particellas individuales.

Tenemos que dividir esta plantilla en archivos independientes; los nombres de archivo están dentro de los comentarios al principio de cada archivo. `piece.ly` contiene todas las definiciones de música. Los otros archivos (`score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` y `vlc.ly`) producen la particella correspondiente.

¡No olvide quitar los comentarios que hemos especificado cuando use los archivos independientes!

```
% piece.ly
% (This is the global definitions file.)

global= {
  \time 4/4
  \key c \major
}

Violinone = \new Voice \relative c'' {
  c2 d e1
  \bar "|."
}

Violintwo = \new Voice \relative c'' {
  g2 g e1
  \bar "|."
}

Viola = \new Voice \relative c' {
  \clef alto
  e2 d c1
  \bar "|."
}

Cello = \new Voice \relative c' {
  \clef bass
  c2 b a1
  \bar "|."
}

music = <<
  \tag #'score \tag #'vn1
  \new Staff \with { instrumentName = "Violin 1" }
    << \global \Violinone >>

  \tag #'score \tag #'vn2
  \new Staff \with { instrumentName = "Violin 2" }
    << \global \Violintwo >>

  \tag #'score \tag #'vla
  \new Staff \with { instrumentName = "Viola" }
    << \global \Viola >>

  \tag #'score \tag #'vlc
  \new Staff \with { instrumentName = "Cello" }
    << \global \Cello >>
>>


% These are the other files you need to save on your computer

% score.ly
```

```
% (This is the main file.)

% Uncomment the line below when using a separate file.
% \include "piece.ly"

#(set-global-staff-size 14)

\score {
  \new StaffGroup \keepWithTag #'score \music
  \layout { }
  \midi { }
}


%{ Uncomment this block when using separate files.

% vn1.ly
% (This is the Violin 1 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn1 \music
  \layout { }
}

% vn2.ly
% (This is the Violin 2 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn2 \music
  \layout { }
}

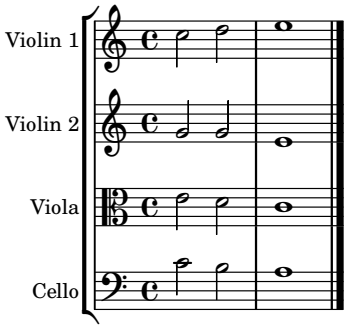
% vla.ly
% (This is the Viola part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}

% vlc.ly
% (This is the Cello part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vlc \music
  \layout { }
}
```

```
%}
```


Violin 1
Violin 2
Viola
Cello
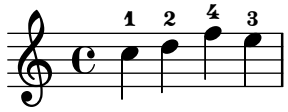
12 Fretted string instruments

See also Sección “Fretted string instruments” en *Notation Reference*.

Añadir digitaciones a la partitura

Se pueden escribir instrucciones de digitación usando una sintaxis muy sencilla.

```
\relative c'' {
  c4-1 d-2 f-4 e-3
}
```

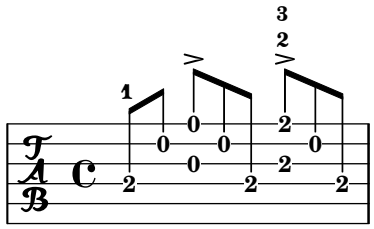


Añadir digitaciones a las tablaturas

Para añadir digitaciones a las tablaturas, utilice una combinación de `\markup` y `\finger`.

```
one = \markup { \finger 1 }
two = \markup { \finger 2 }
threeTwo = \markup {
  \override #'(baseline-skip . 2)
  \column {
    \finger 3
    \finger 2
  }
}

\score {
  \new TabStaff {
    \tabFullNotation
    \stemUp
    e8\4^\one b\2 <g\3 e'\1>^>[ b\2 e\4]
    <a\3 fis'\1>^>^\threeTwo[ b\2 e\4]
  }
}
```



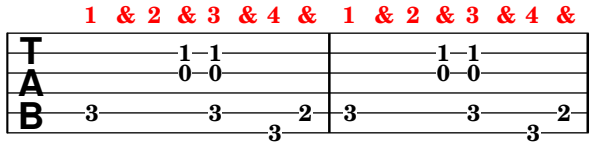
Insertar elementos de marcado en una tablatura

De forma predeterminada, los elementos de marcado no aparecen en la tablatura.

Para hacer que aparezcan, revierta la propiedad `stencil` del objeto gráfico `TextScript` dentro del contexto `TabStaff`.

```
high  = { r4 r8 <g c'> q r8 r4 }
low = { c4 r4 c8 r8 g,8 b, }
pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }
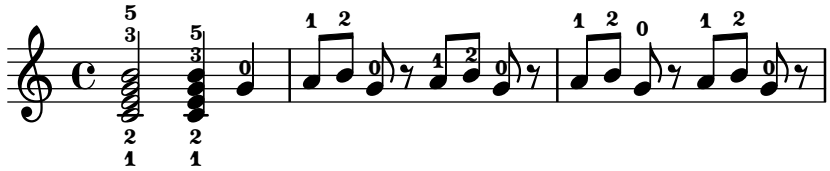
\score {
 \new TabStaff {
   \*2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Hay que prestar atención en situaciones en las que las digitaciones y las plicas están en la misma dirección: por defecto, las digitaciones solo evitan la colisión con plicas unidas por una barra. Se puede cambiar este ajuste para no evitar ninguna plica o evitarlas todas; el ejemplo siguiente muestra las dos copciones, así como la manera de volver al comportamiento predeteminado.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```

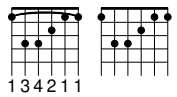


Automatic fretboards barré

When automatic fretboards are used, barré indicators are drawn whenever one finger is responsible for multiple strings.

If no finger indications are given in the chord from which the automatic fretboard is created, no barré indicators are included, because there is no way to identify where barrés should be placed.

```
\new FretBoards {
  <f,-1 c-3 f-4 a-2 c'-1 f'-1>1
  <f, c f a c' f'>1
}
```



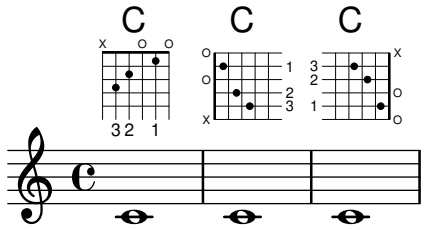
Modificar la orientación de los trastes

Los diagramas de posiciones de acordes se pueden orientar de tres formas. De manera predeterminada se alinena la cuerda o traste superior en las distintas alineaciones.

```
\include "predefined-guitar-fretboards.ly"

<<
  \chords {
    c1
    c1
    c1
  }
  \new FretBoards \chordmode {
    c1
    \override FretBoard.fret-diagram-details.orientation =
      #'landscape
    c1
    \override FretBoard.fret-diagram-details.orientation =
```

```
      #'opposing-landscape
    c1
  }
  \new Voice {
    c'1
    c'1
    c'
  }
>>
```



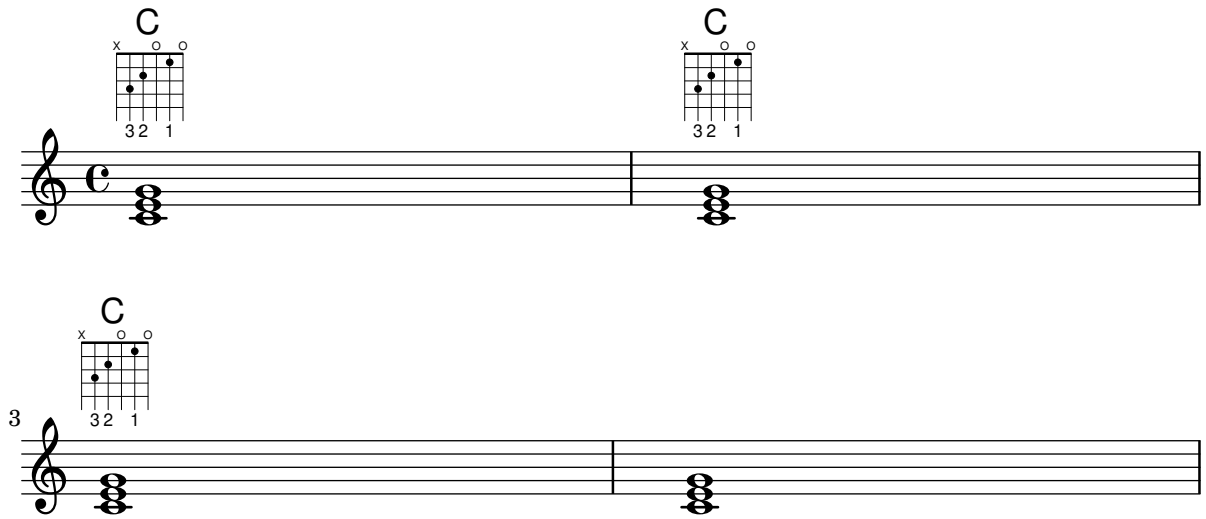
Chord changes for fretboards

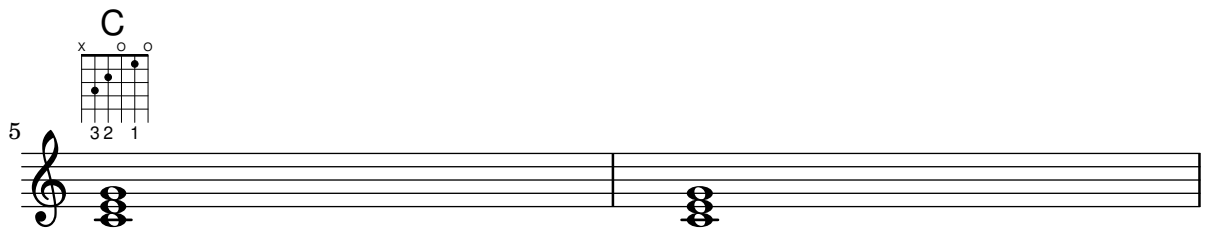
Fretboards can be set to display only when the chord changes, or at the beginning of a new line.

```
\include "predefined-guitar-fretboards.ly"

myChords = \chordmode {
  c1 c1 \break
  \set chordChanges = ##t
  c1 c1 \break
  c1 c1
}

<<
  \new ChordNames { \myChords }
  \new FretBoards { \myChords }
  \new Staff { \myChords }
>>
```





Glissando de acordes en tablatura

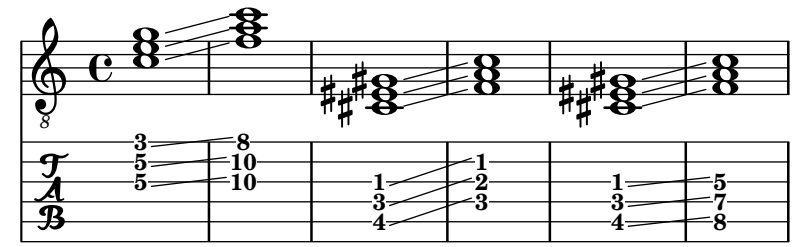
Los deslizamientos para acordes se indican por omisión tanto en el contexto `Staff` como en `TabStaff`.

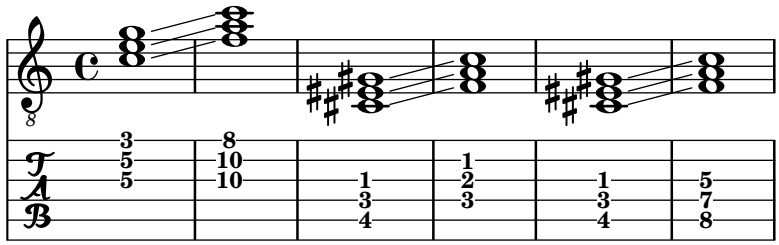
Los números de cuerda son necesarios para TabStaff porque los cálculos de cuerda automáticos son diferentes para los acordes y para notas sueltas.

```
myMusic = \relative c' {
  <c e g>1 \glissando <f a c>
  <cis, eis gis>1 \glissando <f a c>
  <cis eis gis>1 \glissando <f a c\3>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \omit StringNumber
      \myMusic
    }
    \new TabStaff \myMusic
  >>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \omit StringNumber
      \myMusic
    }
    \new TabStaff \with { \override Glissando.style = #'none } {
      \myMusic
    }
  >>
}
```





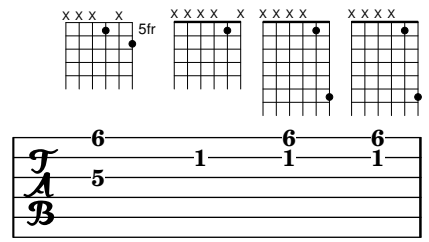
Acordes con digitaciones ampliadas para diagramas de trastes y tablaturas

A veces se requieren acordes con digitación ampliada o estirada. Sin embargo, si no se especifica lo contrario, la propiedad de contexto `maximumFretStretch` viene establecida al valor 4, lo que puede dar lugar a una advertencia "No hay ninguna cuerda para la nota ..." y la nota se omite. Podemos fijar `maximumFretStretch` a un valor adecuado o asignar explícitamente números de cuerda para todas las notas del acorde, para corregir esto.

```
% The code below prints two warnings for the second chord,
% which may be omitted by uncommenting the following line.
%
% #(for-each (lambda (x) (ly:expect-warning "No string for pitch")) (iota 2))

mus = {
  <c' bes'>
  <c'\2 bes'>
  \set maximumFretStretch = 5
  <c' bes'>
  <c'\2 bes'\1>
}

<<
  \new FretBoards \mus
  \new TabVoice \mus
>>
```



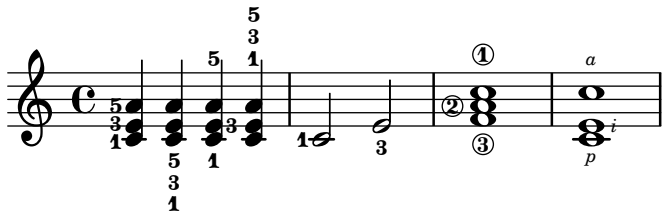
Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación usando la propiedad `fingeringOrientation`. Para que se tenga en cuenta la orientación de las digitaciones, la instrucción de digitación se debe utilizar dentro de una construcción de acorde (`<...>`) aunque sea una sola nota. Se puede controlar la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar usando las propiedades `stringNumberOrientation` y `strokeFingerOrientation`, respectivamente.

Estas propiedades se pueden establecer a una lista de entre uno y tres valores. Controlan si las digitaciones se pueden situar por encima (si aparece `up` en la lista), por debajo (si aparece `down`), a la izquierda (si aparece `left`), o a la derecha (si aparece `right`). Por contra, si una

localización no aparece en la lista, no se coloca ninguna digitadión en ese lugar. LilyPond admite esas restricciones y trata de encontrar la mejor ubicación para las digitaciones de las notas de los acordes siguientes. Observe que `left` y `right` se excluyen mutuamente (las digitaciones se pueden colocar en un lado o en el otro, no en los dos).

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Personalizar los diagramas de posiciones

Se pueden modificar las propiedades de los diagramas de posiciones de acordes estableciendo la propiedad `fret-diagram-details`. Para los diagramas de posiciones de `FretBoard`, se aplican los overrides (sobreescrituras) al objeto `FretBoards.FretBoard`. Como `Voice`, `FretBoards` es un contexto del nivel inferior, y por tanto se puede omitir su nombre en la sobreescritura de propiedades.

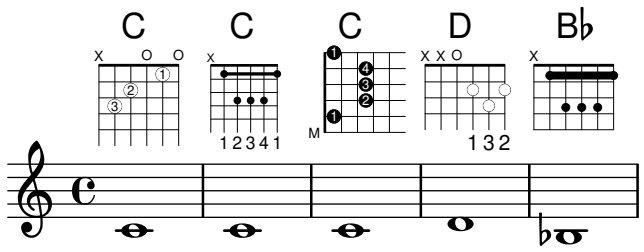
```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        "x;1-1-(;3-2;3-3;3-4;1-1-);"

% shorthand
oo = #(define-music-function
       (grob-path value)
       (list? scheme?)
       #{ \once \override $grob-path = #value #})

<<
  \new ChordNames {
    \chordmode { c1 | c | c | d | bes }
```

```
  }
  \new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard.size = 1.2
    \override FretBoard.fret-diagram-details.finger-code = #'in-dot
    \override FretBoard.fret-diagram-details.dot-color = #'white
    \chordmode {
      c
      \oo FretBoard.size #1.0
      \oo FretBoard.fret-diagram-details.barre-type #'straight
      \oo FretBoard.fret-diagram-details.dot-color #'black
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      c'
      \oo FretBoard.fret-diagram-details.barre-type #'none
      \oo FretBoard.fret-diagram-details.number-type #'arabic
      \oo FretBoard.fret-diagram-details.orientation #'landscape
      \oo FretBoard.fret-diagram-details.mute-string "M"
      \oo FretBoard.fret-diagram-details.label-dir #LEFT
      \oo FretBoard.fret-diagram-details.dot-color #'black
      c'
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      \oo FretBoard.fret-diagram-details.dot-radius #0.35
      \oo FretBoard.fret-diagram-details.dot-position #0.5
      \oo FretBoard.fret-diagram-details.fret-count #3
      d
      \oo FretBoard.fret-diagram-details.barre-type #'straight
      \oo FretBoard.fret-diagram-details.finger-code #'none
      \oo FretBoard.fret-diagram-details.dot-radius #0.25
      \oo FretBoard.fret-diagram-details.dot-color #'black
      \oo FretBoard.fret-diagram-details.string-overhang #0.
      \oo FretBoard.fret-diagram-details.barre-thickness #2.
      bes
    }
  }
  \new Voice {
    c'1 | c' | c' | d' | bes
  }
>>
```



Personalizar diagramas de posiciones de marcado

Se pueden modificar las propiedades de los diagramas de posiciones estableciendo el valor de la propiedad `fret-diagram-details`. Para los diagramas de posiciones de marcado, se pueden aplicar overrides (sobreescrituras) al objeto `Voice.TextScript` o directamente al elemento de marcado.

```
<<
  \chords { c1 | c | c | d }

  \new Voice = "mel" {
    \textLengthOn
    % Set global properties of fret diagram
    \override TextScript.size = 1.2
    \override TextScript.fret-diagram-details.finger-code = #'in-dot
    \override TextScript.fret-diagram-details.dot-color = #'white

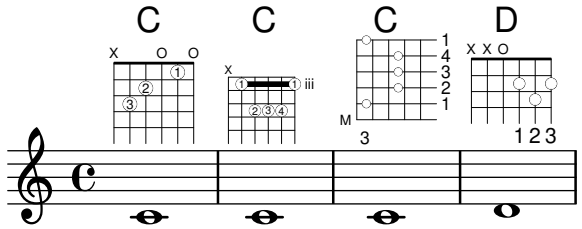
    %% C major for guitar, no barre, using defaults
       % terse style
    c'1^\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

    %% C major for guitar, barred on third fret
       % verbose style
       % size 1.0
       % roman fret label, finger labels below string, straight barre
    c'1^\markup {
      % standard size
      \override #'(size . 1.0) {
        \override #'(fret-diagram-details . (
                    (number-type . roman-lower)
                    (finger-code . in-dot)
                    (barre-type . straight))) {
          \fret-diagram-verbose #'((mute 6)
                                   (place-fret 5 3 1)
                                   (place-fret 4 5 2)
                                   (place-fret 3 5 3)
                                   (place-fret 2 5 4)
                                   (place-fret 1 3 1)
                                   (barre 5 1 3))
        }
      }
    }

    %% C major for guitar, barred on third fret
       % verbose style
       % landscape orientation, arabic numbers, M for mute string
       % no barre, fret label down or left, small mute label font
    c'1^\markup {
      \override #'(fret-diagram-details . (
                  (finger-code . below-string)
                  (number-type . arabic)
                  (label-dir . -1)
                  (mute-string . "M")
                  (orientation . landscape)
                  (barre-type . none)
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3))) {
        \fret-diagram-verbose #'((mute 6)
                                 (place-fret 5 3 1)
```

```
                                    (place-fret 4 5 2)
                                    (place-fret 3 5 3)
                                    (place-fret 2 5 4)
                                    (place-fret 1 3 1)
                                    (barre 5 1 3))
      }
    }

    %% simple D chord
       % terse style
       % larger dots, centered dots, fewer frets
       % label below string
    d'1^\markup {
      \override #'(fret-diagram-details . (
                   (finger-code . below-string)
                   (dot-radius . 0.35)
                   (dot-position . 0.5)
                   (fret-count . 3))) {
        \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```

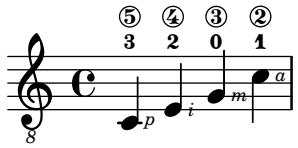


Digitaciones, indicación del número de cuerda y digitaciones de mano derecha

En este ejemplo se combinan las digitaciones de la mano izquierda, indicaciones del número de cuerda y digitaciones de la mano derecha.

```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"
  <c-3\5\RH 1 >4
  <e-2\4\RH 2 >4
  <g-0\3\RH 3 >4
  <c-1\2\RH 4 >4
}
```



Notación del flamenco

Para la guitarra flamenca se utilizan ciertos elementos de notación especiales.

- Un símbolo para indicar un *golpe* sobre la caja de la guitarra con la uña del dedo anular.
- Una flecha para indicar rasgueos y su dirección.
- Letras para las digitaciones ("p": pulgar, "i": índice, "m": medio, "a": anular y "x": meñique).
- *Rasgueados* de 3 y 4 dedos: hacia arriba con todos los dedos y terminando con arriba y abajo con el dedo índice.
- *Abanicos*: rasgueos en serie con el pulgar, hacia abajo y el meñique y el índice hacia arriba (hay también un *abanico 2* en el que los dedos medio y anular se usan en lugar del meñique).
- *Alza púa*: pulsaciones rápidas con el pulgar.

Casi todas las figuras utilizan flechas combinadas con digitaciones; con los abanicos y los rasgueados, las notas se imprimen con cabeza solo en el primer acorde.

Este fragmento de código contiene código de tipo cabecera que se puede copiar como `flamenco.ly` e incluirse en los documentos fuente.

```
%%%%%%  Cut here ----- Start of `flamenco.ly`.

% Text indicators.
abanico = ^\markup \small { \italic Abanico }
rasgueado = ^\markup \small { \italic Ras. }
alzapua = ^\markup \small { \italic Alzapua }

% Finger stroke symbols.
strokeUp = \markup {
  \combine
    \override #'(thickness . 1.3) \draw-line #'(0 . 2)
    \raise #2 \arrow-head #Y #UP ##f }
strokeDown = \markup {
  \combine
    \arrow-head #Y #DOWN ##f
    \override #'(thickness . 1.3) \draw-line #'(0 . 2) }

% Golpe symbol.
golpe = \markup {
  \filled-box #'(0 . 1) #'(0 . 1) #0
  \hspace #-1.6
  \with-color #white
  \filled-box #'(0.15 . 0.85) #'(0.15 . 0.85) #0
}

% Strokes, fingers, and golpe command.
RHp = \rightHandFinger #1
RHi = \rightHandFinger #2
RHm = \rightHandFinger #3
RHa = \rightHandFinger #4
RHx = \rightHandFinger #5
RHu = \rightHandFinger \strokeUp
RHd = \rightHandFinger \strokeDown
RHg = \rightHandFinger \golpe
```

```
% Various shorthands.
tupletOff = {
  \once \omit TupletNumber
  \once \omit TupletBracket
}

tupletsOff = {
  \omit TupletNumber
  \override TupletBracket.bracket-visibility = #'if-no-beam
}

tupletsOn = {
  \override TupletBracket.bracket-visibility = #'default
  \undo \omit TupletNumber
}

headsOff = {
  \hide TabNoteHead
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

headsOn = {
  \override TabNoteHead.transparent = ##f
  \override NoteHead.transparent = ##f
  \override NoteHead.no-ledgers = ##f
}

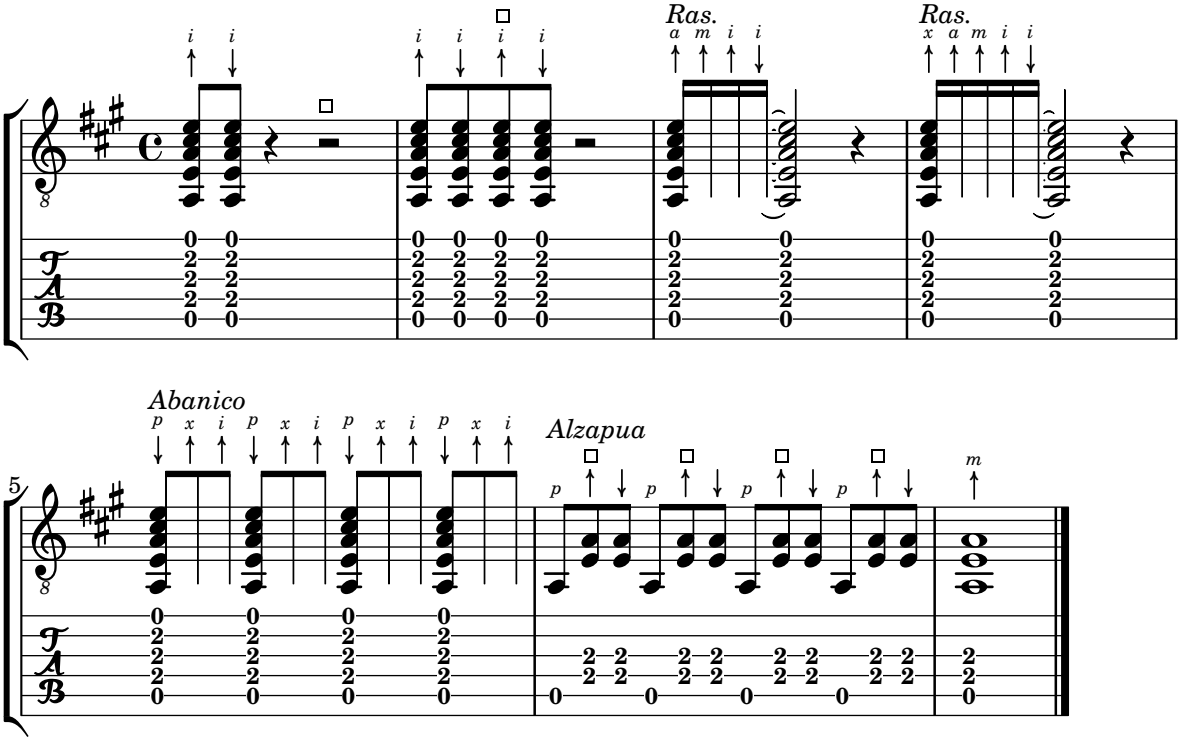
%%%%%%  Cut here ----- End of `flamenco.ly`.


part = \relative c' {
  \set strokeFingerOrientations = #'(up)
  \key a\major

  <a, e' a cis e\RHu\RHi>8
    <a e' a cis e\RHd\RHi>8
    r4
    r2^\markup\golpe |
  <a e' a cis e\RHu\RHi>8
    <a e' a cis e\RHd\RHi>8
    <a e' a cis e\RHu\RHi\RHg>8
    <a e' a cis e\RHd\RHi>8
    r2 |
  <a e' a cis e\RHu\RHa>16\rasgueado
    \headsOff
    <a e' a cis e\RHu\RHm>
    <a e' a cis e\RHu\RHi>
    <a e' a cis e\RHd\RHi>~
    \headsOn
    <a e' a cis e>2
```

```
    r4 |
  \tupletOff
    \tuplet 5/4 {
      <a e' a cis e\RHu\RHx>16\rasgueado
      \headsOff
      <a e' a cis e\RHu\RHa>
      <a e' a cis e\RHu\RHm>
      <a e' a cis e\RHu\RHi>
      <a e' a cis e\RHd\RHi>~
      \headsOn
    }
    <a e' a cis e>2
    r4 |
  <>\abanico
    \tupletsOff
    \repeat unfold 4 {
      \tuplet 3/2 {
        <a e' a cis e\RHd\RHp>8
        \headsOff
        <a e' a cis e\RHu\RHx>
        <a e' a cis e\RHu\RHi>
        \headsOn
      }
    }
    \tupletsOff |
  <>\alzapua
  \override Beam.positions = #'(2 . 2)
    \repeat unfold 4 {
      \tuplet 3/2 {
        a8\RHp
        <e' a\RHu\RHg>
        <e a\RHd>
      }
    }
    \tupletsOn |
  <a, e' a\RHu\RHm>1 \bar "|."
}

\score {
  \new StaffGroup <<
    \context Staff = "part" {
      \clef "G_8"
      \part
    }
    \context TabStaff {
      \part
    }
  >>
  \layout {
    ragged-right = ##t
  }
}
```



Diagramas de posiciones de acorde, explicados y desarrollados

Este fragmento de código presenta muchas posibilidades para obtener diagramas de posiciones de acorde y cómo trucarlos.

```
<<
  \chords {
    a1 a \bar "||" \break
    \repeat unfold 3 {
      c c c d d \bar "||" \break
    }
  }

  \new Voice {
    % Set global properties of fret diagram
    \override TextScript.size = 1.2
    \override TextScript.fret-diagram-details
                .finger-code = #'below-string
    \override TextScript.fret-diagram-details
                .dot-color = #'black

    % 1
    %
    % A chord for ukulele.
    a'1^\markup
      \override #'(fret-diagram-details
                   . ((string-count . 4)
                      (dot-color . white)
                      (finger-code . in-dot)))
      \fret-diagram "4-2-2;3-1-1;2-o;1-o;"
```

```
% 2
%
% A chord for ukulele, with formatting defined in definition
% string: 1.2 * size, 4 strings, 4 frets, fingerings below,
% string dot radius .35 of fret spacing, dot position 0.55 of
% fret spacing.
a'1^\markup
  \override #'(fret-diagram-details
              . ((dot-color . white)
                 (open-string . "o")))
  \fret-diagram
    "s:1.2;w:4;h:3;f:2;d:0.35;p:0.55;4-2-2;3-1-1;2-o;1-o;"


%%
%% These chords will be in normal orientation
%%


% 3
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
              . ((number-type . roman-lower)
                 (finger-code . below-string)
                 (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 5 1 3))

% 4
%
% C major for guitar, barred on third fret: double barre used
% to test barre function, verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
              . ((number-type . arabic)
                 (dot-label-font-mag . 0.9)
                 (finger-code . in-dot)
                 (fret-label-font-mag . 0.6)
                 (fret-label-vertical-offset . 0)
                 (label-dir . -1)
```

```
                    (mute-string . "M")
                    (xo-font-magnification . 0.4)
                    (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 4 2 5)
                           (barre 5 1 3))

% 5
%
% C major for guitar, with capo on third fret: verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-upper)
                  (dot-label-font-mag . 0.9)
                  (finger-code . none)
                  (fret-label-vertical-offset . 0.5)
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (capo 3)
                           (open 5)
                           (place-fret 4 5 1)
                           (place-fret 3 5 2)
                           (place-fret 2 5 3)
                           (open 1))

% 6
%
% Simple D chord.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
                  (dot-radius . 0.35)
                  (string-thickness-factor . 0.3)
                  (dot-position . 0.5)
                  (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 7
%
% Simple D chord, large top fret thickness.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
                  (dot-radius . 0.35)
```

```
                  (dot-position . 0.5)
                  (top-fret-thickness . 7)
                  (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"


%%
%% These chords will be in landscape orientation
%%
\override TextScript.fret-diagram-details
            .orientation = #'landscape


% 8
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight
% barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-lower)
                  (finger-code . below-string)
                  (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 5 1 3))

% 9
%
% C major for guitar, barred on third fret: Double barre
% used to test barre function, verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . arabic)
                  (dot-label-font-mag . 0.9)
                  (finger-code . in-dot)
                  (fret-label-font-mag . 0.6)
                  (fret-label-vertical-offset . 0)
                  (label-dir . -1)
                  (mute-string . "M")
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
```

```
                            (place-fret 4 5 2)
                            (place-fret 3 5 3)
                            (place-fret 2 5 4)
                            (place-fret 1 3 1)
                            (barre 4 2 5)
                            (barre 5 1 3))

% 10
%
% C major for guitar, with capo on third fret: verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-upper)
                  (dot-label-font-mag . 0.9)
                  (finger-code . none)
                  (fret-label-vertical-offset . 0.5)
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (capo 3)
                           (open 5)
                           (place-fret 4 5 1)
                           (place-fret 3 5 2)
                           (place-fret 2 5 3)
                           (open 1))

% 11
%
% Simple D chord.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
                  (dot-radius . 0.35)
                  (dot-position . 0.5)
                  (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 12
%
% Simple D chord, large top fret thickness.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
                  (dot-radius . 0.35)
                  (dot-position . 0.5)
                  (top-fret-thickness . 7)
                  (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
```

```
%%
%% These chords will be in opposing-landscape orientation.
%%
\override TextScript.fret-diagram-details
            .orientation = #'opposing-landscape


% 13
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight
% barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-lower)
                  (finger-code . below-string)
                  (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 5 1 3))

% 14
%
% C major for guitar, barred on third fret: double barre
% used to test barre function, verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . arabic)
                  (dot-label-font-mag . 0.9)
                  (finger-code . in-dot)
                  (fret-label-font-mag . 0.6)
                  (fret-label-vertical-offset . 0)
                  (label-dir . -1)
                  (mute-string . "M")
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 4 2 5)
                           (barre 5 1 3))
```

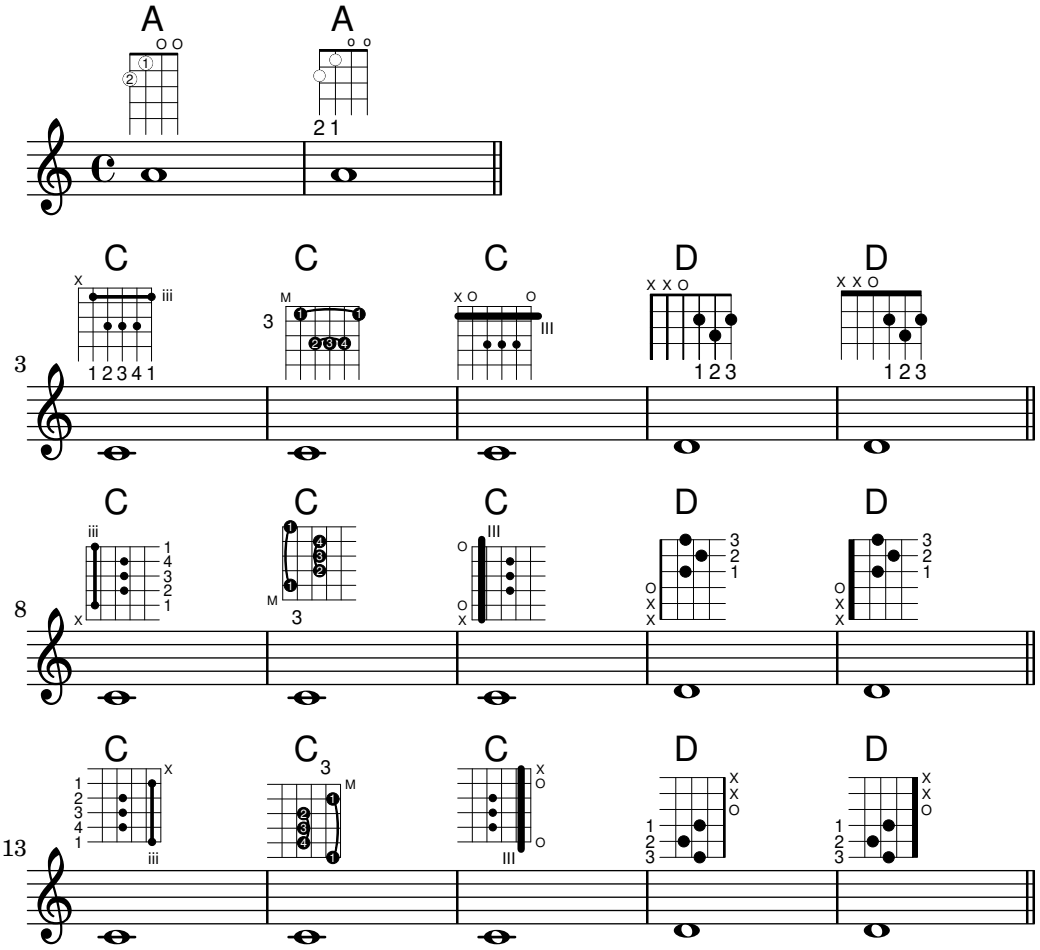
```
    % 15
    %
    % C major for guitar, with capo on third fret: verbose style.
    c'1^\markup
      % 110% of default size
      \override #'(size . 1.1)
      \override #'(fret-diagram-details
                   . ((number-type . roman-upper)
                      (dot-label-font-mag . 0.9)
                      (finger-code . none)
                      (fret-label-vertical-offset . 0.5)
                      (xo-font-magnification . 0.4)
                      (xo-padding . 0.3)))
      \fret-diagram-verbose #'((mute 6)
                               (capo 3)
                               (open 5)
                               (place-fret 4 5 1)
                               (place-fret 3 5 2)
                               (place-fret 2 5 3)
                               (open 1))

    % 16
    %
    % Simple D chord.
    d'1^\markup
      \override #'(fret-diagram-details
                   . ((finger-code . below-string)
                      (dot-radius . 0.35)
                      (dot-position . 0.5)
                      (fret-count . 3)))
      \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

    % 17
    %
    % Simple D chord, large top fret thickness.
    d'1^\markup
      \override #'(fret-diagram-details
                   . ((finger-code . below-string)
                      (dot-radius . 0.35)
                      (dot-position . 0.5)
                      (top-fret-thickness . 7)
                      (fret-count . 3)))
      \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
>>


\paper {
  ragged-right = ##t
  system-system-spacing.basic-distance = 20
}
```

```
\layout {
  \context {
    \Score
    \override SpacingSpanner.spacing-increment = 3
  }
}
```



Tablas alternativas de diagramas de posiciones

Se pueden crear tablas alternativas de diagramas de posiciones. Se utilizarían para tener diagramas altenativos para un acorde dado. Para usar una tabla alternativa de diagramas de posiciones, se debe crear la tabla primero. Después se añaden los diagramas a la tabla.

La tabla de diagramas de posiciones que se crea puede estar vacía o se puede copiar a partir de una tabla existente.

La tabla a usar en la impresión de los diagramas predefinidos se selecciona por medio de la propiedad `\predefinedDiagramTable`.

```
\include "predefined-guitar-fretboards.ly"

% Make a blank new fretboard table.
#(define custom-fretboard-table-one
   (make-fretboard-table))

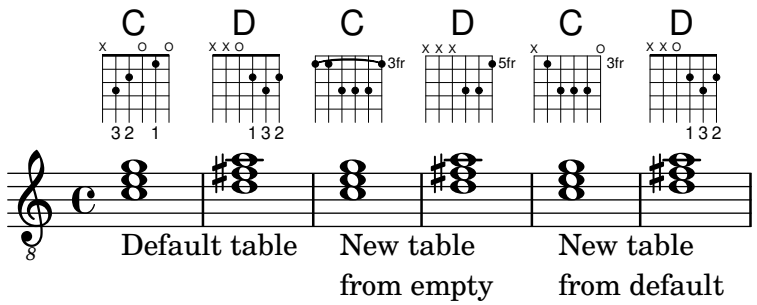
% Make a new fretboard table as a copy of `default-fret-table`.
```

```
#(define custom-fretboard-table-two
   (make-fretboard-table default-fret-table))

% Add a chord to `custom-fretboard-table-one`.
\storePredefinedDiagram #custom-fretboard-table-one
                        \chordmode {c}
                        #guitar-tuning
                        "3-(;3;5;5;5;3-);"

% Add a chord to `custom-fretboard-table-two`.
\storePredefinedDiagram #custom-fretboard-table-two
                        \chordmode {c}
                        #guitar-tuning
                        "x;3;5;5;5;o;"

<<
  \chords {
    c1 | d1 |
    c1 | d1 |
    c1 | d1 |
  }
  \new FretBoards {
    \chordmode {
      \set predefinedDiagramTable = #default-fret-table
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-one
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-two
      c1 | d1 |
    }
  }
  \new Staff {
    \clef "treble_8"
    <<
      \chordmode {
        c1 | d1 |
        c1 | d1 |
        c1 | d1 |
      }
      {
        s1_\markup "Default table" | s1 |
        s1_\markup \column { "New table" "from empty" } | s1 |
        s1_\markup \column { "New table" "from default" } | s1 |
      }
    >>
  }
>>
```



Armónicos sobre cuerdas pisadas en tablatura

El código que aparece a continuación muestra armónicos sobre cuerdas pisadas (armónicos artificiales) en una tablatura.

```
pinchedHarmonics = {
  \textSpannerDown
  \override TextSpanner.bound-details.left.text =
    \markup {\halign #-0.5 \teeny "PH" }
  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.dash-period = 0.6
  \override TextSpanner.bound-details.right.attach-dir = 1
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.padding = -0.5
}

harmonics = {
  % artificial harmonics (AH)
  \textLengthOn
  <\parenthesize b b'\harmonic>4_\markup { \teeny "AH 16" }
  <\parenthesize g g'\harmonic>4_\markup { \teeny "AH 17" }
  <\parenthesize d' d''\harmonic>2_\markup { \teeny "AH 19" }

  % pinched harmonics (PH)
  \pinchedHarmonics
  <a'\harmonic>2\startTextSpan
  <d''\harmonic>4
  <e'\harmonic>4\stopTextSpan

  % tapped harmonics (TH)
  <\parenthesize g\4 g'\harmonic>4_\markup { \teeny "TH 17" }
  <\parenthesize a\4 a'\harmonic>4_\markup { \teeny "TH 19" }
  <\parenthesize c'\3 c''\harmonic>2_\markup { \teeny "TH 17" }

  % touch harmonics (TCH)
  a4( <e''\harmonic>2. )_\markup { \teeny "TCH" }
}

frettedStrings = {
  % artificial harmonics (AH)
  \harmonicByFret 4 g4\3
  \harmonicByFret 5 d4\4
```

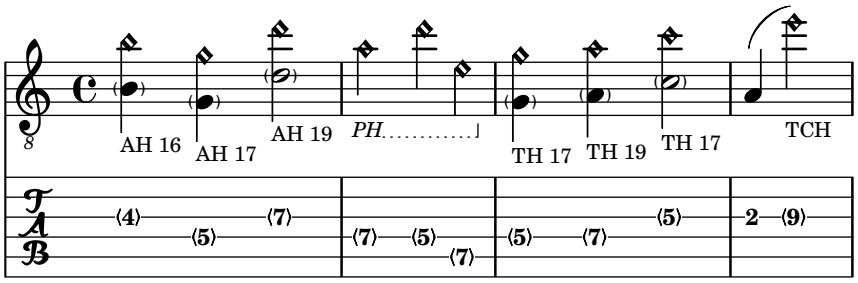
```
  \harmonicByFret 7 g2\3

  % pinched harmonics (PH)
  \harmonicByFret 7 d2\4
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 a4\5

  % tapped harmonics (TH)
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 d4\4
  \harmonicByFret 5 g2\3

  % touch harmonics (TCH)
  a4 \harmonicByFret 9 g2.\3
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \frettedStrings
      }
    }
  >>
}
```



Ligaduras de guitarra

A diferencia de los glissandos, los 'slides' o ligaduras pueden partir de un punto impreciso del mástil hasta un traste específico. Una buena forma de hacerlo es añadir una nota de mordente oculta antes de la nota real, como se muestra en el ejemplo siguiente.

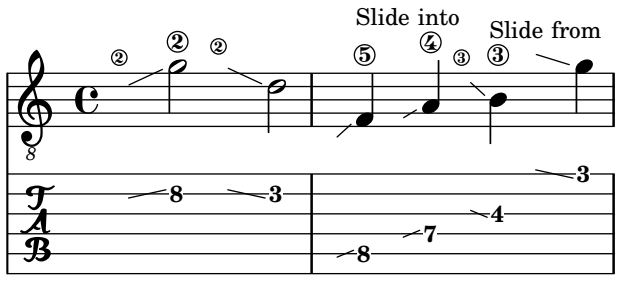
```
% Hide fret number: useful to draw slide into/from a casual point of
% the fretboard.

hideFretNumber = {
  \once \hide TabNoteHead
  \once \hide NoteHead
  \once \omit Stem
  \once \omit Flag
  \once \override NoteHead.no-ledgers = ##t
  \once \override Glissando.bound-details.left.padding = #0.3
}

music= \relative c' {
  \grace { \hideFretNumber d8\2 \glissando s2 } g2\2
  \grace { \hideFretNumber g8\2 \glissando s2 } d2 |

  \grace { \hideFretNumber c,8 \glissando s }
    f4\5^\markup \tiny { Slide into }
  \grace { \hideFretNumber f8 \glissando s } a4\4
  \grace { \hideFretNumber e'8\3 \glissando s }
    b4\3^\markup \tiny { Slide from }
  \grace { \hideFretNumber b'8 \glissando s2 } g4 |
}

\score {
  <<
    \new Staff {
      \clef "G_8"
      \music
    }
    \new TabStaff {
      \music
    }
  >>
}
```

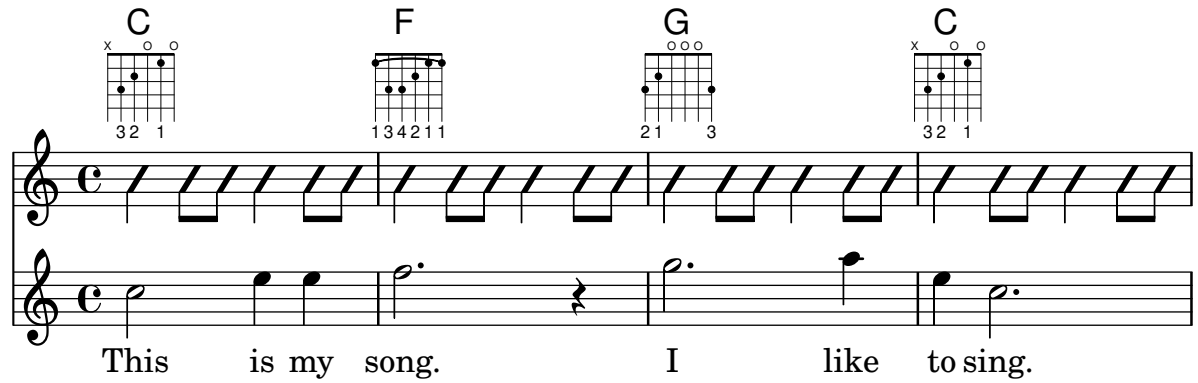


Ritmos rasgueados de guitarra

Para la música de guitarra es posible mostrar los ritmos de rasgueo, además de las notas de la melodía, acordes y diagramas de posiciones.

```
\include "predefined-guitar-fretboards.ly"

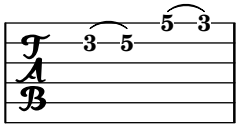
<<
  \new ChordNames \chordmode {
    c1 | f | g | c
  }
  \new FretBoards \chordmode {
    c1 | f | g | c
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
  \new Voice = "melody" \relative c'' {
    c2 e4 e4
    f2. r4
    g2. a4
    e4 c2.
  }
  \new Lyrics \lyricsto "melody" {
    This is my song.
    I like to sing.
  }
>>
```



Ligados ascendentes y descendentes (*hammer-on* y *pull-off*)

Se pueden obtener ligados ascendentes o de entrada y ligados descendentes o de salida (hammer-on y pull-off) usando ligaduras de expresión.

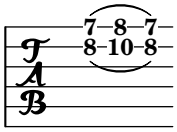
```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Indicaciones de ligado ascendente y descendente (*hammer-on* y *pull-off*) usando acordes

Cuando se usan ligados ascendentes o de entrada (*hammer-on* en inglés) o tirones de salida (*pull-off* en inglés) con notas en acorde, se dibuja un solo arco. Sin embargo son posibles "dobles arcos" si se fija la propiedad `doubleSlurs` a `#t`.

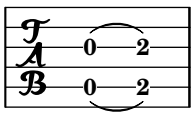
```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



Indicaciones de *hammer-on* y *pull-off* usando voces

El arco del ligado ascendente o de entrada (hammer-on) y descendente o de salida (pull-off) apunta hacia arriba en las voces uno y tres y hacia abajo en las voces dos y cuatro:

```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Cómo cambiar la posición de un diagrama de posiciones

Si queremos mover un diagrama de posiciones de acorde, por ejemplo, para evitar la colisión, o situarlo entre dos notas, tenemos varias posibilidades.

1) Modificar el valor de las propiedades `padding` (relleno) o `extra-offset` (desplazamiento adicional) como se muestra en la primera línea.

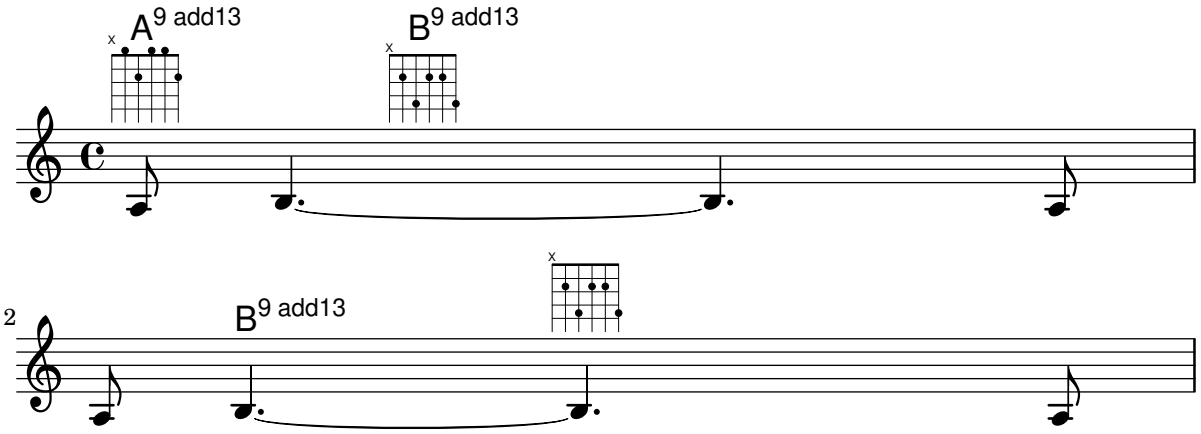
2) Podemos añadir una voz invisible y adjuntar los diagramas de posiciones a las notas invisibles de esa voz (como se ve en la segunda línea).

Si tenemos que mover el diagrama según una posición rítmica dentro del compás (en el ejemplo, la tercera parte del compás) es mejor el segundo ejemplo, porque el diagrama se alinea con el tercer pulso por sí solo.

```
harmonies = \chordmode
{
  a8:13
  \once \override ChordNames.ChordName.extra-offset = #'(10 . 0)
  b8:13 s4. |
  s2 b2:13
}

\score {
  <<
    \new ChordNames \harmonies
    \new Staff {
      % Method 1.
      a8^\markup \fret-diagram "6-x;5-0;4-2;3-0;2-0;1-2;"
      \once \override TextScript.extra-offset = #'(10 . 0)
      b4.~^\markup \fret-diagram "6-x;5-2;4-4;3-2;2-2;1-4;"
        b4. a8 | \break

      % Method 2.
      <<
        { a8 b4.~ b4. a8 }
        { s2 s2^\markup \fret-diagram  "6-x;5-2;4-4;3-2;2-2;1-4;" }
      >> |
    }
  >>
}
```



Plantilla para combo de jazz

Ésta es una plantilla bastante avanzada, para un conjunto de jazz. Observe que la notación de todos los instrumentos usa `\key c \major` (Do mayor). Esto se refiere al tono de concierto; la armadura se transporta automáticamente si la música está dentro de una sección `\transpose`.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup \column {
```

```
              "LilyPond example file by Amelie Zapf,"
              "Berlin 07/07/2003" }
}

% To make the example display properly in the documentation.
\paper {
  paper-width = 130\mm
  paper-height = 205\mm
}

% #(set-global-staff-size 16)

\include "english.ly"


%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%%%%

sl = { \override NoteHead.style = #'slash
       \hide Stem }
nsl = { \revert NoteHead.style
        \undo \hide Stem }
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

% Insert chord name style stuff here.

jazzChords = { }


%%%%%%%%%%% Keys'n'thangs %%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ############ Horns ############

% ------ Trumpet ------
trpt = \transpose c d \relative c'' {
  \Key
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  \trpt
}

% ------ Alto Saxophone ------
```

```
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble
  \alto
}

% ------ Baritone Saxophone ------
bari = \transpose c a' \relative c {
  \Key
  c1 | c1 |
  \sl d4^"Solo" d d d \nsl |
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords
  s1 | s |
  d2:maj e:m7 |
}
bariSax = {
  \global
  \clef treble
  \bari
}

% ------ Trombone ------
tbone = \relative c {
  \Key
  c1 | c | c |
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
  \clef bass
  \tbone
}

% ############ Rhythm Section #############

% ------ Guitar ------
gtr = \relative c'' {
  \Key
  c1 |
  \sl b4 b b b \nsl |
  c1 |
```

```
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 | c2:min7+ d2:maj9 | s1 |
}
guitar = {
  \global
  \clef treble
  \gtr
}

%% ------ Piano ------
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c |
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e |
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g |
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c |
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
  >>
}
PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = <<
```

```
  \new Staff = "upper" \PianoRH
  \new Staff = "lower" \PianoLH
>>


% ------ Bass Guitar ------
Bass = \relative c {
  \Key
  c1 | c | c |
}
bass = {
  \global
  \clef bass
  \Bass
}

% ------ Drums ------
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
}
down = \drummode {
  \voiceTwo
  bd4 s bd s |
  bd4 s bd s |
  bd4 s bd s |
}

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

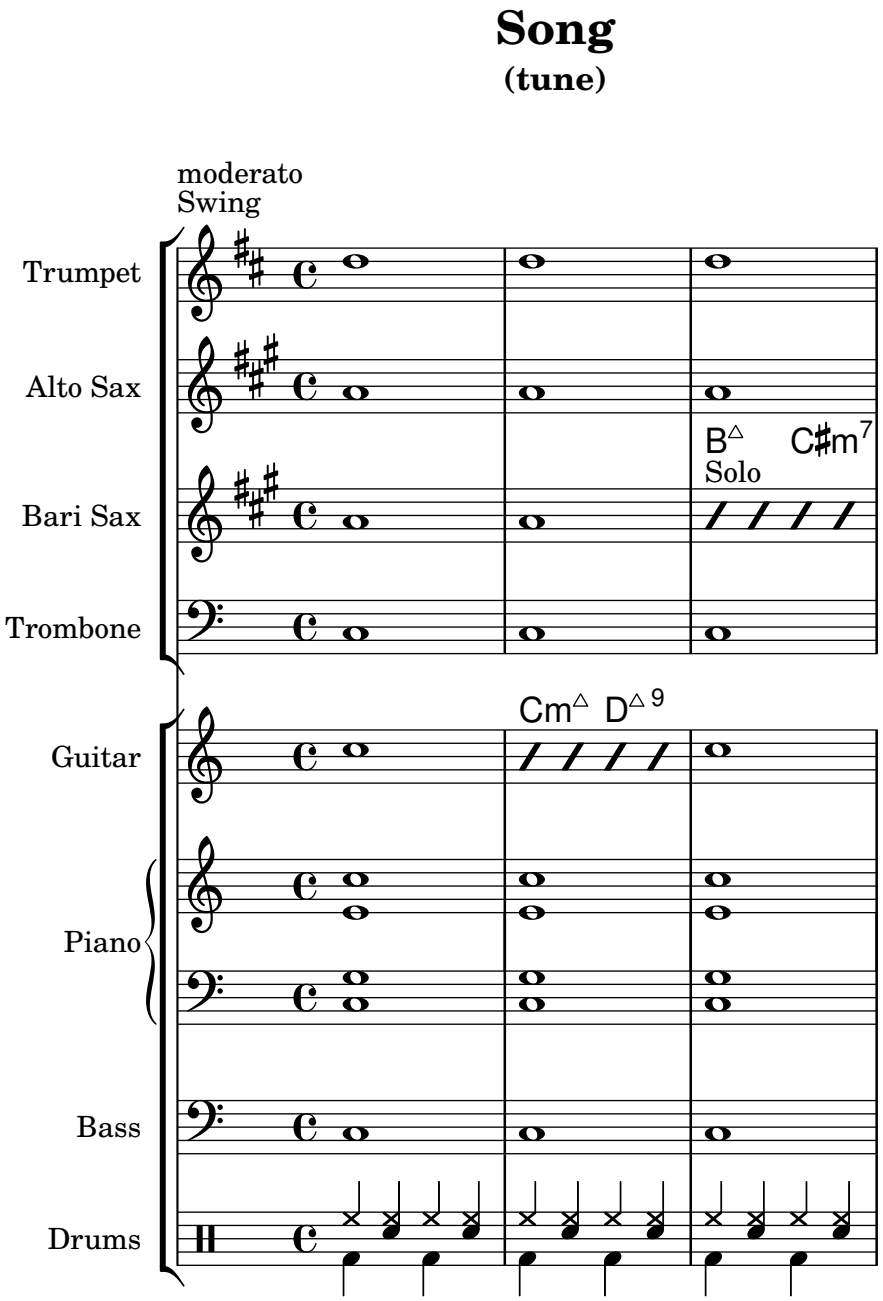
%%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%%%

\book { % For the LilyPond documentation.
  \score {
    <<
      \new StaffGroup = "horns" <<
        \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
          \trumpet
        \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
          \altoSax
        \new ChordNames = "barichords" \with { instrumentName = "Bari Sax" }
          \bariHarmony
        \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
          \bariSax
```

```
        \new Staff = "trombone" \with { instrumentName = "Trombone" }
          \trombone
      >>

      \new StaffGroup = "rhythm" <<
        \new ChordNames = "chords" \with { instrumentName = "Guitar" }
          \gtrHarmony
        \new Staff = "guitar" \with { instrumentName = "Guitar" }
          \guitar
        \new PianoStaff = "piano" \with {
          instrumentName = "Piano"
          midiInstrument = "acoustic grand"
        } \piano
        \new Staff = "bass" \with { instrumentName = "Bass" }
          \bass
        \new DrumStaff \with { instrumentName = "Drums" }
          \drumContents
      >>
    >>

    \layout {
      \context {
        \Staff
        \RemoveEmptyStaves
      }
      \context {
        \Score
        \override BarNumber.padding = 3
        \override RehearsalMark.padding = 2
        skipBars = ##t
      }
    }
    \midi { }
  }
}
```



Ligaduras *laissez vibrer*
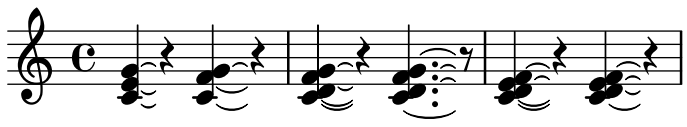
Las ligaduras "laissez vibrer" (dejar vibrar) tienen un tamaño fijo. Se puede ajustar su posición usando la propiedad `tie-configuration`.

Véase también el fragmento de código "Ligaduras laissez vibrer largas".

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
    = #`((-7 . ,DOWN)
```

```
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



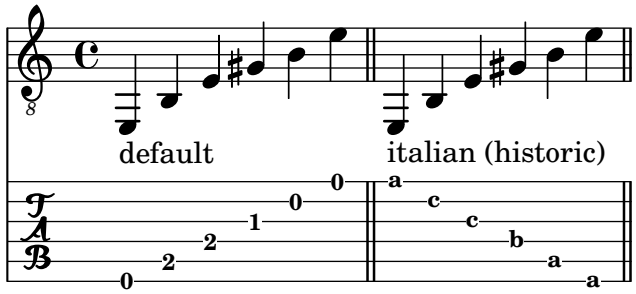
Hacer que TabStaff imprima la cuerda superior en la parte de abajo

En las tablaturas, la primera cuerda se imprime normalmente en la parte de arriba. Si queremos que esté en la parte de abajo, hay que cambiar la propiedad de contexto `stringOneTopmost` al valor `#f`. Para un ajuste al nivel del contexto, se puede hacer también esta asignacion dentro del bloque `\layout`.

```
%\layout {
%  \context {
%    \Score
%    stringOneTopmost = ##f
%  }
%  \context {
%    \TabStaff
%    tablatureFormat = #fret-letter-tablature-format
%  }
%}

m = {
  \cadenzaOn
  e, b, e gis! b e'
  \bar "||"
}

<<
  \new Staff {
    \clef "G_8"
    <>_"default" \m
    <>_"italian (historic)"\m
  }
  \new TabStaff
  {
    \m
    \set Score.stringOneTopmost = ##f
    \set TabStaff.tablatureFormat = #fret-letter-tablature-format
    \m
  }
>>
```

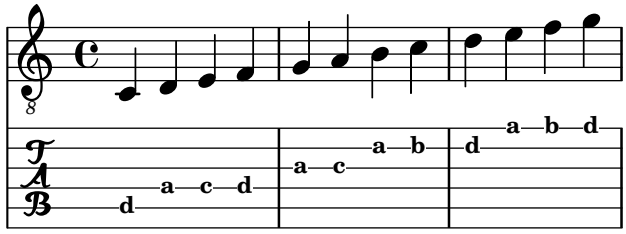


Formateado de tablaturas con letras

La tablatura se puede formatear utilizando letras en lugar de números.

```
music = \relative c {
  c4 d e f
  g4 a b c
  d4 e f g
}

<<
  \new Staff {
    \clef "G_8"
    \music
  }
  \new TabStaff \with {
    tablatureFormat = #fret-letter-tablature-format
  } {
    \music
  }
>>
```



Referencia para armónicos sobre cuerdas al aire

Referencia para armónicos sobre cuerdas al aire (armónicos naturales).

```
openStringHarmonics = {
  \textSpannerDown
  \override TextSpanner.staff-padding = 3
  \override TextSpanner.dash-fraction = 0.3
  \override TextSpanner.dash-period = 1

  % first harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "1st harm. "
  \harmonicByFret 12 e,2\6\startTextSpan
  \harmonicByRatio #1/2 e,\6\stopTextSpan
```

```
% second harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "2nd harm. "
\harmonicByFret 7 e,\6\startTextSpan
\harmonicByRatio #1/3 e,\6
\harmonicByFret 19 e,\6
\harmonicByRatio #2/3 e,\6\stopTextSpan
%\harmonicByFret 19 < e,\6 a,\5 d\4 >
%\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >

% third harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "3rd harm. "
\harmonicByFret 5 e,\6\startTextSpan
\harmonicByRatio #1/4 e,\6
\harmonicByFret 24 e,\6
\harmonicByRatio #3/4 e,\6\stopTextSpan
\break

% fourth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "4th harm. "
\harmonicByFret 4 e,\6\startTextSpan
\harmonicByRatio #1/5 e,\6
\harmonicByFret 9 e,\6
\harmonicByRatio #2/5 e,\6
\harmonicByFret 16 e,\6
\harmonicByRatio #3/5 e,\6\stopTextSpan

% fifth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "5th harm. "
\harmonicByFret 3 e,\6\startTextSpan
\harmonicByRatio #1/6 e,\6\stopTextSpan
\break

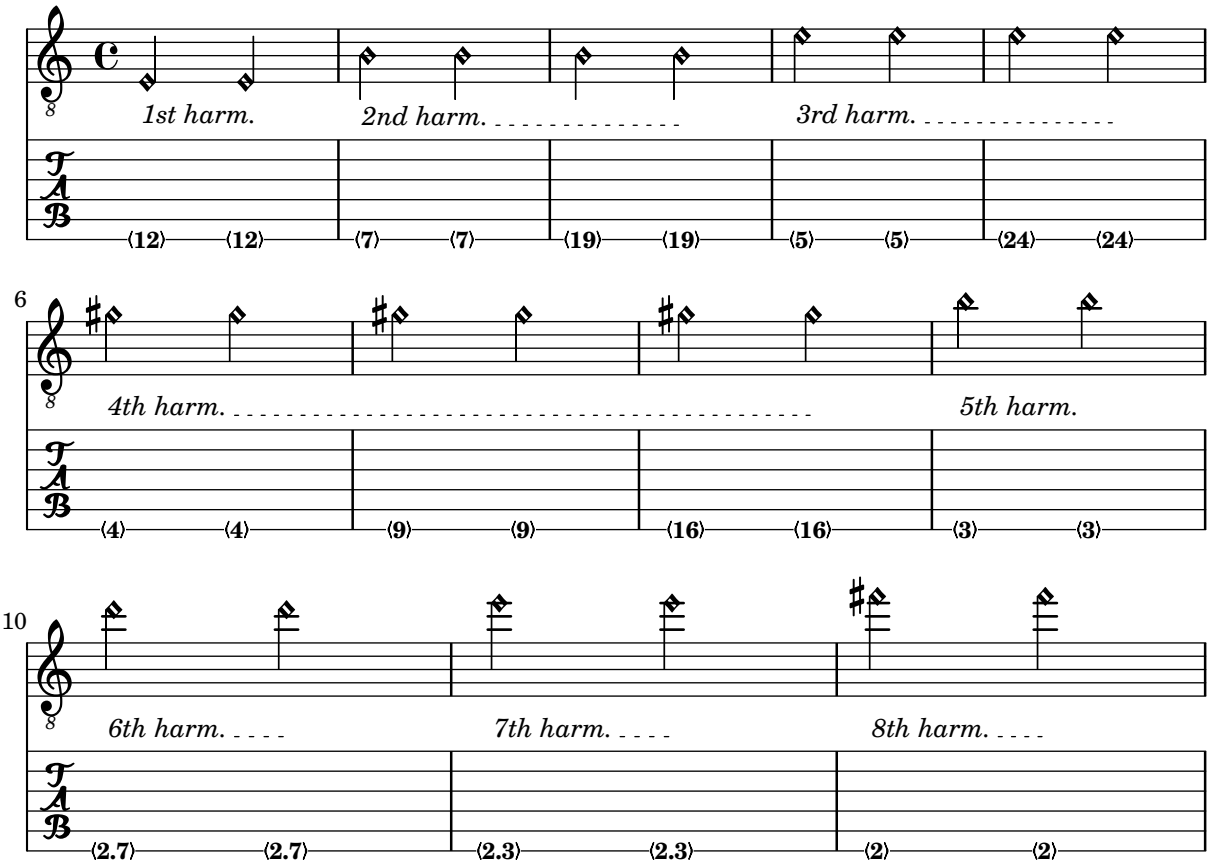
% sixth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "6th harm. "
\harmonicByFret 2.7 e,\6\startTextSpan
\harmonicByRatio #1/7 e,\6\stopTextSpan

% seventh harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "7th harm. "
\harmonicByFret 2.3 e,\6\startTextSpan
\harmonicByRatio #1/8 e,\6\stopTextSpan

% eighth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "8th harm. "
\harmonicByFret 2 e,\6\startTextSpan
```

```
  \harmonicByRatio #1/9 e,\6\stopTextSpan
}

\score {
  <<
    \new Staff \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \openStringHarmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \openStringHarmonics
      }
    }
  >>
}
```



Posicionamiento de digitaciones de mano derecha

Es posible ejercer un mayor control sobre la colocación de las digitaciones de la mano derecha estableciendo el valor de una propiedad específica, como se muestra en el ejemplo siguiente.

```
#(define RH rightHandFinger)

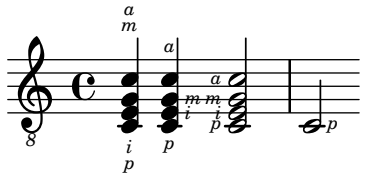
\relative c {
  \clef "treble_8"

  \set strokeFingerOrientations = #'(up down)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >4

  \set strokeFingerOrientations = #'(left)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >2

  \set strokeFingerOrientations = #'(right)
  c\RH 1
}
```



Polifonía en tablaturas

La polifonía se crea de la misma forma en un TabStaff que en una pauta normal.

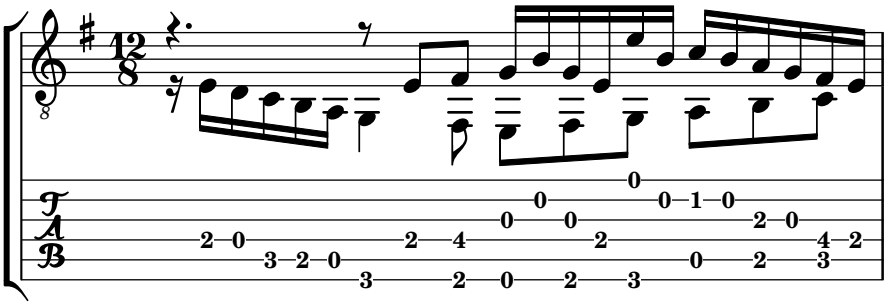
```
upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  \new StaffGroup = "tab with traditional" <<
    \new Staff = "guitar traditional" <<
      \clef "treble_8"
      \new Voice = "upper" \upper
      \new Voice = "lower" \lower
    >>

    \new TabStaff = "guitar tab" <<
      \new TabVoice = "upper" \upper
      \new TabVoice = "lower" \lower
    >>
```

```
  >>
}
```



Setting up predefined fretboards for other instruments

Predefined fret diagrams can be added for new instruments in addition to the standard diagrams used for guitar. This file shows how this is done by defining a new string tuning and a few predefined fretboards for the Venezuelan *cuatro*.

This file also shows how fingerings can be included in the chords used as reference points for the chord lookup, and displayed in the fret diagram and the `TabStaff`, but not the music.

These fretboards are not transposable because they contain string information. This is planned to be corrected in the future.

```
% Add fretboards for the cuatro.
%
%   Note: This section could be put into a separate file
%         `predefined-cuatro-fretboards.ly`
%         and be \included into each of your compositions.

cuatroTuning = #`(,(ly:make-pitch 0 6 0)
                  ,(ly:make-pitch 1 3 SHARP)
                  ,(ly:make-pitch 1 1 0)
                  ,(ly:make-pitch 0 5 0))

dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }
aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }

\storePredefinedDiagram #default-fret-table \dSix
                        #cuatroTuning
                        "o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
                        #cuatroTuning
                        "o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
                        #cuatroTuning
                        "o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven
                        #cuatroTuning
                        "o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
```

```
                        #cuatroTuning
                        "2-2;o;1-1;o;"

% End of potential include file `predefined-cuatro-fretboards.ly`.


#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }

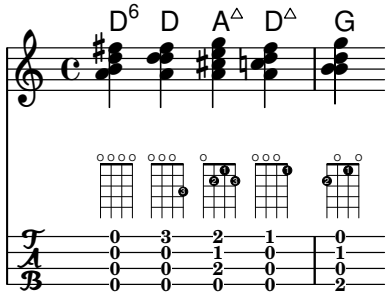
    \new FretBoards {
      \set Staff.stringTunings = #cuatroTuning
%      \override FretBoard
%        #'(fret-diagram-details string-count) = 4
      \override FretBoard.fret-diagram-details.finger-code = #'in-dot
      \primeros
    }

    \new TabStaff \relative c'' {
      \set TabStaff.stringTunings = #cuatroTuning
      \primeros
    }

  >>

  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration =
```

```
        \musicLength 16
    }
  }
  \midi { }
}
```



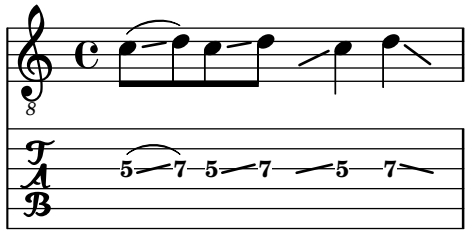
Deslizamientos en tablatura

Los deslizamientos se pueden componer tipográficamente tanto en los contextos de `Staff` como en los de `TabStaff`.

```
slides = {
  c'8\3(\glissando d'8\3)
  c'8\3\glissando d'8\3
  \hideNotes
  \grace { g16\glissando }
  \unHideNotes
  c'4\3
  \afterGrace d'4\3\glissando {
  \stemDown \hideNotes
  g16 }
  \unHideNotes
}

\score {
  <<
    \new Staff { \clef "treble_8" \slides }
    \new TabStaff { \slides }
  >>

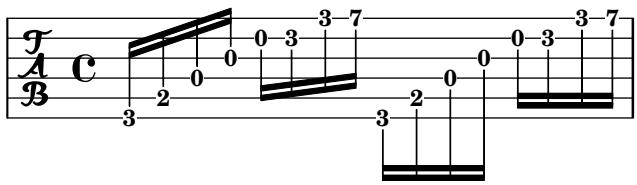
  \layout {
    \context {
      \Score
      \override Glissando.minimum-length = 4
      \override Glissando.springs-and-rods =
                  #ly:spanner::set-spacing-rods
      \override Glissando.thickness = 2
      \omit StringNumber
      % or:
      %\override StringNumber.stencil = ##f
    }
  }
}
```



Comportamiento de las plicas y las barras de corchea en tablaturas

La dirección de las plicas se controla de la misma forma en la tablatura que en la notación tradicional. Las barras se pueden poner horizontales, como se muestra en este ejemplo.

```
\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
    \override Beam.concaveness = 10000
    g,,16 b d g b d g b
  }
}
```
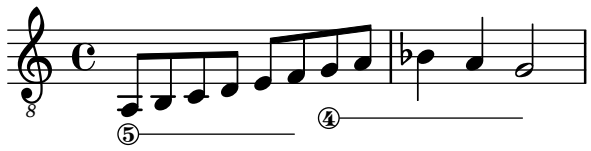


Líneas de extensión para números de cuerda

Hacemos una línea extensora para las indicaciones de número de cuerda, motrando que una serie de notas se han de tocar sobre la misma cuerda.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})

\relative c {
  \clef "treble_8"
  \textSpannerDown
  \stringNumberSpanner "5" a8\startTextSpan b c d
    e f\stopTextSpan \stringNumberSpanner "4" g\startTextSpan a |
  bes4 a g2\stopTextSpan
}
```


5
4

13 Percussion

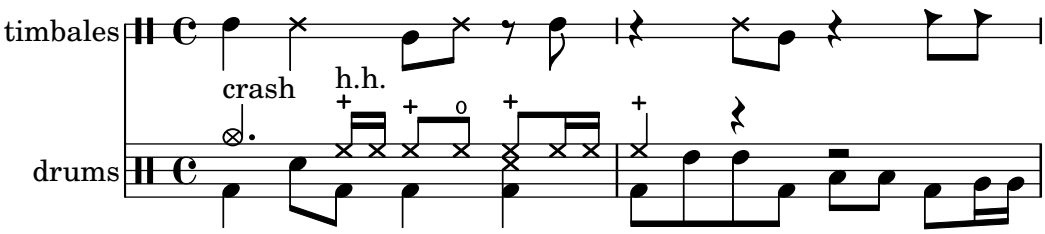
See also Sección “Percussion” en *Notation Reference*.

Escritura de partes de percusión

Mediante la utilización de las potentes herramientas preconfiguradas como la función `\drummode` y el contexto `DrumStaff`, la introducción de partes para percusión es muy fácil: las percusiones se sitúan en sus propias posiciones de pentagrama (con una clave especial) y tienen las cabezas correspondientes al instrumento. Es posible añadir un símbolo adicional a la percusión o reducir el número de líneas.

```
drh = \drummode {
        cymc4.^"crash" hhc16^"h.h." hh hhc8 hho hhc8 hh16 hh
        hhc4 r4 r2
      }
drl = \drummode {
        bd4 sn8 bd bd4 << bd ss >>
        bd8 tommh tommh bd toml toml bd tomfh16 tomfh
      }
timb = \drummode {
         timh4 ssh timl8 ssh r timh r4
         ssh8 timl r4 cb8 cb
       }

\score {
  <<
    \new DrumStaff \with {
      instrumentName = "timbales"
      drumStyleTable = #timbales-style
      \override StaffSymbol.line-count = #2
      \override BarLine.bar-extent = #'(-1 . 1)
    }
    <<
      \timb
    >>
    \new DrumStaff \with { instrumentName = "drums" }
    <<
      \new DrumVoice { \stemUp \drh }
      \new DrumVoice { \stemDown \drl }
    >>
  >>
  \layout { }
  \midi { \tempo 4 = 120 }
}
```



Ejemplo de cencerro y campana

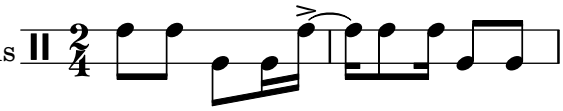
Dos campanas diferentes, escritas con 'cb' (cencerro) y 'rb' (campana, ridebell).

```
#(define mydrums '((ridebell default #f  3)
                  (cowbell  default #f -2)))

\new DrumStaff \with { instrumentName = #"Different Bells" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \set DrumStaff.clefPosition = 0.5
  \override DrumStaff.StaffSymbol.line-positions = #'(-2 3)
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)

  \time 2/4
  rb8 8 cb8 16 rb16-> ~ |
  16 8 16 cb8 8 |
}
```



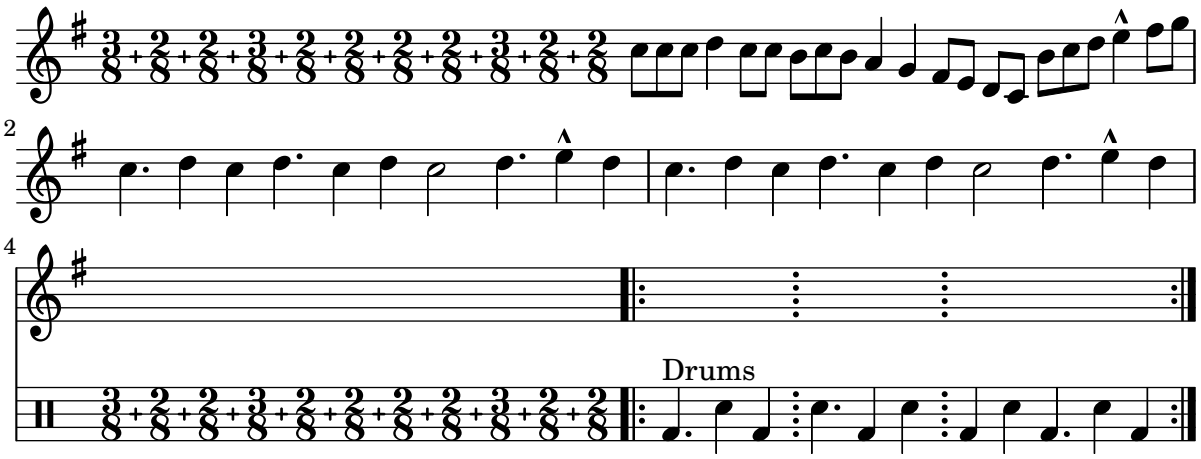
Indicaciones de compás polimétricas profundamente personalizadas

Aunque aquí el elemento más esencial no era la indicación de compás polimétrica que se muestra, se ha incluido para mostrar el pulso de la pieza, que por cierto es la plantilla de una canción de los Balcanes real.

```
melody = \relative c'' {
  \key g \major
  \time #'((3 . 8) (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8)
           (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8))
  \set Timing.beamExceptions = #'()
  \set Timing.beatStructure = 3,2,2,3,2,2,2,2,3,2,2
  c8 c c d4 c8 c b c b a4 g fis8 e d c b' c d e4-^ fis8 g \break
  c,4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
}

drum = \new DrumStaff \drummode {
  \repeat volta 2 {
    bd4.^\markup { Drums } sn4 bd \bar ";"
    sn4. bd4 sn \bar ";"
    bd sn bd4. sn4 bd
  }
}

\new Staff {
  \melody
  \drum
}
```



Ejemplo de cajas chinas grave y aguda

Dos cajas chinas, escritas como 'wbh' (*high woodblock*) y 'wbl' (*low woodblock*). La longitud de la línea divisoria se ha alterado con una instrucción \override para que no sea demasiado corta. Las posiciones de las dos líneas de la pauta se tienen que definir explícitamente también.

```
% These lines define the position of the woodblocks in the stave;
% if you like, you can change it or you can use special note heads
% for the woodblocks.
#(define mydrums '((hiwoodblock default #f  3)
                   (lowoodblock default #f -2)))

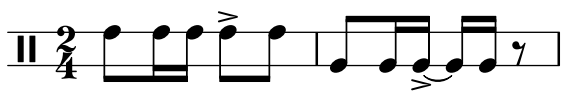
woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

  % This is necessary; if not entered,
  % the barline would be too short!
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)
  % small correction for the clef:
  \set DrumStaff.clefPosition = 0.5
}

\new DrumStaff {
  % with this you load your new drum style table
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \woodstaff

  \drummode {
    \time 2/4
    wbh8 16 16 8-> 8 |
    wbl8 16 16-> ~ 16 16 r8 |
  }
}
```



Plantilla para combo de jazz

Ésta es una plantilla bastante avanzada, para un conjunto de jazz. Observe que la notación de todos los instrumentos usa \key c \major (Do mayor). Esto se refiere al tono de concierto; la armadura se transporta automáticamente si la música está dentro de una sección \transpose.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup \column {
              "LilyPond example file by Amelie Zapf,"
              "Berlin 07/07/2003" }
}

% To make the example display properly in the documentation.
\paper {
  paper-width = 130\mm
  paper-height = 205\mm
}

% #(set-global-staff-size 16)

\include "english.ly"


%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%%%%

sl = { \override NoteHead.style = #'slash
       \hide Stem }
nsl = { \revert NoteHead.style
        \undo \hide Stem }
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

% Insert chord name style stuff here.

jazzChords = { }


%%%%%%%%%%% Keys'n'thangs %%%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ############ Horns ############

% ------ Trumpet ------
trpt = \transpose c d \relative c'' {
  \Key
```

```
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  \trpt
}

% ------ Alto Saxophone ------
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble
  \alto
}

% ------ Baritone Saxophone ------
bari = \transpose c a' \relative c {
  \Key
  c1 | c1 |
  \sl d4^"Solo" d d d \nsl |
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords
  s1 | s |
  d2:maj e:m7 |
}
bariSax = {
  \global
  \clef treble
  \bari
}

% ------ Trombone ------
tbone = \relative c {
  \Key
  c1 | c | c |
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
```

```
  \clef bass
  \tbone
}

% ############ Rhythm Section #############

% ------ Guitar ------
gtr = \relative c'' {
  \Key
  c1 |
  \sl b4 b b b \nsl |
  c1 |
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 | c2:min7+ d2:maj9 | s1 |
}
guitar = {
  \global
  \clef treble
  \gtr
}

%% ------ Piano ------
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c |
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e |
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g |
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c |
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
```

```
  >>
}
PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = <<
  \new Staff = "upper" \PianoRH
  \new Staff = "lower" \PianoLH
>>


% ------ Bass Guitar ------
Bass = \relative c {
  \Key
  c1 | c | c |
}
bass = {
  \global
  \clef bass
  \Bass
}

% ------ Drums ------
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
}
down = \drummode {
  \voiceTwo
  bd4 s bd s |
  bd4 s bd s |
  bd4 s bd s |
}

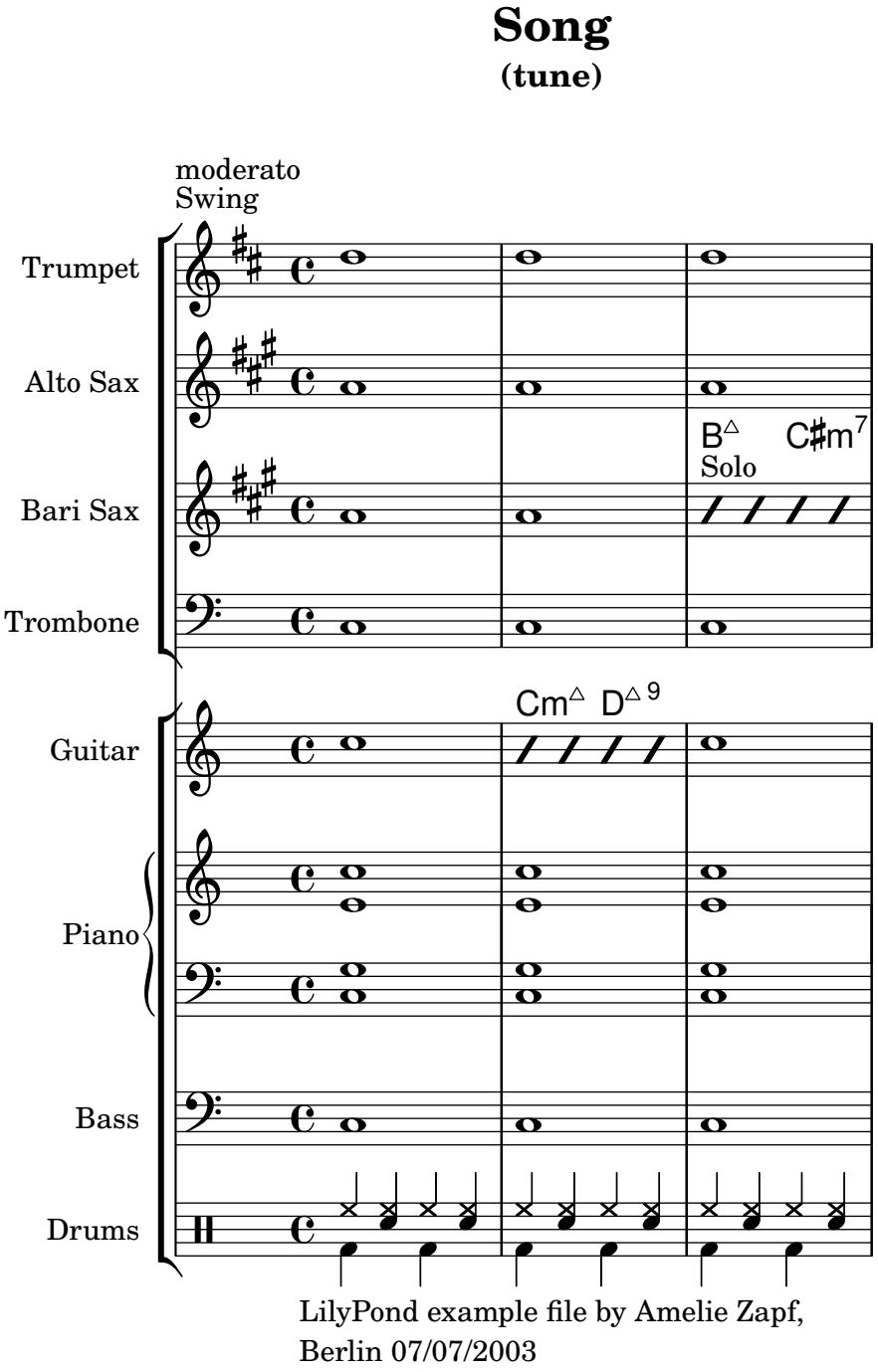
drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%%%
```

```
\book { % For the LilyPond documentation.
  \score {
    <<
      \new StaffGroup = "horns" <<
        \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
          \trumpet
        \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
          \altoSax
        \new ChordNames = "barichords" \with { instrumentName = "Bari Sax" }
          \bariHarmony
        \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
          \bariSax
        \new Staff = "trombone" \with { instrumentName = "Trombone" }
          \trombone
      >>

      \new StaffGroup = "rhythm" <<
        \new ChordNames = "chords" \with { instrumentName = "Guitar" }
          \gtrHarmony
        \new Staff = "guitar" \with { instrumentName = "Guitar" }
          \guitar
        \new PianoStaff = "piano" \with {
          instrumentName = "Piano"
          midiInstrument = "acoustic grand"
        } \piano
        \new Staff = "bass" \with { instrumentName = "Bass" }
          \bass
        \new DrumStaff \with { instrumentName = "Drums" }
          \drumContents
      >>
    >>

    \layout {
      \context {
        \Staff
        \RemoveEmptyStaves
      }
      \context {
        \Score
        \override BarNumber.padding = 3
        \override RehearsalMark.padding = 2
        skipBars = ##t
      }
    }
    \midi { }
  }
}
```



Símbolos de golpes de percusión

No están contemplados de forma nativa los símbolos gráficos para instrumentos de percusión; sin embargo es posible incluir dichos símbolos, ya sea como un archivo EPS externo o como código PostScript incrustado dentro de un elemento de marcado, como se muestra en este ejemplo.

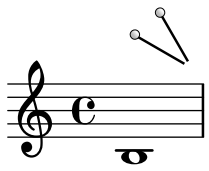
```
stick = \markup \with-dimensions #'(0.80 . 5.2) #'(0.85 . 5.2) {
  \postscript "
    0 6 translate
    0.8 -0.8 scale
    0 0 0 setrgbcolor
    [] 0 setdash
    1 setlinewidth
```

```
0 setlinejoin
0 setlinecap
gsave [1 0 0 1 0 0] concat
gsave [1 0 0 1 -3.5406095 -199.29342] concat
gsave
0 0 0 setrgbcolor
newpath
7.1434065 200.94354 moveto
7.2109628 200.90454 7.2785188 200.86554 7.3460747 200.82654 curveto
8.2056347 202.31535 9.0651946 203.80414 9.9247546 205.29295 curveto
9.8571989 205.33195 9.7896429 205.37095 9.7220864 205.40996 curveto
8.8625264 203.92115 8.0029664 202.43233 7.1434065 200.94354 curveto
closepath
eofill
grestore
gsave
0 0 0 setrgbcolor
newpath
4.9646672 203.10444 moveto
5.0036707 203.03688 5.0426744 202.96933 5.0816777 202.90176 curveto
6.5704792 203.76133 8.0592809 204.6209 9.5480824 205.48045 curveto
9.5090791 205.54801 9.4700754 205.61556 9.4310717 205.68311 curveto
7.94227 204.82356 6.4534687 203.96399 4.9646672 203.10444 curveto
closepath
eofill
grestore
gsave
<<
/ShadingType 3
/ColorSpace /DeviceRGB
/Coords [113.13708 207.87465 0 113.13708 207.87465 16.162441]
/Extend [true true]
/Domain [0 1]
/Function <<
/FunctionType 3
/Functions
[
<<
/FunctionType 2
/Domain [0 1]
/C0 [1 1 1]
/C1 [0.72941178 0.72941178 0.72941178]
/N 1
>>
]
/Domain [0 1]
/Bounds [ ]
/Encode [ 0 1 ]
>>
>>
newpath
7.6422017 200.76488 moveto
```

```
7.6505696 201.02554 7.3905363 201.24867 7.1341335 201.20075 curveto
6.8759501 201.16916 6.6949602 200.87978 6.7801462 200.63381 curveto
6.8480773 200.39155 7.1438307 200.25377 7.3728389 200.35861 curveto
7.5332399 200.42458 7.6444521 200.59122 7.6422017 200.76488 curveto
closepath
clip
gsave [
  0.052859054 0.063089841 -0.020912282 0.017521108 5.7334261 189.76443
] concat
shfill
grestore
grestore
0 0 0 setrgbcolor
[] 0 setdash
0.027282091 setlinewidth
0 setlinejoin
0 setlinecap
newpath
7.6422017 200.76488 moveto
7.6505696 201.02554 7.3905363 201.24867 7.1341335 201.20075 curveto
6.8759501 201.16916 6.6949602 200.87978 6.7801462 200.63381 curveto
6.8480773 200.39155 7.1438307 200.25377 7.3728389 200.35861 curveto
7.5332399 200.42458 7.6444521 200.59122 7.6422017 200.76488 curveto
closepath
stroke
gsave
<<
/ShadingType 3
/ColorSpace /DeviceRGB
/Coords [113.13708 207.87465 0 113.13708 207.87465 16.162441]
/Extend [true true]
/Domain [0 1]
/Function <<
/FunctionType 3
/Functions
[
<<
/FunctionType 2
/Domain [0 1]
/C0 [1 1 1]
/C1 [0.72941178 0.72941178 0.72941178]
/N 1
>>
]
/Domain [0 1]
/Bounds [ ]
/Encode [ 0 1 ]
>>
>>
newpath
5.2721217 202.83181 moveto
5.2804896 203.09247 5.0204563 203.3156 4.7640539 203.26768 curveto
```

```
    4.5058701 203.23609 4.3248803 202.94671 4.4100662 202.70074 curveto
    4.4779975 202.45848 4.7737511 202.3207 5.0027593 202.42554 curveto
    5.1631598 202.49149 5.2743721 202.65813 5.2721217 202.83181 curveto
    closepath
    clip
    gsave [
      0.052859054 0.063089841 -0.020912282 0.017521108 3.363346 191.83136
    ] concat
    shfill
    grestore
    grestore
    0 0 0 setrgbcolor
    [] 0 setdash
    0.027282091 setlinewidth
    0 setlinejoin
    0 setlinecap
    newpath
    5.2721217 202.83181 moveto
    5.2804896 203.09247 5.0204563 203.3156 4.7640539 203.26768 curveto
    4.5058701 203.23609 4.3248803 202.94671 4.4100662 202.70074 curveto
    4.4779975 202.45848 4.7737511 202.3207 5.0027593 202.42554 curveto
    5.1631598 202.49149 5.2743721 202.65813 5.2721217 202.83181 curveto
    closepath
    stroke
    grestore
    grestore
  "
}

\score {
  b1^\stick
}
```



Ejemplo de percusión

Un ejemplo breve extraído de la *Historia de un soldado* de Strawinsky.

```
#(define mydrums '((bassdrum   default #f  4)
                  (snare      default #f -4)
                  (tambourine default #f  0)))

U = \stemUp
D = \stemDown

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
```

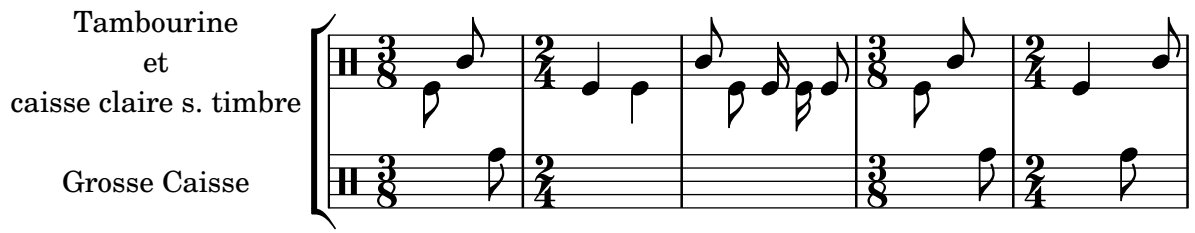
```
  \time 2/4 s2
}

drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
      sn4 \D sn4 |
      \U tamb8 \D sn \U sn16 \D sn \U sn8 |
      \D sn8 \U tamb s |
      \U sn4 s8 \U tamb
    }
  >>
}

drumsB = \drummode {
  s4 bd8 s2*2 s4 bd8 s4 bd8 s
}

\layout {
  indent = 40\mm
  \context {
    \DrumStaff
    drumStyleTable = #(alist->hash-table mydrums)
  }
}

\score {
  \new StaffGroup <<
    \new DrumStaff \with {
      instrumentName = \markup \center-column {
        "Tambourine"
        "et"
        "caisse claire s. timbre" }
    } \drumsA
    \new DrumStaff \with {
      instrumentName = "Grosse Caisse"
    }\drumsB
  >>
}
```



Imprimir música que tenga compases distintos en cada pentagrama

En el siguiente fragmento de código, dos partes distintas tienen un compás completamente diferente y sin embargo se mantienen sincronizadas. Las barras de compás ya no se pueden imprimir en el nivel de `Score`; para permitir barras de compás independientes en cada parte se trasladan los grabadores `Default_barline_engraver` y `Timing_translator` desde el contexto de partitura `Score` hasta el contexto de pentagrama `Staff`.

Si son necesarios números de compás, el grabador de números de compás `Bar_number_engraver` también debe trasladarse, ya que descansa en propiedades fijadas por el `Timing_translator`; se puede usar un bloque `\with` para añadir números de compás al pentagrama apropiado.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}

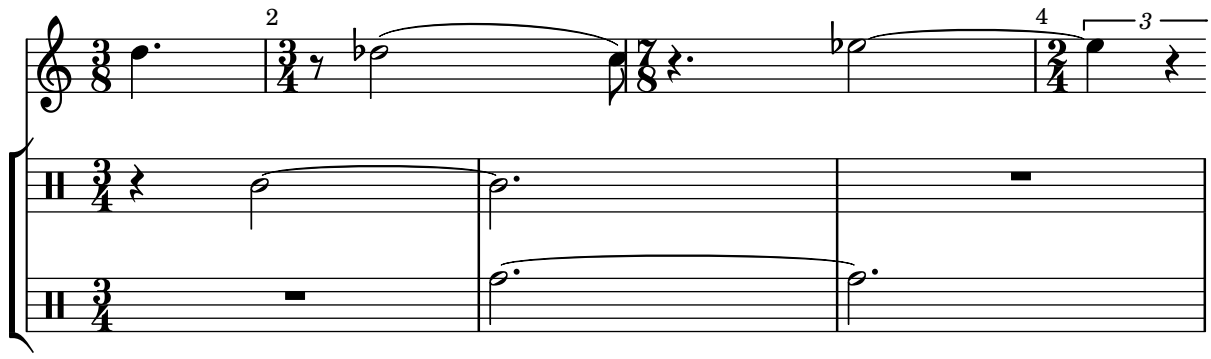
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

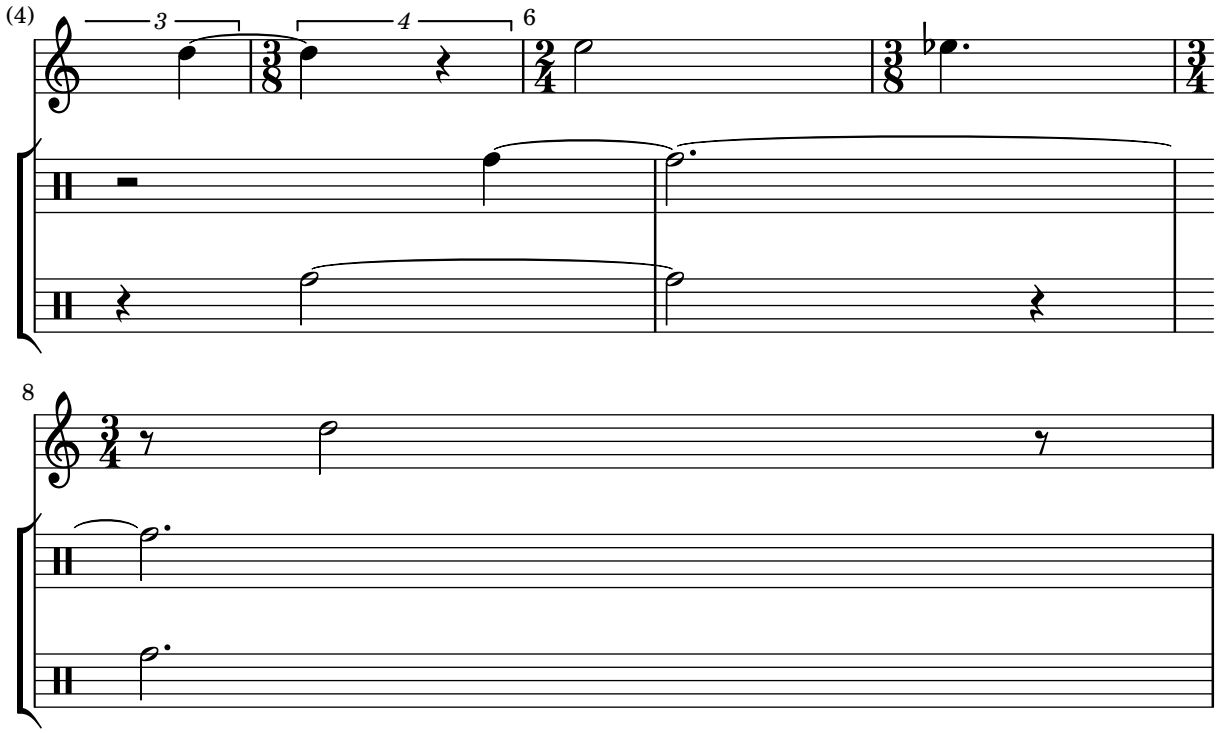
Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. ees''2 ~ |
    \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
    \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
    \time 2/4 e''2 |
    \time 3/8 es''4. |
    \time 3/4 r8 d''2 r8 |
```

```
  }
>>

Perkussion = \new StaffGroup <<
  \new Staff <<
    \global
    {
      \clef percussion
      \time 3/4 r4 c'2 ~ |
      c'2. |
      R2. |
      r2 g'4 ~ |
      g'2. ~ |
      g'2. |
    }
  >>
  \new Staff <<
    \global {
      \clef percussion
      \time 3/4 R2. |
      g'2. ~ |
      g'2. |
      r4 g'2 ~ |
      g'2 r4 |
      g'2. |
    }
  >>
>>

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}
```





Ejemplo de música para tam-tam

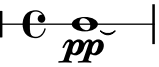
Un ejemplo de tam-tam, escrito como 'tt'.

```
#(define mydrums '((tamtam default #f 0)))

\new DrumStaff \with { instrumentName = #"Tamtam" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  tt 1 \pp \laissezVibrer
}
```

Tamtam 

Ejemlo de pandereta

A tambourine example, entered with 'tamb'.

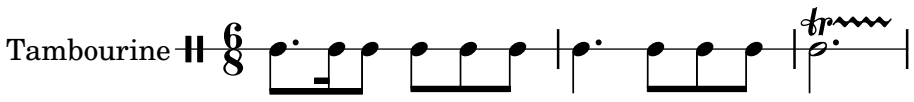
```
#(define mydrums '((tambourine default #f 0)))

\new DrumStaff \with { instrumentName = #"Tambourine" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  \time 6/8
```

```
  tamb8. 16 8 8 8 8 |
  tamb4. 8 8 8 |
  % The trick with the scaled duration and the shorter rest
  % is neccessary for the correct ending of the trill-span!
  tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
}
```



14 Wind instruments

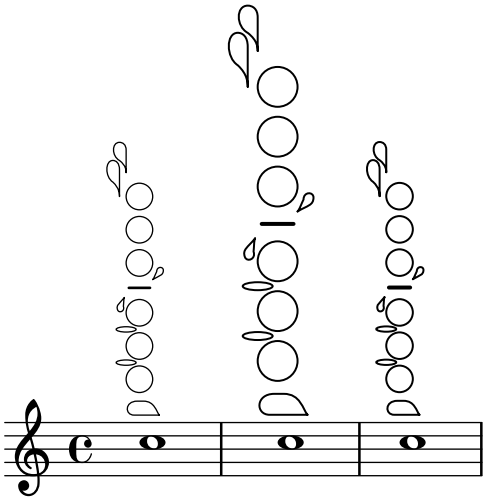
See also Sección “Wind instruments” en *Notation Reference*.

Modificar el tamaño de los diagramas de viento madera

Se puede cambiar el tamaño y grosor de las líneas de los diagramas de posiciones para instrumentos de viento madera.

```
\relative c'' {
  \textLengthOn
  c1^\markup
       \woodwind-diagram #'piccolo #'()

  c^\markup \override #'(size . 1.5)
      \woodwind-diagram #'piccolo #'()
  c^\markup \override #'(thickness . 0.15)
      \woodwind-diagram #'piccolo #'()
}
```



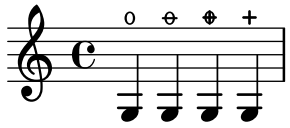
Símbolos de digitación para instrumentos de viento

Se pueden conseguir símbolos especiales combinando glifos existentes, lo que es de utilidad para la notación de instrumentos de viento.

```
lineup =
  \tweak outside-staff-padding #0
  \tweak staff-padding #0
  \tweak padding #0.2
  \tweak parent-alignment-X #CENTER
  \tweak self-alignment-X #CENTER
  \etc

\relative c' {
  g\open
  g\lineup ^\markup \combine
              \musicglyph "scripts.open"
              \musicglyph "scripts.tenuto"
  g\lineup ^\markup \combine
```

```
              \musicglyph "scripts.open"
              \musicglyph "scripts.stopped"
  g\stopped
}
```
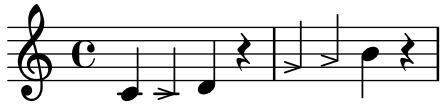


Notación de los golpes de lengua de la flauta

Es posible indicar técnicas de articulación especiales como el golpe de lengua de la flauta ("tongue slap") sustituyendo la cabeza de la figura con el glifo adecuado. Para ello podemos trazar la cabeza de la nota en forma de acento con \markup.

```
slap =
#(define-music-function (music) (ly:music?)
  #{
    \temporary \override NoteHead.stencil =
      #ly:text-interface::print
    \temporary \override NoteHead.text =
      \markup
        \translate #'(1 . 0)
          \override #'(thickness . 1.4)
            \overlay { \draw-line #'(-1.2 . 0.4)
                       \draw-line #'(-1.2 . -0.4) }
    \temporary \override NoteHead.stem-attachment =
      #(lambda (grob)
        (let* ((stem (ly:grob-object grob 'stem))
               (dir (ly:grob-property stem 'direction UP))
               (is-up (eqv? dir UP)))
          (cons dir (if is-up 0 -0.8))))
    #music
    \revert NoteHead.stencil
    \revert NoteHead.text
    \revert NoteHead.stem-attachment
  #})

\relative c' {
  c4 \slap c d r
  \slap { g4 a } b r
}
```

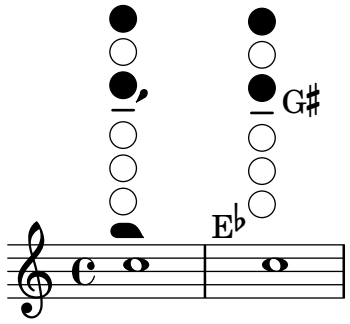


Diagramas para viento madera gráficos y textuales

En muchos casos, las llaves que no están en la columna central se pueden presentar por el nombre de la llave así como de forma gráfica.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram #'piccolo
                      #'((cc . (one three))
                         (lh . (gis))
                         (rh . (ees)))

  c^\markup
    \override #'(graphical . #f)
      \woodwind-diagram #'piccolo
                        #'((cc . (one three))
                           (lh . (gis))
                           (rh . (ees)))
}
```



Diagramas de digitación para la flauta dulce

El ejemplo siguiente muestra cómo se pueden realizar diagramas de digitación para instrumentos de viento.

```
% range chart for paetzold contrabass recorder

centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset = #(lambda (g)
    (+ (ly:self-alignment-interface::centered-on-x-parent g)
       (ly:self-alignment-interface::x-aligned-on-self g)))
}

\new Staff \with {
  \remove "Time_signature_engraver"
  \omit Stem
  \omit Flag
  \consists "Horizontal_bracket_engraver"
} {
  \clef bass
  \set Score.timing = ##f

  f,1*1/4 \glissando

  \clef violin
  gis'1*1/4
```

```
  a'4^\markup "1)"

  \centermarkup
    \once \override TextScript.padding = 2
    bes'1*1/4_\markup \override #'(baseline-skip . 1.7) \column {
      \fontsize #-5
      \slashed-digit #0 \finger 1 \finger 2
      \finger 3 \finger 4 \finger 5 \finger 6 \finger 7 }

  b'1*1/4

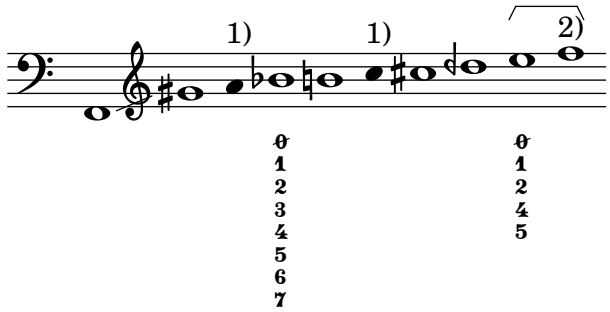
  c''4^\markup "1)"

  cis''1*1/4

  deh''1*1/4

  \centermarkup
    \once \override TextScript.padding = 2
    \once \override Staff.HorizontalBracket.direction = #UP
    e''1*1/4_\markup \override #'(baseline-skip . 1.7) \column {
      \fontsize #-5
      \slashed-digit #0 \finger 1 \finger 2
      \finger 4 \finger 5 } \startGroup

  f''1*1/4^\markup "2)" \stopGroup
}
```
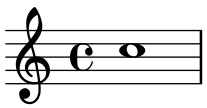


Listas de llaves para los diagramas de viento madera

El fragmento de código que aparece a continuación produce una lista de todas las llaves y disposiciones de llave posibles para los diagramas de posiciones de instrumentos de viento madera, tal y como están definidos en `scm/define-woodwind-diagrams.scm`. La lista se muestra en la salida de error estándar stderr, pero no en la música. Si desea una salida por la consola a través de la salida estándar stdout, omita el código `(current-error-port)` de las instrucciones.

```
#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
```

```
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))

\score {c''1}
```



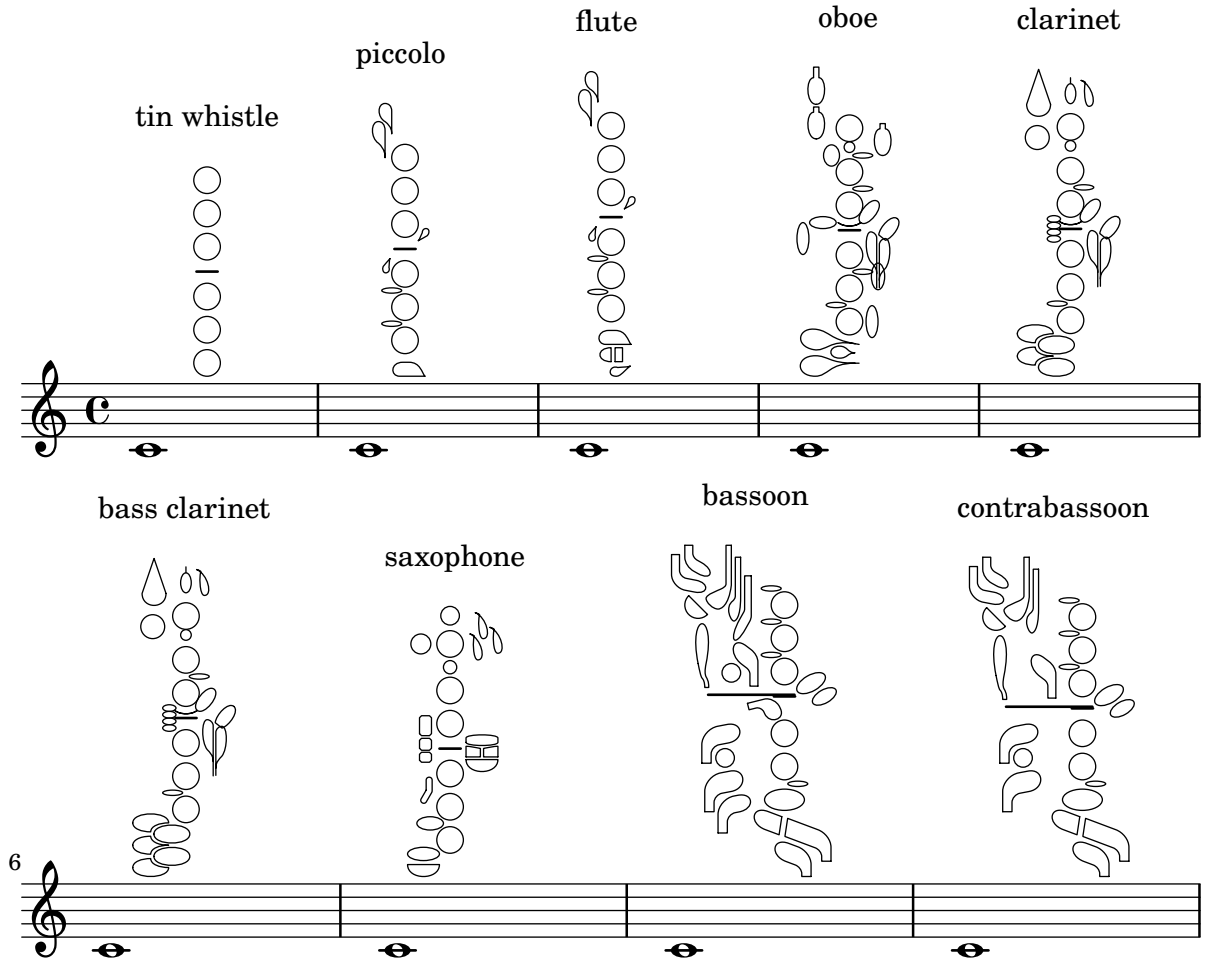
Listado de los diagramas para viento madera

El fragmento de música que aparece a continuación presenta todos los diagramas de viento madera que se encuentran definidos en LilyPond por el momento.

```
\relative c' {
  \textLengthOn
  c1^\markup \center-column { "tin whistle"
                              " "
                              \woodwind-diagram #'tin-whistle #'() }
  c1^\markup \center-column { "piccolo"
                              " "
                              \woodwind-diagram #'piccolo #'() }
  c1^\markup \center-column { "flute"
                              " "
                              \woodwind-diagram #'flute #'() }
  c1^\markup \center-column { "oboe"
                              " "
                              \woodwind-diagram #'oboe #'() }
  c1^\markup \center-column { "clarinet"
                              " "
                              \woodwind-diagram #'clarinet #'() }
  \break

  c1^\markup \center-column { "bass clarinet"
                              " "
                              \woodwind-diagram #'bass-clarinet #'() }
  c1^\markup \center-column { "saxophone"
                              " "
                              \woodwind-diagram #'saxophone #'() }
  c1^\markup \center-column { "bassoon"
                              " "
                              \woodwind-diagram #'bassoon #'() }
  c1^\markup \center-column { "contrabassoon"
                              " "
                              \woodwind-diagram #'contrabassoon #'() }
}
```

```
\paper {
  system-system-spacing.padding = 5
}
```



15 Chord notation

See also Sección “Chord notation” en *Notation Reference*.

Añadir un bajo cifrado encima o debajo de las notas
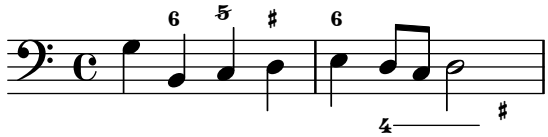
Al escribir un bajo cifrado, podemos situar las cifras encima o debajo de las notas del bajo mediante las instrucciones `\bassFigureStaffAlignmentDown` y `\bassFigureStaffAlignmentUp`. Anteponemos `\once` a la instrucción si queremos modificar solo la cifra siguiente.

La instrucción `\bassFigureStaffAlignmentNeutral` reinicia la dirección del bajo cifrado a su valor predeterminado.

```
bass = {
  \clef bass
  g4 b, c d |
  e d8 c d2
}

continuo = \figuremode {
  <_>4 <6>4 <5/>4
  \bassFigureStaffAlignmentUp
  <_+>4 <6> |
  \set Staff.useBassFigureExtenders = ##t
  \bassFigureStaffAlignmentDown
  <4>4. <4>8 <_+>4
}

\score {
  <<
    \new Staff = bassStaff \bass
    \context Staff = bassStaff \continuo
  >>
}
```

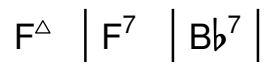


Añadir barras de compás al contexto de nombres de acorde (ChordNames)

Para añadir indicaciones de línea divisoria dentro del contexto de los nombres de acorde `ChordNames`, incluya el grabador `Bar_engraver`.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```



Ajustar los glifos de las alteraciones en el bajo cifrado

De forma predeterminada, en el bajo cifrado se usan glifos diseñados espcialmente para 6\\, 7\\ y 9\\. De forma similar, se usan de forma predeterminada glifos diseñados especialmente para los símbolos 2\+, 4\+ y 5\+, si aparece un signo más después del número.

Para cambiarlo, pase una lista-A a figuredBassPlusStrokedAlist y establezca el glifo en cuestión a #f o no lo especifique en absoluto.

```
#(set-global-staff-size 26)

\figures {
  \set figuredBassPlusDirection = #RIGHT
  <6\\> <7\\> <9\\> r
  <2\+> <4\+> <5\+> r

  \set figuredBassPlusStrokedAlist =
    #'((2 . "figbass.twoplus")
       ;; (4 . "figbass.fourplus")
       ;; (5 . "figbass.fiveplus")
       (6 . "figbass.sixstroked")
       ;; (7 . "figbass.sevenstroked")
       ;; (9 . "figbass.ninestroked")
      )
  <6\\> <7\\> <9\\> r
  <2\+> <4\+> <5\+> r
}
```

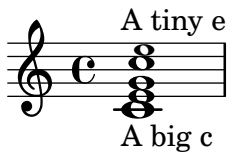


Changing a single note's size in a chord

Individual note heads in a chord can be modified with the \tweak command inside a chord, by altering the font-size property.

Inside the chord (within the brackets < >), before the note to be altered, place the \tweak command, followed by font-size and define the proper size like #-2 (a tiny note head).

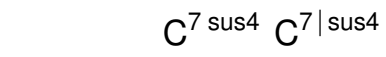
```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Modificación del separador de acordes

Se puede establecer el separador entre las distintas partes del nombre de un acorde para que sea cualquier elemento de marcado.

```
\chords {
  c:7sus4
  \set chordNameSeparator = \markup { \typewriter | }
  c:7sus4
}
```



Cambiar las posiciones de las alteraciones del bajo cifrado

Las alteraciones y los signos «más» pueden aparecer antes o después de los números, según el valor de las propiedades `figuredBassAlterationDirection` y `figuredBassPlusDirection`.

Si un signo más aparece después del número, hay glifos diseñados especialmente para algunas cifras.

```
#(set-global-staff-size 26)

\figures {
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassAlterationDirection = #RIGHT
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassPlusDirection = #RIGHT
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassAlterationDirection = #LEFT
  <5\+> <5+ 4\+> <6 4- 2\+> r
}
```



Excepciones para los nombres de acorde

La propiedad `chordNameExceptions` almacena una lista de excepciones para los nombres de acorde, para tratar casos que no están cubiertos, o bien que se tratan incorrectamente.

Los nombres de acorde predeterminados que se usan por LilyPond siguen las reglas dadas en el libro de Klaus Ignatzek's "Die Jazzmethode für Klavier 1"; el algoritmo para convertir acordes en nombres de acorde puede verse dentro del archivo `scm/chord-ignatzek-names.scm`. Las reglas adicionales se dan como excepciones de acordes y se almacenan dentro de la variable `ignatzekExceptions`, como está establecido en el archivo `ly/chord-modifiers-init.ly`.

Este fragmento de código modifica estas excepcioens en tres pasos.

1. Preparación de una música con acordes y elementos de marcado asociados. Por convención, la fundamental de cada acorde debe ser Do.
2. Llamar a la función de Scheme `sequential-music-to-chord-exceptions` para crear una lista de excepciones nueva, y después concatenarla con las existentes. Como `ignatzekExceptions` está preparado con el segundo parámetro de su función fijado al valor `#t` (para ignorar la fundamental de los acordes), tenemos que hacer lo mismo.
3. Registrar la nueva lista de excepciones.

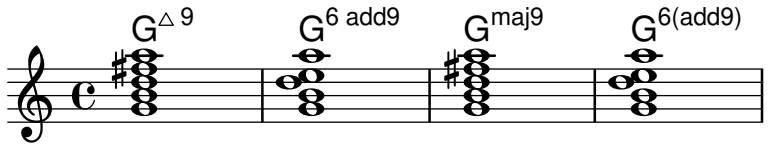
```
% Step 1: Define music with chords and markup for maj9 and 6(add9).
chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

% Step 2: Create extended exception list.
chExceptions =
#(append (sequential-music-to-chord-exceptions chExceptionMusic #t)
         ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  % Step 3: Register extended exception list.
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

<<
  \new ChordNames \theMusic
  \new Voice \theMusic
>>

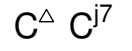
\layout {
  line-width = 10\cm
  ragged-right = ##f
}
```



Nombre de acorde maj7

La presentación del acorde de séptima mayor se puede ajustar mediante la propiedad de contexto `majorSevenSymbol`.

```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```



Nombres alternativos de acordes

Los nombres de acorde se generan a partir de una lista de alturas. Las funciones que construyen estos nombres se pueden personalizar.

Aquí se muestran acordes que siguen el libro de Ignatzek (pp. 17-18, 1995), empleado por omisión desde LilyPond 1.7.20, comparados con una notación alternativa de acordes de *Jazz* y la notación del libro de Harald Banter (1987). En este último se usa una fuente más pequeña porque las indicaciones tienden a ser bastante largas.

Esto refleja el mecanismo usado originalmente en antiguas versiones de LilyPond (anteriore a la 1.7); al no haberse mantenido adecuadamente, hay funcionalidades que se han perdido (principalmente listas de excepciones de acrodes) y se han introducido algunos bugs.

```
%%%% Legacy chord naming functions (formerly in scm/chord-generic-names.scm)
%%%% Copyright (C) 2003--2026 Jan Nieuwenhuizen <janneke@gnu.org>

#(set-global-staff-size 19.7)

#(define-public (banter-chordnames pitches bass inversion context)
  (old_chord->markup 'banter pitches bass inversion context))

#(define-public (jazz-chordnames pitches bass inversion context)
  (old_chord->markup 'jazz pitches bass inversion context))

#(define (define-translator-property symbol type? description)
  (if (not (and (symbol? symbol)
    (procedure? type?)
    (string? description)))
      (ly:error "error in call of define-translator-property"))
  (if (not (equal? (object-property symbol 'translation-doc) #f))
      (ly:error (G_ "symbol ~S redefined") symbol))

  (set-object-property! symbol 'translation-type? type?)
  (set-object-property! symbol 'translation-doc description)
  symbol)

#(for-each
  (lambda (x)
    (apply define-translator-property x))
  `((chordNameExceptionsFull ,list? "An alist of full chord
exceptions.  Contains @code{(@var{chord} . @var{markup})} entries.")
    (chordNameExceptionsPartial ,list? "An alist of partial chord
exceptions.  Contains @code{(@var{chord} . (@var{prefix-markup}
@var{suffix-markup}))} entries.")))

#(define-public (old_chord->markup
                 style pitches bass inversion context)
  "Entry point for @code{Chord_name_engraver}.
@var{pitches}, @var{bass}, and @var{inversion} are lily pitches."
  (define (default-note-namer pitch)
    ((chord-name:markup #f) pitch #f))

  (define (markup-or-empty-markup markup)
    "Return MARKUP if markup, else empty-markup"
    (if (markup? markup) markup empty-markup))

  (define (accidental->markup alteration)
    "Return accidental markup for ALTERATION."
    (if (= alteration 0)
        (make-line-markup (list empty-markup))
        (conditional-kern-before
```

```
       (accidental->text-markup alteration)
       (= alteration FLAT) 0.094725)))

(define (list-minus a b)
  "Return list of elements in A that are not in B."
  (lset-difference eq? a b))

(define (markup-join markups sep)
  "Return line-markup of MARKUPS, joining them with markup SEP"
  (if (pair? markups)
      (make-line-markup (list-insert-separator markups sep))
      empty-markup))

(define (conditional-kern-before markup bool amount)
  "Add AMOUNT of space before MARKUP if BOOL is true."
  (if bool
      (make-line-markup
       (list (make-hspace-markup amount)
         markup))
      markup))

(define (step-nr pitch)
  (let* ((pitch-nr (+ (* 7 (ly:pitch-octave pitch))
                      (ly:pitch-notename pitch)))
         (root-nr (+ (* 7 (ly:pitch-octave (car pitches)))
                     (ly:pitch-notename (car pitches)))))
    (+ 1 (- pitch-nr root-nr))))

(define (next-third pitch)
  (+ pitch
     (ly:make-pitch 0 2 (if (or (= (step-nr pitch) 3)
                                (= (step-nr pitch) 5))
                            FLAT 0))))

(define (step-alteration pitch)
  (let* ((normalized-pitch (- pitch (car pitches)))
         (alteration (ly:pitch-alteration normalized-pitch)))
    (if (= (step-nr pitch) 7) (+ alteration SEMI-TONE) alteration)))

(define (pitch-unalter pitch)
  (let ((alteration (step-alteration pitch)))
    (if (= alteration 0)
        pitch
        (ly:make-pitch (ly:pitch-octave pitch) (ly:pitch-notename pitch)
                       (- (ly:pitch-alteration pitch) alteration)))))

(define (step-even-or-altered? pitch)
  (let ((nr (step-nr pitch)))
    (if (!= (modulo nr 2) 0)
        (!= (step-alteration pitch) 0)
        #t)))
```

```
(define (step->markup-plusminus pitch)
  (let ((alt (step-alteration pitch)))
    (make-line-markup
     (list
      (number->string (step-nr pitch))
      (cond
       ((= alt DOUBLE-FLAT) "--")
       ((= alt FLAT) "-")
       ((= alt NATURAL) "")
       ((= alt SHARP) "+")
       ((= alt DOUBLE-SHARP) "++"))))))

(define (step->markup-accidental pitch)
  (make-line-markup
   (list (accidental->markup (step-alteration pitch))
         (make-simple-markup (number->string (step-nr pitch))))))

(define (step->markup-ignatzek pitch)
  (make-line-markup
   (if (and (= (step-nr pitch) 7)
            (= (step-alteration pitch) 1))
       (list (ly:context-property context 'majorSevenSymbol))
       (list (accidental->markup (step-alteration pitch))
             (make-simple-markup (number->string (step-nr pitch)))))))

;; tja, kennok
(define (make-sub->markup step->markup)
  (lambda (pitch)
    (make-line-markup (list (make-simple-markup "no")
                            (step->markup pitch)))))

(define (step-based-sub->markup step->markup pitch)
  (make-line-markup (list (make-simple-markup "no") (step->markup pitch))))

(define (get-full-list pitch)
  (if (<= (step-nr pitch) (step-nr (last pitches)))
      (cons pitch (get-full-list (next-third pitch)))
      '()))

(define (get-consecutive nr pitches)
  (if (pair? pitches)
      (let* ((pitch-nr (step-nr (car pitches)))
             (next-nr (if (!= (modulo pitch-nr 2) 0) (+ pitch-nr 2) nr)))
        (if (<= pitch-nr nr)
            (cons (car pitches) (get-consecutive next-nr (cdr pitches)))
            '()))
      '()))

;;; FIXME -- exceptions no longer work. -vv

(define (full-match exceptions)
  (if (pair? exceptions)
```

```
      (let* ((e (car exceptions))
             (e-pitches (car e)))
        (if (equal? e-pitches pitches)
            e
            (full-match (cdr exceptions))))
      #f))

(define (partial-match exceptions)
  (if (pair? exceptions)
      (let* ((e (car exceptions))
             (e-pitches (car e)))
        (if (equal? e-pitches (take pitches (length e-pitches)))
            e
            (partial-match (cdr exceptions))))
      #f))

;; FIXME: exceptions don't work anyway.
(if #f (begin
         (write-me "pitches: " pitches)))
(let* ((full-exceptions
        (ly:context-property context 'chordNameExceptionsFull))
       (full-exception (full-match full-exceptions))
       (full-markup (if full-exception (cadr full-exception) '()))
       (partial-exceptions
        (ly:context-property context 'chordNameExceptionsPartial))
       (partial-exception (partial-match partial-exceptions))
       (partial-pitches (if partial-exception (car partial-exception) '()))
       (partial-markup-prefix
        (if partial-exception (markup-or-empty-markup
                               (cadr partial-exception)) empty-markup))
       (partial-markup-suffix
        (if (and partial-exception (pair? (cddr partial-exception)))
            (markup-or-empty-markup (caddr partial-exception)) empty-markup))
       (root (car pitches))
       (full (get-full-list root))
       ;; kludge alert: replace partial matched lower part of all with
       ;; 'normal' pitches from full
       ;; (all pitches)
       (all (append (take full (length partial-pitches))
                    (drop pitches (length partial-pitches))))

       (highest (last all))
       (missing (list-minus full (map pitch-unalter all)))
       (consecutive (get-consecutive 1 all))
       (rest (list-minus all consecutive))
       (altered (filter step-even-or-altered? all))
       (cons-alt (filter step-even-or-altered? consecutive))
       (base (list-minus consecutive altered)))


  (if #f (begin
           (write-me "full:" full)
```

```
          ;; (write-me "partial-pitches:" partial-pitches)
          (write-me "full-markup:" full-markup)
          (write-me "partial-markup-perfix:" partial-markup-prefix)
          (write-me "partial-markup-suffix:" partial-markup-suffix)
          (write-me "all:" all)
          (write-me "altered:" altered)
          (write-me "missing:" missing)
          (write-me "consecutive:" consecutive)
          (write-me "rest:" rest)
          (write-me "base:" base)))

    (case style
      ((banter)
       ;;    root
       ;;    + steps:altered + (highest all -- if not altered)
       ;;    + subs:missing

       (let* ((root->markup default-note-namer)
              (step->markup step->markup-plusminus)
              (sub->markup (lambda (x)
                             (step-based-sub->markup step->markup x)))
              (sep (make-simple-markup "/")))

         (if
          (pair? full-markup)
          (make-line-markup (list (root->markup root) full-markup))

          (make-line-markup
           (list
            (root->markup root)
            partial-markup-prefix
            (make-super-markup
             (markup-join
              (append
               (map step->markup
                    (append altered
                            (if (and (> (step-nr highest) 5)
                                     (not
                                      (step-even-or-altered? highest)))
                                (list highest) '())))
               (list partial-markup-suffix)
               (map sub->markup missing))
              sep)))))))


      ((jazz)
       ;;    root
       ;;    + steps:(highest base) + cons-alt
       ;;    + 'add'
       ;;    + steps:rest
       (let* ((root->markup default-note-namer)
              (step->markup step->markup-ignatzek)
```

```
            (sep (make-simple-markup " "))
            (add-prefix (make-simple-markup " add")))

          (if
           (pair? full-markup)
           (make-line-markup (list (root->markup root) full-markup))

           (make-line-markup
            (list
             (root->markup root)
             partial-markup-prefix
             (make-super-markup
              (make-line-markup
               (list

                ;; kludge alert: omit <= 5
                ;;(markup-join (map step->markup
                ;;                  (cons (last base) cons-alt)) sep)

                ;; This fixes:
                ;;  c     C5       -> C
                ;;  c:2   C5 2     -> C2
                ;;  c:3-  Cm5      -> Cm
                ;;  c:6.9 C5 6add9 -> C6 add 9 (add?)
                ;;  ch = \chords { c c:2 c:3- c:6.9^7 }
                (markup-join (map step->markup
                                  (let ((tb (last base)))
                                    (if (> (step-nr tb) 5)
                                        (cons tb cons-alt)
                                        cons-alt))) sep)

                (if (pair? rest)
                    add-prefix
                    empty-markup)
                (markup-join (map step->markup rest) sep)
                partial-markup-suffix))))))))

        (else empty-markup))))


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%%%% Here begins the actual snippet:

chs = \transpose c' c' {
  <c e g>1
  <c es g>  % m = minor triad
  <c e gis>
  <c es ges> \break
  <c e g bes>
  <c es g bes>
  <c e g b>  % triangle = maj
```

```
  <c es ges beses>
  <c es ges b> \break
  <c e gis bes>
  <c es g b>
  <c e gis b>
  <c es ges bes> \break
  <c e g a>  % 6 = major triad with added sixth
  <c es g a>  % m6 = minor triad with added sixth
  <c e g bes d'>
  <c es g bes d'> \break
  <c es g bes d' f' a' >
  <c es g bes d' f' >
  <c es ges bes d' >
  <c e g bes des' > \break
  <c e g bes dis'>
  <c e g bes d' f'>
  <c e g bes d' fis'>
  <c e g bes d' f' a'> \break
  <c e g bes d' fis' as'>
  <c e gis bes dis'>
  <c e g bes dis' fis'>
  <c e g bes d' f' as'> \break
  <c e g bes des' f' as'>
  <c e g bes d' fis'>
  <c e g b d'>
  <c e g bes d' f' as'> \break
  <c e g bes des' f' as'>
  <c e g bes des' f' a'>
  <c e g b d'>
  <c e g b d' f' a'> \break
  <c e g b d' fis'>
  <c e g bes des' f ' a'>
  <c f g>
  <c f g bes> \break
  <c f g bes d'>
  <c e g d'>  % add9
  <c es g f'>
  <c e g b fis'>  % Lydian
  <c e g bes des' ees' fis' aes'>  % altered chord
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% alternate Jazz notation

efullmusicJazzAlt = {
  <c e gis>1-\markup { "+" }
  <c e g b>-\markup {
    \normal-size-super
    % \override #'(font-family . math) "N"
    \override #'(font-family . math) "M"
  }
```

```
  %%c:3.5.7 = \markup { \override #'(font-family . math) "M" }
  %%c:3.5.7 = \markup { \normal-size-super "maj7" }

  <c es ges>-\markup { \super "o" }  % should be $\circ$ ?
  <c es ges bes>-\markup { \super \combine "o" "/" }
  <c es ges beses>-\markup { \super  "o7" }
}

efullJazzAlt = #(sequential-music-to-chord-exceptions efullmusicJazzAlt #f)

epartialmusicJazzAlt = {
  <c d>1-\markup { \normal-size-super "2" }
  <c es>-\markup { "m" }
  <c f>-\markup { \normal-size-super "sus4" }
  <c g>-\markup { \normal-size-super "5" }
  %% TODO, partial exceptions
  <c es f>-\markup { "m" }-\markup { \normal-size-super "sus4" }
  <c d es>-\markup { "m" }-\markup { \normal-size-super "sus2" }
}

epartialJazzAlt = #(sequential-music-to-chord-exceptions epartialmusicJazzAlt #f)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\score {
  <<
    \new ChordNames {
      %% Already set by default:
      %\set chordNameFunction = #ignatzek-chord-names
      \set instrumentName = "Ignatzek"
      \set shortInstrumentName = "Def"
      \chs
    }
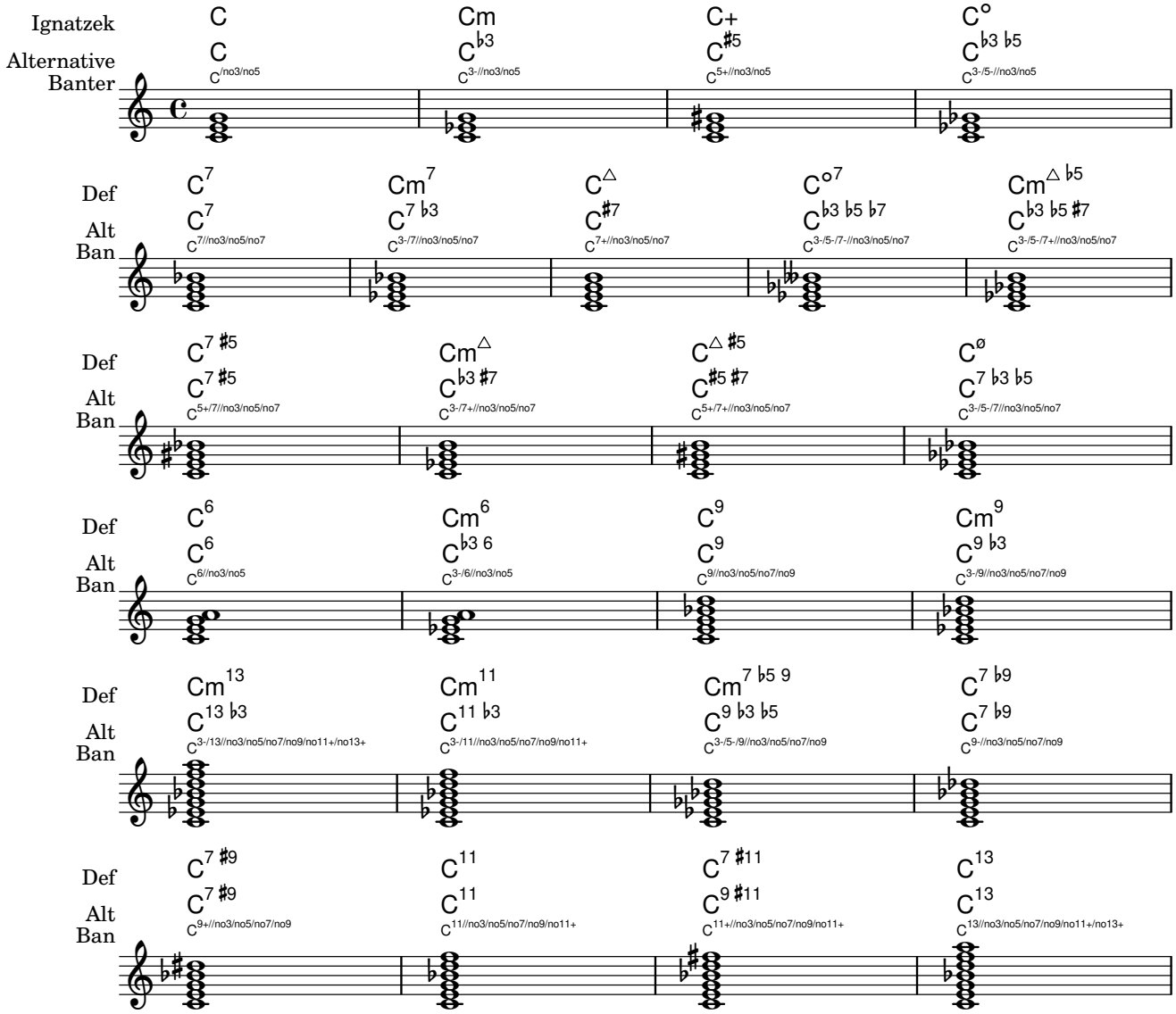
    \new ChordNames {
      \set chordNameFunction = #jazz-chordnames
      \set majorSevenSymbol = \whiteTriangleMarkup
      \set chordNameSeparator = "/"
      \set chordNameExceptionsFull = \efullJazzAlt
      \set chordNameExceptionsPartial = \epartialJazzAlt
      \set instrumentName = "Alternative"
      \set shortInstrumentName = "Alt"
      \chs
    }
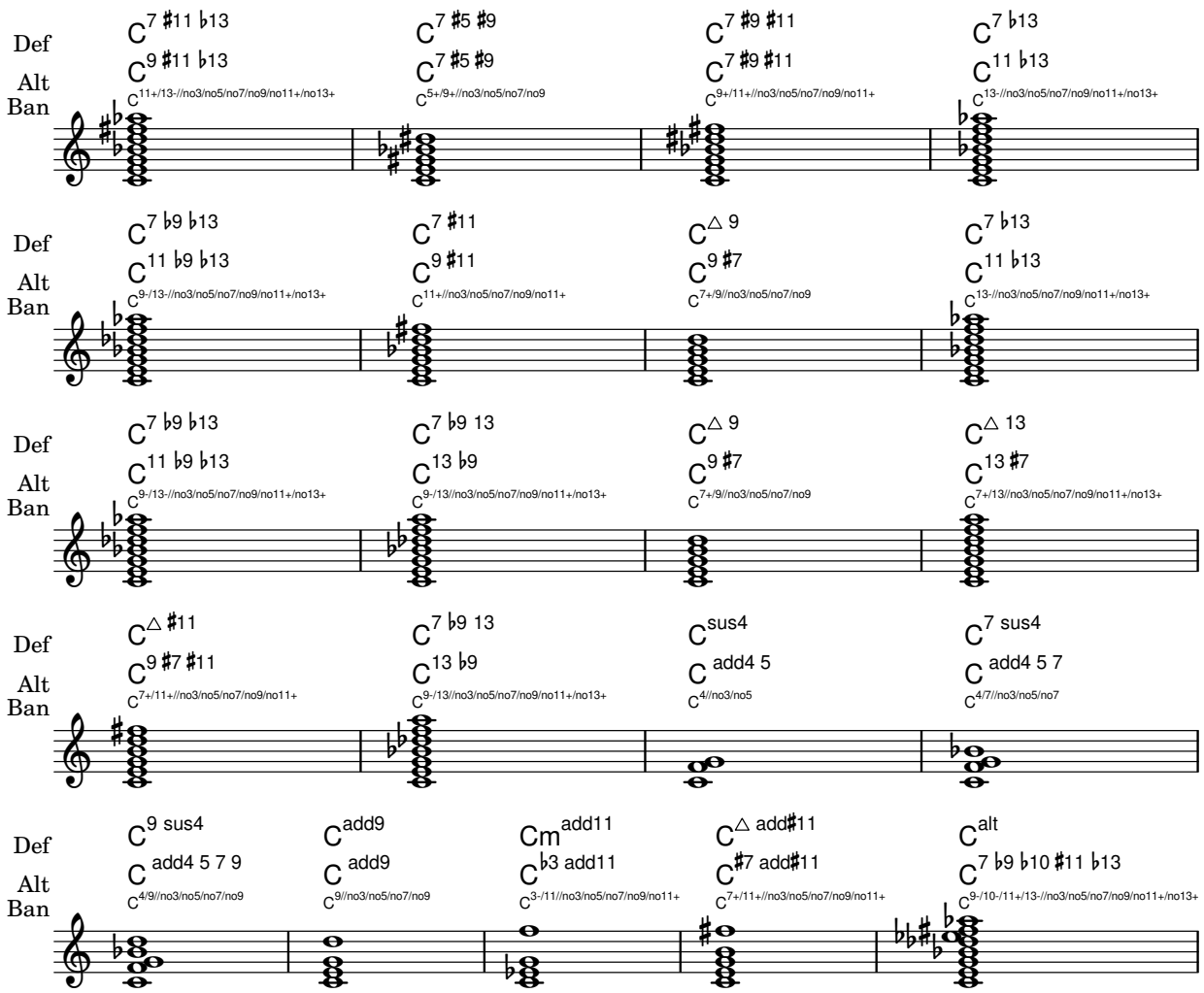
    %% This is the Banter (1987) style.  It gives exceedingly
    %% verbose (wide) names, making the output file take up to 4 pages.

    \new ChordNames {
      \set chordNameFunction = #banter-chordnames
      \override ChordName.font-size = -3
      \set instrumentName = "Banter"
```

```
      \set shortInstrumentName = "Ban"
      \chs
    }

  \new Staff \transpose c c' { \chs }
  >>
  \layout {
    #(layout-set-staff-size 16)
    system-system-spacing.basic-distance = 0
    \context {
      \ChordNames
      \consists "Instrument_name_engraver"
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}
```





Acordes con digitaciones ampliadas para diagramas de trastes y tablaturas

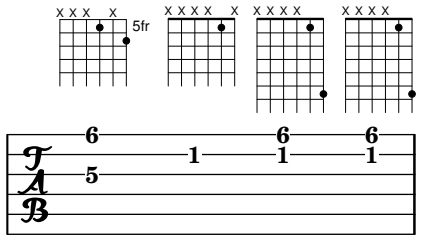
A veces se requieren acordes con digitación ampliada o estirada. Sin embargo, si no se especifica lo contrario, la propiedad de contexto `maximumFretStretch` viene establecida al valor 4, lo que puede dar lugar a una advertencia "No hay ninguna cuerda para la nota ..." y la nota se omite. Podemos fijar `maximumFretStretch` a un valor adecuado o asignar explícitamente números de cuerda para todas las notas del acorde, para corregir esto.

```
% The code below prints two warnings for the second chord,
% which may be omitted by uncommenting the following line.
%
% #(for-each (lambda (x) (ly:expect-warning "No string for pitch")) (iota 2))

mus = {
  <c' bes'>
  <c'\2 bes'>
  \set maximumFretStretch = 5
  <c' bes'>
  <c'\2 bes'\1>
}

<<
```

```
  \new FretBoards \mus
  \new TabVoice \mus
>>
```

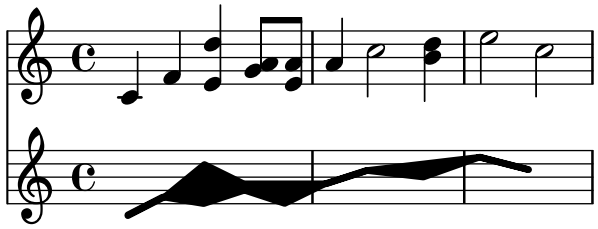


Clusters («racimos»)

Los «clusters» o racimos son un mecanismo para indicar la interpretación de un ámbito de notas al mismo tiempo.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```

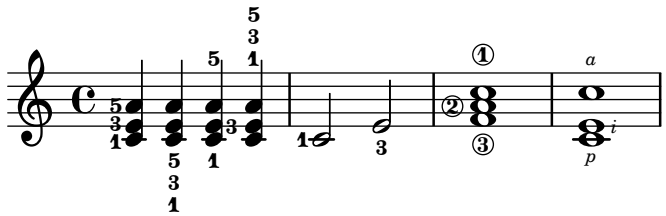


Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación usando la propiedad `fingeringOrientation`. Para que se tenga en cuenta la orientación de las digitaciones, la instrucción de digitación se debe utilizar dentro de una construcción de acorde (`<...>`) aunque sea una sola nota. Se puede controlar la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar usando las propiedades `stringNumberOrientation` y `strokeFingerOrientation`, respectivamente.

Estas propiedades se pueden establecer a una lista de entre uno y tres valores. Controlan si las digitaciones se pueden situar por encima (si aparece `up` en la lista), por debajo (si aparece `down`), a la izquierda (si aparece `left`), o a la derecha (si aparece `right`). Por contra, si una localización no aparece en la lista, no se coloca ninguna digitadión en ese lugar. LilyPond admite esas restricciones y trata de encontrar la mejor ubicación para las digitaciones de las notas de los acordes siguientes. Observe que `left` y `right` se excluyen mutuamente (las digitaciones se pueden colocar en un lado o en el otro, no en los dos).

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Solución de problemas de acordes y barras que cruzan el pentagrama

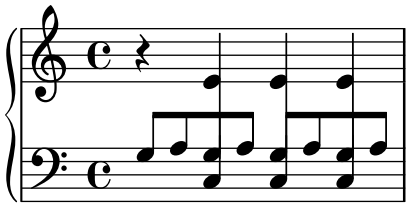
A veces es mejor usar plicas que parten del 'otro' pentagrama para crear acordes de pentagrama cruzado y así engañar al detector de colisiones de plicas de LilyPond. En el siguiente fragmento de código, si se hubieran usado las plicas que provienen del pentagrama inferior, habría sido necesario usar explícitamente

```
\override Staff.Beam.collision-voice-only = ##t
```

de forma que LilyPond no mueva las plicas.

```
\new PianoStaff <<
  \new Staff = up \relative c' <<
    { r4
      \override Stem.cross-staff = ##t
      \override Stem.length = #19 % this is in half-spaces,
          % so it makes stems 9.5 staffspaces long
      \override Stem.Y-offset = #-6 % stems are normally lengthened
          % upwards, so here we must lower the stem by the amount
          % equal to the lengthening - in this case (19 - 7) / 2
          % (7 is default stem length)
      e e e }
    { s4
      \change Staff = "bottom"
      \override NoteColumn.ignore-collision = ##t
      c, c c
    }
  >>
```

```
  \new Staff = bottom \relative c' {
    \clef bass
    \voiceOne
    g8 a g a g a g a
  }
>>
```



Personalización del estilo de las rejillas de acordes

Se pueden definir divisiones personalizadas de los cuadrados de acordes a través de las propiedades `measure-division-lines-alist` y `measure-division-chord-placement-alist` de `ChordSquare`. Ambas son listas de asociación. Sus claves de asociación son divisiones de compás, concretamente listas que dan la fracción del compás que representa cada acorde (o silencio o separación). De forma más precisa, una lista-A de divisiones de compases consta de números positivos exactos que suman la unidad, por ejemplo `'(1/2 1/4 1/4)`. El requisito de la exactitud implica que p.ej. `1/2` es válido pero no `0.5`.

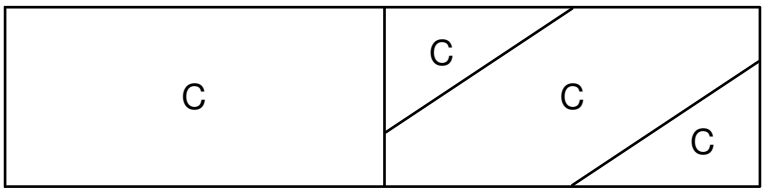
Los valores de `measure-division-lines-alist` son listas de líneas, que están representadas como `(x1 y1 x2 y2)`. La línea comienza en el punto `(x1 . y1)` y finaliza en `(x2 . y2)`. Las coordenadas se expresan en la escala [-1, 1] relativa a las dimensiones del cuadrado.

Los valores en `measure-division-chord-placement-alist` son listas de pares `(x . y)` que dan la ubicación de los compases respectivos.

Este ejemplo define un estilo de rejilla de acordes peculiar que tiene una regla para los compases divididos en tres partes iguales.

```
\paper {
  line-width = 10\cm
  ragged-right = ##f
}

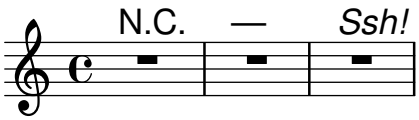
\new ChordGrid \with {
  \override ChordSquare.measure-division-lines-alist =
    #'(((1) . ())
       ((1/3 1/3 1/3) . ((-1 -0.4 0 1) (0 -1 1 0.4))))
  \override ChordSquare.measure-division-chord-placement-alist =
    #'(((1) . ((0 . 0)))
       ((1/3 1/3 1/3) . ((-0.7 . 0.5) (0 . 0) (0.7 . -0.5))))
}
\chordmode {
  \time 3/4
  c2.
  c4 c4 c4
}
```



Personalización del símbolo de Sin Acorde

Por omisión, los silencios en un contexto `ChordNames` producen la impresión del texto “N.C.”. Este elemento de marcado se puede personalizar mediante el establecimiento de la propiedad de contexto `noChordSymbol`.

```
<<
  \chords {
    R1
    \set noChordSymbol = "---"
    R1
    \set noChordSymbol = \markup \italic "Ssh!"
    R1
  }
  {
    R1*3
  }
>>
```
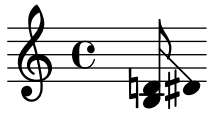


Impresión de acordes complejos

He aquí una forma de imprimir un acorde en el que suena la misma nota dos veces con distintas alteraciones.

```
fixA = {
  \once \override Stem.length = #12
}

fixB = {
  \once \override NoteHead.X-offset = #1.7
  \once \override Stem.length = #7
  \once \override Stem.rotation = #'(45 0 0)
  \once \override Stem.extra-offset = #'(-0.1 . -0.2)
  \once \override Flag.style = #'no-flag
  \once \override Accidental.extra-offset = #'(4 . -.1)
}


\relative c' {
  << { \fixA <b d!>8 } \\ { \voiceThree \fixB dis } >> s
}
```

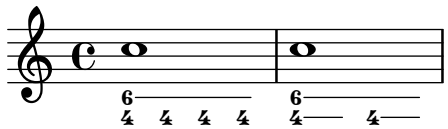


Dividir manualmente las líneas de extensión del bajo cifrado para ciertos números solamente

A menudo, el bajo cifrado usa líneas de extensión para indicar la continuación del elemento correspondiente dentro del acorde. Sin embargo, en este caso LilyPond intenta usar los extensores siempre que es posible, lo que no siempre es deseado. Para dividir los extensores individuales, añada el modificador \! después de un número.

```
bassfigures = \figuremode {
  \set useBassFigureExtenders = ##t
  <6 4>4 <6 4\!> <6 4\!> <6 4\!> |
  <6\! 4\!>  <6 4> <6 4\!> <6 4>
}

<<
  \new Staff \relative c'' { c1 c1 }
  \new FiguredBass \bassfigures
>>
```



Print chord names with same root and different bass as slash and bass note

To print subsequent chord names only differing in their bass note as slash and bass note without a root symbol, use the Scheme engraver defined in this snippet. The behaviour is controlled by the chordChanges context property.

```
#(define Bass_changes_equal_root_engraver
   (lambda (ctx)
     "Drop root for follow-up chord names that differ only in bass.

In other words, the chord name sequence 'D D/C D/B' gets actually
printed as 'D /C /B'.

Set the `chordChanges` context property to `#t` to activate this
feature."
     (let ((chord-pitches '())
           (last-chord-pitches '())
           (bass-pitch #f))
       (make-engraver
        ((initialize this-engraver)
         (let ((chord-note-namer (ly:context-property ctx
                                                      'chordNoteNamer)))
           ;; Set `chordNoteNamer`, respecting user setting if already
           ;; done.
           (ly:context-set-property! ctx 'chordNoteNamer
```

```
                                (if (procedure? chord-note-namer)
                                    chord-note-namer
                                    (chord-name:markup #f)))))

        (listeners
         ((note-event this-engraver event)
          (let* ((pitch (ly:event-property event 'pitch))
                 (pitch-name (ly:pitch-notename pitch))
                 (pitch-alt (ly:pitch-alteration pitch))
                 (bass (ly:event-property event 'bass #f))
                 (inversion (ly:event-property event 'inversion #f)))
            ;; We look at the `bass` and `inversion` event properties
            ;; to decide how to handle the current note event.  If
            ;; `inversion` is set we add the bass note to the chord as
            ;; an ordinary member so that we can compare inversed
            ;; chords; if `bass` is set the bass note is not added to
            ;; the chord.
            ;;
            ;; In the `chord-pitches` list we actually collect only
            ;; the notes' pitch names (which are integers) and pitch
            ;; alterations as pairs, ignoring the octave.
            (cond (bass (set! bass-pitch pitch))
                  (inversion
                   (set! bass-pitch pitch)
                   (set! chord-pitches
                         (cons (cons pitch-name pitch-alt)
                               chord-pitches)))
                  (else
                   (set! chord-pitches
                         (cons (cons pitch-name pitch-alt)
                               chord-pitches)))))))

        (acknowledgers
         ((chord-name-interface this-engraver grob source-engraver)
          (let ((chord-changes (ly:context-property ctx
                                                    'chordChanges #f)))
            ;; If subsequent chords are equal apart from their bass,
            ;; we change the `text` property to print only the slash
            ;; and the bass note (via the formatter stored in the
            ;; `chordNoteNamer` context property).
            ;;
            ;; Equality is tested by comparing the sorted lists of
            ;; this chord's elements and the previous chord.  Sorting
            ;; is needed because inverted chords may have a different
            ;; order of pitches.  Note that we only do a simplified
            ;; sorting using the pitch name, ignoring the alteration.
            (if (and bass-pitch
                     chord-changes
                     (equal? (sort chord-pitches car<)
                             (sort last-chord-pitches car<)))
                (ly:grob-set-property!
                 grob 'text
```

```
                (make-line-markup
                 (list
                  (ly:context-property ctx 'slashChordSeparator)
                  ((ly:context-property ctx 'chordNoteNamer)
                   bass-pitch
                   (ly:context-property ctx
                                        'chordNameLowercaseMinor))))))
            (set! last-chord-pitches chord-pitches)
            (set! chord-pitches '())
            (set! bass-pitch #f))))

        ((finalize this-engraver)
         (set! last-chord-pitches '()))))


myChords = \chordmode {
  % \germanChords

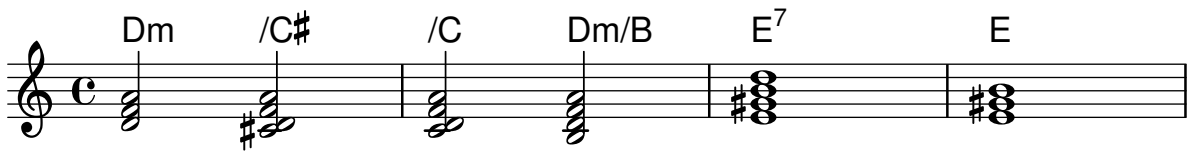
  \set chordChanges = ##t
  d2:m d:m/cis

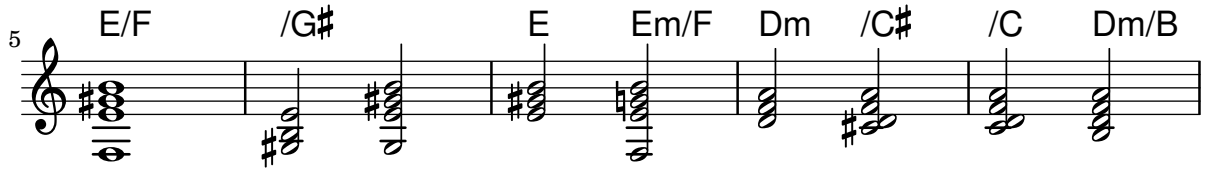
  d:m/c
  \set chordChanges = ##f
  d:m/b

  e1:7
  \set chordChanges = ##t
  e
  \break

  \once \set chordChanges = ##f
  e1/f
  e2/gis e/+gis e e:m/f d:m d:m/cis d:m/c
  \set chordChanges = ##f
  d:m/b
}

<<
  \new ChordNames
    \with { \consists #Bass_changes_equal_root_engraver }
    \myChords
  \new Staff \myChords
>>
```



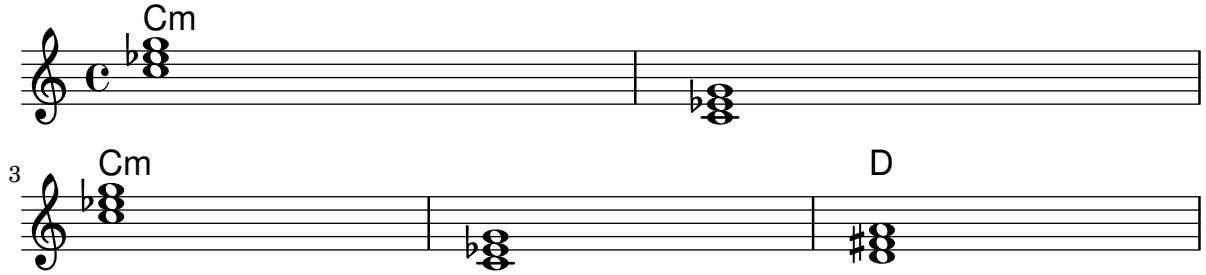


Imprimir los acordes cuando se produce un cambio

De forma predeterminada se imprimen todos los acordes que aparecen en el código de entrada. Se puede modificar este comportamiento de tal manera que los acordes se impriman exclusivamente al comienzo de las líneas o cuando cambia el acorde.

```
harmonies = \chordmode {
  c'1:m c:m \break
  c'1:m c:m d
}

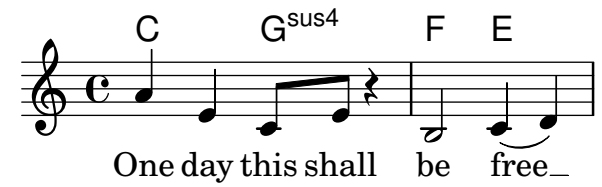
<<
  \new ChordNames {
    \set chordChanges = ##t
    \harmonies
  }
  \new Staff {
    \harmonies
  }
>>
```



Hoja guía de acordes o *lead sheet* sencilla

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet».

```
<<
  \chords { c2 g:sus4 f e }
  \new Staff \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Plantilla de pentagrama único con música, letra y acordes

Esta plantilla facilita la preparación de una canción con melodía, letra y acordes.

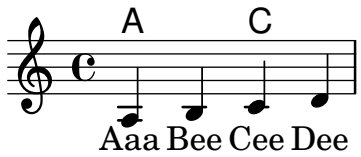
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Single-staff template with notes, lyrics, chords, and frets

Here is a simple lead sheet template with melody, lyrics, chords, and fret diagrams.

```
verseI = \lyricmode {
  \stanza "1."
  This is the first verse
}

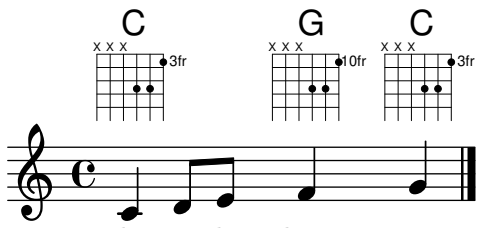
verseII = \lyricmode {
  \stanza "2."
  This is the second verse.
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
```

```
  c2 g4 c
}

staffMelody = \relative c' {
   \key c \major
   \clef treble
   % Type notes for melody here
   c4 d8 e f4 g
   \bar "|."
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}
```



1. This is the first verse
2. This is the second verse.

Plantilla de pentagrama único con música y acordes

¿Quiere preparar una hoja guía de acordes (o «lead sheet») con melodía y acordes? ¡No busque más!
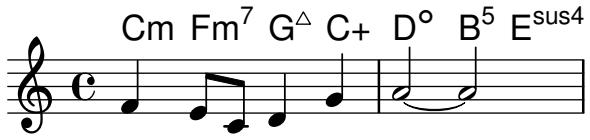
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  f4 e8[ c] d4 g |
  a2 ~ a
}

harmonies = \chordmode {
```

```
  c4:m f:min7 g:maj c:aug |
  d2:dim b4:5 e:sus
}

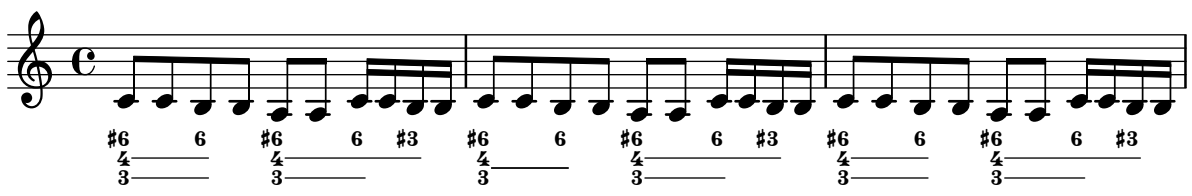
\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Staff \melody
  >>
  \layout{ }
  \midi { }
}
```



Centrado vertical de las líneas de bajo cifrado emparejadas

Allí donde se utilicen líneas extensoras para el bajo cifrado mediante el establecimiento de useBassFigureExtenders a #t, las parejas de líneas extensoras congruentes se centran verticalmente si el valor de figuredBassCenterContinuations tiene el valor #t.

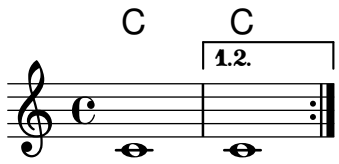
```
<<
  \relative c' {
    \repeat unfold 3 {
      c8 c b b a a c16 c b b
    }
  }
  \figures {
    \set useBassFigureExtenders = ##t
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
    \set figuredBassCenterContinuations = ##t
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
    \set figuredBassCenterContinuations = ##f
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
  }
>>
```



Corchetes de primera y segunda vez debajo de los acordes

Mediante la adición del grabador `Volta_engraver` al pentagrama pertinente, se pueden poner los corchetes de primera y segunda vez debajo de los acordes.

```
\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



16 Contemporary music

See also Sección “Contemporary music” en *Notation Reference*.

Barras recortadas

Las barras recortadas al principio y al final de las barras, junto con las barras unidas a notas sueltas que parecen corchetes planos, son posibles con una combinación de `stemLeftBeamCount`, `stemRightBeamCount` e indicadores de barra `[]` pareados.

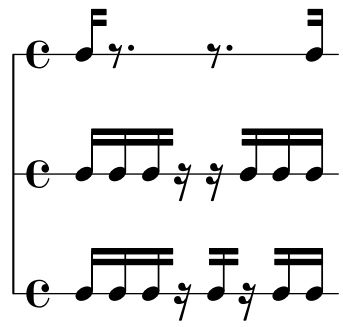
Para imitar los corchetes planos hacia la derecha sobre notas sueltas, indicadores de barra `[]` pareados y fije el valor de `stemLeftBeamCount` a cero. En cambio, para imitar corchetes planos hacia la izquierda sobre notas sueltas, fije el valor de `stemRightBeamCount` a cero (línea uno).

Para las barras recortadas hacia la derecha al final de una serie de noras barradas, fije el valor de `stemRightBeamCount` a un valor positivo. Por su parte, para barras recortadas hacia la derecha al comienzo de una serie de notas barradas, fije `stemLeftBeamCount` (línea dos).

A veces tiene sentido que una nota suelta entre dos silencios lleve dos barras recortadas, hacia la derecha y hacia la izquierda. Esto se hace escribiendo solamente indicadores de barrado `[]` pareados (línea tres).

Observe que `\set stemLeftBeamCount` siempre equivale a `\once \set`. En otras palabras, los ajustes para el número de barras no son “persistentes”, o sea que la pareja de barras recortadas unidas a la semicorchea aislada del último ejemplo no tiene nada que ver con la instrucción `\set` de la barra anterior.

```
\score {
  <<
    \new RhythmicStaff {
      \set stemLeftBeamCount = 0
      c16[] r8.
      r8.
      \set stemRightBeamCount = 0
      16[]
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r r
      \set stemLeftBeamCount = 2
      16 16 16
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r16
      16[] r16
      \set stemLeftBeamCount = 2
      16 16
    }
  >>
}
```



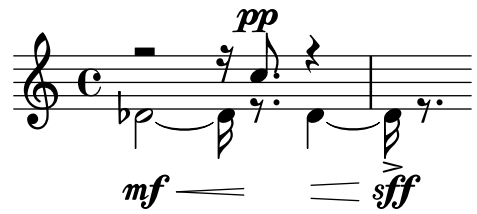
Regulador interrumpido

Para hacer invisibles partes de un regulador de crescendo, se usa el método de dibujar un rectángulo blanco encima de la parte respectiva del regulador, tapándola. El rectángulo se define como un elemento de marcado de texto.

La instrucción de marcado `with-dimensions` indica a LilyPond que considere solamente el extremo inferior del rectángulo cuando realice el espaciado de éste frente al regulador. La propiedad `staff-padding` evita que el rectángulo quepa entre el regulador y el pentagrama.

Asegúrse de que el regulador está en una capa más baja que el elemento de marcado de texto para trazar el rectángulo encima del regulador.

```
\relative c' {
  <<
    {
      \dynamicUp
      r2 r16 c'8.\pp r4
    }
    \\
    {
      \override DynamicLineSpanner.layer = #0
      des,2\mf\< ~
      \override TextScript.layer = #2
      \once\override TextScript.staff-padding = #6
      \once\override TextScript.vertical-skylines = #'()
      des16_\markup \with-dimensions #'(2 . 7) #'(0 . 0)
                    \with-color #white
                    \filled-box #'(2 . 7) #'(0 . 2) #0
      r8. des4 ~ des16->\sff r8.
    }
  >>
}
```



Modificar el compás dentro de una sección polimétrica utilizando `\scaleDurations`

Polirritmia flexible con compases no alineados

Para poder crear explícitamente contextos con compases independientes, suprima el traductor `Timing_translator` del contexto `Score` y defina un contexto `TimingStaffGroup` que contenga el traductor `Timing_translator`. Esto hace que `Timing` sea un alias de `TimingStaffGroup`, con la mira puesta en las instrucciones `\time` hacia el `TimingStaffGroup` contenedor.

A diferencia de la instrucción incorporada `\enablePerStaffTiming` de LilyPond, este enfoque requiere la creación explícita de contextos `TimingStaffGroup`; a cambio, permite crear varios contextos `Staff` que de forma conjunta siguen al compás definido en su `TimingStaffGroup` contenedor.

Indicaciones de compás escalados localmente

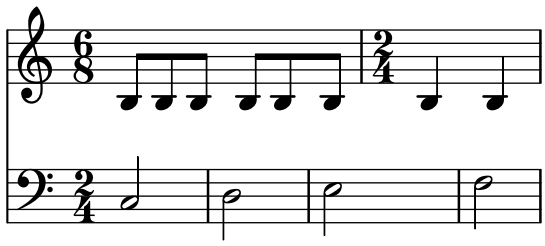
Use la instrucción no escalable `\time` para establecer un compás de la longitud deseada en `Timing`, más conocido como `TimingStaffGroup`. En este fragmento de código, todos los pentagramas que están por debajo de `TimingStaffGroup` usan una indicación de compás escalada, de forma que cualquier compás que establezcamos con la longitud deseada es tan bueno como cualquier otro. Si hubiese un contexto contenedor que no usara una indicación de compás escalada, la elección del compás que se establezca en `Timing` tendría importancia en dicho contexto.

Use la instrucción `\polymetric \time` para establecer propiedades métricas escalablesen los contextos que están por debajo de `Timing`, y use la instrucción `\scaleDurations` para escalar tanto el compás local como las notas que han de llenar el compás.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \accepts TimingStaffGroup
  }
  \context {
    \StaffGroup
    \name TimingStaffGroup
    \alias StaffGroup
    \consists "Timing_translator"
  }
}

<<
  \new TimingStaffGroup <<
    \new Staff {
      \scaleDurations 8/5 {
        \time 6/5 % to set measure length in Timing
        \context Staff \polymetric \time 6/8
        b8 b b b b b
        \time 4/5 % to set measure length in Timing
        \context Staff \polymetric \time 2/4
        b4 b
      }
    }
  >>
  \new TimingStaffGroup <<
    \new Staff {
      \clef bass
      \time 2/4
```

```
      c2 d e f
    }
  >>
>>
```

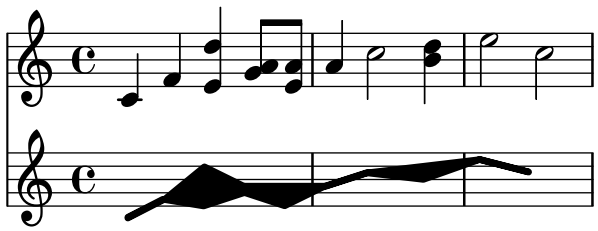


Clusters («racimos»)

Los «clusters» o racimos son un mecanismo para indicar la interpretación de un ámbito de notas al mismo tiempo.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}
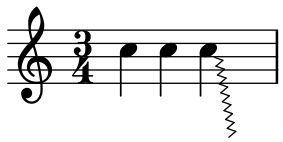
<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Glissando contemporáneo

Se puede tipografiar un gissando contemporáneo sin nota final utilizando una nota oculta y temporalización de cadenza.

```
\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}
```



Ligaduras planas

Este fragmento de código aporta una función `flared-tie` para trazar una ligadura formada por líneas rectas. Su objetivo es sustituir a la función predeterminada de trazado de la ligadura (esto es, servir como un argumento sustitutorio para la propiedad `stencil` del objeto gráfico `Tie`).

El argumento de `flared-tie` es una lista de pare de coordenadas que especifican puntos adicionales entre el primero y el último para abarcar todas las líneas de la ligadura. Los puntos primero y último son idénticos a los puntos de inicio y final de la ligadura, respectivamente. Los valores de las coordenadas X e Y son múltiplos de la longitud y la altura del rectángulo circundante (teniendo también en cuenta la dirección de la ligadura); por tanto, el primer punto tiene las coordenadas (0,0), y el último (1,0).

La función `flare-tie` define una forma abreviada de especificar una ligadura plana. Es posible efectuar un trucaje adicional de la forma mediante la sobreescritura de `Tie.details.height-limit` o con la función `\shape`.

También es posible cambiar la definición personalizada sobre la marcha.

```
#(define ((flared-tie coords) grob)
   (define (pair-to-list pair)
     (list (car pair) (cdr pair)))

   (define (normalize-coords goods x y dir)
     (map
      (lambda (coord)
        (cons (* x (car coord)) (* y dir (cdr coord))))
      goods))

   (define (my-c-p-s points thick)
     (make-connected-path-stencil points thick 1.0 1.0 #f #f))

   ;; Calling `ly:tie::print` and assigning its return value to a
   ;; variable in this outer `let` triggers LilyPond to position the
   ;; tie, allowing us to extract its extents.  We only proceed,
   ;; however, if the tie doesn't get discarded (for whatever reason).
   (let ((sten (ly:tie::print grob)))
     (if (grob::is-live? grob)
         (let* ((layout (ly:grob-layout grob))
                (line-thickness (ly:output-def-lookup layout
                                                      'line-thickness))
                (thickness (ly:grob-property grob 'thickness 0.1))
                (used-thick (* line-thickness thickness))
                (dir (ly:grob-property grob 'direction))
                (xex (ly:stencil-extent sten X))
                (yex (ly:stencil-extent sten Y))
                (lenx (interval-length xex))
                (leny (interval-length yex))
                (xtrans (car xex))
                (ytrans (if (> dir 0)(car yex) (cdr yex)))
```

```
                ;; Add last point.
                (coord-list (append coords '((1.0 . 0.0))))
                (uplist
                 (map pair-to-list
                      (normalize-coords coord-list lenx (* leny 2) dir))))
            (ly:stencil-translate
             (my-c-p-s uplist used-thick)
             (cons xtrans ytrans)))
          '())))

% Define a default tie shape consisting of three straight lines.
#(define flare-tie
   (flared-tie '((0.1 . 0.3) (0.9 . 0.3))))

\relative c' {
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <a c e a c e a c e>~ q
  \once \override Tie.stencil = #flare-tie
  q~ q\break

  <>^\markup \small \typewriter "height-limit = 14"
  \override Tie.details.height-limit = 14
  a'4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

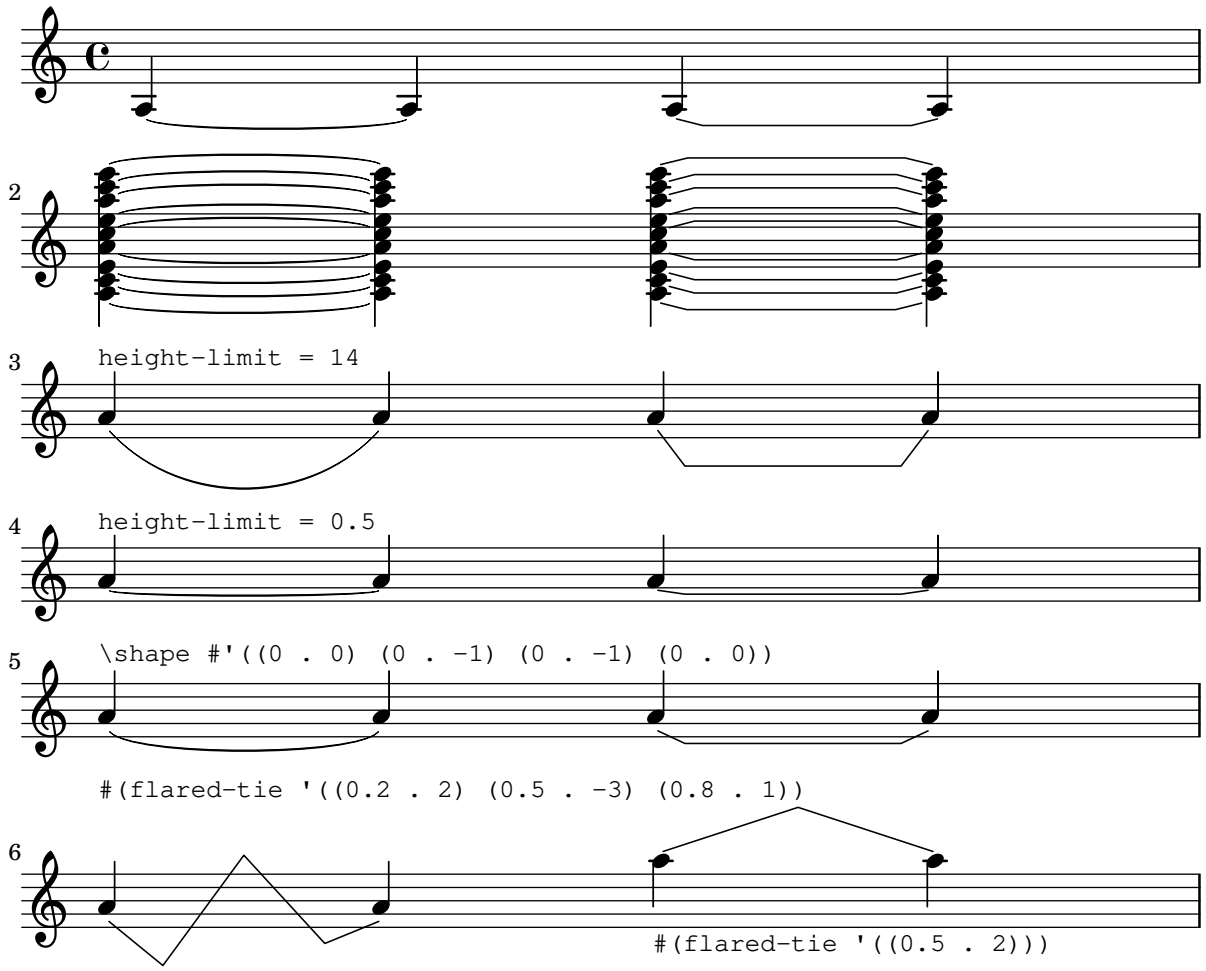
  <>^\markup \small \typewriter "height-limit = 0.5"
  \override Tie.details.height-limit = 0.5
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  \revert Tie.details.height-limit

  <>^\markup \small \typewriter
            "\\shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0))"
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a
  \once \override Tie.stencil = #flare-tie
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a \break

  <>^\markup \small \typewriter
            "#(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1))"
  \once \override Tie.stencil =
        #(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))
  a4~ a
  <>_\markup \small \typewriter
            "#(flared-tie '((0.5 . 2)))"
```

```
  \once \override Tie.stencil = #(flared-tie '((0.5 . 2)))
  a'4~ a
}
```
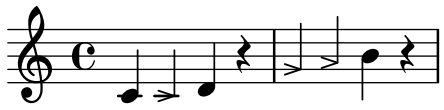


Notación de los golpes de lengua de la flauta

Es posible indicar técnicas de articulación especiales como el golpe de lengua de la flauta ("tongue slap") sustituyendo la cabeza de la figura con el glifo adecuado. Para ello podemos trazar la cabeza de la nota en forma de acento con \markup.

```
slap =
#(define-music-function (music) (ly:music?)
  #{
    \temporary \override NoteHead.stencil =
      #ly:text-interface::print
    \temporary \override NoteHead.text =
      \markup
        \translate #'(1 . 0)
          \override #'(thickness . 1.4)
            \overlay { \draw-line #'(-1.2 . 0.4)
                       \draw-line #'(-1.2 . -0.4) }
    \temporary \override NoteHead.stem-attachment =
      #(lambda (grob)
        (let* ((stem (ly:grob-object grob 'stem))
               (dir (ly:grob-property stem 'direction UP))
```

```
              (is-up (eqv? dir UP)))
          (cons dir (if is-up 0 -0.8))))
    #music
    \revert NoteHead.stencil
    \revert NoteHead.text
    \revert NoteHead.stem-attachment
  #})

\relative c' {
  c4 \slap c d r
  \slap { g4 a } b r
}
```
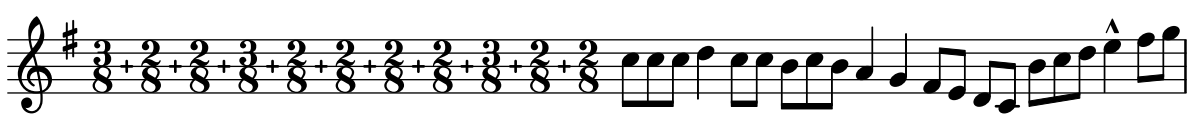


Indicaciones de compás polimétricas profundamente personalizadas

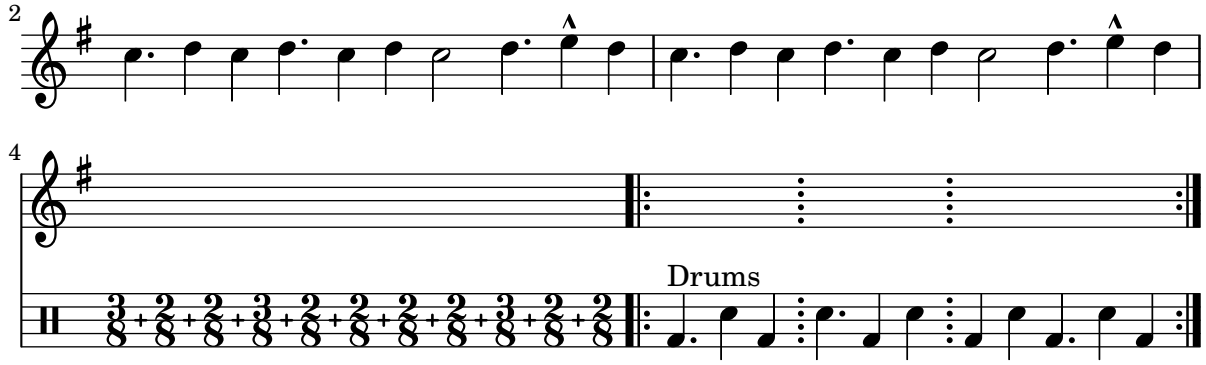
Aunque aquí el elemento más esencial no era la indicación de compás polimétrica que se muestra, se ha incluido para mostrar el pulso de la pieza, que por cierto es la plantilla de una canción de los Balcanes real.

```
melody = \relative c'' {
  \key g \major
  \time #'((3 . 8) (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8)
           (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8))
  \set Timing.beamExceptions = #'()
  \set Timing.beatStructure = 3,2,2,3,2,2,2,2,3,2,2
  c8 c c d4 c8 c b c b a4 g fis8 e d c b' c d e4-^ fis8 g \break
  c,4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
}

drum = \new DrumStaff \drummode {
  \repeat volta 2 {
    bd4.^\markup { Drums } sn4 bd \bar ";"
    sn4. bd4 sn \bar ";"
    bd sn bd4. sn4 bd
  }
}

\new Staff {
  \melody
  \drum
}
```
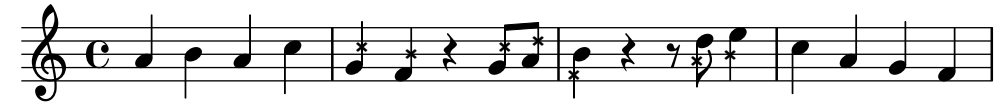




Marcar notas de las partes habladas con unas aspas en la plica

Este ejemplo muestra cómo añadir aspas sobre las plicas. Marcamos el comienzo de una sección hablada con la instrucción \speakOn, y el final con \speakOff.

```
speakOn = \override Stem.stencil =
  #(lambda (grob)
     (let* ((x-parent (ly:grob-parent grob X))
            (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
       (if is-rest?
           empty-stencil
           (ly:stencil-combine-at-edge
            (ly:stem::print grob)
            Y
            (- (ly:grob-property grob 'direction))
            (grob-interpret-markup
     grob
             (markup #:center-align #:fontsize -4
                     #:musicglyph "noteheads.s2cross"))
            -1.7))))

speakOff = \revert Stem.stencil

\new Staff {
  \relative c'' {
    a4 b a c
    \speakOn
    g4 f r g8 a
    b4 r r8 d e4
    \speakOff
    c4 a g f
  }
}
```



Armaduras de tonalidad no tradicionales

La muy utilizada instrucción \key establece la propiedad keyAlterations dentro del contexto Staff.

Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

```
\set Staff.keyAlterations =
  #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)
```
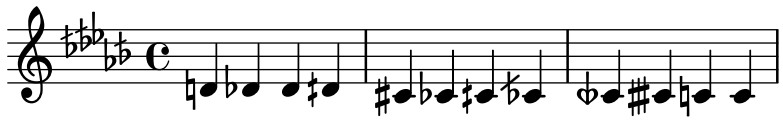
donde, para cada elemento dentro de la lista, *octava* especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), *paso* especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y *alteración* es SHARP, FLAT o DOUBLE-SHARP, etc., precedido de una coma.

De forma alternativa, podemos usar el formato más conciso (*paso . alteración*) para cada elemento de la lista si se usa la misma alteración en todas las octavas.

Para escalas microtonales en las que un "sostenido" no son 100 cents, *alteración* se refiere a la proporción de un tono entero de 200 cents.

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
                                 (6 . ,SEMI-FLAT))
  % \set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Imprimir música que tenga compases distintos en cada pentagrama

En el siguiente fragmento de código, dos partes distintas tienen un compás completamente diferente y sin embargo se mantienen sincronizadas. Las barras de compás ya no se pueden imprimir en el nivel de Score; para permitir barras de compás independientes en cada parte se trasladan los grabadores Default_barline_engraver y Timing_translator desde el contexto de partitura Score hasta el contexto de pentagrama Staff.

Si son necesarios números de compás, el grabador de números de compás Bar_number_engraver también debe trasladarse, ya que descansa en propiedades fijadas por el Timing_translator; se puede usar un bloque \with para añadir números de compás al pentagrama apropiado.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}

\layout {
  \context {
    \Score
```
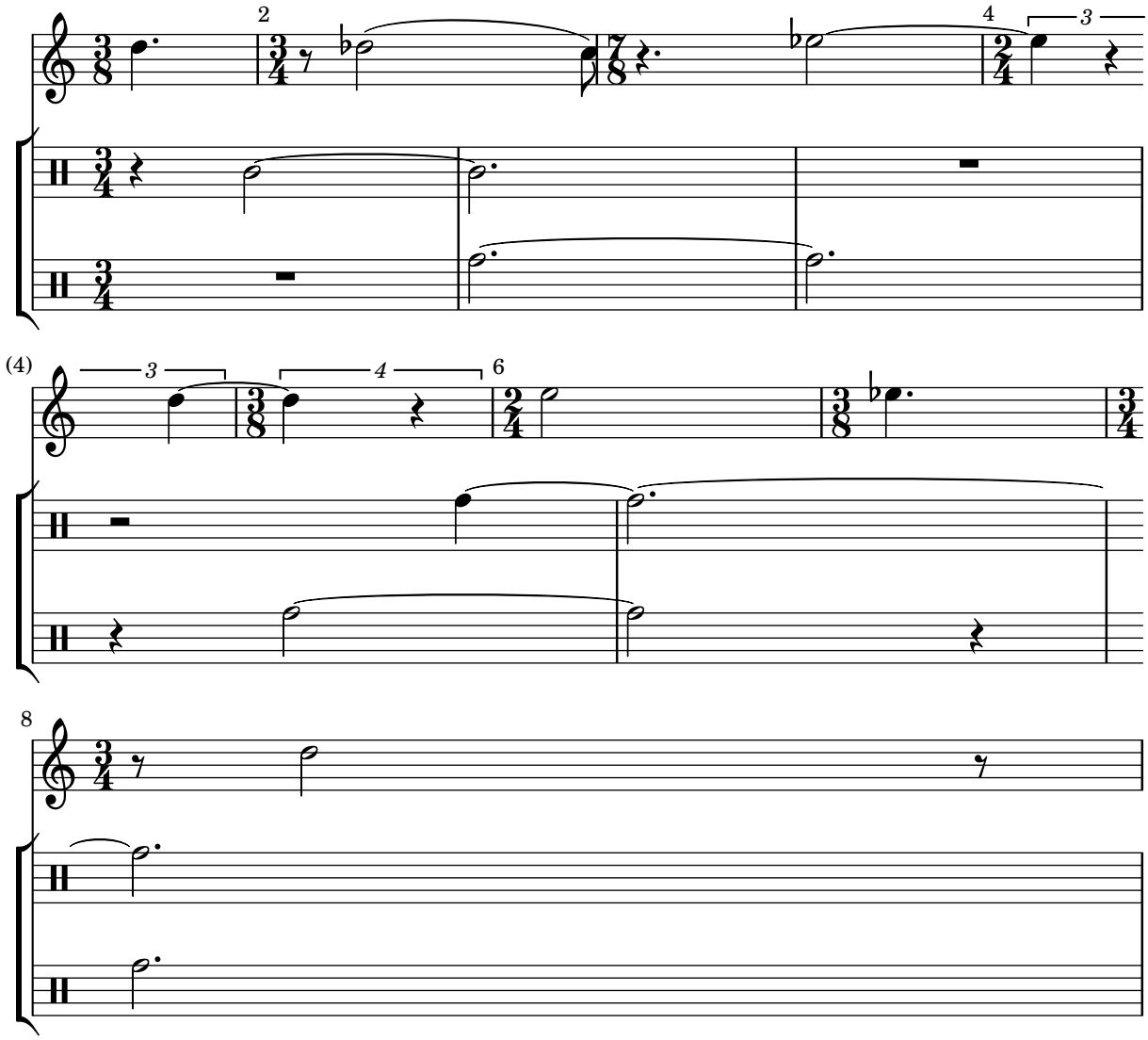
```
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. ees''2 ~ |
    \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
    \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
    \time 2/4 e''2 |
    \time 3/8 es''4. |
    \time 3/4 r8 d''2 r8 |
  }
>>

Perkussion = \new StaffGroup <<
  \new Staff <<
    \global
    {
      \clef percussion
      \time 3/4 r4 c'2 ~ |
      c'2. |
      R2. |
      r2 g'4 ~ |
      g'2. ~ |
      g'2. |
    }
  >>
  \new Staff <<
    \global {
      \clef percussion
```

```
      \time 3/4 R2. |
      g'2. ~ |
      g'2. |
      r4 g'2 ~ |
      g'2 r4 |
      g'2. |
    }
  >>
>>

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}
```

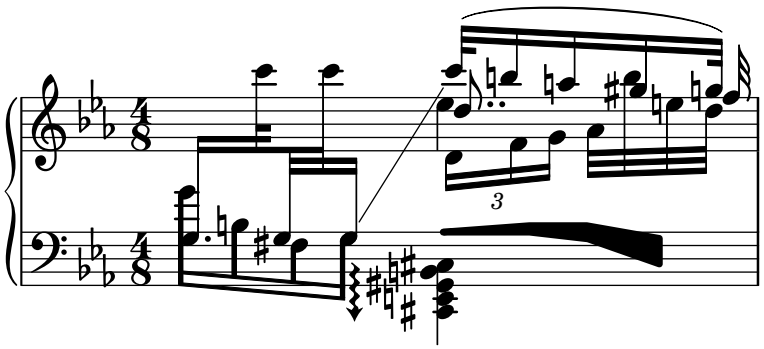


Screech and boink

Notación musical compleja y aleatoria.

```
\score {
  \context PianoStaff <<
    \new Staff = "up" {
      \time 4/8
      \key c \minor
      <<
        {
          \revert Stem.direction
          \change Staff = down
          \set subdivideBeams = ##t
          g16.[
          \change Staff = up
          c'''32
          \change Staff = down
          g32
          \change Staff = up
          c'''32
          \change Staff = down
          g16]
          \change Staff = up
          \stemUp
          \set followVoice = ##t
          c'''32([ b''16 a''16 gis''16 g''32)]
        }
        \\
        { s4 \tuplet 3/2 { d'16[ f' g'] } as'32[ b''32 e'' d''] }
        \\
        { s4 \autoBeamOff d''8.. f''32  }
        \\
        { s4 es''4 }
      >>
    }
    \new Staff = "down" {
      \clef bass
      \key c \minor
      \set subdivideBeams = ##f
      \override Stem.french-beaming = ##t
      \override Beam.beam-thickness = #0.3
      \override Stem.thickness = #4.0
      g'16[ b16 fis16 g16]
      <<
        \makeClusters {
          as16 <as b> <g b> <g cis>
        }
        \\
        {
          \override Staff.Arpeggio.arpeggio-direction = #DOWN
          <cis, e, gis, b, cis>4\arpeggio
        }
      >>
    }
  >>
```

```
  \midi {
    \tempo 8 = 60
  }
  \layout {
    ragged-right = ##t
    \context {
      \Staff
      \consists "Horizontal_bracket_engraver"
    }
  }
}
```



Stemlets (plicas de tamaño reducido)

En ciertas convenciones notacionales se permite que las barras se extiendan por encima de los silencios. Dependiendo de nuestras preferencias, estas barras pueden presentar pequeñas plicas o 'stemlets' para que el ojo aprecie más fácilmente el ritmo, y en algunos casos de música moderna el propio silencio se omite y permanecen solamente las pequeñas plicas sin cabeza.

Este fragmento de código muestra una progresión a partir de la notación tradicional, pasando por barras sobre silencios y plicas sobre los silencios, hasta las plicas solamente. Las pequeñas plicas o Stemlets se generan sobreescribiendo la propiedad `stemlet-length` del objeto `Stem`, y los silencios se ocultan usando `\hide`.
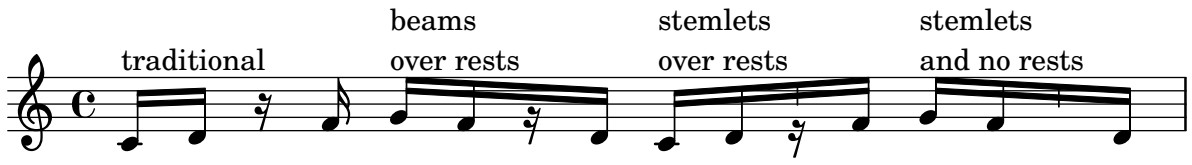
Algunos elementos de `\markup` se incluyen en la fuente para imprimir títulos sobre cada estilo de notación.

```
\paper {
  ragged-right = ##f
}

{
  c'16^\markup { traditional } d' r f'
  g'16[^\markup \column { "beams" "over rests" } f' r d']

  % N.B. use Score.Stem to set for the whole score.
  \override Staff.Stem.stemlet-length = #0.75

  c'16[^\markup \column { "stemlets" "over rests" } d' r f']
  g'16[^\markup \column { "stemlets" "and no rests" } f'
  \once \hide Rest
  r16 d']
}
```


traditional
beams
over rests
stemlets
over rests
stemlets
and no rests

17 Ancient notation

See also Sección “Ancient notation” en *Notation Reference*.

Añadir un bajo cifrado encima o debajo de las notas
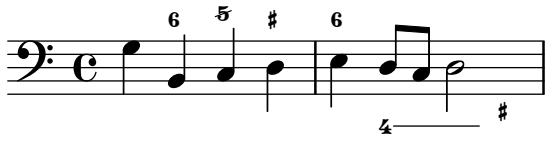
Al escribir un bajo cifrado, podemos situar las cifras encima o debajo de las notas del bajo mediante las instrucciones `\bassFigureStaffAlignmentDown` y `\bassFigureStaffAlignmentUp`. Anteponemos `\once` a la instrucción si queremos modificar solo la cifra siguiente.

La instrucción `\bassFigureStaffAlignmentNeutral` reinicia la dirección del bajo cifrado a su valor predeterminado.

```
bass = {
  \clef bass
  g4 b, c d |
  e d8 c d2
}

continuo = \figuremode {
  <_>4 <6>4 <5/>4
  \bassFigureStaffAlignmentUp
  <_+>4 <6> |
  \set Staff.useBassFigureExtenders = ##t
  \bassFigureStaffAlignmentDown
  <4>4. <4>8 <_+>4
}

\score {
  <<
    \new Staff = bassStaff \bass
    \context Staff = bassStaff \continuo
  >>
}
```



Tipografía de música antigua

Este fragmento de código presenta muchos de los símbolos incluidos en la fuente Emmentaler que LilyPond usa para tipografiar la música antigua.

```
m = { c1 e f ges cis' \bar "||" }

\score {
  \new VaticanaVoice {
    \clef "vaticana-fa2"
    \key es \major
    \textMark \markup \rounded-box "Vaticana clefs, custos and note heads"

    \override NoteHead.style = #'vaticana.punctum
    <>^"vaticana.punctum" \m
```

```
\override NoteHead.style = #'vaticana.inclinatum
<>^"vaticana.inclinatum" \m

\override NoteHead.style = #'vaticana.quilisma
<>^"vaticana.quilisma" \m

\clef "vaticana-fa1"
\override NoteHead.style = #'vaticana.plica
<>^"vaticana.plica" \m

\override NoteHead.style = #'vaticana.reverse.plica
<>^"vaticana.reverse.plica" \m

\override NoteHead.style = #'vaticana.punctum.cavum
<>^"vaticana.punctum.cavum" \m

\override NoteHead.style = #'vaticana.lpes
<>^"vaticana.punctum.lpes" \m

\override NoteHead.style = #'vaticana.upes
<>^"vaticana.punctum.upes" \m

\override NoteHead.style = #'vaticana.vupes
<>^"vaticana.punctum.vupes" \m

\override NoteHead.style = #'vaticana.linea.punctum
<>^"vaticana.punctum.linea" \m

\override NoteHead.style = #'vaticana.epiphonus
<>^"vaticana.punctum.epiphonus" \m

\override NoteHead.style = #'vaticana.cephalicus
<>^"vaticana.punctum.cephalicus" \m

\break

\textMark \markup \rounded-box "Medicaea clefs, custos and note heads"
\set VaticanaStaff.alterationGlyphs =
  #alteration-medicaea-glyph-name-alist
\override VaticanaStaff.Custos.style = #'medicaea

\clef "medicaea-fa2"
\override NoteHead.style = #'medicaea.punctum
<>^"medicaea.punctum" \m

\clef "medicaea-do2"
\override NoteHead.style = #'medicaea.inclinatum
<>^"medicaea.inclinatum" \m

\override NoteHead.style = #'medicaea.virga
<>^"medicaea.virga" \m
```

```
    \clef "medicaea-fa1"
    \override NoteHead.style = #'medicaea.rvirga
    <>^"medicaea.rvirga" \m

    \break

    \textMark \markup \rounded-box "Hufnagel clefs, custos and note heads"
    \set Staff.alterationGlyphs =
      #alteration-hufnagel-glyph-name-alist
    \override VaticanaStaff.Custos.style = #'hufnagel
    \clef "hufnagel-fa2"

    \break
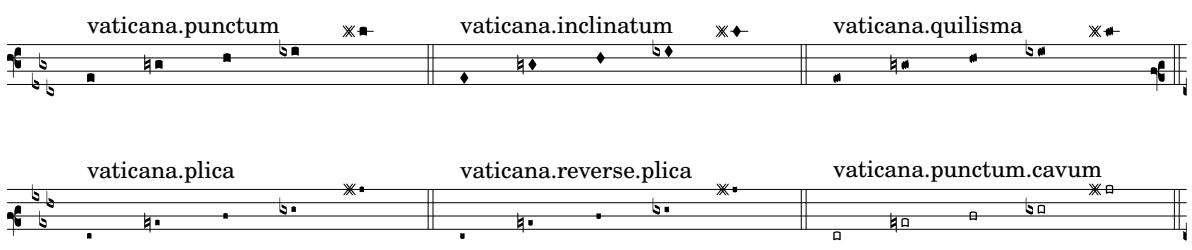
    \override NoteHead.style = #'hufnagel.punctum
    <>^"hufnagel.punctum" \m

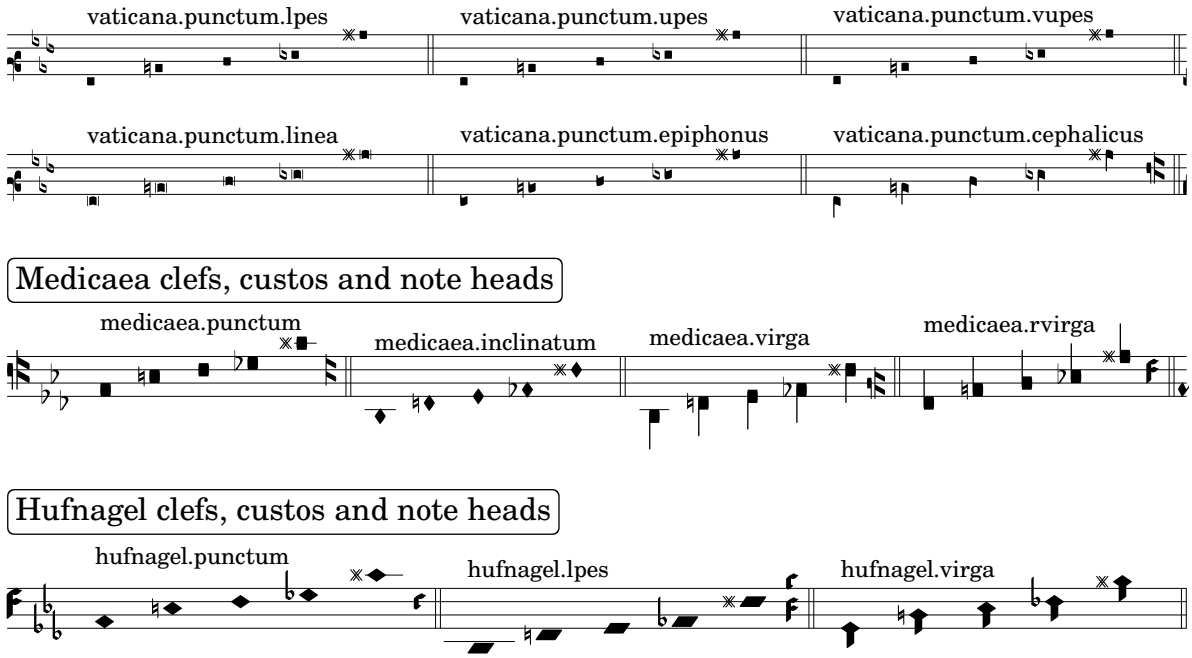
    \clef "hufnagel-do2"
    \override NoteHead.style = #'hufnagel.lpes
    <>^"hufnagel.lpes" \m

    \clef "hufnagel-do-fa"
    \override NoteHead.style = #'hufnagel.virga
    <>^"hufnagel.virga" \m
  }

  \layout {
    \context {
      \Score
      \override TextScript.font-size = #-2
      \override TextMark.break-align-symbols = #'(left-edge clef staff-bar)
      \override TextMark.padding = 4
      \omit BarNumber
    }
    \context {
      \VaticanaStaff
      alterationGlyphs =
        #alteration-vaticana-glyph-name-alist
    }
  }
}
```

Vaticana clefs, custos and note heads





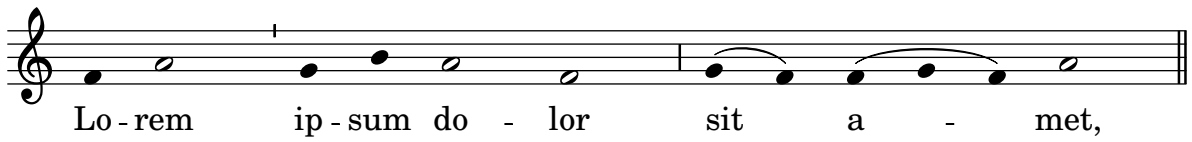
Ancient notation template – modern transcription of Gregorian music

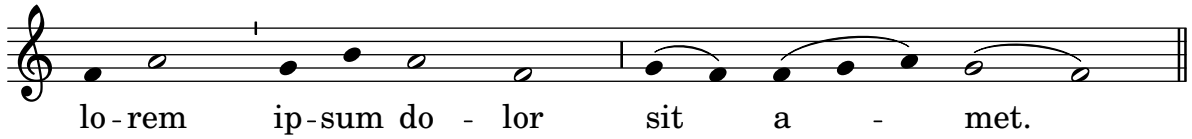
This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter note heads, and special marks, indicating rests of different length.

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g f) a2 \finalis \break
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g a) g2( f) \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met,
  lo -- rem ip -- sum do -- lor sit a -- met.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" \chant
    \new GregorianTranscriptionLyrics = "one" \lyricsto melody \verba
  >>
}
```

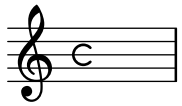




Indicaciones de compás antiguas

Las indicaciones de compás también se pueden grabar en estilo antiguo.

```
{
  \override Staff.TimeSignature.style = #'neomensural
  s1
}
```

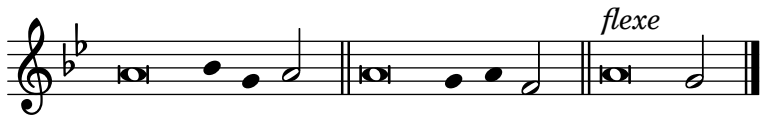


Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}
```



Custos

Se pueden tipografiar «custos» en diferentes estilos.

```
\layout {
  ragged-right = ##t
}

\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

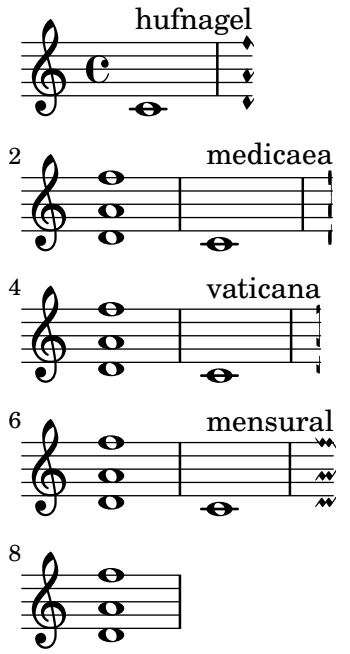
    \override Staff.Custos.style = #'hufnagel
```

```
    c1^"hufnagel" \break
    <d a' f'>1

    \override Staff.Custos.style = #'medicaea
    c1^"medicaea" \break
    <d a' f'>1

    \override Staff.Custos.style = #'vaticana
    c1^"vaticana" \break
    <d a' f'>1

    \override Staff.Custos.style = #'mensural
    c1^"mensural" \break
    <d a' f'>1
  }
}
```



Incipits

Al transcribir música mensural, es útil poner un incipit al comienzo de la pieza para indicar la tonalidad y el compás originales. Aunque en la actualidad los músicos están acostumbrados a las líneas divisorias para reconocer más rápidamente los patrones rítmicos, en el período de la música mensural aún no se habían inventado las barras de compás; de hecho, el metro cambiaba con frecuencia cada pocas notas. Como compromiso, las barras de compás se imprimían muchas veces entre las pautas en lugar de dibujarse atravesando las líneas del pentagrama.

```
% A short excerpt from the Jubilate Deo by Orlande de Lassus

global = {
  \set Score.skipBars = ##t
  \key g \major
  \time 4/4

  % the actual music
```

```
  \skip 1*8

  % let finis bar go through all staves
  \override Staff.BarLine.transparent = ##f

  % finis bar
  \bar "|."
}

discantusIncipit = \new PetrucciStaff {
  \clef "petrucci-c1"
  \key f \major
  \time 2/2
  c''1.
}

discantusNotes = {
  \transpose c' c'' {
    \clef "treble"
    d'2. d'4 |
    b e' d'2 |
    c'4 e'4.( d'8 c' b |
    a4) b a2 |
    b4.( c'8 d'4) c'4 |
    \once \hide NoteHead
    c'1 |
    b\breve |
  }
}

discantusLyrics = \lyricmode {
  Ju -- bi -- la -- te De -- o,
  om -- nis ter -- ra, __ om-
  "..."
  -us.
}

altusIncipit = \new PetrucciStaff {
  \clef "petrucci-c3"
  \key f \major
  \time 2/2
  e'1\rest f'1.
}

altusNotes = {
  \transpose c' c'' {
    \clef "treble"
    r2 g2. e4 fis g |
    a2 g4 e |
    fis g4.( fis16 e fis4) |
    g1 |
    \once \hide NoteHead
```

```
    g1 |
    g\breve |
  }
}

altusLyrics = \lyricmode {
  Ju -- bi -- la -- te
  De -- o, om -- nis ter -- ra,
  "..."
  -us.
}

tenorIncipit = \new PetrucciStaff {
  \clef "petrucci-c4"
  \key f \major
  \time 2/2
  r\longa
  r\breve
  r1 c'1.
}

tenorNotes = {
  \transpose c' c' {
    \clef "treble_8"
    R1 |
    R1 |
    R1 |
    % two measures
    r2 d'2. d'4 b e' |
    \once \hide NoteHead
    e'1 |
    d'\breve |
  }
}

tenorLyrics = \lyricmode {
  Ju -- bi -- la -- te
  "..."
  -us.
}

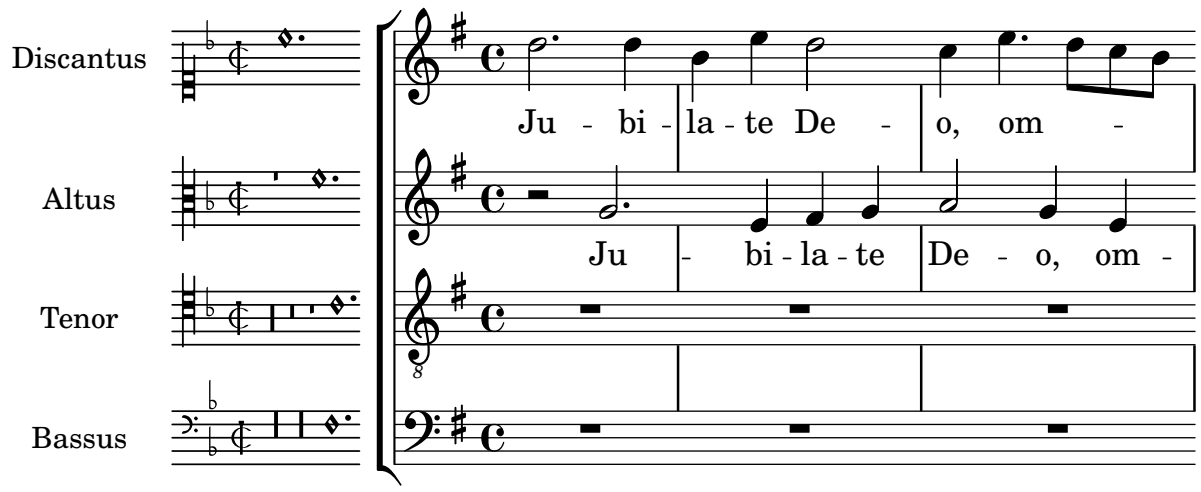
bassusIncipit = \new PetrucciStaff {
  % The original print shows the b flat
  % for the f major key signature twice.
  \override Staff.KeySignature.flat-positions = #'((-7 . 6))
  \clef "mensural-f"
  \key f\major
  \time 2/2
  \tweak Y-offset #1 r\longa \tweak Y-offset #1 r\longa
  f1.
}
```

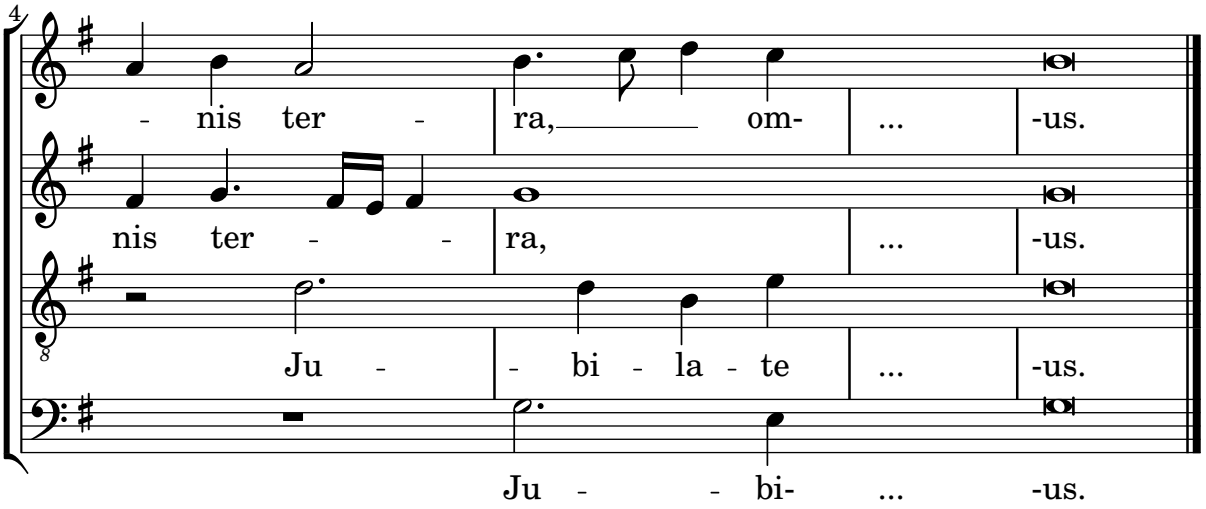
```
bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \hide NoteHead
    e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  Ju -- bi-
  "..."
  -us.
}

\score {
  <<
    \new StaffGroup = choirStaff <<
      \new Voice = "discantusNotes" <<
        \set Staff.instrumentName = "Discantus"
        \incipit #1 \discantusIncipit
        \global
        \discantusNotes
      >>
      \new Lyrics \lyricsto discantusNotes { \discantusLyrics }
      \new Voice = "altusNotes" <<
        \set Staff.instrumentName = "Altus"
        \global
        \incipit #1 \altusIncipit
        \altusNotes
      >>
      \new Lyrics \lyricsto altusNotes { \altusLyrics }
      \new Voice = "tenorNotes" <<
        \set Staff.instrumentName = "Tenor"
        \global
        \incipit #1 \tenorIncipit
        \tenorNotes
      >>
      \new Lyrics \lyricsto tenorNotes { \tenorLyrics }
      \new Voice = "bassusNotes" <<
        \set Staff.instrumentName = "Bassus"
        \global
        \incipit #1 \bassusIncipit
        \bassusNotes
      >>
      \new Lyrics \lyricsto bassusNotes { \bassusLyrics }
    >>
```

```
  >>
  \layout {
    \context {
      \Score
      %% no bar lines in staves or lyrics
      \hide BarLine
    }
    %% the next two instructions keep the lyrics between the bar lines
    \context {
      \Lyrics
      \consists "Bar_engraver"
      \consists "Separating_line_group_engraver"
    }
    \context {
      \Voice
      %% no slurs
      \hide Slur
      %% Comment in the below "\remove" command to allow line
      %% breaking also at those bar lines where a note overlaps
      %% into the next measure.  The command is commented out in this
      %% short example score, but especially for large scores, you
      %% will typically yield better line breaking and thus improve
      %% overall spacing if you comment in the following command.
      %%\remove "Forbid_line_break_engraver"
    }
    indent = 5\cm
    incipit-width = 2.5\cm
  }
}
```





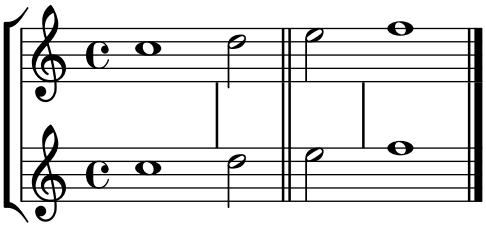
Disposición *Mensurstriche* (líneas divisorias entre pentagramas)

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir fijando `measureBarType` al valor `"-span|"` y usando un contexto de agrupamiento que admita barras extendidas, como `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



Estilos de silencios

Los silencios se pueden imprimir en distintos estilos.

```
restsA = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
  s64 s128 s256 s512 s1024 s1024
}
restsB = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
  r64 r128 r256 r512 r1024 s1024
}

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

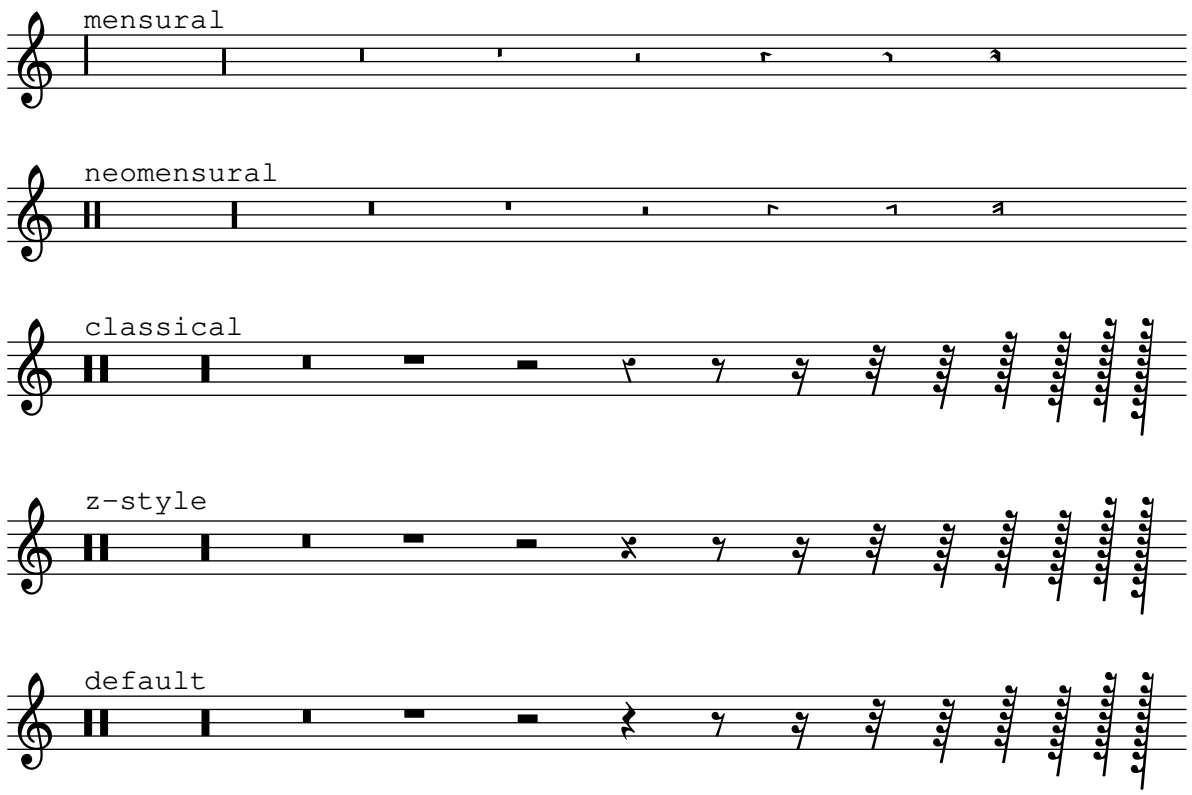
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>^\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>^\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>^\markup \typewriter { default } \restsB \bar "" \break
}
```



Uso de etiquetas para producir música mensural y moderna a partir del mismo código fuente

Usando etiquetas es posible producir las dos notaciones mensural y moderna a partir del mismo código fuente de la música. En este fragmento se introduce la función `\menrest`, que permite que los silencios mensurales estén a la altura de notas como en el original, pero con silencios modetnos en la posicion del pentagrama estándar.

También se pueden usar etiquetas allí donde se necesiten otras diferencias: por ejemplo usando "silencios de compás completo" (`R1`, `R\breve`, etc.) en música moderna, pero silencios normales (`r1`, `r\breve`, etc.) en la versión mensural. La conversión de música mensural a su equivalente moderno se conoce normalmente como *transcription*.

La llamara `c4.\Be c8 c\Am` es lo mismo que `c4.[ c8 c]`. Sin embargo, suprime las advertencias sin se inicia en una nota que no puede llevar barra pero la necesita en todo caso debido al uso del grabador `Completion_heads_engraver`.

[La longitud ligeramente acortada de la línea en la pauta mensural evita el recorte del glifo de los custos cuando LilyPond genera imágenes recortadas muy estrechamente. El uso de `\with-true-dimensions`, tal y como aparece a continuación, evita esto.]

```
\layout {
  line-width = 150\mm
}

menrest = #(define-music-function (note) (ly:music?)
              #{
                \tag #'mens $(make-music 'RestEvent note)
                \tag #'mod $(make-music 'RestEvent note 'pitch '())
              #})
Be = \tag #'mod
       #(begin
          (ly:expect-warning (G_ "stem does not fit in beam"))
          (ly:expect-warning (G_ "beam was started here"))
          (make-span-event 'BeamEvent START))
Am = \tag #'mod ]

MenStyle = {
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
  \omit Slur
  \omit Beam
}

finalis = \section

Music = \relative c'' {
  \key f \major
  g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
  g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}

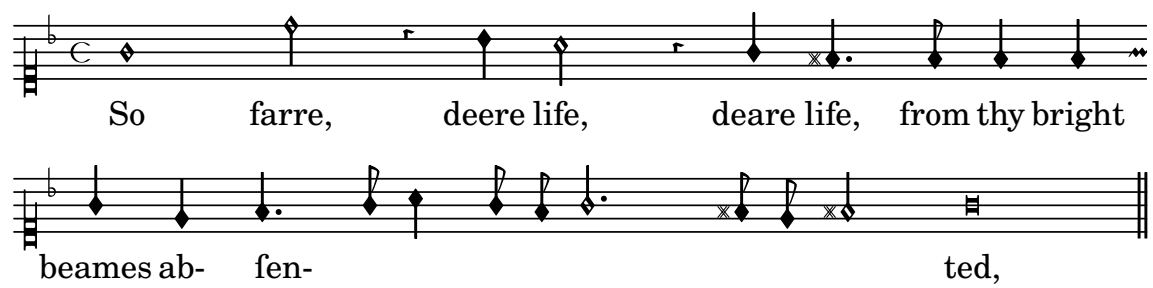
MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
```

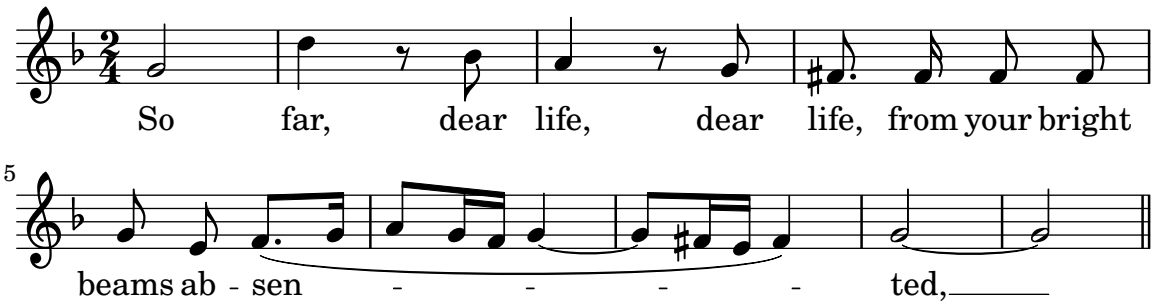
```
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}

\score {
  \keepWithTag #'mens {
    <<
      \new PetrucciStaff {
        \new PetrucciVoice = "Cantus" {
          \clef "petrucci-c1" \time 4/4 \MenStyle \Music
        }
      }
      \new Lyrics \lyricsto "Cantus" \MenLyr
    >>
  }
  \layout {
    \context {
      \PetrucciVoice
      % No longer necessary starting with version 2.25.23.
      \override Flag.style = #'mensural
    }
  }
}

\markup\vspace #1

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff {
        \new Voice = "Sop" \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver"
        } \shiftDurations 1 0 { \time 2/4 \autoBeamOff \Music }
      }
      \new Lyrics \lyricsto "Sop" \ModLyr
    >>
  }
}
```





Línea vertical como una articulación barroca

Esta línea corta vertical situada encima de la nota es de uso común en música barroca. Su significado varía, pero en general indica notas que se deben tocar con más "peso". El ejemplo siguiente muestra cómo conseguir dicha notación.

```
upline =
\tweak stencil
  #(lambda (grob)
    (grob-interpret-markup grob #{ \markup \draw-line #'(0 . 1) #}))
  \stopped

\relative c' {
  a'4^\upline a( c d')_\upline
}
```



18 World music

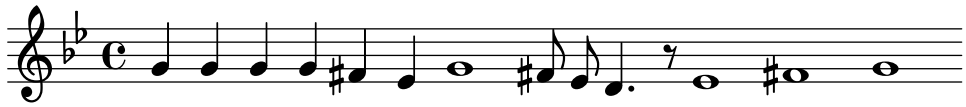
See also Sección “World music” en *Notation Reference*.

Improvisación de música árabe

Para las improvisaciones o *taqasim* que son libres durante unos momentos, se puede omitir la indicación de compás y se puede usar `\cadenzaOn`. Podría ser necesario ajustar el estilo de alteraciones accidentales, porque la ausencia de líneas divisorias hace que la alteración aparezca una sola vez. He aquí un ejemplo de cómo podría ser el comienzo de una improvisación *hijaz*.

```
\include "arabic.ly"

\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
```
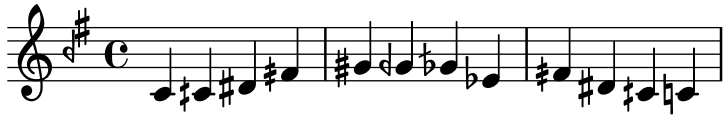


Ejemplo de *makam*

El «Makam» es un tipo de melodía de Turquía que utiliza alteraciones microtonales de 1/9 de tono.

Consulte el archivo de inicio `ly/makam.ly` para ver detalles de los nombres de las notas y las alteraciones.

```
\include "makam.ly"

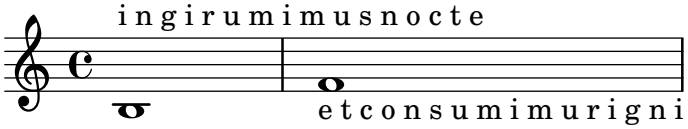
\relative c' {
  \set Staff.keyAlterations = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Impresión de texto de derecha a izquierda

Es posible imprimir texto de derecha a izquierda en un elemento de marcado, como se muestra aquí.

```
{
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```



Ejemplo de *makam* en la música turca

Esta plantilla usa el comienzo de un *Saz Semai* turco bien conocido que es familiar dentro del repertorio, para ilustrar algunos de los elementos de la notación musical de Turquía.

```
#(set-default-paper-size "a6" 'landscape)

\include "turkish-makam.ly"

\header {
  title = "Hüseyni Saz Semaisi"
  composer = "Lavtacı Andon"
  tagline = ##f
}

\relative {
  \set Staff.extraNatural = ##f
  \set Staff.autoBeaming = ##f
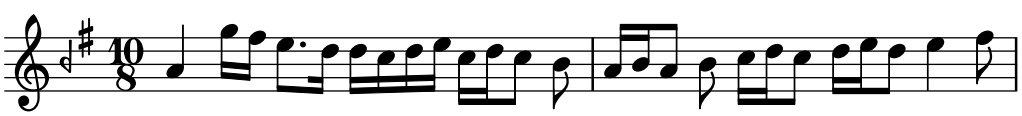
  \key a \huseyni
  \time 10/8
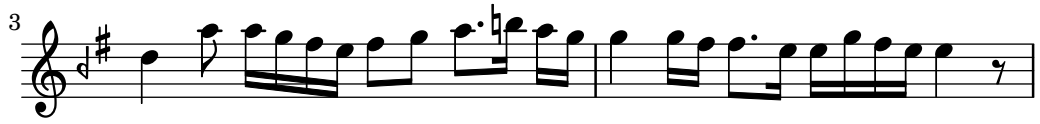
  a'4 g'16[ fb] e8.[ d16] d[ c d e] c[ d c8] bfc |
  a16[ bfc a8] bfc c16[ d c8] d16[ e d8] e4 fb8 |
  d4 a'8 a16[ g fb e] fb8[ g] a8.[ b16] a16[ g] |
  g4 g16[ fb] fb8.[ e16] e[ g fb e] e4 r8 |
}

\layout {
  indent = 0
}
```

Hüseyni Saz Semaisi

Lavtacı Andon




3

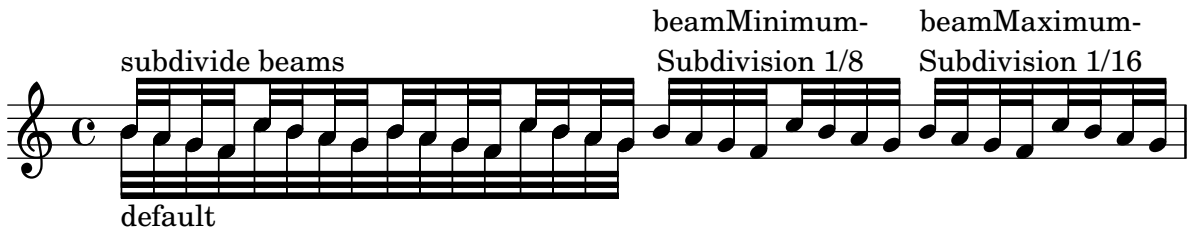
Other collections

19 Automatic notation

Subdivisiones de barra automáticas

Se pueden subdividir las barras automáticamente. Estableciendo la propiedad `subdivideBeams`, las barras se subdividen en siempre que sea posible. Se pueden limitar los intervalos y la profundidad de la subdivisión con las propiedades `beamMinimumSubdivision` y `beamMaximumSubdivision`, respectivamente.

```
\new Staff {
  \relative c'' {
    <<
      {
        \voiceOne
        \set subdivideBeams = ##t
        b32[^"subdivide beams" a g f c' b a g
        b32 a g f c' b a g]
      }
      \new Voice {
        \voiceTwo
        b32_"default"[ a g f c' b a g
        b32 a g f c' b a g]
      }
    >>
    \oneVoice
    \once \set beamMinimumSubdivision = #1/8
    b32^\markup \center-column { "beamMinimum-"
                                 "Subdivision 1/8" } [ a g f c' b a g]
    \once \set beamMaximumSubdivision = #1/16
    b32^\markup \center-column { "beamMaximum-"
                                 "Subdivision 1/16" } [ a g f c' b a g]
  }
}
```



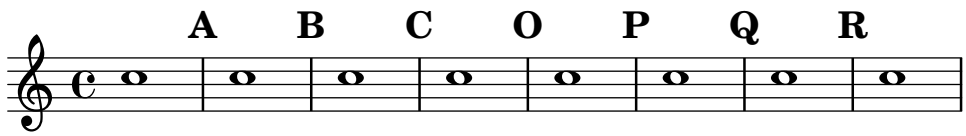
Forzar que las marcas de ensayo comiencen en una letra o número dados

Este fragmento de código muestra cómo obtener marcas de ensayo ordenadas automáticamente, pero partiendo de la letra o número deseados.

```
\relative c'' {
  \override Score.RehearsalMark.Y-offset = #3.5

  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
```

```
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1
}
```



Generación de partituras completas y partes de libro en Scheme sin usar el analizador sintáctico

Internamente, una partitura de LilyPond no es más que una expresión de Scheme, generada por el analizador sintáctico de LilyPond. Usando Scheme, también podemos generar automáticamente una partitura sin ningún archivo de entrada. Si tenemos la expresión musical en Scheme, se puede generar la partitura simplemente llamando a

```
(scorify-music music)
```

sobre la música. Así se genera un objeto partitura, para el que podemos entonces fijar un bloque layout personalizado con

```
(let* ((layout (ly:output-def-clone $defaultlayout)))
   ; modificamos el layout aquí, y después lo asignamos:
   (ly:score-add-output-def! score layout))
```

Finalmente, todo lo que tenemos que hacer es pasar esta partitura a lilypond para que realice la composición tipográfica. Este fragmento de código define las funciones `(add-score score)`, `(add-text text)` y `(add-music music)` para pasar una partitura completa, elementos de marcado o algo de música a LilyPond para su composición tipográfica.

Este fragmento de código también funciona para la tipografía de partituras dentro de un bloque `\book {...}`, así como partituras normales del nivel superior jerárquico. Para conseguirlo, cada una de las partituras destinadas a su tipografiado se añaden al final de una lista de partituras del nivel superior jerárquico y se modifica el toplevel-book-handler (que es una función de Scheme que se llama para procesar un libro una vez que se ha cerrado el bloque `\book{..}`) para insertar todas las partituras así coleccionadas al libro.

Nota: por razones técnicas, solo se muestra el primer `\book`, debido a que las otras instrucciones `\book` crean archivos de salida adicionales.

```
#(define-public (add-score score)
   (module-define! (current-module) 'toplevel-scores
                   (cons score (module-ref (current-module) 'toplevel-scores '()))))

#(define-public (add-text text)
   (add-score (list text)))

#(define-public (add-music music)
   (collect-music-aux (lambda (score)
                        (add-score score))
                      music))

#(define-public (toplevel-book-handler book)
   (map (lambda (score)
```

```
          (ly:book-add-score! book score))
        (reverse (ly:book-lookup book 'toplevel-scores)))
   (print-book-with-defaults book))

#(define-public (book-score-handler book score)
   (add-score score))

#(define-public (book-text-handler book text)
   (add-text text))

#(define-public (book-music-handler book music)
   (add-music music))


% Some example code to show how to use these functions.  Each call to
% `\oneNoteScore` constructs a global markup followed by a single
% staff with a single quarter note.  The pitch of this note is taken
% from the variable `pitch`; the start value 0 corresponds to pitch C.
% After emitting the score, variable `pitch` gets increased by 1.
%
% `\oneNoteScore` calls Scheme function `add-one-note-score` to do all
% the work.

#(define add-one-note-score #f)
#(let ((pitch 0))
   (set! add-one-note-score
         (lambda ()
           (let* ((music
                   (make-music
                    'EventChord
                    'elements (list (make-music
                                     'NoteEvent
                                     'duration (ly:make-duration 2 0 1/1)
                                     'pitch (ly:make-pitch 0 pitch 0)))))
                  (score (scorify-music music))
                  (layout (ly:output-def-clone $defaultlayout))
                  (note-name (case pitch
                               ((0) "do")
                               ((1) "ré")
                               ((2) "mi")
                               ((3) "fa")
                               ((4) "sol")
                               ((5) "la")
                               ((6) "si")
                               (else "huh")))
                  (title (markup #:large #:line
                                 ("Score with a" note-name))))
             (ly:score-add-output-def! score layout)
             (add-text title)
             (add-score score))
           (set! pitch (modulo (1+ pitch) 7)))))
```

```
oneNoteScore =
#(define-void-function () ()
   (add-one-note-score))

\book {
  \oneNoteScore
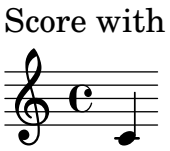
  \paper { tagline = ##f }
}


\book {
  \oneNoteScore
  \oneNoteScore

  \paper { tagline = ##f }
}

% Top-level scores are also handled correctly.
\oneNoteScore
\oneNoteScore
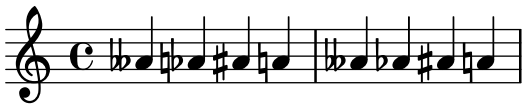
\paper { tagline = ##f }
```



Evitar que se añadan becuadros adicionales automáticamente

Según las reglas tradicionales de composición tipográfica, se imprime un becuadro antes de un sostenido o un bemol cuando se tiene que cancelar un doble sostenido o un doble bemol anterior en la misma nota. Para modificar este comportamiento a la práctica actual, establezca el valor de la propiedad `extraNatural` a `#f` (falso) dentro del contexto de `Staff`.
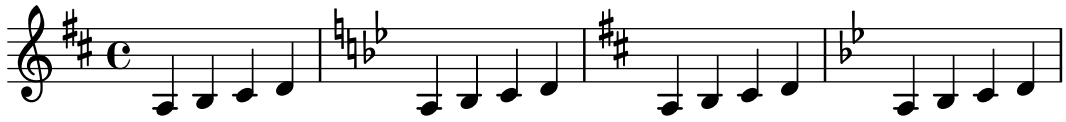
```
\relative c'' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Evitar que se impriman becuadros cuando cambia la armadura

Cuando cambia la armadura de la tonalidad, se imprimen becuadros automáticamente para cancelar las alteraciones de las armaduras anteriores. Esto se puede evitar estableciendo la propiedad `printKeyCancellation` al valor "#f" dentro del contexto `Staff`.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Plantilla de conjunto vocal con reducción de piano automática

Esta plantilla añade una reducción de piano automática a la partitura vocal SATB estándar que se mostró en el fragmento de código "Plantilla de conjunto vocal". Presenta uno de los puntos fuertes de LilyPond: podemos usar una definición de música más de una vez. Si se hace cualquier cambio en las notas de la parte vocal (digamos `tenorMusic`), entonces los cambios se aplican también a la reducción de piano.

```
\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords =\lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
```

```
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

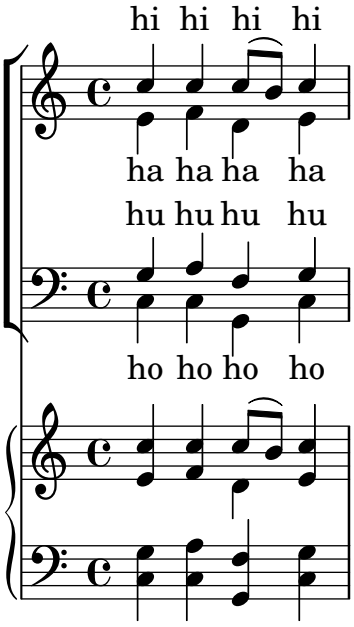
\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"

      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"

      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>

    \new PianoStaff <<
      \new Staff <<
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \sopMusic >>
        << \global \altoMusic >>
      >>
      \new Staff <<
        \clef bass
        \set Staff.printPartCombineTexts = ##f
        \partCombine
```

```
        << \global \tenorMusic >>
        << \global \bassMusic >>
      >>
    >>
  >>
}
```

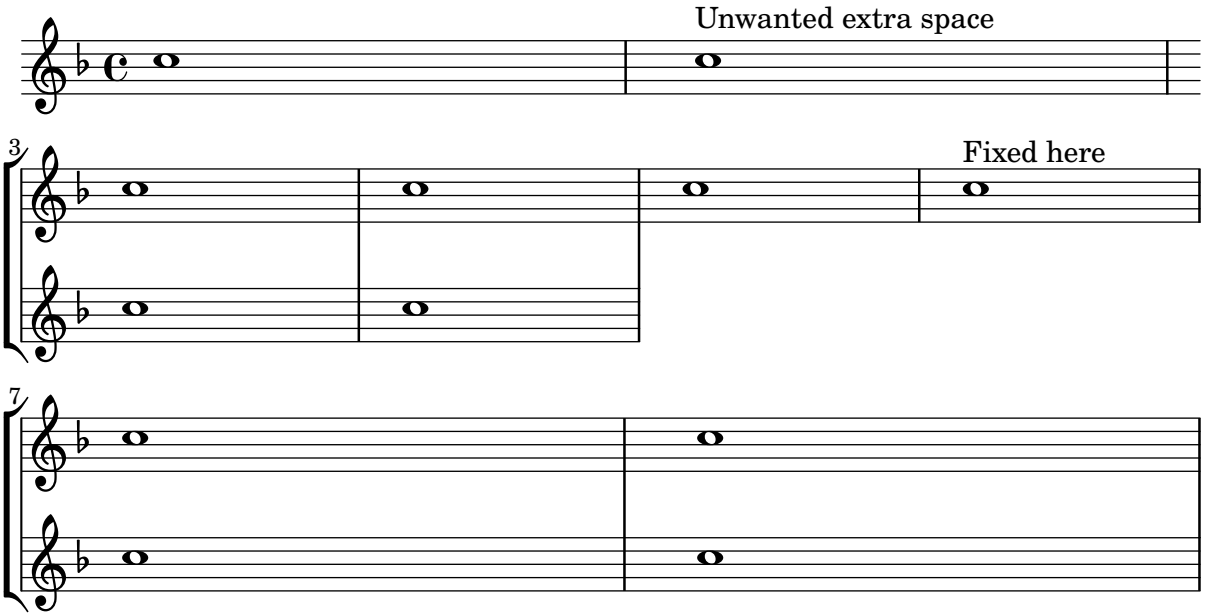


20 Breaks

Añadir un pentagrama adicional en un salto de línea

Al añadir un pentagrama nuevo en un salto de línea, por desgracia se añade un espacio adicional al final de la línea antes del salto (reservado para hacer sitio a un cambio de armadura que de todas formas nunca se imprime). La solución alternativa es establecer la propiedad `Staff.explicitKeySignatureVisibility` del objeto gráfico `Staff` como se muestra en el ejemplo.

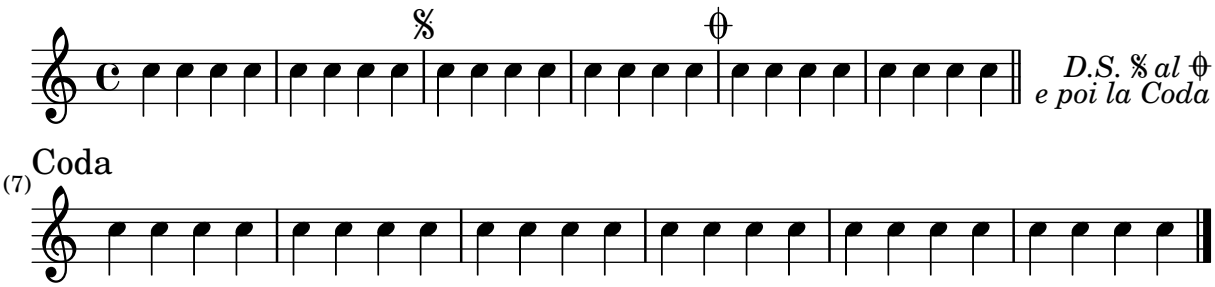
```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
       \new Staff {
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
    c1 | c^"Fixed here" \break
    << { c1 | c }
       \new Staff {
         \once \set Staff.explicitKeySignatureVisibility =
           #end-of-line-invisible
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
  }
}
```



Posicionar los símbolos de repetición segno y coda (con saltos de línea)

Si queremos colocar un símbolo de segno de salida y añadir texto como “D.S. al Coda” junto a él donde normalmente están las líneas del pentagrama, cpodemos usar este código. La coda continúa en una línea nueva. Hay una variante documentada en el fragmento de código, donde la coda se mantiene en la misma línea.

```
\relative c'' {
  c4 c c c | c c c c |
  \repeat segno 2 {
    c4 c c c | c c c c |
    \alternative {
      \volta 1 {
        c4 c c c | c c c c |
        % If you don't use \break at Coda, use \noBreak here
        % and after \bar "" below.
        \noBreak
        \section % double bar line
        \cadenzaOn % pause bar count
        \stopStaff % remove staff lines
        % Increasing the unfold counter will expand the staff-free space
        \repeat unfold 4 {
          s1
          \bar ""
        }
        % Place JumpScript where the staff would normally be.
        \once \override Score.JumpScript.outside-staff-priority = ##f
        \once \override Score.JumpScript.Y-offset = 0
        \startStaff % resume bar count
        \cadenzaOff % show staff lines again
      }
    }
  }
  \sectionLabel "Coda"
  % Show Coda on a new line
  \break
  \*6 { c4 c c c }
  \fine
}
```



Quitar la primera línea vacía

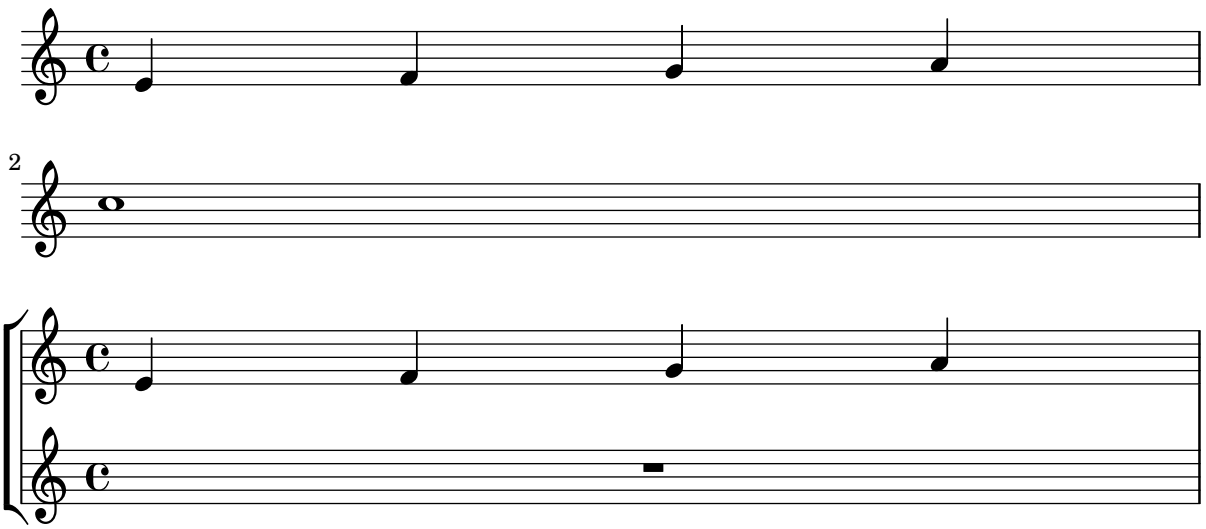
Para eliminar el primer pentagrama vacío de una aprtitura, establecemos la propiedad `remove-first` del objeto gráfico `VerticalAxisGroup` al valor `#t`. Esto se puede hacer globalmente dentro del bloque `\layout`, o localmente dentro del pentagrama concreto que se

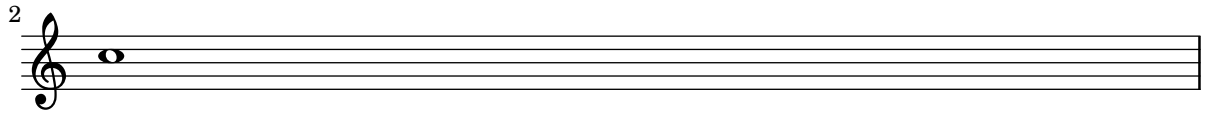
quiere suprimir. En este último caso, tenemos que especificar el contexto (`Staff` se aplica solo al pentagrama actual) delante de la propiedad.

El pentagrama inferior del segundo grupo no se elimina, porque el ajuste solo se aplica al pentagrama concreto dentro del que se escribe.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup.remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>

\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
>>
```




2

21 Connecting notes
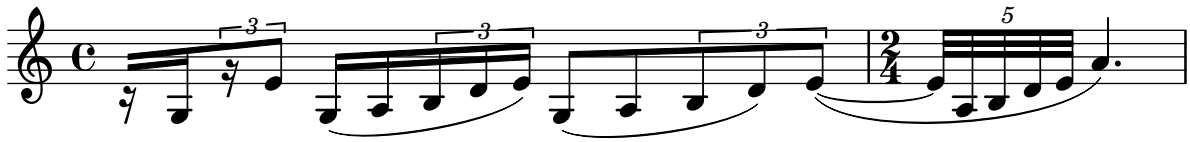
Adding beams, slurs, ties, etc., when using tuplet and non-tuplet rhythms

LilyPond primarily uses postfix syntax for inputting parentheses, brackets, etc., which might feel unintuitive for novices.

For example, when entering a manual beam, the left square bracket has to be placed *after* the starting note and its duration, not before. Similarly, the right square bracket should directly follow the note which is to be at the end of the requested beaming, even if this note happens to be inside a tuplet section.

This snippet demonstrates how to combine manual beaming, manual slurs, ties, and phrasing slurs with tuplet sections (enclosed within curly braces).

```
{
  r16[ g16 \tuplet 3/2 { r16 e'8] }
  g16( a \tuplet 3/2 { b d' e') }
  g8[( a \tuplet 3/2 { b d') e']\( ~ }
  \time 2/4
  \tuplet 5/4 { e'32 a b d' e' } a'4.\)
}
```

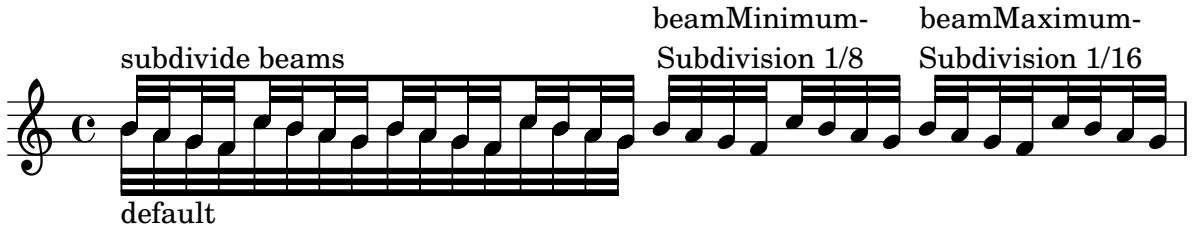


Subdivisiones de barra automáticas

Se pueden subdividir las barras automáticamente. Estableciendo la propiedad `subdivideBeams`, las barras se subdividen en siempre que sea posible. Se pueden limitar los intervalos y la profundidad de la subdivisión con las propiedades `beamMinimumSubdivision` y `beamMaximumSubdivision`, respectivamente.

```
\new Staff {
  \relative c'' {
    <<
      {
        \voiceOne
        \set subdivideBeams = ##t
        b32[^"subdivide beams" a g f c' b a g
        b32 a g f c' b a g]
      }
      \new Voice {
        \voiceTwo
        b32_"default"[ a g f c' b a g
        b32 a g f c' b a g]
      }
    >>
    \oneVoice
    \once \set beamMinimumSubdivision = #1/8
    b32^\markup \center-column { "beamMinimum-"
                                 "Subdivision 1/8" } [ a g f c' b a g]
```

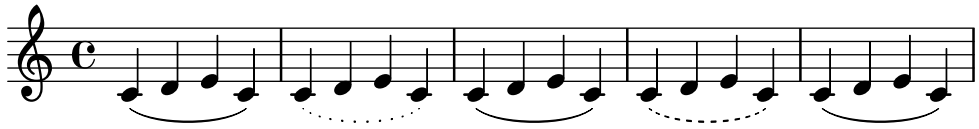
```
    \once \set beamMaximumSubdivision = #1/16
    b32^\markup \center-column { "beamMaximum-"
                                 "Subdivision 1/16" } [ a g f c' b a g]
  }
}
```



Modificar el aspecto continuo de una ligadura de expresión a punteado o intermitente

Se puede cambiar el aspecto de las ligaduras de expresión de continuas a punteadas o intermitentes.

```
\relative c' {
  c4( d e c)
  \slurDotted
  c4( d e c)
  \slurSolid
  c4( d e c)
  \slurDashed
  c4( d e c)
  \slurSolid
  c4( d e c)
}
```



Controlar la visibilidad de los corchetes de grupo especial

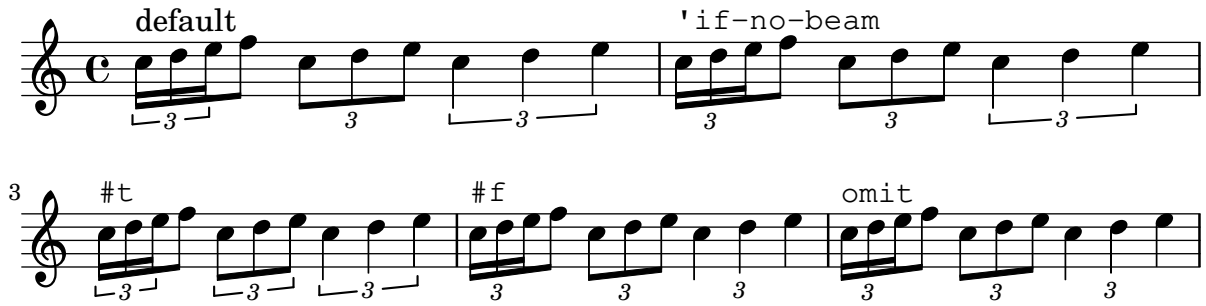
El comportamiento predeterminado de la visibilidad de los corchetes de grupo de valoración especial es imprimir el corchete a no ser que haya una barra de la misma longitud que el grupo especial.

Para controlar la visibilidad de los corchetes de grupo, establezca la propiedad `bracket-visibility` a `#t` (imprimir el corchete siempre), `#if-no-beam` (imprimir el corchete solamente si no hay barra), o `#f` (no imprimir nunca el corchete). Este último equivale de hecho a omitir el objeto `TupletBracket` de la salida impresa.

```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e } f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

\new Voice {
  \relative c' {
```

```
    \override Score.TextMark.non-musical = ##f
    \textMark "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    \textMark \markup \typewriter "'if-no-beam" \music
    \override TupletBracket.bracket-visibility = ##t
    \textMark \markup \typewriter "#t" \music
    \override TupletBracket.bracket-visibility = ##f
    \textMark \markup \typewriter "#f" \music
    \omit TupletBracket
    \textMark \markup \typewriter "omit" \music
  }
}
```
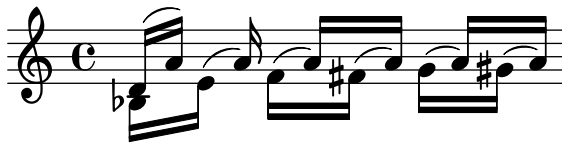


Hacer ligaduras entre voces distintas

En determinadas situaciones es necesario crear ligaduras de expresión entre notas que están en voces distintas.

La solución es añadir notas invisibles a una de las voces utilizando `\hideNotes`.

Este ejemplo es el compás 235 de la Chacona de la segunda Partita para violín solo, BWV 1004, de Bach.

```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```
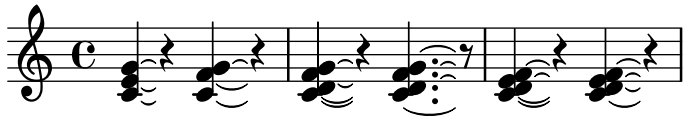


Ligaduras *laissez vibrer*

Las ligaduras "laissez vibrer" (dejar vibrar) tienen un tamaño fijo. Se puede ajustar su posición usando la propiedad `tie-configuration`.

Véase también el fragmento de código "Ligaduras laissez vibrer largas".

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8
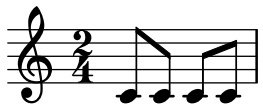
  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
     = #`((-7 . ,DOWN)
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



Controlar manualmente las posiciones de las barras

Se pueden controlar manualmente las posiciones de las barras de corchea sobreescribiendo el valor del parámetro `positions` del objeto gráfico `Beam`.

```
\relative c' {
  \time 2/4
  % from upper staff-line (position 2) to center (position 0)
  \override Beam.positions = #'(2 . 0)
  c8 c
  % from center to one above center (position 1)
  \override Beam.positions = #'(0 . 1)
  c8 c
}
```
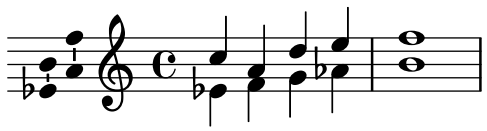


22 Contexts and engravers

See also Sección “Changing defaults” en *Notation Reference* and Sección “Contexts and engravers” en *Learning Manual*.

Añadir un ámbito por voz

Se puede añadir un ámbito por cada voz. En este caso, el ámbito se debe desplazar manualmente para evitar colisiones.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```

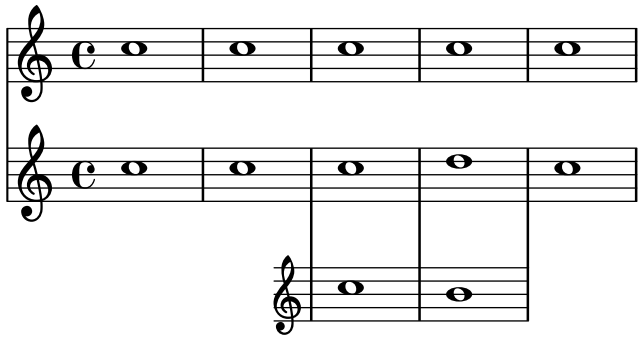


Añadir un pentagrama nuevo

Se puede añadir (posiblemente de forma temporal) un pentagrama nuevo una vez que la pieza ha comenzado.

```
\score {
  <<
    \new Staff \relative c'' {
      c1 | c | c | c | c
    }
    \new StaffGroup \relative c'' {
      \new Staff {
        c1 | c
        <<
          { c1 | d }
          \new Staff {
            \once \omit Staff.TimeSignature
            c1 | b
          }
        >>
        c1
      }
```

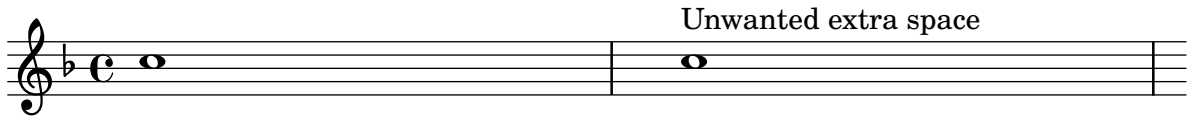
```
    }
  >>
}
```

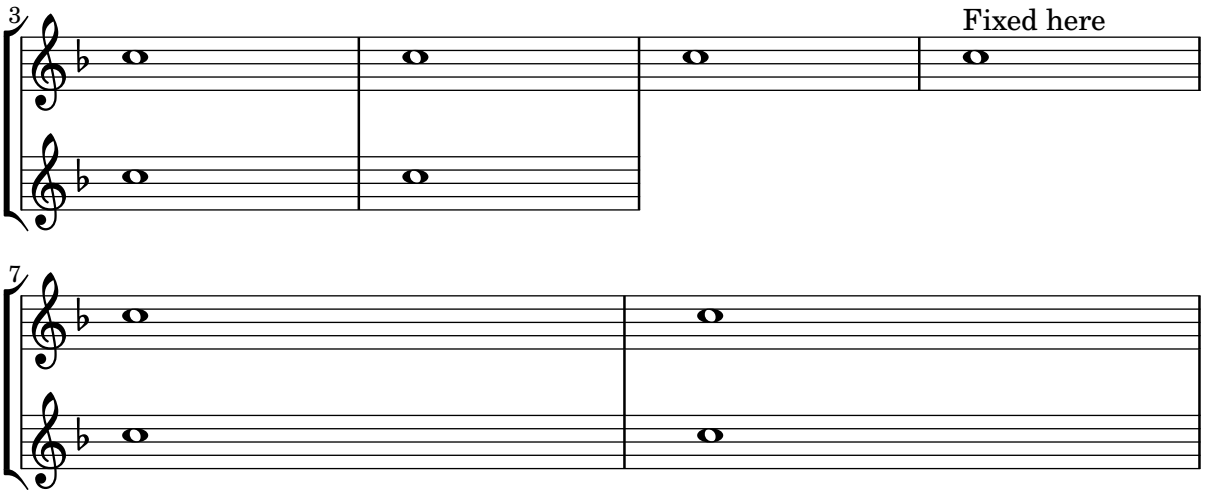


Añadir un pentagrama adicional en un salto de línea

Al añadir un pentagrama nuevo en un salto de línea, por desgracia se añade un espacio adicional al final de la línea antes del salto (reservado para hacer sitio a un cambio de armadura que de todas formas nunca se imprime). La solución alternativa es establecer la propiedad `Staff.explicitKeySignatureVisibility` del objeto gráfico `Staff` como se muestra en el ejemplo.

```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
       \new Staff {
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
    c1 | c^"Fixed here" \break
    << { c1 | c }
       \new Staff {
         \once \set Staff.explicitKeySignatureVisibility =
           #end-of-line-invisible
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
  }
}
```



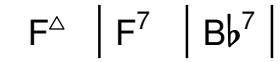


Añadir barras de compás al contexto de nombres de acorde (ChordNames)

Para añadir indicaciones de línea divisoria dentro del contexto de los nombres de acorde `ChordNames`, incluya el grabador `Bar_engraver`.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```
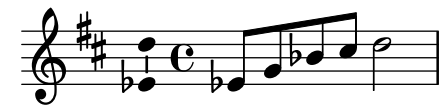


Indicación de tesitura después de la armadura

De manera predeterminada, las indicaciones de ámbito o tesitura se sitúan a la izquierda de la clave. La función `\ambitusAfter` permite cambiar esta colocación. La sintaxis es `\ambitusAfter` *grob-interface* (consulte Interfaces de los objetos gráficos (`https://lilypond.org/doc/v2.24/Documentation/internals/graphical-object-interfaces`) para ver una lista de los valores posibles para *grob-interface*.)

Un caso de utilización bastante común es imprimir la indicación de texitura entre la armadura de la clave y la indicación de compás.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



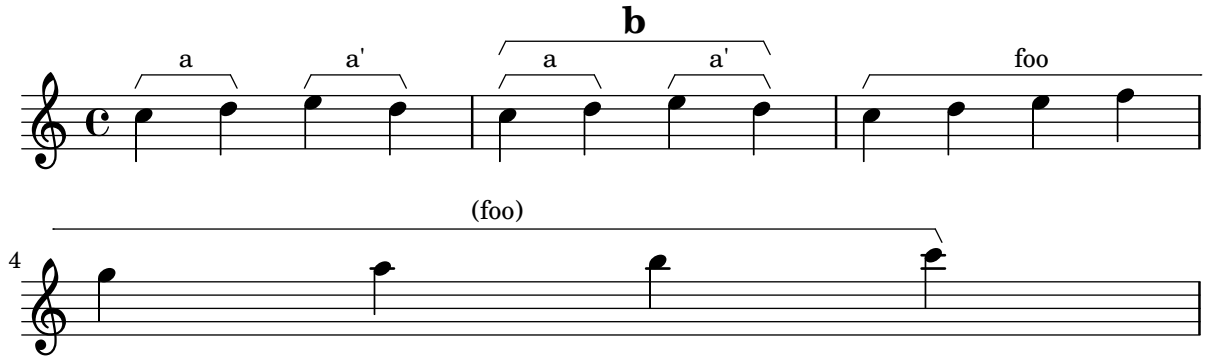
Corchetes de análisis con etiquetas

Se pueden añadir indicaciones de marcado de texto a los corchetes de análisis por medio de la propiedad `text` del objeto gráfico `HorizontalBracketText`. Si se quieren añadir textos diferentes al corchetes que comienzan en el mismo tiempo, es necesario usar la instrucción `\tweak`.

El texto del corchete se encierra entre paréntesis después de un salto de línea. El orden vertical de los corchetes anidados se puede controlar con la propiedad `outside-staff-priority`.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
    c''\startGroup d''\stopGroup
    \once\override HorizontalBracketText.text = "a'"
    e''\startGroup d''\stopGroup |
  c''-\tweak outside-staff-priority #801
      \tweak HorizontalBracketText.text
        \markup \bold \huge "b" \startGroup
     -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```
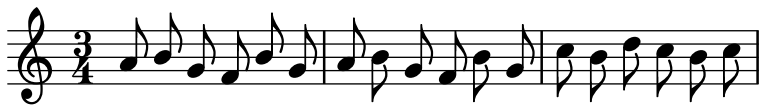


Cambiar la dirección de la plica de las notas de la tercera línea automáticamente, basado en la melodía

LilyPond puede alterar la dirección de la plica de las notas que van en la tercera línea de un pentagrama de forma que siga la melodía, mediante la adición del grabador `Melody_engraver` al contexto Voice.

Se puede usar la propiedad de contexto `suspendMelodyDecisions` para desactivar este comportamiento localmente.

```
\relative c'' {
  \time 3/4
  a8 b g f b g |
  \set suspendMelodyDecisions = ##t
  a  b g f b g |
  \unset suspendMelodyDecisions
  c  b d c b c |
}

\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \autoBeamOff
  }
}
```
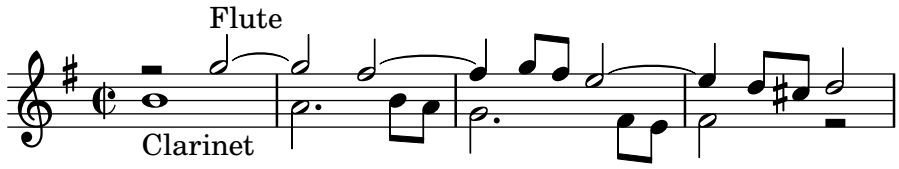


Modificar la salida MIDI para que tenga un canal por cada voz

Al producir una salida MIDI, el comportamiento predeterminado es que cada pentagrama representa un canal MIDI, con todas las voces de dicho pentagrama mezcladas. Esto reduce al mínimo el riesgo de que se agote el número de canales MIDI disponibles, pues existe un máximo de 16 canales por cada puerto MIDI, y la mayoría de los dispositivos solo tiene un puerto.

Sin embargo, cuando se traslada el interpretador `Staff_performer` al contexto `Voice`, cada voz de un pentagrama puede tener su propio canal MIDI, como se muestra en el siguiente ejemplo: a pesar de estar sobre el mismo pentagrama, se crean dos canales MIDI, cada uno con un `midiInstrument` distinto.

```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = "flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = "clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
```

```
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}
```



Modificar el compás dentro de una sección polimétrica utilizando `\scaleDurations`

Polirritmia flexible con compases no alineados

Para poder crear explícitamente contextos con compases independientes, suprima el traductor `Timing_translator` del contexto `Score` y defina un contexto `TimingStaffGroup` que contenga el traductor `Timing_translator`. Esto hace que `Timing` sea un alias de `TimingStaffGroup`, con la mira puesta en las instrucciones `\time` hacia el `TimingStaffGroup` contenedor.

A diferencia de la instrucción incorporada `\enablePerStaffTiming` de LilyPond, este enfoque requiere la creación explícita de contextos `TimingStaffGroup`; a cambio, permite crear varios contextos `Staff` que de forma conjunta siguen al compás definido en su `TimingStaffGroup` contenedor.

Indicaciones de compás escalados localmente

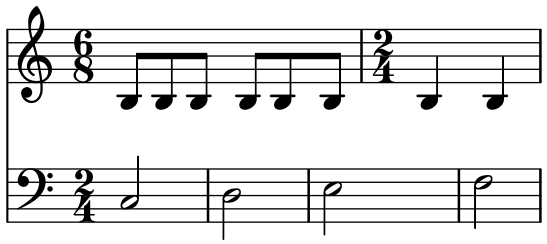
Use la instrucción no escalable `\time` para establecer un compás de la longitud deseada en `Timing`, más conocido como `TimingStaffGroup`. En este fragmento de código, todos los pentagramas que están por debajo de `TimingStaffGroup` usan una indicación de compás escalada, de forma que cualquier compás que establezcamos con la longitud deseada es tan bueno como cualquier otro. Si hubiese un contexto contenedor que no usara una indicación de compás escalada, la elección del compás que se establezca en `Timing` tendría importancia en dicho contexto.

Use la instrucción `\polymetric \time` para establecer propiedades métricas escalablesen los contextos que están por debajo de `Timing`, y use la instrucción `\scaleDurations` para escalar tanto el compás local como las notas que han de llenar el compás.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \accepts TimingStaffGroup
  }
  \context {
    \StaffGroup
```

```
    \name TimingStaffGroup
    \alias StaffGroup
    \consists "Timing_translator"
  }
}
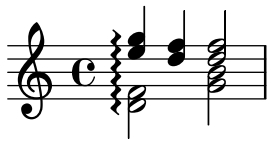
<<
  \new TimingStaffGroup <<
    \new Staff {
      \scaleDurations 8/5 {
        \time 6/5 % to set measure length in Timing
        \context Staff \polymetric \time 6/8
        b8 b b b b b
        \time 4/5 % to set measure length in Timing
        \context Staff \polymetric \time 2/4
        b4 b
      }
    }
  >>
  \new TimingStaffGroup <<
    \new Staff {
      \clef bass
      \time 2/4
      c2 d e f
    }
  >>
>>
```



Crear arpegios entre notas de voces distintas

Se puede trazar un símbolo de *arpeggio* entre notas de distintas voces que están sobre el mismo pentagrama si el grabador Span_arpeggio_engraver se traslada al contexto de Staff.

```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Crear pentagramas en blanco

Para crear pentagramas en blanco, genere compases vacíos y después elimine el grabador de números de compás Bar_number_engraver del contexto Score, y los grabadores de la indicación de compás Time_signature_engraver, de la clave Clef_engraver y de los compases Bar_engraver del contexto de Staff.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

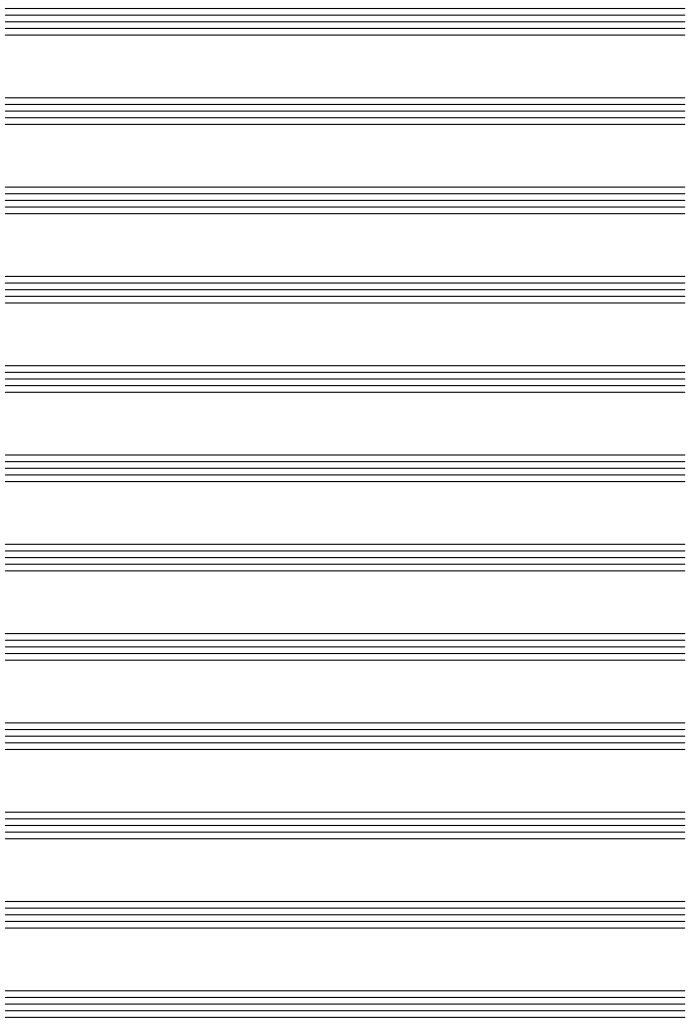
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % for the documentation
  \paper {
    #(set-paper-size "a6")
    ragged-last-bottom = ##f
    line-width = 90\mm
    left-margin = 7.5\mm
    bottom-margin = 5\mm
    top-margin = 5\mm
    tagline = ##f
  }

  % uncomment these lines for "letter" size
  %{
  \paper {
    #(set-paper-size "letter")
    ragged-last-bottom = ##f
    line-width = 7.5\in
    left-margin = 0.5\in
    bottom-margin = 0.25\in
```

```
    top-margin = 0.25\in
    tagline = ##f
   }
  %}

  % uncomment these lines for "A4" size
  %{
  \paper {
    #(set-paper-size "a4")
    ragged-last-bottom = ##f
    line-width = 180\mm
    left-margin = 15\mm
    bottom-margin = 10\mm
    top-margin = 10\mm
    tagline = ##f
  }
  %}
}
```



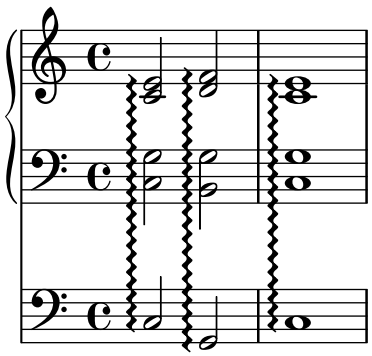
Creación de arpegios que se cruzan entre pentagramas dentro de otros contextos

Se pueden crear arpegios que se cruzan entre pentagramas dentro de contextos distintos de `GrandStaff` y sus parientes derivados `PianoStaff`, `ChoirStaff` y `StaffGroup` si se incluye el grabador `Span_arpeggio_engraver` en el contexto de `Score`.

```
<<
  \new PianoStaff <<
    \new Voice \relative c' {
      <c e>2\arpeggio <d f>2\arpeggio
      <c e>1\arpeggio
    }
    \new Voice \relative c {
      \clef bass
      <c g'>2\arpeggio <b g'>2\arpeggio
      <c g'>1\arpeggio
    }
  >>

  \new Staff \relative c {
    \set Score.connectArpeggios = ##t
    \clef bass
    c2\arpeggio g\arpeggio
    c1\arpeggio
  }
>>

\layout {
  \context {
    \Score
    \consists "Span_arpeggio_engraver"
  }
}
```
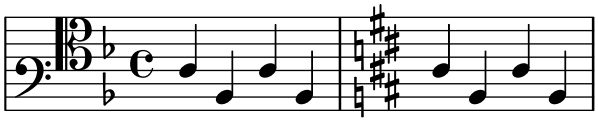


Creación de armaduras de clave personalizadas

LilyPond contempla armaduras de tonalidad personalizadas. En este ejemplo se muestra una armadura de Re menor y Re mayor con un rango ampliado de bemoles mostrados.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil =
    #(lambda (grob)
        (grob-interpret-markup grob
          #{ \markup\combine
               \musicglyph "clefs.C"
               \translate #'(-3 . -2)
                 \musicglyph "clefs.F"
          #}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}

{
  \key d\minor f bes, f bes, |
  \key d\major fis b, fis b, |
}
```



Plicas de pentagrama cruzado

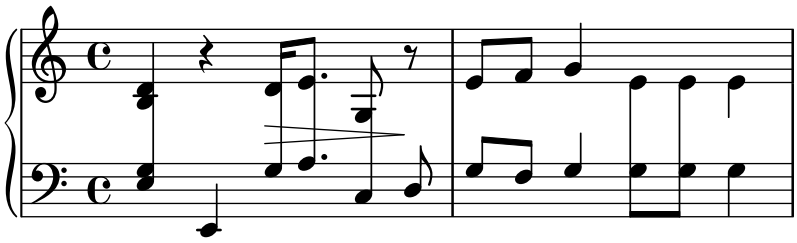
Este fragmento de código muestra el uso del grabador Span_stem_engraver y de \crossStaff para conectar automáticamente plicas de un pentagrama a otro.

No es necesario especificar las longitudes de las plicas porque la distancia variable entre las cabezas de las notas y los pentagramas se calcula automáticamente. Sin embargo, es importante aplicar \crossStaff a la voz o pentagrama correctos (esto es, en el lado opuesto de donde está o estaría posicionada la barra de corcheas) para conseguir el efecto deseado

```
\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

\new PianoStaff <<
  \new Staff {
    <b d'>4 r d'16\> e'8. g8 r\! |
    e'8 f' g'4
      \voiceTwo
      % Down to lower staff
      \crossStaff { e'8 e'8 } e'4 |
  }
```

```
  \new Staff {
    \clef bass
    \voiceOne
    % Up to upper staff
    \crossStaff { <e g>4 e, g16 a8. c8 } d |
    g8 f g4 \voiceTwo g8 g g4 |
  }
>>
```



Definir un grabador en Scheme: grabador de ámbito o tesitura

Este ejemplo muestra cómo puede definirse el grabador de ámbito o tesitura en el espacio del usuario, con un grabador de Scheme. Esto es básicamente una reescritura en Scheme del código de lily/ambitus-engraver.cc.

```
#(use-modules (oop goops))

%%%
%%% Grob utilities
%%%
%%% These are literal rewrites of some C++ methods used by the ambitus
%%% engraver.

#(define (ly:separation-item::add-conditional-item grob grob-item)
   "Add GROB-ITEM to the array of conditional elements of GROB.

This is a rewrite of function `Separation_item::add_conditional_item` from
file `lily/separation-item.cc`."
   (ly:pointer-group-interface::add-grob
    grob 'conditional-elements grob-item))

#(define (ly:accidental-placement::accidental-pitch accidental-grob)
   "Get the pitch from the grob cause of ACCIDENTAL-GROB.

This is a rewrite of function `accidental_pitch` from file
`lily/accidental-placement.cc`."
   (ly:event-property (ly:grob-property
                       (ly:grob-parent accidental-grob Y) 'cause)
                      'pitch))

#(define (ly:accidental-placement::add-accidental grob accidental-grob)
   "Add ACCIDENTAL-GROB to the list of accidentals grobs of GROB.
ACCIDENTAL-GROB is an `Accidental` grob; GROB is an `AccidentalPlacement`
grob.
```

```
This is a rewrite of function `Accidental_placement::add_accidental` from
file `lily/accidental-placement.cc`."
   (let ((pitch (ly:accidental-placement::accidental-pitch
                 accidental-grob)))
     (set! (ly:grob-parent accidental-grob X) grob)
     (let* ((accidentals (ly:grob-object grob 'accidental-grobs))
            (handle (assq (ly:pitch-notename pitch) accidentals))
            (entry (if handle (cdr handle) '())))
       (set! (ly:grob-object grob 'accidental-grobs)
             (assq-set! accidentals
                        (ly:pitch-notename pitch)
                        (cons accidental-grob entry))))))

%%%
%%% Ambitus data structures.
%%%

%%% The <ambitus> class holds the various grobs that are created to
%%% print an ambitus:
%%%
%%% - `ambitus-group`: the grob that groups all the components of an
%%%   ambitus (`Ambitus` grob);
%%% - `ambitus-line`: the vertical line between the upper and lower
%%%   ambitus notes (`AmbitusLine` grob);
%%% - `ambitus-up-note` and `ambitus-down-note`: the note head and
%%%   accidental for the lower and upper note of the ambitus (see
%%%   `<ambitus-note>` class below).
%%%
%%% The other slots define the key and clef context of the engraver:
%%%
%%% - `start-c0`: position of middle c at the beginning of the piece.
%%%   It is used to place the ambitus notes according to their pitch;
%%% - `start-key-sig`: the key signature at the beginning of the
%%%   piece.  It is used to determine whether accidentals shall be
%%%   printed next to ambitus notes.

#(define-class <ambitus> ()
   (ambitus-group #:accessor ambitus-group)
   (ambitus-line #:accessor ambitus-line)
   (ambitus-up-note #:getter ambitus-up-note
                    #:init-form (make <ambitus-note>))
   (ambitus-down-note #:getter ambitus-down-note
                      #:init-form (make <ambitus-note>))
   (start-c0 #:accessor ambitus-start-c0
             #:init-value #f)
   (start-key-sig #:accessor ambitus-start-key-sig
                  #:init-value '()))

%%% Accessor for the lower and upper note data of an ambitus.
#(define-method (ambitus-note (ambitus <ambitus>) direction)
   "Return lower or upper note of AMBITUS depending on DIRECTION."
   (if (= direction UP)
```

```
       (ambitus-up-note ambitus)
       (ambitus-down-note ambitus)))

%%% The `<ambitus-note>` class holds the grobs that are specific to
%%% ambitus (lower and upper) notes:
%%%
%%% - `head`: an `AmbitusNoteHead` grob;
%%% - `accidental`: an `AmbitusAccidental` grob, to be possibly
%%%   printed next to the ambitus note head.
%%%
%%% Moreover,
%%%
%%% - `pitch` is the absolute pitch of the note;
%%% - `cause` is the note event that causes this ambitus note, i.e.,
%%%   the lower or upper note of the considered music sequence.

#(define-class <ambitus-note> ()
   (head #:accessor ambitus-note-head
         #:init-value #f)
   (accidental #:accessor ambitus-note-accidental
               #:init-value #f)
   (cause #:accessor ambitus-note-cause
          #:init-value #f)
   (pitch #:accessor ambitus-note-pitch
          #:init-value #f))

%%%
%%% Ambitus engraving logic.
%%%
%%% This is rewrite of the code from file `lily/ambitus-engraver.cc`.

#(define (make-ambitus translator)
   "Build an ambitus object: initialize all the grobs and their
relations.

The `Ambitus` grob contains all other grobs:

 Ambitus
  |- AmbitusLine
  |- AmbitusNoteHead    for upper note
  |- AmbitusAccidental  for upper note
  |- AmbitusNoteHead    for lower note
  |- AmbitusAccidental  for lower note

The parent of an accidental is the corresponding note head, and the
accidental is set as the `accidental-grob` property of the note head
so that is printed by the function that prints notes."
   ;; Make the ambitus object.
   (let ((ambitus (make <ambitus>)))
     ;; Build the `Ambitus` grob, which will contain all other grobs.
     (set! (ambitus-group ambitus)
           (ly:engraver-make-grob translator 'Ambitus '()))
```

```
     ;; Build the `AmbitusLine` grob (the line between lower and upper
     ;; note).
     (set! (ambitus-line ambitus)
           (ly:engraver-make-grob translator 'AmbitusLine '()))
     ;; Build the upper and lower `AmbitusNoteHead` and
     ;; `AmbitusAccidental`.
     (for-each
      (lambda (direction)
        (let ((head (ly:engraver-make-grob translator
                                           'AmbitusNoteHead '()))
              (accidental (ly:engraver-make-grob translator
                                                 'AmbitusAccidental '()))
              (group (ambitus-group ambitus)))
          ;; The parent of the `AmbitusAccidental` grob is the
          ;; `AmbitusNoteHead` grob.
          (set! (ly:grob-parent accidental Y) head)
          ;; The `AmbitusAccidental` grob is set as the
          ;; `accidental-grob` object of `AmbitusNoteHead`.  This is
          ;; later used by the function that prints notes.
          (set! (ly:grob-object head 'accidental-grob) accidental)
          ;; Both the note head and the accidental grobs are added to
          ;; the main ambitus grob.
          (ly:axis-group-interface::add-element group head)
          (ly:axis-group-interface::add-element group accidental)
          ;; The note head and the accidental grobs are added to the
          ;; ambitus object.
          (set! (ambitus-note-head (ambitus-note ambitus direction))
                head)
          (set! (ambitus-note-accidental (ambitus-note ambitus direction))
                accidental)))
       (list DOWN UP))

     ;; The parent of the ambitus line is the lower ambitus note head.
     (set! (ly:grob-parent (ambitus-line ambitus) X)
           (ambitus-note-head (ambitus-note ambitus DOWN)))
     ;; The ambitus line is added to the ambitus main grob.
     (ly:axis-group-interface::add-element (ambitus-group ambitus)
                                           (ambitus-line ambitus))
     ambitus))

#(define-method (initialize-ambitus-state
                  (ambitus <ambitus>) translator)
   "Initialize the state of AMBITUS by getting the starting position of
middle C and key signature from TRANSLATOR's context."
   (if (not (ambitus-start-c0 ambitus))
       (begin
         (set! (ambitus-start-c0 ambitus)
               (ly:context-property (ly:translator-context translator)
                                    'middleCPosition 0))
         (set! (ambitus-start-key-sig ambitus)
               (ly:context-property (ly:translator-context translator)
                                    'keyAlterations)))))
```

```
#(define-method (update-ambitus-notes (ambitus <ambitus>) note-grob)
   "Update upper and lower ambitus pitches of AMBITUS using NOTE-GROB."
   ;; Get the event that caused the `note-grob` creation and check
   ;; that it is a `note-event`.
   (let ((note-event (ly:grob-property note-grob 'cause)))
     (if (ly:in-event-class? note-event 'note-event)
         ;; Get the pitch from the note event.
         (let ((pitch (ly:event-property note-event 'pitch)))
           ;; If this pitch is lower than the current ambitus' lower
           ;; note pitch (or it has not been initialized yet), then
           ;; this pitch is the new ambitus' lower pitch.  The same is
           ;; done for the upper pitch (but in the opposite
           ;; direction).
           (for-each
            (lambda (direction pitch-compare)
              (if (or (not (ambitus-note-pitch
                            (ambitus-note ambitus direction)))
                      (pitch-compare
                       pitch (ambitus-note-pitch
                              (ambitus-note ambitus direction))))
                  (begin
                    (set! (ambitus-note-pitch
                           (ambitus-note ambitus direction))
                          pitch)
                    (set! (ambitus-note-cause
                           (ambitus-note ambitus direction))
                          note-event))))
            (list DOWN UP)
            (list ly:pitch<?
                  (lambda (p1 p2) (ly:pitch<? p2 p1)))))))

#(define-method (typeset-ambitus (ambitus <ambitus>) translator)
   "Typeset AMBITUS.

- Place the lower and upper ambitus notes according to their pitch and
  the position of the middle C.
- Typeset or delete the note accidentals, according to the key
  signature.  An accidental, if it is to be printed, is added to an
  `AccidentalPlacement` grob (a grob dedicated to the placement of
  accidentals near a chord).
- Both note heads are added to the ambitus line grob so that a line
  gets printed between them."
   ;; Check whether there are lower and upper pitches.
   (if (and (ambitus-note-pitch (ambitus-note ambitus UP))
            (ambitus-note-pitch (ambitus-note ambitus DOWN)))
       ;; Make an `AccidentalPlacement` grob, for placement of note
       ;; accidentals.
       (let ((accidental-placement
              (ly:engraver-make-grob
               translator
               'AccidentalPlacement (ambitus-note-accidental
```

```
                              (ambitus-note ambitus DOWN)))))
      ;; For lower and upper ambitus notes.
      (for-each
       (lambda (direction)
         (let ((pitch (ambitus-note-pitch
                       (ambitus-note ambitus direction))))
           ;; Set the cause and the staff position of the ambitus
           ;; note according to the associated pitch.
           (set! (ly:grob-property
                  (ambitus-note-head (ambitus-note ambitus direction))
                  'cause)
                 (ambitus-note-cause (ambitus-note ambitus direction)))
           (set! (ly:grob-property
                  (ambitus-note-head (ambitus-note ambitus direction))
                  'staff-position)
                 (+ (ambitus-start-c0 ambitus)
                    (ly:pitch-steps pitch)))
           ;; Determine whether an accidental shall be printed for
           ;; this note, according to the key signature.
           (let* ((handle
                   (or (assoc (cons (ly:pitch-octave pitch)
                                    (ly:pitch-notename pitch))
                              (ambitus-start-key-sig ambitus))
                       (assoc (ly:pitch-notename pitch)
                              (ambitus-start-key-sig ambitus))))
                  (sig-alter (if handle (cdr handle) 0)))
             (cond
              ((= (ly:pitch-alteration pitch) sig-alter)
               ;; The note alteration is in the key signature
               ;; => it does not have to be printed.
               (ly:grob-suicide! (ambitus-note-accidental
                                  (ambitus-note ambitus direction)))
               (set! (ly:grob-object (ambitus-note-head
                                      (ambitus-note ambitus direction))
                                     'accidental-grob)
                     '()))
              (else
               ;; Otherwise the accidental shall be printed.
               (set! (ly:grob-property
                      (ambitus-note-accidental
                       (ambitus-note ambitus direction)) 'alteration)
                     (ly:pitch-alteration pitch))))))
           ;; Add the `AccidentalPlacement` grob to the conditional
           ;; items of the `AmbitusNoteHead`.
           (ly:separation-item::add-conditional-item
            (ambitus-note-head (ambitus-note ambitus direction))
            accidental-placement)
           ;; Add the `AmbitusAccidental` to the list of the
           ;; `AccidentalPlacement` grob accidentals.
           (ly:accidental-placement::add-accidental
            accidental-placement
            (ambitus-note-accidental (ambitus-note ambitus direction)))
```

```
              ;; Add the `AmbitusNoteHead` grob to the `AmbitusLine` grob.
              (ly:pointer-group-interface::add-grob
               (ambitus-line ambitus)
               'note-heads
                (ambitus-note-head (ambitus-note ambitus direction)))))
           (list DOWN UP))
          ;; Add the `AccidentalPlacement` grob to the main `Ambitus` grob.
          (ly:axis-group-interface::add-element
           (ambitus-group ambitus) accidental-placement))
        ;; No lower and upper pitches => nothing to print.
        (begin
          (for-each
           (lambda (direction)
             (ly:grob-suicide! (ambitus-note-accidental
                                (ambitus-note ambitus direction)))
             (ly:grob-suicide! (ambitus-note-head
                                (ambitus-note ambitus direction))))
           (list DOWN UP))
          (ly:grob-suicide! ambitus-line))))

%%%
%%% Ambitus engraver definition.
%%%
#(define ambitus-engraver
   (lambda (context)
     (let ((ambitus #f))
       ;; When music is processed, make the ambitus object if not
       ;; already built.
       (make-engraver
        ((process-music translator)
         (if (not ambitus)
             (set! ambitus (make-ambitus translator))))

        ;; Set the ambitus clef and key signature state.
        ((stop-translation-timestep translator)
         (if ambitus
             (initialize-ambitus-state ambitus translator)))

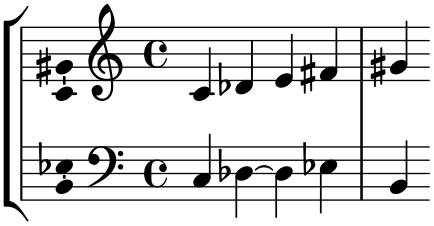
        ;; When a note head grob is built, update the ambitus notes.
        (acknowledgers
         ((note-head-interface engraver grob source-engraver)
          (if ambitus
              (update-ambitus-notes ambitus grob))))

        ;; Finally, typeset the ambitus according to its upper and
        ;; lower notes (if any).
        ((finalize translator)
         (if ambitus
             (typeset-ambitus ambitus translator)))))))

%%%
%%% Example
```

```
%%%

\score {
  \new StaffGroup <<
    \new Staff { c'4 des' e' fis' gis' }
    \new Staff { \clef "bass" c4 des ~ des ees b, }
  >>
  \layout { \context { \Staff \consists #ambitus-engraver } }
}
```



Mostrar un sistema GrandStaff completo si sigue con vida uno solo de sus pentagramas

En muchas partituras orquestales es costumbre no mostrar los pentagramas de aquellos instrumentos que están en silencio durante bastante tiempo; esto se llama 'partitura francesa'. LilyPond aporta esta funcionalidad a través de la instrucción `\RemoveEmptyStaves`.

Cuando vuelven a sonar, con frecuencia se prefiere mostrar *todos los instrumentos del grupo*. Esto se puede hacer añadiendo el grabador `Keep_alive_together_engraver` al contexto agrupador (p.ej.: un `GrandStaff` o un `StaffGroup`).

En este ejemplo, los violines están en silencio durante el segundo sistema. Solo el violín primero suena en el último compás del tercer sistema, pero se muestra también el pentagrama del violín segundo.

```
\score {
  <<
    \new Staff = "Staff_flute" \with {
      instrumentName = "Flute"
      shortInstrumentName = "Fl"
    } \relative c' {
      \*3 { c'4 c c c | c c c c | c c c c | \break }
    }

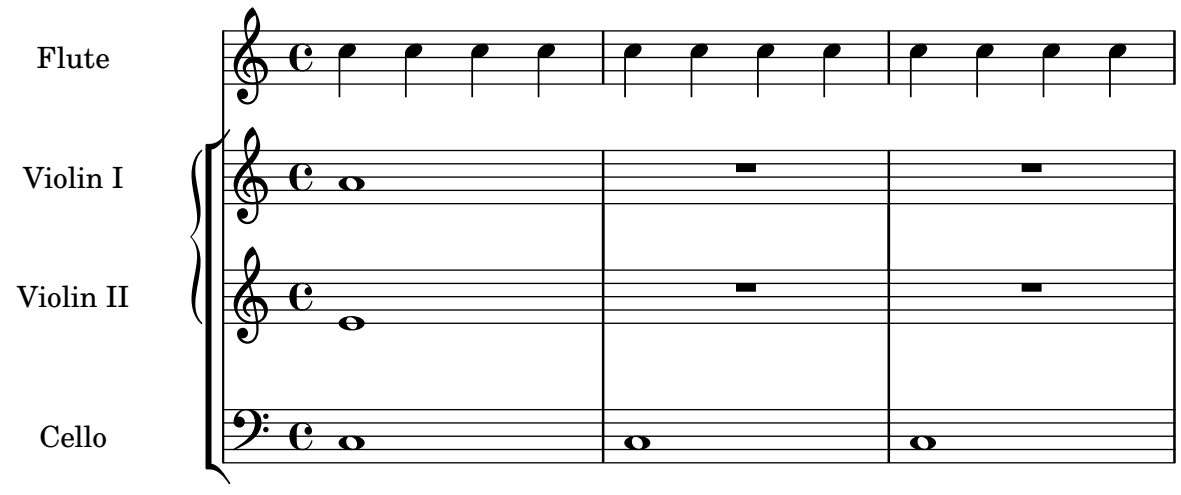
    \new StaffGroup = "StaffGroup_Strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
          shortInstrumentName = "Vi I"
        } \relative c'' {
          a1 | R1*7 | \*12 a16 a4 |
        }
        \new Staff = "StaffViolinII" \with {
          instrumentName = "Violin II"
          shortInstrumentName = "Vi II"
        } \relative c' {
          e1 | R1*8 |
```

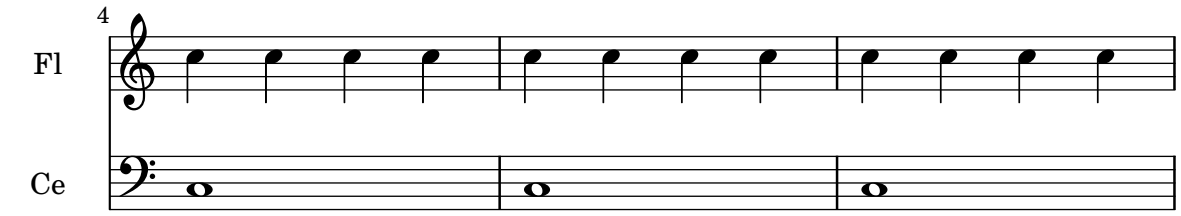
```
        }
      >>

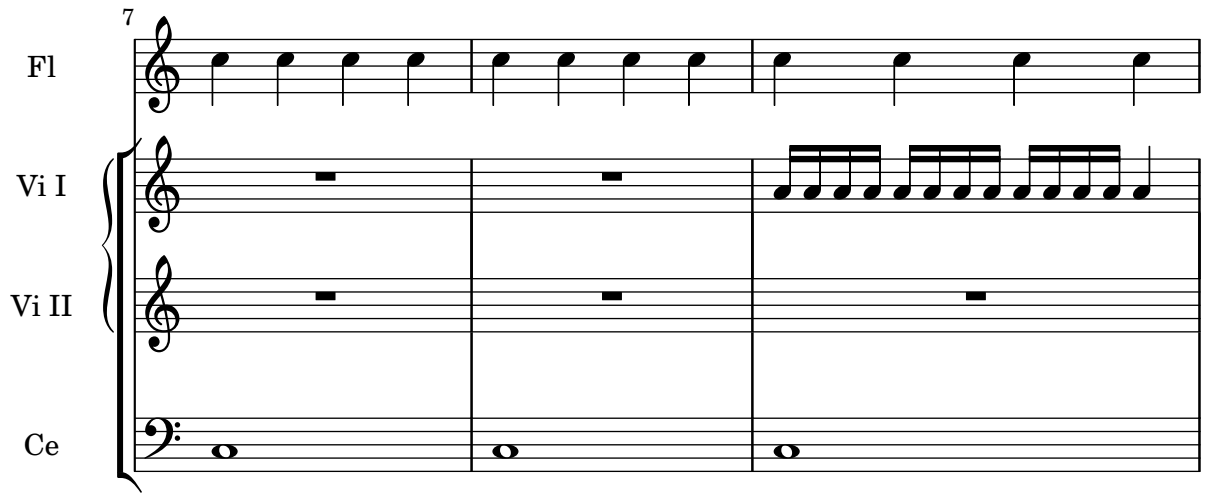
      \new Staff = "Staff_cello" \with {
        instrumentName = "Cello"
        shortInstrumentName = "Ce"
      } \relative c {
        \clef bass \*9 c1 |
      }
    >>
  >>
}

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm

  \context {
    \GrandStaff
    \consists Keep_alive_together_engraver
  }
  \context {
    \Staff
    \RemoveEmptyStaves
  }
}
```







Los grabadores, uno por uno

LilyPond trata los distintos elementos necesarios para el tipografiado de una partitura utilizando «plugins» o complementos añadidos. Cada uno de los complementos se conoce como un *grabador*. En este ejemplo, algunos grabadores se van activando uno por uno, en el orden siguiente:

- cabeza de las notas,
- el símbolo del pentagrama,
- clave,
- plicas,
- barras, ligaduras de expresión, acentos,
- alteraciones, líneas divisorias, indicación del compás, y armadura.

Los grabadores se encuentran agrupados. Por ejemplo, las cabezas de nota, ligaduras de expresión, barras de corchea, etc. forman un contexto de voz, `Voice`. Los grabadores de la armadura, alteraciones, líneas de compás, etc. forman un contexto de pentagrama, `Staff`.

```
topVoice = \relative c' {
  \key d \major
  es8([ g] a[ fis])
  b4
  b16[-. b-. b-. cis-.]
  d4->
}

% empty staff and voice contexts
MyStaff = \context {
  \type Engraver_group
  \name Staff
  \accepts Voice
  \defaultchild Voice
}
MyVoice = \context {
  \type Engraver_group
  \name Voice
}

% add note heads
```

```
MyVoice = \context {
  \MyVoice
  \consists Note_heads_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add staff
MyStaff = \context {
  \MyStaff
  \consists Staff_symbol_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}
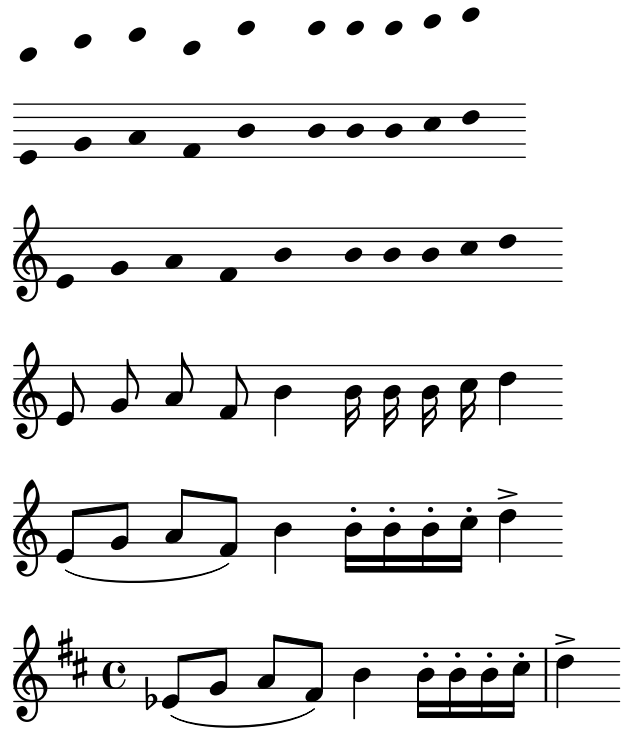
% add clef
MyStaff = \context {
  \MyStaff
  \consists Clef_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add stems
MyVoice = \context {
  \MyVoice
  \consists Stem_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add beams, slurs, and accents
```

```
MyVoice = \context {
  \MyVoice
  \consists Beam_engraver
  \consists Slur_engraver
  \consists Script_engraver
  \consists Rhythmic_column_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add accidentals, bar, time signature, and key signature
MyStaff = \context {
  \MyStaff
  \consists Accidental_engraver
  \consists Bar_engraver
  \consists Time_signature_engraver
  \consists Key_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}
```
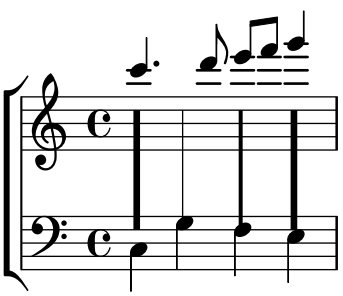


Líneas de rejilla: modificar su aspecto

Se puede cambiar el aspecto de las líneas de rejilla sobreescribiendo algunas de sus propiedades.

```
\new ChoirStaff <<
  \new Staff {
    \relative c'' {
      \stemUp
      c'4. d8 e8 f g4
    }
  }
  \new Staff {
    \relative c {
      % this moves them up one staff space from the default position
      \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
      \stemDown
      \clef bass
      \once \override Score.GridLine.thickness = 5.0
      c4
      \once \override Score.GridLine.thickness = 1.0
      g'4
      \once \override Score.GridLine.thickness = 3.0
      f4
      \once \override Score.GridLine.thickness = 5.0
      e4
    }
  }
>>

\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #1/4
    }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % this moves them to the right half a staff space
    \override NoteColumn.X-offset = -0.5
  }
}
```



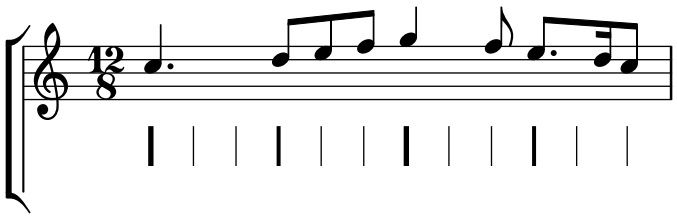
Líneas de rejilla: destacar ritmos y la relación temporal entre notas

Se pueden trazar líneas verticales normales entre pentagramas para mostrar la relación entre notas; sin embargo, en caso de música monofónica, podemos hacer invisible el segundo pentagrama, y que las líneas sean más cortas, como en este fragmento de código.

```
\new ChoirStaff {
  \relative c'' <<
    \new Staff {
      \time 12/8
      \stemUp
      c4. d8 e8 f g4 f8 e8. d16 c8
    }
    \new Staff {
      % hides staff and notes so that only the grid lines are visible
      \hideNotes
      \hide Staff.BarLine
      \override Staff.StaffSymbol.line-count = #0
      \hide Staff.TimeSignature
      \hide Staff.Clef

      % dummy notes to force regular note spacing
      \once  \override Score.GridLine.thickness = #4.0
      c8 c c
      \once  \override Score.GridLine.thickness = #3.0
      c8 c c
      \once  \override Score.GridLine.thickness = #4.0
      c8 c c
      \once  \override Score.GridLine.thickness = #3.0
      c8 c c
    }
  >>
}

\layout {
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % center grid lines horizontally below note heads
    \override NoteColumn.X-offset = #-0.5
  }
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #1/8
    % set line length and positioning:
    % two staff spaces above center line on hidden staff
    % to four spaces below center line on visible staff
    \override GridPoint.Y-extent = #'(2 . -4)
  }
}
```



Measure counters

This snippet demonstrates the use of the `Measure_counter_engraver` to number groups of successive measures. Any stretch of measures may be numbered, whether consisting of repetitions or not.

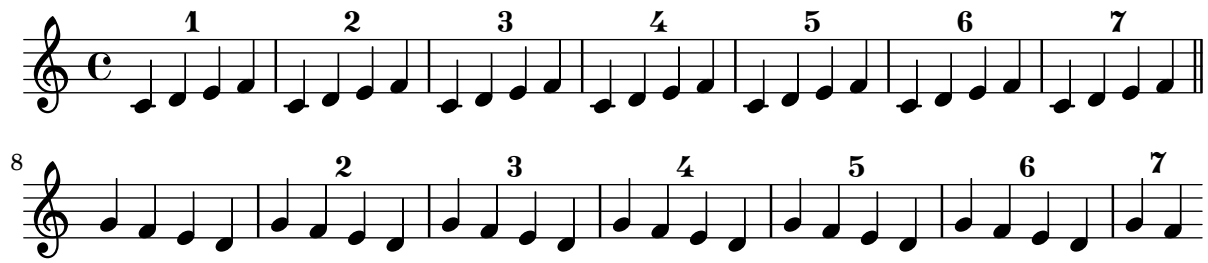
The engraver must be added to the appropriate context. Here, a `Staff` context is used; another possibility is a `Dynamics` context.
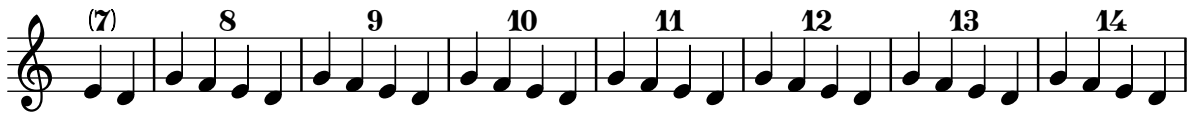
The counter is begun with `\startMeasureCount` and ended with `\stopMeasureCount`. Numbering will start by default with 1, but this behavior may be modified by overriding the `count-from` property.

When a measure extends across a line break, the number will appear twice, the second time in parentheses.

```
\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

\new Staff {
  \startMeasureCount
  \*7 { c'4 d' e' f' }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \*5 { g'4 f' e' d' }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \*7 { g'4 f' e' d' }
  \stopMeasureCount
}
```



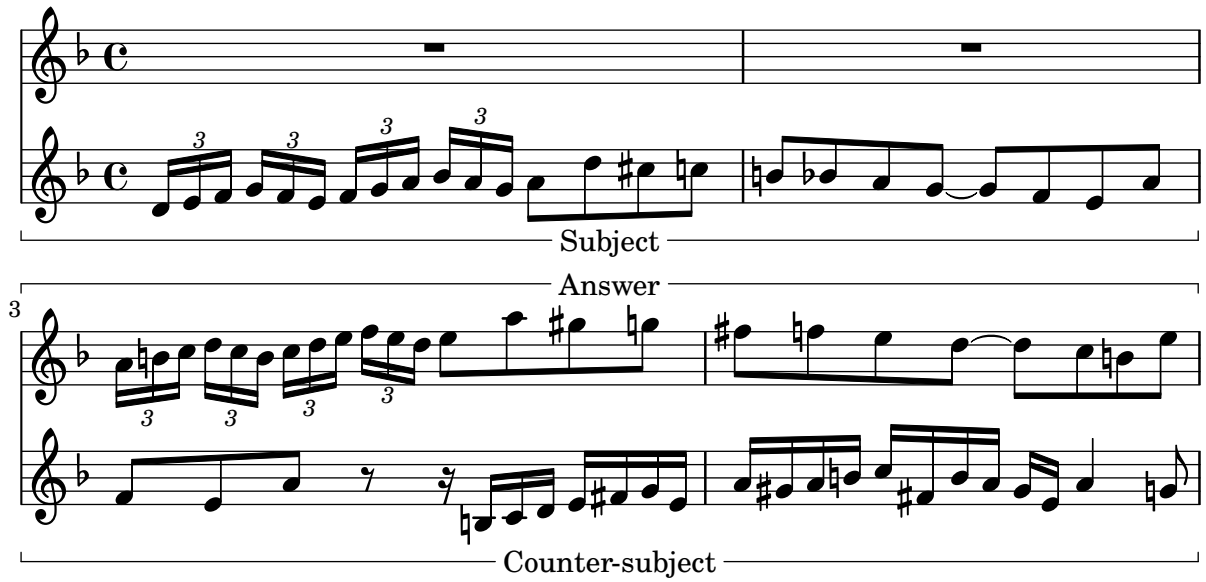


Corchete que abarca varios compases

Los elementos de extensión de compases son una forma alternativa de imprimir corchetes anotados. A diferencia de los corchetes horizontales, estos se extienden entre dos líneas divisorias en lugar de hacerlo entre dos notas. El texto se imprime en el centro del corchete.

```
\layout {
  \context {
    \Staff
    \consists Measure_spanner_engraver
  }
}

<<
  \new Staff \relative c'' {
    \key d \minor
    R1*2
    \tweak text "Answer"
    \startMeasureSpanner
    \tuplet 3/2 8 {
      a16[ b c] d[ c b]  c[ d e] f[ e d]
    }
    e8 a gis g
    fis f e d~ d c b e
    \stopMeasureSpanner
  }
  \new Staff \relative c' {
    \key d \minor
    \tweak text "Subject"
      \tweak direction #DOWN
      \startMeasureSpanner
    \tuplet 3/2 8 {
      d16[ e f] g[ f e] f[ g a] bes[ a g]
    }
    a8 d cis c
    b bes a g~ g f e a
    \stopMeasureSpanner
    \tweak text "Counter-subject"
      \tweak direction #DOWN
      \startMeasureSpanner
    f8 e a r r16 b, c d e fis g e
    a gis a b c fis, b a gis e a4 g8
    \stopMeasureSpanner
  }
>>
```



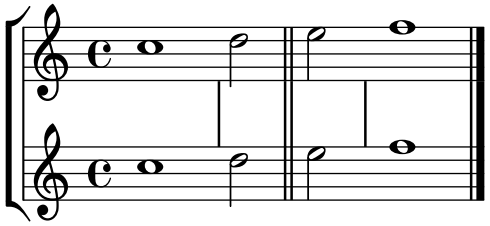
Disposición *Mensurstriche* (líneas divisorias entre pentagramas)

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir fijando `measureBarType` al valor `"-span|"` y usando un contexto de agrupamiento que admita barras extendidas, como `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



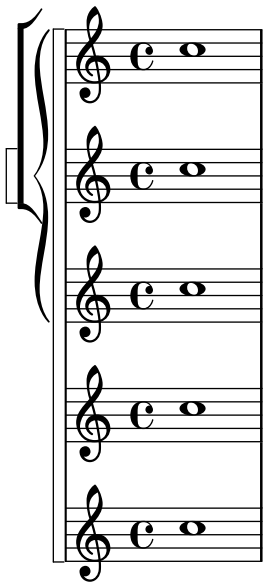
Anidado de grupos de pentagramas

Se puede utilizar la propiedad `systemStartDelimiterHierarchy` para crear grupos de pentagramas anidados de forma más compleja. La propiedad `systemStartDelimiterHierarchy` del

contexto StaffGroup toma una lista alfabética del número de pentagramas producidos. Se puede proporcionar antes de cada pentagrama un delimitador de comienzo de sistema. Se debe encerrar entre corchetes y admite tantos pentagramas como encierren las llaves. Se pueden omitir los elementos de la lista, pero el primer corchete siempre abarca todos los pentagramas. Las posibilidades son SystemStartBar, SystemStartBracket, SystemStartBrace y SystemStartSquare.

```
\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = 4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare
          (SystemStartBrace
            (SystemStartBracket a
              (SystemStartSquare b))
            c)
          d)

  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Permitir saltos de línea dentro de grupos especiales con barra

Estos ejemplos artificiales muestran cómo se pueden permitir tanto los saltos de línea manuales como los automáticos dentro de un grupo de valoración especial unido por una barra que no se puede dividir en una duración de manera exacta.

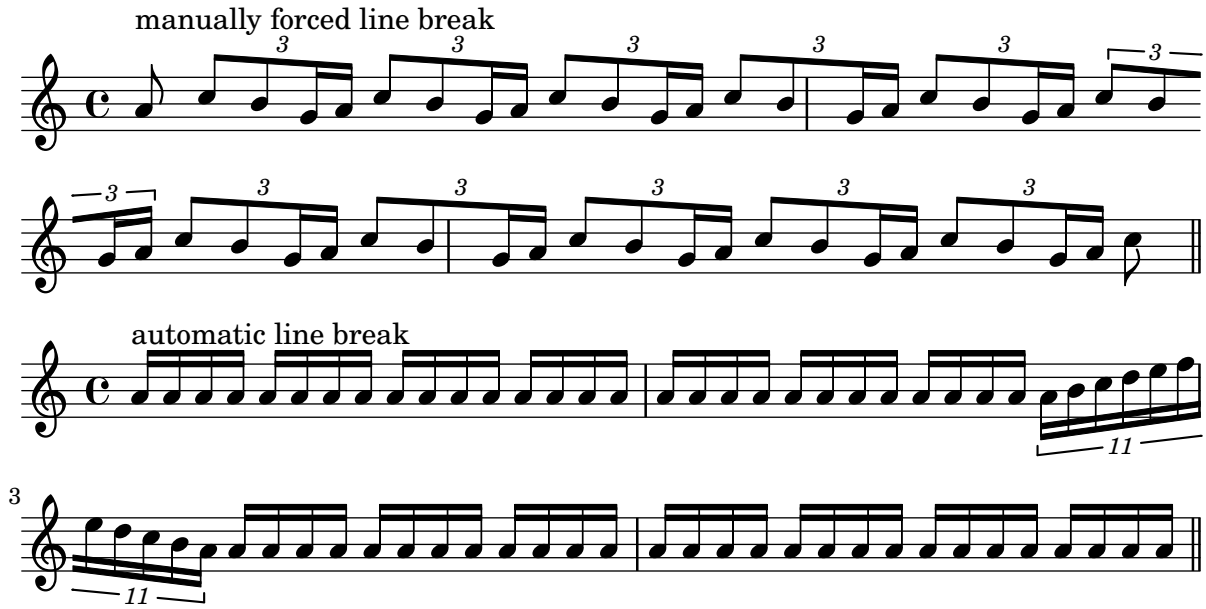
Esta funcionalidad solo funciona con grupos de valoración especial barrados manualmente.

```
\layout {
  \context {
    \Voice
    % Permit automatic line breaks within tuplets.
    \remove "Forbid_line_break_engraver"
```

```
    % Allow beams to be broken at line breaks.
    \override Beam.breakable = ##t
  }
}

\relative c'' {
  <>^"manually forced line break"
  a8
  \*5 { \tuplet 3/2 { c8[ b g16 a] } }
  \tuplet 3/2 { c8[ b \break g16 a] }
  \*5 { \tuplet 3/2 { c8[ b g16 a] } }
  c8 \bar "||"
}

\relative c'' {
  <>^"automatic line break"
  \*28 a16
  \tuplet 11/8 { a16[ b c d e f e d c b a] }
  \*28 a16 \bar "||"
}
```



Print chord names with same root and different bass as slash and bass note

To print subsequent chord names only differing in their bass note as slash and bass note without a root symbol, use the Scheme engraver defined in this snippet. The behaviour is controlled by the chordChanges context property.

```
#(define Bass_changes_equal_root_engraver
   (lambda (ctx)
     "Drop root for follow-up chord names that differ only in bass.

In other words, the chord name sequence 'D D/C D/B' gets actually
printed as 'D /C /B'.
```

```
Set the `chordChanges` context property to `#t` to activate this
feature."
     (let ((chord-pitches '())
           (last-chord-pitches '())
           (bass-pitch #f))
       (make-engraver
        ((initialize this-engraver)
         (let ((chord-note-namer (ly:context-property ctx
                                                      'chordNoteNamer)))
           ;; Set `chordNoteNamer`, respecting user setting if already
           ;; done.
           (ly:context-set-property! ctx 'chordNoteNamer
                                     (if (procedure? chord-note-namer)
                                         chord-note-namer
                                         (chord-name:markup #f)))))

        (listeners
         ((note-event this-engraver event)
          (let* ((pitch (ly:event-property event 'pitch))
                 (pitch-name (ly:pitch-notename pitch))
                 (pitch-alt (ly:pitch-alteration pitch))
                 (bass (ly:event-property event 'bass #f))
                 (inversion (ly:event-property event 'inversion #f)))
            ;; We look at the `bass` and `inversion` event properties
            ;; to decide how to handle the current note event.  If
            ;; `inversion` is set we add the bass note to the chord as
            ;; an ordinary member so that we can compare inversed
            ;; chords; if `bass` is set the bass note is not added to
            ;; the chord.
            ;;
            ;; In the `chord-pitches` list we actually collect only
            ;; the notes' pitch names (which are integers) and pitch
            ;; alterations as pairs, ignoring the octave.
            (cond (bass (set! bass-pitch pitch))
                  (inversion
                   (set! bass-pitch pitch)
                   (set! chord-pitches
                         (cons (cons pitch-name pitch-alt)
                               chord-pitches)))
                  (else
                   (set! chord-pitches
                         (cons (cons pitch-name pitch-alt)
                               chord-pitches)))))))

        (acknowledgers
         ((chord-name-interface this-engraver grob source-engraver)
          (let ((chord-changes (ly:context-property ctx
                                                    'chordChanges #f)))
            ;; If subsequent chords are equal apart from their bass,
            ;; we change the `text` property to print only the slash
            ;; and the bass note (via the formatter stored in the
            ;; `chordNoteNamer` context property).
```

```
            ;;
            ;; Equality is tested by comparing the sorted lists of
            ;; this chord's elements and the previous chord.  Sorting
            ;; is needed because inverted chords may have a different
            ;; order of pitches.  Note that we only do a simplified
            ;; sorting using the pitch name, ignoring the alteration.
            (if (and bass-pitch
                     chord-changes
                     (equal? (sort chord-pitches car<)
                             (sort last-chord-pitches car<)))
                (ly:grob-set-property!
                 grob 'text
                 (make-line-markup
                  (list
                   (ly:context-property ctx 'slashChordSeparator)
                   ((ly:context-property ctx 'chordNoteNamer)
                    bass-pitch
                    (ly:context-property ctx
                                         'chordNameLowercaseMinor))))))
            (set! last-chord-pitches chord-pitches)
            (set! chord-pitches '())
            (set! bass-pitch #f))))

        ((finalize this-engraver)
         (set! last-chord-pitches '()))))


myChords = \chordmode {
  % \germanChords

  \set chordChanges = ##t
  d2:m d:m/cis

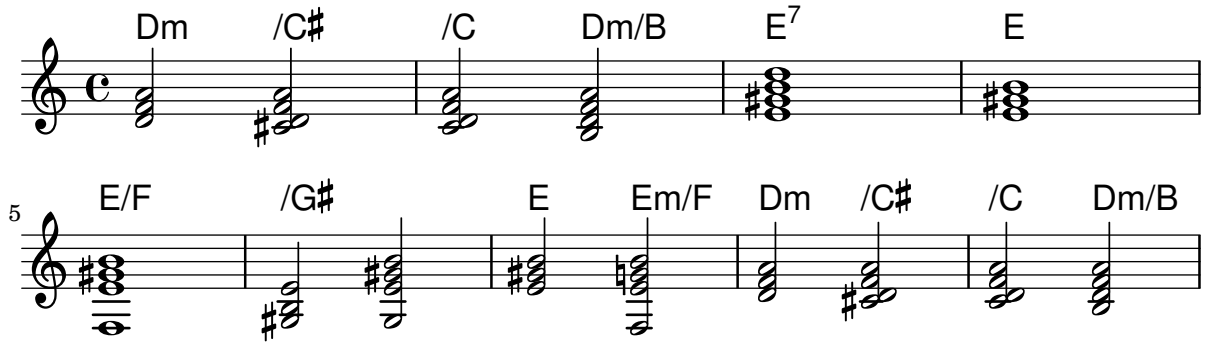
  d:m/c
  \set chordChanges = ##f
  d:m/b

  e1:7
  \set chordChanges = ##t
  e
  \break

  \once \set chordChanges = ##f
  e1/f
  e2/gis e/+gis e e:m/f d:m d:m/cis d:m/c
  \set chordChanges = ##f
  d:m/b
}

<<
  \new ChordNames
    \with { \consists #Bass_changes_equal_root_engraver }
```

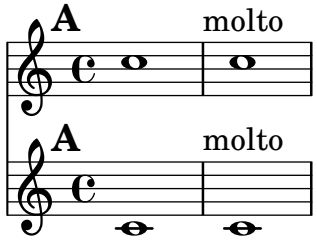
```
    \myChords
  \new Staff \myChords
>>
```



Imprimir marcas de ensayo en cualquier pentagrama

Aunque normalmente las marcas de ensayo y textuales solo se imprimen sobre el pentagrama superior, también se pueden imprimir en otro pentagrama cualquiera.

```
\score {
  <<
    \new Staff { \mark \default c''1 \textMark "molto" c'' }
    \new Staff { \mark \default c'1 \textMark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove Mark_engraver
      \remove Text_mark_engraver
      \remove Staff_collecting_engraver
    }
    \context {
      \Staff
      \consists Mark_engraver
      \consists Text_mark_engraver
      \consists Staff_collecting_engraver
    }
  }
}
```



Imprimir música que tenga compases distintos en cada pentagrama

En el siguiente fragmento de código, dos partes distintas tienen un compás completamente diferente y sin embargo se mantienen sincronizadas. Las barras de compás ya no se pueden

imprimir en el nivel de Score; para permitir barras de compás independientes en cada parte se trasladan los grabadores Default_barline_engraver y Timing_translator desde el contexto de partitura Score hasta el contexto de pentagrama Staff.

Si son necesarios números de compás, el grabador de números de compás Bar_number_engraver también debe trasladarse, ya que descansa en propiedades fijadas por el Timing_translator; se puede usar un bloque \with para añadir números de compás al pentagrama apropiado.

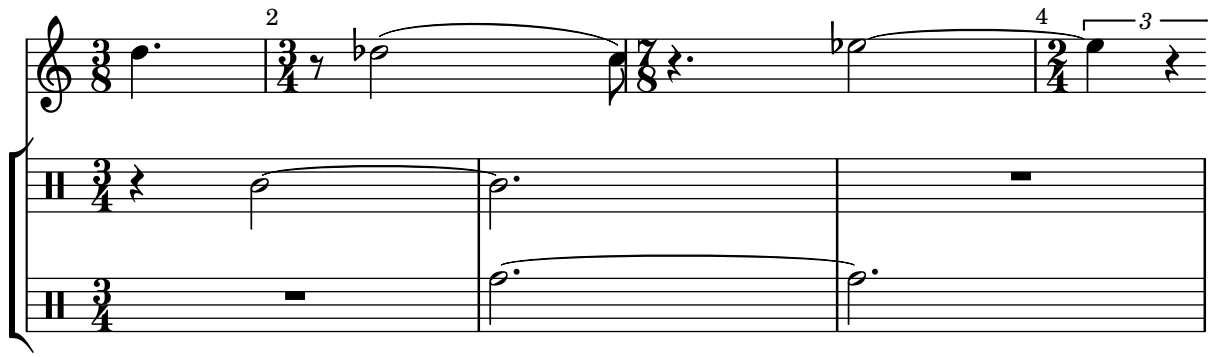
```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}

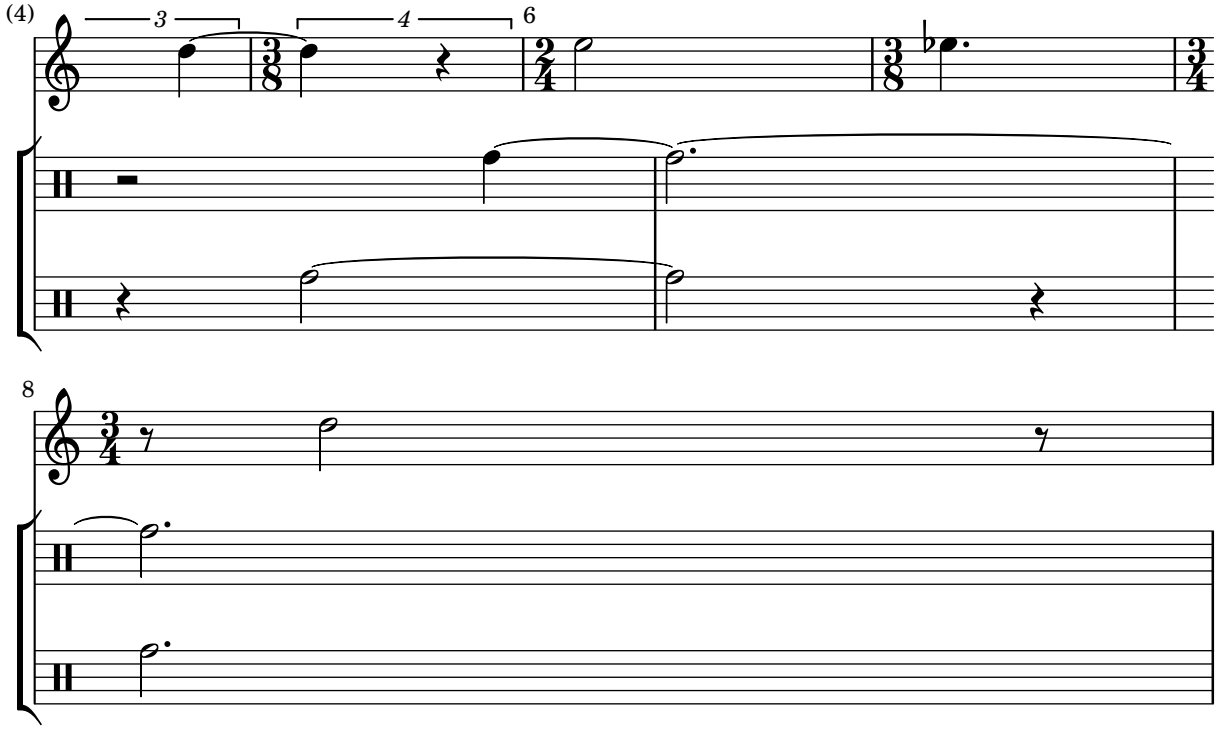
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. ees''2 ~ |
    \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
    \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
    \time 2/4 e''2 |
    \time 3/8 es''4. |
    \time 3/4 r8 d''2 r8 |
  }
>>
```

```
Perkussion = \new StaffGroup <<
  \new Staff <<
    \global
    {
      \clef percussion
      \time 3/4 r4 c'2 ~ |
      c'2. |
      R2. |
      r2 g'4 ~ |
      g'2. ~ |
      g'2. |
    }
  >>
  \new Staff <<
    \global {
      \clef percussion
      \time 3/4 R2. |
      g'2. ~ |
      g'2. |
      r4 g'2 ~ |
      g'2 r4 |
      g'2. |
    }
  >>
>>

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}
```



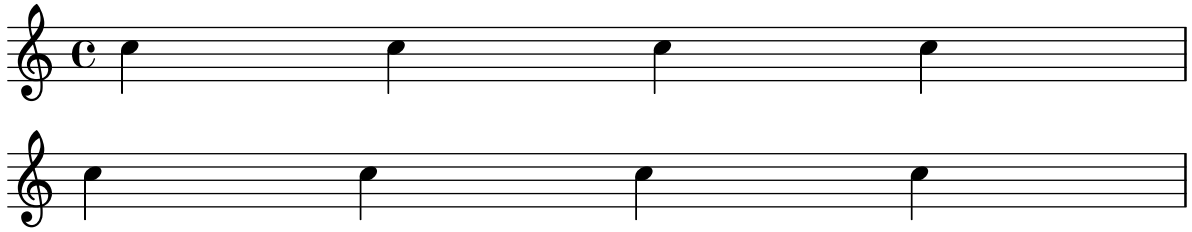


Suprimir los números de compás de toda la partitura

Se pueden eliminar completamente los números de compás quitando el grabador `Bar_number_engraver` del contexto de `Score`.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    % \remove "Bar_number_engraver"
  }
}

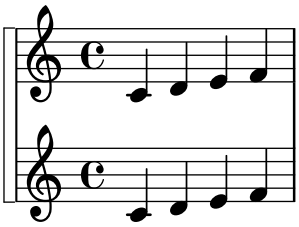
\relative c'' {
  c4 c c c \break
  c4 c c c
}
```



Uso del corchete recto al comienzo de un grupo de pentagramas

Se puede usar el delimitador de comienzo de un sistema `SystemStartSquare` estableciéndolo explícitamente dentro de un contexto `StaffGroup` o `ChoirStaffGroup`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



Using mark lines in a Frenched score

Using MarkLine contexts (such as in "Placing rehearsal marks other than above the top staff") in a Frenched score can be problematic if all the staves between two MarkLines are removed in one system. The Keep_alive_together_engraver can be used within each StaffGroup to keep the MarkLine alive only as long as the other staves in the group stay alive.
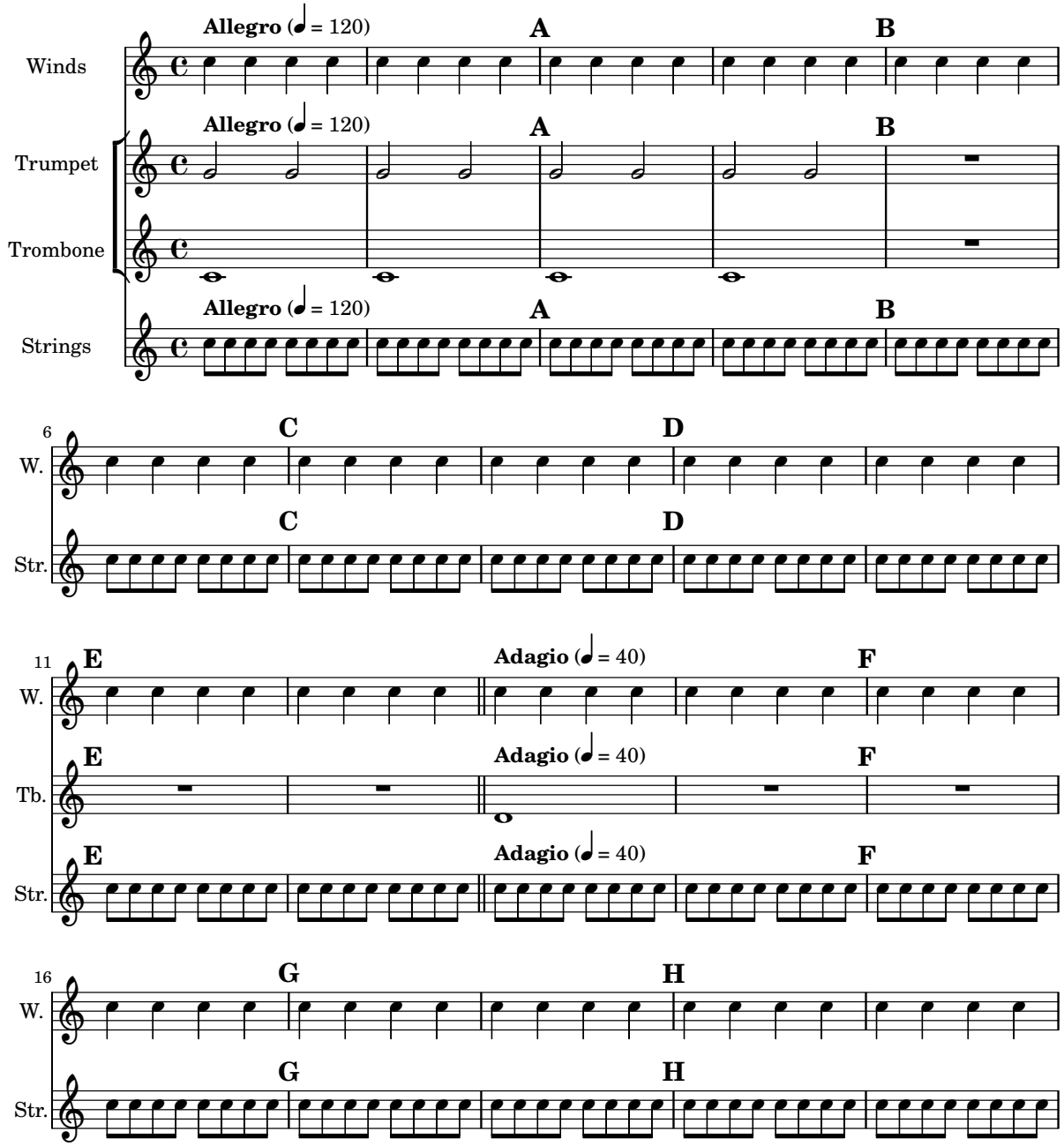
```
bars = {
  \tempo Allegro 4=120
  s1*2
  \*5 { \mark \default s1*2 }
  \bar "||"
  \tempo Adagio 4=40
  s1*2
  \*8 { \mark \default s1*2 }
  \bar "|."
}
winds = \*120 c''4
trumpet = {
  \*8 g'2
  R1*16
  \*4 g'2
  R1*8
}
trombone = {
  \*4 c'1
  R1*8
  d'1
  R1*17
}
strings = \*240 c''8

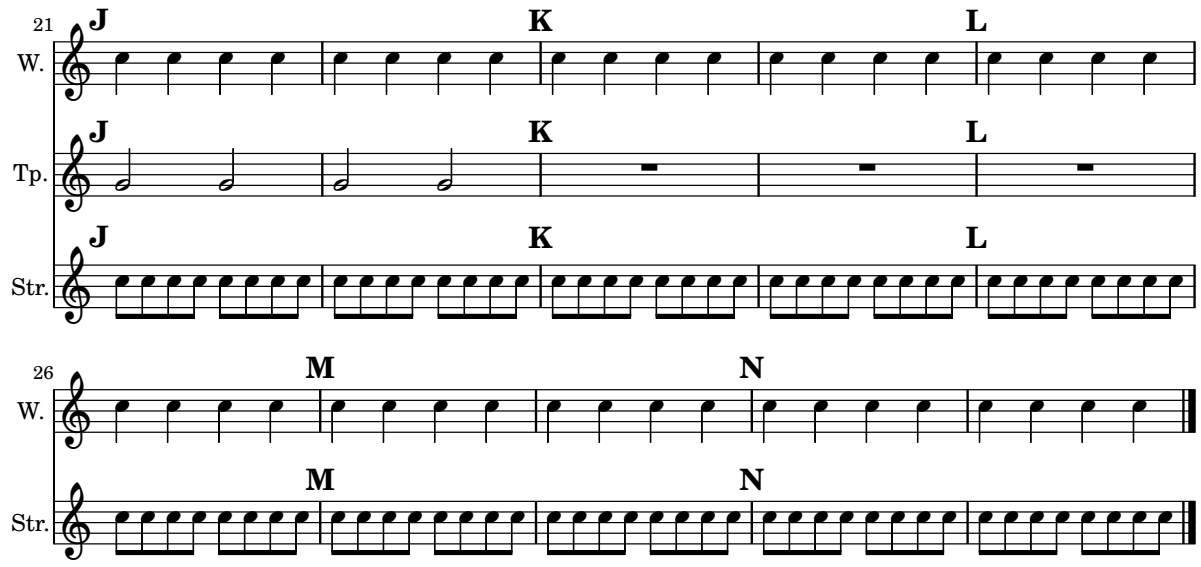
#(set-global-staff-size 16)
\paper {
  systems-per-page = 5
  ragged-last-bottom = ##f
  tagline = ##f
}
```

```
\layout {
  indent = 16\mm
  short-indent = 5\mm
  \context {
    \name MarkLine
    \type Engraver_group
    \consists Apply_output_engraver
    \consists Axis_group_engraver
    \consists Mark_engraver
    \consists Metronome_mark_engraver
    \consists Staff_collecting_engraver
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = #'any
    \override VerticalAxisGroup.staff-affinity = #DOWN
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing.padding = 1
    keepAliveInterfaces = #'()
  }
  \context {
    \Staff
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = ##f
  }
  \context {
    \StaffGroup
    \accepts MarkLine
    \consists Keep_alive_together_engraver
  }
  \context {
    \Score
    \remove Mark_engraver
    \remove Metronome_mark_engraver
    \remove Staff_collecting_engraver
    \override BarNumber.Y-offset = #3
  }
}

\score {
  <<
    \new StaffGroup = "winds" \with {
      instrumentName = "Winds"
      shortInstrumentName = "W."
    } <<
      \new MarkLine \bars
      \new Staff \winds
    >>
    \new StaffGroup = "brass" <<
      \new MarkLine \bars
      \new Staff = "trumpet" \with {
        instrumentName = "Trumpet"
        shortInstrumentName = "Tp."
      } \trumpet
```

```
      \new Staff = "trombone" \with {
        instrumentName = "Trombone"
        shortInstrumentName = "Tb."
      } \trombone
    >>
    \new StaffGroup = "strings" \with {
      instrumentName = "Strings"
      shortInstrumentName = "Str."
    } <<
      \new MarkLine \bars
      \new Staff = "strings" { \strings }
    >>
  >>
}
```





Uso de etiquetas para producir música mensural y moderna a partir del mismo código fuente

Usando etiquetas es posible producir las dos notaciones mensural y moderna a partir del mismo código fuente de la música. En este fragmento se introduce la función `\menrest`, que permite que los silencios mensurales estén a la altura de notas como en el original, pero con silencios modetnos en la posicion del pentagrama estándar.

También se pueden usar etiquetas allí donde se necesiten otras diferencias: por ejemplo usando "silencios de compás completo" (`R1`, `R\breve`, etc.) en música moderna, pero silencios normales (`r1`, `r\breve`, etc.) en la versión mensural. La conversión de música mensural a su equivalente moderno se conoce normalmente como *transcription*.

La llamara `c4.\Be c8 c\Am` es lo mismo que `c4.[ c8 c]`. Sin embargo, suprime las advertencias sin se inicia en una nota que no puede llevar barra pero la necesita en todo caso debido al uso del grabador `Completion_heads_engraver`.

[La longitud ligeramente acortada de la línea en la pauta mensural evita el recorte del glifo de los custos cuando LilyPond genera imágenes recortadas muy estrechamente. El uso de `\with-true-dimensions`, tal y como aparece a continuación, evita esto.]

```
\layout {
  line-width = 150\mm
}

menrest = #(define-music-function (note) (ly:music?)
              #{
                \tag #'mens $(make-music 'RestEvent note)
                \tag #'mod $(make-music 'RestEvent note 'pitch '())
              #})
Be = \tag #'mod
       #(begin
          (ly:expect-warning (G_ "stem does not fit in beam"))
          (ly:expect-warning (G_ "beam was started here"))
          (make-span-event 'BeamEvent START))
Am = \tag #'mod ]

MenStyle = {
```

```
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
  \omit Slur
  \omit Beam
}

finalis = \section

Music = \relative c'' {
  \key f \major
  g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
  g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}

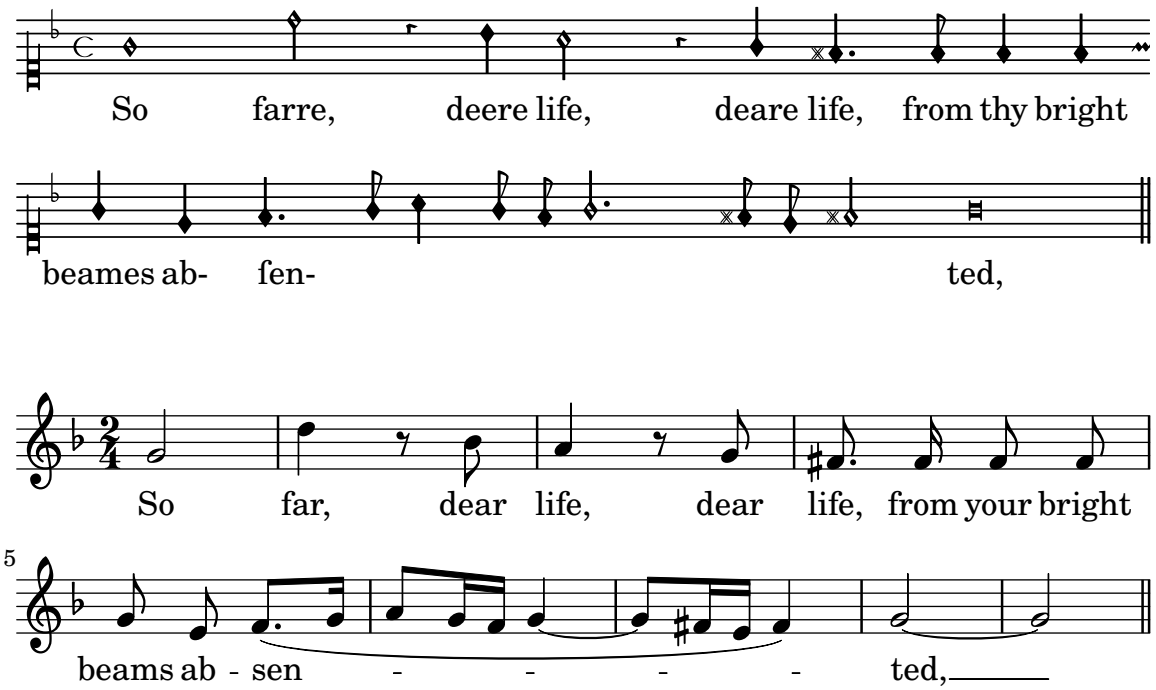
MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}

\score {
  \keepWithTag #'mens {
    <<
      \new PetrucciStaff {
        \new PetrucciVoice = "Cantus" {
          \clef "petrucci-c1" \time 4/4 \MenStyle \Music
        }
      }
      \new Lyrics \lyricsto "Cantus" \MenLyr
    >>
  }
  \layout {
    \context {
      \PetrucciVoice
      % No longer necessary starting with version 2.25.23.
      \override Flag.style = #'mensural
    }
  }
}

\markup\vspace #1

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff {
        \new Voice = "Sop" \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
```

```
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver"
        } \shiftDurations 1 0 { \time 2/4 \autoBeamOff \Music }
      }
      \new Lyrics \lyricsto "Sop" \ModLyr
    >>
  }
}
```



Estrofa para solista y estribillo a dos voces

Esta plantilla crea una partitura que comienza con una estrofa para solista y continúa con un estribillo a dos voces. también muestra el uso de silencios de separación dentro de la variable \global para definir cambios de compás (y otros elementos que son comunes a todas las partes) a lo largo de toda la partitura.

```
global = {
  \key g \major

  % verse
  \time 3/4
  s2.*2
  \break

  % refrain
  \time 2/4
  s2*2
  \bar "|."
}

SoloNotes = \relative g' {
  \clef "treble"
```

```
  % verse
  g4 g g |
  b4 b b |

  % refrain
  R2*2 |
}

SoloLyrics = \lyricmode {
  One two three |
  four five six |
}

SopranoNotes = \relative c'' {
  \clef "treble"

  % verse
  R2.*2 |

  % refrain
  c4 c |
  g4 g |
}

SopranoLyrics = \lyricmode {
  la la |
  la la |
}

BassNotes = \relative c {
  \clef "bass"

  % verse
  R2.*2 |

  % refrain
  c4 e |
  d4 d |
}

BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

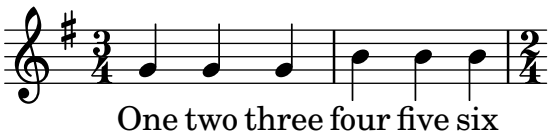
\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics
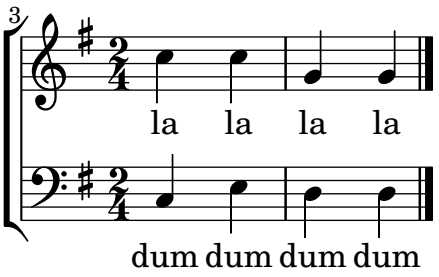
    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
```

```
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

      \new Voice = "BassVoice" << \global \BassNotes >>
      \new Lyrics \lyricsto "BassVoice" \BassLyrics
    >>
  >>

  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}
```

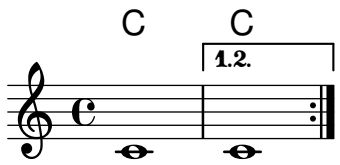




Corchetes de primera y segunda vez debajo de los acordes

Mediante la adición del grabador `Volta_engraver` al pentagrama pertinente, se pueden poner los corchetes de primera y segunda vez debajo de los acordes.

```
\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



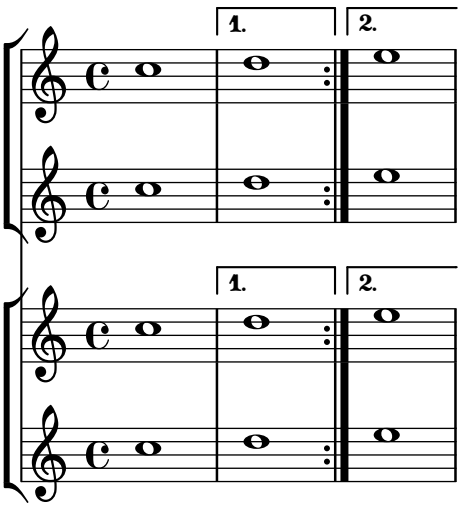
Casillas de primera y segunda vez en varios pentagramas

Añadiendo el grabador `Volta_engraver` al pentagrama deseado, las casillas de primera y segunda vez se pueden poner sobre pautas distintas a la del extremo superior de la partitura.

Las instrucciones `\repeat` y otras relacionadas deben estar presentes en todos los pentagramas.

```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}

<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```



23 Education

Líneas de rejilla: destacar ritmos y la relación temporal entre notas

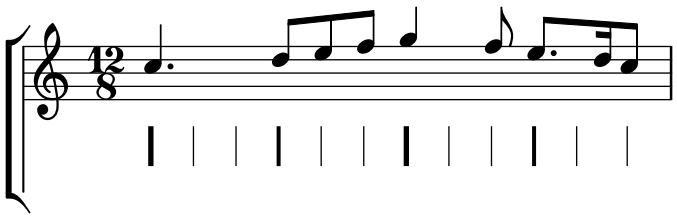
Se pueden trazar líneas verticales normales entre pentagramas para mostrar la relación entre notas; sin embargo, en caso de música monofónica, podemos hacer invisible el segundo pentagrama, y que las líneas sean más cortas, como en este fragmento de código.

```
\new ChoirStaff {
  \relative c'' <<
    \new Staff {
      \time 12/8
      \stemUp
      c4. d8 e8 f g4 f8 e8. d16 c8
    }
    \new Staff {
      % hides staff and notes so that only the grid lines are visible
      \hideNotes
      \hide Staff.BarLine
      \override Staff.StaffSymbol.line-count = #0
      \hide Staff.TimeSignature
      \hide Staff.Clef

      % dummy notes to force regular note spacing
      \once  \override Score.GridLine.thickness = #4.0
      c8 c c
      \once  \override Score.GridLine.thickness = #3.0
      c8 c c
      \once  \override Score.GridLine.thickness = #4.0
      c8 c c
      \once  \override Score.GridLine.thickness = #3.0
      c8 c c
    }
  >>
}

\layout {
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % center grid lines horizontally below note heads
    \override NoteColumn.X-offset = #-0.5
  }
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #1/8
    % set line length and positioning:
    % two staff spaces above center line on hidden staff
    % to four spaces below center line on visible staff
    \override GridPoint.Y-extent = #'(2 . -4)
```
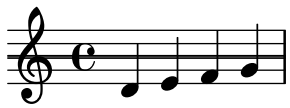
```
  }
}
```



Hacer unas líneas del pentagrama más gruesas que las otras

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



24 Headword

Encabezamiento de música antigua

Encabezamiento de música antigua.

```
#(set-global-staff-size 26)

\new VaticanaScore <<
  \new VaticanaVoice = "cantus" {
    \clef "vaticana-do3"

    % Verse 1 --- Sálve, Regína
    a\melisma \[ a \flexa g \pes a\melismaEnd \] d
      \divisioMinima
    \[ a\melisma \flexa g\melismaEnd \]
      \[ f\melisma \flexa e f \pes g \flexa f\melismaEnd \]
      \[ e\melisma \flexa d\melismaEnd \]
      \divisioMaior
    c d \[d\melisma \flexa c\melismaEnd \] d
      \[ e\melisma \pes f\melismaEnd\] g
      \[d\melisma \pes e \flexa c\melismaEnd \] d
      \finalis

    % Verse 2 --- Víta, dulcédo
    % a\melisma \[ a \flexa g \pes a\melismaEnd \] d
    %   \divisioMinima
    % \[ a\melisma \flexa g\melismaEnd \]
    %   \[ f\melisma \flexa e f \pes g \flexa f\melismaEnd \]
    %   \[ e\melisma \flexa d\melismaEnd \]
    %   \divisioMaior
    % c d \[e\melisma \pes f\melismaEnd \] g
    %   \[d\melisma \pes e \flexa c\melismaEnd \] d
    %   \finalis

    % Verse 3 --- Ad te clamámus
    \[ d\melisma \pes f\melismaEnd\] a g
      \[ g\melisma \flexa f \pes a\melismaEnd\] e
      \divisioMaior
    g f \[ e\melisma \flexa d \pes g\melismaEnd \]
      \divisioMinima
    c d \[ e\melisma \flexa d \pes g\melismaEnd \]
      \[ f\melisma \flexa e\melismaEnd \] d
      \finalis

    % Verse 4 --- Ad te suspirámus
    \[ d\melisma \pes f\melismaEnd \] a c' g
      \[ g\melisma \flexa f \pes g\melismaEnd \] a
      \divisioMaior
    d \[ f\melisma \pes \deminutum g\melismaEnd \] g d
      \[ \virga f\melisma \inclinatum e \inclinatum d\melismaEnd \] c
      \divisioMaior
```

```
d \[ d\melisma \flexa c \pes f\melismaEnd \]
  \[ g\melisma \pes a\melismaEnd \]
  g \[ f\melisma \flexa e\melismaEnd \] g
  \[ f\melisma \flexa \deminutum e\melismaEnd \]
  \[ d\melisma \flexa c \pes d\melismaEnd \]
  \finalis

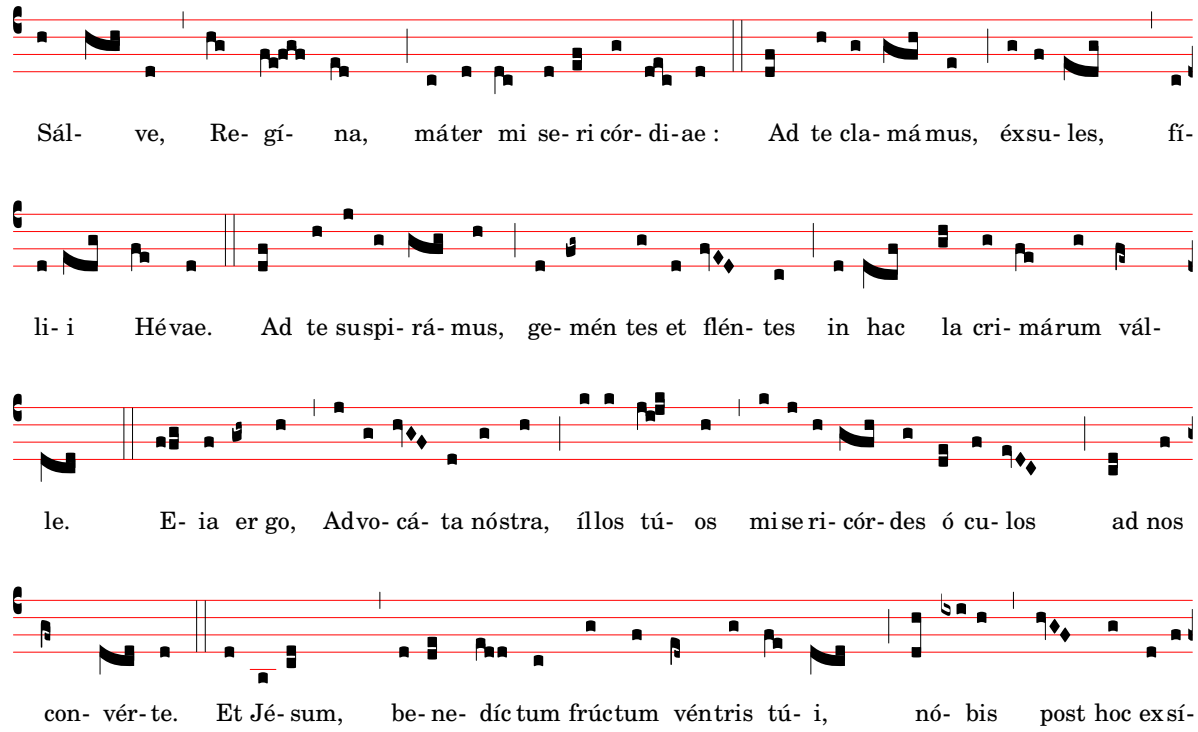
% Verse 5 --- Eia ergo, Advocáta nóstra
\[ f\melisma f \pes g\melismaEnd \] f
  \[ g\melisma \pes \deminutum a\melismaEnd \] a
  \divisioMinima
c' g \[ \virga a\melisma \inclinatum g \inclinatum f\melismaEnd \]
  d g a
  \divisioMaior
d' d' \[ c'\melisma \flexa b c' \pes d'\melismaEnd \] a
  \divisioMinima
d' c' a \[ g\melisma \flexa f \pes a\melismaEnd \] g
  \[ d\melisma \pes e\melismaEnd \] f
  \[ \virga e\melisma \inclinatum d \inclinatum c\melismaEnd \]
  \divisioMaior
\[ c\melisma \pes d\melismaEnd \]
  f \[ g\melisma \flexa \deminutum f\melismaEnd \]
  \[ d\melisma \flexa c \pes d\melismaEnd \] d
  \finalis

% Verse 6 --- Et Jésum
d a, \[ c\melisma \pes d\melismaEnd \]
  \divisioMinima
d \[ d\melisma \pes e\melismaEnd \]
  \[ e\melisma \flexa d d\melismaEnd \]
  c g f \[ e\melisma \flexa \deminutum d\melismaEnd \] g
  \[ f\melisma \flexa e\melismaEnd \]
  \[ d\melisma \flexa c \pes d\melismaEnd \]
  \divisioMaior
\[ d\melisma \pes a \] \virga bes\melismaEnd a
  \divisioMinima
\[ \virga a\melisma \inclinatum g \inclinatum f\melismaEnd \]
  g d f
  \[ f\melisma \flexa e\melismaEnd \]
  \[ d\melisma \flexa c\melismaEnd \]
  \divisioMinima
\[ e\melisma \pes f \flexa e\melismaEnd \] d d
  \finalis

% Verse 7 ad finem --- O clémens: O pía: O dúlcis Vírgo María
a\melisma \[a \flexa g a \quilisma b \pes c'\melismaEnd \]
  \[ \virga b\melisma \inclinatum a \inclinatum g\melismaEnd \] a
  \finalis
\[ g\melisma \pes a \quilisma b \pes c' \]
  \[ c' \flexa b\melismaEnd \]
  \[ a\melisma \flexa g\melismaEnd \]
  \[ g\melisma \pes a\melismaEnd \]
```

```
    \finalis
  \[ a\melisma \flexa d \virga f
    \inclinatum e \inclinatum d \inclinatum c d \]
    \divisioMinima
  \[ d \pes e f \pes g\melismaEnd \]
    \[ g\melisma \flexa \deminutum f\melismaEnd \]
    \[ g\melisma \pes a\melismaEnd \]
    d c d \[ d\melisma \pes g \flexa f f\melismaEnd \]
    \[ e\melisma \flexa d\melismaEnd \]
    \finalis
  s
}

\new VaticanaLyrics \lyricsto "cantus" {
  Sál -- ve, Re -- gí -- na,
    má -- ter mi -- se -- ri -- cór -- di -- "ae :"
  % Ví -- ta, dul -- cé -- do, et spes nó -- stra, sál -- ve.
  Ad te cla -- má -- mus, éx -- su -- les, fí -- li -- i
  Ad te su -- spi -- rá -- mus, ge -- mén -- tes et flén -- tes
    in hac la -- cri -- má -- rum vál -- le.
  E -- ia er -- go, Ad -- vo -- cá -- ta nó -- stra,
    íl -- los tú -- os mi -- se -- ri -- cór -- des
    ó -- cu -- los ad nos con -- vér -- te.
  Et Jé -- sum, be -- ne -- díc -- tum frúc -- tum vén -- tris tú -- i,
    nó -- bis post hoc ex -- sí -- li -- um os -- tén -- de.
  O clé -- "mens :"
  O pí -- "a :"
  O dúl -- cis Vír -- go Ma -- rí -- a.
}
>>
```





Encabezamiento de acordes

Encabezamiento de acordes.

```
theChords = \chordmode {
  \time 2/2
  f1 | c2 f2 | f1 | c2 f2| %\break
  f2 bes2 | f1 | c2:7 f | c1 | \break
}

verseOne = \lyricmode {
  \stanza "1."
  Fair is the sun - shine,
  Fair - er the moon - light
  And all the stars __ _  in heav'n a -- bove;
}

verseTwo = \lyricmode {
  \stanza "2."
  Fair are the mead - ows,
  Fair - er the wood - land,
  Robed in the flow -- ers of bloom -- ing spring;
}

Soprano = {
  \time 2/2
  \key f \major
  \stemUp
  f'2 f'4 f' | g'4 e' f'2 | a'4. a'8 a'4 a' | bes'4 g' a'2 |
  c''2 f''4 d'' |  c''2  bes'4  a' | bes'2 a' | g'1 |
}

Alto = {
  \key f \major
  c'2 c'4 c' | d'4 c' c'2 | f'4. f'8 f'4 fis' | g'4 e' f'2 |
  f'2 f'4 f' |  f'2  g'4  f' | e'2 f' | e'1 |
}

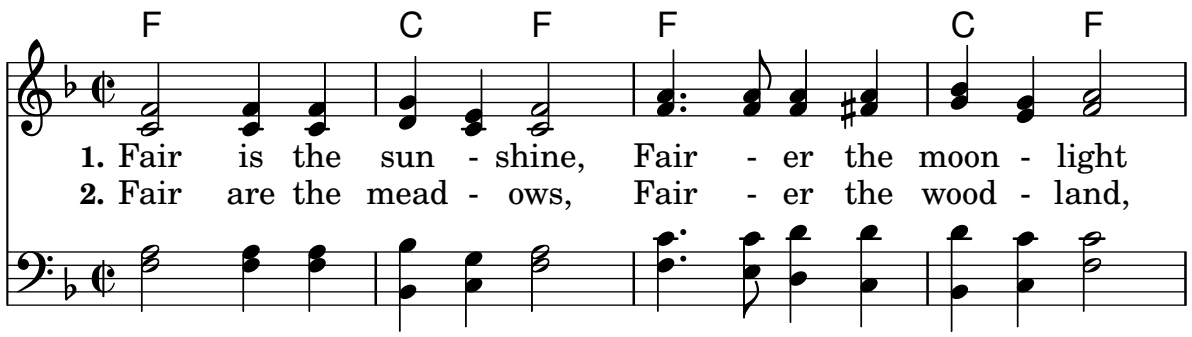
Tenor = {
  \key f \major
  \stemDown
  a2 a4 a | bes4 g a2  | c'4. c'8 d'4 d' | d'4 c' c'2 |
  a2 d'4 bes | a2 c'4 c' | c'2 c'  | c'1 |
}

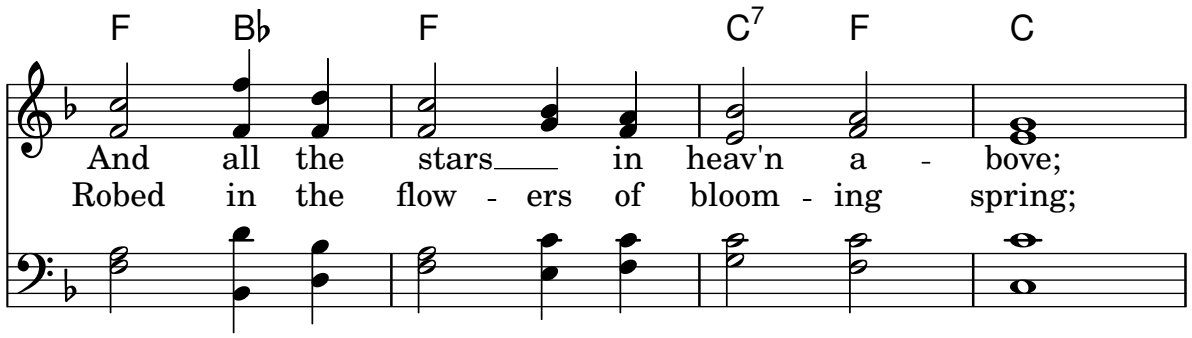
Bass = {
  \key f \major
  f2 f4 f | bes,4 c  f2 | f4. e8 d4 c | bes,4 c f2 |
```

```
  f2 bes,4 d | f2 e4 f | g2 f | c1 |
}


\score {
  <<
    \new ChordNames { \theChords }
    \context Staff = upper {
      \context Voice = sop {
        <<
          \Soprano
          \Alto
        >>
      }
    }
    \context Lyrics = "LyrOne" \lyricsto "sop" { \verseOne }
    \context Lyrics = "LyrTwo" \lyricsto "sop" { \verseTwo }
    \context Staff = lower {
      \new Voice {
        \clef bass
        \accidentalStyle modern-cautionary
        <<
          \Tenor
          \Bass
        >>
      }
    }
  >>

  \layout {
    indent = 0
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
    \context {
      \Voice
      \override StanzaNumber.padding = #1.8
    }
  }
}
```





Encabezamiento de indicaciones editoriales

Encabezamiento de Indicaciones editoriales.

```
\include "english.ly"

% Beethoven, Op. 31, No. 3
% Piano sonata 18, Movt II, Scherzo
% Measures 9 - 14

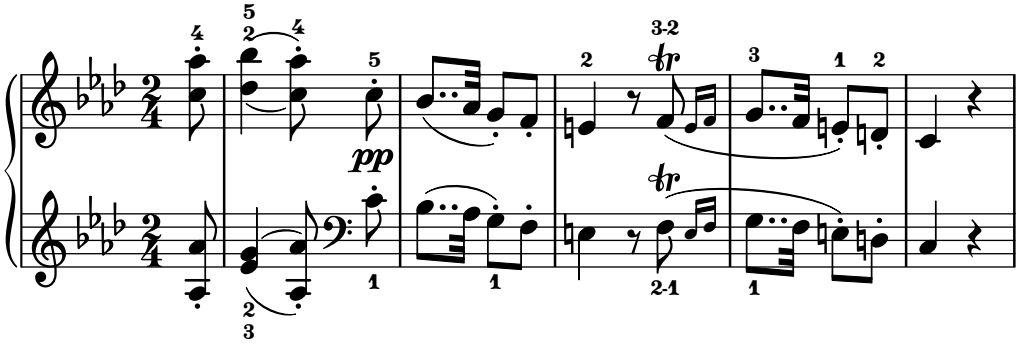
\new PianoStaff <<
  \new Staff = "right hand" {
    \clef treble
    \key af \major
    \time 2/4
    \set Staff.fingeringOrientations = #'(up)
    \set Score.currentBarNumber = #9

    \partial 8 <af''-4 c''>8-. |
    \once \set doubleSlurs = ##t
    <bf''-5 df''-2>4( <af''-4 c''>8-.) \noBeam c''8-5-.\pp |
    bf'8..( af'32 g'8)-. f'8-. |
    e'4-2 r8
      \once \override Script.script-priority = #-100
      \afterGrace f'8(\trill^\finger "3-2" { e'16[ f'16] } |
    g'8..-3 f'32 e'8-1)-. d'8-2-. |
    c'4 r4 |
  }

  \new Staff = "left hand" {
    \key af \major
    \clef treble
    \override Fingering.direction = #down
    \set Staff.fingeringOrientations = #'(down)

    \partial 8 <af' af>8-.
    \once \set doubleSlurs = ##t
    <g'-2 ef'-3>4( <af' af>8)-. \noBeam \clef bass c'8-1-. |
    bf8..( af32 g8-1)-. f8-. |
    e4 r8 \afterGrace f8(\trill_\finger "2-1" { e16[ f16] } |
    g8..-1 f32 e8)-. d8 -. |
    c4 r4 |
```

```
  }
>>
```



Encabezamiento de indicaciones expresivas

Encabezamiento de Indicaciones expresivas.

```
% L. v. Beethoven, Op. 49 no. 1
% Piano sonata 19 - "Leichte Sonate"
% measures 1 - 12

\include "english.ly"

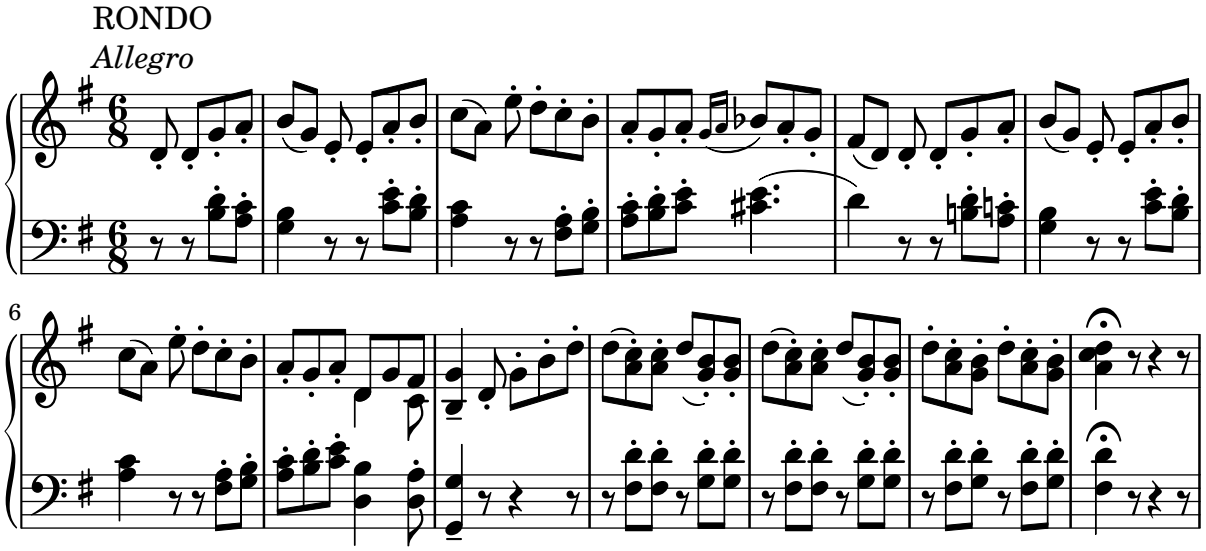
\new PianoStaff <<
  \new Staff = "right hand" {
    \clef treble
    \key g \major
    \time 6/8

    \partial 2
      \textMark \markup \override #'(baseline-skip . 3)
                        \column { RONDO
                                  \italic Allegro }
      d'8-. d'-. g'-. a'-. |
    b'8[( g']) e'-. e'-. a'-. b'-. |
    c''8[( a']) e''-. d''-. c''-. b'-. |
    a'8-. g'-. a'-. \acciaccatura { g'16[ a'] } bf'8 a'-. g'-. |
    fs'8[( d']) d'-. d'-. g'-. a'-. |
    % 5
    b'8[( g']) e'-. e'-. a'-. b'-. |
    c''8[( a']) e''-. d''-. c''-. b'-. |
    a'8-. g'-. a'-. << { d' g' fs' } \\
                      { d'4 c'8 } >> |
    <b g'>4-- d'8-. g'-. b'-. d''-. |
    d''8( <c'' a'>-.) <c'' a'>-. d''( <b' g'>-.) <b' g'>-. |
    % 10
    d''8( <c'' a'>-.) <c'' a'>-. d''( <b' g'>-.) <b' g'>-. |
    d''8-. <c'' a'>-. <b' g'>-. d''-. <c'' a'>-. <b' g'>-.  |
    <d'' c'' a'>4\fermata r8 r4 r8 |
  }

  \new Staff = "left hand" {
    \clef bass
```

```
    \key g \major
    \time 6/8

    \partial 2 r8 r <d' b>-. <c' a>-. |
    <b g>4 r8 r <e' c'>-. <d' b>-. |
    <c' a>4 r8 r <a fs>-. <b g>-. |
    <c' a>8-. <b d'>-. <e' c'>-. <e' cs'>4.( |
    d'4) r8 r <d' b!>-. <c'! a>-. |
    % 5
    <b g>4 r8 r <e' c'>-. <d' b>-. |
    <c' a>4 r8 r <a fs>-. <b g>-. |
    <c' a>8-. <d' b>-. <e' c'>-. <b d>4 <a d>8-. |
    <g g,>4 \tenuto r8 r4 r8 |
    r8 <d' fs>-. <d' fs>-. r <d' g>-. <d' g>-. |
    % 10
    r8 <d' fs>-. <d' fs>-. r <d' g>-. <d' g>-. |
    r8 <d' fs>-. <d' g>-. r <d' fs>-. <d' g>-. |
    <d' fs>4\fermata r8 r4 r8 |
  }
>>
```



Encabezamiento de bajo cifrado

Encabezamiento de Bajo cifrado.

```
% Arcangelo Corelli, 12 Sonate da Camera, Op. 2
% Sonata II, Allemanda
% measures 1 - 7

extendOn = \bassFigureExtendersOn
extendOff = \bassFigureExtendersOff

\score {
  \new StaffGroup <<
    \new GrandStaff <<
      \new Staff = "violinoI" \with { instrumentName = "Violino I." }
      {
```

```
    \time 4/4
    \tempo Adagio
    \partial 8 r16 a' |
    a'8.[ d''16 d''8. e''16] cis''8 a'4 a''16 bes'' |
    cis''8 d''16( e'') e''8. d''16 d''4 r8 d''16 e'' |
    f''8 f''4 g''16( f'') e''8 e''4 f''16( e'') |
    d''8. d''16 g''( f'') e''( d'') cis''8 cis''4 cis''16 cis'' |
    d''8 d'' c''8. c''16 c''8( b'4) b'16 b' |
    c''8 c'' bes'8. bes'16 bes'8( a'4) a''16 a'' |
    a''8 g'' g''8. g''16 g''8( f'') r f'' |
  }
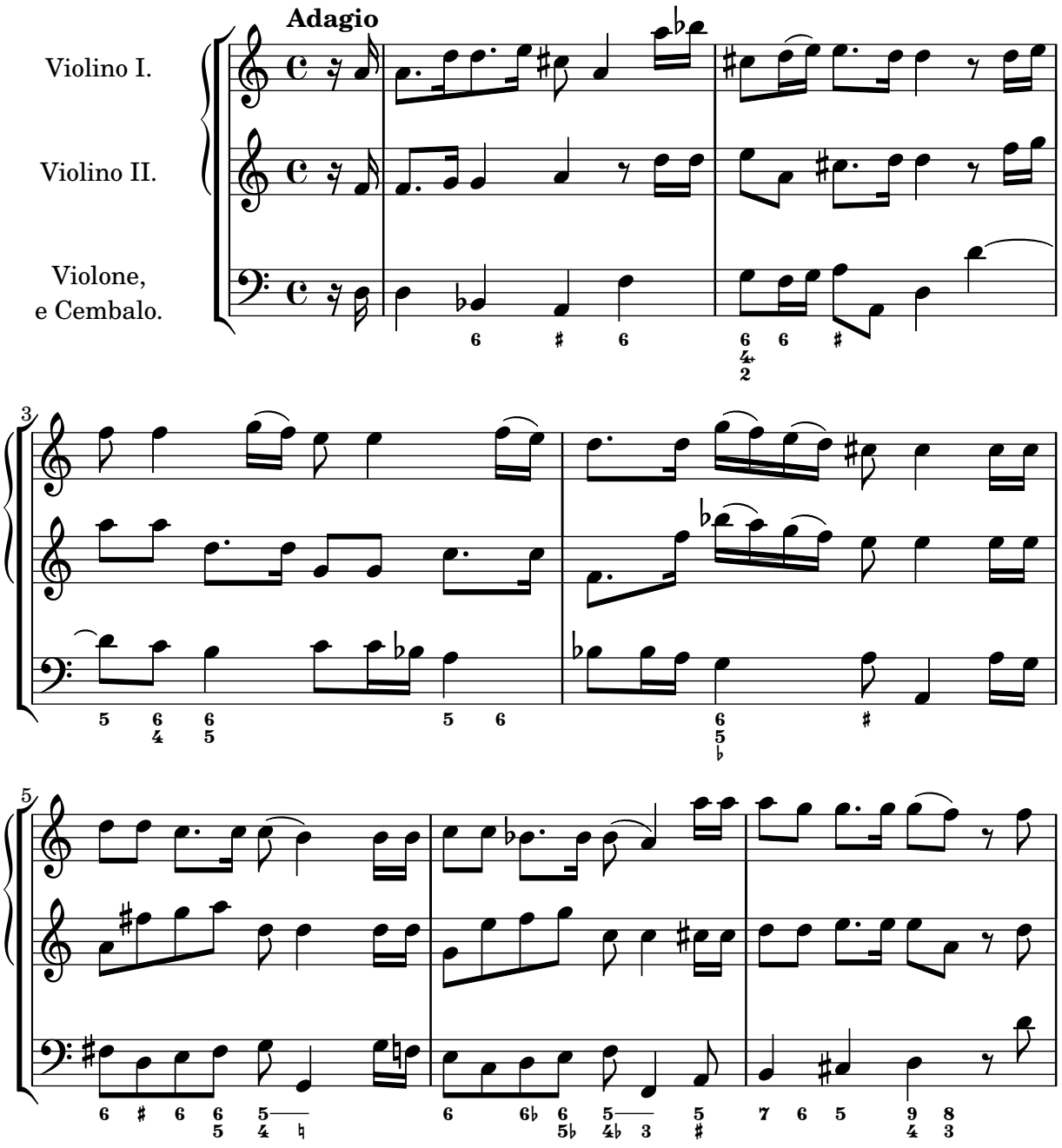
  \new Staff = "violinoII" \with { instrumentName = "Violino II." }
  {
    \time 4/4
    \partial 8 r16 f' |
    f'8. g'16 g'4 a' r8 d''16 d'' |
    e''8 a' cis''8. d''16 d''4 r8 f''16 g'' |
    a''8 a'' d''8. d''16 g'8 g' c''8. c''16 |
    f'8. f''16 bes''( a'') g''( f'') e''8 e''4 e''16 e'' |
    a'8 fis'' g'' a'' d'' d''4 d''16 d'' |
    g'8 e'' f'' g'' c'' c''4 cis''16 cis'' |
    d''8 d'' e''8. e''16 e''8 a' r d'' |
  }
>>

\new Staff = "violone" \with {
  instrumentName = \markup {
    \center-column { Violone,
                     "e Cembalo." } } }
{
  \time 4/4
  \clef bass
  \partial 8 r16 d |
  d4 bes, a, f |
  g8 f16 g a8 a, d4 d' ~ |
  d'8 c' b4 c'8 c'16 bes a4 |
  bes8 bes16 a g4 a8 a,4 a16 g |
  fis8 d e fis g8 g,4 g16 f |
  e8 c d e f8 f,4 a,8 |
  b,4 cis d r8 d' |
}

\new FiguredBass \figuremode {
  \set figuredBassAlterationDirection = #RIGHT
  \set figuredBassPlusDirection = #RIGHT
  \override BassFigureAlignment.stacking-dir = #DOWN
  s8 |
  s4 <6> <_+> <6> |
  <6 4\+ 2>8 <6> <_+> s s2 |
  <5>8 <6 4> <6 5>4 s <5>8 <6> |
  s4 <6 5 _-> <_+>2 |
```

```
      <6>8 <_+> <6> <6 5> <5 4> \extendOn <5 _!> \extendOff s4 |
      <6>4 <6->8 <6 5-> <5 4-> \extendOn <5 3>4 \extendOff <5 _+>8 |
      <7>8 <6> <5>4 <9 4>8 <8 3> s4 |
    }
  >>
}

\layout {
  indent = 3\cm
}
```



Encabezamiento de instrumentos con trastes

Encabezamiento de instrumentos con trastes.

```
% Johann Kaspar Mertz, Opern-Revue op. 8
% No. 17 ("Bellini, Norma")
% measures 123 - 133

%%%% shortcuts
% fingering orientations
sfol = \set fingeringOrientations = #'(left)
sfor = \set fingeringOrientations = #'(right)
sfod = \set fingeringOrientations = #'(down)
sfou = \set fingeringOrientations = #'(up)

% string number orientations
ssnol = \set stringNumberOrientations = #'(left)  %(down right up)
ssnou = \set stringNumberOrientations = #'(up)
ssnod = \set stringNumberOrientations = #'(down)
ssnor = \set stringNumberOrientations = #'(right)

% define fingering offset
FO = #(define-music-function (offsetX offsetY) (number? number?)
#{
  \once \override Voice.Fingering.extra-offset = #(cons offsetX offsetY)
#})

% markups
rit = \markup \bold \italic "rit."
dimin = \markup \italic "dim."
benmarcato = \markup \italic \bold "il canto ben marcato"
pdolce = #(make-dynamic-script
           (markup #:line (#:dynamic "p" #:normal-text #:italic "dol.")))

% triplet
T = \tuplet 3/2 \etc


%%% THE MUSIC %%%

melody = \relative c {
  \voiceOne
  \clef "treble_8"
  \key d \major
  \once \omit Staff.TimeSignature
  \time 4/4

  \sfol
  e,32[ a' c e] e,[ a c e] e,,[ a' c e] e,[ a c e]
    f4\rest <e'-4>4-> | % m. 1
  e,,,32[ gis' b e] e,[ gis b e] e,,[ gis' b e] e,[ gis b e]
    f4\rest \FO #0.4 #0.5 <gis-1 e'-4>4 | % m. 2
  d4\rest <b e>-> d4\rest^\rit <b e>4-> | % m. 3
  <gis b e>1 \bar "||" % m. 4

  \tempo \markup \larger \italic "Andantino"
```

```
  \key a \minor
  \time 4/4
  R1 | % m. 5
  e'4^\benmarcato e8. d16-4
    d4-4 \T { \sfou \FO #-0.3 #0.6 <c-2>4 b8 } | % m. 6
  \FO #-0.3 #0.3 <a-3>4 \T { c4 b8 } a4 e'8. e16 | % m. 7
  \FO #-0.3 #0.3 <g-4>4 \T { \sfol \FO #0.3 #0.0 <f-1>4 e8 }
    e4 \T { \sfou <d-4>4 c8 } | % m. 8
  b4 \T { d4-4 c8 } \sfou \FO #-1.7 #-1.5 <b-0>4 e | % m. 9
  e4 e8. d16-4 d4 \T { c4 b8 } | % m. 10
  \T { a4 a8 b4 c8 }
    \sfou \FO #-0.3 #0.3 <d-4>4^\< \T { e4 <d f>8\! } | % m. 11
}

bass = \relative c {
  \voiceTwo
  \key d \major
  \time 4/4

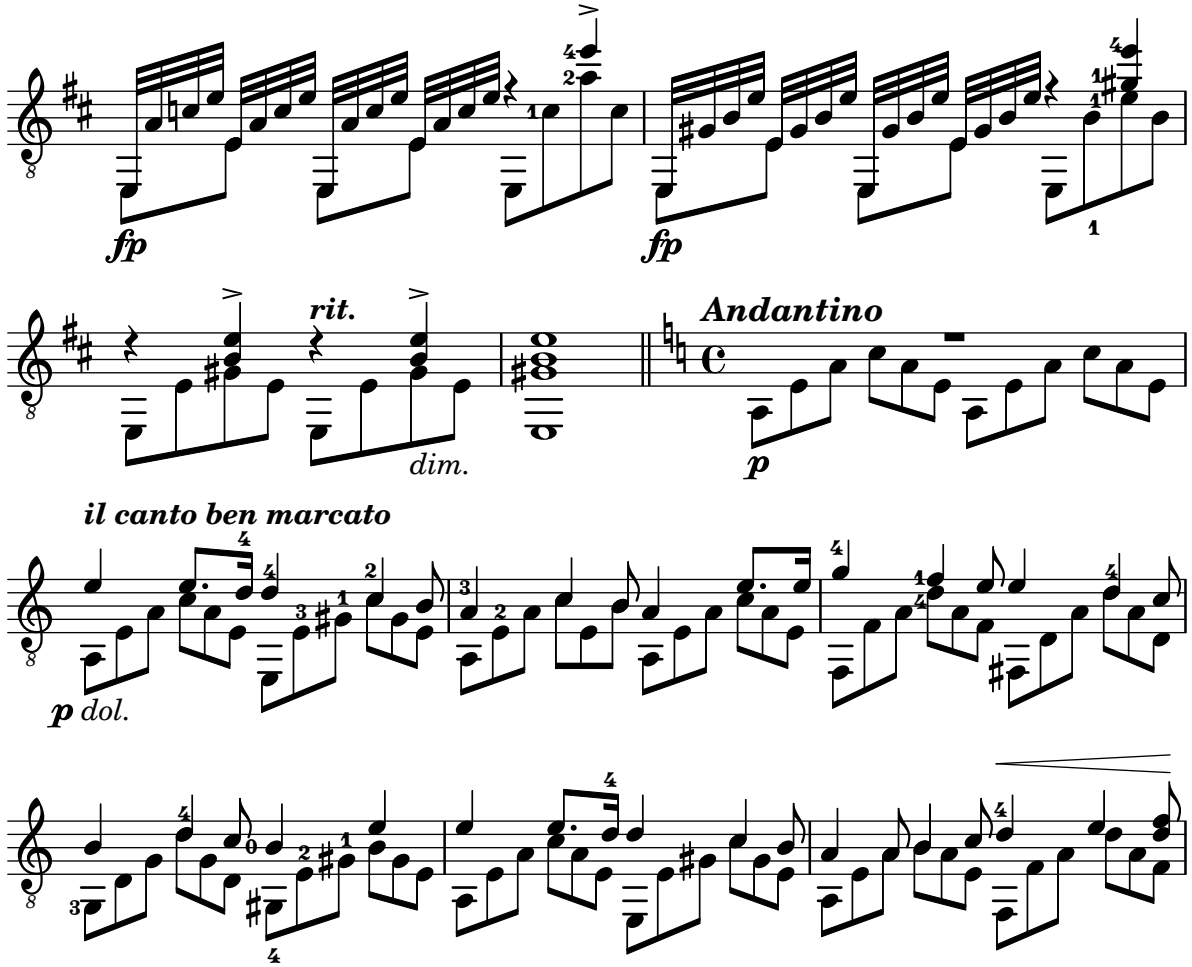
  e,8\fp[ e'] e,[ e'] e, \sfol <c''-1> <a'-2> c, | % m. 1
  e,,8\fp[ e'] e,[ e']
    e, \sfod \FO #0.2 #-0.2 <b''-1> \sfol \FO #0.3 #0.0 <e-1> b | % m. 2
  e,,8 e' gis e e, e' gis_\dimin e | % m. 3
  e,1 | % m. 4

  \T { a8\p e' a c a e a, e' a c a e } | % m. 5
  \T { a,8\pdolce e' a c a e }
    \T { e,8 \sfou <e'-3> <gis-1> c gis e } | % m. 6
  \T { a,8 <e'-2> a c e, b' a, e' a c a e } | % m. 7
  \T { f,8 f' a \sfol \FO #0.3 #-0.5 <d-4> a f }
    \T { fis, d' a' d a d, } | % m. 8
  \T { <g,-3>8 d' g d' g, d }
    \T { \sfod <gis,-4> \sfou <e'-2> <gis-1> b gis e } | % m. 9
  \T { a,8 e' a c a e e, e' gis c gis e } | % m. 10
  \T { a,8 e' a b a e f, f' a d a f } | % m. 11
}

\new Staff = "guitar" <<
  \context Voice = "upper" { \melody }
  \context Voice = "lower" { \bass }
>>

\layout {
  \context {
    \Score
    \remove "Bar_number_engraver"
    \override Fingering.staff-padding = #'()
    \omit TupletNumber
    \override TupletBracket.bracket-visibility = ##f
  }
}
```

```
\paper {
  system-system-spacing.padding = 3
}
```



Encabezamiento de instrumentos de teclado

Keyboard headword.

```
% M. Ravel, Sonatine (1905)
% End of first movement

\include "english.ly"

\layout {
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}

fermataLong = \markup {
  \override #'(direction . 1)
  \override #'(baseline-skip . 2) {
    \dir-column {
      \fermata
```

```
      \serif \italic \center-align long
    }
  }
}
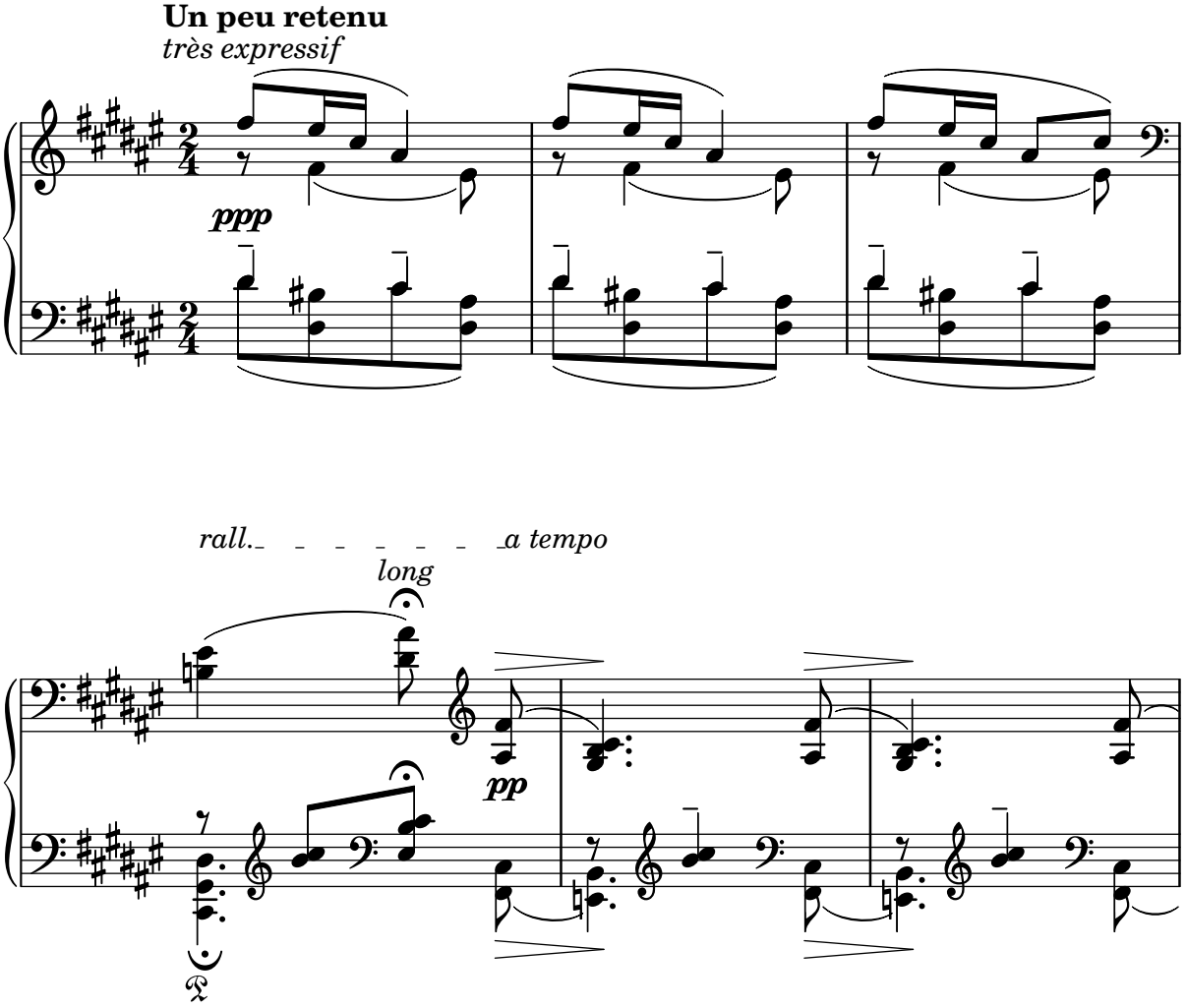
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    \time 2/4
    \key fs \major
    <<
      \new Voice {
        \voiceOne
        \textMark \markup {
          \override #'(baseline-skip . 2.4) \column {
            \line \bold { Un peu retenu }
            \line \italic { très expressif } } }
        fs''8( es''16 cs'' as'4) |
        fs''8( es''16 cs'' as'4) |
        fs''8( es''16 cs'' as'8 cs''8) |
      }
      \new Voice {
        \voiceTwo
        gs'8\rest \offset Y-offset #-3 \ppp fs'4( es'8) |
        gs'8\rest fs'4( es'8) |
        gs'8\rest fs'4( es'8) |
      }
    >>
    \clef bass
    \override TextSpanner.bound-details.left.text = "rall."
      \override TextSpanner.bound-details.right.text = "a tempo"
      <b! es'>4(\startTextSpan
      \override Script.stencil =
        #(lambda(grob)
           (grob-interpret-markup grob fermataLong))
      <ds' as'>8)\fermata \noBeam
      \clef treble
      <as fs'>8^(
               \tweak to-barline ##f
                 \tweak after-line-breaking ##f
                 \tweak endpoint-alignments #'(-1 . 0) ^\>
               \stopTextSpan | \noBreak
    <gs b cs'>4.\!)
      <as fs'>8^(
               \tweak to-barline ##f
                 \tweak after-line-breaking ##f
                 \tweak endpoint-alignments #'(-1 . 0) ^\> |
    <gs b cs'>4.\!)
    <<
      \new Voice {
        \voiceOne
        <as fs'>8( |
```
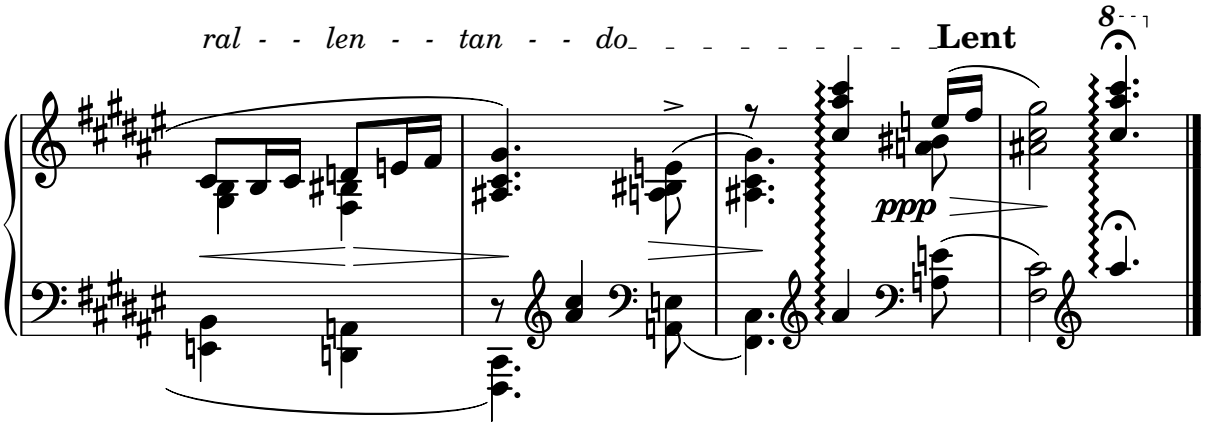
```
      \override TextSpanner.bound-details.left.text =
        "ral  -   -   len   -   -   tan   -   -   do"
        \override TextSpanner.bound-details.right.text =
          \markup \larger \upright \bold "Lent"
      cs'8\<\startTextSpan b16 cs'
        d'8\tweak to-barline ##f \> e'16 fs' |
      <as! cs' gs'>4.)\! s8 |
      r8 <cs'' as'' cs'''>4\arpeggio
        e''16(\stopTextSpan fs''16 |
      \voiceTwo
      <as'! cs'' gs''>2) |
    }
    \new Voice {
      \voiceTwo
      s8 |
      <gs b>4 <fs bs>4 |
      s4. <a bs e'>8^(^> \tweak to-barline ##f _\> |
      <as! cs' gs'>4.)\!
      <a' bs'>8\tweak X-offset #-4 \ppp \tweak to-barline ##f \> |
      s8\!
        \voiceOne
        \ottava 1
        \once \override PianoStaff.Arpeggio.padding = 0.8
        <cs''' as''' cs''''>4. \arpeggio \fermata
        \ottava 0
        \bar "|."
    }
  >>
}

\new Staff <<
  \key fs \major
  \clef bass
  \new Voice {
    \voiceOne
    ds'4-- cs'4-- |
    ds'4-- cs'4-- |
    ds'4-- cs'4-- |
    r8 \clef treble <b' cs''>8[ \clef bass <es b cs'>8]\fermata
      s8^\tweak Y-offset 3 \pp |
    fs8\rest \clef treble <b' cs''>4-- s8 |
    fs8\rest \clef treble <b' cs''>4-- s8 |
    s2 |
    ds8\rest \clef treble <as' cs''>4 \clef bass s8 |
    s8 \clef treble <as'>4 \arpeggio \clef bass s8 |
    s8 \clef treble <as''>4. \arpeggio \fermata |
  }
  \new Voice {
    \voiceTwo
    ds'8[( <ds bs> cs' <ds as>]) |
    ds'8[( <ds bs> cs' <ds as>]) |
    ds'8[( <ds bs> cs' <ds as>]) |
```

```
      \set Staff.pedalSustainStrings = #'("P" "" "")
        <cs, gs, ds>4.\sustainOn \fermata
        <fs, cs>8(\sustainOff
                  \tweak to-barline ##f
                    \tweak after-line-breaking ##f _\> |
      <e, b,>4.)\! \clef bass
        <fs, cs>8( \tweak to-barline ##f
                    \tweak after-line-breaking ##f _\> |
      <e, b,>4.)\! \clef bass <fs, cs>8( |
      <e, b,>4 <d, a,> |
      <fs,, cs,>4.) <a, e>8( |
      <fs, cs>4.) <a e'>8^( |
      <fs cs'>2) |
    }
  >>
>>

\paper {
  system-system-spacing.padding = 4
}
```





Encabezamiento de alturas

Encabezamiento de alturas.

```
% L. v. Beethoven
% Piano sonata 21 op. 53 - "à Monsieur Comte de Waldstein"
% first movement, measures 34 - 41

\include "english.ly"

\new PianoStaff <<
  \new Staff = "right hand" <<
    \new Voice = "right hand voice 1" {
      \set Score.currentBarNumber = 34
      \voiceOne gs''2(^\markup \italic "dolce e molto ligato" fs''4 e'' |
      ds''2 cs'') |
      ds''2( e''4 fs'' |
      <gs'' e''>2 <fs'' ds''>) \clef bass |

      \oneVoice <gs' e' b>2( <fs' ds' a>4 <e' cs' gs> |
      <ds' bs fs>2 <cs' a e>) |
      \voiceOne b2\tweak height-limit 7
                  \tweak positions #'(6 . 2) ( cs'4 ds' \clef treble |
      <e' gs>4) r4 r2 |
    }

    \new Voice = "right hand voice 2" {
      \voiceTwo <e'' b'>2 <ds'' a'>4 <cs'' gs'> |
      <bs' fs'>2 e' |
      <b'! a'>2 b'4 <e'' cs''> |
      b'2.( a'4) \clef bass | \break

      s1 |
      s1 |
      <gs e>4( <a fs>2.) |
      s4 r4 r2 |
    }
  >>

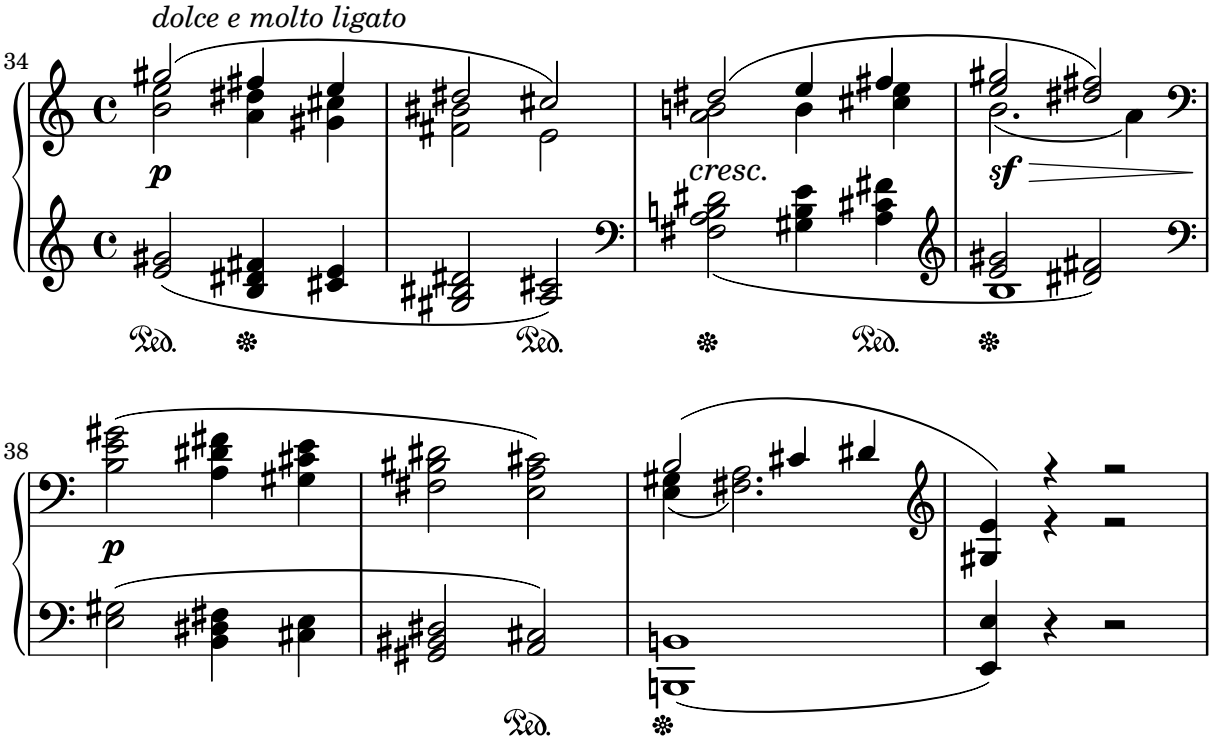
  \new Dynamics {
```

```
    s1*2\p |
    s1\tweak style #'none \cresc |
    s1\sf\> |

    s1*4\p |
  }

  \new Staff = "left hand" {
    \override Staff.SustainPedalLineSpanner.staff-padding = 6
    <gs' e'>2(\sustainOn <fs' ds' b>4\sustainOff <e' cs'> |
    <ds' bs gs>2 <cs' a>)\sustainOn \clef bass |
    <ds' b! a fs>2_(\sustainOff
      <e' b gs>4 <fs' cs' a>\sustainOn \clef treble |
    << { \voiceOne <gs' e'>2 <fs' ds'>) }
       \new Voice { \voiceTwo b1\sustainOff } >> \clef bass |

    \oneVoice <gs e>2( <fs ds b,>4 <e cs> |
    <ds bs, gs,>2 <cs a,>)\sustainOn |
    <b,! b,,!>1(\sustainOff |
    <e e,>4) r4 r2 |
  }
>>
```



Encabezamiento de repeticiones

Encabezamiento de repeticiones

```
% Beethoven, Op. 57
% Piano sonata 23 -
%   (Appassionata, "Dem Grafen Franz von Brunsvik gewidmet")
% Movt II, Andante con moto
% Measures 9 - 16

\include "english.ly"

\new PianoStaff <<
  \new Staff = RH {
    \clef treble
    \key df \major
    \time 2/4
    \set Score.currentBarNumber = #9

    \bar ".|:-|"
    \repeat volta 2 {
      \change Staff = LH \voiceOne <af ef c>4 <af gf c>8.. <af ef c>32 |
      <af f df>8. <df' af f>16 <c' af gf>8 <df' af f>8 |
      <af ef c>4 <af gf c>8.. <af ef c>32 |
      <af f df>8. \change Staff = RH f'16 f'^( ef' df'8)

      \change Staff = LH \voiceOne <af ef c>4 <af gf c>8.. <af ef c>32 |
      \change Staff = RH <af' df' af>8. <af' f' df'>16
        << { af'16( gf' f'8) } \\
           { <ef' c'>8 <df' af> } >> |
      ef'4\tweak height-limit 5 ^(
        \change Staff = LH \voiceOne <af ef c> |
      <df' f df>4.) r8
    }
  }

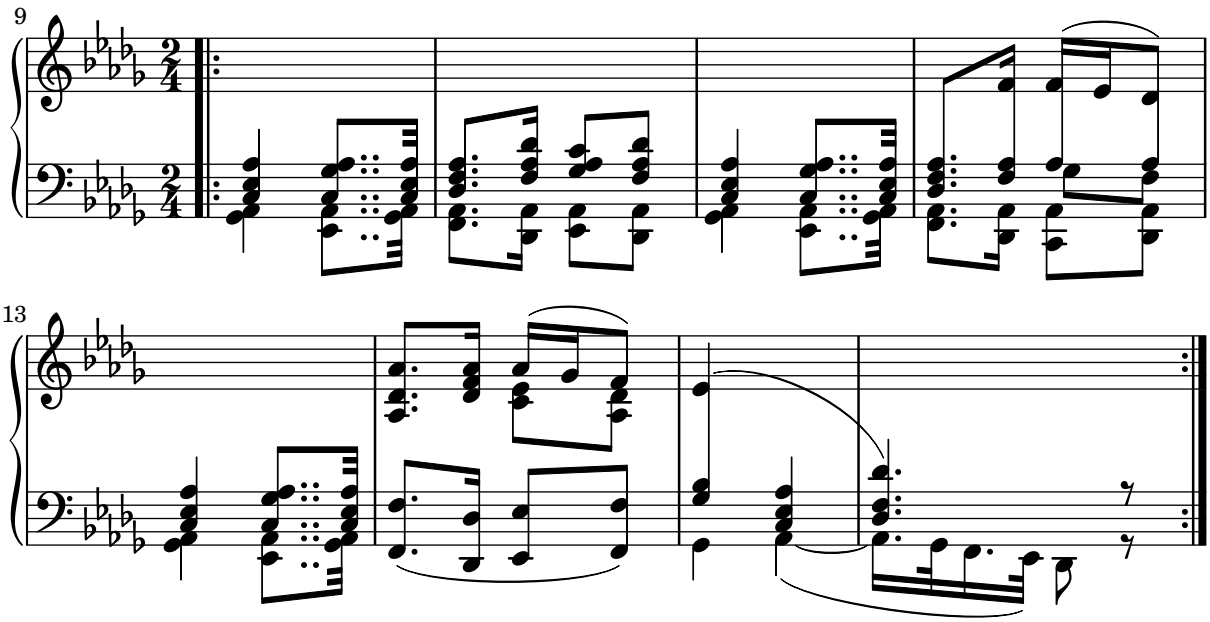
  \new Staff = LH {
    \clef bass
    \key df \major
    \time 2/4

    \repeat volta 2 {
      \voiceTwo <af, gf,>4 <af, ef,>8.. <af, gf,>32 |
      <af, f,>8. <af, df,>16 <af, ef,>8 <af, df,> |
      <af, gf,>4 <af, ef,>8.. <af, gf,>32 |
      << { \voiceTwo s4 gf8\tweak positions #'(-0.3 . -0.8) [ f] } \\
         { \voiceOne s8. \crossStaff {<af f>16 af s af8 } } \\
         { \voiceTwo <af, f,>8. <af, df,>16
             \once\shiftOn <af, c,>8 <af, df,> } >> |

      <af, gf,>4 <af, ef,>8.. <af, gf,>32 |
      \oneVoice <f f,>8.( <df df,>16 <ef ef,>8 <f f,>) |
      << { \voiceOne \crossStaff <gf bf>4 s |
           s2 } \\
         { \voiceTwo gf,4 af,( ~ |
           af,16.[ gf,32 f,16. ef,32]) df,8 r8 } >> |
```

```
    }
  }
>>

\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}
```



Encabezamiento de Duraciones

Encabezamiento de Duraciones.

```
% Beethoven, Op. 81a
% Piano sonata 26 - (Les Adieux)
% Movt II - Abwesenheit (L'Absence)
% Measures 31 - 34

\include "english.ly"

% Circumvent issue #6876: `strict-grace-spacing` ignores
% accidentals of the following main note.
shiftedGrace =
#(define-music-function (offset music) (number? ly:music?)
   #{
     \override NoteHead.X-offset = #(- offset 0.85)
     \override Stem.X-offset = #offset
     \grace { $music }
     \revert NoteHead.X-offset
     \revert Stem.X-offset
   #})

\new PianoStaff <<
```

```
  \new Staff = "right hand" {
    \clef treble
    \key c \minor
    \time 2/4
    \set Score.currentBarNumber = #31

    <c''' c''>8(^\markup \italic "a tempo" <g'' g'>) ~
      <g'' g'>8( <a'' a'>16 <f'' f'>) |
    <f'' f'>8[( \shiftedGrace #-0.15 { e''32 f'' e'' d'' }
      <e''! e'>16 <f'' f'>16]) <g'' g'>16-.([ <a'' a'>-.)]
      <bf'' bf'>32( <b'' b'>) <b'' b'>( <c''' c''>) |
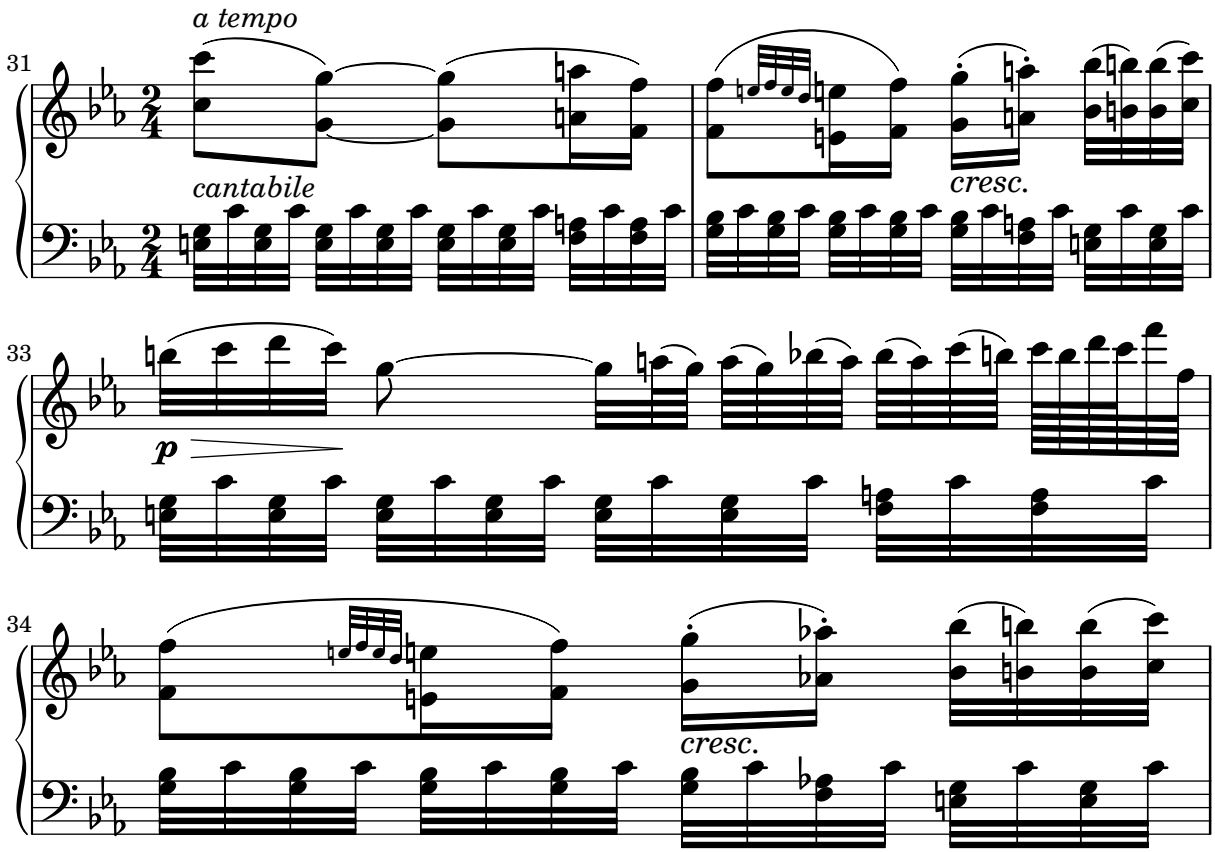
    b''32([ c''' d''' c'''32)] g''8 ~
      g''32[ a''64( g'']) a''64([ g'') bf''( a''64)]
      bf''64([ a'') c'''( b''64)] c'''128[ b'' d''' c''' f'''64 f''] |
    <f'' f'>8[( \shiftedGrace #-0.15 { e''32 f'' e'' d''] }
      <e''! e'>16 <f'' f'>]) <g'' g'>16-.([ <af''! af'!>-.)]
      <bf'' bf'>32( <b'' b'>) <b'' b'>( <c''' c''>) |
   }

  \new Dynamics {
    s2\offset Y-offset 1 -\markup \italic "cantabile" |
    s4 s\tweak style #'none \cresc |
    s16.\offset Y-offset 1 \p \offset Y-offset 1 \> s32\! s4. |
    s4 s\tweak style #'none \cresc <>\! |
  }

  \new Staff = "left hand" {
    \set Staff.beatBase = #1/8
    \set Staff.beatStructure = 1,1,1,1
    \clef bass
    \key c \minor
    \time 2/4
    \*3 { <g e>32 c' <g e> c' } <a f> c' <a f> c' |
    \*2 { <bf g>32 c' <bf g> c' }
      <bf g> c' <a f> c' <g e> c' <g e> c' |

    \*3 { <g e>32 c' <g e> c' } <a f> c' <a f> c' |
    \*2 { <bf g>32 c' <bf g> c' }
      <bf g> c' <af! f> c' <g e> c' <g e> c' |
   }
>>

\layout {
  \context {
    \Score
    \override SpacingSpanner.base-shortest-duration =
      \musicLength 1*1/40
    \override SpacingSpanner.strict-grace-spacing = ##t
  }
}
```



Encabezamiento de notas simultáneas

Encabezamiento de Notas simultáneas.

```
\include "english.ly"

% L. v. Beethoven, Op. 111
% Piano sonata 32
% Movt II - Arietta - Adagio molto semplice e cantabile
% measures 108 - 118

trillFlat =
\once \override TrillSpanner.bound-details.left.text = \markup {
  \concat {
    \musicglyph "scripts.trill"
    \translate #'(-0.5 . 1.9) \fontsize #-7 \flat
  }
}

\new PianoStaff <<
  \new Staff = "right hand" <<
    \set Score.currentBarNumber = #108
    \new Voice = "right hand 1" {
      \clef treble
      \key c \major
      \time 9/16

      \grace s32 s4. s8. |
```

```
    s4. \voiceOne a''8[(\p g''16]) |
    g''4.\dim af''8[( g''16]) |
    g''8.[ g''8. g''8.] |
    g''8.[\pp af''8.af''8.] |
    af''8.[ af''8.af''8.] |

    \trillFlat af''4.\startTrillSpan\< ~ af''8. ~ |
    af''4.\> ~ af''8. ~ |
    \oneVoice <af'' d''>8.[\p\cresc a''8. bf''8.] ~ |
    bf''8.[ b''8. c'''8.] ~ \bar "||"
    \key ef \major c'''8.[ cs'''8.] <>\stopTrillSpan <>\!
  }

  \new Voice = "right hand 2" {
    \override Voice.TrillSpanner.direction = #DOWN
    \grace cs''32 \voiceTwo d''4.\f\startTrillSpan ~ d''8. ~ |
    d''4. ~ d''8. ~ |
    d''8. <>\stopTrillSpan\trillFlat d''4.\startTrillSpan ~ |
    d''4. ~ d''8. ~ |
    d''4. ~ d''8. ~ |
    d''4. ~ d''8. ~ <> \stopTrillSpan |
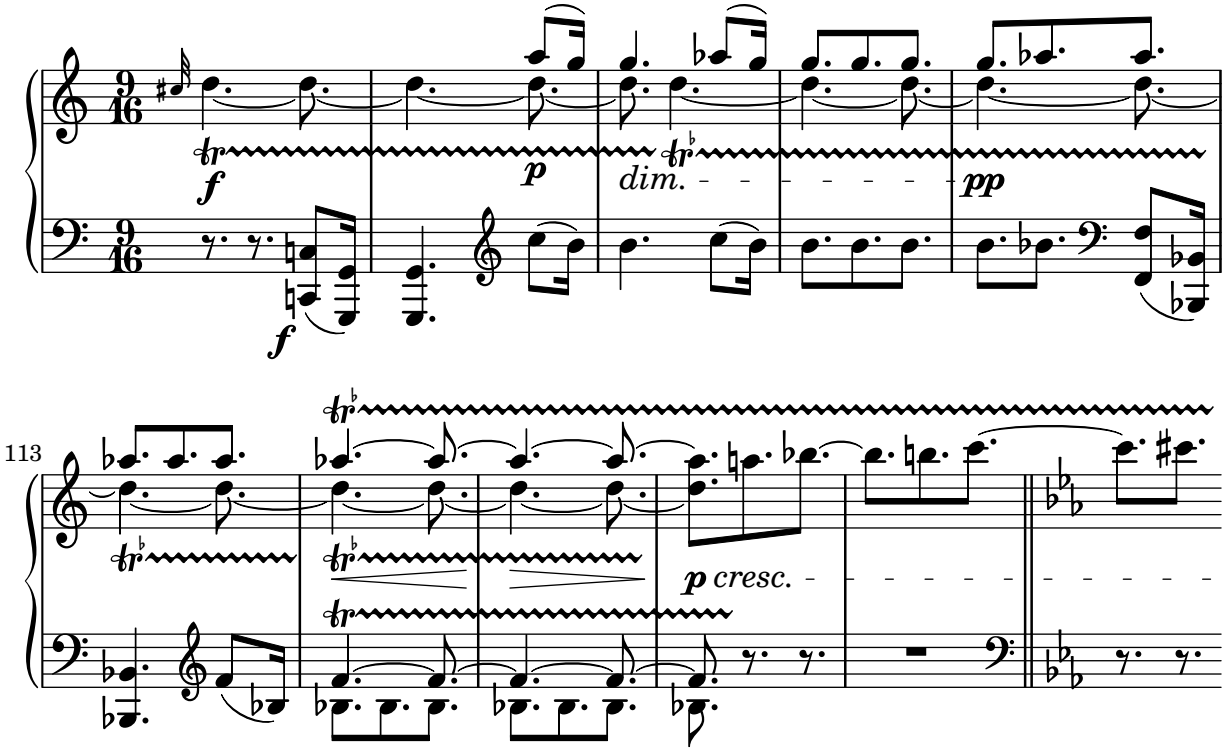
    \trillFlat d''4.\startTrillSpan ~ d''8. ~ |
    d''4. ~ d''8. ~ |
    \once \override NoteColumn.ignore-collision = ##t
      \hideNotes d''8.\stopTrillSpan s4. |
    s4. s8. |
    s4.
  }
>>

\new Staff = "left hand" {
  \clef bass
  \key c \major
  \time 9/16

  \grace s32 r8. r8. <c! c,!>8[(\tweak X-offset #-2 _\f <g, g,,>16]) |
  <g, g,,>4. \clef treble c''8[( b'16]) |
  b'4. c''8[( b'16]) |
  b'8.[ b'8. b'8.] |
  b'8.[ bf'8.] \clef bass <f f,>8[( <bf, bf,,>16]) |
  <bf, bf,,>4. \clef treble f'8[( bf16]) |

  <<
    \new Voice {
      \voiceOne
      \override Voice.TrillSpanner.direction = #UP
      f'4.~ \startTrillSpan f'8.~ |
      f'4.~ f'8.~ |
      f'8. <> \stopTrillSpan
    }
    \new Voice {
```

```
        \voiceTwo
        \override Voice.TrillSpanner.direction = #DOWN
        bf8.[ bf8. bf8.] |
        bf8.[ bf8. bf8.] |
        bf8.
      }
    >> \oneVoice r8. r8. |
    R1*9/16 \clef bass |
    \key ef \major r8. r8.
  }
>>
```



Encabezamiento de pentagramas

Staff headword.

```
\include "catalan.ly"

% Piotr Ilitch Tchaïkovski
% Le Lac des Cygnes, op. 20
% Danse Napolitaine
% arr. Laurence Sardain (Mutopia 2006/12/22)

#(set-global-staff-size 18)

trompette = \relative do'' {
  \clef treble
  \key mib \major
  \time 2/4

  <>\tweak staff-padding #1.5 ^\markup { \larger \italic Comodo } R2 |
```

```
  r8 \once \override TextScript.padding = #2.0
    sib16-.^\markup { \dynamic p \italic grazioso } do-.
    mib16( re)-. do-. sib-. |
  re8-. r8 re4->( |
  re8) do16-. re-. mib( re) do-. re-. |
  do8-. r8 sib4-> |
}

tambourin = \drummode {
  \time 2/4
  r8 tamb16 16 8 8 |
  r8 16 16 8 8 |
  r8 8 r8 8 |
  r8 16 16 8 8 |
  r8 8 r8 8 |
}

upper = \relative do' {
  \clef treble
  \key mib \major
  \time 2/4

  r8\p <sol sib mib>16-. q-. q8-. q-. |
  r8 <sol sib mib>16-. q-. q8-. q-. |
  r8 <lab sib re>16-. q-. q8-. q-. |
  r8 <lab sib re>16-. q-. q8-. q-. |
  r8 <sol sib mib>16-. q-. q8-. q-. |
}

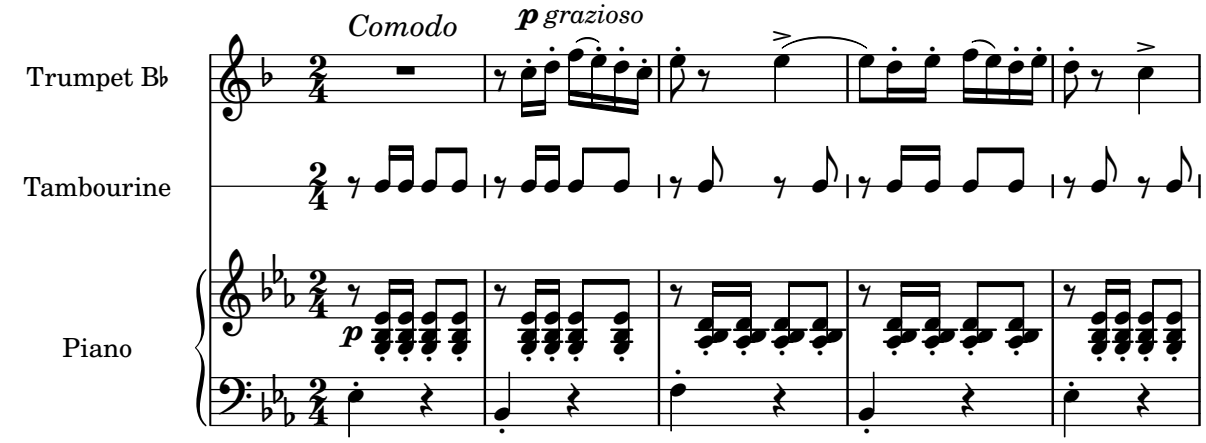
lower = \relative do {
  \clef bass
  \key mib \major
  \time 2/4

  mib4-. r4 |
  sib-. r |
  fa'-. r |
  sib,-. r |
  mib4-. r4 |
}

\score {
  <<
    \context Staff = "trumpet" \with {
      instrumentName = \markup {
        "Trumpet" \concat { B \teeny \raise #0.4 \flat } }
    } \transpose sib do' \trompette
    \context RhythmicStaff = "tambourin" \with {
      instrumentName = "Tambourine"
    } \tambourin
    \context PianoStaff = "prima" \with {
      instrumentName = "Piano"
```

```
    } <<
        \context Staff = "uppera" \upper
        \context Staff = "lowera" \lower
      >>
  >>
  \layout { indent = 2.5\cm }
}

\score {
  <<
    \context Staff = "trumpet" \with { midiInstrument = "trumpet" }
      \trompette
    \context DrumStaff = "tambourin"
      \tambourin
    \context Staff = "piano"
      <<
        \upper
        \lower
      >>
  >>
  \midi { \tempo 4 = 72 }
}
```



Encabezamiento de Texto

Text headword.

```
% L. v. Beethoven, Op. 110
% Piano sonata 31
% measures 1 - 7

\include "english.ly"

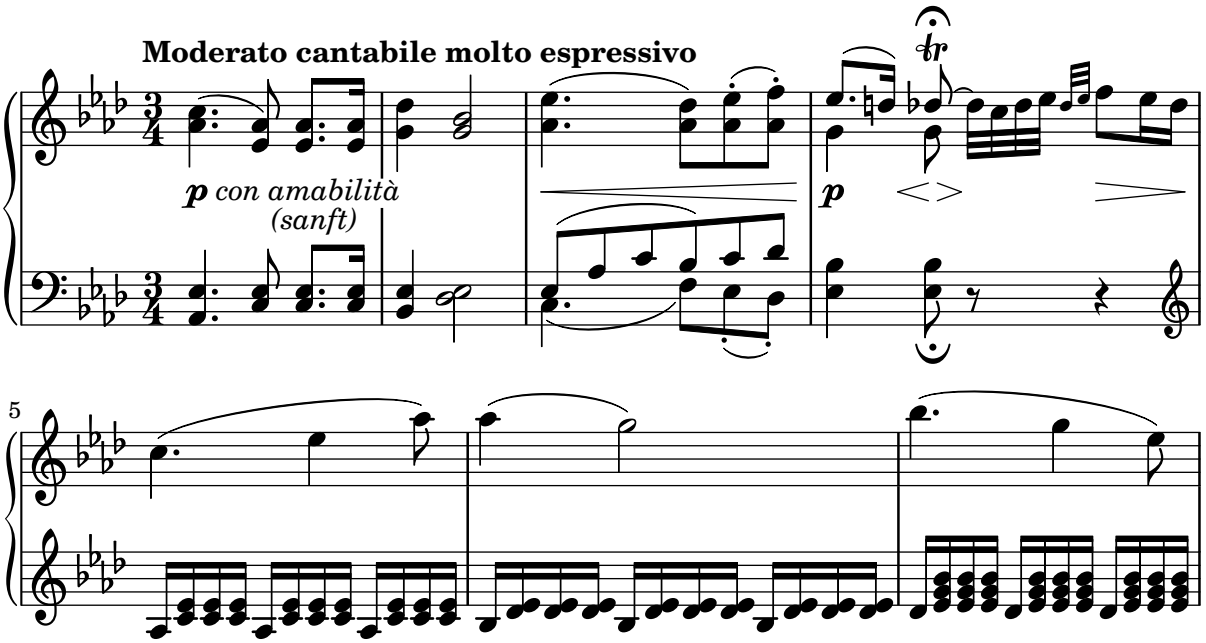
\new PianoStaff <<
  \new Staff = "right hand" {
    \clef treble
    \key af \major
    \time 3/4
    \tempo "Moderato cantabile molto espressivo"
```

```
    <c'' af'>4.( <af' ef'>8 ) q8.[ q16] |
    <df'' g'>4 <bf' g'>2 |
    <af' ef''>4.( <af' df''>8[) <af' ef''>-.( <af' f''>-.)] |
    << { ef''8.[( d''16]) df''8\trill\fermata ~
           \oneVoice df''32[ c'' df'' ef''] }\\
       { g'4 g'8 s } >>
     \grace { df''32[ ef''] } f''8[ ef''16 df''] |
%
% 5
%
    c''4.( ef''4 af''8) |
    af''4( g''2) |
    bf''4.( g''4 ef''8) |
  }

  \new Dynamics {
    s2.-\tweak padding #-1
       -\tweak baseline-skip #0
       -\markup \center-column {
          \whiteout \line { \dynamic p \italic { con amabilità } }
          \line { \hspace #3 (sanft) } } |
    s2. |
    s2.\< |
    s8..\p s32\< s16..\> s64\! s8 s4\> |
%
% 5
%
    s2.*3\! |
  }

  \new Staff = "left hand" {
    \clef bass
    \key af \major
    \time 3/4

    <af, ef>4. \stemUp <c ef>8 q8.[ q16] \stemNeutral |
    <bf, ef>4 <df ef>2 |
    << { ef8[( af c' bf) c' df'] } \\
       { c4.( f8[) ef8-.( df-.]) } >> |
    <ef bf>4 q8_\fermata r r4 | \clef treble
%
% 5
%
    af16[ <c' ef'> q q] af[ <c' ef'> q q] af[ <c' ef'> q q] |
    bf16[ <df' ef'> q q] bf[ <df' ef'> q q] bf[ <df' ef'> q q] |
    df'16[ <ef' g' bf'> q q] df'[ <ef' g' bf'> q q]
      df'[ <ef' g' bf'> q q] |
  }
>>
```



Encabezamiento de Instrumentos sin trastes

Encabezamiento de Instrumentos sin trastes.

```
% David Séverin
% Les cinq pieds (2007)
% for violon solo
% (this extract is in the public domain)


% Abbreviations
db = \markup { \musicglyph "scripts.udownbow" }
dub = \markup { \musicglyph "scripts.udownbow" " "
                \musicglyph "scripts.uupbow" }
dubetc = \markup { \musicglyph "scripts.udownbow" " "
                   \musicglyph "scripts.uupbow" "..." }

ub = \markup { \musicglyph "scripts.uupbow" }
udb = \markup { \musicglyph "scripts.uupbow" " "
                \musicglyph "scripts.udownbow" }
udbetc = \markup { \musicglyph "scripts.uupbow" " "
                   \musicglyph "scripts.udownbow" "..." }

accel = \markup \tiny \italic \bold "accel..."
ritar = \markup \tiny \italic \bold "ritar..."

% Strings
svib = \markup \small "s. vib."
pvib = \markup \small "p. vib."
mvib = \markup \small "m. vib."
sulp = \markup \small "s.p."
norm = \markup \small "n."

quatre = \markup \teeny "IV"
```

```
% Shifting Notes
shift = \once \override NoteColumn.force-hshift = 0.9
shifta = \once \override NoteColumn.force-hshift = 1.2

% Hairpin
aniente = \once \override Hairpin.circled-tip = ##t

% Tuplets
tupletbp = \once \override Staff.TupletBracket.padding = 2.25

% Functions
#(define-markup-command (colmark layout props args) (markup-list?)
   (let ((entries (cons (list '(baseline-skip . 2.3)) props)))
     (interpret-markup layout entries
                       (make-column-markup (map (lambda (arg)
                                                  (markup arg))
                                                (reverse args))))))
% Instruments
ViolinSolo = \relative c' {
  \set Score.rehearsalMarkFormatter = #format-mark-box-numbers
  \override Score.VoltaBracket.font-family = #'sans
  \override Score.VoltaBracket.extra-offset = #'(0 . 1)
  \override SpacingSpanner.uniform-stretching = ##t

  \voiceOne

  % Measure 1
  \time 25/8 \mark \default
  r2^\markup \colmark { \italic "fatigué" " " \bold "lentement"} r4 r r8
  <<
    { \shift d2\glissando^\markup \colmark { \quatre \dubetc \svib }
      \shifta e1 } \\
    { d2\open\mf\< ~ \aniente d1\!\>
      r4 r\!^\markup \colmark { " " \fermata } }
  >>

  % Measure 2
  \time 7/4
  \set Score.repeatCommands = #'((volta "1) n.      2) s.p."))
  <<
    { \shift d2\glissando^\markup \colmark { \quatre \udbetc }
      \shifta e1 } \\
    { d2\open\mf\< ~ d1\!\> ~ d4\!^\markup \colmark { " " \fermata } }
  >>
  \set Score.repeatCommands = #'((volta #f))

  % Measure 3
  \time 15/4
  <<
    { \shift d2\glissando^\markup \colmark { \quatre \dubetc \pvib \norm }
      \shifta e1\glissando d2 } \\
```

```
    { d2\open\mf\< ~ d1 ~ d2\ff ~ d1\> ~
      d2^\markup \colmark { " " " " \svib } ~ d4\pp }
  >>
  \break

  % Measure 4
  \time 4/4 \stemUp \tupletDown
  \tuplet 3/2 { d4 ^\markup \colmark { \quatre \db \accel } d d }
  \tuplet 3/2 { d4 ^\markup \colmark { " " \db \sulp } d d }

  % Measure 5
  \time 5/4
  \tupletbp \tuplet 3/2 {
    d8\mf\<^\markup \colmark { \quatre \db \norm } d_\open d }
  \tupletbp \tuplet 3/2 {
    d8^\markup \colmark { " " \db \sulp } d_\open d }
  \tupletbp \tuplet 3/2 {
    d16^\markup \colmark { " " \db \norm } d_\open d d d_\open d }
  d2\ff\>^\markup \colmark { " " \pvib }

  % Measure 6
  \time 5/8
  \once \override Beam.grow-direction = #RIGHT % \featherDurations 2/3
  d16-.[ d-. d-. d-. d-. d-. d-. d-. d-. d-.]
  \break

  % Measure 7
  \time 7/4
  \tupletbp \tuplet 3/2 {
    d16^\markup \colmark { \quatre } d_\open d d d_\open d }
  \tupletbp \tuplet 3/2 {
    d8^\markup \colmark { " " \db } d_\open d }
  \tupletbp \tuplet 3/2 {
    d8^\markup \colmark { " " \db " " \sulp } d_\open d }
  \tuplet 3/2 { d4^\markup \colmark { \quatre \db \ritar \norm } d d }
  \tuplet 3/2 { d4^\markup \colmark { " " \db " " \sulp } d d\ppp ~ }

  % Measure 8
  d4^\markup \colmark { " " " " \pvib \norm } deh2 d dih \<

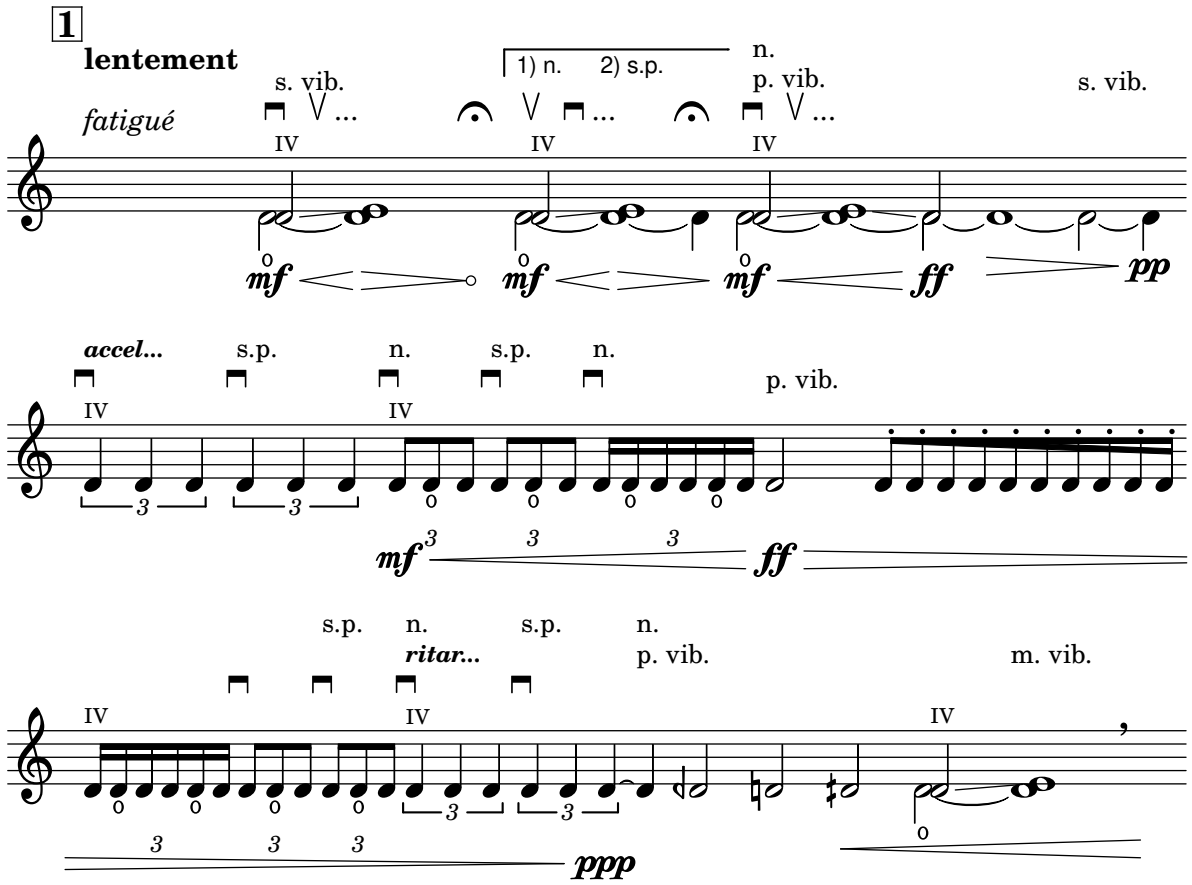
  % Measure 9
  <<
    { \shift d2\glissando^\markup \colmark { \quatre } \shifta e1 } \\
    { d2\open ~ d1^\markup \colmark { " " " " \mvib } }
  >>
  \breathe r4\!
}

\score {
  <<
    \new Staff \relative c' \ViolinSolo
    \hide Score.Rest
```

```
    \set Score.measureBarType = ""
  >>

  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}

\paper {
  system-system-spacing.padding = 5
}
```



Encabezamiento de música vocal

Encabezamiento de Música vocal.

```
% L. van Beethoven, op. 125
% Symphony No. 9 in D minor
% Finale measures 216 - 236
% Text: L. van Beethoven (introduction), F. von Schiller ("Ode to Joy")

\score {
  \new Staff \relative c' {
    \override Score.BarNumber.self-alignment-X = #LEFT
    \set Score.currentBarNumber = 216
    \set Score.barNumberVisibility = #all-bar-numbers-visible
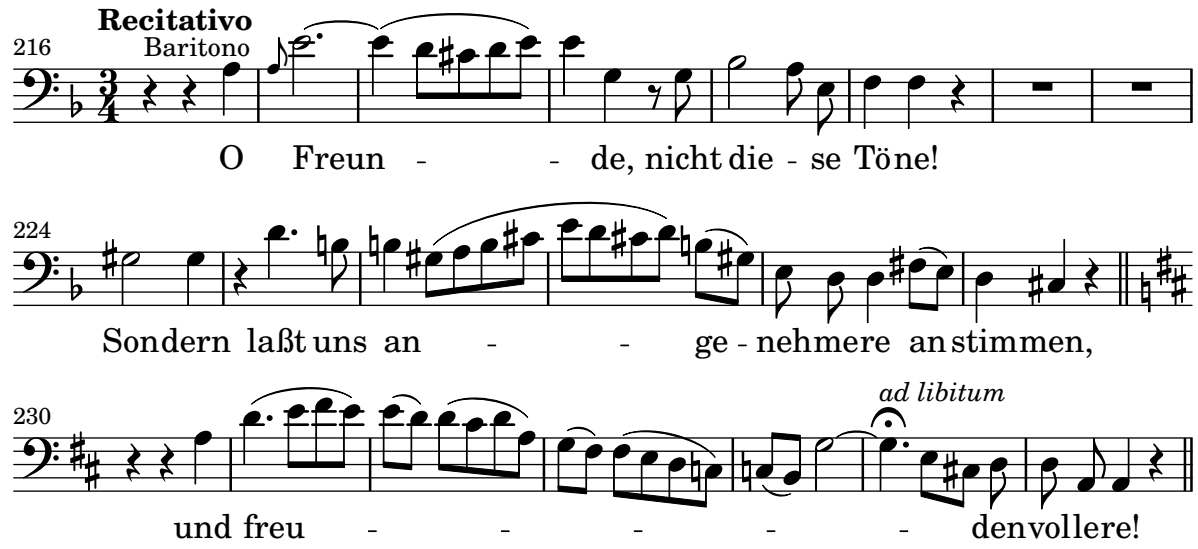
    \autoBeamOff
    \clef bass \key d \minor \time 3/4
    \tempo "Recitativo"

    r4^\markup { \small Baritono } r a |
    \grace a8 e'2. ~ |
    e4( d8[ cis d e]) |
    e4 g, r8 g |
    bes2 a8 e |
    f4 f r |
    R2.*2 |

    gis2 gis4 |
    r4 d'4. b8 |
    b4 gis8\tweak height-limit #4 ([ a b cis] |
    e8[ d cis d)] b([ gis)] |
    e8 d d4 fis8([ e)] |
    d4 cis r \bar "||"

    \key d \major
    r4 r a' |
    d4.( e8[ fis e)] |
    e([ d)] d([ cis d a)] |
    g8([ fis)] fis([ e d c)] |
    c8([ b)] g'2~ |
    % put fermata closer to staff
    \once \override Script.outside-staff-priority = #1
    g4.\fermata ^\markup { \small \italic "ad libitum" } e8[ cis!] d |
    d8 a a4 r \bar "||"
  }

  \addlyrics {
    O Freun -- _ de, nicht die -- _ se Tö -- ne!
    Son -- dern laßt uns an -- _ ge -- neh -- me -- re an -- stim -- men,
    und freu -- _ _ _ _ _ _ _ den -- vol -- le -- re!
  }
}
```



Encabezamiento de instrumentos de viento

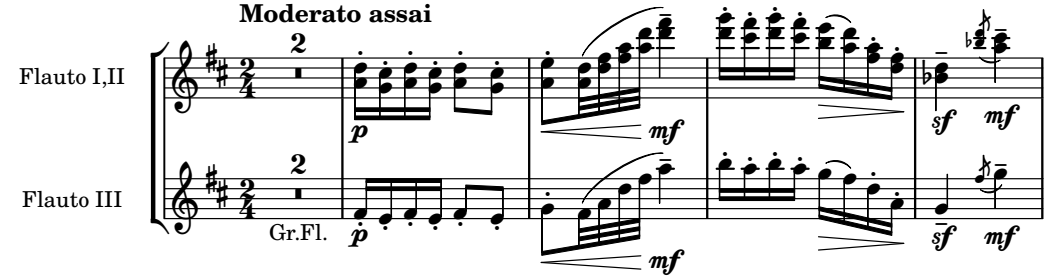
Encabezamiento de Instrumentos de viento.

```
% Tchaikovsky op. 71a
% The Nutcracker (suite)
% VII Dance of the Merlitons

#(set-global-staff-size 15)

\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Flauto I,II" }
    \relative c'' {
      \tweak padding 3.5 \tweak font-size 1 \tempo "Moderato assai"
      \key d \major
      \time 2/4
      \compressMMRests R2*2
      <d a>16-.\p <cis g>-. <d a>-. <cis g>-. <d a>8-. <cis g>-.
      <e a,>-.\< <d a>32( <fis d> <a fis> <d a> <fis d>4--)\mf
      <g d>16-. <fis cis>-. <g d>-. <fis cis>-.
      <e b>(\> <d a>) <a fis>-. <fis d>-.\!
      <d bes>4--\sf \acciaccatura {<d' bes>8} <cis a>4--\mf
    }
    \new Staff \with { instrumentName = "Flauto III" }
    \relative c' {
      \key d \major
      \time 2/4
      \compressMMRests R2*2_\markup{Gr.Fl.}
      fis16-.\p e-. fis-. e-. fis8-. e-.
      g8-.\< fis32( a d fis a4--)\mf
      b16-. a-. b-. a-. g(\> fis) d-. a-.\!
      g4--\sf \acciaccatura fis'8 g4--\mf
    }
  >>
}
```

```
\layout {
  indent = 2\cm
}
```



25 MIDI
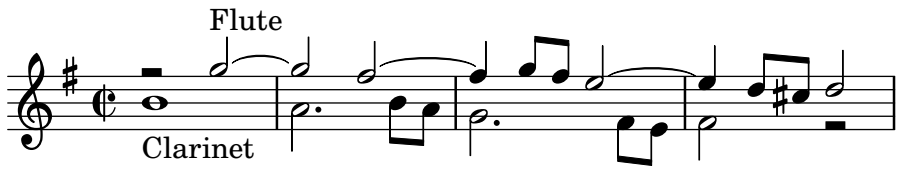
See also Sección “Creating MIDI output” en *Notation Reference*.

Modificar la salida MIDI para que tenga un canal por cada voz

Al producir una salida MIDI, el comportamiento predeterminado es que cada pentagrama representa un canal MIDI, con todas las voces de dicho pentagrama mezcladas. Esto reduce al mínimo el riesgo de que se agote el número de canales MIDI disponibles, pues existe un máximo de 16 canales por cada puerto MIDI, y la mayoría de los dispositivos solo tiene un puerto.

Sin embargo, cuando se traslada el interpretador `Staff_performer` al contexto `Voice`, cada voz de un pentagrama puede tener su propio canal MIDI, como se muestra en el siguiente ejemplo: a pesar de estar sobre el mismo pentagrama, se crean dos canales MIDI, cada uno con un `midiInstrument` distinto.
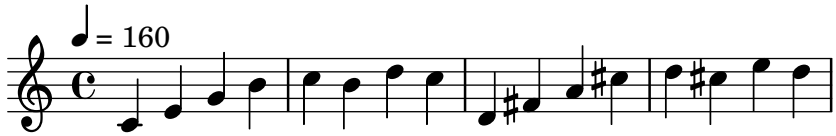
```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = "flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = "clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}
```



Cambiar el tempo sin indicación metronómica

Para cambiar el tempo en la salida MIDI sin imprimir nada, hacemos invisible la indicación metronómica:

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



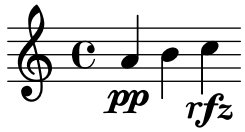
Creacaión de matices dinámicos personalizados en el MIDI

El ejemplo siguiente muestra cómo crear una marda de matiz dinámico, no incluida en la lista predeterminada, y asignarle un valor específico de forma que afecte a la salida MIDI.

El matiz \rfz conlleva el valor 0.9.

```
#(define (myDynamics dynamic)
    (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = "cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative {
        a'4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}
```



Demo of MIDI instruments

Problem: How to know which `midiInstrument` values would be best for your composition?

Solution: A LilyPond demo file. You have to compile this snippet by yourself and listen to the created MIDI output file.

```
melody = \relative c' {
  \tempo 4 = 150
  c4.\mf g c16 b' c d
  e16 d e f g4 g'4 r
  R1
}

\score {
  \new Voice \melody
  \layout { }
}

\score {
  \new Voice {
    r\mf
    % 1-8 keyboard
    \set Staff.midiInstrument = "acoustic grand" \melody
    \set Staff.midiInstrument = "bright acoustic" \melody
    \set Staff.midiInstrument = "electric grand" \melody
    \set Staff.midiInstrument = "honky-tonk" \melody
    \set Staff.midiInstrument = "electric piano 1" \melody
    \set Staff.midiInstrument = "electric piano 2" \melody
    \set Staff.midiInstrument = "harpsichord" \melody
    \set Staff.midiInstrument = "clav" \melody

    % 9-16 chrom percussion
    \set Staff.midiInstrument = "celesta" \melody
    \set Staff.midiInstrument = "glockenspiel" \melody
    \set Staff.midiInstrument = "music box" \melody
    \set Staff.midiInstrument = "vibraphone" \melody
    \set Staff.midiInstrument = "marimba" \melody
    \set Staff.midiInstrument = "xylophone" \melody
    \set Staff.midiInstrument = "tubular bells" \melody
    \set Staff.midiInstrument = "dulcimer" \melody

    % 17-24 organ
    \set Staff.midiInstrument = "drawbar organ" \melody
    \set Staff.midiInstrument = "percussive organ" \melody
    \set Staff.midiInstrument = "rock organ" \melody
    \set Staff.midiInstrument = "church organ" \melody
    \set Staff.midiInstrument = "reed organ" \melody
    \set Staff.midiInstrument = "accordion" \melody
```

```
\set Staff.midiInstrument = "harmonica" \melody
\set Staff.midiInstrument = "concertina" \melody

% 25-32 guitar
\set Staff.midiInstrument = "acoustic guitar (nylon)" \melody
\set Staff.midiInstrument = "acoustic guitar (steel)" \melody
\set Staff.midiInstrument = "electric guitar (jazz)" \melody
\set Staff.midiInstrument = "electric guitar (clean)" \melody
\set Staff.midiInstrument = "electric guitar (muted)" \melody
\set Staff.midiInstrument = "overdriven guitar" \melody
\set Staff.midiInstrument = "distorted guitar" \melody
\set Staff.midiInstrument = "guitar harmonics" \melody

% 33-40 bass
\set Staff.midiInstrument = "acoustic bass" \melody
\set Staff.midiInstrument = "electric bass (finger)" \melody
\set Staff.midiInstrument = "electric bass (pick)" \melody
\set Staff.midiInstrument = "fretless bass" \melody
\set Staff.midiInstrument = "slap bass 1" \melody
\set Staff.midiInstrument = "slap bass 2" \melody
\set Staff.midiInstrument = "synth bass 1" \melody
\set Staff.midiInstrument = "synth bass 2" \melody

% 41-48 strings
\set Staff.midiInstrument = "violin" \melody
\set Staff.midiInstrument = "viola" \melody
\set Staff.midiInstrument = "cello" \melody
\set Staff.midiInstrument = "contrabass" \melody
\set Staff.midiInstrument = "tremolo strings" \melody
\set Staff.midiInstrument = "pizzicato strings" \melody
\set Staff.midiInstrument = "orchestral harp" \melody
\set Staff.midiInstrument = "timpani" \melody

% 49-56 ensemble
\set Staff.midiInstrument = "string ensemble 1" \melody
\set Staff.midiInstrument = "string ensemble 2" \melody
\set Staff.midiInstrument = "synthstrings 1" \melody
\set Staff.midiInstrument = "synthstrings 2" \melody
\set Staff.midiInstrument = "choir aahs" \melody
\set Staff.midiInstrument = "voice oohs" \melody
\set Staff.midiInstrument = "synth voice" \melody
\set Staff.midiInstrument = "orchestra hit" \melody

% 57-64 brass
\set Staff.midiInstrument = "trumpet" \melody
\set Staff.midiInstrument = "trombone" \melody
\set Staff.midiInstrument = "tuba" \melody
\set Staff.midiInstrument = "muted trumpet" \melody
\set Staff.midiInstrument = "french horn" \melody
\set Staff.midiInstrument = "brass section" \melody
\set Staff.midiInstrument = "synthbrass 1" \melody
\set Staff.midiInstrument = "synthbrass 2" \melody
```

```
% 65-72 reed
\set Staff.midiInstrument = "soprano sax" \melody
\set Staff.midiInstrument = "alto sax" \melody
\set Staff.midiInstrument = "tenor sax" \melody
\set Staff.midiInstrument = "baritone sax" \melody
\set Staff.midiInstrument = "oboe" \melody
\set Staff.midiInstrument = "english horn" \melody
\set Staff.midiInstrument = "bassoon" \melody
\set Staff.midiInstrument = "clarinet" \melody

% 73-80 pipe
\set Staff.midiInstrument = "piccolo" \melody
\set Staff.midiInstrument = "flute" \melody
\set Staff.midiInstrument = "recorder" \melody
\set Staff.midiInstrument = "pan flute" \melody
\set Staff.midiInstrument = "blown bottle" \melody
\set Staff.midiInstrument = "shakuhachi" \melody
\set Staff.midiInstrument = "whistle" \melody
\set Staff.midiInstrument = "ocarina" \melody

% 81-88 synth lead
\set Staff.midiInstrument = "lead 1 (square)" \melody
\set Staff.midiInstrument = "lead 2 (sawtooth)" \melody
\set Staff.midiInstrument = "lead 3 (calliope)" \melody
\set Staff.midiInstrument = "lead 4 (chiff)" \melody
\set Staff.midiInstrument = "lead 5 (charang)" \melody
\set Staff.midiInstrument = "lead 6 (voice)" \melody
\set Staff.midiInstrument = "lead 7 (fifths)" \melody
\set Staff.midiInstrument = "lead 8 (bass+lead)" \melody

% 89-96 synth pad
\set Staff.midiInstrument = "pad 1 (new age)" \melody
\set Staff.midiInstrument = "pad 2 (warm)" \melody
\set Staff.midiInstrument = "pad 3 (polysynth)" \melody
\set Staff.midiInstrument = "pad 4 (choir)" \melody
\set Staff.midiInstrument = "pad 5 (bowed)" \melody
\set Staff.midiInstrument = "pad 6 (metallic)" \melody
\set Staff.midiInstrument = "pad 7 (halo)" \melody
\set Staff.midiInstrument = "pad 8 (sweep)" \melody

% 97-104 synth effects
\set Staff.midiInstrument = "fx 1 (rain)" \melody
\set Staff.midiInstrument = "fx 2 (soundtrack)" \melody
\set Staff.midiInstrument = "fx 3 (crystal)" \melody
\set Staff.midiInstrument = "fx 4 (atmosphere)" \melody
\set Staff.midiInstrument = "fx 5 (brightness)" \melody
\set Staff.midiInstrument = "fx 6 (goblins)" \melody
\set Staff.midiInstrument = "fx 7 (echoes)" \melody
\set Staff.midiInstrument = "fx 8 (sci-fi)" \melody

% 105-112 ethnic
```

```
    \set Staff.midiInstrument = "sitar" \melody
    \set Staff.midiInstrument = "banjo" \melody
    \set Staff.midiInstrument = "shamisen" \melody
    \set Staff.midiInstrument = "koto" \melody
    \set Staff.midiInstrument = "kalimba" \melody
    \set Staff.midiInstrument = "bagpipe" \melody
    \set Staff.midiInstrument = "fiddle" \melody
    \set Staff.midiInstrument = "shanai" \melody

    % 113-120 percussive
    \set Staff.midiInstrument = "tinkle bell" \melody
    \set Staff.midiInstrument = "agogo" \melody
    \set Staff.midiInstrument = "steel drums" \melody
    \set Staff.midiInstrument = "woodblock" \melody
    \set Staff.midiInstrument = "taiko drum" \melody
    \set Staff.midiInstrument = "melodic tom" \melody
    \set Staff.midiInstrument = "synth drum" \melody
    \set Staff.midiInstrument = "reverse cymbal" \melody

    % 121-128 sound effects
    \set Staff.midiInstrument = "guitar fret noise" \melody
    \set Staff.midiInstrument = "breath noise" \melody
    \set Staff.midiInstrument = "seashore" \melody
    \set Staff.midiInstrument = "bird tweet" \melody
    \set Staff.midiInstrument = "telephone ring" \melody
    \set Staff.midiInstrument = "helicopter" \melody
    \set Staff.midiInstrument = "applause" \melody
    \set Staff.midiInstrument = "gunshot" \melody
  }
  \midi { }
}
```



Reemplazo de la ecualización predeterminada de instrumentos MIDI

El ecualizador predeterminado de MIDI se puede sustituir fijando la propiedad `instrumentEqualizer` en el contexto `Score` a un procedimiento de Scheme definido por el usuario que usa un nombre de instrumento MIDI como argumento junto a un par de fracciones que indican los vloúmenes mínimo y máximo, respectivamente, que aplicar a ese instrumento.

El ejemplo siguiente fija los volúmenes mínimo y máximo de la flauta y el clarinete.
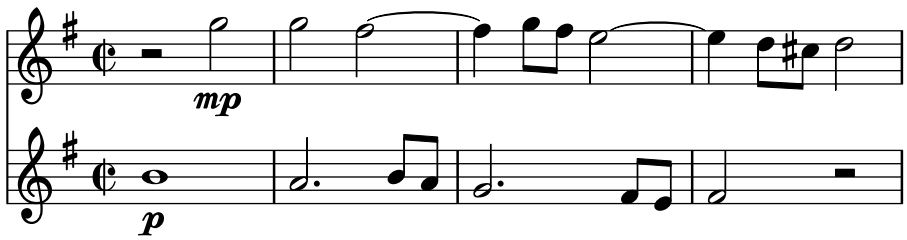
```
#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
```

```
    my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = "flute"
      \new Voice \relative {
        r2 g''\mp g fis~
        4 g8 fis e2~
        4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = "clarinet"
      \new Voice \relative {
        b'1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {  }
}
```



26 Non-music

Alinear y centrar los nombres de instrumento

La alineación horizontal de los nombres de instrumento se puede trucar modificando la propiedad `self-alignment-X` del objeto gráfico `InstrumentName` (normalmente dentro del contexto `Staff`). Las variables de `\layout` `indent` y `short-indent` definen el espacio en que se alinean los nombres de instrumento antes del primer sistema y de los siguientes, respectivamente.

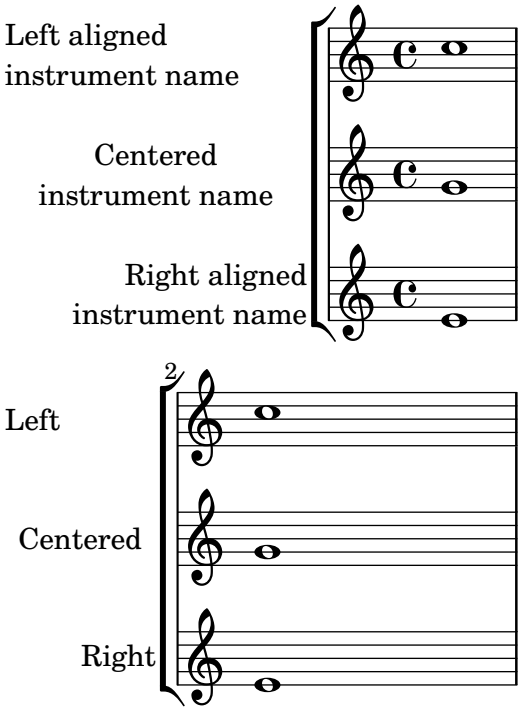
```
\paper {
  left-margin = 3\cm
}

\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                            "instrument name" }
    shortInstrumentName = "Left"
  } {
    c''1 \break c''1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #CENTER
    instrumentName = \markup \center-column { Centered
                                              "instrument name" }
    shortInstrumentName = "Centered"
  } {
    g'1 g'1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #RIGHT
    instrumentName = \markup \right-column { "Right aligned"
                                             "instrument name" }
    shortInstrumentName = "Right"
  } {
    e'1 e'1
  }
>>

\layout {
  indent = 4\cm
  short-indent = 2\cm
  line-width = 6.5\cm
}
```



Demonstrating all \header fields

A demonstration of all header fields that LilyPond defines by default. Thanks to setting `print-all-headers` to `#t`, much more fields as usual are displayed, indicating the hierarchy of `\header` blocks.

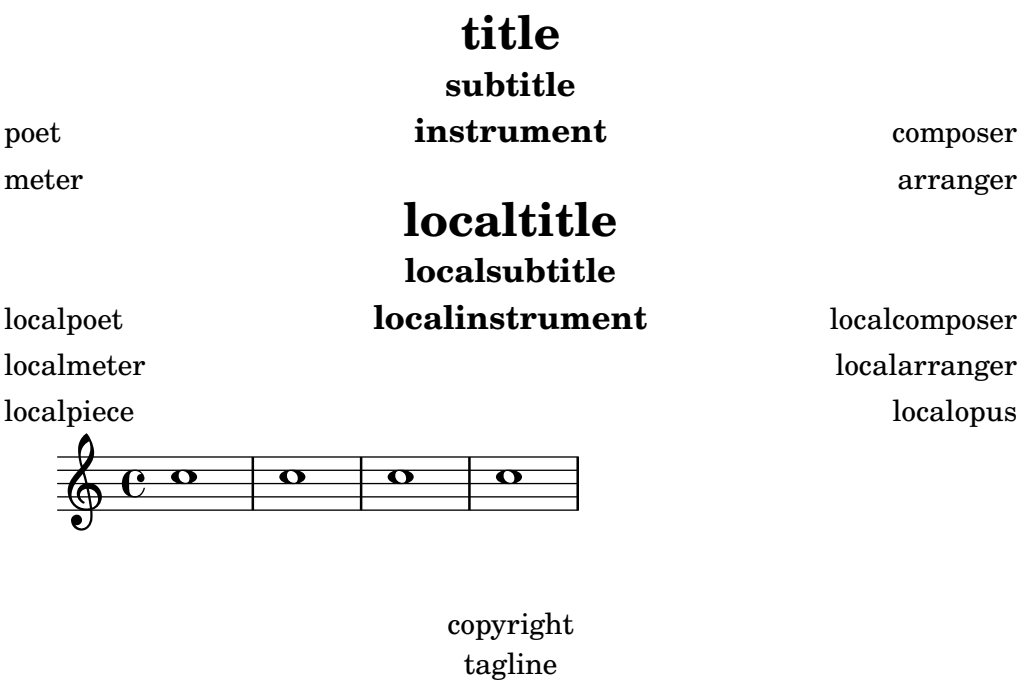
```
\paper {
  #(set-paper-size "a6" 'landscape)
  print-all-headers = ##t
}

\book {
  \header {
    title = "title"
    subtitle = "subtitle"
    composer = "composer"
    arranger = "arranger"
    instrument = "instrument"
    meter = "meter"
    opus = "opus"
    piece = "piece"
    poet = "poet"
    copyright = "copyright"
    tagline = "tagline"
  }

  \bookpart {
    \score {
      \relative c'' { c1 | c | c | c }

      \header {
        title = "localtitle"
```

```
        subtitle = "localsubtitle"
        composer = "localcomposer"
        arranger = "localarranger"
        instrument = "localinstrument"
        meter = "localmeter"
        opus = "localopus"
        piece = "localpiece"
        poet = "localpoet"
        copyright = "localcopyright"
        tagline = "localtagline"
      }
    }
  }
}
```
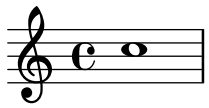


Listas de llaves para los diagramas de viento madera

El fragmento de código que aparece a continuación produce una lista de todas las llaves y disposiciones de llave posibles para los diagramas de posiciones de instrumentos de viento madera, tal y como están definidos en `scm/define-woodwind-diagrams.scm`. La lista se muestra en la salida de error estándar stderr, pero no en la música. Si desea una salida por la consola a través de la salida estándar stdout, omita el código `(current-error-port)` de las instrucciones.

```
#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
```

```
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))

\score {c''1}
```



27 Paper and layout

See also Sección “Spacing issues” en *Notation Reference*.

Alinear y centrar los nombres de instrumento

La alineación horizontal de los nombres de instrumento se puede trucar modificando la propiedad `self-alignment-X` del objeto gráfico `InstrumentName` (normalmente dentro del contexto `Staff`). Las variables de `\layout` `indent` y `short-indent` definen el espacio en que se alinean los nombres de instrumento antes del primer sistema y de los siguientes, respectivamente.

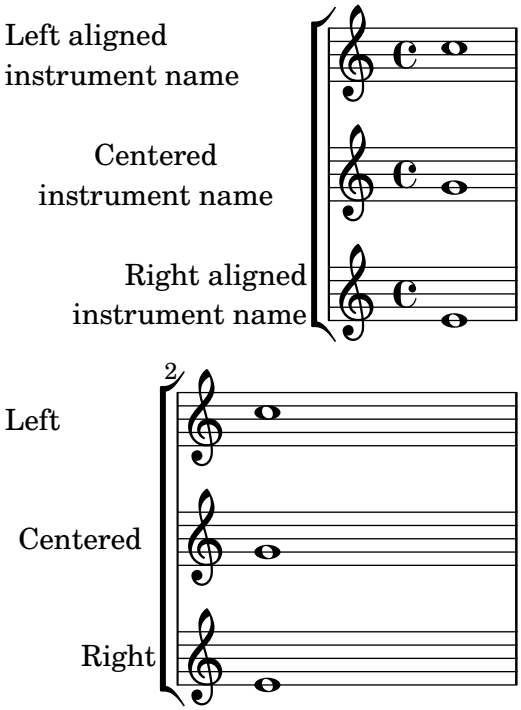
```
\paper {
  left-margin = 3\cm
}

\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                            "instrument name" }
    shortInstrumentName = "Left"
  } {
    c''1 \break c''1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #CENTER
    instrumentName = \markup \center-column { Centered
                                              "instrument name" }
    shortInstrumentName = "Centered"
  } {
    g'1 g'1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #RIGHT
    instrumentName = \markup \right-column { "Right aligned"
                                             "instrument name" }
    shortInstrumentName = "Right"
  } {
    e'1 e'1
  }
>>

\layout {
  indent = 4\cm
  short-indent = 2\cm
  line-width = 6.5\cm
}
```



Disposición de distintas letras en la misma línea

A veces queremos poner la letra que corresponde a distintos personajes sobre una sola línea, por ejemplo allí donde el texto alterna rápidamente. Este fragmento muestra cómo puede hacerse ajustando la propiedad `nonstaff-nonstaff-spacing` del objeto gráfico `VerticalAxisGroup`.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
              .nonstaff-nonstaff-spacing
              .minimum-distance = ##f
  }
}

aliceSings = \markup { \smallCaps "Alice" }
eveSings = \markup { \smallCaps "Eve" }

<<
  \new Staff <<
    \new Voice = "alice" {
      f'4^\aliceSings g' r2 |
      s1 |
      f'4^\aliceSings g' r2 |
      s1 | \break
      % ...
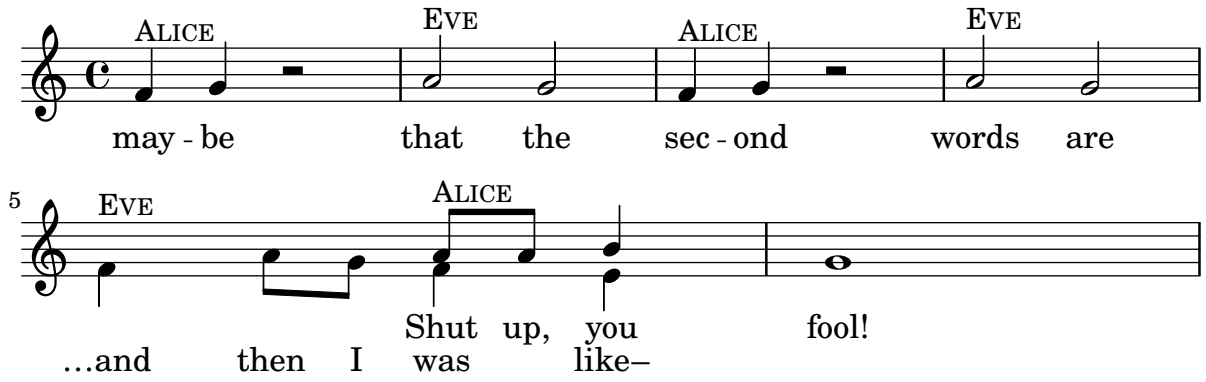
      \voiceOne
      s2 a'8^\aliceSings a' b'4 |
      \oneVoice
      g'1
    }
```

```
    \new Voice = "eve" {
      s1 |
      a'2^\eveSings g' |
      s1 |
      a'2^\eveSings g'
      % ...

      \voiceTwo
      f'4^\eveSings a'8 g' f'4 e' |
      \oneVoice
      s1
    }
  >>

  \new Lyrics \lyricsto "alice" {
    may -- be
    sec -- ond
    % ...
    Shut up, you fool!
  }

  \new Lyrics \lyricsto "eve" {
    that the
    words are
    % ...
    ...and then I was like--
  }
>>
```



Partes de libro

Se puede usar \bookpart para dividir un libro en varias partes. La última página de cada parte vùede quedar afectada por el valor de ragged-last-bottom. Los elementos de marcado de encabezamiento y pie de página pueden detectar que están en la última página de una parte, para diferenciarla de la última página del libro.

```
#(set-default-paper-size "a6")

\book {
  %% book paper, which is inherited by all children bookparts
  \paper {
    ragged-last-bottom = ##t
```

```
    %% Page footer: add a different part-tagline at part last page
    oddFooterMarkup = \markup {
      \column {
        \fill-line {
          %% Copyright header field only on book first page.
          \if \on-first-page \fromproperty #'header:copyright
        }
        \fill-line {
          %% Part tagline header field only on each part last page.
          \if \on-last-page-of-part \fromproperty #'header:parttagline
        }
        \fill-line {
          %% Tagline header field only on book last page.
          \if \on-last-page \fromproperty #'header:tagline
        }
      }
    }
  }

  %% book header, which is inherited by the first bookpart
  \header {
    title = "Book title"
    copyright = "Copyright line on book first page"
    parttagline = "Part tagline"
    tagline = "Book tagline"
  }

  \bookpart {
    %% a different page breaking function may be used on each part
    \paper { page-breaking = #ly:minimal-breaking }
    \header { subtitle = "First part" }
    \markup { The first book part }
    \markup { a page break }
    \pageBreak
    \markup { first part last page }
    \markup \wordwrap {
      with ragged-last-bottom (see the space below this text) }
  }

  \bookpart {
    \header { subtitle = "Second part" }
    { c'4 }
  }

}
```

Book title

First part

The first book part
a page break

Copyright line on book first page

2

first part last page
with ragged-last-bottom (see the space below this
text)

Part tagline

3

Book title

Second part



Part tagline

Book tagline

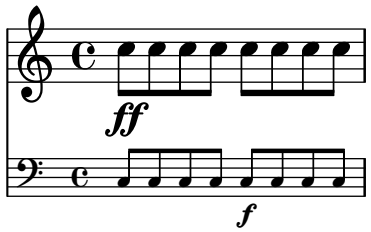
Modificar el tamaño de la pauta

La manera más sencilla de redimensionar los pentagramas es usar

```
#(set-global-staff-size tamaño)
```

El tamaño de una pauta individual se puede cambiar usando las propiedades `staff-space` y de `fontSize`.

```
<<
  \new Staff \relative c'' {
    \dynamicDown c8\ff c c c c c c c
  }
  \new Staff \with {
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
  } \relative c {
    \clef bass c8 c c c c\f c c c
  }
>>
```



Recorte de sistemas

Este código muestra cómo recortar (extraer) fragmentos a partir de una partitura completa.

Este archivo tiene que procesarse de forma separada con la opción `-dclip-systems`; la página de fragmentos de código podría no mostrar el resultado adecuadamente. La salida consiste en archivos con los nombres '*base*`-from-`*start*`-to-`*end*`[-`*count*`].eps'.

Si se incluyen los comienzos y finales de los sistemas, incluyen las dimensiones del grob del sistema, por ejemplo los nombres de instrumento.

Las notas de adorno en el punto final de la región no se incluyen.

Las regiones pueden abarcar varios sistemas. En este caso, se generan varios archivos EPS.

```
#(set-default-paper-size "a6" 'landscape)
\layout {
  indent = 2.4\cm
}

#(ly:set-option 'clip-systems)
#(ly:set-option 'separate-page-formats "ps")
#(define output-suffix "1")

origScore = \new Staff \with { instrumentName = "Instrument" }
\relative c' {
  c1
  d1
  \grace c16 e1
  \key d \major
  f1 \break
  \clef bass
  g,1
  fis1
}
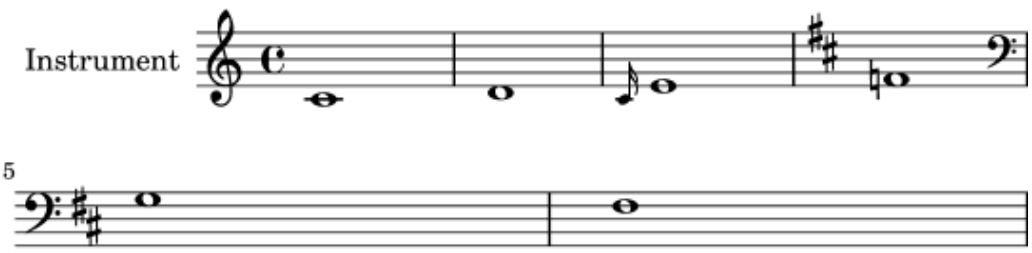
\book {
```

```
  \score {
    \origScore
    \layout {
      % Each clip-region is a (START . END) pair
      % where both are rhythmic locations.  Syntax:
      %
      %   (make-rhythmic-locations BAR-NUMBER NUM DEN)
      %
      % means NUM/DEN whole-notes into bar numbered BAR-NUMBER

      clip-regions = #(list (cons (make-rhythmic-location 2 0 1)
                                  (make-rhythmic-location 4 0 1))
                            (cons (make-rhythmic-location 0 0 1)
                                  (make-rhythmic-location 4 0 1))
                            (cons (make-rhythmic-location 0 0 1)
                                  (make-rhythmic-location 6 0 1)))
    }
  }
}

#(ly:set-option 'clip-systems #f)
#(ly:set-option 'separate-page-formats #f)
#(define output-suffix #f)

\book {
  \score { \origScore }
  \markup { \bold \fontsize #6 clips }
  \score {
    \lyrics {
      "from-2.0.1-to-4.0.1-clip.eps"
      \markup \epsfile #X #30.0
                       #(format #f "~a-1-from-2.0.1-to-4.0.1-clip.eps"
                                (ly:parser-output-name))
    }
  }
}
```



LilyPond v2.27.4

Crear pentagramas en blanco

Para crear pentagramas en blanco, genere compases vacíos y después elimine el grabador de números de compás `Bar_number_engraver` del contexto `Score`, y los grabadores de la indicación de compás `Time_signature_engraver`, de la clave `Clef_engraver` y de los compases `Bar_engraver` del contexto de `Staff`.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

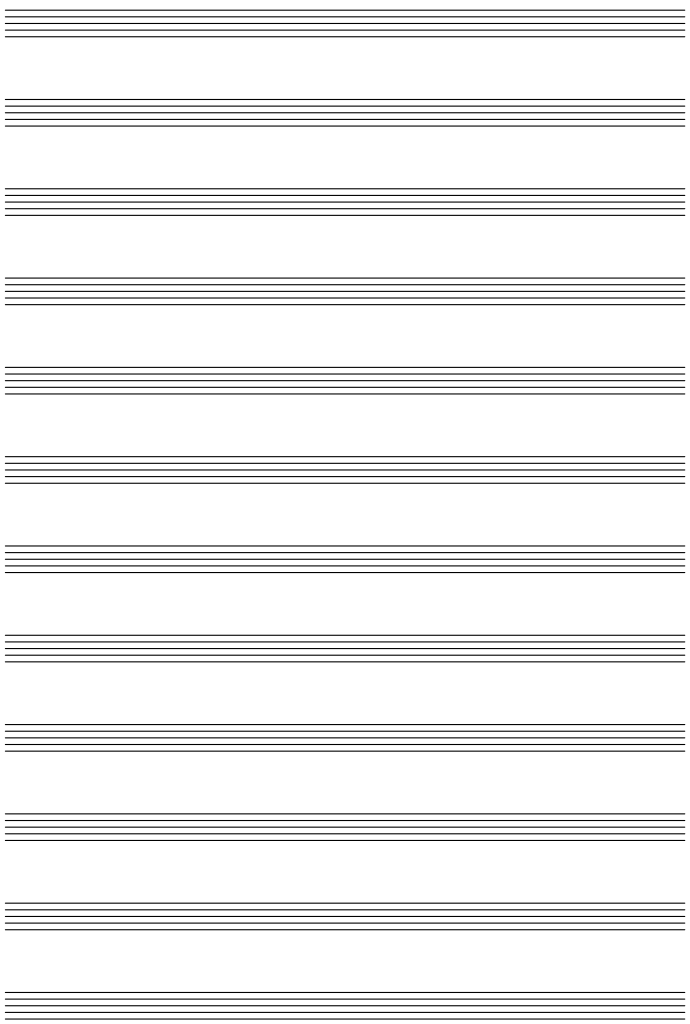
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % for the documentation
  \paper {
    #(set-paper-size "a6")
```

```
    ragged-last-bottom = ##f
    line-width = 90\mm
    left-margin = 7.5\mm
    bottom-margin = 5\mm
    top-margin = 5\mm
    tagline = ##f
  }

  % uncomment these lines for "letter" size
  %{
  \paper {
    #(set-paper-size "letter")
    ragged-last-bottom = ##f
    line-width = 7.5\in
    left-margin = 0.5\in
    bottom-margin = 0.25\in
    top-margin = 0.25\in
    tagline = ##f
   }
  %}

  % uncomment these lines for "A4" size
  %{
  \paper {
    #(set-paper-size "a4")
    ragged-last-bottom = ##f
    line-width = 180\mm
    left-margin = 15\mm
    bottom-margin = 10\mm
    top-margin = 10\mm
    tagline = ##f
  }
  %}
}
```



Demonstrating all \header fields

A demonstration of all header fields that LilyPond defines by default. Thanks to setting `print-all-headers` to `#t`, much more fields as usual are displayed, indicating the hierarchy of `\header` blocks.

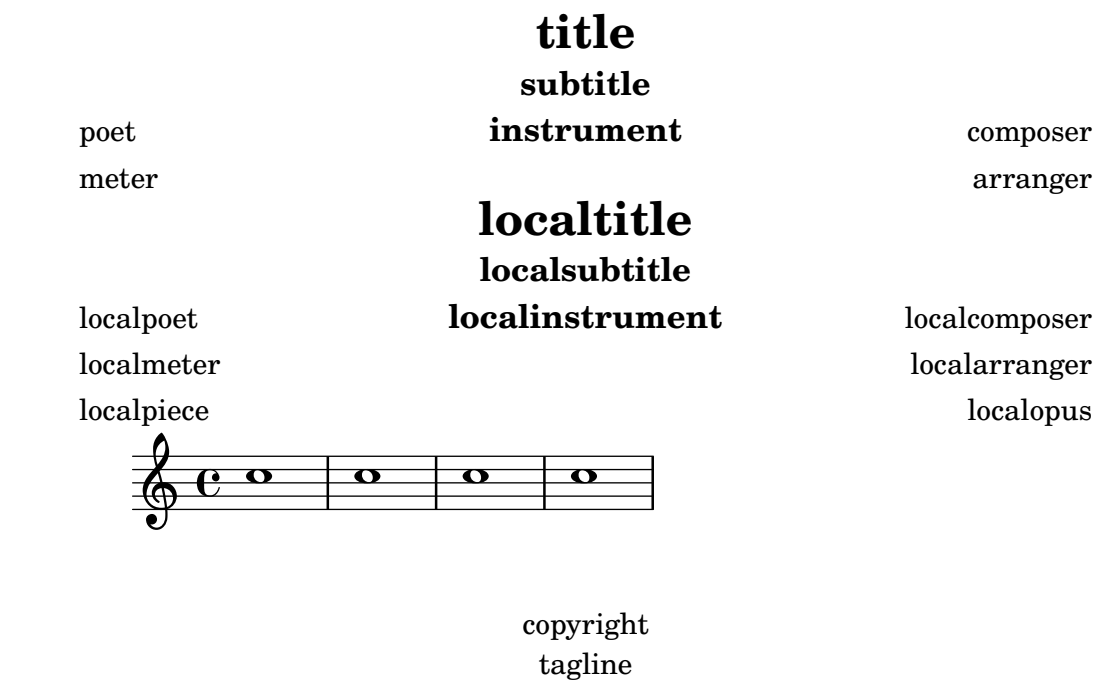
```
\paper {
  #(set-paper-size "a6" 'landscape)
  print-all-headers = ##t
}

\book {
  \header {
    title = "title"
    subtitle = "subtitle"
    composer = "composer"
    arranger = "arranger"
    instrument = "instrument"
    meter = "meter"
    opus = "opus"
    piece = "piece"
    poet = "poet"
    copyright = "copyright"
```

```
    tagline = "tagline"
  }

  \bookpart {
    \score {
      \relative c'' { c1 | c | c | c }

      \header {
        title = "localtitle"
        subtitle = "localsubtitle"
        composer = "localcomposer"
        arranger = "localarranger"
        instrument = "localinstrument"
        meter = "localmeter"
        opus = "localopus"
        piece = "localpiece"
        poet = "localpoet"
        copyright = "localcopyright"
        tagline = "localtagline"
      }
    }
  }
}
```



Mostrar un sistema GrandStaff completo si sigue con vida uno solo de sus pentagramas

En muchas partituras orquestales es costumbre no mostrar los pentagramas de aquellos instrumentos que están en silencio durante bastante tiempo; esto se llama 'partitura francesa'. LilyPond aporta esta funcionalidad a través de la instrucción `\RemoveEmptyStaves`.

Cuando vuelven a sonar, con frecuencia se prefiere mostrar *todos los instrumentos del grupo*. Esto se puede hacer añadiendo el grabador `Keep_alive_together_engraver` al contexto agrupador (p.ej.: un `GrandStaff` o un `StaffGroup`).

En este ejemplo, los violines están en silencio durante el segundo sistema. Solo el violín primero suena en el último compás del tercer sistema, pero se muestra también el pentagrama del violín segundo.

```
\score {
  <<
    \new Staff = "Staff_flute" \with {
      instrumentName = "Flute"
      shortInstrumentName = "Fl"
    } \relative c' {
      \*3 { c'4 c c c | c c c c | c c c c | \break }
    }
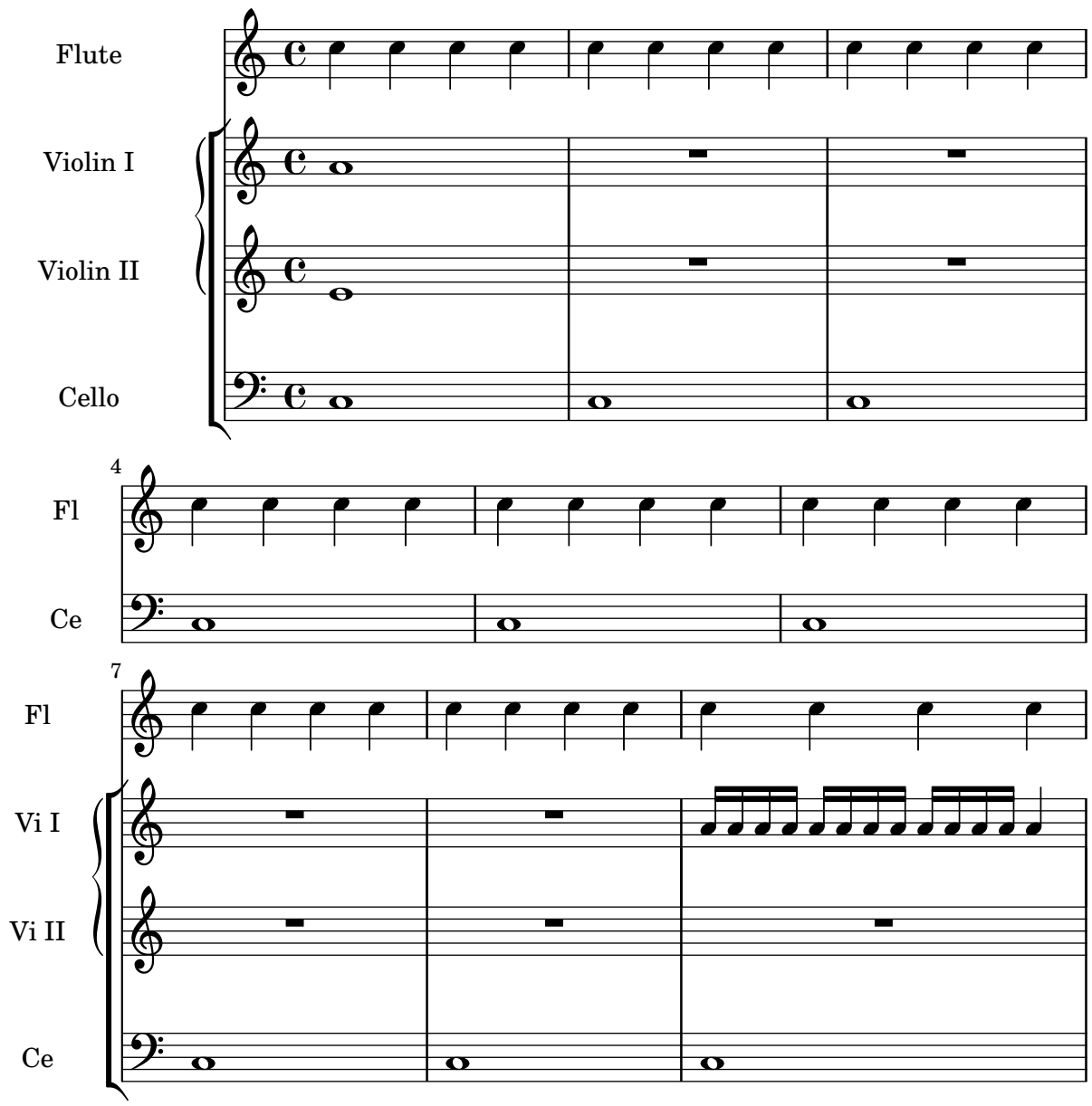
    \new StaffGroup = "StaffGroup_Strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
          shortInstrumentName = "Vi I"
        } \relative c'' {
          a1 | R1*7 | \*12 a16 a4 |
        }
        \new Staff = "StaffViolinII" \with {
          instrumentName = "Violin II"
          shortInstrumentName = "Vi II"
        } \relative c' {
          e1 | R1*8 |
        }
      >>

      \new Staff = "Staff_cello" \with {
        instrumentName = "Cello"
        shortInstrumentName = "Ce"
      } \relative c {
        \clef bass \*9 c1 |
      }
    >>
  >>
}

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm

  \context {
    \GrandStaff
    \consists Keep_alive_together_engraver
  }
  \context {
    \Staff
```

```
    \RemoveEmptyStaves
  }
}
```



Fijar un separador entre los sistemas

Se pueden insertar separadores sistema entre los sistemas de una página. Se puede usar cualquier elemento de marcado, pero `\slashSeparator` está disponible como una elección predeterminada adecuada.

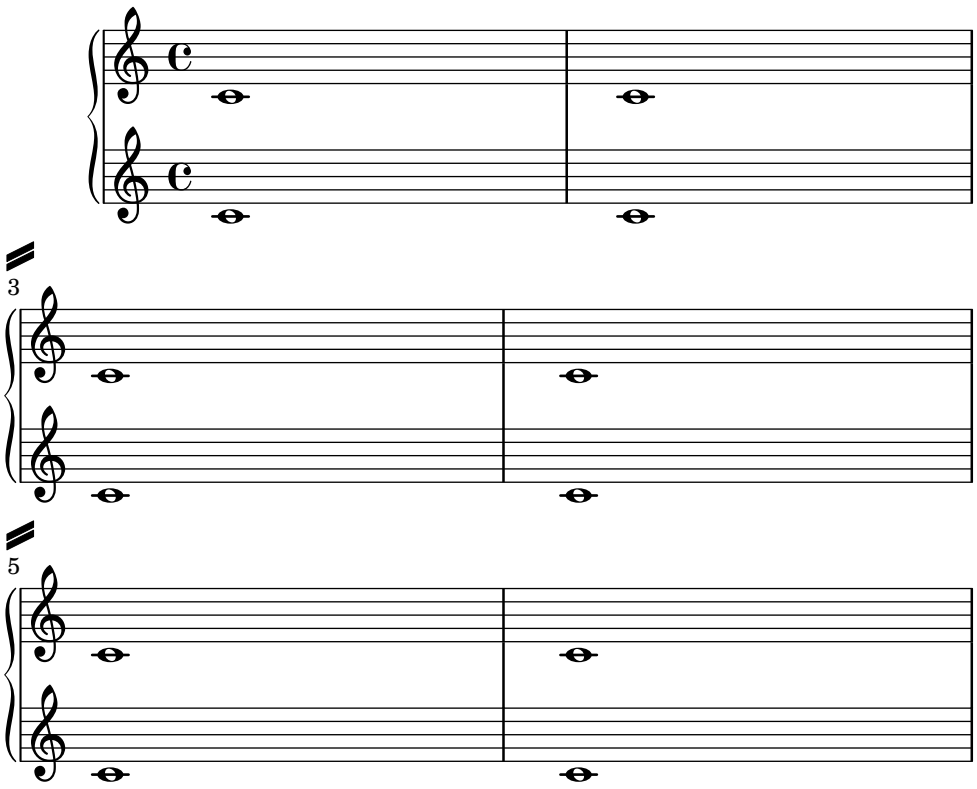
```
#(set-default-paper-size "a5")

\paper {
  system-separator-markup = \slashSeparator
  tagline = ##f
}

notes = \relative c' {
```

```
  c1 | c \break
  c1 | c \break
  c1 | c
}

\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}
```



Índice general (tabla de contenidos)

Se puede incluir un índice general (tabla de contenidos) mediante el uso de `\markuplist \table-of-contents`; sus elementos se añaden con la instrucción `\tocItem`.

```
#(set-default-paper-size "a7" 'landscape)
#(set-global-staff-size 11)

\paper {
  print-all-headers = ##t
}

\book {
  \markuplist \table-of-contents
  \pageBreak
```

```
  \tocItem \markup { The first score }
  \score {
    {
      c'1 \pageBreak
      \mark \default \tocItem \markup { Mark A }
      d'1
    }
    \header { title = "First score" }
  }
  \pageBreak

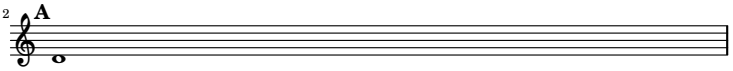
  \tocItem \markup { The second score }
  \score {
    { e'1 }
    \header { title = "Second score" }
  }
}
```

Table of Contents

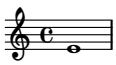
The first score **2**
Mark A **3**
The second score **4**

First score





Second score



Grupos de pentagramas alineados verticalmente sin barra inicial de conexión

Este fragmento muestra cómo conseguir StaffGroups alineados verticalmente con un SystemStartBar por cada StaffGroup, pero sin conectarlos entre sí.

Observe que esto solo funciona correctamente para música que quepa completamente dentro de un único sistema.

```
#(set-global-staff-size 15)

\paper {
  ragged-right = ##f
  print-all-headers = ##t
  tagline = ##f
}

\layout {
  indent = 0

  \context {
    \StaffGroup
    \consists Text_mark_engraver
    \consists Staff_collecting_engraver
    systemStartDelimiterHierarchy =
      #'(SystemStartBrace (SystemStartBracket a b))
  }

  \context {
    \Score
    \remove Text_mark_engraver
    \remove Staff_collecting_engraver
    \override SystemStartBrace.style = #'bar-line
    \omit SystemStartBar
    \override SystemStartBrace.padding = #-0.1
    \override SystemStartBrace.thickness = #1.6
    \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #15
  }
}

%%%% EXAMPLE

txt =
\lyricmode {
  Wer4 nur den lie -- ben Gott läßt wal2 -- ten4
  und4 hof -- fet auf ihn al -- le Zeit2.
}

% First StaffGroup "exercise"

eI = \relative c' {
  \textMark \markup {
    \bold Teacher:
    This is a simple setting of the choral. Please improve it. }
  \key a \minor
```

```
  \time 4/4
  \voiceOne

  \partial 4 e4
  a b c b
  a b gis2
  e4\fermata g! g f
  e a a gis
  a2.\fermata
  \bar ":|."
}

eII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo
  \partial 4 c4
  e e e gis
  a f e2
  b4 b d d
  c c d d
  c2.
  \bar ":|."
}

eIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a4
  c b a b
  c d b2
  gis4 g g b
  c a f e
  e2.
}

eIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a,4
  a' gis a e
  a, d e2
  e,4\fermata e' b g
  c f d e
  a,2.\fermata
  \bar ":|."
```

```
}

exercise = \new StaffGroup = "exercise" <<
  \new Staff <<
    \new Voice \eI
    \new Voice \eII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \eIII
    \new Voice \eIV
  >>
>>


% Second StaffGroup "simple Bach"

sbI = \relative c' {
  \textMark \markup { \bold" Pupil:" Here's my version! }
  \key a \minor
  \time 4/4
  \voiceOne

  \partial 4 e4
  a b c b
  a b gis2
  e4\fermata g! g f
  e a a gis
  a2.\fermata
  \bar ":|."
}

sbII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo
  \partial 4 c8 d
  e4 e e8 f g4
  f f e2
  b4 b8 c d4 d
  e8 d c4 b8 c d4
  c2.
  \bar ":|."
}

sbIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne
```

```
  \partial 4 a8 b
  c4 b a b8 c
  d4 d8 c b2
  gis4 g g8 a b4
  b a8 g f4 e
  e2.
}

sbIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a,4
  a' gis a e
  f8 e d4 e2
  e,4\fermata e' b a8 g
  c4 f8 e d4 e
  a,2.\fermata
  \bar ":|."
}

simpleBach = \new StaffGroup = "simple Bach" <<
  \new Staff <<
    \new Voice \sbI
    \new Voice \sbII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \sbIII
    \new Voice \sbIV
  >>
>>


% Third StaffGroup "chromatic Bach"

cbI = \relative c' {
  \textMark \markup {
    \bold "Teacher:"
    \column {
      "Well, you simply copied and transposed a version of J.S.Bach."
      "Do you know this one?"
    }
  }
  \key a \minor
  \time 4/4
  \voiceOne
```

```
  \partial 4 e4
  a b c b
  a b gis4. fis8
  e4\fermata g! g f
  e a a8 b gis4
  a2.\fermata
  \bar ":|."
}

cbII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo

  \partial 4 c8 d
  e4 e e8 fis gis4
  a8 g! f!4 e2
  b4 e e d
  d8[ cis] d dis e fis e4
  e2.
  \bar ":|."
}

cbIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a8 b
  c[ b] a gis8 a4 d,
  e8[ e'] d c b4. a8
  gis4 b c d8 c
  b[ a] a b c b b c16 d
  c2.
}

cbIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a4
  c, e a, b
  c d e2
  e4\fermata e a b8 c
  gis[ g] fis f e dis e4
  a,2.\fermata
  \bar ":|."
}
```

```
chromaticBach = \new StaffGroup = "chromatic Bach" <<
  \new Staff <<
    \new Voice \cbI
    \new Voice \cbII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \cbIII
    \new Voice \cbIV
  >>
>>


% Score

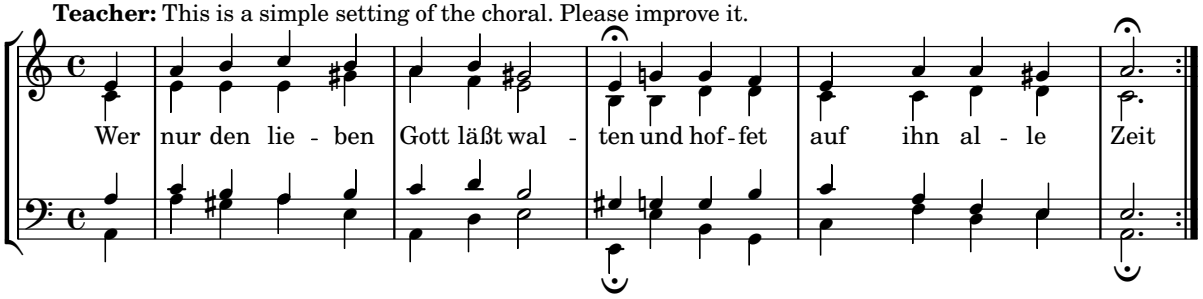
\score {
  <<
    \exercise
    \simpleBach
    \chromaticBach
  >>

  \header {
    title = \markup \column {
              \combine \null \vspace #1
              "Exercise: Improve the given choral"
              " "
            }
  }
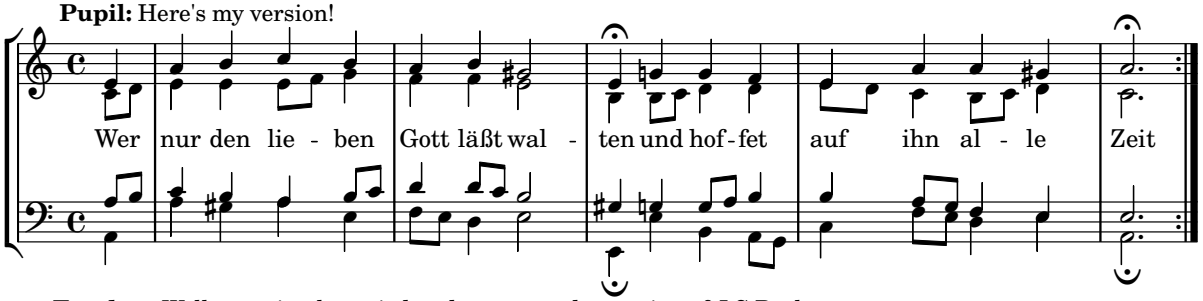
  \layout {
    \context {
      \Lyrics
      \override LyricText.X-offset = #-1
    }
  }
}
```

Exercise: Improve the given choral

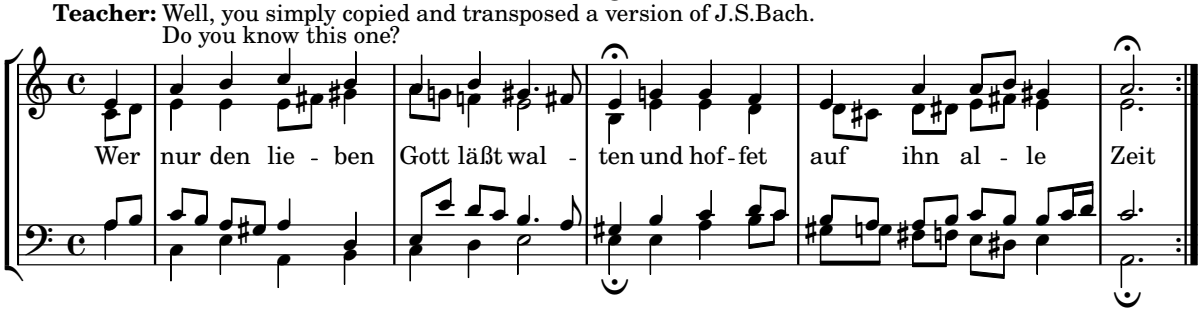
Teacher: This is a simple setting of the choral. Please improve it.



Pupil: Here's my version!



Teacher: Well, you simply copied and transposed a version of J.S.Bach.
Do you know this one?



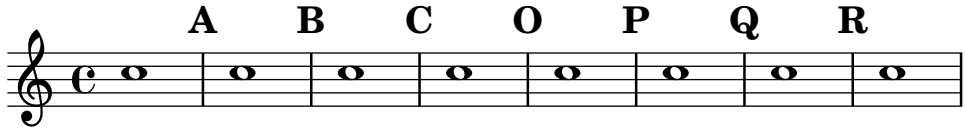
28 Preparing parts

Forzar que las marcas de ensayo comiencen en una letra o número dados

Este fragmento de código muestra cómo obtener marcas de ensayo ordenadas automáticamente, pero partiendo de la letra o número deseados.

```
\relative c'' {
  \override Score.RehearsalMark.Y-offset = #3.5

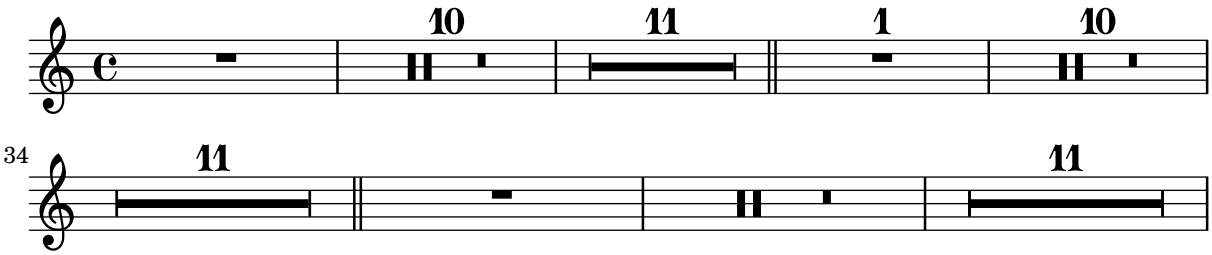
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1
}
```



Numerar silencios seguidos de un compás

Los silencios de varios compases muestran su duración con un número excepto los que tienen un solo compás. Esto se puede cambiar fijando el valor de `restNumberThreshold`.

```
{
  \compressEmptyMeasures
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 0
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 10
  R1 R1*10 R1*11
}
```



Plantilla de cuarteto de cuerda con particellas independientes

El fragmento de código “Plantilla de cuarteto de cuerda” produce un resultado satisfactorio para el cuarteto, pero ¿y si tenemos que imprimir las particellas? Esta nueva plantilla muestra cómo usar la funcionalidad `\tag` (etiqueta) para dividir fácilmente una pieza en particellas individuales.

Tenemos que dividir esta plantilla en archivos independientes; los nombres de archivo están dentro de los comentarios al principio de cada archivo. `piece.ly` contiene todas las definiciones de música. Los otros archivos (`score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` y `vlc.ly`) producen la particella correspondiente.

¡No olvide quitar los comentarios que hemos especificado cuando use los archivos independientes!

```
% piece.ly
% (This is the global definitions file.)

global= {
  \time 4/4
  \key c \major
}

Violinone = \new Voice \relative c'' {
  c2 d e1
  \bar "|."
}

Violintwo = \new Voice \relative c'' {
  g2 g e1
  \bar "|."
}

Viola = \new Voice \relative c' {
  \clef alto
  e2 d c1
  \bar "|."
}

Cello = \new Voice \relative c' {
  \clef bass
  c2 b a1
  \bar "|."
}

music = <<
  \tag #'score \tag #'vn1
  \new Staff \with { instrumentName = "Violin 1" }
    << \global \Violinone >>

  \tag #'score \tag #'vn2
  \new Staff \with { instrumentName = "Violin 2" }
    << \global \Violintwo >>

  \tag #'score \tag #'vla
  \new Staff \with { instrumentName = "Viola" }
    << \global \Viola >>

  \tag #'score \tag #'vlc
  \new Staff \with { instrumentName = "Cello" }
```

```
    << \global \Cello >>
>>


% These are the other files you need to save on your computer

% score.ly
% (This is the main file.)

% Uncomment the line below when using a separate file.
% \include "piece.ly"

#(set-global-staff-size 14)

\score {
  \new StaffGroup \keepWithTag #'score \music
  \layout { }
  \midi { }
}


%{ Uncomment this block when using separate files.

% vn1.ly
% (This is the Violin 1 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn1 \music
  \layout { }
}

% vn2.ly
% (This is the Violin 2 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn2 \music
  \layout { }
}

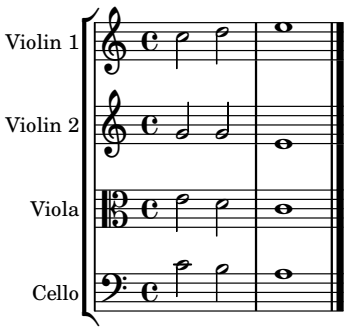
% vla.ly
% (This is the Viola part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}

% vlc.ly
% (This is the Cello part file.)
```

```
\include "piece.ly"
\score {
  \keepWithTag #'vlc \music
  \layout { }
}
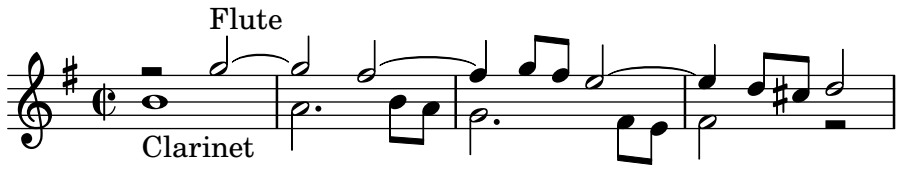
%}
```



29 Real music

Modificar la salida MIDI para que tenga un canal por cada voz

Al producir una salida MIDI, el comportamiento predeterminado es que cada pentagrama representa un canal MIDI, con todas las voces de dicho pentagrama mezcladas. Esto reduce al mínimo el riesgo de que se agote el número de canales MIDI disponibles, pues existe un máximo de 16 canales por cada puerto MIDI, y la mayoría de los dispositivos solo tiene un puerto.

Sin embargo, cuando se traslada el interpretador `Staff_performer` al contexto `Voice`, cada voz de un pentagrama puede tener su propio canal MIDI, como se muestra en el siguiente ejemplo: a pesar de estar sobre el mismo pentagrama, se crean dos canales MIDI, cada uno con un `midiInstrument` distinto.

```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = "flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = "clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}
```
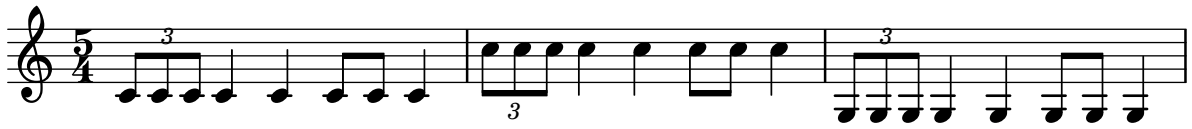


Crear una secuencia de notas a distintas alturas

En una música que tenga muchas apariciones de la misma secuencia de notas a distintas alturas, podría ser de utilidad la siguiente función musical. Admite una nota, de la que solo se utiliza su altura. Las funciones de apoyo en Scheme se han tomado prestadas del documento de "Consejos y trucos"de la versión 2.10 del manual.

Este ejemplo crea las duraciones rítmicas que se usan a todo lo largo de *Marte*, de *Los Planetas* de Gustav Holst.

```
rhythm =
#(define-music-function (p) (ly:pitch?)
   "Make the rhythm in Mars (the Planets) at the given pitch"
  #{ \tuplet 3/2 { $p 8 8 8 } 4 4 8 8 4 #})

\new Staff {
  \time 5/4
  \rhythm c'
  \rhythm c''
  \rhythm g
}
```



Hacer ligaduras entre voces distintas
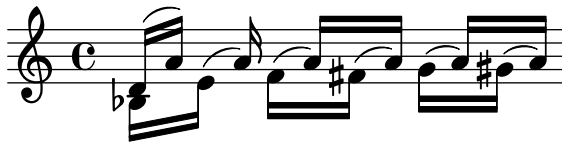
En determinadas situaciones es necesario crear ligaduras de expresión entre notas que están en voces distintas.

La solución es añadir notas invisibles a una de las voces utilizando `\hideNotes`.

Este ejemplo es el compás 235 de la Chacona de la segunda Partita para violín solo, BWV 1004, de Bach.

```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
```
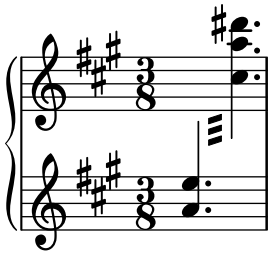
```
}
```



Trémolos de pentagrama cruzado

Dado que `\repeat tremolo` espera exactamente dos argumentos musicales para los trémolos de acorde, la nota o acorde que cambia de pentagrama en un trémolo que cruza el pentagrama se debe colocar dentro de llaves curvas junto a su instrucción `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```



Demo of MIDI instruments

Problem: How to know which `midiInstrument` values would be best for your composition?

Solution: A LilyPond demo file. You have to compile this snippet by yourself and listen to the created MIDI output file.

```
melody = \relative c' {
  \tempo 4 = 150
  c4.\mf g c16 b' c d
  e16 d e f g4 g'4 r
  R1
}
```

```
\score {
  \new Voice \melody
  \layout { }
}

\score {
  \new Voice {
    r\mf
    % 1-8 keyboard
    \set Staff.midiInstrument = "acoustic grand" \melody
    \set Staff.midiInstrument = "bright acoustic" \melody
    \set Staff.midiInstrument = "electric grand" \melody
    \set Staff.midiInstrument = "honky-tonk" \melody
    \set Staff.midiInstrument = "electric piano 1" \melody
    \set Staff.midiInstrument = "electric piano 2" \melody
    \set Staff.midiInstrument = "harpsichord" \melody
    \set Staff.midiInstrument = "clav" \melody

    % 9-16 chrom percussion
    \set Staff.midiInstrument = "celesta" \melody
    \set Staff.midiInstrument = "glockenspiel" \melody
    \set Staff.midiInstrument = "music box" \melody
    \set Staff.midiInstrument = "vibraphone" \melody
    \set Staff.midiInstrument = "marimba" \melody
    \set Staff.midiInstrument = "xylophone" \melody
    \set Staff.midiInstrument = "tubular bells" \melody
    \set Staff.midiInstrument = "dulcimer" \melody

    % 17-24 organ
    \set Staff.midiInstrument = "drawbar organ" \melody
    \set Staff.midiInstrument = "percussive organ" \melody
    \set Staff.midiInstrument = "rock organ" \melody
    \set Staff.midiInstrument = "church organ" \melody
    \set Staff.midiInstrument = "reed organ" \melody
    \set Staff.midiInstrument = "accordion" \melody
    \set Staff.midiInstrument = "harmonica" \melody
    \set Staff.midiInstrument = "concertina" \melody

    % 25-32 guitar
    \set Staff.midiInstrument = "acoustic guitar (nylon)" \melody
    \set Staff.midiInstrument = "acoustic guitar (steel)" \melody
    \set Staff.midiInstrument = "electric guitar (jazz)" \melody
    \set Staff.midiInstrument = "electric guitar (clean)" \melody
    \set Staff.midiInstrument = "electric guitar (muted)" \melody
    \set Staff.midiInstrument = "overdriven guitar" \melody
    \set Staff.midiInstrument = "distorted guitar" \melody
    \set Staff.midiInstrument = "guitar harmonics" \melody

    % 33-40 bass
    \set Staff.midiInstrument = "acoustic bass" \melody
    \set Staff.midiInstrument = "electric bass (finger)" \melody
```

```
\set Staff.midiInstrument = "electric bass (pick)" \melody
\set Staff.midiInstrument = "fretless bass" \melody
\set Staff.midiInstrument = "slap bass 1" \melody
\set Staff.midiInstrument = "slap bass 2" \melody
\set Staff.midiInstrument = "synth bass 1" \melody
\set Staff.midiInstrument = "synth bass 2" \melody

% 41-48 strings
\set Staff.midiInstrument = "violin" \melody
\set Staff.midiInstrument = "viola" \melody
\set Staff.midiInstrument = "cello" \melody
\set Staff.midiInstrument = "contrabass" \melody
\set Staff.midiInstrument = "tremolo strings" \melody
\set Staff.midiInstrument = "pizzicato strings" \melody
\set Staff.midiInstrument = "orchestral harp" \melody
\set Staff.midiInstrument = "timpani" \melody

% 49-56 ensemble
\set Staff.midiInstrument = "string ensemble 1" \melody
\set Staff.midiInstrument = "string ensemble 2" \melody
\set Staff.midiInstrument = "synthstrings 1" \melody
\set Staff.midiInstrument = "synthstrings 2" \melody
\set Staff.midiInstrument = "choir aahs" \melody
\set Staff.midiInstrument = "voice oohs" \melody
\set Staff.midiInstrument = "synth voice" \melody
\set Staff.midiInstrument = "orchestra hit" \melody

% 57-64 brass
\set Staff.midiInstrument = "trumpet" \melody
\set Staff.midiInstrument = "trombone" \melody
\set Staff.midiInstrument = "tuba" \melody
\set Staff.midiInstrument = "muted trumpet" \melody
\set Staff.midiInstrument = "french horn" \melody
\set Staff.midiInstrument = "brass section" \melody
\set Staff.midiInstrument = "synthbrass 1" \melody
\set Staff.midiInstrument = "synthbrass 2" \melody

% 65-72 reed
\set Staff.midiInstrument = "soprano sax" \melody
\set Staff.midiInstrument = "alto sax" \melody
\set Staff.midiInstrument = "tenor sax" \melody
\set Staff.midiInstrument = "baritone sax" \melody
\set Staff.midiInstrument = "oboe" \melody
\set Staff.midiInstrument = "english horn" \melody
\set Staff.midiInstrument = "bassoon" \melody
\set Staff.midiInstrument = "clarinet" \melody

% 73-80 pipe
\set Staff.midiInstrument = "piccolo" \melody
\set Staff.midiInstrument = "flute" \melody
\set Staff.midiInstrument = "recorder" \melody
\set Staff.midiInstrument = "pan flute" \melody
```

```
\set Staff.midiInstrument = "blown bottle" \melody
\set Staff.midiInstrument = "shakuhachi" \melody
\set Staff.midiInstrument = "whistle" \melody
\set Staff.midiInstrument = "ocarina" \melody

% 81-88 synth lead
\set Staff.midiInstrument = "lead 1 (square)" \melody
\set Staff.midiInstrument = "lead 2 (sawtooth)" \melody
\set Staff.midiInstrument = "lead 3 (calliope)" \melody
\set Staff.midiInstrument = "lead 4 (chiff)" \melody
\set Staff.midiInstrument = "lead 5 (charang)" \melody
\set Staff.midiInstrument = "lead 6 (voice)" \melody
\set Staff.midiInstrument = "lead 7 (fifths)" \melody
\set Staff.midiInstrument = "lead 8 (bass+lead)" \melody

% 89-96 synth pad
\set Staff.midiInstrument = "pad 1 (new age)" \melody
\set Staff.midiInstrument = "pad 2 (warm)" \melody
\set Staff.midiInstrument = "pad 3 (polysynth)" \melody
\set Staff.midiInstrument = "pad 4 (choir)" \melody
\set Staff.midiInstrument = "pad 5 (bowed)" \melody
\set Staff.midiInstrument = "pad 6 (metallic)" \melody
\set Staff.midiInstrument = "pad 7 (halo)" \melody
\set Staff.midiInstrument = "pad 8 (sweep)" \melody

% 97-104 synth effects
\set Staff.midiInstrument = "fx 1 (rain)" \melody
\set Staff.midiInstrument = "fx 2 (soundtrack)" \melody
\set Staff.midiInstrument = "fx 3 (crystal)" \melody
\set Staff.midiInstrument = "fx 4 (atmosphere)" \melody
\set Staff.midiInstrument = "fx 5 (brightness)" \melody
\set Staff.midiInstrument = "fx 6 (goblins)" \melody
\set Staff.midiInstrument = "fx 7 (echoes)" \melody
\set Staff.midiInstrument = "fx 8 (sci-fi)" \melody

% 105-112 ethnic
\set Staff.midiInstrument = "sitar" \melody
\set Staff.midiInstrument = "banjo" \melody
\set Staff.midiInstrument = "shamisen" \melody
\set Staff.midiInstrument = "koto" \melody
\set Staff.midiInstrument = "kalimba" \melody
\set Staff.midiInstrument = "bagpipe" \melody
\set Staff.midiInstrument = "fiddle" \melody
\set Staff.midiInstrument = "shanai" \melody

% 113-120 percussive
\set Staff.midiInstrument = "tinkle bell" \melody
\set Staff.midiInstrument = "agogo" \melody
\set Staff.midiInstrument = "steel drums" \melody
\set Staff.midiInstrument = "woodblock" \melody
\set Staff.midiInstrument = "taiko drum" \melody
\set Staff.midiInstrument = "melodic tom" \melody
```

```
    \set Staff.midiInstrument = "synth drum" \melody
    \set Staff.midiInstrument = "reverse cymbal" \melody
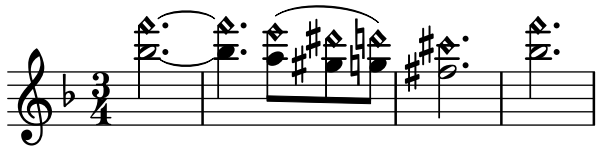
    % 121-128 sound effects
    \set Staff.midiInstrument = "guitar fret noise" \melody
    \set Staff.midiInstrument = "breath noise" \melody
    \set Staff.midiInstrument = "seashore" \melody
    \set Staff.midiInstrument = "bird tweet" \melody
    \set Staff.midiInstrument = "telephone ring" \melody
    \set Staff.midiInstrument = "helicopter" \melody
    \set Staff.midiInstrument = "applause" \melody
    \set Staff.midiInstrument = "gunshot" \melody
  }
  \midi { }
}
```



Armónicos con puntillo

Los armónicos artificiales que usan la instrucción \harmonic no tienen puntillo. Para sobreescribir este comportamiento, fije la propiedad de contexto harmonicDots.

```
\relative c''' {
  \time 3/4
  \key f \major
  \set harmonicDots = ##t
  <bes f'\harmonic>2. ~
  <bes f'\harmonic>4. <a e'\harmonic>8( <gis dis'\harmonic> <g d'\harmonic>)
  <fis cis'\harmonic>2.
  <bes f'\harmonic>2.
}
```



Indicaciones de compás polimétricas profundamente personalizadas

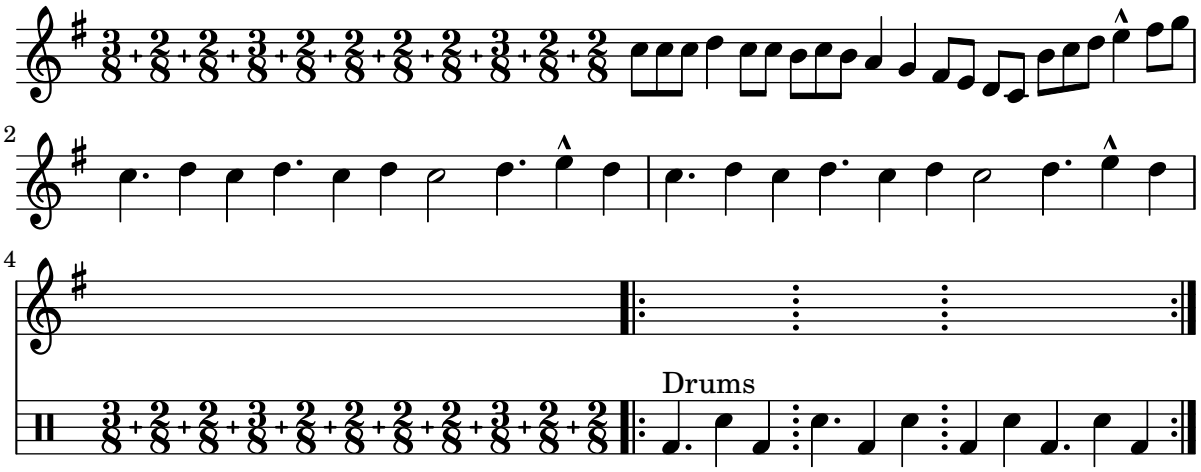
Aunque aquí el elemento más esencial no era la indicación de compás polimétrica que se muestra, se ha incluido para mostrar el pulso de la pieza, que por cierto es la plantilla de una canción de los Balcanes real.

```
melody = \relative c'' {
  \key g \major
  \time #'((3 . 8) (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8)
           (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8))
  \set Timing.beamExceptions = #'()
```

```
  \set Timing.beatStructure = 3,2,2,3,2,2,2,2,3,2,2
  c8 c c d4 c8 c b c b a4 g fis8 e d c b' c d e4-^ fis8 g \break
  c,4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
}

drum = \new DrumStaff \drummode {
  \repeat volta 2 {
    bd4.^\markup { Drums } sn4 bd \bar ";"
    sn4. bd4 sn \bar ";"
    bd sn bd4. sn4 bd
  }
}

\new Staff {
  \melody
  \drum
}
```



Indicating cross-staff chords with a bracket

A non-arpeggiato bracket can indicate that notes on two different staves are to be played with the same hand. In order to do this, the PianoStaff must be set to accept cross-staff brackets.

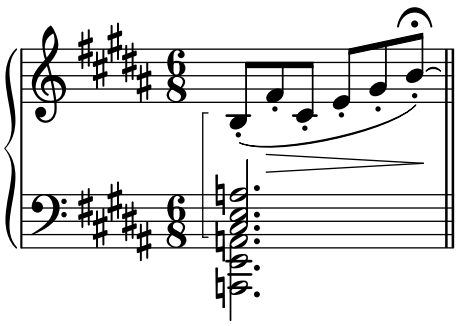
The following example typesets measure 65 of Debussy's prelude *Les collines d'Anacapri*.

```
\new PianoStaff <<
  \once \set PianoStaff.connectChordBrackets = ##t

  \new Staff \relative c' {
    \key b \major
    \time 6/8
    b8-.(\nonArpeggiato fis'-.\> cis-.
      e-. gis-. b-.)\!\fermata^\laissezVibrer
    \section
  }

  \new Staff \relative c' {
    \clef bass
    \key b \major
```

```
    << { <a e cis>2.\nonArpeggiato } \\
       { <a, e a,>2. } >>
    \section
  }
>>
```



Insertar fragmentos de partituras sobre el pentagrama, como elementos de marcado

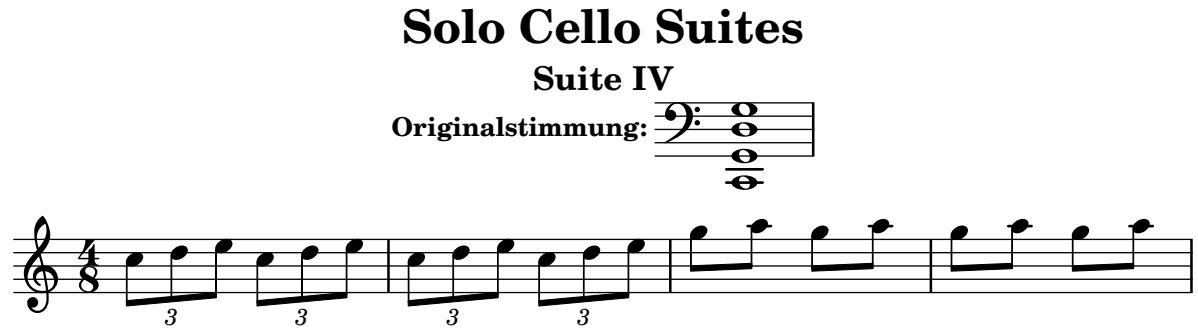
La instrucción \markup es muy versátil. En este fragmento de código, contiene un bloque \score con una partitura completa en lugar de textos u otras marcas.

```
tuning = \markup \score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \clef bass
    <c, g, d g>1
  }
  \layout {
    indent = 0\cm
  }
}

\header {
  title = "Solo Cello Suites"
  subtitle = "Suite IV"
  subsubtitle = \markup { Originalstimmung: \raise #0.5 \tuning }
  tagline = ##f
}

\layout {
  ragged-right = ##f
}

\relative c'' {
  \time 4/8
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { c d e }
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { c d e }
  g8 a g a
  g8 a g a
}
```



Ejemplo de percusión

Un ejemplo breve extraído de la *Historia de un soldado* de Strawinsky.

```
#(define mydrums '((bassdrum   default #f  4)
                   (snare      default #f -4)
                   (tambourine default #f  0)))

U = \stemUp
D = \stemDown

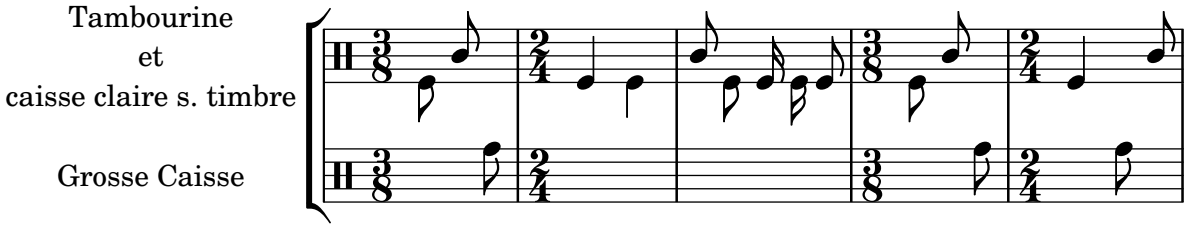
global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}

drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
      sn4 \D sn4 |
      \U tamb8 \D sn \U sn16 \D sn \U sn8 |
      \D sn8 \U tamb s |
      \U sn4 s8 \U tamb
    }
  >>
}

drumsB = \drummode {
  s4 bd8 s2*2 s4 bd8 s4 bd8 s
}

\layout {
  indent = 40\mm
  \context {
    \DrumStaff
    drumStyleTable = #(alist->hash-table mydrums)
  }
}
```

```
\score {
  \new StaffGroup <<
    \new DrumStaff \with {
      instrumentName = \markup \center-column {
        "Tambourine"
        "et"
        "caisse claire s. timbre" }
    } \drumsA
    \new DrumStaff \with {
      instrumentName = "Grosse Caisse"
    }\drumsB
  >>
}
```



Imprimir música que tenga compases distintos en cada pentagrama

En el siguiente fragmento de código, dos partes distintas tienen un compás completamente diferente y sin embargo se mantienen sincronizadas. Las barras de compás ya no se pueden imprimir en el nivel de `Score`; para permitir barras de compás independientes en cada parte se trasladan los grabadores `Default_barline_engraver` y `Timing_translator` desde el contexto de partitura `Score` hasta el contexto de pentagrama `Staff`.

Si son necesarios números de compás, el grabador de números de compás `Bar_number_engraver` también debe trasladarse, ya que descansa en propiedades fijadas por el `Timing_translator`; se puede usar un bloque `\with` para añadir números de compás al pentagrama apropiado.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}
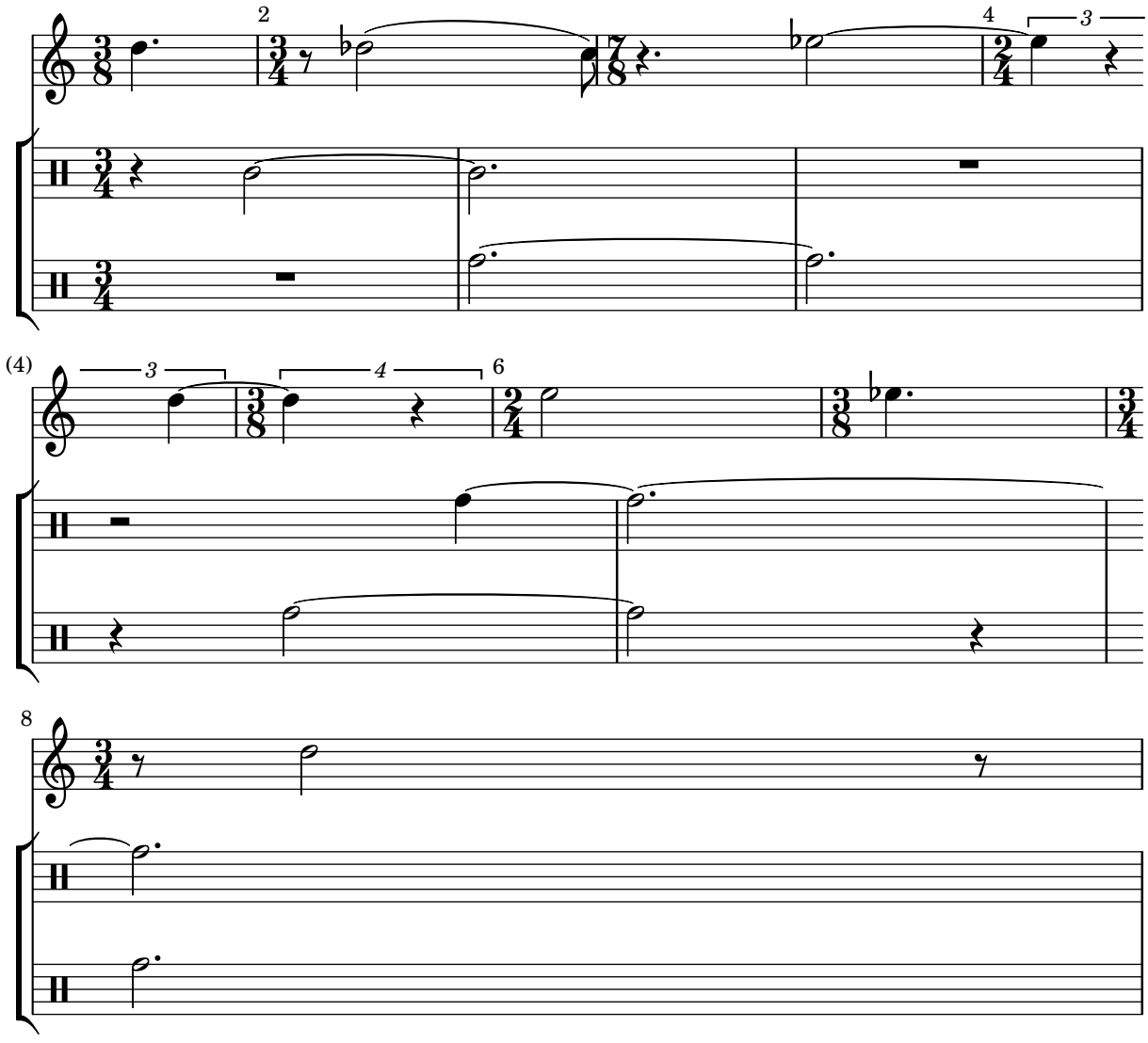
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
```

```
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. ees''2 ~ |
    \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
    \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
    \time 2/4 e''2 |
    \time 3/8 es''4. |
    \time 3/4 r8 d''2 r8 |
  }
>>

Perkussion = \new StaffGroup <<
  \new Staff <<
    \global
    {
      \clef percussion
      \time 3/4 r4 c'2 ~ |
      c'2. |
      R2. |
      r2 g'4 ~ |
      g'2. ~ |
      g'2. |
    }
  >>
  \new Staff <<
    \global {
      \clef percussion
      \time 3/4 R2. |
      g'2. ~ |
      g'2. |
      r4 g'2 ~ |
      g'2 r4 |
      g'2. |
    }
  >>
>>
```

```
\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}
```



30 Really cool

Añadir la fecha actual a una partitura

Con algo de código de Scheme, se puede añadir fácilmente la fecha actual a una partitura.

```
\paper { tagline = ##f }

% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))

% use it in the title block:
\header {
  title = "Including the date!"
  subtitle = \date
}

\score {
  \relative c'' {
    c4 c c c
  }
}
% and use it in a \markup block:
\markup {
  \date
}
```

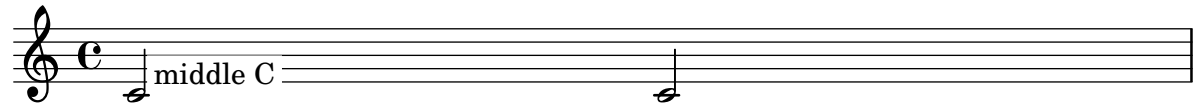


Poner en blanco las líneas del pentagrama utilizando la instrucción \whiteout

La instrucción \whiteout intercala un rectángulo blanco debajo de un elemento de marcado. Este rectángulo blanco no tapa a ningún otro objeto gráfico, puesto que las líneas del pentagrama están en una capa inferior a la de la mayor parte de los otros objetos.

```
\layout {
  ragged-right = ##f
}

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```



Centrar texto debajo de un regulador

Este ejemplo proporciona una función para tipografiar un regulador con texto por debajo, como "molto" o "poco". El texto añadido cambia la dirección de acuerdo con la del regulador. El objeto Hairpin está alineado con un objeto gráfico DynamicText.

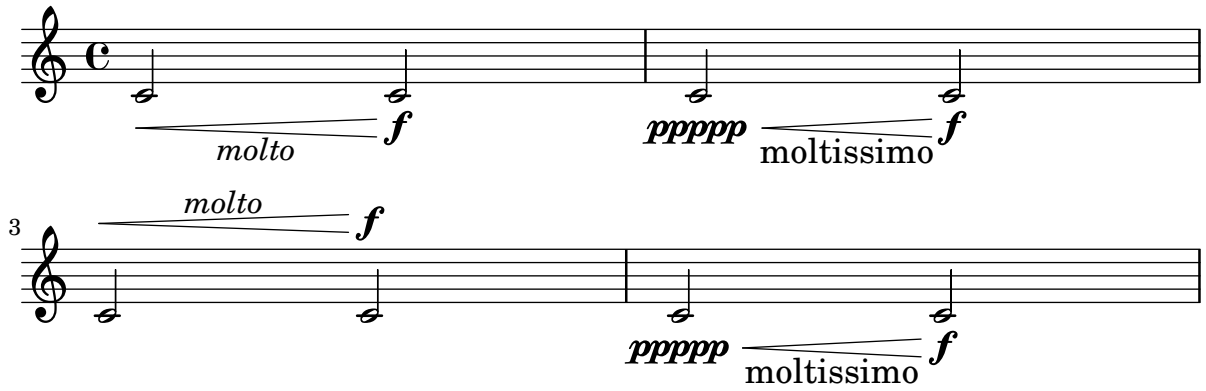
El ejemplo ilustra también cómo modificar la manera en que se imprime normalmente un objeto, utilizando código de Scheme.

```
hairpinWithCenteredText =
#(define-music-function (text) (markup?)
  #{
    \once \override Voice.Hairpin.after-line-breaking =
      #(lambda (grob)
         (let* ((stencil (ly:hairpin::print grob))
                (par-y (ly:grob-parent grob Y))
                (dir (ly:grob-property par-y 'direction))
                (staff-line-thickness
                 (ly:output-def-lookup (ly:grob-layout grob)
                                       'line-thickness))
                (new-stencil
                 (ly:stencil-aligned-to
                  (ly:stencil-combine-at-edge
                   (ly:stencil-aligned-to stencil X CENTER)
                   Y dir
                   (ly:stencil-aligned-to
                    (grob-interpret-markup
                     grob
                     (make-fontsize-markup
                      (magnification->font-size
                       (+ (ly:staff-symbol-staff-space grob)
                          (/ staff-line-thickness 2)))
                      text))
                    X CENTER))
                  X LEFT))
                (staff-space (ly:output-def-lookup
                 (ly:grob-layout grob) 'staff-space))
                (par-x (ly:grob-parent grob X))
                (dyn-text (grob::has-interface par-x
                                               'dynamic-text-interface))
                (dyn-text-stencil-x-length
                 (if dyn-text
                     (interval-length
                      (ly:stencil-extent
                       (ly:grob-property par-x 'stencil) X))
                     0))
                (x-shift
                 (if dyn-text (- (+ staff-space dyn-text-stencil-x-length)
                                 (* 0.5 staff-line-thickness))
                     0)))
           (ly:grob-set-property! grob 'Y-offset 0)
           (ly:grob-set-property! grob
                                  'stencil (ly:stencil-translate-axis
                                            new-stencil
```

```
                                          x-shift X))))
  #})

hairpinMolto = \hairpinWithCenteredText \markup { \italic molto }
hairpinMore = \hairpinWithCenteredText \markup { \larger moltissimo }

\relative c' {
  \hairpinMolto c2\< c\f
  \hairpinMore c2\ppppp\< c\f
  \break
  \hairpinMolto c2^\< c\f
  \hairpinMore c2\ppppp\< c\f
}
```



Modificación de propiedades para objetos gráficos individuales

La instrucción `\applyOutput` hace posible el ajuste fino de cualquier objeto de presentación, en cualquier contexto. Requiere una función de Scheme contres argumentos.
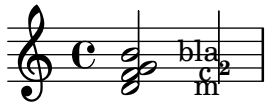
En el ejemplo de abajo, la función `mc-squared` se ejecuta para todos los objetos gráficos `NoteHead` (dentro del contexto `Voice` actual) en el punto temporal en curso; la función modifica el `stencil` o sello del objeto gráfico, usando la propiedad `staff-position` para sustituir ciertas alturas con elementos de marcado.

Consulte el 'Manual de extensión del programa' (`https://lilypond.org/doc/v2.24/Documentation/extending/running-a-function-on-all-layout-objects`) para más información.

```
#(define (mc-squared grob grob-origin context)
   (let ((sp (ly:grob-property grob 'staff-position)))
     (ly:grob-set-property!
      grob 'stencil
      (grob-interpret-markup grob
       #{ \markup \lower #0.5
           #(case sp
              ((-5) "m")
              ((-3) "c ")
              ((-2) #{ \markup \teeny \bold 2 #})
              (else "bla")) #}))))

\relative c' {
  <d f g b>2
  \applyOutput Voice.NoteHead #mc-squared
```

```
  <d f g b>2
}
```

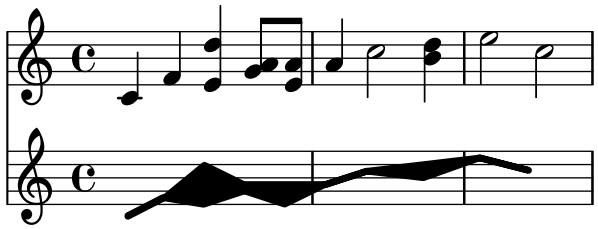


Clusters («racimos»)

Los «clusters» o racimos son un mecanismo para indicar la interpretación de un ámbito de notas al mismo tiempo.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Colorear las notas según su altura

Es posible colorear la cabeza de las notas dependiendo de su altura y/o de su nombre: la función que se usa en este ejemplo hace posible incluso distinguir los armónicos.
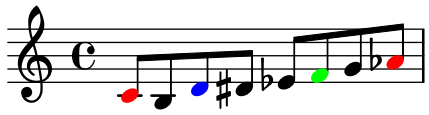
```
% Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 2 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 3 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 3 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 4 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 5 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 5 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 6 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 1 NATURAL) (x11-color 'blue))
    (cons (ly:make-pitch 0 3 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 4 FLAT) (x11-color 'blue))
    (cons (ly:make-pitch 0 5 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 6 FLAT) (x11-color 'blue))))
```

```
% Compare pitch and alteration (not octave).
#(define (pitch-equals? p1 p2)
   (and
    (= (ly:pitch-alteration p1) (ly:pitch-alteration p2))
    (= (ly:pitch-notename p1) (ly:pitch-notename p2))))

#(define (pitch-to-color pitch)
   (let ((color (assoc pitch color-mapping pitch-equals?)))
     (if color
         (cdr color))))

#(define (color-notehead grob)
   (pitch-to-color
    (ly:event-property (event-cause grob) 'pitch)))

\score {
  \new Staff \relative c' {
    \override NoteHead.color = #color-notehead
    c8 b d dis ees f g aes
  }
}
```
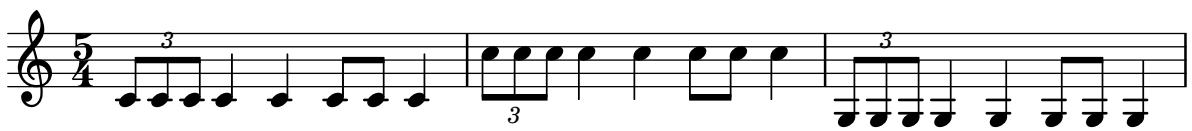


Crear una secuencia de notas a distintas alturas

En una música que tenga muchas apariciones de la misma secuencia de notas a distintas alturas, podría ser de utilidad la siguiente función musical. Admite una nota, de la que solo se utiliza su altura. Las funciones de apoyo en Scheme se han tomado prestadas del documento de "Consejos y trucos"de la versión 2.10 del manual.

Este ejemplo crea las duraciones rítmicas que se usan a todo lo largo de *Marte*, de *Los Planetas* de Gustav Holst.

```
rhythm =
#(define-music-function (p) (ly:pitch?)
   "Make the rhythm in Mars (the Planets) at the given pitch"
  #{ \tuplet 3/2 { $p 8 8 8 } 4 4 8 8 4 #})

\new Staff {
  \time 5/4
  \rhythm c'
  \rhythm c''
  \rhythm g
}
```



Generación de notas aleatorias
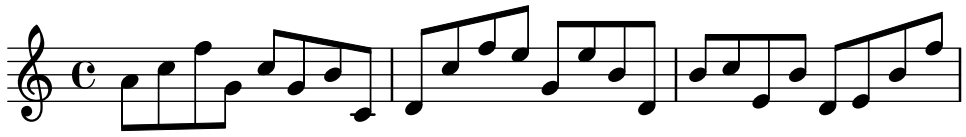
Este fragmento de código basado en Scheme genera notas aleatorias. Úselo como

```
\randomNotes n desde hasta duración
```

para generar *n* notas aleatorias entre las alturas *desde* y *hasta*, con una longitud de *duración*.

```
randomNotes =
#(define-music-function (n from to dur)
   (integer? ly:pitch? ly:pitch? ly:duration?)
   (let ((from-step (ly:pitch-steps from))
         (to-step (ly:pitch-steps to)))
     (make-sequential-music
      (map (lambda (_)
             (let* ((step (+ from-step
                             (random (- to-step from-step))))
                    (pitch (ly:make-pitch 0 step 0)))
               #{ $pitch $dur #}))
           (iota n)))))

\randomNotes 24 c' g'' 8
```



Generación de partituras completas y partes de libro en Scheme sin usar el analizador sintáctico

Internamente, una partitura de LilyPond no es más que una expresión de Scheme, generada por el analizador sintáctico de LilyPond. Usando Scheme, también podemos generar automáticamente una partitura sin ningún archivo de entrada. Si tenemos la expresión musical en Scheme, se puede generar la partitura simplemente llamando a

```
(scorify-music music)
```

sobre la música. Así se genera un objeto partitura, para el que podemos entonces fijar un bloque layout personalizado con

```
(let* ((layout (ly:output-def-clone $defaultlayout)))
   ; modificamos el layout aquí, y después lo asignamos:
   (ly:score-add-output-def! score layout))
```

Finalmente, todo lo que tenemos que hacer es pasar esta partitura a lilypond para que realice la composición tipográfica. Este fragmento de código define las funciones `(add-score score)`, `(add-text text)` y `(add-music music)` para pasar una partitura completa, elementos de marcado o algo de música a LilyPond para su composición tipográfica.

Este fragmento de código también funciona para la tipografía de partituras dentro de un bloque `\book {...}`, así como partituras normales del nivel superior jerárquico. Para conseguirlo, cada una de las partituras destinadas a su tipografiado se añaden al final de una lista de partituras del nivel superior jerárquico y se modifica el toplevel-book-handler (que es una función de Scheme que se llama para procesar un libro una vez que se ha cerrado el bloque `\book{..}`) para insertar todas las partituras así coleccionadas al libro.

Nota: por razones técnicas, solo se muestra el primer `\book`, debido a que las otras instrucciones `\book` crean archivos de salida adicionales.

```
#(define-public (add-score score)
   (module-define! (current-module) 'toplevel-scores
                   (cons score (module-ref (current-module) 'toplevel-scores '()))))

#(define-public (add-text text)
   (add-score (list text)))

#(define-public (add-music music)
   (collect-music-aux (lambda (score)
                        (add-score score))
                      music))

#(define-public (toplevel-book-handler book)
   (map (lambda (score)
          (ly:book-add-score! book score))
        (reverse (ly:book-lookup book 'toplevel-scores)))
   (print-book-with-defaults book))

#(define-public (book-score-handler book score)
   (add-score score))

#(define-public (book-text-handler book text)
   (add-text text))

#(define-public (book-music-handler book music)
   (add-music music))


% Some example code to show how to use these functions.  Each call to
% `\oneNoteScore` constructs a global markup followed by a single
% staff with a single quarter note.  The pitch of this note is taken
% from the variable `pitch`; the start value 0 corresponds to pitch C.
% After emitting the score, variable `pitch` gets increased by 1.
%
% `\oneNoteScore` calls Scheme function `add-one-note-score` to do all
% the work.

#(define add-one-note-score #f)
#(let ((pitch 0))
   (set! add-one-note-score
         (lambda ()
           (let* ((music
                   (make-music
                    'EventChord
                    'elements (list (make-music
                                     'NoteEvent
                                     'duration (ly:make-duration 2 0 1/1)
                                     'pitch (ly:make-pitch 0 pitch 0)))))
                  (score (scorify-music music))
                  (layout (ly:output-def-clone $defaultlayout))
                  (note-name (case pitch
                               ((0) "do")
```

```
                                ((1) "ré")
                                ((2) "mi")
                                ((3) "fa")
                                ((4) "sol")
                                ((5) "la")
                                ((6) "si")
                                (else "huh")))
                        (title (markup #:large #:line
                                       ("Score with a" note-name))))
                  (ly:score-add-output-def! score layout)
                  (add-text title)
                  (add-score score))
                (set! pitch (modulo (1+ pitch) 7)))))

oneNoteScore =
#(define-void-function () ()
   (add-one-note-score))

\book {
  \oneNoteScore

  \paper { tagline = ##f }
}
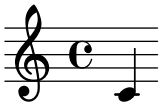

\book {
  \oneNoteScore
  \oneNoteScore

  \paper { tagline = ##f }
}

% Top-level scores are also handled correctly.
\oneNoteScore
\oneNoteScore

\paper { tagline = ##f }
```
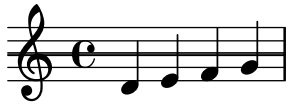
Score with a do



Hacer unas líneas del pentagrama más gruesas que las otras

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Armaduras de tonalidad no tradicionales

La muy utilizada instrucción \key establece la propiedad keyAlterations dentro del contexto Staff.

Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

```
\set Staff.keyAlterations =
  #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)
```
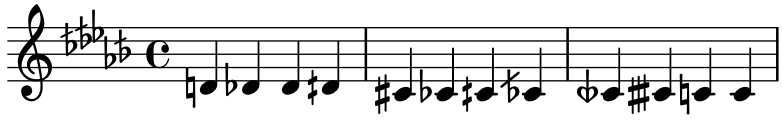
donde, para cada elemento dentro de la lista, *octava* especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), *paso* especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y *alteración* es SHARP, FLAT o DOUBLE-SHARP, etc., precedido de una coma.

De forma alternativa, podemos usar el formato más conciso (*paso . alteración*) para cada elemento de la lista si se usa la misma alteración en todas las octavas.

Para escalas microtonales en las que un "sostenido" no son 100 cents, *alteración* se refiere a la proporción de un tono entero de 200 cents.

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
                                 (6 . ,SEMI-FLAT))
  % \set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Imprimir música que tenga compases distintos en cada pentagrama

En el siguiente fragmento de código, dos partes distintas tienen un compás completamente diferente y sin embargo se mantienen sincronizadas. Las barras de compás ya no se pueden imprimir en el nivel de Score; para permitir barras de compás independientes en cada parte se trasladan los grabadores Default_barline_engraver y Timing_translator desde el contexto de partitura Score hasta el contexto de pentagrama Staff.

Si son necesarios números de compás, el grabador de números de compás Bar_number_engraver también debe trasladarse, ya que descansa en propiedades fijadas por el Timing_translator; se puede usar un bloque \with para añadir números de compás al pentagrama apropiado.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}
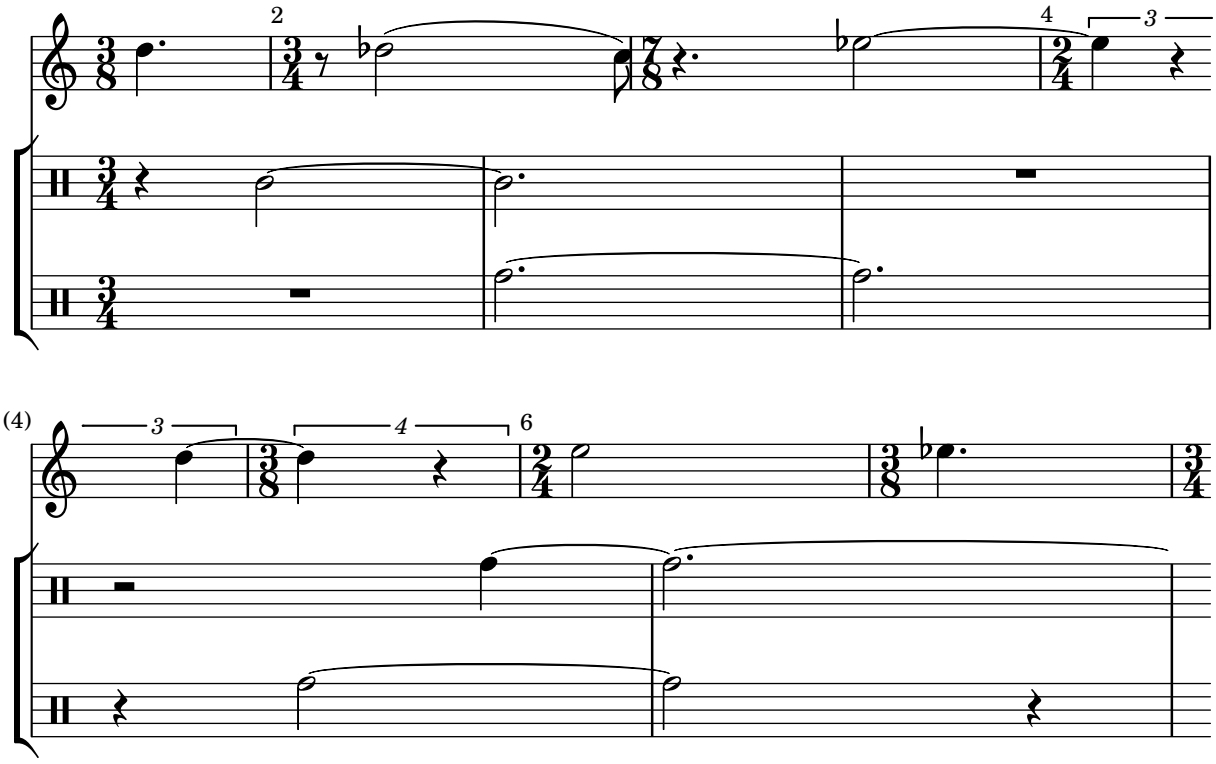
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

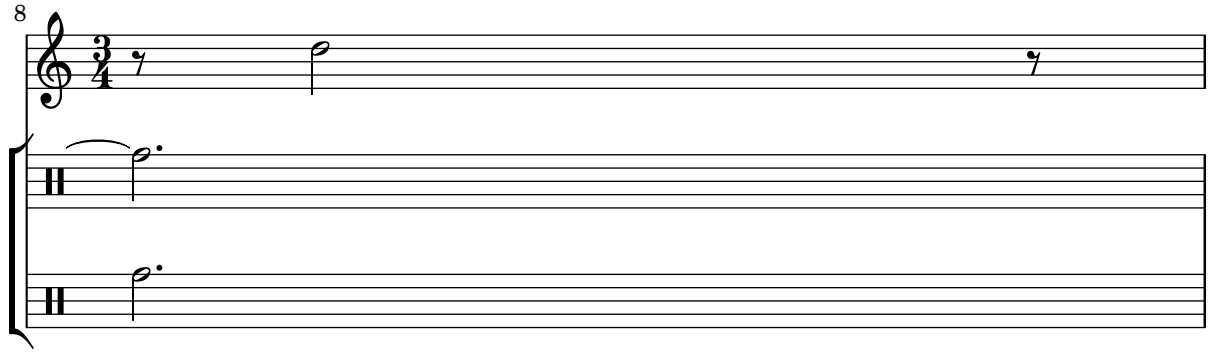
Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. ees''2 ~ |
    \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
    \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
    \time 2/4 e''2 |
    \time 3/8 es''4. |
    \time 3/4 r8 d''2 r8 |
  }
>>

Perkussion = \new StaffGroup <<
  \new Staff <<
    \global
```

```
    {
      \clef percussion
      \time 3/4 r4 c'2 ~ |
      c'2. |
      R2. |
      r2 g'4 ~ |
      g'2. ~ |
      g'2. |
    }
  >>
  \new Staff <<
    \global {
      \clef percussion
      \time 3/4 R2. |
      g'2. ~ |
      g'2. |
      r4 g'2 ~ |
      g'2 r4 |
      g'2. |
    }
  >>
>>

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}
```



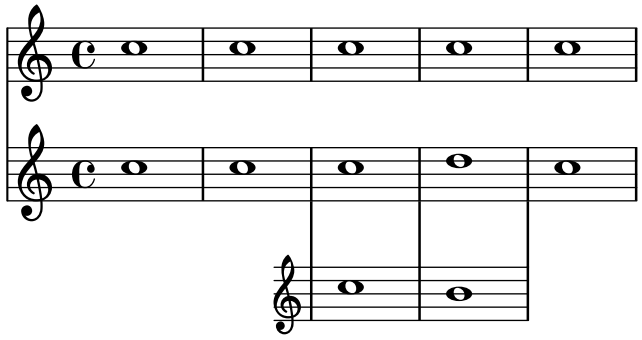

8

31 Really simple

Añadir un pentagrama nuevo

Se puede añadir (posiblemente de forma temporal) un pentagrama nuevo una vez que la pieza ha comenzado.

```
\score {
  <<
    \new Staff \relative c'' {
      c1 | c | c | c | c
    }
    \new StaffGroup \relative c'' {
      \new Staff {
        c1 | c
        <<
          { c1 | d }
          \new Staff {
            \once \omit Staff.TimeSignature
            c1 | b
          }
        >>
        c1
      }
    }
  >>
}
```



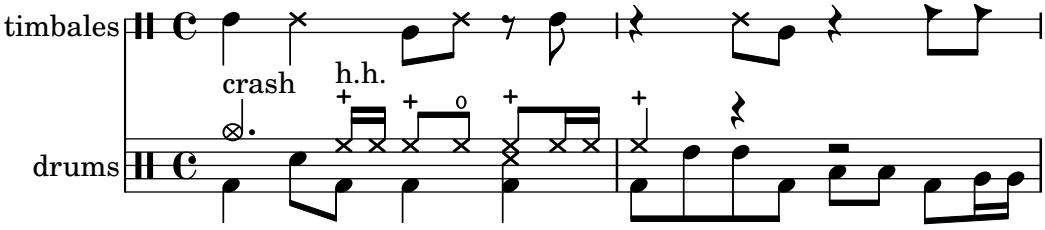
Escritura de partes de percusión

Mediante la utilización de las potentes herramientas preconfiguradas como la función `\drummode` y el contexto `DrumStaff`, la introducción de partes para percusión es muy fácil: las percusiones se sitúan en sus propias posiciones de pentagrama (con una clave especial) y tienen las cabezas correspondientes al instrumento. Es posible añadir un símbolo adicional a la percusión o reducir el número de líneas.

```
drh = \drummode {
        cymc4.^"crash" hhc16^"h.h." hh hhc8 hho hhc8 hh16 hh
        hhc4 r4 r2
      }
drl = \drummode {
        bd4 sn8 bd bd4 << bd ss >>
        bd8 tommh tommh bd toml toml bd tomfh16 tomfh
```

```
      }
timb = \drummode {
         timh4 ssh timl8 ssh r timh r4
         ssh8 timl r4 cb8 cb
       }
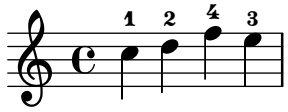
\score {
  <<
    \new DrumStaff \with {
      instrumentName = "timbales"
      drumStyleTable = #timbales-style
      \override StaffSymbol.line-count = #2
      \override BarLine.bar-extent = #'(-1 . 1)
    }
    <<
      \timb
    >>
    \new DrumStaff \with { instrumentName = "drums" }
    <<
      \new DrumVoice { \stemUp \drh }
      \new DrumVoice { \stemDown \drl }
    >>
  >>
  \layout { }
  \midi { \tempo 4 = 120 }
}
```



Añadir digitaciones a la partitura

Se pueden escribir instrucciones de digitación usando una sintaxis muy sencilla.

```
\relative c'' {
  c4-1 d-2 f-4 e-3
}
```

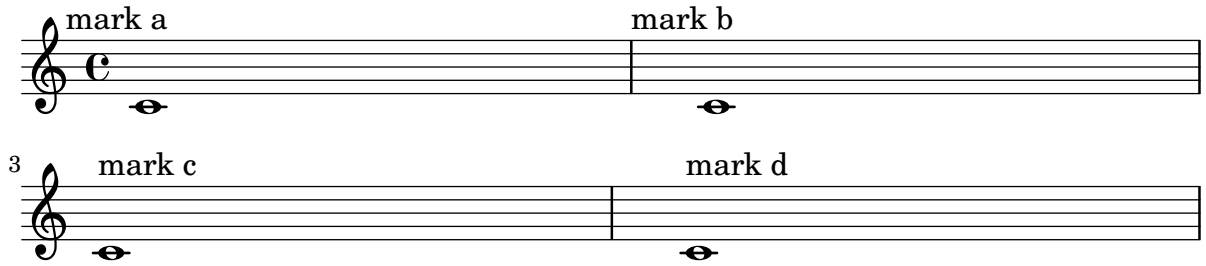


Aligning text marks to notes

By default, `TextMark` objects are aligned to so-called `NonMusicalPaperColumn` grobs, like the left edge of the staff or a bar line. They can be aligned to a note instead by setting the `non-musical` property to `#f`.

```
\layout {
  line-length = 80\mm
}

{
  \textMark "mark a" c'1 |
  \textMark "mark b" c'1 |
  \break
  \override Score.TextMark.non-musical = ##f
  \textMark "mark c" c'1 |
  \textMark "mark d" c'1 |
}
```
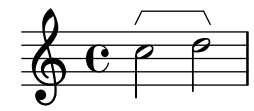


Corchetes de análisis encima del pentagrama

De forma predeterminada se añaden corchetes de análisis sencillos debajo del pentagrama. El ejemplo siguiente muestra una manera de colocarlos por encima.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```

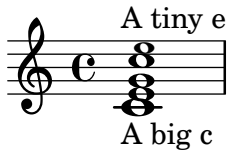


Changing a single note's size in a chord

Individual note heads in a chord can be modified with the `\tweak` command inside a chord, by altering the `font-size` property.

Inside the chord (within the brackets < >), before the note to be altered, place the `\tweak` command, followed by `font-size` and define the proper size like `#-2` (a tiny note head).

```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Modificar la tipografía para cada estrofa
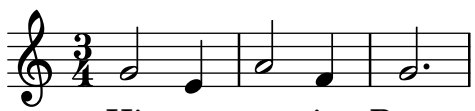
Se pueden cambiar las fuentes tipográficas de forma independiente para cada estrofa, incuida la fuente que se usa para imprimir el número de estrofa.

```
%{
You may have to install additional fonts.

Red Hat Fedora: dejavu-fonts-all

Debian GNU/Linux, Ubuntu: fonts-dejavu-core
                          fonts-dejavu-extra
%}

\relative c'' {
  \time 3/4
  g2 e4
  a2 f4
  g2.
}
\addlyrics {
  \stanza "1. "
  Hi, my name is Bert.
}
\addlyrics {
  \override StanzaNumber.fonts.serif = "DejaVu Sans"
  \stanza "2. "
  \override LyricText.font-family = #'typewriter
  Oh, ché -- ri, je t'aime
}
```

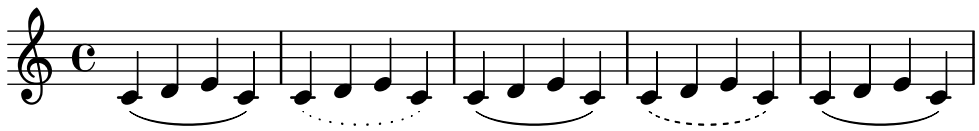


1. Hi, my name is Bert.
2. Oh, ché-ri, je t'aime

Modificar el aspecto continuo de una ligadura de expresión a punteado o intermitente

Se puede cambiar el aspecto de las ligaduras de expresión de continuas a punteadas o intermitentes.

```
\relative c' {
  c4( d e c)
  \slurDotted
  c4( d e c)
  \slurSolid
  c4( d e c)
  \slurDashed
  c4( d e c)
  \slurSolid
  c4( d e c)
}
```
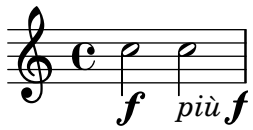


Combinar indicaciones dinámicas con marcados textuales

Ciertas indicaciones dinámicas pueden llevar textos (como "più *f*" o "*p* subito"). Se pueden producir usando un bloque `\markup`; el objeto resultante se comporta como un objeto gráfico `TextScript`.

See also "Combining dynamics with markup texts (2)".

```
piuF = \markup { \italic più \dynamic f }

\score {
  \relative c'' {
    c2\f c-\piuF
  }
}
```
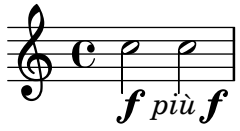


Combining dynamics with markup texts (2)

Some dynamics may involve text indications (such as "più *f*" or "*p* subito"). These can be produced using the `make-dynamic-script` Scheme function; the resulting object behaves like a `DynamicText` grob.

See also "Combining dynamics with markup texts".

```
piuF = #(make-dynamic-script
          #{ \markup { \normal-text \italic più \dynamic f } #})

\score {
  \relative c'' {
    c2\f c\piuF
  }
}
```

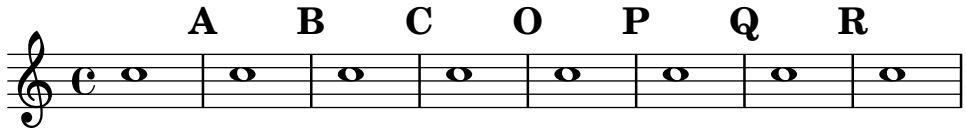


Forzar que las marcas de ensayo comiencen en una letra o número dados

Este fragmento de código muestra cómo obtener marcas de ensayo ordenadas automáticamente, pero partiendo de la letra o número deseados.

```
\relative c'' {
  \override Score.RehearsalMark.Y-offset = #3.5

  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1
}
```
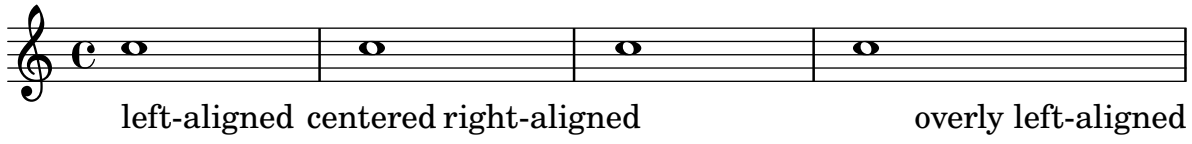


Alineación de la letra

La alineación horizontal de la letra se puede ajustar sobreescribiendo la propiedad `self-alignment-X` del objeto `LyricText`. El valor -1 significa alineado por la izquierda, 0 es centrado y 1 es derecha; de forma alternativa también podemos usar los valores de Scheme `#LEFT`, `#CENTER` y `#RIGHT` en lugar de números. También son posibles otros valores numéricos. Importante: no olvide añadir el prfijo de Scheme '`#`' para los números negativos.

```
\layout {
  ragged-right = ##f
}

\relative c'' {
  c1 c c c
}

\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "centered"
  \once \override LyricText.self-alignment-X = 1
  "right-aligned"
  \once \override LyricText.self-alignment-X = #-1.5
  "overly left-aligned"
}
```

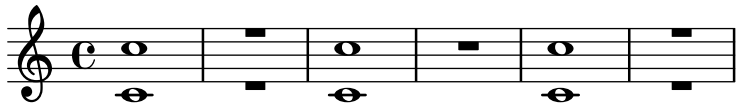


Combinar los silencios multicompás en una parte polifónica

Los silencios multicompás en un pentagrama polifónico se sitúan de forma diferente dependiendo de a qué voz pertenecen. Se pueden imprimir sobre la misma línea del pentagrama, usando el ajuste que aparece a continuación. Si omitimos la palabra clave `\once`, el cambio afecta a todos los silencios que siguen, dentro de la voz dada.

```
normalPos = \once \revert MultiMeasureRest.direction

<<
  { c''1 R c'' \normalPos R c'' R } \\
  { c'1 R c' \normalPos R c' R }
>>
```

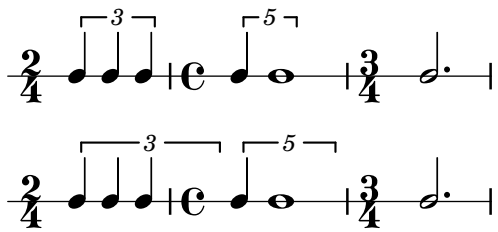


Modificar la longitud del corchete de grupo especial

Se puede hacer que los corchetes de grupo de valoración especial, como los tresillos, se extiendan horizontalmente hasta abarcar música anterior al propio grupo o incluyan la nota siguiente. Los corchetes predeterminados finalizan en el extremo derecho de la última nota del grupo; el corchete de un grupo de duración completa se exiende más a la derecha, bien para que cubra toda la notación no rítmica hasta la nota siguiente, o bien para que cubra solamente el espacio vacía que hay antes del elemento siguiente de notación, sea éste una clave, una indicación de compás, u otra nota. El ejemplo muestra cómo activar el modo de duración completa de los grupos de valoración especial y cómo cambiar la parte de música que cubren.

```
\new RhythmicStaff {
  % Defaults.
  \time 2/4 \tuplet 3/2 { c4 4 4 }
  \time 4/4 \tuplet 5/4 { 4 1 }
  \time 3/4 2.
}

\new RhythmicStaff {
  % Set tuplets to be extendable...
  \set tupletFullLength = ##t
  % ...to cover all items up to the next note
  \set tupletFullLengthNote = ##t
  \time 2/4 \tuplet 3/2 { c4 4 4 }
  % ...or to cover just whitespace.
  \set tupletFullLengthNote = ##f
  \time 4/4 \tuplet 5/4 { 4 1 }
  \time 3/4 2.
}
```



Imprimir el número de versión

Es posible imprimir el número de versión de LilyPond dentro de un elemento de marcado.

```
\markup { Processed with LilyPond version #(lilypond-version) }
```

Processed with LilyPond version 2.27.4

Plantilla de piano (sencilla)

Presentamos a continuación una plantilla de piano sencilla con algunas notas.
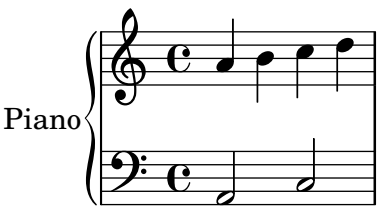
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  \new PianoStaff \with { instrumentName = "Piano" }
  <<
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}
```



Plantilla de piano con letra centrada

En lugar de tener un pentagrama dedicado a la melodía y la letra, ésta se puede centrar entre los pentagramas de un sistema de piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
```

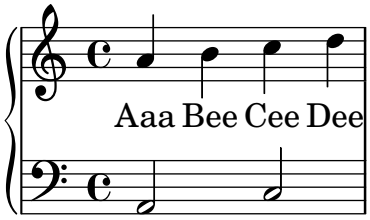
```
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}
```



Plantilla de piano con melodía y letra

He aquí el típico formato dde una canción: un pentagrama con la melodía y la letra, y el acompañamiento de piano por debajo.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

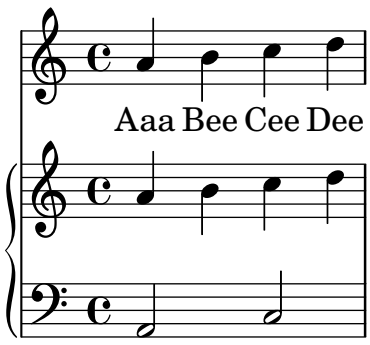
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4
```

```
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```



Plantilla de pentagrama único con música, letra y acordes

Esta plantilla facilita la preparación de una canción con melodía, letra y acordes.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

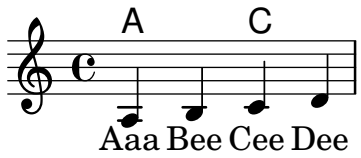
  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
```

```
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Plantilla de pentagrama único con música y acordes
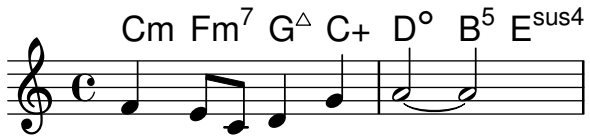
¿Quiere preparar una hoja guía de acordes (o «lead sheet») con melodía y acordes? ¡No busque más!

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  f4 e8[ c] d4 g |
  a2 ~ a
}

harmonies = \chordmode {
  c4:m f:min7 g:maj c:aug |
  d2:dim b4:5 e:sus
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Staff \melody
  >>
  \layout{ }
  \midi { }
}
```



Plantilla de pentagrama único don notas y letra
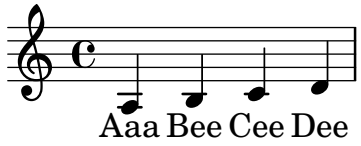
Esta pequeña plantilla muestra una melodía sencilla con letra. Córtela y péguela, escriba las notas y luego la letra. Este ejemplo desactiva el barrado automático, que es lo más frecuente en las partes vocales antiguas. Para usar el barrado automático modifique o marque como un comentario la línea correspondiente.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score{
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Plantilla de un solo pentagrama, con notas únicamente

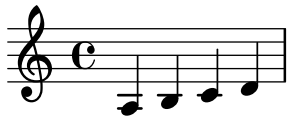
Esta plantilla simple prepara un pentagrama con notas, adecuado para un instrumento solista o un fragmento melódico. Córtelo y péguelo en un archivo, escriba las notas y ¡ya está!

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

\score {
```

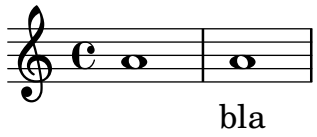
```
  \new Staff \melody
  \layout { }
  \midi { }
}
```



Silencios de separación en modo de letra

La sintaxis 's' para los silencios de desplazamiento solamente está disponible en los modos de nota y de acorde. En otras situaciones, por ejemplo al escribir la letra de las canciones, se recomienda usar la instrucción `\skip`.

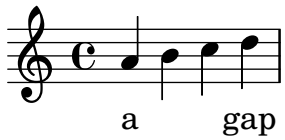
```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip1 bla1 }
>>
```



Silencios de separación en modo de letra (2)

Aunque no se pueden usar los silencios de separación 's' dentro de `\lyricmode` (se toman como una "s", literal, no como un espacio), sí están disponibles las comillas dobles (`""`) y el carácter subrayado (`_`).

```
<<
  \relative c'' { a4 b c d }
  \new Lyrics \lyricmode { a4 "" _ gap }
>>
```



Plantilla de cuarteto de cuerda (sencilla)

Esta plantilla muestra un cuarteto de cuerda normal. También utiliza una sección `\global` para el compás y la armadura.

Véase también el fragmento de código "Plantilla de cuarteto de cuerda con particellas independientes".

```
global= {
  \time 4/4
  \key c \major
}

violinOne = \new Voice \relative c'' {
  c2 d
  e1
```

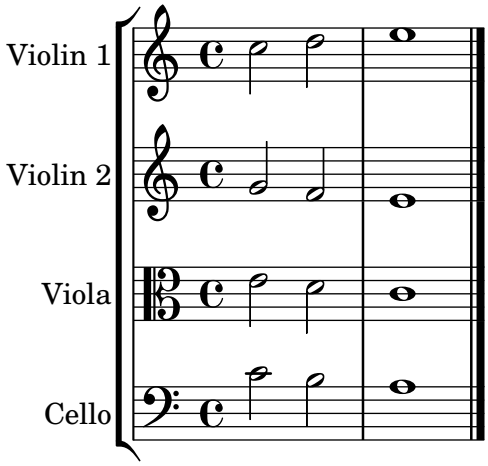
```
  \bar "|."
}

violinTwo = \new Voice \relative c'' {
  g2 f
  e1
  \bar "|."
}

viola = \new Voice \relative c' {
  \clef alto
  e2 d
  c1
  \bar "|."
}

cello = \new Voice \relative c' {
  \clef bass
  c2 b
  a1
  \bar "|."
}

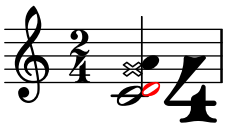
\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Violin 1" }
      << \global \violinOne >>
    \new Staff \with { instrumentName = "Violin 2" }
      << \global \violinTwo >>
    \new Staff \with { instrumentName = "Viola" }
      << \global \viola >>
    \new Staff \with { instrumentName = "Cello" }
      << \global \cello >>
  >>
  \layout { }
  \midi { }
}
```



Uso de la instrucción \tweak para retocar objetos gráficos individuales

Con la instrucción de trucar \tweak todos los objetos gráficos se pueden retocar directamente. Aquí presentamos ejemplos de los tipos de retoque disponibles.

```
\relative c' {
  \time 2/4
  \set fingeringOrientations = #'(right)
  <
    \tweak font-size #3 c
    \tweak color #red d-\tweak font-size #8 -4
    \tweak style #'cross g
    \tweak duration-log #2 a
  >2
}
```



Plantilla de conjunto vocal

He aquí una partitura vocal estándar para cuatro voces SATB. Con grupos mayores, suele ser útil incluir una sección que aparezca en todas las partes. Por ejemplo, el compás y la armadura casi siempre son los mismos para todas. Como en la "Plantilla para himnos", las cuatro voces se reagrupan en solo dos pentagramas.

```
\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
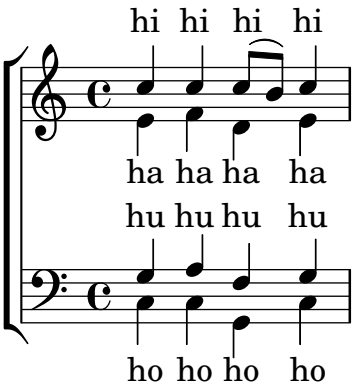
altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
```

```
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Lyrics = "sopranos" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "women" <<
      \new Voice = "sopranos" {
        \voiceOne
        << \global \sopMusic >>
      }
      \new Voice = "altos" {
        \voiceTwo
        << \global \altoMusic >>
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        << \global \tenorMusic >>
      }
      \new Voice = "basses" {
        \voiceTwo << \global \bassMusic >>
      }
    >>
    \new Lyrics = "basses"
    \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
    \context Lyrics = "altos" \lyricsto "altos" \altoWords
    \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
    \context Lyrics = "basses" \lyricsto "basses" \bassWords
  >>
}
```



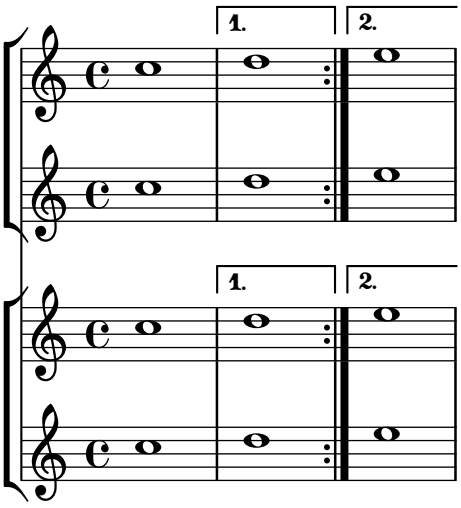
Casillas de primera y segunda vez en varios pentagramas

Añadiendo el grabador `Volta_engraver` al pentagrama deseado, las casillas de primera y segunda vez se pueden poner sobre pautas distintas a la del extremo superior de la partitura.

Las instrucciones `\repeat` y otras relacionadas deben estar presentes en todos los pentagramas.

```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}

<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```


1.
2.
1.
2.

32 Scheme

See also Sección “Scheme tutorial” en *Extending* and Sección “Interfaces for programmers” en *Extending*.

Incluir digitaciones adicionales con Scheme

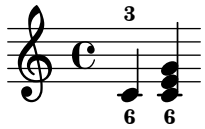
Se pueden añadir elementos adicionales a las notas usando `map-some-music`. En este ejemplo se ajunta una inscripción adicional a una nota.

En general se debe efectuar primero un `\displayMusic` de la música que queremos crear, para ver su estructura. Más tarde, podemos usar esto como plantilla para nuestro código de Scheme.

```
addScript =
#(define-music-function (script music) (ly:event? ly:music?)
   (map-some-music
    (lambda (mus)
      (define (append-script-at! prop)
        (set! (ly:music-property mus prop)
              (append (ly:music-property mus prop)
                      (list (ly:music-deep-copy script))))
        mus)

      (case (ly:music-property mus 'name)
        ((EventChord)
         (append-script-at! 'elements))
        ((NoteEvent)
         (append-script-at! 'articulations))
        (else #f)))
    music))

{
  \addScript _6 { c'4-3 <c' e' g'> }
}
```



Añadir indicadores a los pentagramas que se dividen después de un salto de línea

Este fragmento de código define las instrucciones `\splitStaffBarLine`, `\convUpStaffBarLine` y `\convDownStaffBarLine`, que añaden flechas en las sobre una línea divisoria para denotar que cada una de las voces que comparten un pentagrama continúan en su propio pentagrama en el siguiente sistema, o que las voces así divididas se recombinan.

Observe que la implementación de este fragmento de código traza unas flechas sin dimensiones en el margen derecho. Para la impresión formal, esto no produce ningún problema. Sin embargo, es necesario aumentar las dimensiones de la caja circundante en el sentido horizontal si se procesa el código como imagen para evitar el recorte, como se muestra abajo.

```
#(define-markup-command (arrow-at-angle layout props angle-deg length fill)
   (number? number? boolean?)
   (let* ((PI-OVER-180 (/ (atan 1 1) 34))
          (degrees->radians (lambda (degrees) (* degrees PI-OVER-180)))
          (angle-rad (degrees->radians angle-deg))
          (target-x (* length (cos angle-rad)))
          (target-y (* length (sin angle-rad))))
     (interpret-markup layout props
       (markup
        #:translate (cons (/ target-x 2) (/ target-y 2))
        #:rotate angle-deg
        #:translate (cons (/ length -2) 0)
        #:concat (#:draw-line (cons length 0)
                  #:arrow-head X RIGHT fill)))))

splitStaffBarLineMarkup = \markup \with-dimensions #'(0 . 0) #'(0 . 0) {
  \combine
  \arrow-at-angle #45 #(sqrt 8) ##t
  \arrow-at-angle #-45 #(sqrt 8) ##t
}

splitStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob splitStaffBarLineMarkup)
      0))
  \break
}

convDownStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . -.13)\arrow-at-angle #-45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

convUpStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
```

```
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . .14)\arrow-at-angle #45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

\paper {
  indent = 10\mm
  short-indent = 10\mm
  line-width = 8\cm
}

separateSopranos = {
  \set Staff.instrumentName = "AI AII"
  \set Staff.shortInstrumentName = "AI AII"
  \splitStaffBarLine
  \change Staff = "up"
}
convSopranos = {
  \convDownStaffBarLine
  \change Staff = "shared"
  \set Staff.instrumentName = "S A"
  \set Staff.shortInstrumentName = "S A"
}

sI = {
  \voiceOne
  \*4 f''2
  \separateSopranos
  \*4 g''2
  \convSopranos
  \*4 c''2
}
sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \*4 d''2
}
aI = {
  \voiceTwo
  \*4 a'2
  \voiceOne
  \*4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \*4 g'2
}
aII = {
  s1*2
```

```
  \voiceTwo
  \*4 g'2
}
ten = {
  \voiceOne
  \*4 c'2
  \*4 d'2
  \*4 c'2
}
bas = {
  \voiceTwo
  \*4 f2
  \*4 g2
  \*4 c2
}

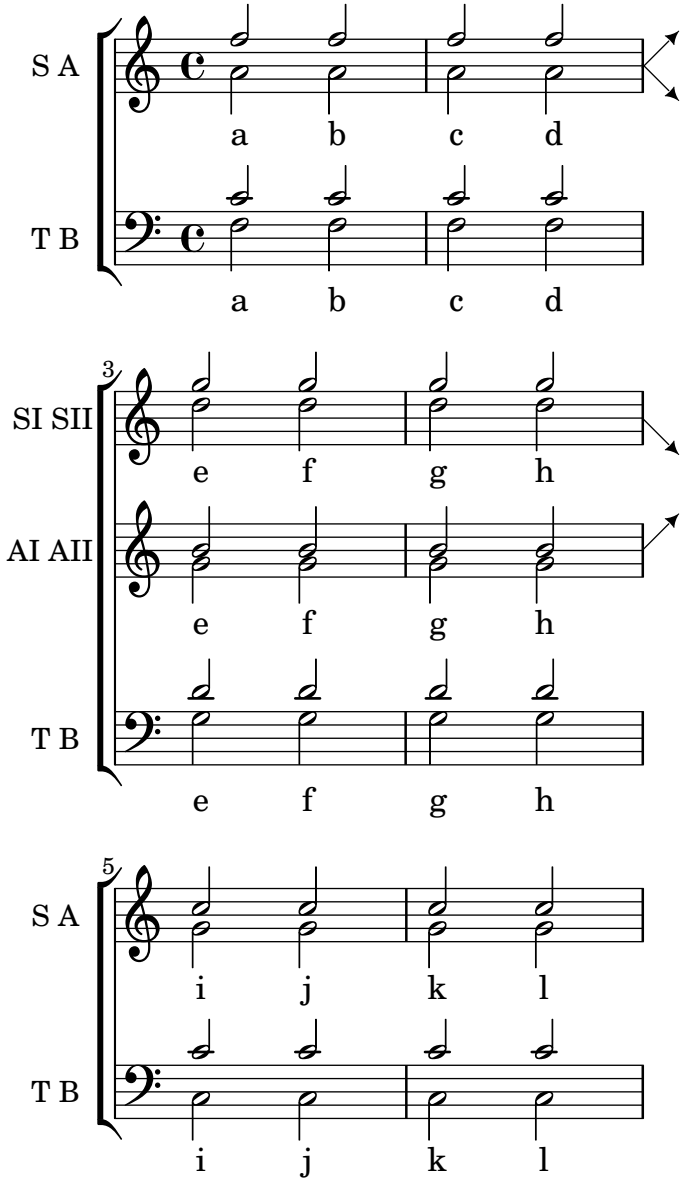
\markup \pad-x #3  % avoid cropping
  \score {
    <<
      \new ChoirStaff <<
        \new Staff = up \with {
          instrumentName = "SI SII"
          shortInstrumentName = "SI SII"
        } {
          s1*4
        }

        \new Staff = shared \with {
          instrumentName = "S A"
          shortInstrumentName = "S A"
        } <<
          \new Voice = sopI \sI
          \new Voice = sopII \sII
          \new Voice = altI \aI
          \new Voice = altII \aII
        >>
        \new Lyrics \with {
          alignBelowContext = up
        }
        \lyricsto sopII { e f g h }
        \new Lyrics \lyricsto altI { a b c d e f g h i j k l }

        \new Staff = men \with {
          instrumentName = "T B"
          shortInstrumentName = "T B"
        } <<
          \clef F
          \new Voice = ten \ten
          \new Voice = bas \bas
        >>
        \new Lyrics \lyricsto bas { a b c d e f g h i j k l }
      >>
```

```
    >>

    \layout {
      \context {
        \Staff \RemoveEmptyStaves
        \override VerticalAxisGroup.remove-first = ##t
      }
    }
}
```



Añadir enlaces a los objetos

Para añadir un enlace al sello de un objeto gráfico, podemos usar `add-link` tal y como se define aquí. Funciona con `\override` y con `\tweak`.

Inconveniente: `point-and-click` (apuntar y pulsar) quedará obstacuilzado por los objetos gráficos enlazados.

Limitación: funciona solamente para PDF.

Los objetos enlazados se colorean con una instrucción aparte.

```
#(define (add-link url-strg)
  (lambda (grob)
    (let* ((stil (ly:grob-property grob 'stencil)))
      (if (ly:stencil? stil)
          (let* ((x-ext (ly:stencil-extent stil X))
                 (y-ext (ly:stencil-extent stil Y))
                 (url-expr `(url-link ,url-strg ,x-ext ,y-ext))
                 (new-stil
                   (ly:stencil-add
                     (ly:make-stencil url-expr x-ext y-ext)
                     stil)))
          (ly:grob-set-property! grob 'stencil new-stil))))))

%%%% test

%% For easier maintenance of this snippet the URL is formatted to use the
%% actually used LilyPond version.
%% Of course a literal URL would work as well.

#(define major.minor-version
  (string-join (take (string-split (lilypond-version) #\.) 2) "."))

urlI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

urlII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/rhythms"
  major.minor-version)

urlIII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-heads"
  major.minor-version)

urlIV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/beams"
  major.minor-version)

urlV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-head-styles"
  major.minor-version)

urlVI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
```

```
  major.minor-version)
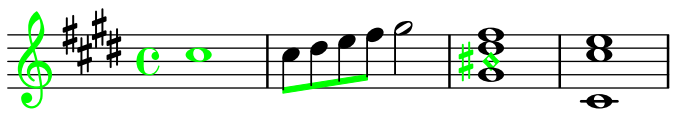
\relative c' {
   \key cis \minor

   \once \override Staff.Clef.color = #green
   \once \override Staff.Clef.after-line-breaking =
     #(add-link urlI)

   \once \override Staff.TimeSignature.color = #green
   \once \override Staff.TimeSignature.after-line-breaking =
     #(add-link urlII)

   \once \override NoteHead.color = #green
   \once \override NoteHead.after-line-breaking =
     #(add-link urlIII)

   cis'1
   \once \override Beam.color = #green
   \once \override Beam.after-line-breaking =
     #(add-link urlIV)
   cis8 dis e fis gis2
   <gis,
    \tweak Accidental.color #green
    \tweak Accidental.after-line-breaking #(add-link urlVI)
    \tweak color #green
    \tweak after-line-breaking #(add-link urlV)
    \tweak style #'harmonic
    bis
    dis
    fis
   >1
   <cis, cis' e>
}
```



Añadir notas guía orquestales a una partitura vocal

Este ejemplo muestra una forma de simplificar la adición de muchas notas guía orquestales a la reducción de piano en una partitura vocal. La función musical `\cueWhile` toma cuatro argumentos: la música de la que se toma la cita, como viene definida por `\addQuote`, el nombre que insertar antes de las notas guía, y después `UP` o `DOWN` para especificar `\voiceOne` con el nombre encima del pentagrama o bien `\voiceTwo` con el nombre debajo del pentagrama, y finalmente la música de piano con la que las notas guía deben aparecer en paralelo. El nombre del instrumento citado se posiciona a la izquierda de las notas guía. Se pueden citar muchos pasajes como guía, pero no se pueden superponer en el tiempo entre ellos.

```
cueWhile =
#(define-music-function
   (instrument name dir music)
   (string? string? ly:dir? ly:music?)
   #{
     \cueDuring $instrument #dir {
       \once \override TextScript.self-alignment-X = #RIGHT
       \once \override TextScript.direction = $dir
       <>-\markup { \tiny #name }
       $music
     }
   #})

flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }
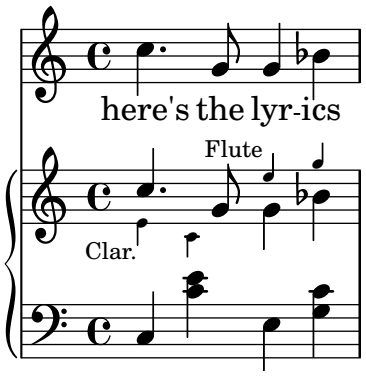
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
```

```
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```



Añadir la fecha actual a una partitura

Con algo de código de Scheme, se puede añadir fácilmente la fecha actual a una partitura.

```
\paper { tagline = ##f }
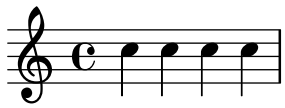
% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))

% use it in the title block:
\header {
  title = "Including the date!"
  subtitle = \date
}

\score {
  \relative c'' {
    c4 c c c
  }
}
% and use it in a \markup block:
\markup {
  \date
}
```

Including the date!

20-10-2026



20-10-2026

Ajustar las posiciones de las ligaduras verticalmente

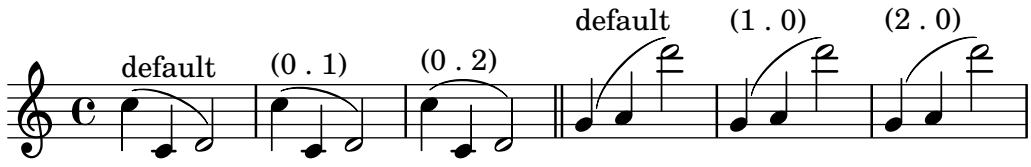
Usando `\override Slur.positions` es posible fijar la posición vertical de los puntos de inicio y de fin de una ligadura de expresión a valores absolutos (o mejor dicho, forzar al algoritmo de ligaduras de LilyPond para que tenga en cuenta estos valores según se desee). En muchos casos, esto implica bastante ensayo y error hasta que se encuentran unos valores aceptables. Porblablemente habrá probado la instrucción `\offset` solo para comprobar que no funciona con las ligaduras de expresión, sino que emite una advertencia en su lugar.

El código de este fragmento nos permite trucar las posiciones de comienzo y final especificando cambios *relativos*, de manera parecida a `\offset`.

La sintaxis es: `\offsetPositions #'(`*dy1 . dy2*`)`

```
offsetPositions =
#(define-music-function (offsets) (number-pair?)
  #{
     \once \override Slur.control-points =
       #(lambda (grob)
          (match-let ((((_ . y1) _ _ (_ . y2))
                       (ly:slur::calc-control-points grob))
                      ((off1 . off2) offsets))
            (set! (ly:grob-property grob 'positions)
                  (cons (+ y1 off1) (+ y2 off2)))
            (ly:slur::calc-control-points grob)))
  #})

\relative c'' {
  c4(^"default" c, d2)
  \offsetPositions #'(0 . 1)
  c'4(^"(0 . 1)" c, d2)
  \offsetPositions #'(0 . 2)
  c'4(^"(0 . 2)" c, d2)
  \bar "||"
  g4(^"default" a d'2)
  \offsetPositions #'(1 . 0)
  g,,4(^"(1 . 0)" a d'2)
  \offsetPositions #'(2 . 0)
  g,,4(^"(2 . 0)" a d'2)
}
```



Centrar texto debajo de un regulador

Este ejemplo proporciona una función para tipografiar un regulador con texto por debajo, como “molto” o “poco”. El texto añadido cambia la dirección de acuerdo con la del regulador. El objeto Hairpin está alineado con un objeto gráfico DynamicText.

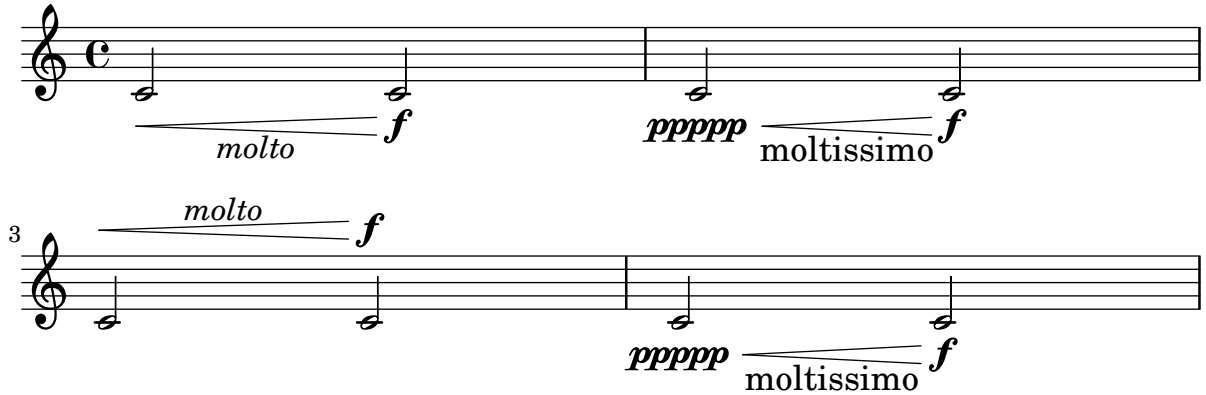
El ejemplo ilustra también cómo modificar la manera en que se imprime normalmente un objeto, utilizando código de Scheme.

```
hairpinWithCenteredText =
#(define-music-function (text) (markup?)
  #{
    \once \override Voice.Hairpin.after-line-breaking =
      #(lambda (grob)
         (let* ((stencil (ly:hairpin::print grob))
                (par-y (ly:grob-parent grob Y))
                (dir (ly:grob-property par-y 'direction))
                (staff-line-thickness
                 (ly:output-def-lookup (ly:grob-layout grob)
                                       'line-thickness))
                (new-stencil
                 (ly:stencil-aligned-to
                  (ly:stencil-combine-at-edge
                   (ly:stencil-aligned-to stencil X CENTER)
                   Y dir
                   (ly:stencil-aligned-to
                    (grob-interpret-markup
                     grob
                     (make-fontsize-markup
                      (magnification->font-size
                       (+ (ly:staff-symbol-staff-space grob)
                          (/ staff-line-thickness 2)))
                      text))
                    X CENTER))
                  X LEFT))
                (staff-space (ly:output-def-lookup
                 (ly:grob-layout grob) 'staff-space))
                (par-x (ly:grob-parent grob X))
                (dyn-text (grob::has-interface par-x
                                               'dynamic-text-interface))
                (dyn-text-stencil-x-length
                 (if dyn-text
                     (interval-length
                      (ly:stencil-extent
                       (ly:grob-property par-x 'stencil) X))
                     0))
                (x-shift
                 (if dyn-text (- (+ staff-space dyn-text-stencil-x-length)
                                 (* 0.5 staff-line-thickness))
                     0)))
           (ly:grob-set-property! grob 'Y-offset 0)
           (ly:grob-set-property! grob
                                  'stencil (ly:stencil-translate-axis
                                            new-stencil
                                            x-shift X))))
  #})

hairpinMolto = \hairpinWithCenteredText \markup { \italic molto }
hairpinMore = \hairpinWithCenteredText \markup { \larger moltissimo }

\relative c' {
```

```
  \hairpinMolto c2\< c\f
  \hairpinMore c2\ppppp\< c\f
  \break
  \hairpinMolto c2^\< c\f
  \hairpinMore c2\ppppp\< c\f
}
```



Modificación de propiedades para objetos gráficos individuales

La instrucción `\applyOutput` hace posible el ajuste fino de cualquier objeto de presentación, en cualquier contexto. Requiere una función de Scheme contres argumentos.

En el ejemplo de abajo, la función `mc-squared` se ejecuta para todos los objetos gráficos `NoteHead` (dentro del contexto `Voice` actual) en el punto temporal en curso; la función modifica el `stencil` o sello del objeto gráfico, usando la propiedad `staff-position` para sustituir ciertas alturas con elementos de marcado.

Consulte el 'Manual de extensión del programa' (`https://lilypond.org/doc/v2.24/Documentation/extending/running-a-function-on-all-layout-objects`) para más información.

```
#(define (mc-squared grob grob-origin context)
   (let ((sp (ly:grob-property grob 'staff-position)))
     (ly:grob-set-property!
      grob 'stencil
      (grob-interpret-markup grob
       #{ \markup \lower #0.5
           #(case sp
              ((-5) "m")
              ((-3) "c ")
              ((-2) #{ \markup \teeny \bold 2 #})
              (else "bla")) #}))))

\relative c' {
  <d f g b>2
  \applyOutput Voice.NoteHead #mc-squared
  <d f g b>2
}
```



Excepciones para los nombres de acorde

La propiedad `chordNameExceptions` almacena una lista de excepciones para los nombres de acorde, para tratar casos que no están cubiertos, o bien que se tratan incorrectamente.

Los nombres de acorde predeterminados que se usan por LilyPond siguen las reglas dadas en el libro de Klaus Ignatzek's "Die Jazzmethode für Klavier 1"; el algoritmo para convertir acordes en nombres de acorde puede verse dentro del archivo `scm/chord-ignatzek-names.scm`. Las reglas adicionales se dan como excepciones de acordes y se almacenan dentro de la variable `ignatzekExceptions`, como está establecido en el archivo `ly/chord-modifiers-init.ly`.

Este fragmento de código modifica estas excepcioens en tres pasos.

1. Preparación de una música con acordes y elementos de marcado asociados. Por convención, la fundamental de cada acorde debe ser Do.
2. Llamar a la función de Scheme `sequential-music-to-chord-exceptions` para crear una lista de excepciones nueva, y después concatenarla con las existentes. Como `ignatzekExceptions` está preparado con el segundo parámetro de su función fijado al valor `#t` (para ignorar la fundamental de los acordes), tenemos que hacer lo mismo.
3. Registrar la nueva lista de excepciones.

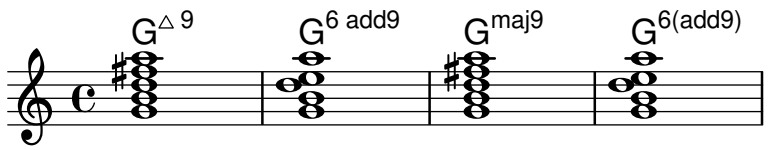
```
% Step 1: Define music with chords and markup for maj9 and 6(add9).
chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

% Step 2: Create extended exception list.
chExceptions =
#(append (sequential-music-to-chord-exceptions chExceptionMusic #t)
         ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  % Step 3: Register extended exception list.
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

<<
  \new ChordNames \theMusic
  \new Voice \theMusic
>>

\layout {
  line-width = 10\cm
  ragged-right = ##f
}
```



Colorear las notas según su altura

Es posible colorear la cabeza de las notas dependiendo de su altura y/o de su nombre: la función que se usa en este ejemplo hace posible incluso distinguir los armónicos.
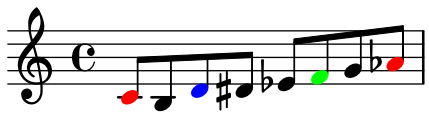
```
% Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 2 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 3 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 3 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 4 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 5 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 5 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 6 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 1 NATURAL) (x11-color 'blue))
    (cons (ly:make-pitch 0 3 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 4 FLAT) (x11-color 'blue))
    (cons (ly:make-pitch 0 5 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 6 FLAT) (x11-color 'blue))))

% Compare pitch and alteration (not octave).
#(define (pitch-equals? p1 p2)
   (and
    (= (ly:pitch-alteration p1) (ly:pitch-alteration p2))
    (= (ly:pitch-notename p1) (ly:pitch-notename p2))))

#(define (pitch-to-color pitch)
   (let ((color (assoc pitch color-mapping pitch-equals?)))
     (if color
         (cdr color))))

#(define (color-notehead grob)
   (pitch-to-color
    (ly:event-property (event-cause grob) 'pitch)))

\score {
  \new Staff \relative c' {
    \override NoteHead.color = #color-notehead
    c8 b d dis ees f g aes
  }
}
```
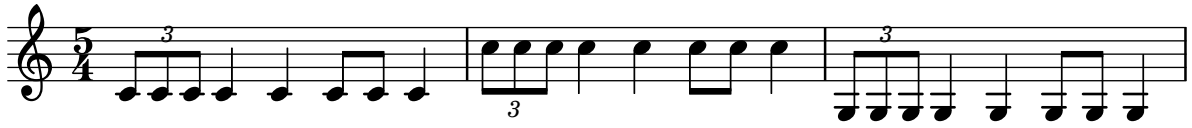


Crear una secuencia de notas a distintas alturas

En una música que tenga muchas apariciones de la misma secuencia de notas a distintas alturas, podría ser de utilidad la siguiente función musical. Admite una nota, de la que solo se utiliza su altura. Las funciones de apoyo en Scheme se han tomado prestadas del documento de "Consejos y trucos"de la versión 2.10 del manual.

Este ejemplo crea las duraciones rítmicas que se usan a todo lo largo de *Marte*, de *Los Planetas* de Gustav Holst.

```
rhythm =
#(define-music-function (p) (ly:pitch?)
   "Make the rhythm in Mars (the Planets) at the given pitch"
  #{ \tuplet 3/2 { $p 8 8 8 } 4 4 8 8 4 #})

\new Staff {
  \time 5/4
  \rhythm c'
  \rhythm c''
  \rhythm g
}
```



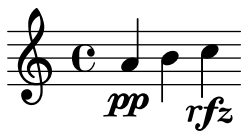
Creacaión de matices dinámicos personalizados en el MIDI

El ejemplo siguiente muestra cómo crear una marda de matiz dinámico, no incluida en la lista predeterminada, y asignarle un valor específico de forma que afecte a la salida MIDI.

El matiz \rfz conlleva el valor 0.9.

```
#(define (myDynamics dynamic)
    (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = "cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative {
        a'4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}
```



Personalizar la posición y el número de puntos en las líneas divisorias de repetición

Si queremos personalizar la posición y/o el número de puntos en las barras de repetición, podemos definir líneas divisorias personalizadas o redefinir la manera en que se dibujan los símbolos de repetición. Esto puede ser de especial ayuda cuando se usa un pentagrama con posiciones de línea personalizadas, como se muestra en el fragmento.

```
#(define ((make-custom-dot-bar-line dot-positions) is-span grob extent)
   "Draw dots (repeat sign dots) at DOT-POSITIONS.

The coordinates of DOT-POSITIONS are equivalent to the coordinates of
`StaffSymbol.line-positions`; a dot position of X and a line position
of X indicate the same vertical position.

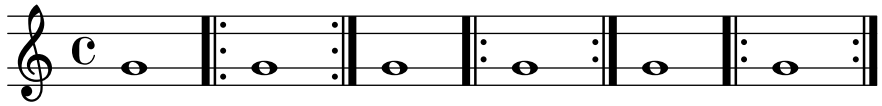
IS-SPAN is not used in this custom function."
   (let* ((staff-space (ly:staff-symbol-staff-space grob))
          (dot (ly:font-get-glyph (ly:grob-default-font grob)
               "dots.dot"))
          (stencil empty-stencil))
     (for-each
      (lambda (dp)
        (set! stencil (ly:stencil-add stencil
                       (ly:stencil-translate-axis
                        dot (* dp (/ staff-space 2)) Y))))
      dot-positions)
     stencil))

% With the procedure above we can define custom bar lines, for example,
% that resemble standard repeat sign bar lines except that there are
% three dots at staff positions -3, 0, and 3.

#(add-bar-glyph-print-procedure "*" (make-custom-dot-bar-line '(-3 0 3)))
\defineBarLine ".|*" #'("" "*" "")
\defineBarLine "*|." #'("" "*" "")

% We can also customize the dot positions used in all default repeat
% signs by redefining the print procedure of the colon bar glyph (":").
% On a staff with line positions of `(-4 -2 2 4)`, the default repeat
% sign dots appear at `(-3 3)`, but we can put them at `(-1 1)` instead.
#(add-bar-glyph-print-procedure ":" (make-custom-dot-bar-line '(-1 1)))


\new Staff \with {
  \override StaffSymbol.line-positions = #'(-4 -2 2 4)
  \override StaffSymbol.staff-space = #1.3
} \relative f' {
  g1 \bar ".|*"
  g \bar "*|."
  g \bar ".|:-|"
  g \bar ":|."
  g |
  \repeat volta 2 { g }
}
```



Definir un grabador en Scheme: grabador de ámbito o tesitura

Este ejemplo muestra cómo puede definirse el grabador de ámbito o tesitura en el espacio del usuario, con un grabador de Scheme. Esto es básicamente una reescritura en Scheme del código de `lily/ambitus-engraver.cc`.

```
#(use-modules (oop goops))

%%%
%%% Grob utilities
%%%
%%% These are literal rewrites of some C++ methods used by the ambitus
%%% engraver.

#(define (ly:separation-item::add-conditional-item grob grob-item)
   "Add GROB-ITEM to the array of conditional elements of GROB.

This is a rewrite of function `Separation_item::add_conditional_item` from
file `lily/separation-item.cc`."
   (ly:pointer-group-interface::add-grob
    grob 'conditional-elements grob-item))

#(define (ly:accidental-placement::accidental-pitch accidental-grob)
   "Get the pitch from the grob cause of ACCIDENTAL-GROB.

This is a rewrite of function `accidental_pitch` from file
`lily/accidental-placement.cc`."
   (ly:event-property (ly:grob-property
                       (ly:grob-parent accidental-grob Y) 'cause)
                      'pitch))

#(define (ly:accidental-placement::add-accidental grob accidental-grob)
   "Add ACCIDENTAL-GROB to the list of accidentals grobs of GROB.
ACCIDENTAL-GROB is an `Accidental` grob; GROB is an `AccidentalPlacement`
grob.

This is a rewrite of function `Accidental_placement::add_accidental` from
file `lily/accidental-placement.cc`."
   (let ((pitch (ly:accidental-placement::accidental-pitch
                 accidental-grob)))
     (set! (ly:grob-parent accidental-grob X) grob)
     (let* ((accidentals (ly:grob-object grob 'accidental-grobs))
            (handle (assq (ly:pitch-notename pitch) accidentals))
            (entry (if handle (cdr handle) '())))
       (set! (ly:grob-object grob 'accidental-grobs)
             (assq-set! accidentals
                        (ly:pitch-notename pitch)
                        (cons accidental-grob entry))))))
```

```
%%%
%%% Ambitus data structures.
%%%

%%% The <ambitus> class holds the various grobs that are created to
%%% print an ambitus:
%%%
%%% - `ambitus-group`: the grob that groups all the components of an
%%%   ambitus (`Ambitus` grob);
%%% - `ambitus-line`: the vertical line between the upper and lower
%%%   ambitus notes (`AmbitusLine` grob);
%%% - `ambitus-up-note` and `ambitus-down-note`: the note head and
%%%   accidental for the lower and upper note of the ambitus (see
%%%   `<ambitus-note>` class below).
%%%
%%% The other slots define the key and clef context of the engraver:
%%%
%%% - `start-c0`: position of middle c at the beginning of the piece.
%%%   It is used to place the ambitus notes according to their pitch;
%%% - `start-key-sig`: the key signature at the beginning of the
%%%   piece.  It is used to determine whether accidentals shall be
%%%   printed next to ambitus notes.

#(define-class <ambitus> ()
   (ambitus-group #:accessor ambitus-group)
   (ambitus-line #:accessor ambitus-line)
   (ambitus-up-note #:getter ambitus-up-note
                    #:init-form (make <ambitus-note>))
   (ambitus-down-note #:getter ambitus-down-note
                      #:init-form (make <ambitus-note>))
   (start-c0 #:accessor ambitus-start-c0
             #:init-value #f)
   (start-key-sig #:accessor ambitus-start-key-sig
                  #:init-value '()))

%%% Accessor for the lower and upper note data of an ambitus.
#(define-method (ambitus-note (ambitus <ambitus>) direction)
   "Return lower or upper note of AMBITUS depending on DIRECTION."
   (if (= direction UP)
       (ambitus-up-note ambitus)
       (ambitus-down-note ambitus)))

%%% The `<ambitus-note>` class holds the grobs that are specific to
%%% ambitus (lower and upper) notes:
%%%
%%% - `head`: an `AmbitusNoteHead` grob;
%%% - `accidental`: an `AmbitusAccidental` grob, to be possibly
%%%   printed next to the ambitus note head.
%%%
%%% Moreover,
%%%
%%% - `pitch` is the absolute pitch of the note;
```

```
%%% - `cause` is the note event that causes this ambitus note, i.e.,
%%%   the lower or upper note of the considered music sequence.

#(define-class <ambitus-note> ()
   (head #:accessor ambitus-note-head
         #:init-value #f)
   (accidental #:accessor ambitus-note-accidental
               #:init-value #f)
   (cause #:accessor ambitus-note-cause
          #:init-value #f)
   (pitch #:accessor ambitus-note-pitch
          #:init-value #f))

%%%
%%% Ambitus engraving logic.
%%%
%%% This is rewrite of the code from file `lily/ambitus-engraver.cc`.

#(define (make-ambitus translator)
   "Build an ambitus object: initialize all the grobs and their
relations.

The `Ambitus` grob contains all other grobs:

 Ambitus
  |- AmbitusLine
  |- AmbitusNoteHead    for upper note
  |- AmbitusAccidental  for upper note
  |- AmbitusNoteHead    for lower note
  |- AmbitusAccidental  for lower note

The parent of an accidental is the corresponding note head, and the
accidental is set as the `accidental-grob` property of the note head
so that is printed by the function that prints notes."
   ;; Make the ambitus object.
   (let ((ambitus (make <ambitus>)))
     ;; Build the `Ambitus` grob, which will contain all other grobs.
     (set! (ambitus-group ambitus)
           (ly:engraver-make-grob translator 'Ambitus '()))
     ;; Build the `AmbitusLine` grob (the line between lower and upper
     ;; note).
     (set! (ambitus-line ambitus)
           (ly:engraver-make-grob translator 'AmbitusLine '()))
     ;; Build the upper and lower `AmbitusNoteHead` and
     ;; `AmbitusAccidental`.
     (for-each
      (lambda (direction)
        (let ((head (ly:engraver-make-grob translator
                                           'AmbitusNoteHead '()))
              (accidental (ly:engraver-make-grob translator
                                                 'AmbitusAccidental '()))
              (group (ambitus-group ambitus)))
```

```
          ;; The parent of the `AmbitusAccidental` grob is the
          ;; `AmbitusNoteHead` grob.
          (set! (ly:grob-parent accidental Y) head)
          ;; The `AmbitusAccidental` grob is set as the
          ;; `accidental-grob` object of `AmbitusNoteHead`.  This is
          ;; later used by the function that prints notes.
          (set! (ly:grob-object head 'accidental-grob) accidental)
          ;; Both the note head and the accidental grobs are added to
          ;; the main ambitus grob.
          (ly:axis-group-interface::add-element group head)
          (ly:axis-group-interface::add-element group accidental)
          ;; The note head and the accidental grobs are added to the
          ;; ambitus object.
          (set! (ambitus-note-head (ambitus-note ambitus direction))
                head)
          (set! (ambitus-note-accidental (ambitus-note ambitus direction))
                accidental)))
      (list DOWN UP))

     ;; The parent of the ambitus line is the lower ambitus note head.
     (set! (ly:grob-parent (ambitus-line ambitus) X)
           (ambitus-note-head (ambitus-note ambitus DOWN)))
     ;; The ambitus line is added to the ambitus main grob.
     (ly:axis-group-interface::add-element (ambitus-group ambitus)
                                           (ambitus-line ambitus))
     ambitus))

#(define-method (initialize-ambitus-state
                  (ambitus <ambitus>) translator)
   "Initialize the state of AMBITUS by getting the starting position of
middle C and key signature from TRANSLATOR's context."
   (if (not (ambitus-start-c0 ambitus))
       (begin
         (set! (ambitus-start-c0 ambitus)
               (ly:context-property (ly:translator-context translator)
                                    'middleCPosition 0))
         (set! (ambitus-start-key-sig ambitus)
               (ly:context-property (ly:translator-context translator)
                                    'keyAlterations)))))

#(define-method (update-ambitus-notes (ambitus <ambitus>) note-grob)
   "Update upper and lower ambitus pitches of AMBITUS using NOTE-GROB."
   ;; Get the event that caused the `note-grob` creation and check
   ;; that it is a `note-event`.
   (let ((note-event (ly:grob-property note-grob 'cause)))
     (if (ly:in-event-class? note-event 'note-event)
         ;; Get the pitch from the note event.
         (let ((pitch (ly:event-property note-event 'pitch)))
           ;; If this pitch is lower than the current ambitus' lower
           ;; note pitch (or it has not been initialized yet), then
           ;; this pitch is the new ambitus' lower pitch.  The same is
           ;; done for the upper pitch (but in the opposite
```

```
           ;; direction).
           (for-each
            (lambda (direction pitch-compare)
              (if (or (not (ambitus-note-pitch
                            (ambitus-note ambitus direction)))
                      (pitch-compare
                       pitch (ambitus-note-pitch
                              (ambitus-note ambitus direction))))
                  (begin
                    (set! (ambitus-note-pitch
                           (ambitus-note ambitus direction))
                          pitch)
                    (set! (ambitus-note-cause
                           (ambitus-note ambitus direction))
                          note-event))))
            (list DOWN UP)
            (list ly:pitch<?
                  (lambda (p1 p2) (ly:pitch<? p2 p1))))))))

#(define-method (typeset-ambitus (ambitus <ambitus>) translator)
   "Typeset AMBITUS.

- Place the lower and upper ambitus notes according to their pitch and
  the position of the middle C.
- Typeset or delete the note accidentals, according to the key
  signature.  An accidental, if it is to be printed, is added to an
  `AccidentalPlacement` grob (a grob dedicated to the placement of
  accidentals near a chord).
- Both note heads are added to the ambitus line grob so that a line
  gets printed between them."
   ;; Check whether there are lower and upper pitches.
   (if (and (ambitus-note-pitch (ambitus-note ambitus UP))
            (ambitus-note-pitch (ambitus-note ambitus DOWN)))
       ;; Make an `AccidentalPlacement` grob, for placement of note
       ;; accidentals.
       (let ((accidental-placement
              (ly:engraver-make-grob
               translator
               'AccidentalPlacement (ambitus-note-accidental
                                     (ambitus-note ambitus DOWN)))))
         ;; For lower and upper ambitus notes.
         (for-each
          (lambda (direction)
            (let ((pitch (ambitus-note-pitch
                          (ambitus-note ambitus direction))))
              ;; Set the cause and the staff position of the ambitus
              ;; note according to the associated pitch.
              (set! (ly:grob-property
                     (ambitus-note-head (ambitus-note ambitus direction))
                     'cause)
                    (ambitus-note-cause (ambitus-note ambitus direction)))
              (set! (ly:grob-property
```

```
              (ambitus-note-head (ambitus-note ambitus direction))
              'staff-position)
             (+ (ambitus-start-c0 ambitus)
                (ly:pitch-steps pitch)))
        ;; Determine whether an accidental shall be printed for
        ;; this note, according to the key signature.
        (let* ((handle
                (or (assoc (cons (ly:pitch-octave pitch)
                                 (ly:pitch-notename pitch))
                           (ambitus-start-key-sig ambitus))
                    (assoc (ly:pitch-notename pitch)
                           (ambitus-start-key-sig ambitus))))
               (sig-alter (if handle (cdr handle) 0)))
          (cond
           ((= (ly:pitch-alteration pitch) sig-alter)
            ;; The note alteration is in the key signature
            ;; => it does not have to be printed.
            (ly:grob-suicide! (ambitus-note-accidental
                               (ambitus-note ambitus direction)))
            (set! (ly:grob-object (ambitus-note-head
                                   (ambitus-note ambitus direction))
                                  'accidental-grob)
                  '()))
           (else
            ;; Otherwise the accidental shall be printed.
            (set! (ly:grob-property
                   (ambitus-note-accidental
                    (ambitus-note ambitus direction)) 'alteration)
                  (ly:pitch-alteration pitch)))))
        ;; Add the `AccidentalPlacement` grob to the conditional
        ;; items of the `AmbitusNoteHead`.
        (ly:separation-item::add-conditional-item
         (ambitus-note-head (ambitus-note ambitus direction))
         accidental-placement)
        ;; Add the `AmbitusAccidental` to the list of the
        ;; `AccidentalPlacement` grob accidentals.
        (ly:accidental-placement::add-accidental
         accidental-placement
         (ambitus-note-accidental (ambitus-note ambitus direction)))
        ;; Add the `AmbitusNoteHead` grob to the `AmbitusLine` grob.
        (ly:pointer-group-interface::add-grob
         (ambitus-line ambitus)
         'note-heads
          (ambitus-note-head (ambitus-note ambitus direction)))))
   (list DOWN UP))
  ;; Add the `AccidentalPlacement` grob to the main `Ambitus` grob.
  (ly:axis-group-interface::add-element
   (ambitus-group ambitus) accidental-placement))
;; No lower and upper pitches => nothing to print.
(begin
  (for-each
   (lambda (direction)
```

```
            (ly:grob-suicide! (ambitus-note-accidental
                               (ambitus-note ambitus direction)))
            (ly:grob-suicide! (ambitus-note-head
                               (ambitus-note ambitus direction))))
          (list DOWN UP))
         (ly:grob-suicide! ambitus-line))))

%%%
%%% Ambitus engraver definition.
%%%
#(define ambitus-engraver
   (lambda (context)
     (let ((ambitus #f))
       ;; When music is processed, make the ambitus object if not
       ;; already built.
       (make-engraver
        ((process-music translator)
         (if (not ambitus)
             (set! ambitus (make-ambitus translator))))
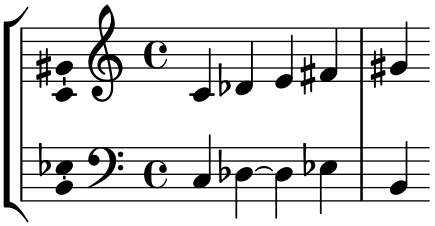
        ;; Set the ambitus clef and key signature state.
        ((stop-translation-timestep translator)
         (if ambitus
             (initialize-ambitus-state ambitus translator)))

        ;; When a note head grob is built, update the ambitus notes.
        (acknowledgers
         ((note-head-interface engraver grob source-engraver)
          (if ambitus
              (update-ambitus-notes ambitus grob))))

        ;; Finally, typeset the ambitus according to its upper and
        ;; lower notes (if any).
        ((finalize translator)
         (if ambitus
             (typeset-ambitus ambitus translator)))))))

%%%
%%% Example
%%%

\score {
  \new StaffGroup <<
    \new Staff { c'4 des' e' fis' gis' }
    \new Staff { \clef "bass" c4 des ~ des ees b, }
  >>
  \layout { \context { \Staff \consists #ambitus-engraver } }
}
```



Diferentes tamaños de fuente para instrumentName y shortInstrumentName

Elegir distintos tamaños de fuente para instrumentName y shortInstrumentName como una sobreescritura de contexto.

```
InstrumentNameFontSize =
#(define-music-function (font-size-pair) (pair?)
   "Set the font size of `InstrumentName` grobs.

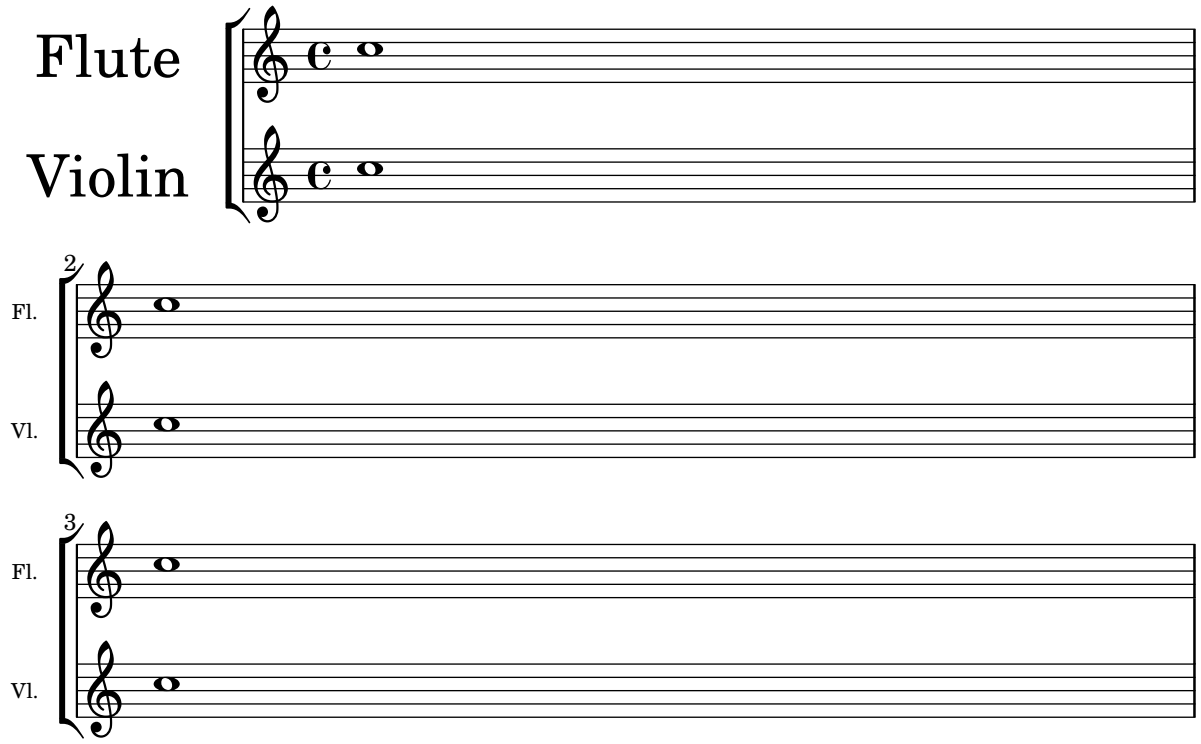
The first value of FONT-SIZE-PAIR sets the font size of the initial
`instrumentName` property, the second value sets the font size of
`shortInstrumentName`."
   ;; This code could be changed or extended to set different values
   ;; for each occurrence of `shortInstrumentName'.
   #{
     \override InstrumentName.after-line-breaking =
       #(lambda (grob)
          (let* ((orig (ly:grob-original grob))
                 (siblings (if (ly:grob? orig)
                               (ly:spanner-broken-into orig)
                               '())))
            (when (pair? siblings)
              (ly:grob-set-property! (car siblings)
                                     'font-size (car font-size-pair))
              (for-each
               (lambda (g)
                 (ly:grob-set-property! g
                                        'font-size (cdr font-size-pair)))
               (cdr siblings)))))
   #})

\layout {
  indent = 3\cm
  short-indent = 0.8\cm

  \context {
    \Staff
    \InstrumentNameFontSize #'(6 . -3)
  }
}

\new StaffGroup <<
  \new Staff \with {
       instrumentName = "Flute"
```

```
      shortInstrumentName = "Fl." } {
    c''1 \break c'' \break c'' }
  \new Staff \with {
      instrumentName = "Violin"
      shortInstrumentName = "Vl." } {
    c''1 \break c'' \break c'' }
>>
```



Imprimir el árbol genealógico de un grob

Al trabajar con los callbacks de un grob, puede ser de mucha ayuda entender el árbol genealógico de un grob. La mayor parte de los grobs tienen padres que influyen en el posicionamiento del grob. los padres X e Y influyen en las posiciones horizontal y vertical del grob, respectivamente. Además, cada pade puede tener padres a su vez.

Por desgracia, existen varios aspectos de la genealogía de un grob que pueden llevar a confusión:

- Los tipos de padre que tiene un grob pueden depender del contexto.
- Para ciertos grobs, los padres X e Y son el mismo.
- Un *ancestro* concreto puede estar relacionado con un grob de más de una manera.
- El concepto de *generaciones* es engañoso.

Por ejemplo, el grob `System` puede ser tanto un padre (sobre el lado Y) como un abuelo (dos veces en el lado X) de un grob `VerticalAlignment`.

El macro definido en este fragmento de código imprime en la consola una representación textual de la genealogía de un grob. Por ejemplo, la llamada

```
{
  \once \override NoteHead.before-line-breaking = #display-ancestry
  c
}
```

genera la siguiente salida.

```
-----------------------------------
NoteHead
X,Y: NoteColumn
     X: PaperColumn
        X,Y: System
     Y: VerticalAxisGroup
        X: NonMusicalPaperColumn
           X,Y: System
        Y: VerticalAlignment
           X: NonMusicalPaperColumn
              X,Y: System
           Y: System
```

Como consecuencia, tenemos que ejecutar el código del fragmento de código por nosotros mismos, porque el archivo de salida no muestra los datos que nos interesan.
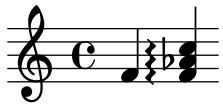
```
#(define (get-ancestry grob)
   (if (not (null? (ly:grob-parent grob X)))
       (list (grob::name grob)
             (get-ancestry (ly:grob-parent grob X))
             (get-ancestry (ly:grob-parent grob Y)))
       (grob::name grob)))

#(define (format-ancestry lst padding)
   (string-append
    (symbol->string (car lst)) "\n"
    (let ((X-ancestry (if (list? (cadr lst))
                          (format-ancestry (cadr lst) (+ padding 3))
                          (symbol->string (cadr lst))))
          (Y-ancestry (if (list? (caddr lst))
                          (format-ancestry (caddr lst) (+ padding 3))
                          (symbol->string (caddr lst)))))
      (if (equal? X-ancestry Y-ancestry)
          (string-append (format #f "~&")
                         (make-string padding #\space)
                         "X,Y: "
                         (if (list? (cadr lst))
                             (format-ancestry (cadr lst) (+ padding 5))
                             (symbol->string (cadr lst))))
          (string-append (format #f "~&")
                         (make-string padding #\space)
                         "X: " X-ancestry "\n"
                         (make-string padding #\space)
                         "Y: " Y-ancestry (format #f "~&"))))
    (format #f "~&")))

#(define (display-ancestry grob)
   (format (current-output-port)
           "~2&~a~2%~a~&"
           (make-string 36 #\-)
           (if (ly:grob? grob)
               (format-ancestry (get-ancestry grob) 0)
```

```
              (format #f "~a is not a grob" grob))))

\relative c' {
  \once \override NoteHead.before-line-breaking = #display-ancestry
  f4
  \once \override Accidental.before-line-breaking = #display-ancestry
  \once \override Arpeggio.before-line-breaking = #display-ancestry
  <f as c>4\arpeggio
}
```
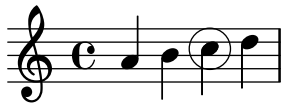


Trazar circunferencias alrededor de la cabeza de las notas

Se puede trazar una circunferencia alrededor de la cabeza de una nota aportando una función de Scheme personalizada para sobreescribir temporalmente la propiedad `stencil`.

```
circle = \tweak NoteHead.stencil
  #(lambda (grob)
    (let* ((note (ly:note-head::print grob))
           (combo-stencil (ly:stencil-add
               note
               (circle-stencil note 0.1 0.8))))
          (ly:make-stencil (ly:stencil-expr combo-stencil)
            (ly:stencil-extent note X)
            (ly:stencil-extent note Y))))
  \etc

{ a' b' \circle c'' d'' }
```



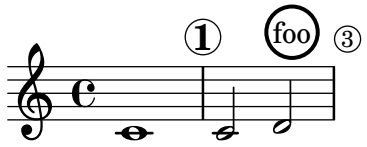
Rodear diversos objetos con una circunferencia

La instrucción `\circle` traza circunferencias alrededor de los objetos de marcado creados con `\markup`. Para otros objetos pueden requerirse trucos específicos, como se muestra aquí para el caso de las letras de ensayo y los números de compás.

```
\relative c' {
  c1
  \set Score.rehearsalMarkFormatter =
    #(lambda (mark context)
              (make-circle-markup (format-mark-numbers mark context)))
  \mark \default

  c2 d^\markup {
    \override #'(thickness . 3) {
      \circle foo
    }
  }
  \override Score.BarNumber.break-visibility = #all-visible
```

```
  \override Score.BarNumber.stencil =
    #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
}
```



Extender glissandos sobre repeticiones

Se puede simular un glissando que se extiende hasta el interior de varios bloques `\alternative` de primera y segunda vez mediante la adición de una nota de adorno oculta con un glissando al comienzo de cada bloque `\alternative`. La nota de adorno debe estar a la misma altura que la nota que da inicio al primer glissando. Esto se implementa aquí con una función musical que toma como argumento la altura de la nota de adorno.

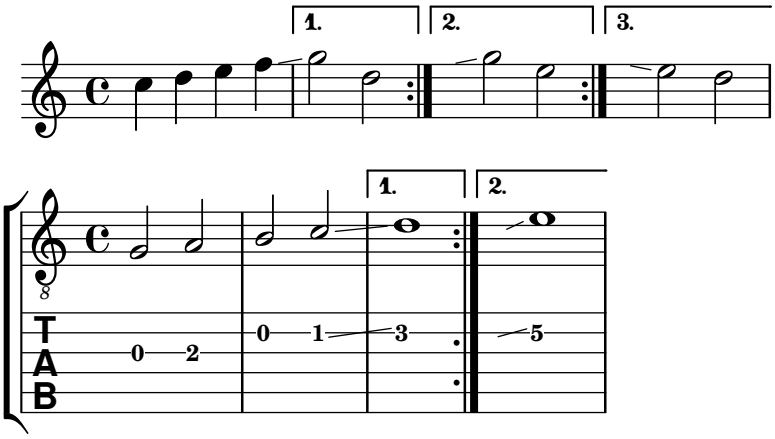
Observe que en música polifónica la nota de adorno debe coincidir con las notas de adorno correspondientes en todas las otras voces.

```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music =  \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}
```

```
\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff  <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
}
```



Ligaduras planas

Este fragmento de código aporta una función `flared-tie` para trazar una ligadura formada por líneas rectas. Su objetivo es sustituir a la función predeterminada de trazado de la ligadura (esto es, servir como un argumento sustitutorio para la propiedad `stencil` del objeto gráfico `Tie`).

El argumento de `flared-tie` es una lista de pare de coordenadas que especifican puntos adicionales entre el primero y el último para abarcar todas las líneas de la ligadura. Los puntos primero y último son idénticos a los puntos de inicio y final de la ligadura, respectivamente. Los valores de las coordenadas X e Y son múltiplos de la longitud y la altura del rectángulo circundante (teniendo también en cuenta la dirección de la ligadura); por tanto, el primer punto tiene las coordenadas (0,0), y el último (1,0).

La función `flare-tie` define una forma abreviada de especificar una ligadura plana. Es posible efectuar un trucaje adicional de la forma mediante la sobreescritura de `Tie.details.height-limit` o con la función `\shape`.

También es posible cambiar la definición personalizada sobre la marcha.

```
#(define ((flared-tie coords) grob)
   (define (pair-to-list pair)
     (list (car pair) (cdr pair)))

   (define (normalize-coords goods x y dir)
     (map
      (lambda (coord)
        (cons (* x (car coord)) (* y dir (cdr coord))))
      goods))

   (define (my-c-p-s points thick)
     (make-connected-path-stencil points thick 1.0 1.0 #f #f))
```

```
   ;; Calling `ly:tie::print` and assigning its return value to a
   ;; variable in this outer `let` triggers LilyPond to position the
   ;; tie, allowing us to extract its extents.  We only proceed,
   ;; however, if the tie doesn't get discarded (for whatever reason).
   (let ((sten (ly:tie::print grob)))
     (if (grob::is-live? grob)
         (let* ((layout (ly:grob-layout grob))
                (line-thickness (ly:output-def-lookup layout
                                                      'line-thickness))
                (thickness (ly:grob-property grob 'thickness 0.1))
                (used-thick (* line-thickness thickness))
                (dir (ly:grob-property grob 'direction))
                (xex (ly:stencil-extent sten X))
                (yex (ly:stencil-extent sten Y))
                (lenx (interval-length xex))
                (leny (interval-length yex))
                (xtrans (car xex))
                (ytrans (if (> dir 0)(car yex) (cdr yex)))
                ;; Add last point.
                (coord-list (append coords '((1.0 . 0.0))))
                (uplist
                 (map pair-to-list
                      (normalize-coords coord-list lenx (* leny 2) dir))))
           (ly:stencil-translate
            (my-c-p-s uplist used-thick)
            (cons xtrans ytrans)))
         '())))

% Define a default tie shape consisting of three straight lines.
#(define flare-tie
   (flared-tie '((0.1 . 0.3) (0.9 . 0.3))))

\relative c' {
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <a c e a c e a c e>~ q
  \once \override Tie.stencil = #flare-tie
  q~ q\break

  <>^\markup \small \typewriter "height-limit = 14"
  \override Tie.details.height-limit = 14
  a'4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

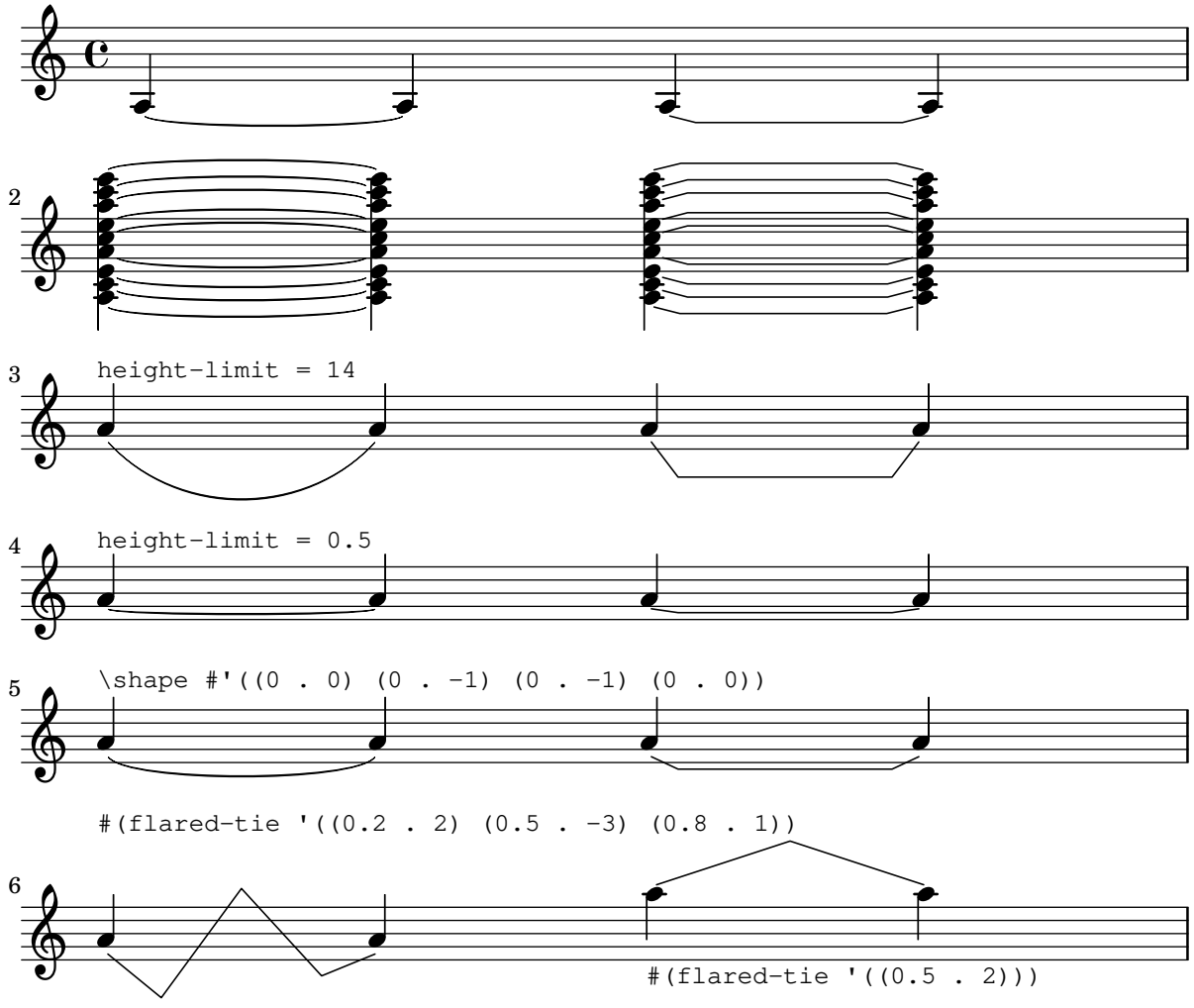
  <>^\markup \small \typewriter "height-limit = 0.5"
  \override Tie.details.height-limit = 0.5
  a4~ a
  \once \override Tie.stencil = #flare-tie
```

```
  a4~ a \break

  \revert Tie.details.height-limit

  <>^\markup \small \typewriter
            "\\shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0))"
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a
  \once \override Tie.stencil = #flare-tie
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a \break

  <>^\markup \small \typewriter
            "#(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1))"
  \once \override Tie.stencil =
       #(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))
  a4~ a
  <>_\markup \small \typewriter
            "#(flared-tie '((0.5 . 2)))"
  \once \override Tie.stencil = #(flared-tie '((0.5 . 2)))
  a'4~ a
}
```
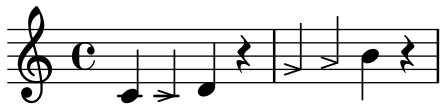


Notación de los golpes de lengua de la flauta

Es posible indicar técnicas de articulación especiales como el golpe de lengua de la flauta ("tongue slap") sustituyendo la cabeza de la figura con el glifo adecuado. Para ello podemos trazar la cabeza de la nota en forma de acento con \markup.

```
slap =
#(define-music-function (music) (ly:music?)
  #{
    \temporary \override NoteHead.stencil =
      #ly:text-interface::print
    \temporary \override NoteHead.text =
      \markup
        \translate #'(1 . 0)
          \override #'(thickness . 1.4)
            \overlay { \draw-line #'(-1.2 . 0.4)
                       \draw-line #'(-1.2 . -0.4) }
    \temporary \override NoteHead.stem-attachment =
      #(lambda (grob)
        (let* ((stem (ly:grob-object grob 'stem))
               (dir (ly:grob-property stem 'direction UP))
               (is-up (eqv? dir UP)))
          (cons dir (if is-up 0 -0.8))))
    #music
    \revert NoteHead.stencil
    \revert NoteHead.text
    \revert NoteHead.stem-attachment
  #})

\relative c' {
  c4 \slap c d r
  \slap { g4 a } b r
}
```



Tablas alternativas de diagramas de posiciones

Se pueden crear tablas alternativas de diagramas de posiciones. Se utilizarían para tener diagramas altenativos para un acorde dado. Para usar una tabla alternativa de diagramas de posiciones, se debe crear la tabla primero. Después se añaden los diagramas a la tabla.

La tabla de diagramas de posiciones que se crea puede estar vacía o se puede copiar a partir de una tabla existente.

La tabla a usar en la impresión de los diagramas predefinidos se selecciona por medio de la propiedad \predefinedDiagramTable.

```
\include "predefined-guitar-fretboards.ly"

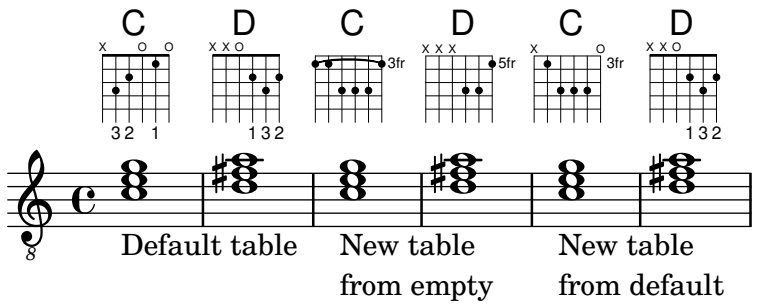
% Make a blank new fretboard table.
#(define custom-fretboard-table-one
   (make-fretboard-table))
```

```
% Make a new fretboard table as a copy of `default-fret-table`.
#(define custom-fretboard-table-two
   (make-fretboard-table default-fret-table))

% Add a chord to `custom-fretboard-table-one`.
\storePredefinedDiagram #custom-fretboard-table-one
                        \chordmode {c}
                        #guitar-tuning
                        "3-(;3;5;5;5;3-);"

% Add a chord to `custom-fretboard-table-two`.
\storePredefinedDiagram #custom-fretboard-table-two
                        \chordmode {c}
                        #guitar-tuning
                        "x;3;5;5;5;o;"

<<
  \chords {
    c1 | d1 |
    c1 | d1 |
    c1 | d1 |
  }
  \new FretBoards {
    \chordmode {
      \set predefinedDiagramTable = #default-fret-table
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-one
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-two
      c1 | d1 |
    }
  }
  \new Staff {
    \clef "treble_8"
    <<
      \chordmode {
        c1 | d1 |
        c1 | d1 |
        c1 | d1 |
      }
      {
        s1_\markup "Default table" | s1 |
        s1_\markup \column { "New table" "from empty" } | s1 |
        s1_\markup \column { "New table" "from default" } | s1 |
      }
    >>
  }
>>
```



Generate special note head shapes

When a note head with a special shape cannot easily be generated with graphic markup, a drawing specification for `ly:make-stencil` can be used to generate the shape. This snippet gives an example for a parallelogram-shaped note head.

Unfortunately, the available commands in a drawing specification are currently not documented (this is tracked in Issue #6874 (`https://gitlab.com/lilypond/lilypond/-/issues/6874`)); in any case, the used `path` sub-command has the following signature, quite similar to the `make-path-stencil` Scheme function.
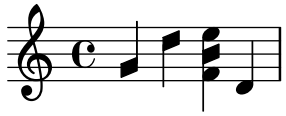
```
(path thickness command-list line-cap-style line-join-style fill)
```

The commands in *command-list* resemble PostScript drawing commands but with arguments after the command name.

```
parallelogram =
  #(ly:make-stencil
    '(path 0.1
           (rmoveto 0 0.25
            lineto 1.2 0.75
            lineto 1.2 -0.25
            lineto 0 -0.75
            lineto 0 0.25)
           round
           round
           #t)
    (cons -0.05 1.25)
    (cons -.75 .75))

myNoteHeads = \override NoteHead.stencil = \parallelogram
normalNoteHeads = \revert NoteHead.stencil

\relative c'' {
  \myNoteHeads
  g4 d'
  \normalNoteHeads
  <f, \tweak stencil \parallelogram b e>4 d
}
```



Generar corchetes personalizados

La propiedad `stencil` del grob `Flag` (el objeto gráfico corchete) se puede fijar a una función de Scheme personalizada que genere el glifo del corchete.

```
#(define-public (weight-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (log (- (ly:grob-property stem-grob 'duration-log) 2))
          (is-up? (eqv? (ly:grob-property stem-grob 'direction) UP))
          (yext (if is-up? (cons (* log -0.8) 0) (cons 0 (* log 0.8))))
          (flag-stencil (make-filled-box-stencil '(-0.4 . 0.4) yext))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stroke-stencil (if (equal? stroke-style "grace")
                              (make-line-stencil 0.2 -0.9 -0.4 0.9 -0.4)
                              empty-stencil)))
     (ly:stencil-add flag-stencil stroke-stencil)))


% Create a flag stencil by looking up the glyph from the font
#(define (inverted-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (dir (if (eqv? (ly:grob-property stem-grob 'direction) UP) "d" "u"))
          (flag (retrieve-glyph-flag "" dir "" grob))
          (line-thickness (ly:staff-symbol-line-thickness grob))
          (stem-thickness (ly:grob-property stem-grob 'thickness))
          (stem-width (* line-thickness stem-thickness))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stencil (if (null? stroke-style)
                       flag
                       (add-stroke-glyph flag stem-grob dir stroke-style "")))
          (rotated-flag (ly:stencil-rotate-absolute stencil 180 0 0)))
     (ly:stencil-translate rotated-flag (cons (- (/ stem-width 2)) 0))))

snippetexamplenotes =
{
  \autoBeamOff c'8 d'16 c'32 d'64 \acciaccatura {c'8} d'64
}

{
  \time 1/4
  <>^"Normal flags"
  \snippetexamplenotes

  <>_"Custom flag: inverted"
  \override Flag.stencil = #inverted-flag
  \snippetexamplenotes

  <>^"Custom flag: weight"
  \override Flag.stencil = #weight-flag
  \snippetexamplenotes

  <>_"Revert to normal"
  \revert Flag.stencil
```

```
  \snippetexamplenotes
}
```



Generación de notas aleatorias

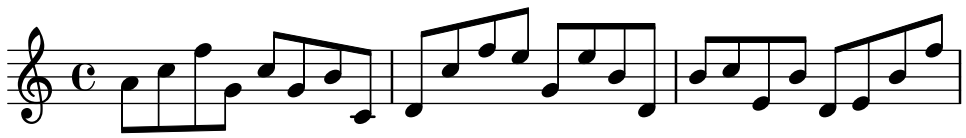
Este fragmento de código basado en Scheme genera notas aleatorias. Úselo como

```
\randomNotes n desde hasta duración
```

para generar *n* notas aleatorias entre las alturas *desde* y *hasta*, con una longitud de *duración*.

```
randomNotes =
#(define-music-function (n from to dur)
   (integer? ly:pitch? ly:pitch? ly:duration?)
   (let ((from-step (ly:pitch-steps from))
         (to-step (ly:pitch-steps to)))
     (make-sequential-music
      (map (lambda (_)
             (let* ((step (+ from-step
                             (random (- to-step from-step))))
                    (pitch (ly:make-pitch 0 step 0)))
               #{ $pitch $dur #}))
           (iota n)))))

\randomNotes 24 c' g'' 8
```



Generación de partituras completas y partes de libro en Scheme sin usar el analizador sintáctico

Internamente, una partitura de LilyPond no es más que una expresión de Scheme, generada por el analizador sintáctico de LilyPond. Usando Scheme, también podemos generar automáticamente una partitura sin ningún archivo de entrada. Si tenemos la expresión musical en Scheme, se puede generar la partitura simplemente llamando a

```
(scorify-music music)
```

sobre la música. Así se genera un objeto partitura, para el que podemos entonces fijar un bloque layout personalizado con

```
(let* ((layout (ly:output-def-clone $defaultlayout)))
   ; modificamos el layout aquí, y después lo asignamos:
   (ly:score-add-output-def! score layout))
```

Finalmente, todo lo que tenemos que hacer es pasar esta partitura a lilypond para que realice la composición tipográfica. Este fragmento de código define las funciones `(add-score score)`, `(add-text text)` y `(add-music music)` para pasar una partitura completa, elementos de marcado o algo de música a LilyPond para su composición tipográfica.

Este fragmento de código también funciona para la tipografía de partituras dentro de un bloque \book {...}, así como partituras normales del nivel superior jerárquico. Para conseguirlo, cada una de las partituras destinadas a su tipografiado se añaden al final de una lista de partituras del nivel superior jerárquico y se modifica el toplevel-book-handler (que es una función de Scheme que se llama para procesar un libro una vez que se ha cerrado el bloque \book{..}) para insertar todas las partituras así coleccionadas al libro.

Nota: por razones técnicas, solo se muestra el primer \book, debido a que las otras instrucciones \book crean archivos de salida adicionales.

```
#(define-public (add-score score)
   (module-define! (current-module) 'toplevel-scores
                   (cons score (module-ref (current-module) 'toplevel-scores '()))))

#(define-public (add-text text)
   (add-score (list text)))

#(define-public (add-music music)
   (collect-music-aux (lambda (score)
                        (add-score score))
                      music))

#(define-public (toplevel-book-handler book)
   (map (lambda (score)
          (ly:book-add-score! book score))
        (reverse (ly:book-lookup book 'toplevel-scores)))
   (print-book-with-defaults book))

#(define-public (book-score-handler book score)
   (add-score score))

#(define-public (book-text-handler book text)
   (add-text text))

#(define-public (book-music-handler book music)
   (add-music music))


% Some example code to show how to use these functions.  Each call to
% `\oneNoteScore` constructs a global markup followed by a single
% staff with a single quarter note.  The pitch of this note is taken
% from the variable `pitch`; the start value 0 corresponds to pitch C.
% After emitting the score, variable `pitch` gets increased by 1.
%
% `\oneNoteScore` calls Scheme function `add-one-note-score` to do all
% the work.

#(define add-one-note-score #f)
#(let ((pitch 0))
   (set! add-one-note-score
         (lambda ()
           (let* ((music
                   (make-music
```

```
                'EventChord
                'elements (list (make-music
                                 'NoteEvent
                                 'duration (ly:make-duration 2 0 1/1)
                                 'pitch (ly:make-pitch 0 pitch 0)))))
              (score (scorify-music music))
              (layout (ly:output-def-clone $defaultlayout))
              (note-name (case pitch
                           ((0) "do")
                           ((1) "ré")
                           ((2) "mi")
                           ((3) "fa")
                           ((4) "sol")
                           ((5) "la")
                           ((6) "si")
                           (else "huh")))
              (title (markup #:large #:line
                             ("Score with a" note-name))))
        (ly:score-add-output-def! score layout)
        (add-text title)
        (add-score score))
      (set! pitch (modulo (1+ pitch) 7)))))

oneNoteScore =
#(define-void-function () ()
   (add-one-note-score))

\book {
  \oneNoteScore

  \paper { tagline = ##f }
}


\book {
  \oneNoteScore
  \oneNoteScore

  \paper { tagline = ##f }
}

% Top-level scores are also handled correctly.
\oneNoteScore
\oneNoteScore

\paper { tagline = ##f }
```

Score with a do



Símbolos de porcentaje sueltos

También se pueden imprimir símbolos de porcentaje sueltos.

```
makePercent =
#(define-music-function (note) (ly:music?)
   "Make a percent repeat the same length as NOTE."
   (make-music 'PercentEvent
               'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```



Números como notas de notación fácil

Las cabezas de nota de notación fácil utilizan la propiedad `note-names` del objeto `NoteHead` para determinar lo que aparece dentro de la cabeza. Mediante la sobreescritura de esta propiedad, es posible imprimir números que representen el grado de la escala.

Se puede crear un grabador simple que haga esto para la cabeza de cada nota que ve.

```
#(define Ez_numbers_engraver
   (make-engraver
    (acknowledgers
     ((note-head-interface engraver grob source-engraver)
      (let* ((context (ly:translator-context engraver))
             (tonic-pitch (ly:context-property context 'tonic))
             (tonic-name (ly:pitch-notename tonic-pitch))
             (grob-pitch
              (ly:event-property (event-cause grob) 'pitch))
             (grob-name (ly:pitch-notename grob-pitch))
             (delta (modulo (- grob-name tonic-name) 7))
             (note-names
              (make-vector 7 (number->string (1+ delta)))))
        (ly:grob-set-property! grob 'note-names note-names))))))

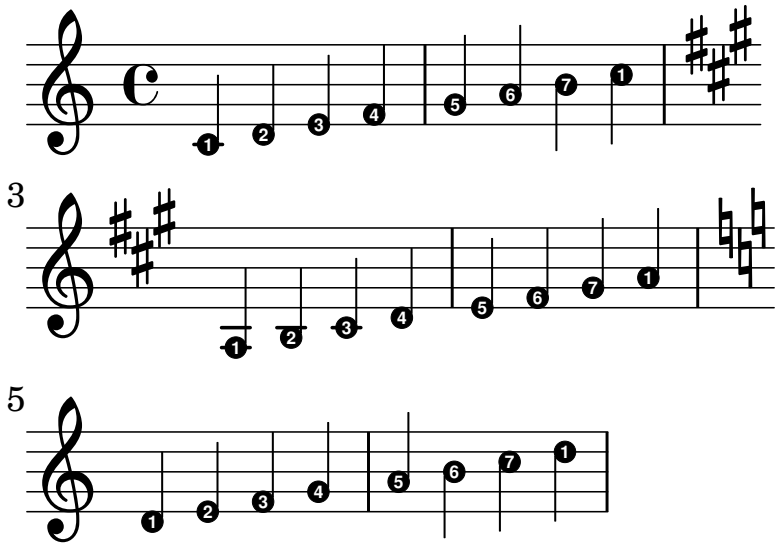
#(set-global-staff-size 30)

\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break
```

```
  \key a \major
  a,4 b cis d
  e4 fis gis a \break

  \key d \dorian
  d,4 e f g
  a4 b c d
}
```



Modificación de articulaciones por tipo

A veces queremos modificar un solo timpo de articulación. Aunque siempre es posible usar la instrucción \tweak, podría hacerse tedioso hacerlo para todos y cada uno de los símbolos de una partitura completa. El ejemplo muestra cómo trucar articulaciones con una lista de ajustes personalizados. Un caso sería la creación de una hoja de estilos.

```
#(define (custom-script-tweaks ls)
   (lambda (grob)
     (let* ((type (ly:event-property (ly:grob-property grob 'cause)
                                     'articulation-type))
            (tweaks (assoc-ref ls type)))
       (when tweaks
         (for-each
          (lambda (x) (ly:grob-set-property! grob (car x) (cdr x)))
          tweaks)))))

customScripts =
#(define-music-function (settings) (list?)
   #{
     \override Script.before-line-breaking =
       #(custom-script-tweaks settings)
   #})
revertCustomScripts = \revert Script.before-line-breaking


% Example
```

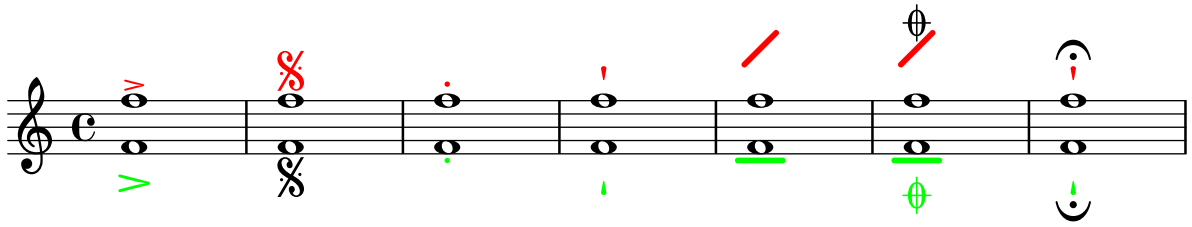
```
% Predefine two sets of desired tweaks.
#(define my-settings-1
   '((accent . ((font-size . 0)
                (color . (1 0 0))))
     (segno . ((font-size . 0)
               (color . (1 0 0))))
     (staccato . ((color . (1 0 0))
                  (padding . 0.5)))
     (staccatissimo . ((padding . 1)
                       (color . (1 0 0))))
     (tenuto . ((color . (1 0 0))
                (rotation . (45 0 0))
                (padding . 2)
                (font-size . 10)))
    ))

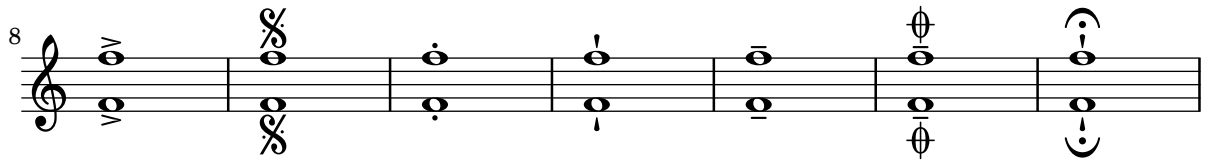
#(define my-settings-2
  '((accent . ((font-size . 4)
               (color . (0 1 0))
               (padding . 1.5)))
    (coda . ((color . (0 1 0))
             (padding . 1)))
    (staccato . ((color . (0 1 0))))
    (staccatissimo . ((padding . 2)
                      (color . (0 1 0))))
    (tenuto . ((color . (0 1 0))
               (font-size . 10)))
    ))

music = { f1-> | f\segno | f-. | f-! | f-- | f--\coda | f-!\fermata | }

block = {
  \music
  \break
  \revertCustomScripts \music
}

\new Staff <<
  \new Voice \with { \customScripts #my-settings-1 }
    \relative c''{ \voiceOne \block }
  \new Voice \with { \customScripts #my-settings-2 }
    \relative c' { \voiceTwo \block }
>>
```





Posicionamiento de las notas de adorno con espacio flotante

Al establecer la propiedad `strict-grace-spacing` hacemos que las columnas musicales para las notas de adorno sean 'flotantes', es decir, desacopladas de las notas que no son de adorno: primero se aplica el espaciado de las notas normales, y luego se ponen las columnas musicales de las notas de adorno a la izquierda de las columnas musicales de las notas principales.

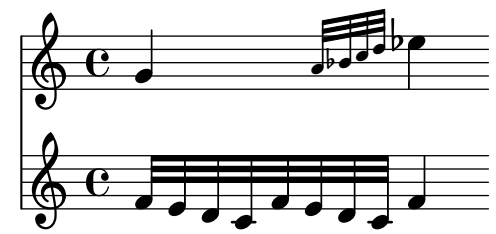
Sin embargo, a causa del bug número 6876 (`https://gitlab.com/lilypond/lilypond/-/issues/6876`), las alteraciones accidentales se ignoran si esta propiedad está establecida. Este fragmento de código ofrece una solución para este problema.

Otro desafortunado efecto colateral de esta propiedad es que LilyPond no comprueba si hay espacio horizontal suficiente para las notas de adorno (este problema recibe seguimiento como bug número 2630 (`https://gitlab.com/lilypond/lilypond/-/issues/2630`)). Tenemos que asegurarnos de que ha espacio suficiente, por ejemplo, por medio del uso de `\newSpacingSection` junto con un valor adecuado para la duración `base-shortest-duration` del objeto gráfico `SpacingSpanner`.

```
shiftedGrace =
#(define-music-function (offset music) (number? ly:music?)
   #{
     \override NoteHead.X-offset = #(- offset 0.85)
     \override Stem.X-offset = #offset
     \grace { $music }
     \revert NoteHead.X-offset
     \revert Stem.X-offset
   #})

\relative c'' <<
  { g4 \shiftedGrace #-1.3 a32 \shiftedGrace #-0.5 { bes c d } es4 }
  { f,32 e d c f e d c f4 }
>>

\layout {
  \context {
    \Score
    \override SpacingSpanner.strict-grace-spacing = ##t
  }
}
```



Print chord names with same root and different bass as slash and bass note

To print subsequent chord names only differing in their bass note as slash and bass note without a root symbol, use the Scheme engraver defined in this snippet. The behaviour is controlled by the chordChanges context property.

```
#(define Bass_changes_equal_root_engraver
   (lambda (ctx)
     "Drop root for follow-up chord names that differ only in bass.

In other words, the chord name sequence 'D D/C D/B' gets actually
printed as 'D /C /B'.

Set the `chordChanges` context property to `#t` to activate this
feature."
     (let ((chord-pitches '())
           (last-chord-pitches '())
           (bass-pitch #f))
       (make-engraver
        ((initialize this-engraver)
         (let ((chord-note-namer (ly:context-property ctx
                                                      'chordNoteNamer)))
           ;; Set `chordNoteNamer`, respecting user setting if already
           ;; done.
           (ly:context-set-property! ctx 'chordNoteNamer
                                     (if (procedure? chord-note-namer)
                                         chord-note-namer
                                         (chord-name:markup #f)))))

        (listeners
         ((note-event this-engraver event)
          (let* ((pitch (ly:event-property event 'pitch))
                 (pitch-name (ly:pitch-notename pitch))
                 (pitch-alt (ly:pitch-alteration pitch))
                 (bass (ly:event-property event 'bass #f))
                 (inversion (ly:event-property event 'inversion #f)))
            ;; We look at the `bass` and `inversion` event properties
            ;; to decide how to handle the current note event.  If
            ;; `inversion` is set we add the bass note to the chord as
            ;; an ordinary member so that we can compare inversed
            ;; chords; if `bass` is set the bass note is not added to
            ;; the chord.
            ;;
            ;; In the `chord-pitches` list we actually collect only
            ;; the notes' pitch names (which are integers) and pitch
            ;; alterations as pairs, ignoring the octave.
            (cond (bass (set! bass-pitch pitch))
                  (inversion
                   (set! bass-pitch pitch)
                   (set! chord-pitches
                         (cons (cons pitch-name pitch-alt)
                               chord-pitches)))
                  (else
```

```
                (set! chord-pitches
                      (cons (cons pitch-name pitch-alt)
                            chord-pitches)))))))

        (acknowledgers
         ((chord-name-interface this-engraver grob source-engraver)
          (let ((chord-changes (ly:context-property ctx
                                                    'chordChanges #f)))
            ;; If subsequent chords are equal apart from their bass,
            ;; we change the `text` property to print only the slash
            ;; and the bass note (via the formatter stored in the
            ;; `chordNoteNamer` context property).
            ;;
            ;; Equality is tested by comparing the sorted lists of
            ;; this chord's elements and the previous chord.  Sorting
            ;; is needed because inverted chords may have a different
            ;; order of pitches.  Note that we only do a simplified
            ;; sorting using the pitch name, ignoring the alteration.
            (if (and bass-pitch
                     chord-changes
                     (equal? (sort chord-pitches car<)
                             (sort last-chord-pitches car<)))
                (ly:grob-set-property!
                 grob 'text
                 (make-line-markup
                  (list
                   (ly:context-property ctx 'slashChordSeparator)
                   ((ly:context-property ctx 'chordNoteNamer)
                    bass-pitch
                    (ly:context-property ctx
                                         'chordNameLowercaseMinor))))))
            (set! last-chord-pitches chord-pitches)
            (set! chord-pitches '())
            (set! bass-pitch #f))))

        ((finalize this-engraver)
         (set! last-chord-pitches '()))))


myChords = \chordmode {
  % \germanChords

  \set chordChanges = ##t
  d2:m d:m/cis

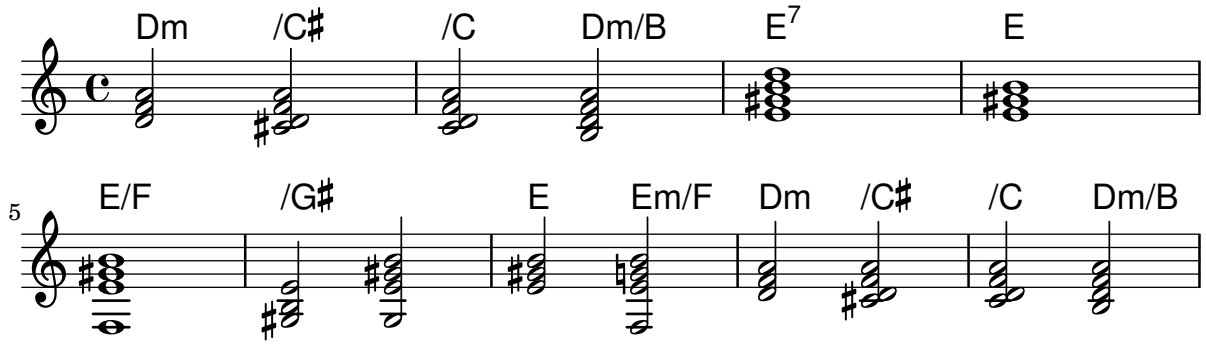
  d:m/c
  \set chordChanges = ##f
  d:m/b

  e1:7
  \set chordChanges = ##t
  e
```

```
  \break

  \once \set chordChanges = ##f
  e1/f
  e2/gis e/+gis e e:m/f d:m d:m/cis d:m/c
  \set chordChanges = ##f
  d:m/b
}

<<
  \new ChordNames
    \with { \consists #Bass_changes_equal_root_engraver }
    \myChords
  \new Staff \myChords
>>
```



Reemplazo de la ecualización predeterminada de instrumentos MIDI

El ecualizador predeterminado de MIDI se puede sustituir fijando la propiedad `instrumentEqualizer` en el contexto `Score` a un procedimiento de Scheme definido por el usuario que usa un nombre de instrumento MIDI como argumento junto a un par de fracciones que indican los vloúmenes mínimo y máximo, respectivamente, que aplicar a ese instrumento.

El ejemplo siguiente fija los volúmenes mínimo y máximo de la flauta y el clarinete.
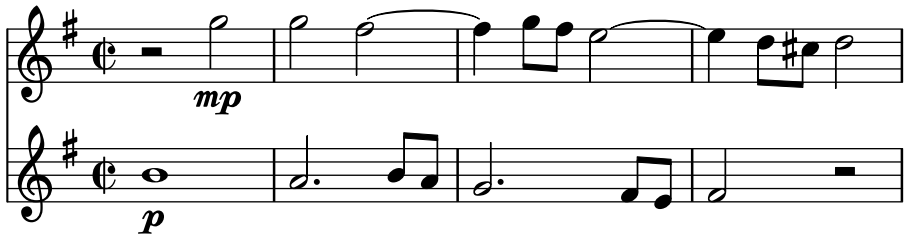
```
#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
```

```
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = "flute"
      \new Voice \relative {
        r2 g''\mp g fis~
        4 g8 fis e2~
        4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = "clarinet"
      \new Voice \relative {
        b'1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {  }
}
```



Separar las cancelaciones de tonalidad de los cambios de armadura

De forma predeterminada, las alteraciones accidentales que se usan para las cancelaciones en las armaduras se colocan adyacentes a las que se usan para los cambios de tonalidad. Este comportamiento se puede cambiar sobreescribiendo la propiedad `break-align-orders` del objeto gráfico `BreakAlignment`.

Si busca la definición de `break-alignment-interface` en la Referencia de funcionamiento interno (`https://lilypond.org/doc/v2.24/Documentation/internals/break_002dalignment_002dinterface`) de LilyPond, obtendrá la siguiente lista de los símbolos de break-align disponibles:

```
ambitus
breathing-sign
clef
cue-clef
cue-end-clef
custos
key-cancellation
key-signature
left-edge
signum-repetitionis
```

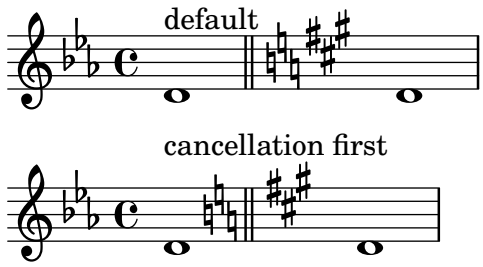
```
staff-bar
staff-ellipsis
time-signature
```

A partir de esta lista, vemos que lo que queremos hacer es trasladar la `key-cancellation` de forma que esté antes del `staff-bar`. Esto se hace con la función `\breakAlignInsert`.

```
music = { \key es \major d'1 \bar "||"
          \key a \major d'1 }

{ <>^\markup "default"
  \music }

{ <>^\markup "cancellation first"
  \breakAlignInsert key-cancellation before staff-bar
  \music }
```
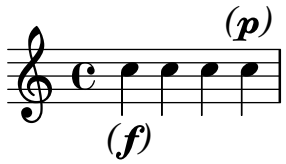


Slanted parentheses around dynamics

The easiest way to parenthesize dynamics is to use the normal `\parenthesize` command and adjust the font size, e.g., `{ c'4\tweak Parentheses.font-size 2 \parenthesize \f }`. However, this results in upright parentheses, while dynamic marks are normally slanted. This snippet gives a method to make the parentheses slanted as well.

```
paren =
#(define-event-function (dyn) (ly:event?)
   (make-dynamic-script
    #{ \markup \concat {
         \normal-text \italic \fontsize #2 (
         \pad-x #0.2 #(ly:music-property dyn 'text)
         \normal-text \italic \fontsize #2 )
       }
    #}))

\relative c'' {
  c4\paren\f c c \dynamicUp c\paren\p
}
```

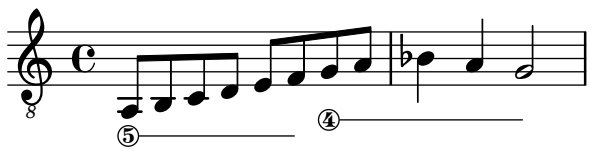


Líneas de extensión para números de cuerda

Hacemos una línea extensora para las indicaciones de número de cuerda, motrando que una serie de notas se han de tocar sobre la misma cuerda.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})

\relative c {
  \clef "treble_8"
  \textSpannerDown
  \stringNumberSpanner "5" a8\startTextSpan b c d
    e f\stopTextSpan \stringNumberSpanner "4" g\startTextSpan a |
  bes4 a g2\stopTextSpan
}
```



Rectángulo de tres segmentos visibles

Este ejemplo muestra cómo añadir ina instrucción de marcado para obtener un rectángulo de tres segmentos visibles rodeando un elemento de texto (u otro elemento de marcado).

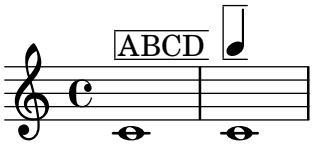
```
% New command to add a three-sided box, with sides north, west, and south.
% Based on the `box-stencil` command defined in `scm/stencil.scm`.
% Note that ";;" is used to comment a line in Scheme.
#(define-public (NWS-box-stencil stencil thickness padding)
   "Add a box around STENCIL, producing a new stencil."
   (let* ((x-ext (interval-widen (ly:stencil-extent stencil X) padding))
          (y-ext (interval-widen (ly:stencil-extent stencil Y) padding))
          (y-rule (make-filled-box-stencil (cons 0 thickness) y-ext))
          (x-rule (make-filled-box-stencil
                   (interval-widen x-ext thickness) (cons 0 thickness))))
     ;; (set! stencil (ly:stencil-combine-at-edge stencil X 1 y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil X LEFT y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil Y UP x-rule 0.0))
     (set! stencil (ly:stencil-combine-at-edge stencil Y DOWN x-rule 0.0))
     stencil))

% The corresponding markup command, based on the `\box` command defined
% in `scm/define-markup-commands.scm`.
#(define-markup-command (NWS-box layout props arg) (markup?)
   #:properties ((thickness 0.1) (font-size 0) (box-padding 0.2))
   "Draw a box round ARG.

Look at THICKNESS, BOX-PADDING, and FONT-SIZE properties to determine
line thickness and padding around the markup."
   (let ((pad (* (magstep font-size) box-padding))
```

```
        (m (interpret-markup layout props arg)))
    (NWS-box-stencil m thickness pad)))


\relative c' {
  c1^\markup { \NWS-box ABCD }
  c1^\markup { \NWS-box \note {4} #1.0 }
}
```



Transposing pitches with minimum accidentals ("smart" transpose)

This example uses some Scheme code to enforce enharmonic modifications for notes in order to have the minimum number of accidentals. In this case, the following rules apply:

- double accidentals should be removed
- b sharp → c
- e sharp → f
- c flat → b
- f flat → e

In this manner, the most natural enharmonic notes are chosen.

```
#(define (naturalize-pitch p)
   (let ((o (ly:pitch-octave p))
         ;; `ly:pitch-alteration` returns quarter tone steps.
         (a (* 4 (ly:pitch-alteration p)))
         (n (ly:pitch-notename p)))
     (cond
      ((and (> a 1)
            (or (eqv? n 6) (eqv? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1)
            (or (eqv? n 0) (eqv? n 3)))
       (set! a (+ a 2))
       (set! n (- n 1))))
     (cond
      ((> a 2)
       (set! a (- a 4))
       (set! n (+ n 1)))
      ((< a -2)
       (set! a (+ a 4))
       (set! n (- n 1))))
     (when (< n 0)
       (set! o (- o 1))
       (set! n (+ n 7)))
     (when (> n 6)
```
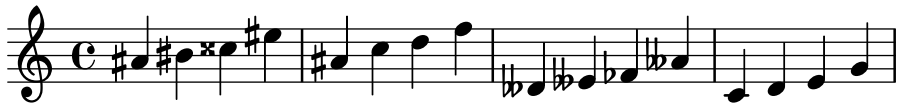
```
      (set! o (+ o 1))
      (set! n (- n 7)))
    (ly:make-pitch o n (/ a 4))))

#(define (naturalize music)
   (let ((es (ly:music-property music 'elements))
         (e (ly:music-property music 'element))
         (p (ly:music-property music 'pitch)))
     (when (pair? es)
       (ly:music-set-property! music 'elements
                               (map naturalize es)))
     (when (ly:music? e)
       (ly:music-set-property! music 'element
                               (naturalize e)))
     (when (ly:pitch? p)
       (set! p (naturalize-pitch p))
       (ly:music-set-property! music 'pitch p))
     music))

naturalizeMusic =
#(define-music-function (m) (ly:music?)
   (naturalize m))

music = \relative c' { c4 d e g }

\new Staff {
  \transpose c ais { \music }
  \naturalizeMusic \transpose c ais { \music }
  \transpose c deses { \music }
  \naturalizeMusic \transpose c deses { \music }
}
```



Dos pares de \partCombine en un solo pentagrama

La función \partCombine toma dos expresiones musicales, cada una de las cuales contiene una parte o voz, y las distribuye entre cuatro contextos de Voice llamados “one”, “two”, “solo” y “shared” dependiendo de cuándo y cómo las partes o voces se combinan en una voz común.

Las voces que devuelve \partCombine pueden ver ajustadas sus propiedades de presentación de la manera usual. Aquí definimos extensiones de \partCombine para facilitar la inclusión de cuatro voces dentro del mismo pentagrama.

\partCombineUp y \partCombineDown son variantes de \partCombine pensadas para producir combinaciones de dos voces con la plica hacia arriba y hacia abajo, respectivamente. Sin embargo, combinarlas para comprimir cuatro partes en un solo pentagrama requiere ciertos ajustes especiales, los que este fragmento de código define adecuadamente.

```
customPartCombineUp =
#(define-music-function (part1 part2) (ly:music? ly:music?)
  "Make an up-stem `VoiceBox` context that combines PART1 and PART2.

The context is called 'Up'; internally, the function calls
`\\partCombineUp`."
  #{
    \new VoiceBox = "Up" <<
      \context Voice = "one" { \voiceOne }
      \context Voice = "two" { \voiceThree }
      \context Voice = "shared" { \voiceOne }
      \context Voice = "solo" { \voiceOne }
      \context NullVoice = "null" {}
      \partCombine #part1 #part2
    >>
  #})

customPartCombineDown =
#(define-music-function (part3 part4) (ly:music? ly:music?)
  "Make a down-stem `VoiceBox` context that combines PART3 and PART4.

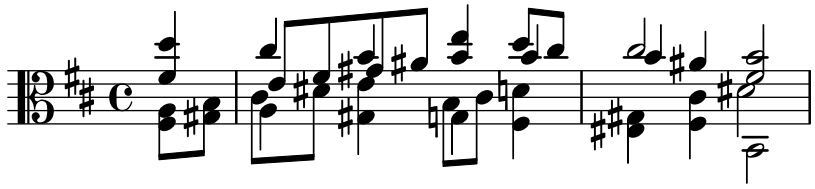
The context is called 'Down'; internally, the function calls
`\\partCombineDown`."
  #{
    \new VoiceBox = "Down" <<
      \set VoiceBox.soloText = #"Solo III"
      \set VoiceBox.soloIIText = #"Solo IV"
      \context Voice ="one" { \voiceFour }
      \context Voice ="two" { \voiceTwo }
      \context Voice ="shared" { \voiceFour }
      \context Voice ="solo" { \voiceFour }
      \context NullVoice = "null" {}
      \partCombine #part3 #part4
    >>
  #})

soprano = { d'4 | cis'  b  e'  d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }

\new Staff <<
  \key b\minor
  \clef alto
  \partial 4
  \transpose b b' \customPartCombineUp \soprano \alto
  \customPartCombineDown \tenor \bass
>>

\layout {
  \context {
    \Staff
```

```
    \accepts "VoiceBox"
  }
  \context {
    \name "VoiceBox"
    \type "Engraver_group"
    \defaultchild "Voice"
    \accepts "Voice"
    \accepts "NullVoice"
  }
}
```
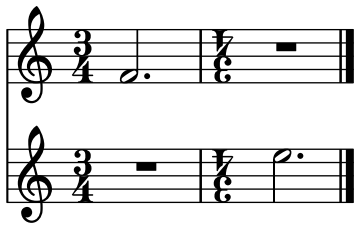


Indicaciones de compás personalizadas

Se pueden definir indicaciones de compás nuevas. La indicación de compás del segundo compás se imprime en posición invertida en los dos pentagramas.

```
#(add-simple-time-signature-style 'topsy-turvy
   (lambda (fraction)
     (make-rotate-markup 180 (make-compound-meter-markup fraction))))

<<
  \new Staff {
    \time 3/4 f'2.
    \override Score.TimeSignature.style = #'topsy-turvy
    \time 3/4 R2. \bar "|."
  }
  \new Staff {
    R2. e''
  }
>>
```



Utilizar ly:grob-object para acceder a los grobs con \tweak

Se puede acceder "lateralmente" a algunos grobs desde dentro de la función de callback de otro grob. Éstos se encuentran relacionados normalmente como "layout objects" (objetos de presentación) en la sección "Internal properties" (propiedades internas) de un interface de grob. Se usa la función `ly:grob-object` para acceder a estos grobs.

Se presentan más abajo como ejemplo algunas formas de addecer a grobs desde dentro de una función de callback de `NoteHead`, pero la técnica no se limita a las cabezas de nota. Sin embargo, la función de callback de `NoteHead` es especialmente importante, porque es la función de callback implícita que utiliza la instrucción `\tweak`.

La salida de consola de la función de ejemplo que se muestra abajo (`display-grobs`) es como sigue.

```
--------------------
#<Grob Accidental >
()
#<Grob Stem >
```

Probablemente no es tan útil, pero muestra que efectivamente se puede acceder a los objetos gráficos.

```
#(define (notehead-get-accidental notehead)
   ;; notehead is grob
   (ly:grob-object notehead 'accidental-grob))
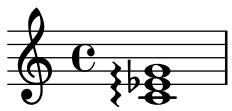
#(define (notehead-get-arpeggio notehead)
   ;; notehead is grob
   (let ((notecolumn (notehead-get-notecolumn notehead)))
     (ly:grob-object notecolumn 'arpeggio)))

#(define (notehead-get-notecolumn notehead)
   ;; notehead is grob
   (ly:grob-parent notehead X))

#(define (notehead-get-stem notehead)
   ;; notehead is grob
   (let ((notecolumn (notehead-get-notecolumn notehead)))
     (ly:grob-object notecolumn 'stem)))

#(define (display-grobs notehead)
   ;; notehead is grob
   (let ((accidental (notehead-get-accidental notehead))
         (arpeggio (notehead-get-arpeggio notehead))
         (stem (notehead-get-stem notehead)))
     (format (current-error-port) "~2&~a\n" (make-string 20 #\-))
     (for-each
      (lambda (x) (format (current-error-port) "~a\n" x))
      (list accidental arpeggio stem))))

\relative c' {
  %% display grobs for each note head:
  %\override NoteHead.before-line-breaking = #display-grobs
  <c
  %% or just for one:
  \tweak before-line-breaking #display-grobs
  es
  g>1\arpeggio
}
```
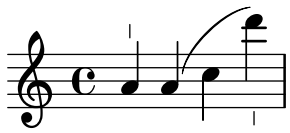


Línea vertical como una articulación barroca

Esta línea corta vertical situada encima de la nota es de uso común en música barroca. Su significado varía, pero en general indica notas que se deben tocar con más “peso”. El ejemplo siguiente muestra cómo conseguir dicha notación.

```
upline =
\tweak stencil
  #(lambda (grob)
    (grob-interpret-markup grob #{ \markup \draw-line #'(0 . 1) #}))
  \stopped

\relative c' {
  a'4^\upline a( c d')_\upline
}
```



33 Spacing

See also Sección “Spacing issues” en *Notation Reference*.
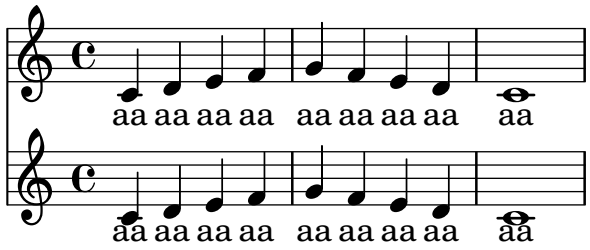
Adjusting vertical spacing of lyrics

This snippet shows how to bring the lyrics line closer to the staff.

```
music = \relative c' { c4 d e f | g4 f e d | c1 }
text = \lyricmode { aa aa aa aa aa aa aa aa aa }

<<
  \new Staff \new Voice = melody \music
  % Default layout:
  \new Lyrics \lyricsto melody \text
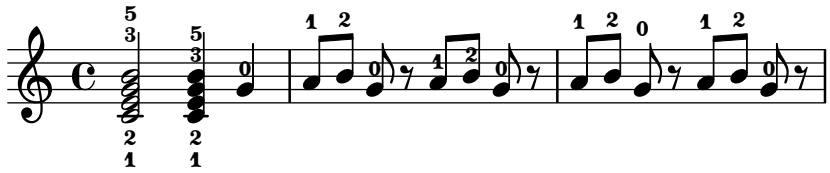
  \new Staff \new Voice = melody \music
  % Reducing the minimum space below the staff and above the lyrics.
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  } \lyricsto melody \text
>>
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Hay que prestar atención en situaciones en las que las digitaciones y las plicas están en la misma dirección: por defecto, las digitaciones solo evitan la colisión con plicas unidas por una barra. Se puede cambiar este ajuste para no evitar ninguna plica o evitarlas todas; el ejemplo siguiente muestra las dos copciones, así como la manera de volver al comportamiento predeteminado.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```



Breaking horizontal alignment of dynamics and textscripts

LilyPond uses `DynamicLineSpanner` grobs to horizontally align successive dynamic objects like hairpins and dynamic text, even if they are positioned on different sides of a staff. This connection cannot be broken, contrary to the vertical alignment (see snippet “Breaking vertical alignment of dynamics and textscripts”).

There are two solutions to circumvent the problem.

- Modify the `shorten-pair` property of the `Hairpin` grob to compensate the offset by which the hairpin was moved.
- Put the two dynamic objects into different voices.

Both solutions are demonstrated in this snippet.

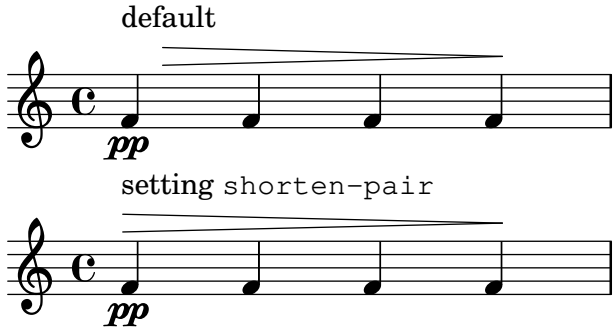
```
{
  <>^"default"
  f'_\pp ^\> f' f' f'\!
}
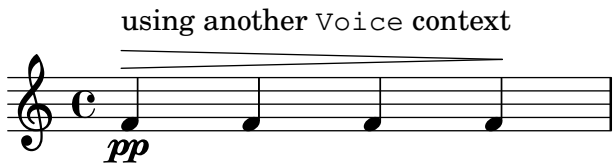
{
  <>^\markup { setting \typewriter shorten-pair }
  f'_\pp \tweak shorten-pair #'(-3 . 0) ^\> f' f' f'\!
}

{
  <>^\markup { using another \typewriter Voice context }
  << { f'^\> f' f' f'\! }
     \new Voice { s4_\pp } >>
}

\layout {
  line-width = 8\cm
  ragged-right = ##f

  \context {
    \Voice
    \override TextScript.staff-padding = #3.5
  }
}
```





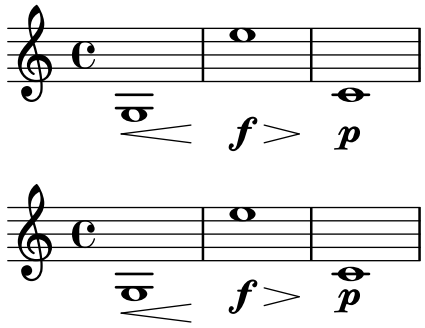
Desalinear indicaciones dinámicas y textuales verticalmente

De forma predeterminada, LilyPond usa ojetos gráficos DynamicLineSpanner para alinear verticalmente objetos de matiz dinámico sucesivos como reguladores y dinamicas textuales. Sin embargo esto no siempre es deseable. Insertando \breakDynamicSpan, que finaliza el objeto extenso de alineamiento de forma prematura, se puede evitar este alineamiento vertical.

Véase también el fragmento de código "Desalinear indicaciones dinámicas y textuales horizontalmente".

```
{ g1\< |
  e''\f\> |
  c'\p }

{ g1\< |
  e''\breakDynamicSpan\f\> |
  c'\p }
```



Harmonizing bar line thickness for staves with different sizes

When using \magnifyStaff only for some staves in a StaffGroup, BarLine grobs do not align any more due to its changed properties thick-thickness, hair-thickness, and kern.

To fix this, multiple workarounds are available, as demonstrated below.

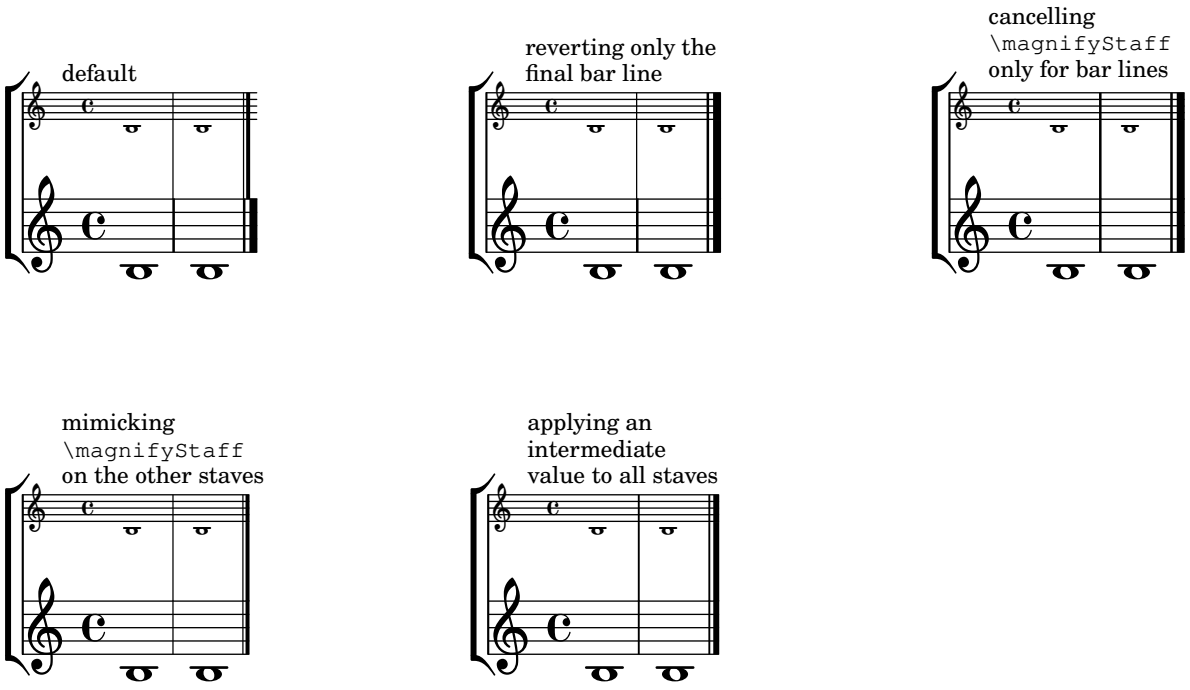
```
\markuplist {
  % First row.
  \fill-line {
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny "default"
          b1 b \bar "|."
        }
        \new Staff { b b }
      >>
    }
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
```

```
        \textMark \markup \tiny \column { "reverting only the"
                                          "final bar line" }
        b1 b
        \revert Staff.BarLine.thick-thickness
        \revert Staff.BarLine.hair-thickness
        \revert Staff.BarLine.kern
        \bar "|."
      }
      \new Staff { b b }
    >>
  }
  \score {
    \new StaffGroup <<
      \new Staff \with { \magnifyStaff #1/2
                         #(revert-props 'magnifyStaff 0
                                        '((BarLine thick-thickness)
                                          (BarLine hair-thickness)
                                          (BarLine kern))) } {
        \textMark \markup \tiny \column { "cancelling"
                                          \typewriter "\\magnifyStaff"
                                          "only for bar lines" }
        b1 b \bar "|."
      }
      \new Staff { b b }
    >>
  }
}

\vspace #2

% Second row.
\fill-line {
  \score {
    \new StaffGroup <<
      \new Staff \with { \magnifyStaff #1/2 } {
        \textMark \markup \tiny \column { "mimicking"
                                          \typewriter "\\magnifyStaff"
                                          "on the other staves" }
        b1 b \bar "|." }
      \new Staff \with { #(scale-props 'magnifyStaff 1/2 #t
                                       '((BarLine thick-thickness)
                                         (BarLine hair-thickness)
                                         (BarLine kern))) } {
        b b }
    >>
  }
  \score {
    \new StaffGroup <<
      \new Staff \with { \magnifyStaff #1/2
                         #(scale-props 'magnifyStaff 3/2 #t
                                       '((BarLine thick-thickness)
                                         (BarLine hair-thickness)
```

```
                                          (BarLine kern))) } {
          \textMark \markup \tiny \column { "applying an"
                                            "intermediate"
                                            "value to all staves" }
          b1 b \bar "|." }
        \new Staff \with { #(scale-props 'magnifyStaff 3/4 #t
                                         '((BarLine thick-thickness)
                                           (BarLine hair-thickness)
                                           (BarLine kern))) } {
          b b }
      >>
    }
    ""
  }
}
```



Etiqueta de página

Se pueden situar etiquetas de página dentro de la música o en el nivel superior, y ser referenciadas desde otros elementos de marcado.

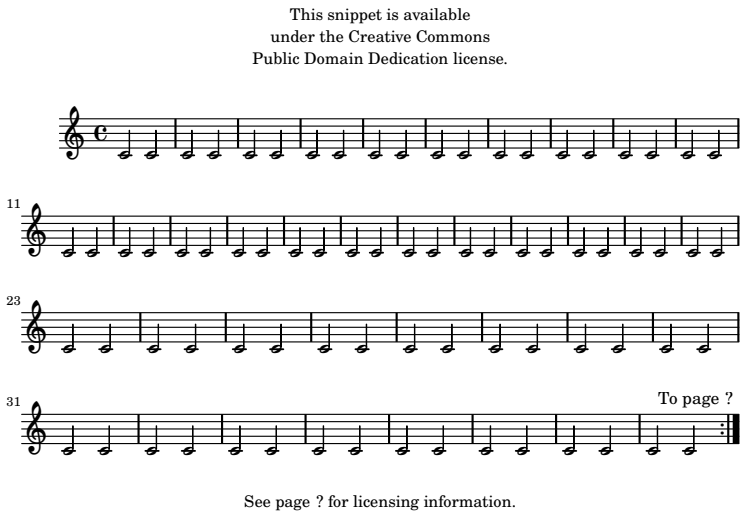
```
#(set-default-paper-size "a7" 'landscape)
#(set-global-staff-size 11)

\label license
\markup \fill-line {
  \center-column {
    "This snippet is available"
    "under the Creative Commons"
    "Public Domain Dedication license." } }

{
  \repeat volta 2 {
```

```
    \label startRepeat
    \*22 { c'2 2 }
    \pageBreak
    \*16 { c'2 2 }
  }
  \textEndMark \markup {
    \with-link #'startRepeat \line {
      To page \page-ref #'startRepeat "0" "?"
    }
  }
}

\markup \fill-line {
  \line {
    See page \page-ref #'license "0" "?" for
    licensing information. } }
```
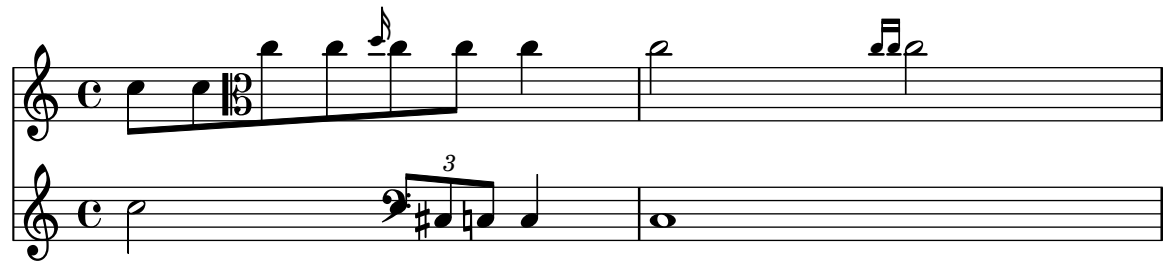


Espaciado de las notas estrictamente proporcional

Si está establecida al valor `#t` la propiedad `strict-note-spacing` del objeto gráfico SpacingSpanner, el espaciado de las notas no queda influido por los compases o claves que pueda haber dentro de un sistema. En lugar de ello, se colocan justo antes de la nota que tiene lugar en el mismo momento temporal. Esto puede producir colisiones.

```
\relative c'' <<
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  \set Score.proportionalNotationDuration = #1/16

  \new Staff {
    c8[ c \clef alto c c \grace { d16 } c8 c] c4
    c2 \grace { c16[ c16] } c2
  }
  \new Staff {
    c2 \tuplet 3/2 { c8 \clef bass cis,, c } c4
    c1
  }
>>
```



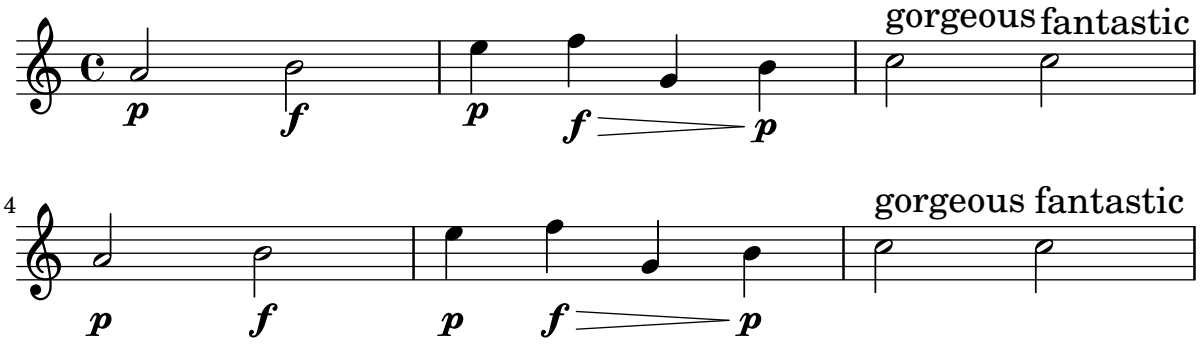
Indicaciones dinámicas y textuales alineadas verticalmente

Para todos los objetos `DynamicLineSpanner` (esto es, reguladores y textos de dinámica), la distancia vertical mínima entre su línea de referencia y el pentagrama viene dada por el valor dentro de la propiedad `staff-padding`, a no ser que otros elementos de notación los fuercen a situarse en una posición más alejada. El establecimiento de esta propiedad a un valor suficientemente grande, permite alinear las indicaciones de matiz dinámico.

Se usa una idea similar junto a \textLengthOn para alinear las inscripciones de texto a lo largo de su línea de base.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = 3
  \textLengthOn
  \override TextScript.staff-padding = 1
  \music
}
```

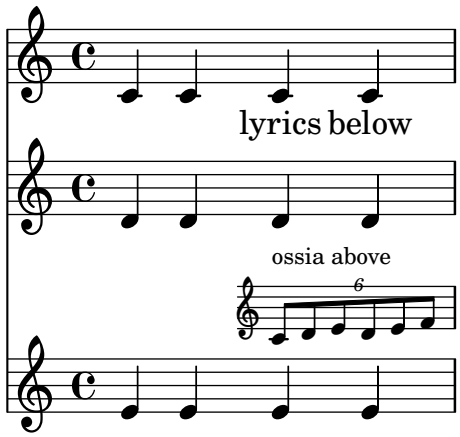


Alineación vertical de la letra y los compases de ossia

Este fragmento de código muestra el uso de las propiedades de contexto `alignBelowContext` y `alignAboveContext` para controlar la posición de la letra y los compases de ossia.

```
\relative c' <<
  \new Staff = "1" { c4 c c c }
  \new Staff = "2" { d4 d d d }
  \new Staff = "3" { e4 e e e }

  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = "1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = "3"
        fontSize = -2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
        \override VerticalAxisGroup.staff-staff-spacing =
           #'((minimum-distance . 0)
              (basic-distance . 0)
              (padding . 1))
      } {
        \tuplet 6/4 {
          \override TextScript.padding = 2
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



34 Specific notation

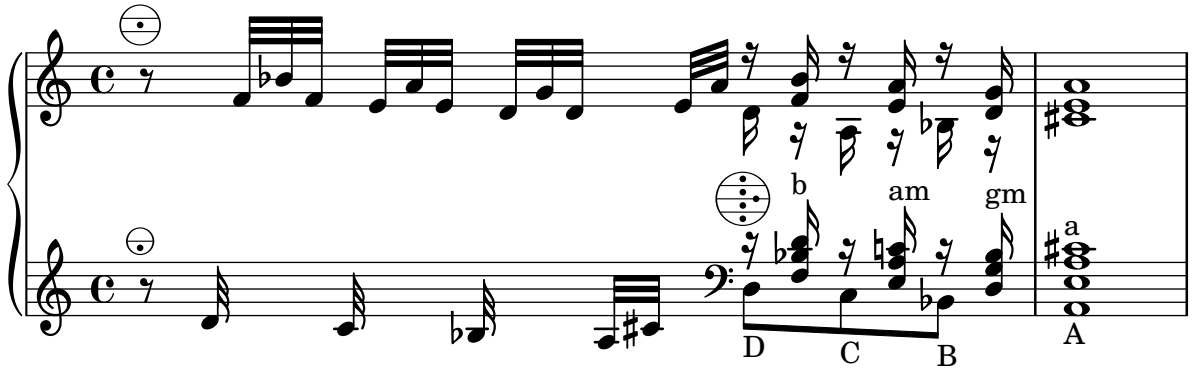
Símbolos de registración de acordeón

Los símbolos de registración de acordeón están disponibles como elementos de \markup y como eventos musicales autocontenidos (pues los cambios de registro tienden a ocurrir entre eventos de música). Los registros del bajo no están demasiado estandarizados. Las instrucciones existentes están disponibles en el apartado 'Símbolos de acordeón' en la Referencia de la notación (https://lilypond.org/doc/v2.24/Documentation/notation/accordion#discant-symbols).

```
#(use-modules (lily accreg))

\new PianoStaff <<
  \new Staff \relative {
    \clef treble
    \discant "10"
    r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
    <<
      { r16 <f bes> r <e a> r <d g> }
      \\
      { d r a r bes r }
    >> |
    <cis e a>1
  }

  \new Staff \relative {
    \clef treble
    \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef bass \stdBass "Master"
    <<
      { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
      <e a cis>1^"a" }
      \\
      { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```

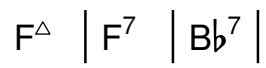


Añadir barras de compás al contexto de nombres de acorde (ChordNames)

Para añadir indicaciones de línea divisoria dentro del contexto de los nombres de acorde ChordNames, incluya el grabador Bar_engraver.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```



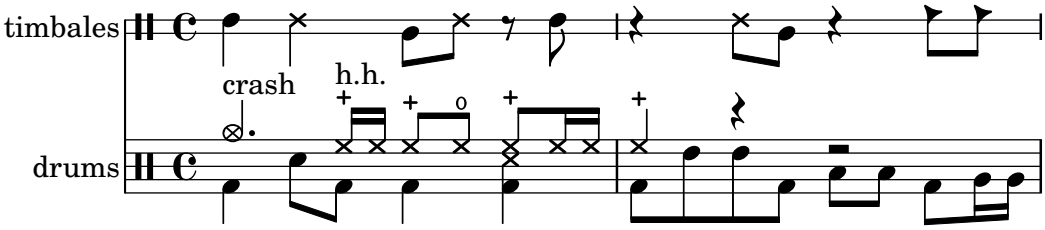
Escritura de partes de percusión

Mediante la utilización de las potentes herramientas preconfiguradas como la función \drummode y el contexto DrumStaff, la introducción de partes para percusión es muy fácil: las percusiones se sitúan en sus propias posiciones de pentagrama (con una clave especial) y tienen las cabezas correspondientes al instrumento. Es posible añadir un símbolo adicional a la percusión o reducir el número de líneas.

```
drh = \drummode {
        cymc4.^"crash" hhc16^"h.h." hh hhc8 hho hhc8 hh16 hh
        hhc4 r4 r2
      }
drl = \drummode {
        bd4 sn8 bd bd4 << bd ss >>
        bd8 tommh tommh bd toml toml bd tomfh16 tomfh
      }
timb = \drummode {
         timh4 ssh timl8 ssh r timh r4
         ssh8 timl r4 cb8 cb
       }

\score {
  <<
    \new DrumStaff \with {
      instrumentName = "timbales"
      drumStyleTable = #timbales-style
      \override StaffSymbol.line-count = #2
      \override BarLine.bar-extent = #'(-1 . 1)
    }
    <<
      \timb
    >>
    \new DrumStaff \with { instrumentName = "drums" }
    <<
      \new DrumVoice { \stemUp \drh }
      \new DrumVoice { \stemDown \drl }
    >>
  >>
```

```
  \layout { }
  \midi { \tempo 4 = 120 }
}
```

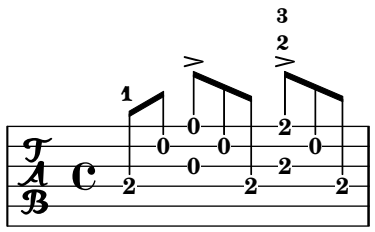


Añadir digitaciones a las tablaturas

Para añadir digitaciones a las tablaturas, utilice una combinación de `\markup` y `\finger`.

```
one = \markup { \finger 1 }
two = \markup { \finger 2 }
threeTwo = \markup {
  \override #'(baseline-skip . 2)
  \column {
    \finger 3
    \finger 2
  }
}

\score {
  \new TabStaff {
    \tabFullNotation
    \stemUp
    e8\4^\one b\2 <g\3 e'\1>^>[ b\2 e\4]
    <a\3 fis'\1>^>^\threeTwo[ b\2 e\4]
  }
}
```
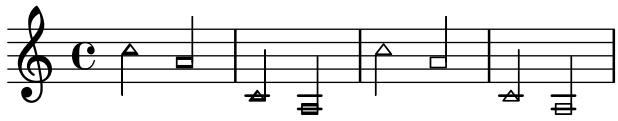


Cabezas de nota tipo Aiken, variante delgada

Las notas blancas de tipo Aiken se vuelven algo difíciles de leer con tamaños de pentagrama pequeños, especialmente con líneas adicionales. Al perder el espacio vacío del interior pueden parecer más bien figuras negras.

```
\score {
  {
    \aikenHeads
    c''2 a' c' a

    % Switch to thin-variant noteheads
    \set shapeNoteStyles = ##(doThin reThin miThin
```
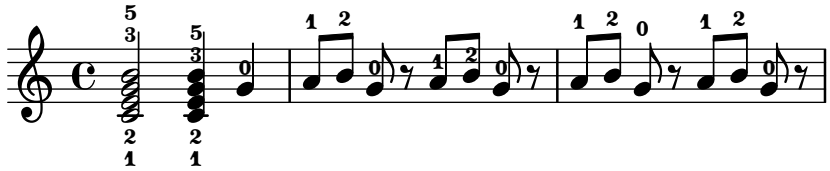
```
                                faThin sol laThin tiThin)
    c'' a' c' a
  }
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Hay que prestar atención en situaciones en las que las digitaciones y las plicas están en la misma dirección: por defecto, las digitaciones solo evitan la colisión con plicas unidas por una barra. Se puede cambiar este ajuste para no evitar ninguna plica o evitarlas todas; el ejemplo siguiente muestra las dos copciones, así como la manera de volver al comportamiento predeteminado.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```



Cambiar el número de líneas de una pauta

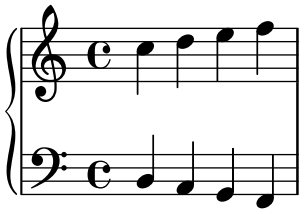
El número de líneas de una pauta se puede modificar sobreescribiendo la propiedad `line-count` del objeto `StaffSymbol`.

```
upper = \relative c'' {
  c4 d e f
}

lower = \relative c {
  \clef bass
  c4 b a g
}

\score {
  \context PianoStaff <<
    \new Staff {
```

```
      \upper
    }
    \new Staff {
      \override Staff.StaffSymbol.line-count = #4
      \lower
    }
  >>
}
```

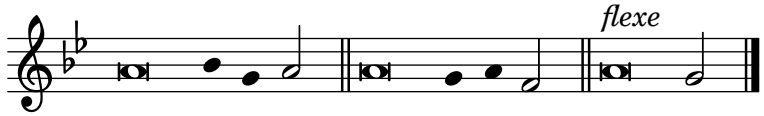


Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}
```



Excepciones para los nombres de acorde

La propiedad `chordNameExceptions` almacena una lista de excepciones para los nombres de acorde, para tratar casos que no están cubiertos, o bien que se tratan incorrectamente.

Los nombres de acorde predeterminados que se usan por LilyPond siguen las reglas dadas en el libro de Klaus Ignatzek's "Die Jazzmethode für Klavier 1"; el algoritmo para convertir acordes en nombres de acorde puede verse dentro del archivo `scm/chord-ignatzek-names.scm`. Las reglas adicionales se dan como excepciones de acordes y se almacenan dentro de la variable `ignatzekExceptions`, como está establecido en el archivo `ly/chord-modifiers-init.ly`.

Este fragmento de código modifica estas excepcioens en tres pasos.

1. Preparación de una música con acordes y elementos de marcado asociados. Por convención, la fundamental de cada acorde debe ser Do.

2. Llamar a la función de Scheme `sequential-music-to-chord-exceptions` para crear una lista de excepciones nueva, y después concatenarla con las existentes. Como `ignatzekExceptions` está preparado con el segundo parámetro de su función fijado al valor `#t` (para ignorar la fundamental de los acordes), tenemos que hacer lo mismo.
3. Registrar la nueva lista de excepciones.

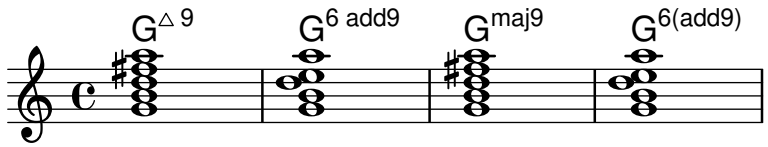
```
% Step 1: Define music with chords and markup for maj9 and 6(add9).
chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

% Step 2: Create extended exception list.
chExceptions =
#(append (sequential-music-to-chord-exceptions chExceptionMusic #t)
         ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  % Step 3: Register extended exception list.
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

<<
  \new ChordNames \theMusic
  \new Voice \theMusic
>>

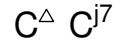
\layout {
  line-width = 10\cm
  ragged-right = ##f
}
```



Nombre de acorde maj7

La presentación del acorde de séptima mayor se puede ajustar mediante la propiedad de contexto `majorSevenSymbol`.

```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```



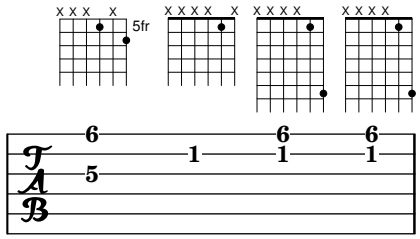
Acordes con digitaciones ampliadas para diagramas de trastes y tablaturas

A veces se requieren acordes con digitación ampliada o estirada. Sin embargo, si no se especifica lo contrario, la propiedad de contexto `maximumFretStretch` viene establecida al valor 4, lo que puede dar lugar a una advertencia “No hay ninguna cuerda para la nota ...” y la nota se omite. Podemos fijar `maximumFretStretch` a un valor adecuado o asignar explícitamente números de cuerda para todas las notas del acorde, para corregir esto.

```
% The code below prints two warnings for the second chord,
% which may be omitted by uncommenting the following line.
%
% #(for-each (lambda (x) (ly:expect-warning "No string for pitch")) (iota 2))

mus = {
  <c' bes'>
  <c'\2 bes'>
  \set maximumFretStretch = 5
  <c' bes'>
  <c'\2 bes'\1>
}

<<
  \new FretBoards \mus
  \new TabVoice \mus
>>
```

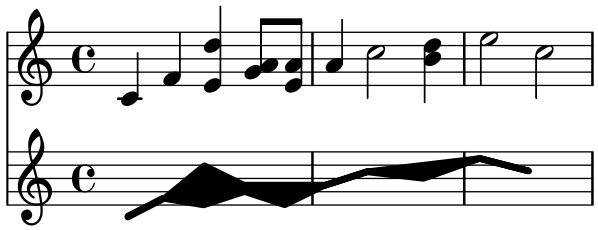


Clusters («racimos»)

Los «clusters» o racimos son un mecanismo para indicar la interpretación de un ámbito de notas al mismo tiempo.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}
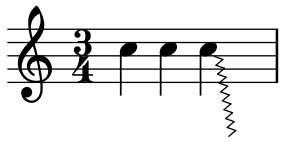
<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Glissando contemporáneo

Se puede tipografiar un gissando contemporáneo sin nota final utilizando una nota oculta y temporalización de cadenza.

```
\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}
```



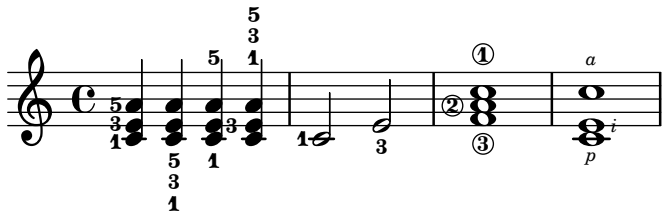
Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación usando la propiedad `fingeringOrientation`. Para que se tenga en cuenta la orientación de las digitaciones, la instrucción de digitación se debe utilizar dentro de una construcción de acorde (`<...>`) aunque sea una sola nota. Se puede controlar la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar usando las propiedades `stringNumberOrientation` y `strokeFingerOrientation`, respectivamente.

Estas propiedades se pueden establecer a una lista de entre uno y tres valores. Controlan si las digitaciones se pueden situar por encima (si aparece `up` en la lista), por debajo (si aparece `down`), a la izquierda (si aparece `left`), o a la derecha (si aparece `right`). Por contra, si una localización no aparece en la lista, no se coloca ninguna digitadión en ese lugar. LilyPond admite

esas restricciones y trata de encontrar la mejor ubicación para las digitaciones de las notas de los acordes siguientes. Observe que `left` y `right` se excluyen mutuamente (las digitaciones se pueden colocar en un lado o en el otro, no en los dos).

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Ejemplo de cencerro y campana

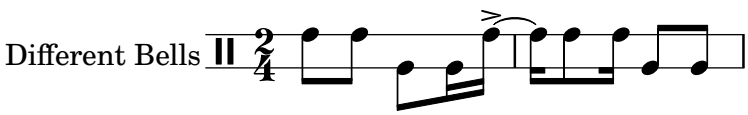
Dos campanas diferentes, escritas con ‘cb’ (cencerro) y ‘rb’ (campana, ridebell).

```
#(define mydrums '((ridebell default #f  3)
                  (cowbell  default #f -2)))

\new DrumStaff \with { instrumentName = #"Different Bells" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \set DrumStaff.clefPosition = 0.5
  \override DrumStaff.StaffSymbol.line-positions = #'(-2 3)
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)

  \time 2/4
  rb8 8 cb8 16 rb16-> ~ |
  16 8 16 cb8 8 |
}
```



Crear pentagramas en blanco

Para crear pentagramas en blanco, genere compases vacíos y después elimine el grabador de números de compás Bar_number_engraver del contexto Score, y los grabadores de la indicación de compás Time_signature_engraver, de la clave Clef_engraver y de los compases Bar_engraver del contexto de Staff.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

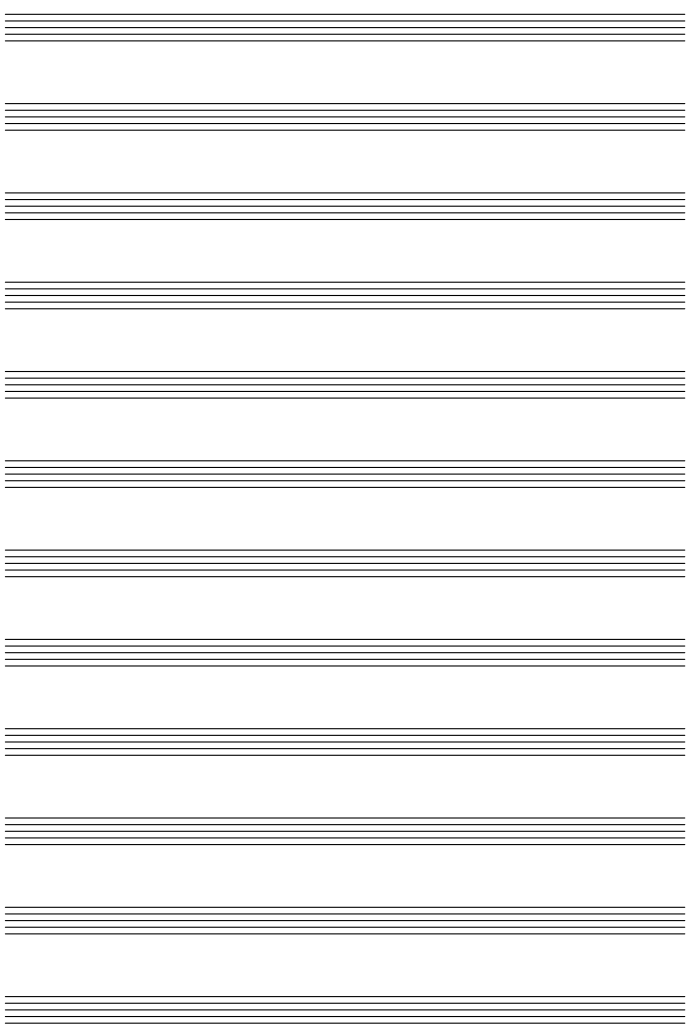
\book {
  \score {
    { \*12 { s1 \break } }

    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % for the documentation
  \paper {
    #(set-paper-size "a6")
    ragged-last-bottom = ##f
    line-width = 90\mm
    left-margin = 7.5\mm
    bottom-margin = 5\mm
    top-margin = 5\mm
    tagline = ##f
  }

  % uncomment these lines for "letter" size
  %{
  \paper {
    #(set-paper-size "letter")
    ragged-last-bottom = ##f
    line-width = 7.5\in
    left-margin = 0.5\in
    bottom-margin = 0.25\in
    top-margin = 0.25\in
    tagline = ##f
   }
  %}
```

```
  % uncomment these lines for "A4" size
  %{
  \paper {
    #(set-paper-size "a4")
    ragged-last-bottom = ##f
    line-width = 180\mm
    left-margin = 15\mm
    bottom-margin = 10\mm
    top-margin = 10\mm
    tagline = ##f
  }
  %}
}
```



Custos

Se pueden tipografiar «custos» en diferentes estilos.

```
\layout {
  ragged-right = ##t
}

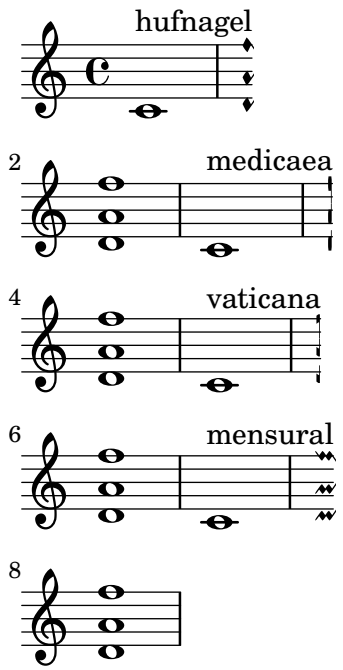
\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

    \override Staff.Custos.style = #'hufnagel
    c1^"hufnagel" \break
    <d a' f'>1

    \override Staff.Custos.style = #'medicaea
    c1^"medicaea" \break
    <d a' f'>1

    \override Staff.Custos.style = #'vaticana
    c1^"vaticana" \break
    <d a' f'>1

    \override Staff.Custos.style = #'mensural
    c1^"mensural" \break
    <d a' f'>1
  }
}
```



Demo of MIDI instruments

Problem: How to know which `midiInstrument` values would be best for your composition?

Solution: A LilyPond demo file. You have to compile this snippet by yourself and listen to the created MIDI output file.

```
melody = \relative c' {
  \tempo 4 = 150
  c4.\mf g c16 b' c d
  e16 d e f g4 g'4 r
  R1
}

\score {
  \new Voice \melody
  \layout { }
}

\score {
  \new Voice {
    r\mf
    % 1-8 keyboard
    \set Staff.midiInstrument = "acoustic grand" \melody
    \set Staff.midiInstrument = "bright acoustic" \melody
    \set Staff.midiInstrument = "electric grand" \melody
    \set Staff.midiInstrument = "honky-tonk" \melody
    \set Staff.midiInstrument = "electric piano 1" \melody
    \set Staff.midiInstrument = "electric piano 2" \melody
    \set Staff.midiInstrument = "harpsichord" \melody
    \set Staff.midiInstrument = "clav" \melody

    % 9-16 chrom percussion
    \set Staff.midiInstrument = "celesta" \melody
    \set Staff.midiInstrument = "glockenspiel" \melody
    \set Staff.midiInstrument = "music box" \melody
    \set Staff.midiInstrument = "vibraphone" \melody
    \set Staff.midiInstrument = "marimba" \melody
    \set Staff.midiInstrument = "xylophone" \melody
    \set Staff.midiInstrument = "tubular bells" \melody
    \set Staff.midiInstrument = "dulcimer" \melody

    % 17-24 organ
    \set Staff.midiInstrument = "drawbar organ" \melody
    \set Staff.midiInstrument = "percussive organ" \melody
    \set Staff.midiInstrument = "rock organ" \melody
    \set Staff.midiInstrument = "church organ" \melody
    \set Staff.midiInstrument = "reed organ" \melody
    \set Staff.midiInstrument = "accordion" \melody
    \set Staff.midiInstrument = "harmonica" \melody
    \set Staff.midiInstrument = "concertina" \melody

    % 25-32 guitar
    \set Staff.midiInstrument = "acoustic guitar (nylon)" \melody
    \set Staff.midiInstrument = "acoustic guitar (steel)" \melody
    \set Staff.midiInstrument = "electric guitar (jazz)" \melody
    \set Staff.midiInstrument = "electric guitar (clean)" \melody
    \set Staff.midiInstrument = "electric guitar (muted)" \melody
    \set Staff.midiInstrument = "overdriven guitar" \melody
```

```
\set Staff.midiInstrument = "distorted guitar" \melody
\set Staff.midiInstrument = "guitar harmonics" \melody

% 33-40 bass
\set Staff.midiInstrument = "acoustic bass" \melody
\set Staff.midiInstrument = "electric bass (finger)" \melody
\set Staff.midiInstrument = "electric bass (pick)" \melody
\set Staff.midiInstrument = "fretless bass" \melody
\set Staff.midiInstrument = "slap bass 1" \melody
\set Staff.midiInstrument = "slap bass 2" \melody
\set Staff.midiInstrument = "synth bass 1" \melody
\set Staff.midiInstrument = "synth bass 2" \melody

% 41-48 strings
\set Staff.midiInstrument = "violin" \melody
\set Staff.midiInstrument = "viola" \melody
\set Staff.midiInstrument = "cello" \melody
\set Staff.midiInstrument = "contrabass" \melody
\set Staff.midiInstrument = "tremolo strings" \melody
\set Staff.midiInstrument = "pizzicato strings" \melody
\set Staff.midiInstrument = "orchestral harp" \melody
\set Staff.midiInstrument = "timpani" \melody

% 49-56 ensemble
\set Staff.midiInstrument = "string ensemble 1" \melody
\set Staff.midiInstrument = "string ensemble 2" \melody
\set Staff.midiInstrument = "synthstrings 1" \melody
\set Staff.midiInstrument = "synthstrings 2" \melody
\set Staff.midiInstrument = "choir aahs" \melody
\set Staff.midiInstrument = "voice oohs" \melody
\set Staff.midiInstrument = "synth voice" \melody
\set Staff.midiInstrument = "orchestra hit" \melody

% 57-64 brass
\set Staff.midiInstrument = "trumpet" \melody
\set Staff.midiInstrument = "trombone" \melody
\set Staff.midiInstrument = "tuba" \melody
\set Staff.midiInstrument = "muted trumpet" \melody
\set Staff.midiInstrument = "french horn" \melody
\set Staff.midiInstrument = "brass section" \melody
\set Staff.midiInstrument = "synthbrass 1" \melody
\set Staff.midiInstrument = "synthbrass 2" \melody

% 65-72 reed
\set Staff.midiInstrument = "soprano sax" \melody
\set Staff.midiInstrument = "alto sax" \melody
\set Staff.midiInstrument = "tenor sax" \melody
\set Staff.midiInstrument = "baritone sax" \melody
\set Staff.midiInstrument = "oboe" \melody
\set Staff.midiInstrument = "english horn" \melody
\set Staff.midiInstrument = "bassoon" \melody
\set Staff.midiInstrument = "clarinet" \melody
```

```
% 73-80 pipe
\set Staff.midiInstrument = "piccolo" \melody
\set Staff.midiInstrument = "flute" \melody
\set Staff.midiInstrument = "recorder" \melody
\set Staff.midiInstrument = "pan flute" \melody
\set Staff.midiInstrument = "blown bottle" \melody
\set Staff.midiInstrument = "shakuhachi" \melody
\set Staff.midiInstrument = "whistle" \melody
\set Staff.midiInstrument = "ocarina" \melody

% 81-88 synth lead
\set Staff.midiInstrument = "lead 1 (square)" \melody
\set Staff.midiInstrument = "lead 2 (sawtooth)" \melody
\set Staff.midiInstrument = "lead 3 (calliope)" \melody
\set Staff.midiInstrument = "lead 4 (chiff)" \melody
\set Staff.midiInstrument = "lead 5 (charang)" \melody
\set Staff.midiInstrument = "lead 6 (voice)" \melody
\set Staff.midiInstrument = "lead 7 (fifths)" \melody
\set Staff.midiInstrument = "lead 8 (bass+lead)" \melody

% 89-96 synth pad
\set Staff.midiInstrument = "pad 1 (new age)" \melody
\set Staff.midiInstrument = "pad 2 (warm)" \melody
\set Staff.midiInstrument = "pad 3 (polysynth)" \melody
\set Staff.midiInstrument = "pad 4 (choir)" \melody
\set Staff.midiInstrument = "pad 5 (bowed)" \melody
\set Staff.midiInstrument = "pad 6 (metallic)" \melody
\set Staff.midiInstrument = "pad 7 (halo)" \melody
\set Staff.midiInstrument = "pad 8 (sweep)" \melody

% 97-104 synth effects
\set Staff.midiInstrument = "fx 1 (rain)" \melody
\set Staff.midiInstrument = "fx 2 (soundtrack)" \melody
\set Staff.midiInstrument = "fx 3 (crystal)" \melody
\set Staff.midiInstrument = "fx 4 (atmosphere)" \melody
\set Staff.midiInstrument = "fx 5 (brightness)" \melody
\set Staff.midiInstrument = "fx 6 (goblins)" \melody
\set Staff.midiInstrument = "fx 7 (echoes)" \melody
\set Staff.midiInstrument = "fx 8 (sci-fi)" \melody

% 105-112 ethnic
\set Staff.midiInstrument = "sitar" \melody
\set Staff.midiInstrument = "banjo" \melody
\set Staff.midiInstrument = "shamisen" \melody
\set Staff.midiInstrument = "koto" \melody
\set Staff.midiInstrument = "kalimba" \melody
\set Staff.midiInstrument = "bagpipe" \melody
\set Staff.midiInstrument = "fiddle" \melody
\set Staff.midiInstrument = "shanai" \melody

% 113-120 percussive
```

```
    \set Staff.midiInstrument = "tinkle bell" \melody
    \set Staff.midiInstrument = "agogo" \melody
    \set Staff.midiInstrument = "steel drums" \melody
    \set Staff.midiInstrument = "woodblock" \melody
    \set Staff.midiInstrument = "taiko drum" \melody
    \set Staff.midiInstrument = "melodic tom" \melody
    \set Staff.midiInstrument = "synth drum" \melody
    \set Staff.midiInstrument = "reverse cymbal" \melody

    % 121-128 sound effects
    \set Staff.midiInstrument = "guitar fret noise" \melody
    \set Staff.midiInstrument = "breath noise" \melody
    \set Staff.midiInstrument = "seashore" \melody
    \set Staff.midiInstrument = "bird tweet" \melody
    \set Staff.midiInstrument = "telephone ring" \melody
    \set Staff.midiInstrument = "helicopter" \melody
    \set Staff.midiInstrument = "applause" \melody
    \set Staff.midiInstrument = "gunshot" \melody
  }
  \midi { }
}
```
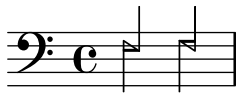


Direction of merged ‘fa’ shape note heads

Using property `NoteCollision.fa-merge-direction`, the direction of “fa” shape note heads (“fa”, “faThin”, etc.) can be controlled independently of the stem direction if two voices with the same pitch and different stem directions are merged. If this property is not set, the “down” glyph variant is used.

```
{
  \clef bass

  << { \aikenHeads
       f2
       \override Staff.NoteCollision.fa-merge-direction = #UP
       f2 }
  \\ { \aikenHeads
       f2
       f2 }
  >>
}
```



Empotrar PostScript nativo dentro de un bloque \markup

Se puede insertar código PostScript directamente dentro de un bloque `\markup`.

En general se recomienda usar en su lugar las instrucciones de marcado nativas de LilyPond tales como \polygon, que se pueden usar con todos los *backends* de LilyPond.

```
\relative c'' {
  a2-\markup \postscript "0 3 moveto
                          5 2 rlineto
                          stroke"
    -\markup \postscript "[1 1] 0 setdash
                          0 0 moveto
                          5 2 rlineto
                          stroke"
  b2-\markup \postscript "1 1 moveto
                          0 0 1 2 8 4 10 2 rcurveto
                          stroke"
  a'1
}
```



Los grabadores, uno por uno

LilyPond trata los distintos elementos necesarios para el tipografiado de una partitura utilizando «plugins» o complementos añadidos. Cada uno de los complementos se conoce como un *grabador*. En este ejemplo, algunos grabadores se van activando uno por uno, en el orden siguiente:

- cabeza de las notas,
- el símbolo del pentagrama,
- clave,
- plicas,
- barras, ligaduras de expresión, acentos,
- alteraciones, líneas divisorias, indicación del compás, y armadura.

Los grabadores se encuentran agrupados. Por ejemplo, las cabezas de nota, ligaduras de expresión, barras de corchea, etc. forman un contexto de voz, `Voice`. Los grabadores de la armadura, alteraciones, líneas de compás, etc. forman un contexto de pentagrama, `Staff`.

```
topVoice = \relative c' {
  \key d \major
  es8([ g] a[ fis])
  b4
  b16[-. b-. b-. cis-.]
  d4->
}

% empty staff and voice contexts
MyStaff = \context {
  \type Engraver_group
  \name Staff
  \accepts Voice
  \defaultchild Voice
}
MyVoice = \context {
```

```
  \type Engraver_group
  \name Voice
}

% add note heads
MyVoice = \context {
  \MyVoice
  \consists Note_heads_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add staff
MyStaff = \context {
  \MyStaff
  \consists Staff_symbol_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}
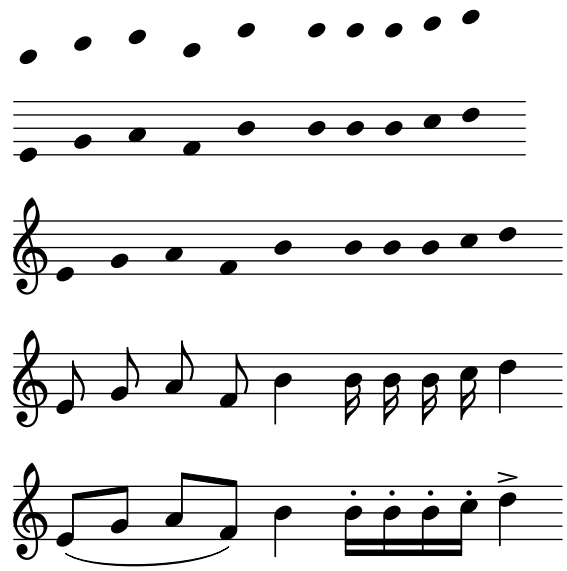
% add clef
MyStaff = \context {
  \MyStaff
  \consists Clef_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}
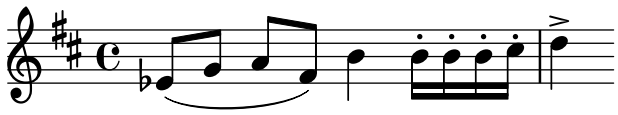
% add stems
MyVoice = \context {
  \MyVoice
  \consists Stem_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
```

```
    \context { \MyVoice }
  }
}

% add beams, slurs, and accents
MyVoice = \context {
  \MyVoice
  \consists Beam_engraver
  \consists Slur_engraver
  \consists Script_engraver
  \consists Rhythmic_column_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add accidentals, bar, time signature, and key signature
MyStaff = \context {
  \MyStaff
  \consists Accidental_engraver
  \consists Bar_engraver
  \consists Time_signature_engraver
  \consists Key_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}
```





Notación del flamenco

Para la guitarra flamenca se utilizan ciertos elementos de notación especiales.

- Un símbolo para indicar un *golpe* sobre la caja de la guitarra con la uña del dedo anular.
- Una flecha para indicar rasgueos y su dirección.
- Letras para las digitaciones ("p": pulgar, "i": índice, "m": medio, "a": anular y "x": meñique).
- *Rasgueados* de 3 y 4 dedos: hacia arriba con todos los dedos y terminando con arriba y abajo con el dedo índice.
- *Abanicos*: rasgueos en serie con el pulgar, hacia abajo y el meñique y el índice hacia arriba (hay también un *abanico 2* en el que los dedos medio y anular se usan en lugar del meñique).
- *Alza púa*: pulsaciones rápidas con el pulgar.

Casi todas las figuras utilizan flechas combinadas con digitaciones; con los abanicos y los rasgueados, las notas se imprimen con cabeza solo en el primer acorde.

Este fragmento de código contiene código de tipo cabecera que se puede copiar como `flamenco.ly` e incluirse en los documentos fuente.

```
%%%%%%  Cut here ----- Start of `flamenco.ly`.

% Text indicators.
abanico = ^\markup \small { \italic Abanico }
rasgueado = ^\markup \small { \italic Ras. }
alzapua = ^\markup \small { \italic Alzapua }

% Finger stroke symbols.
strokeUp = \markup {
  \combine
    \override #'(thickness . 1.3) \draw-line #'(0 . 2)
    \raise #2 \arrow-head #Y #UP ##f }
strokeDown = \markup {
  \combine
    \arrow-head #Y #DOWN ##f
    \override #'(thickness . 1.3) \draw-line #'(0 . 2) }

% Golpe symbol.
golpe = \markup {
  \filled-box #'(0 . 1) #'(0 . 1) #0
  \hspace #-1.6
  \with-color #white
  \filled-box #'(0.15 . 0.85) #'(0.15 . 0.85) #0
}

% Strokes, fingers, and golpe command.
RHp = \rightHandFinger #1
RHi = \rightHandFinger #2
RHm = \rightHandFinger #3
RHa = \rightHandFinger #4
```

```
RHx = \rightHandFinger #5
RHu = \rightHandFinger \strokeUp
RHd = \rightHandFinger \strokeDown
RHg = \rightHandFinger \golpe

% Various shorthands.
tupletOff = {
  \once \omit TupletNumber
  \once \omit TupletBracket
}

tupletsOff = {
  \omit TupletNumber
  \override TupletBracket.bracket-visibility = #'if-no-beam
}

tupletsOn = {
  \override TupletBracket.bracket-visibility = #'default
  \undo \omit TupletNumber
}

headsOff = {
  \hide TabNoteHead
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

headsOn = {
  \override TabNoteHead.transparent = ##f
  \override NoteHead.transparent = ##f
  \override NoteHead.no-ledgers = ##f
}

%%%%%%  Cut here ----- End of `flamenco.ly`.

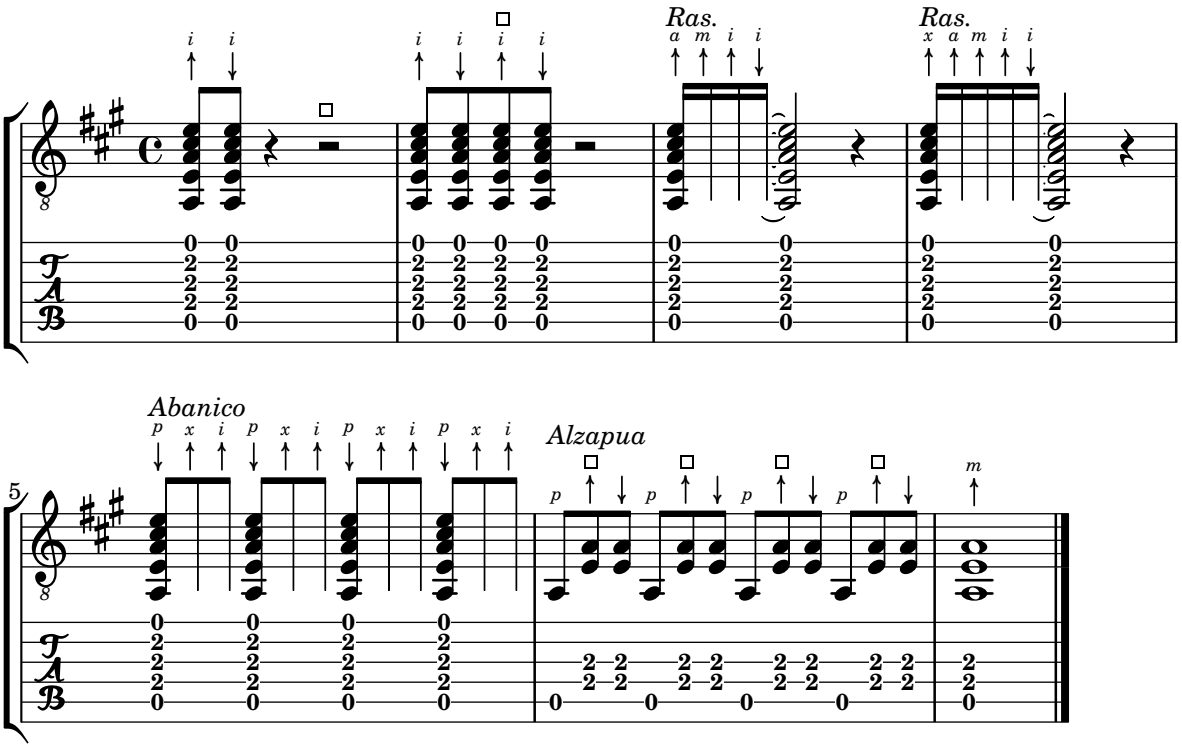

part = \relative c' {
  \set strokeFingerOrientations = #'(up)
  \key a\major

  <a, e' a cis e\RHu\RHi>8
    <a e' a cis e\RHd\RHi>8
    r4
    r2^\markup\golpe |
  <a e' a cis e\RHu\RHi>8
    <a e' a cis e\RHd\RHi>8
    <a e' a cis e\RHu\RHi\RHg>8
    <a e' a cis e\RHd\RHi>8
    r2 |
  <a e' a cis e\RHu\RHa>16\rasgueado
    \headsOff
    <a e' a cis e\RHu\RHm>
```

```
    <a e' a cis e\RHu\RHi>
    <a e' a cis e\RHd\RHi>~
    \headsOn
    <a e' a cis e>2
    r4 |
  \tupletOff
    \tuplet 5/4 {
      <a e' a cis e\RHu\RHx>16\rasgueado
      \headsOff
      <a e' a cis e\RHu\RHa>
      <a e' a cis e\RHu\RHm>
      <a e' a cis e\RHu\RHi>
      <a e' a cis e\RHd\RHi>~
      \headsOn
    }
    <a e' a cis e>2
    r4 |
  <>\abanico
    \tupletsOff
    \repeat unfold 4 {
      \tuplet 3/2 {
        <a e' a cis e\RHd\RHp>8
        \headsOff
        <a e' a cis e\RHu\RHx>
        <a e' a cis e\RHu\RHi>
        \headsOn
      }
    }
    \tupletsOff |
  <>\alzapua
  \override Beam.positions = #'(2 . 2)
    \repeat unfold 4 {
      \tuplet 3/2 {
        a8\RHp
        <e' a\RHu\RHg>
        <e a\RHd>
      }
    }
    \tupletsOn |
  <a, e' a\RHu\RHm>1 \bar "|."
}

\score {
  \new StaffGroup <<
    \context Staff = "part" {
      \clef "G_8"
      \part
    }
    \context TabStaff {
      \part
    }
  >>
```

```
  \layout {
    ragged-right = ##t
  }
}
```



Ejemplo de cajas chinas grave y aguda

Dos cajas chinas, escritas como 'wbh' (*high woodblock*) y 'wbl' (*low woodblock*). La longitud de la línea divisoria se ha alterado con una instrucción `\override` para que no sea demasiado corta. Las posiciones de las dos líneas de la pauta se tienen que definir explícitamente también.

```
% These lines define the position of the woodblocks in the stave;
% if you like, you can change it or you can use special note heads
% for the woodblocks.
#(define mydrums '((hiwoodblock default #f  3)
                  (lowoodblock default #f -2)))

woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

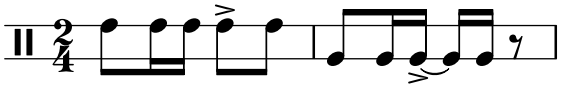
  % This is necessary; if not entered,
  % the barline would be too short!
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)
  % small correction for the clef:
  \set DrumStaff.clefPosition = 0.5
}

\new DrumStaff {
  % with this you load your new drum style table
```

```
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \woodstaff

  \drummode {
    \time 2/4
    wbh8 16 16 8-> 8 |
    wbl8 16 16-> ~ 16 16 r8 |
  }
}
```



Cómo cambiar la posición de un diagrama de posiciones

Si queremos mover un diagrama de posiciones de acorde, por ejemplo, para evitar la colisión, o situarlo entre dos notas, tenemos varias posibilidades.

1) Modificar el valor de las propiedades `padding` (relleno) o `extra-offset` (desplazamiento adicional) como se muestra en la primera línea.

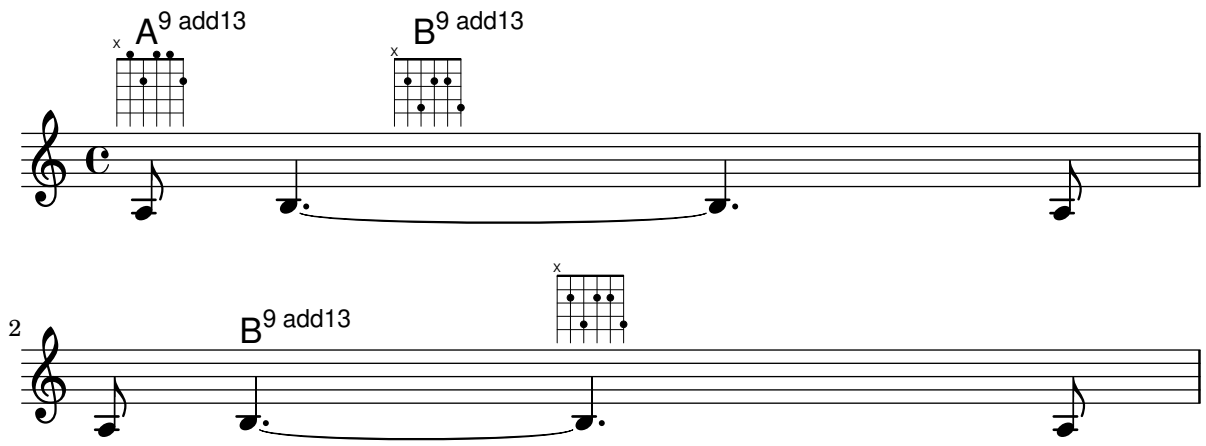
2) Podemos añadir una voz invisible y adjuntar los diagramas de posiciones a las notas invisibles de esa voz (como se ve en la segunda línea).

Si tenemos que mover el diagrama según una posición rítmica dentro del compás (en el ejemplo, la tercera parte del compás) es mejor el segundo ejemplo, porque el diagrama se alinea con el tercer pulso por sí solo.

```
harmonies = \chordmode
{
  a8:13
  \once \override ChordNames.ChordName.extra-offset = #'(10 . 0)
  b8:13 s4. |
  s2 b2:13
}

\score {
  <<
    \new ChordNames \harmonies
    \new Staff {
      % Method 1.
      a8^\markup \fret-diagram "6-x;5-0;4-2;3-0;2-0;1-2;"
      \once \override TextScript.extra-offset = #'(10 . 0)
      b4.~^\markup \fret-diagram "6-x;5-2;4-4;3-2;2-2;1-4;"
        b4. a8 | \break

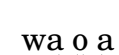
      % Method 2.
      <<
        { a8 b4.~ b4. a8 }
        { s2 s2^\markup \fret-diagram  "6-x;5-2;4-4;3-2;2-2;1-4;" }
      >> |
    }
  >>
}
```



Cómo poner ligaduras entre sílabas de la letra

Se puede hacer separando las sílabas mediante símbolos de tilde curva.

```
\lyrics {
  wa~o~a
}
```
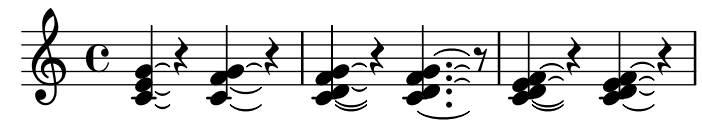


Ligaduras *laissez vibrer*

Las ligaduras "laissez vibrer" (dejar vibrar) tienen un tamaño fijo. Se puede ajustar su posición usando la propiedad `tie-configuration`.

Véase también el fragmento de código "Ligaduras laissez vibrer largas".

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
     = #`((-7 . ,DOWN)
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



Ejemplo de percusión

Un ejemplo breve extraído de la *Historia de un soldado* de Strawinsky.

```
#(define mydrums '((bassdrum   default #f  4)
                   (snare      default #f -4)
                   (tambourine default #f  0)))

U = \stemUp
D = \stemDown

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}

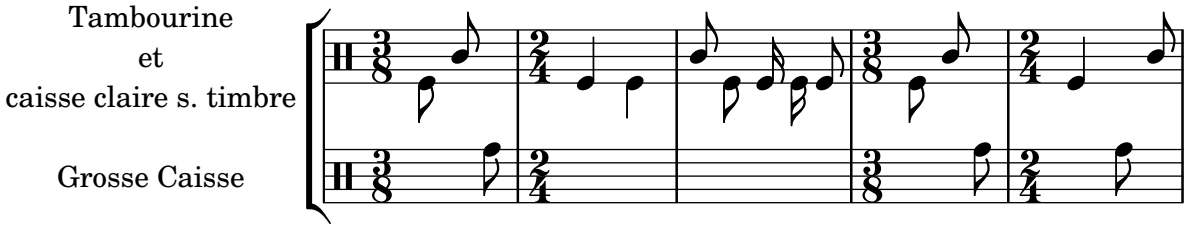
drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
      sn4 \D sn4 |
      \U tamb8 \D sn \U sn16 \D sn \U sn8 |
      \D sn8 \U tamb s |
      \U sn4 s8 \U tamb
    }
  >>
}

drumsB = \drummode {
  s4 bd8 s2*2 s4 bd8 s4 bd8 s
}

\layout {
  indent = 40\mm
  \context {
    \DrumStaff
    drumStyleTable = #(alist->hash-table mydrums)
  }
}

\score {
  \new StaffGroup <<
    \new DrumStaff \with {
      instrumentName = \markup \center-column {
        "Tambourine"
        "et"
        "caisse claire s. timbre" }
    } \drumsA
    \new DrumStaff \with {
      instrumentName = "Grosse Caisse"
    }\drumsB
  >>
```

```
}
```



Ejemplo de música para tam-tam

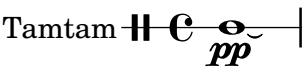
Un ejemplo de tam-tam, escrito como 'tt'.

```
#(define mydrums '((tamtam default #f 0)))

\new DrumStaff \with { instrumentName = #"Tamtam" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  tt 1 \pp \laissezVibrer
}
```



Ejemlo de pandereta

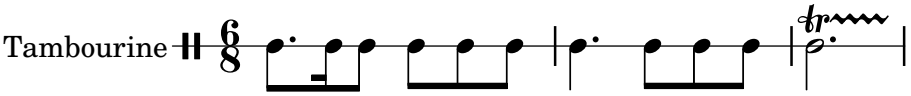
A tambourine example, entered with 'tamb'.

```
#(define mydrums '((tambourine default #f 0)))

\new DrumStaff \with { instrumentName = #"Tambourine" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
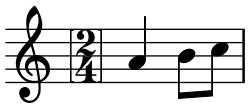
  \time 6/8
  tamb8. 16 8 8 8 8 |
  tamb4. 8 8 8 |
  % The trick with the scaled duration and the shorter rest
  % is neccessary for the correct ending of the trill-span!
  tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
}
```



Indicación de compás entre corchetes

La indicación de compás se puede encerrar entre corchetes.

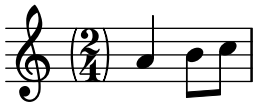
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Compás entre paréntesis

Se puede encerrar la indicación de compás entre paréntesis.

```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Uso de una voz adicional para los saltos de línea y de página

Con frecuencia es más fácil gestionar la información de los saltos de línea y de página manteniéndola separada de la música, por medio de la introducción de una vos adicional que contiene solamente silencios de separacion junto a las instrucciones `\break`, `\pageBreak` y otras informaciones de disposición.

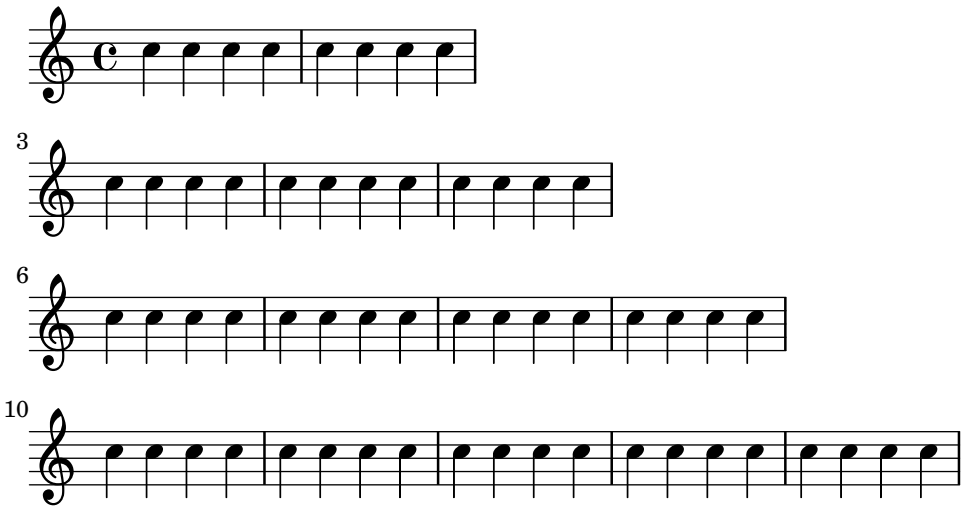
Este patrón se hace especialmente útil cuando se está sobreescribiendo la propiedad `line-break-system-details` y las otras útiles pero largas propiedades del objeto gráfico `NonMusicalPaperColumn`.

```
music = \relative c'' { c4 c c c }

\score {
  \new Staff <<
    \new Voice {
      s1*2 \break
      s1*3 \break
      s1*4 \break
      s1*5 \break
    }
    \new Voice {
      \*2 \music
      \*3 \music
      \*4 \music
      \*5 \music
    }
```

```
  >>
}

\paper {
  indent = 0
  line-width = 140\mm
  ragged-right = ##t
}
```



Listado de los diagramas para viento madera

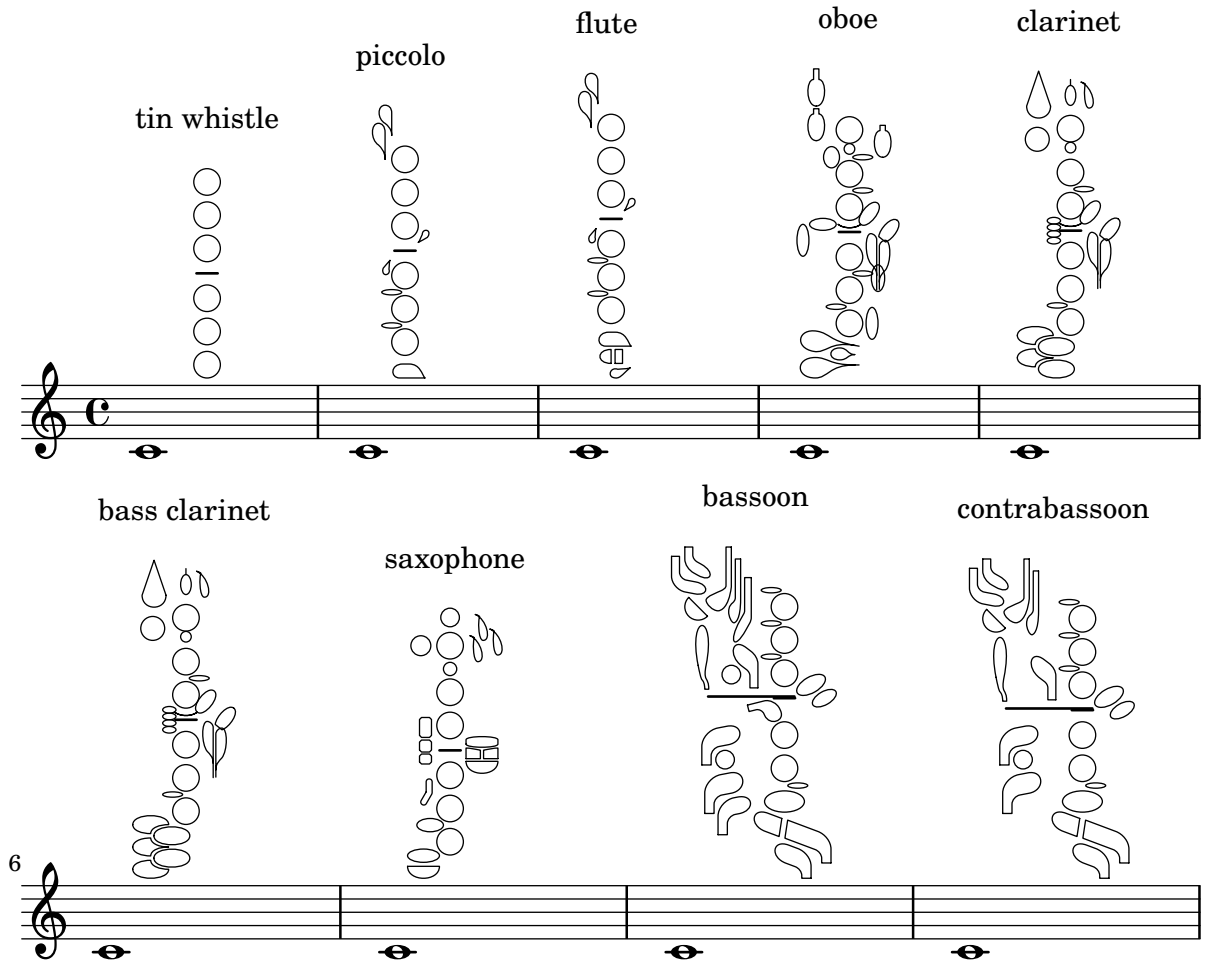
El fragmento de música que aparece a continuación presenta todos los diagramas de viento madera que se encuentran definidos en LilyPond por el momento.

```
\relative c' {
  \textLengthOn
  c1^\markup \center-column { "tin whistle"
                              " "
                              \woodwind-diagram #'tin-whistle #'() }
  c1^\markup \center-column { "piccolo"
                              " "
                              \woodwind-diagram #'piccolo #'() }
  c1^\markup \center-column { "flute"
                              " "
                              \woodwind-diagram #'flute #'() }
  c1^\markup \center-column { "oboe"
                              " "
                              \woodwind-diagram #'oboe #'() }
  c1^\markup \center-column { "clarinet"
                              " "
                              \woodwind-diagram #'clarinet #'() }
  \break

  c1^\markup \center-column { "bass clarinet"
                              " "
                              \woodwind-diagram #'bass-clarinet #'() }
  c1^\markup \center-column { "saxophone"
                              " "
```

```
                                \woodwind-diagram #'saxophone #'() }
  c1^\markup \center-column { "bassoon"
                                " "
                                \woodwind-diagram #'bassoon #'() }
  c1^\markup \center-column { "contrabassoon"
                                " "
                                \woodwind-diagram #'contrabassoon #'() }
}

\paper {
  system-system-spacing.padding = 5
}
```



35 Symbols and glyphs

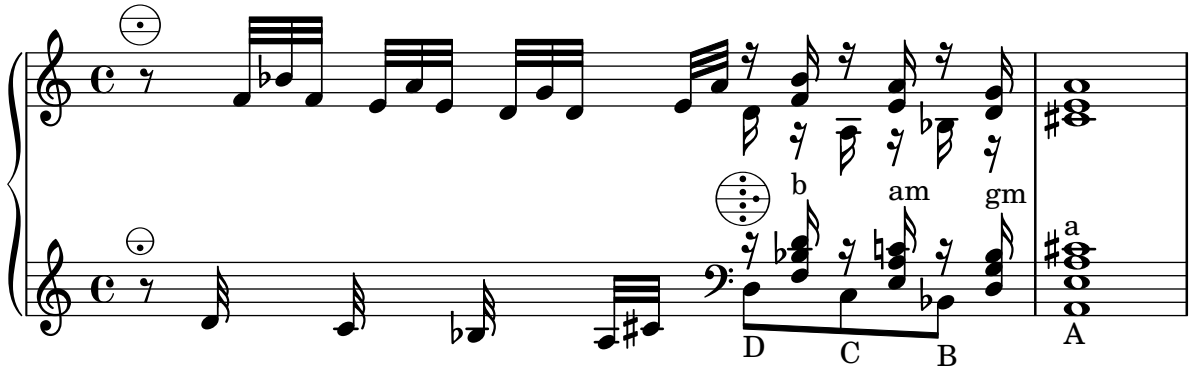
Símbolos de registración de acordeón

Los símbolos de registración de acordeón están disponibles como elementos de \markup y como eventos musicales autocontenidos (pues los cambios de registro tienden a ocurrir entre eventos de música). Los registros del bajo no están demasiado estandarizados. Las instrucciones existentes están disponibles en el apartado 'Símbolos de acordeón' en la Referencia de la notación (https://lilypond.org/doc/v2.24/Documentation/notation/accordion#discant-symbols).

```
#(use-modules (lily accreg))

\new PianoStaff <<
  \new Staff \relative {
    \clef treble
    \discant "10"
    r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
    <<
      { r16 <f bes> r <e a> r <d g> }
      \\
      { d r a r bes r }
    >> |
    <cis e a>1
  }

  \new Staff \relative {
    \clef treble
    \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef bass \stdBass "Master"
    <<
      { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
      <e a cis>1^"a" }
      \\
      { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```



Añadir indicadores a los pentagramas que se dividen después de un salto de línea

Este fragmento de código define las instrucciones \splitStaffBarLine, \convUpStaffBarLine y \convDownStaffBarLine, que añaden flechas en las sobre una línea divisoria para denotar que cada una de las voces que comparten un pentagrama continúan en su propio pentagrama en el siguiente sistema, o que las voces así divididas se recombinan.

Observe que la implementación de este fragmento de código traza unas flechas sin dimensiones en el margen derecho. Para la impresión formal, esto no produce ningún problema. Sin embargo, es necesario aumentar las dimensiones de la caja circundante en el sentido horizontal si se procesa el código como imagen para evitar el recorte, como se muestra abajo.

```
#(define-markup-command (arrow-at-angle layout props angle-deg length fill)
   (number? number? boolean?)
   (let* ((PI-OVER-180 (/ (atan 1 1) 34))
          (degrees->radians (lambda (degrees) (* degrees PI-OVER-180)))
          (angle-rad (degrees->radians angle-deg))
          (target-x (* length (cos angle-rad)))
          (target-y (* length (sin angle-rad))))
     (interpret-markup layout props
       (markup
        #:translate (cons (/ target-x 2) (/ target-y 2))
        #:rotate angle-deg
        #:translate (cons (/ length -2) 0)
        #:concat (#:draw-line (cons length 0)
                   #:arrow-head X RIGHT fill)))))

splitStaffBarLineMarkup = \markup \with-dimensions #'(0 . 0) #'(0 . 0) {
  \combine
  \arrow-at-angle #45 #(sqrt 8) ##t
  \arrow-at-angle #-45 #(sqrt 8) ##t
}

splitStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob splitStaffBarLineMarkup)
      0))
  \break
}

convDownStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . -.13)\arrow-at-angle #-45 #(sqrt 8) ##t
        }#})
```

```
      0))
  \break
}

convUpStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . .14)\arrow-at-angle #45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

\paper {
  indent = 10\mm
  short-indent = 10\mm
  line-width = 8\cm
}

separateSopranos = {
  \set Staff.instrumentName = "AI AII"
  \set Staff.shortInstrumentName = "AI AII"
  \splitStaffBarLine
  \change Staff = "up"
}
convSopranos = {
  \convDownStaffBarLine
  \change Staff = "shared"
  \set Staff.instrumentName = "S A"
  \set Staff.shortInstrumentName = "S A"
}

sI = {
  \voiceOne
  \*4 f''2
  \separateSopranos
  \*4 g''2
  \convSopranos
  \*4 c''2
}
sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \*4 d''2
}
aI = {
```

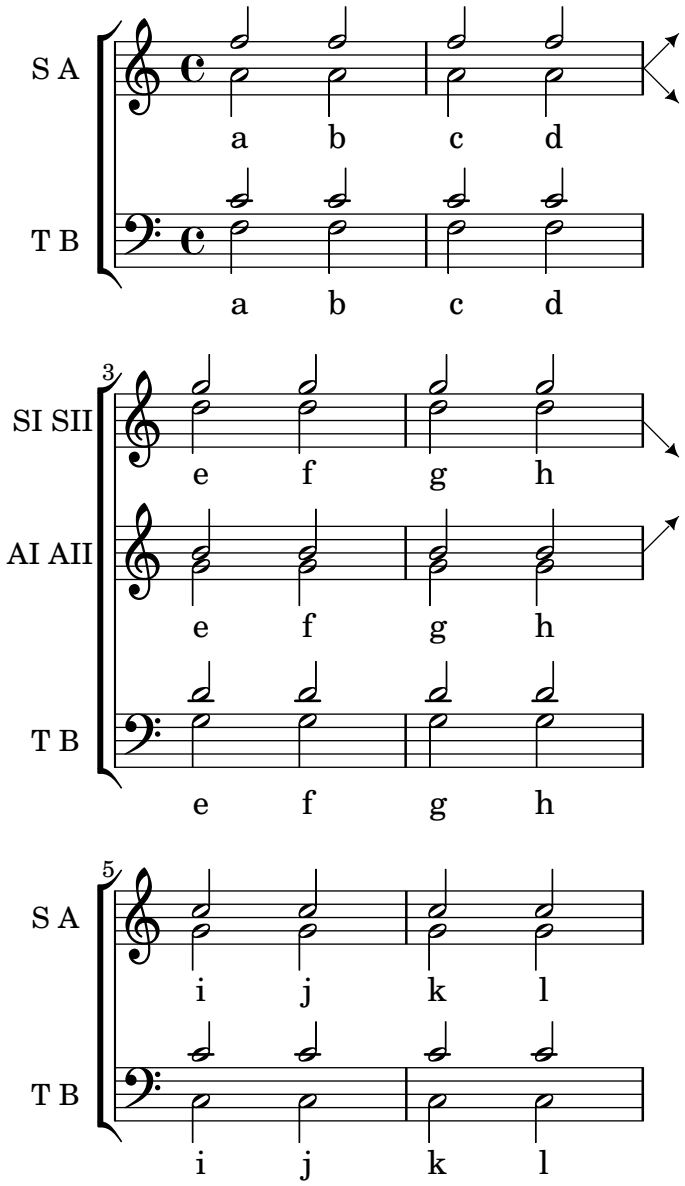
```
  \voiceTwo
  \*4 a'2
  \voiceOne
  \*4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \*4 g'2
}
aII = {
  s1*2
  \voiceTwo
  \*4 g'2
}
ten = {
  \voiceOne
  \*4 c'2
  \*4 d'2
  \*4 c'2
}
bas = {
  \voiceTwo
  \*4 f2
  \*4 g2
  \*4 c2
}

\markup \pad-x #3  % avoid cropping
  \score {
    <<
      \new ChoirStaff <<
        \new Staff = up \with {
          instrumentName = "SI SII"
          shortInstrumentName = "SI SII"
        } {
          s1*4
        }

        \new Staff = shared \with {
          instrumentName = "S A"
          shortInstrumentName = "S A"
        } <<
          \new Voice = sopI \sI
          \new Voice = sopII \sII
          \new Voice = altI \aI
          \new Voice = altII \aII
        >>
        \new Lyrics \with {
          alignBelowContext = up
        }
        \lyricsto sopII { e f g h }
        \new Lyrics \lyricsto altI { a b c d e f g h i j k l }
```

```
        \new Staff = men \with {
          instrumentName = "T B"
          shortInstrumentName = "T B"
        } <<
          \clef F
          \new Voice = ten \ten
          \new Voice = bas \bas
        >>
        \new Lyrics \lyricsto bas { a b c d e f g h i j k l }
      >>
    >>

    \layout {
      \context {
        \Staff \RemoveEmptyStaves
        \override VerticalAxisGroup.remove-first = ##t
      }
    }
}
```



Tipografía de música antigua

Este fragmento de código presenta muchos de los símbolos incluidos en la fuente Emmentaler que LilyPond usa para tipografiar la música antigua.

```
m = { c1 e f ges cis' \bar "||" }

\score {
  \new VaticanaVoice {
    \clef "vaticana-fa2"
    \key es \major
    \textMark \markup \rounded-box "Vaticana clefs, custos and note heads"

    \override NoteHead.style = #'vaticana.punctum
    <>^"vaticana.punctum" \m

    \override NoteHead.style = #'vaticana.inclinatum
    <>^"vaticana.inclinatum" \m
```

```
\override NoteHead.style = #'vaticana.quilisma
<>^"vaticana.quilisma" \m

\clef "vaticana-fa1"
\override NoteHead.style = #'vaticana.plica
<>^"vaticana.plica" \m

\override NoteHead.style = #'vaticana.reverse.plica
<>^"vaticana.reverse.plica" \m

\override NoteHead.style = #'vaticana.punctum.cavum
<>^"vaticana.punctum.cavum" \m

\override NoteHead.style = #'vaticana.lpes
<>^"vaticana.punctum.lpes" \m

\override NoteHead.style = #'vaticana.upes
<>^"vaticana.punctum.upes" \m

\override NoteHead.style = #'vaticana.vupes
<>^"vaticana.punctum.vupes" \m

\override NoteHead.style = #'vaticana.linea.punctum
<>^"vaticana.punctum.linea" \m

\override NoteHead.style = #'vaticana.epiphonus
<>^"vaticana.punctum.epiphonus" \m

\override NoteHead.style = #'vaticana.cephalicus
<>^"vaticana.punctum.cephalicus" \m

\break

\textMark \markup \rounded-box "Medicaea clefs, custos and note heads"
\set VaticanaStaff.alterationGlyphs =
  #alteration-medicaea-glyph-name-alist
\override VaticanaStaff.Custos.style = #'medicaea

\clef "medicaea-fa2"
\override NoteHead.style = #'medicaea.punctum
<>^"medicaea.punctum" \m

\clef "medicaea-do2"
\override NoteHead.style = #'medicaea.inclinatum
<>^"medicaea.inclinatum" \m

\override NoteHead.style = #'medicaea.virga
<>^"medicaea.virga" \m

\clef "medicaea-fa1"
\override NoteHead.style = #'medicaea.rvirga
```

```
    <>^"medicaea.rvirga" \m

    \break

    \textMark \markup \rounded-box "Hufnagel clefs, custos and note heads"
    \set Staff.alterationGlyphs =
      #alteration-hufnagel-glyph-name-alist
    \override VaticanaStaff.Custos.style = #'hufnagel
    \clef "hufnagel-fa2"

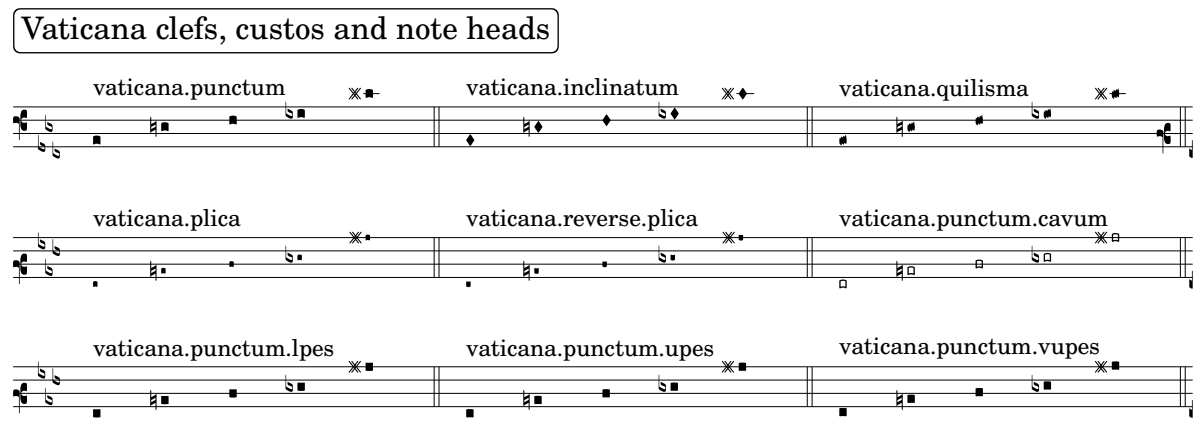
    \break

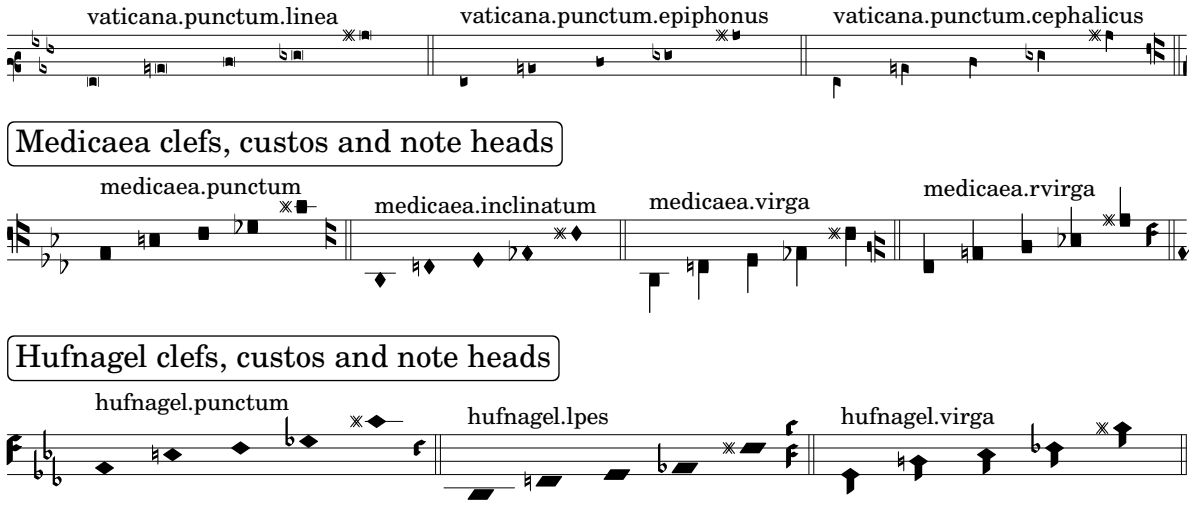
    \override NoteHead.style = #'hufnagel.punctum
    <>^"hufnagel.punctum" \m

    \clef "hufnagel-do2"
    \override NoteHead.style = #'hufnagel.lpes
    <>^"hufnagel.lpes" \m

    \clef "hufnagel-do-fa"
    \override NoteHead.style = #'hufnagel.virga
    <>^"hufnagel.virga" \m
  }

  \layout {
    \context {
      \Score
      \override TextScript.font-size = #-2
      \override TextMark.break-align-symbols = #'(left-edge clef staff-bar)
      \override TextMark.padding = 4
      \omit BarNumber
    }
    \context {
      \VaticanaStaff
      alterationGlyphs =
        #alteration-vaticana-glyph-name-alist
    }
  }
}
```





Marcas de respiración

Las marcas de respiración están disponibles para varios gustos: coma (la predeterminada), raya corta, uve y "vías del tren" (cesura).

```
\new Staff \relative c'' {
  \key es \major
  \time 3/4
  % this bar contains no \breathe
  << { g4 as g } \\ { es4 bes es } >> |
  % Modern notation:
  % by default, \breathe uses the rcomma, just as if saying:
  % \override BreathingSign.text =
  %   #(make-musicglyph-markup "scripts.rcomma")
  << { g4 as g } \\ { es4 \breathe bes es } >> |
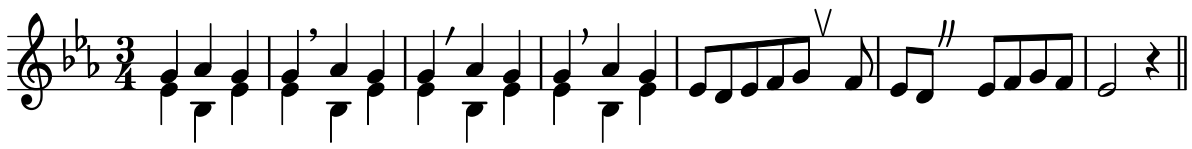
  % rvarcomma and lvarcomma are variations of the default rcomma
  % and lcomma
  % N.B.: must use Staff context here, since we start a Voice below
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % raltcomma and laltcomma are alternative variations of the
  % default rcomma and lcomma
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.raltcomma" }
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % vee
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.uupbow" }
  es8[ d es f g] \breathe f |

  % caesura
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.caesura.curved" }
  es8[ d] \breathe es[ f g f] |
```

```
  es2 r4 \bar "||"
}
```



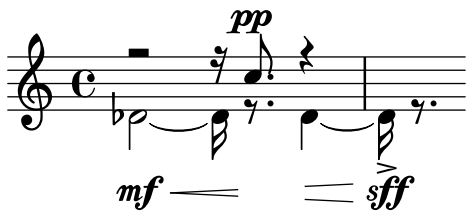
Regulador interrumpido

Para hacer invisibles partes de un regulador de crescendo, se usa el método de dibujar un rectángulo blanco encima de la parte respectiva del regulador, tapándola. El rectángulo se define como un elemento de marcado de texto.

La instrucción de marcado `with-dimensions` indica a LilyPond que considere solamente el extremo inferior del rectángulo cuando realice el espaciado de éste frente al regulador. La propiedad `staff-padding` evita que el rectángulo quepa entre el regulador y el pentagrama.

Asegúrse de que el regulador está en una capa más baja que el elemento de marcado de texto para trazar el rectángulo encima del regulador.

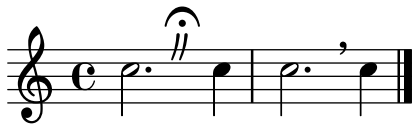
```
\relative c' {
  <<
    {
      \dynamicUp
      r2 r16 c'8.\pp r4
    }
    \\
    {
      \override DynamicLineSpanner.layer = #0
      des,2\mf\< ~
      \override TextScript.layer = #2
      \once\override TextScript.staff-padding = #6
      \once\override TextScript.vertical-skylines = #'()
      des16_\markup \with-dimensions #'(2 . 7) #'(0 . 0)
                    \with-color #white
                    \filled-box #'(2 . 7) #'(0 . 2) #0
      r8. des4 ~ des16->\sff r8.
    }
  >>
}
```



Caesura ("railtracks") with fermata

A caesura is sometimes denoted by a double "railtracks" breath mark with a fermata sign positioned above. This snippet shows an optically pleasing combination of railtracks and fermata.

```
\relative c'' {
  c2.
  % construct the symbol
  \override BreathingSign.text = \markup {
    \override #'(direction . 1)
    \override #'(baseline-skip . 1.8)
    \dir-column {
      \translate #'(0.155 . 0)
        \center-align \musicglyph "scripts.caesura.curved"
      \center-align \musicglyph "scripts.ufermata"
    }
  }
  \breathe c4
  % set the breath mark back to normal
  \revert BreathingSign.text
  c2. \breathe c4
  \bar "|."
}
```



Custos

Se pueden tipografiar «custos» en diferentes estilos.

```
\layout {
  ragged-right = ##t
}

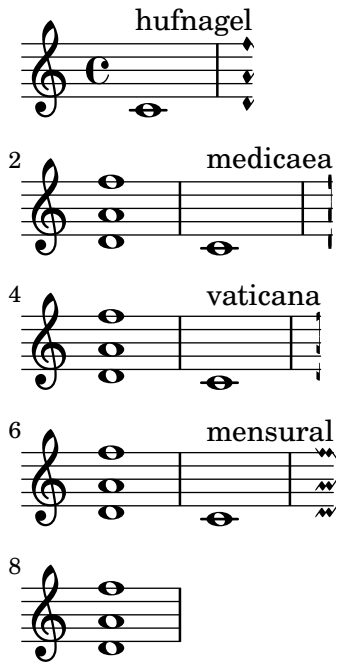
\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

    \override Staff.Custos.style = #'hufnagel
    c1^"hufnagel" \break
    <d a' f'>1

    \override Staff.Custos.style = #'medicaea
    c1^"medicaea" \break
    <d a' f'>1

    \override Staff.Custos.style = #'vaticana
    c1^"vaticana" \break
    <d a' f'>1

    \override Staff.Custos.style = #'mensural
    c1^"mensural" \break
    <d a' f'>1
  }
}
```



Personalizar la posición y el número de puntos en las líneas divisorias de repetición

Si queremos personalizar la posición y/o el número de puntos en las barras de repetición, podemos definir líneas divisorias personalizadas o redefinir la manera en que se dibujan los símbolos de repetición. Esto puede ser de especial ayuda cuando se usa un pentagrama con posiciones de línea personalizadas, como se muestra en el fragmento.

```
#(define ((make-custom-dot-bar-line dot-positions) is-span grob extent)
   "Draw dots (repeat sign dots) at DOT-POSITIONS.

The coordinates of DOT-POSITIONS are equivalent to the coordinates of
`StaffSymbol.line-positions`; a dot position of X and a line position
of X indicate the same vertical position.

IS-SPAN is not used in this custom function."
   (let* ((staff-space (ly:staff-symbol-staff-space grob))
          (dot (ly:font-get-glyph (ly:grob-default-font grob)
               "dots.dot"))
          (stencil empty-stencil))
     (for-each
      (lambda (dp)
        (set! stencil (ly:stencil-add stencil
                        (ly:stencil-translate-axis
                         dot (* dp (/ staff-space 2)) Y))))
      dot-positions)
     stencil))

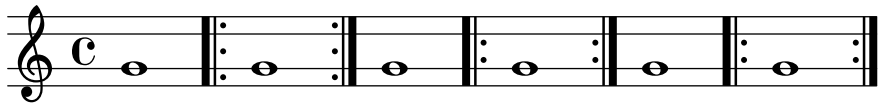
% With the procedure above we can define custom bar lines, for example,
% that resemble standard repeat sign bar lines except that there are
% three dots at staff positions -3, 0, and 3.

#(add-bar-glyph-print-procedure "*" (make-custom-dot-bar-line '(-3 0 3)))
```

```
\defineBarLine ".|*" #'("" "*" "")
\defineBarLine "*|." #'("" "*" "")

% We can also customize the dot positions used in all default repeat
% signs by redefining the print procedure of the colon bar glyph (":").
% On a staff with line positions of `(-4 -2 2 4)`, the default repeat
% sign dots appear at `(-3 3)`, but we can put them at `(-1 1)` instead.
#(add-bar-glyph-print-procedure ":" (make-custom-dot-bar-line '(-1 1)))


\new Staff \with {
  \override StaffSymbol.line-positions = #'(-4 -2 2 4)
  \override StaffSymbol.staff-space = #1.3
} \relative f' {
  g1 \bar ".|*"
  g \bar "*|."
  g \bar ".|:-|"
  g \bar ":|."
  g |
  \repeat volta 2 { g }
}
```

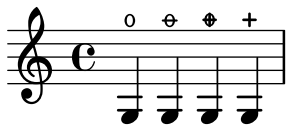


Símbolos de digitación para instrumentos de viento

Se pueden conseguir símbolos especiales combinando glifos existentes, lo que es de utilidad para la notación de instrumentos de viento.

```
lineup =
  \tweak outside-staff-padding #0
  \tweak staff-padding #0
  \tweak padding #0.2
  \tweak parent-alignment-X #CENTER
  \tweak self-alignment-X #CENTER
  \etc

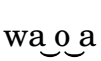
\relative c' {
  g\open
  g\lineup ^\markup \combine
              \musicglyph "scripts.open"
              \musicglyph "scripts.tenuto"
  g\lineup ^\markup \combine
              \musicglyph "scripts.open"
              \musicglyph "scripts.stopped"
  g\stopped
}
```



Cómo poner ligaduras entre sílabas de la letra

Se puede hacer separando las sílabas mediante símbolos de tilde curva.
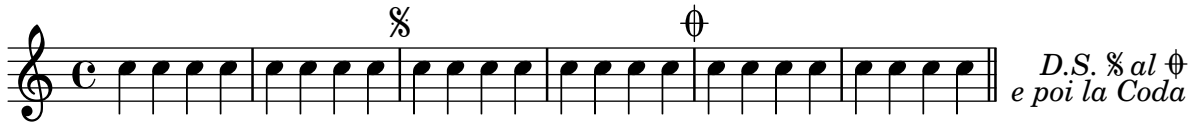
```
\lyrics {
  wa~o~a
}
```
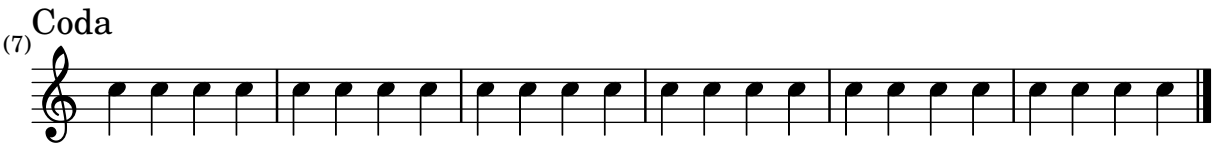


Posicionar los símbolos de repetición segno y coda (con saltos de línea)

Si queremos colocar un símbolo de segno de salida y añadir texto como “D.S. al Coda” junto a él donde normalmente están las líneas del pentagrama, cpodemos usar este código. La coda continúa en una línea nueva. Hay una variante documentada en el fragmento de código, donde la coda se mantiene en la misma línea.

```
\relative c'' {
  c4 c c c | c c c c |
  \repeat segno 2 {
    c4 c c c | c c c c |
    \alternative {
      \volta 1 {
        c4 c c c | c c c c |
        % If you don't use \break at Coda, use \noBreak here
        % and after \bar "" below.
        \noBreak
        \section % double bar line
        \cadenzaOn % pause bar count
        \stopStaff % remove staff lines
        % Increasing the unfold counter will expand the staff-free space
        \repeat unfold 4 {
          s1
          \bar ""
        }
        % Place JumpScript where the staff would normally be.
        \once \override Score.JumpScript.outside-staff-priority = ##f
        \once \override Score.JumpScript.Y-offset = 0
        \startStaff % resume bar count
        \cadenzaOff % show staff lines again
      }
    }
  }
  \sectionLabel "Coda"
  % Show Coda on a new line
  \break
  \*6 { c4 c c c }
  \fine
}
```





Estilos de silencios

Los silencios se pueden imprimir en distintos estilos.

```
restsA = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
  s64 s128 s256 s512 s1024 s1024
}
restsB = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
  r64 r128 r256 r512 r1024 s1024
}

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

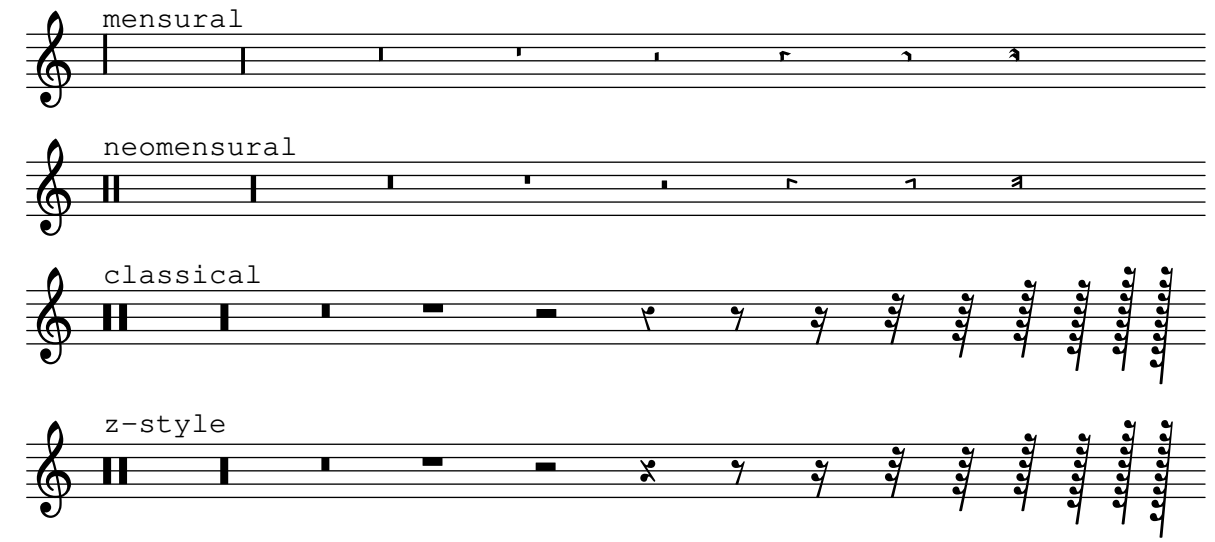
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

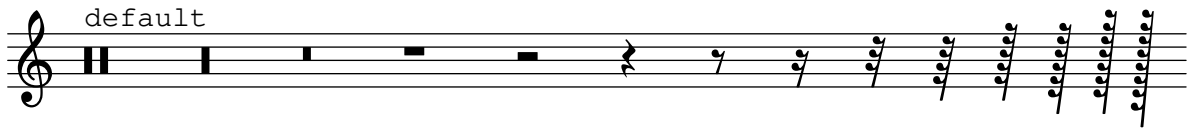
  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>^\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>^\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>^\markup \typewriter { default } \restsB \bar "" \break
}
```
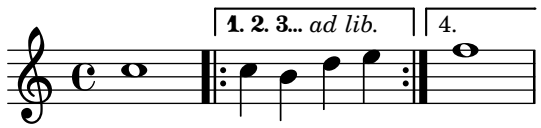




Elementos de marcado para el texto de las casillas de repetición usando `repeatCommands`

Aunque las casillas de repetición se especifican de forma óptima usando `\repeat volta`, debe usarse la propiedad de contexto `repeatCommands` en caso de que el texto de la casilla requiera un formato más avanzado con `\markup`.

Puesto que `repeatCommands` admite una lista, el método más sencillo de incluir elementos de marcado es usar un identificador para el texto e insertarlo en la lista de instrucciones usando la sintaxis de Scheme `#`((volta ,`*textIdentifier*`) ...)` (observe el uso del apóstrofo invertido después de # y la coma antes de *textIdentifier*). Las instrucciones de principio y fin de repetición se pueden añadir como elementos de lista independientes:

```
voltaAdLib = \markup { \volta-number { 1. 2. 3... } \italic { ad lib. } }

\relative c'' {
  c1
  \set Score.repeatCommands = #`((volta ,voltaAdLib) start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```



36 Templates

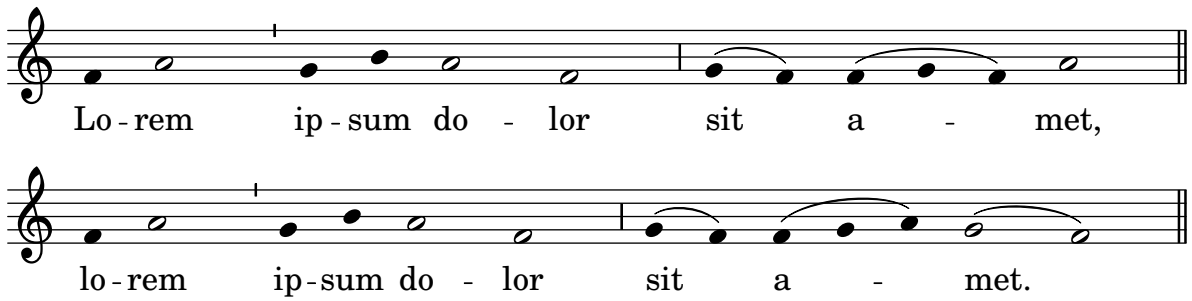
Ancient notation template – modern transcription of Gregorian music

This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter note heads, and special marks, indicating rests of different length.

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g f) a2 \finalis \break
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g a) g2( f) \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met,
  lo -- rem ip -- sum do -- lor sit a -- met.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" \chant
    \new GregorianTranscriptionLyrics = "one" \lyricsto melody \verba
  >>
}
```



Plantilla de salmo del canto anglicano

Esta plantilla muestra una forma de preparar un cántico salmódico anglicano. También muestra cómo se pueden añadir estrofas adicionales como texto independiente por debajo de la música. Las dos estrofas se codifican en estilos diferentes para ilustrar más posibilidades.

```
SopranoMusic = \relative g' {
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
```

```
  f1 | f2 e | d d | e1 |
}

TenorMusic = \relative a {
  c1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

BassMusic =  \relative c {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}

tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}

% Use markup to center the chant on the page
\markup \fill-line {
  \score { % centered
    \new ChoirStaff <<
      \new Staff <<
        \global
        \clef "treble"
        \new Voice = "Soprano" <<
          \voiceOne
          \SopranoMusic
        >>
        \new Voice = "Alto" <<
          \voiceTwo
          \AltoMusic
        >>
      >>

      \new Staff <<
        \clef "bass"
        \global
        \new Voice = "Tenor" <<
          \voiceOne
          \TenorMusic
        >>
        \new Voice = "Bass" <<
          \voiceTwo
          \BassMusic
```

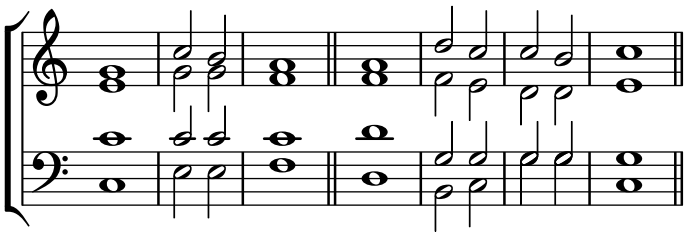
```
        >>
      >>
    >>

    \layout {
      \context {
        \Score
        \override SpacingSpanner.base-shortest-duration =
                    \musicLength 2
      }
      \context {
        \Staff
        \remove "Time_signature_engraver"
      }
    }
  }  % End score
}  % End markup

\markup \fill-line {
  \left-column {
    \null \null \null
    \line { \fontsize #5 O
            \fontsize #3 come
            let us \bold sing | unto \dot the | Lord : let }
    \line { us heartily \concat { re \bold joice }
            \italic \bold in the | strength of | our }
    \line { sal | vation. }

    \null

    \line { \hspace #2.5 8. Today if ye will hear his voice * }
    \line { \concat { \bold hard en }
            \tick not your \tick hearts : as \italic \bold in the pro- }
    \line { vocation * and as \italic \bold in the \bold day of tempt- \tick }
    \line { -ation \tick \italic \bold in the \tick wilderness. }
  }
}
```



O come let us **sing** | unto • the | Lord : let us heartily re**joice** in the | strength of | our sal | vation.

8. Today if ye will hear his voice * **hard**en ' not your ' hearts : as in the provocation * and as in the **day** of tempt- ' -ation ' in the ' wilderness.

Plantilla para himnos

Este fragmento de código muestra una forma de preparar un himno en el que cada línea comienza con un compás parcial. También muestra cómo añadir los versos como texto independiente debajo de la música.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \caesura \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \fine
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
 \key g \major
}

\score {  % Start score
  \new PianoStaff <<  % Start pianostaff
```

```
    \new Staff <<  % Start Staff = RH
      \global
      \clef "treble"
      \new Voice = "Soprano" <<  % Start Voice = "Soprano"
        \Timeline
        \voiceOne
        \SopranoMusic
      >>  % End Voice = "Soprano"
      \new Voice = "Alto" <<  % Start Voice = "Alto"
        \Timeline
        \voiceTwo
        \AltoMusic
      >>  % End Voice = "Alto"
    >>  % End Staff = RH

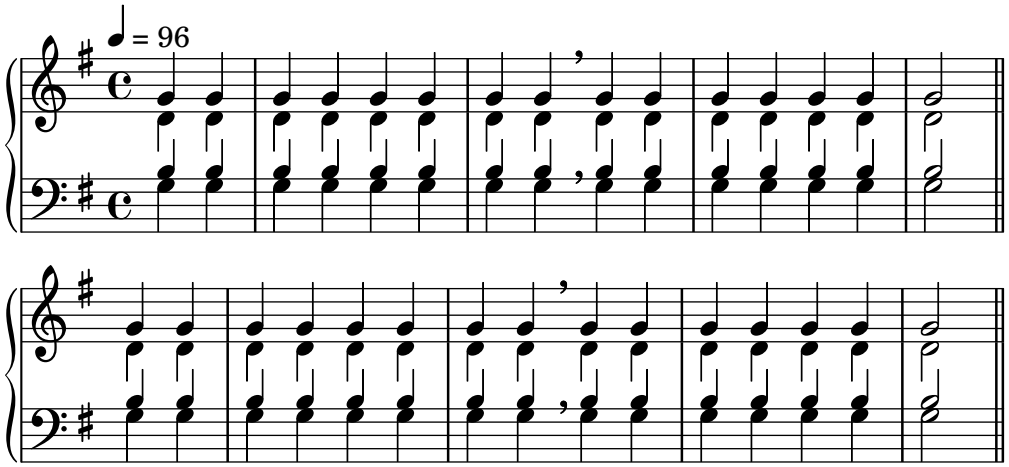
    \new Staff <<  % Start Staff = LH
      \global
      \clef "bass"
      \new Voice = "Tenor" <<  % Start Voice = "Tenor"
        \Timeline
        \voiceOne
        \TenorMusic
      >>  % End Voice = "Tenor"
      \new Voice = "Bass" <<  % Start Voice = "Bass"
        \Timeline
        \voiceTwo
        \BassMusic
      >>  % End Voice = "Bass"
    >>  % End Staff = LH
  >>  % End pianostaff
}  % End score

\markup \fill-line {
  \left-column {
    "This is line one of the first verse"
    "This is line two of the same"
    \null
    "And here's line one of the second verse"
    "And the next line of the same"
  }
}

\layout {
  \context {
    \Score
    caesuraType = #'((bar-line . "||"))
    fineBarType = "||"
  }
}

\paper {  % Start paper block
  indent = 0         % don't indent first system
```

```
  line-width = 130   % shorten line length to suit music
  tagline = ##f      % Don't print tag line, can be removed
}  % End paper block
```



This is line one of the first verse
This is line two of the same

And here's line one of the second verse
And the next line of the same

Plantilla para combo de jazz

Ésta es una plantilla bastante avanzada, para un conjunto de jazz. Observe que la notación de todos los instrumentos usa `\key c \major` (Do mayor). Esto se refiere al tono de concierto; la armadura se transporta automáticamente si la música está dentro de una sección `\transpose`.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup \column {
              "LilyPond example file by Amelie Zapf,"
              "Berlin 07/07/2003" }
}

% To make the example display properly in the documentation.
\paper {
  paper-width = 130\mm
  paper-height = 205\mm
}

% #(set-global-staff-size 16)

\include "english.ly"


%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%%%%
```

```
sl = { \override NoteHead.style = #'slash
       \hide Stem }
nsl = { \revert NoteHead.style
        \undo \hide Stem }
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

% Insert chord name style stuff here.

jazzChords = { }


%%%%%%%%%%% Keys'n'thangs %%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ############ Horns ############

% ------ Trumpet ------
trpt = \transpose c d \relative c'' {
  \Key
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  \trpt
}

% ------ Alto Saxophone ------
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble
  \alto
}

% ------ Baritone Saxophone ------
bari = \transpose c a' \relative c {
  \Key
```

```
  c1 | c1 |
  \sl d4^"Solo" d d d \nsl |
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords
  s1 | s |
  d2:maj e:m7 |
}
bariSax = {
  \global
  \clef treble
  \bari
}

% ------ Trombone ------
tbone = \relative c {
  \Key
  c1 | c | c |
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
  \clef bass
  \tbone
}

% ############ Rhythm Section #############

% ------ Guitar ------
gtr = \relative c'' {
  \Key
  c1 |
  \sl b4 b b b \nsl |
  c1 |
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 | c2:min7+ d2:maj9 | s1 |
}
guitar = {
  \global
  \clef treble
  \gtr
}

%% ------ Piano ------
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c |
```

```
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e |
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g |
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c |
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
  >>
}
PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = <<
  \new Staff = "upper" \PianoRH
  \new Staff = "lower" \PianoLH
>>


% ------ Bass Guitar ------
Bass = \relative c {
  \Key
  c1 | c | c |
}
bass = {
  \global
  \clef bass
  \Bass
}
```

```
% ------ Drums ------
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
}
down = \drummode {
  \voiceTwo
  bd4 s bd s |
  bd4 s bd s |
  bd4 s bd s |
}

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

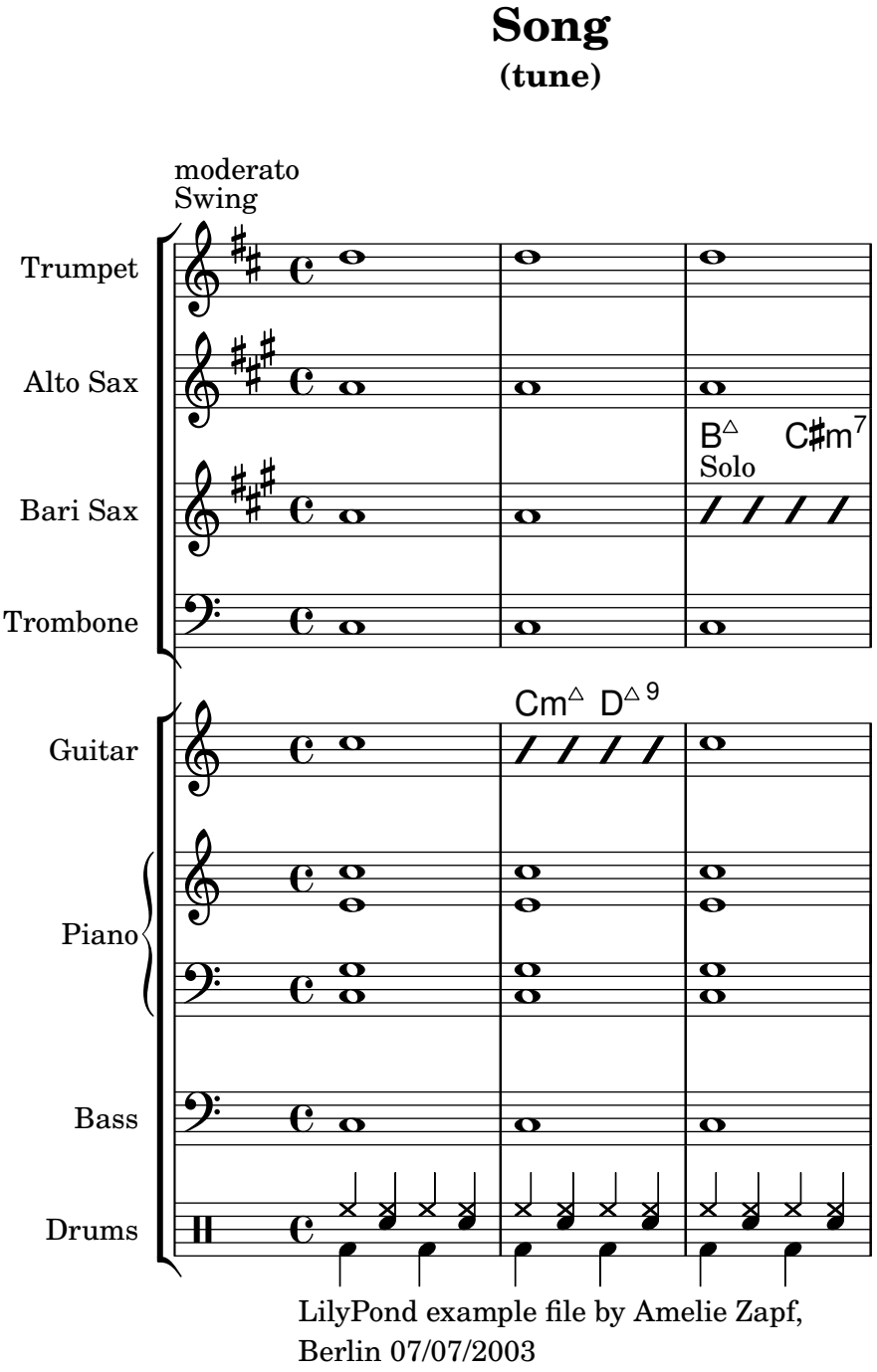
%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%%

\book { % For the LilyPond documentation.
  \score {
    <<
      \new StaffGroup = "horns" <<
        \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
          \trumpet
        \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
          \altoSax
        \new ChordNames = "barichords" \with { instrumentName = "Bari Sax" }
          \bariHarmony
        \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
          \bariSax
        \new Staff = "trombone" \with { instrumentName = "Trombone" }
          \trombone
      >>

      \new StaffGroup = "rhythm" <<
        \new ChordNames = "chords" \with { instrumentName = "Guitar" }
          \gtrHarmony
        \new Staff = "guitar" \with { instrumentName = "Guitar" }
          \guitar
        \new PianoStaff = "piano" \with {
          instrumentName = "Piano"
          midiInstrument = "acoustic grand"
        } \piano
        \new Staff = "bass" \with { instrumentName = "Bass" }
          \bass
        \new DrumStaff \with { instrumentName = "Drums" }
```

```
          \drumContents
      >>
    >>

    \layout {
      \context {
        \Staff
        \RemoveEmptyStaves
      }
      \context {
        \Score
        \override BarNumber.padding = 3
        \override RehearsalMark.padding = 2
        skipBars = ##t
      }
    }
    \midi { }
  }
}
```



Plantilla de orquesta, coro y piano

Esta plantilla muestra el uso de contextos `StaffGroup` y `GrandStaff` anidados para sub-agrupar instrumentos del mismo tipo, y una forma de usar `\transpose` de manera que unas variables contengan la música para instrumentos transpositores en afinación de concierto.

```
#(set-global-staff-size 17)

\paper {
  indent = 3.0\cm  % add space for instrumentName
  short-indent = 1.5\cm  % add less space for shortInstrumentName
}
```

```
fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.
clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns
hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }
sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }
altoILyrics = \sopranoLyrics
altoIIMusic = \relative c' { \key g \major g'1 b }
altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }
tenorLyrics = \sopranoLyrics

pianoRHMusic = \relative c { \key g \major g''1 b }
pianoLHMusic = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }
violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\book {
  \score {
    <<
      \new StaffGroup = "StaffGroup_woodwinds" <<
        \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
          \fluteMusic

        \new Staff = "Staff_clarinet" \with {
          instrumentName = \markup { \concat { "Clarinet in B" \flat } }
        }
          % Declare that written Middle C in the music
          % to follow sounds a concert B flat, for
          % output using sounded pitches such as MIDI.
          %\transposition bes
```

```
    % Print music for a B-flat clarinet
    \transpose bes c' \clarinetMusic
>>

\new StaffGroup = "StaffGroup_brass" <<
  \new Staff = "Staff_hornI" \with {
    instrumentName = "Horn in F"
  }
    % \transposition f
    \transpose f c' \hornMusic

  \new Staff = "Staff_trumpet" \with {
    instrumentName = "Trumpet in  C"
  }
    \trumpetMusic
>>

\new RhythmicStaff = "RhythmicStaff_percussion" \with {
  instrumentName = "Percussion"
}
  \percussionMusic

\new PianoStaff \with {
  instrumentName = "Piano"
} <<
  \new Staff { \pianoRHMusic }
  \new Staff { \pianoLHMusic }
>>

\new ChoirStaff = "ChoirStaff_choir" <<
  \new Staff = "Staff_soprano" \with {
    instrumentName = "Soprano"
  }
    \new Voice = "soprano" \sopranoMusic
    \new Lyrics \lyricsto "soprano" { \sopranoLyrics }

  \new GrandStaff = "GrandStaff_altos" \with {
    \accepts Lyrics
  } <<
    \new Staff = "Staff_altoI" \with {
      instrumentName = "Alto I"
    }
      \new Voice = "altoI"
      \altoIMusic
      \new Lyrics \lyricsto "altoI" { \altoILyrics }
    \new Staff = "Staff_altoII" \with {
      instrumentName = "Alto II"
    }
      \new Voice = "altoII"
      \altoIIMusic
      \new Lyrics \lyricsto "altoII" { \altoIILyrics }
```

```
        >>

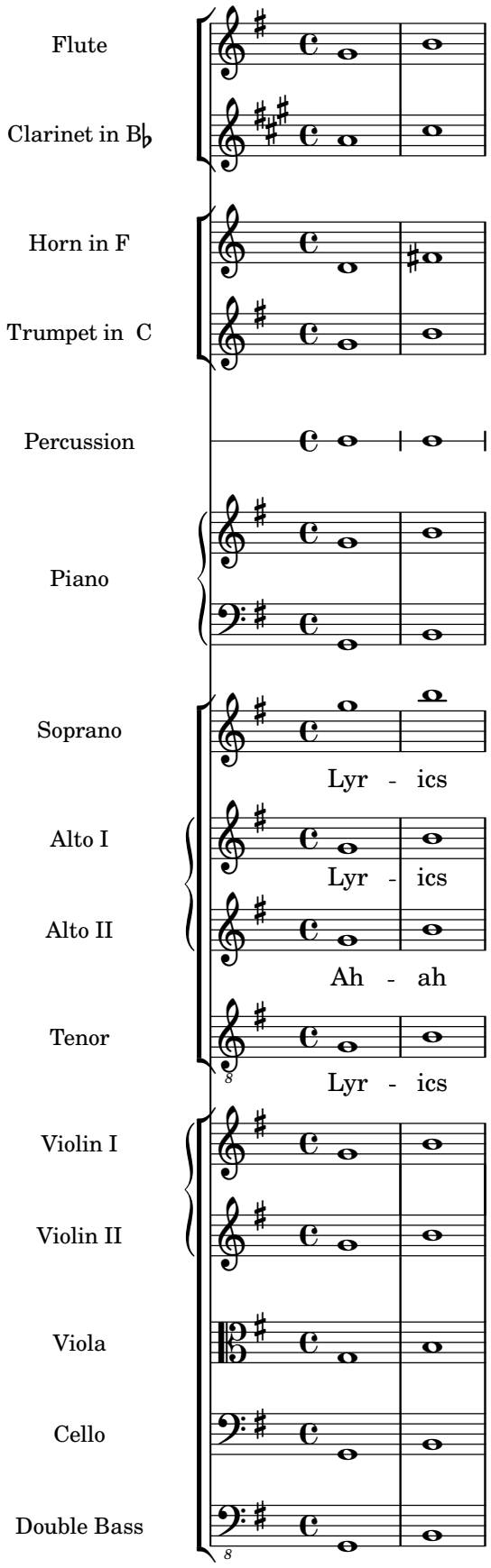
        \new Staff = "Staff_tenor" \with {
          instrumentName = "Tenor"
        }
          \new Voice = "tenor" \tenorMusic
          \new Lyrics \lyricsto "tenor" { \tenorLyrics }
      >>

      \new StaffGroup = "StaffGroup_strings" <<
        \new GrandStaff = "GrandStaff_violins" <<
          \new Staff = "Staff_violinI" \with {
            instrumentName = "Violin I"
          }
            \violinIMusic
          \new Staff = "Staff_violinII" \with {
            instrumentName = "Violin II"
          }
            \violinIIMusic
        >>

        \new Staff = "Staff_viola" \with {
          instrumentName = "Viola"
        }
          \violaMusic

        \new Staff = "Staff_cello" \with {
          instrumentName = "Cello"
        }
          \celloMusic

        \new Staff = "Staff_bass" \with {
          instrumentName = "Double Bass"
        }
          \bassMusic
      >>
    >>
  }
}
```


Flute
Clarinet in B♭
Horn in F
Trumpet in C
Percussion
Piano
Soprano
Lyr - ics
Alto I
Lyr - ics
Alto II
Ah - ah
Tenor
Lyr - ics
Violin I
Violin II
Viola
Cello
Double Bass

Plantilla de piano (sencilla)

Presentamos a continuación una plantilla de piano sencilla con algunas notas.

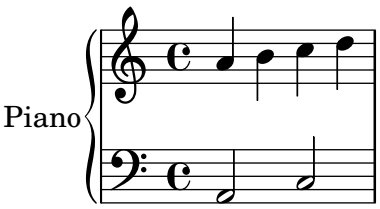
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  \new PianoStaff \with { instrumentName = "Piano" }
  <<
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}
```



Plantilla de piano con letra centrada

En lugar de tener un pentagrama dedicado a la melodía y la letra, ésta se puede centrar entre los pentagramas de un sistema de piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

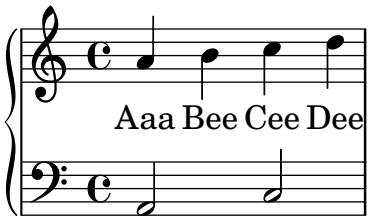
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4
```

```
  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}
```



Plantilla de piano con melodía y letra

He aquí el típico formato dde una canción: un pentagrama con la melodía y la letra, y el acompañamiento de piano por debajo.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

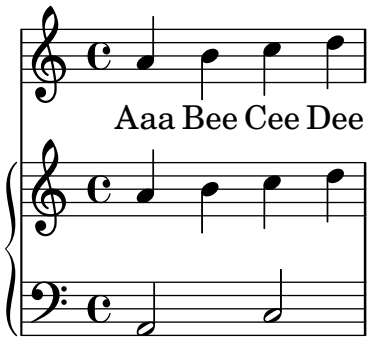
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4
```

```
  a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```
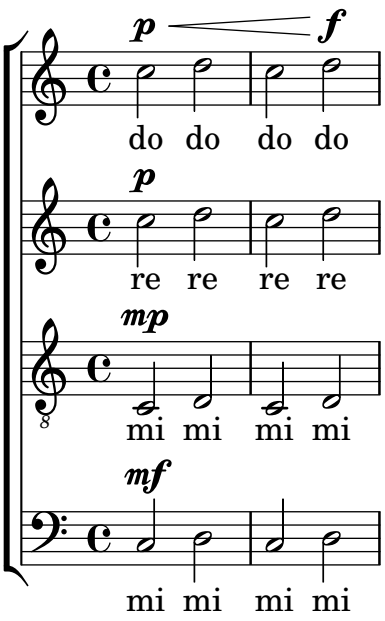


Plantilla de coro SATB, a cuatro pentagramas

He aquí una plantilla de coro SATB en cuatro pentagramas.

```
global = {
  \key c \major
  \time 4/4
  \dynamicUp
}
sopranonotes = \relative c'' {
  c2 \p \< d c d \f
}
sopranowords = \lyricmode { do do do do }
altonotes = \relative c'' {
  c2\p d c d
}
altowords = \lyricmode { re re re re }
tenornotes = {
  \clef "G_8"
  c2\mp d c d
}
tenorwords = \lyricmode { mi mi mi mi }
bassnotes = {
  \clef bass
```

```
  c2\mf d c d
}
basswords = \lyricmode { mi mi mi mi }

\score {
  \new ChoirStaff <<
    \new Staff <<
      \new Voice = "soprano" <<
        \global
        \sopranonotes
      >>
      \new Lyrics \lyricsto "soprano" \sopranowords
    >>
    \new Staff <<
      \new Voice = "alto" <<
        \global
        \altonotes
      >>
      \new Lyrics \lyricsto "alto" \altowords
    >>
    \new Staff <<
      \new Voice = "tenor" <<
        \global
        \tenornotes
      >>
      \new Lyrics \lyricsto "tenor" \tenorwords
    >>
    \new Staff <<
      \new Voice = "bass" <<
        \global
        \bassnotes
      >>
      \new Lyrics \lyricsto "bass" \basswords
    >>
  >>
}
```



Plantilla de pentagrama único con música, letra y acordes

Esta plantilla facilita la preparación de una canción con melodía, letra y acordes.

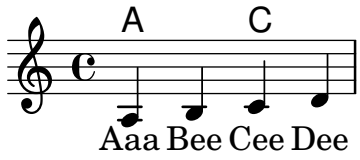
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Single-staff template with notes, lyrics, chords, and frets

Here is a simple lead sheet template with melody, lyrics, chords, and fret diagrams.

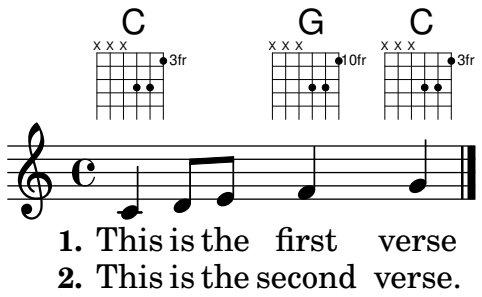
```
verseI = \lyricmode {
  \stanza "1."
  This is the first verse
}

verseII = \lyricmode {
  \stanza "2."
  This is the second verse.
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
  c2 g4 c
}

staffMelody = \relative c' {
   \key c \major
   \clef treble
   % Type notes for melody here
   c4 d8 e f4 g
   \bar "|."
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}
```



Plantilla de pentagrama único con música y acordes
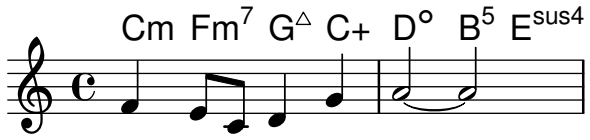
¿Quiere preparar una hoja guía de acordes (o «lead sheet») con melodía y acordes? ¡No busque más!

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  f4 e8[ c] d4 g |
  a2 ~ a
}

harmonies = \chordmode {
  c4:m f:min7 g:maj c:aug |
  d2:dim b4:5 e:sus
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Staff \melody
  >>
  \layout{ }
  \midi { }
}
```



Plantilla de pentagrama único don notas y letra

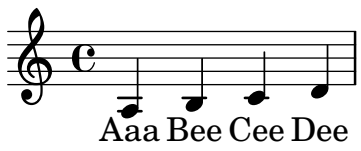
Esta pequeña plantilla muestra una melodía sencilla con letra. Córtela y péguela, escriba las notas y luego la letra. Este ejemplo desactiva el barrado automático, que es lo más frecuente en las partes vocales antiguas. Para usar el barrado automático modifique o marque como un comentario la línea correspondiente.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score{
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```
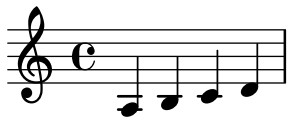


Plantilla de un solo pentagrama, con notas únicamente

Esta plantilla simple prepara un pentagrama con notas, adecuado para un instrumento solista o un fragmento melódico. Córtelo y péguelo en un archivo, escriba las notas y ¡ya está!

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

\score {
  \new Staff \melody
  \layout { }
  \midi { }
}
```



Plantilla de cuarteto de cuerda (sencilla)

Esta plantilla muestra un cuarteto de cuerda normal. También utiliza una sección `\global` para el compás y la armadura.

Véase también el fragmento de código "Plantilla de cuarteto de cuerda con particellas independientes".

```
global= {
  \time 4/4
  \key c \major
}

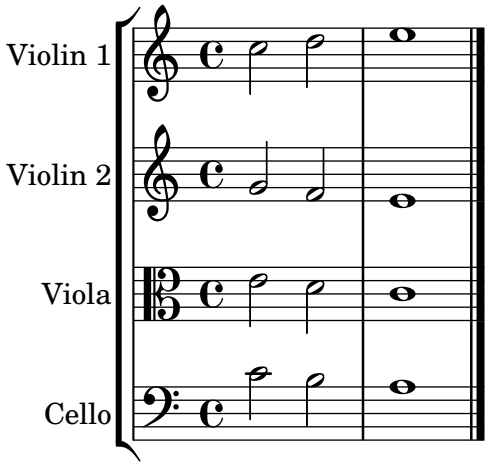
violinOne = \new Voice \relative c'' {
  c2 d
  e1
  \bar "|."
}

violinTwo = \new Voice \relative c'' {
  g2 f
  e1
  \bar "|."
}

viola = \new Voice \relative c' {
  \clef alto
  e2 d
  c1
  \bar "|."
}

cello = \new Voice \relative c' {
  \clef bass
  c2 b
  a1
  \bar "|."
}

\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Violin 1" }
      << \global \violinOne >>
    \new Staff \with { instrumentName = "Violin 2" }
      << \global \violinTwo >>
    \new Staff \with { instrumentName = "Viola" }
      << \global \viola >>
    \new Staff \with { instrumentName = "Cello" }
      << \global \cello >>
  >>
  \layout { }
  \midi { }
}
```



Plantilla de cuarteto de cuerda con particellas independientes

El fragmento de código "Plantilla de cuarteto de cuerda" produce un resultado satisfactorio para el cuarteto, pero ¿y si tenemos que imprimir las particellas? Esta nueva plantilla muestra cómo usar la funcionalidad `\tag` (etiqueta) para dividir fácilmente una pieza en particellas individuales.

Tenemos que dividir esta plantilla en archivos independientes; los nombres de archivo están dentro de los comentarios al principio de cada archivo. `piece.ly` contiene todas las definiciones de música. Los otros archivos (`score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` y `vlc.ly`) producen la particella correspondiente.

¡No olvide quitar los comentarios que hemos especificado cuando use los archivos independientes!

```
% piece.ly
% (This is the global definitions file.)

global= {
  \time 4/4
  \key c \major
}

Violinone = \new Voice \relative c'' {
  c2 d e1
  \bar "|."
}

Violintwo = \new Voice \relative c'' {
  g2 g e1
  \bar "|."
}

Viola = \new Voice \relative c' {
  \clef alto
  e2 d c1
  \bar "|."
}

Cello = \new Voice \relative c' {
```

```
  \clef bass
  c2 b a1
  \bar "|."
}

music = <<
  \tag #'score \tag #'vn1
  \new Staff \with { instrumentName = "Violin 1" }
    << \global \Violinone >>

  \tag #'score \tag #'vn2
  \new Staff \with { instrumentName = "Violin 2" }
    << \global \Violintwo >>

  \tag #'score \tag #'vla
  \new Staff \with { instrumentName = "Viola" }
    << \global \Viola >>

  \tag #'score \tag #'vlc
  \new Staff \with { instrumentName = "Cello" }
    << \global \Cello >>
>>


% These are the other files you need to save on your computer

% score.ly
% (This is the main file.)

% Uncomment the line below when using a separate file.
% \include "piece.ly"

#(set-global-staff-size 14)

\score {
  \new StaffGroup \keepWithTag #'score \music
  \layout { }
  \midi { }
}


%{ Uncomment this block when using separate files.

% vn1.ly
% (This is the Violin 1 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn1 \music
  \layout { }
}
```

```
% vn2.ly
% (This is the Violin 2 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn2 \music
  \layout { }
}

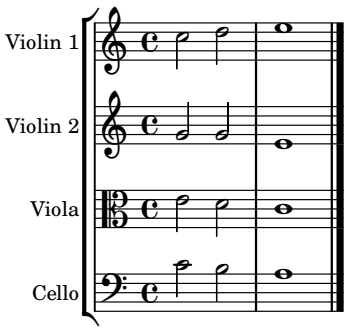
% vla.ly
% (This is the Viola part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}

% vlc.ly
% (This is the Cello part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vlc \music
  \layout { }
}

%}
```



Plantilla de conjunto vocal

He aquí una partitura vocal estándar para cuatro voces SATB. Con grupos mayores, suele ser útil incluir una sección que aparezca en todas las partes. Por ejemplo, el compás y la armadura casi siempre son los mismos para todas. Como en la "Plantilla para himnos", las cuatro voces se reagrupan en solo dos pentagramas.

```
\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}
```

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
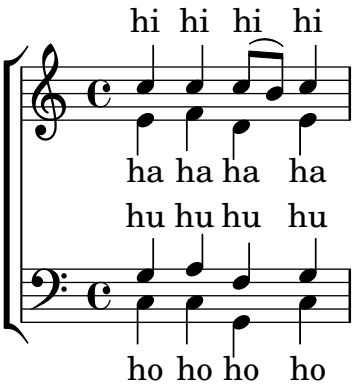
altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Lyrics = "sopranos" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "women" <<
      \new Voice = "sopranos" {
        \voiceOne
        << \global \sopMusic >>
      }
      \new Voice = "altos" {
        \voiceTwo
        << \global \altoMusic >>
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors" \with {
      % this is needed for lyrics above a staff
```

```
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        << \global \tenorMusic >>
      }
      \new Voice = "basses" {
        \voiceTwo << \global \bassMusic >>
      }
    >>
    \new Lyrics = "basses"
    \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
    \context Lyrics = "altos" \lyricsto "altos" \altoWords
    \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
    \context Lyrics = "basses" \lyricsto "basses" \bassWords
  >>
}
```



Plantilla de conjunto vocal con reducción de piano automática

Esta plantilla añade una reducción de piano automática a la partitura vocal SATB estándar que se mostró en el fragmento de código "Plantilla de conjunto vocal". Presenta uno de los puntos fuertes de LilyPond: podemos usar una definición de música más de una vez. Si se hace cualquier cambio en las notas de la parte vocal (digamos `tenorMusic`), entonces los cambios se aplican también a la reducción de piano.

```
\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
```

```
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords =\lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

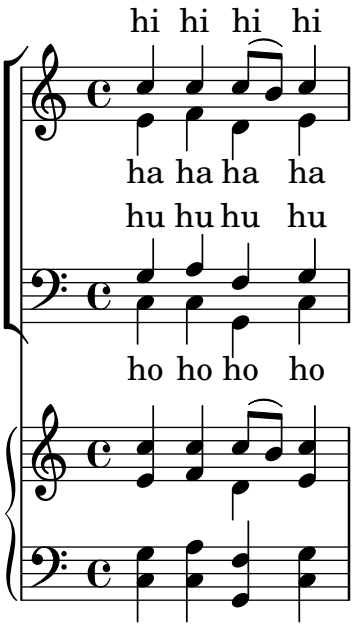
\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"

      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"

      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
```

```
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>

    \new PianoStaff <<
      \new Staff <<
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \sopMusic >>
        << \global \altoMusic >>
      >>
      \new Staff <<
        \clef bass
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \tenorMusic >>
        << \global \bassMusic >>
      >>
    >>
  >>
}
```



Plantilla para conjunto vocal con letras alineadas encima y debajo de los pentagramas

Esta plantilla es, básicamente, la misma que la sencilla "Plantilla para conjunto vocal", excepto que aquí todas las líneas de letra se colocan utilizando `alignAboveContext` y `alignBelowContext`.

```
global = {
  \key c \major
  \time 4/4
}
```

```
sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}
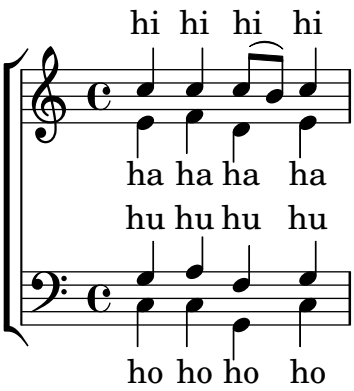
tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative c {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = "women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords

    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = "men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
```

```
    % \new Lyrics \lyricsto "basses" \bassWords
  >>
}
```



Estrofa para solista y estribillo a dos voces

Esta plantilla crea una partitura que comienza con una estrofa para solista y continúa con un estribillo a dos voces. también muestra el uso de silencios de separación dentro de la variable \global para definir cambios de compás (y otros elementos que son comunes a todas las partes) a lo largo de toda la partitura.

```
global = {
  \key g \major

  % verse
  \time 3/4
  s2.*2
  \break

  % refrain
  \time 2/4
  s2*2
  \bar "|."
}

SoloNotes = \relative g' {
  \clef "treble"

  % verse
  g4 g g |
  b4 b b |

  % refrain
  R2*2 |
}

SoloLyrics = \lyricmode {
  One two three |
  four five six |
}

SopranoNotes = \relative c'' {
```

```
  \clef "treble"

  % verse
  R2.*2 |

  % refrain
  c4 c |
  g4 g |
}

SopranoLyrics = \lyricmode {
  la la |
  la la |
}

BassNotes = \relative c {
  \clef "bass"

  % verse
  R2.*2 |

  % refrain
  c4 e |
  d4 d |
}

BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

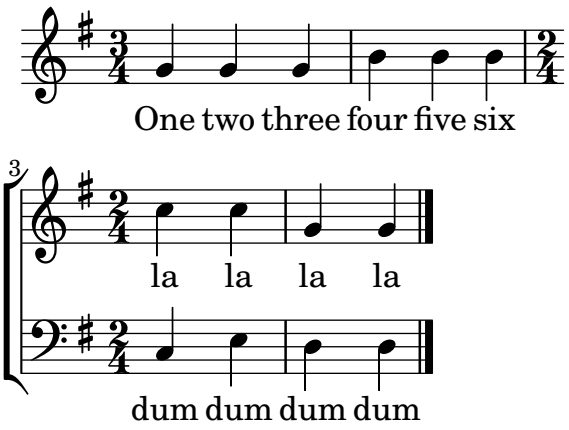
\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

      \new Voice = "BassVoice" << \global \BassNotes >>
      \new Lyrics \lyricsto "BassVoice" \BassLyrics
    >>
  >>

  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
```

```
  }
}
```


One two three four five six
3
la la la la
dum dum dum dum

37 Titles

See also Sección “Titles and headers” en *Notation Reference*.

Añadir la fecha actual a una partitura

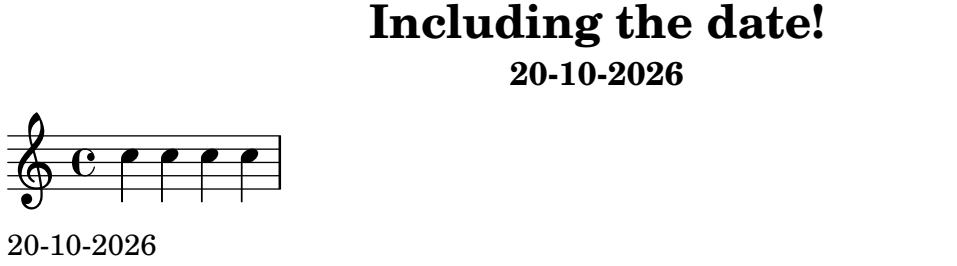
Con algo de código de Scheme, se puede añadir fácilmente la fecha actual a una partitura.

```
\paper { tagline = ##f }

% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))

% use it in the title block:
\header {
  title = "Including the date!"
  subtitle = \date
}

\score {
  \relative c'' {
    c4 c c c
  }
}
% and use it in a \markup block:
\markup {
  \date
}
```



Alinear y centrar los nombres de instrumento

La alineación horizontal de los nombres de instrumento se puede trucar modificando la propiedad `self-alignment-X` del objeto gráfico `InstrumentName` (normalmente dentro del contexto `Staff`). Las variables de `\layout` `indent` y `short-indent` definen el espacio en que se alinean los nombres de instrumento antes del primer sistema y de los siguientes, respectivamente.

```
\paper {
  left-margin = 3\cm
}

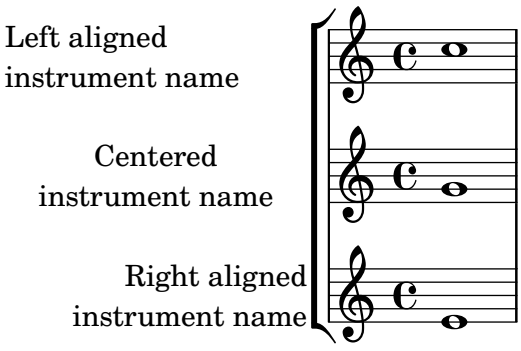
\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                            "instrument name" }
    shortInstrumentName = "Left"
```

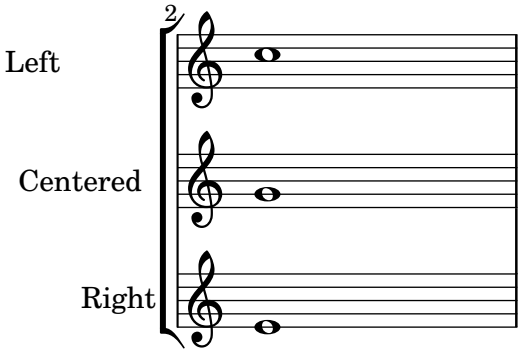
```
  } {
    c''1 \break c''1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #CENTER
    instrumentName = \markup \center-column { Centered
                                              "instrument name" }
    shortInstrumentName = "Centered"
  } {
    g'1 g'1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #RIGHT
    instrumentName = \markup \right-column { "Right aligned"
                                             "instrument name" }
    shortInstrumentName = "Right"
  } {
    e'1 e'1
  }
>>

\layout {
  indent = 4\cm
  short-indent = 2\cm
  line-width = 6.5\cm
}
```





Demonstrating all \header fields

A demonstration of all header fields that LilyPond defines by default. Thanks to setting `print-all-headers` to `#t`, much more fields as usual are displayed, indicating the hierarchy of `\header` blocks.

```
\paper {
  #(set-paper-size "a6" 'landscape)
  print-all-headers = ##t
}

\book {
  \header {
    title = "title"
    subtitle = "subtitle"
    composer = "composer"
    arranger = "arranger"
    instrument = "instrument"
    meter = "meter"
    opus = "opus"
    piece = "piece"
    poet = "poet"
    copyright = "copyright"
    tagline = "tagline"
  }

  \bookpart {
    \score {
      \relative c'' { c1 | c | c | c }

      \header {
        title = "localtitle"
        subtitle = "localsubtitle"
        composer = "localcomposer"
        arranger = "localarranger"
        instrument = "localinstrument"
        meter = "localmeter"
        opus = "localopus"
        piece = "localpiece"
        poet = "localpoet"
        copyright = "localcopyright"
        tagline = "localtagline"
      }
    }
  }
}
```

title

subtitle

poet **instrument** composer

meter arranger
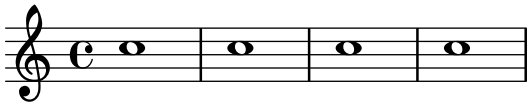
localtitle

localsubtitle

localpoet **localinstrument** localcomposer

localmeter localarranger

localpiece localopus



copyright

tagline

Imprimir el número de versión

Es posible imprimir el número de versión de LilyPond dentro de un elemento de marcado.

```
\markup { Processed with LilyPond version #(lilypond-version) }
```

Processed with LilyPond version 2.27.4

38 Tweaks and overrides

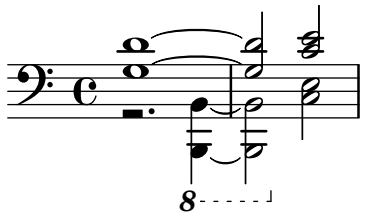
See also Sección “Changing defaults” en *Notation Reference* and Sección “Tweaking output” en *Learning Manual*.

Añadir una indicación de octava alta a una sola voz

Si tiene más de una voz en el mismo pentagrama, el cambio de octavación de una voz transportará la posición de las notas en todas las voces mientras dure el corchete de octava. Si la octavación se quiere aplicar a una voz solamente, hay que trasladar el grabador `Ottava_spanner_engraver` al contexto `Voice`.

```
\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}

{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
    {
      r2.
      \ottava -1
      <b,,, b,,>4 ~ |
      q2
      \ottava 0
      <c e>2
    }
  >>
}
```



Añadir enlaces a los objetos

Para añadir un enlace al sello de un objeto gráfico, podemos usar `add-link` tal y como se define aquí. Funciona con `\override` y con `\tweak`.

Inconveniente: `point-and-click` (apuntar y pulsar) quedará obstacuilzado por los objetos gráficos enlazados.

Limitación: funciona solamente para PDF.

Los objetos enlazados se colorean con una instrucción aparte.

```
#(define (add-link url-strg)
  (lambda (grob)
    (let* ((stil (ly:grob-property grob 'stencil)))
      (if (ly:stencil? stil)
          (let* ((x-ext (ly:stencil-extent stil X))
                 (y-ext (ly:stencil-extent stil Y))
                 (url-expr `(url-link ,url-strg ,x-ext ,y-ext))
                 (new-stil
                   (ly:stencil-add
                     (ly:make-stencil url-expr x-ext y-ext)
                     stil)))
          (ly:grob-set-property! grob 'stencil new-stil))))))

%%%% test

%% For easier maintenance of this snippet the URL is formatted to use the
%% actually used LilyPond version.
%% Of course a literal URL would work as well.

#(define major.minor-version
  (string-join (take (string-split (lilypond-version) #\.) 2) "."))

urlI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

urlII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/rhythms"
  major.minor-version)

urlIII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-heads"
  major.minor-version)

urlIV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/beams"
  major.minor-version)

urlV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-head-styles"
  major.minor-version)

urlVI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)
```
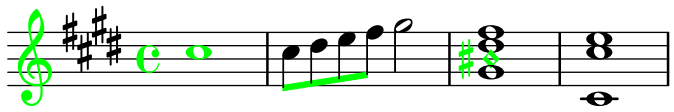
```
\relative c' {
   \key cis \minor

   \once \override Staff.Clef.color = #green
   \once \override Staff.Clef.after-line-breaking =
     #(add-link urlI)

   \once \override Staff.TimeSignature.color = #green
   \once \override Staff.TimeSignature.after-line-breaking =
     #(add-link urlII)

   \once \override NoteHead.color = #green
   \once \override NoteHead.after-line-breaking =
     #(add-link urlIII)

   cis'1
   \once \override Beam.color = #green
   \once \override Beam.after-line-breaking =
     #(add-link urlIV)
   cis8 dis e fis gis2
   <gis,
    \tweak Accidental.color #green
    \tweak Accidental.after-line-breaking #(add-link urlVI)
    \tweak color #green
    \tweak after-line-breaking #(add-link urlV)
    \tweak style #'harmonic
    bis
    dis
    fis
   >1
   <cis, cis' e>
}
```



Insertar elementos de marcado en una tablatura

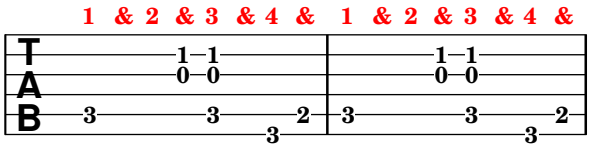
De forma predeterminada, los elementos de marcado no aparecen en la tablatura.

Para hacer que aparezcan, revierta la propiedad `stencil` del objeto gráfico `TextScript` dentro del contexto `TabStaff`.

```
high  = { r4 r8 <g c'> q r8 r4 }
low = { c4 r4 c8 r8 g,8 b, }
pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }

\score {
 \new TabStaff {
   \*2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
```

```
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```



Añadir marcas de tiempo a glissandos largos
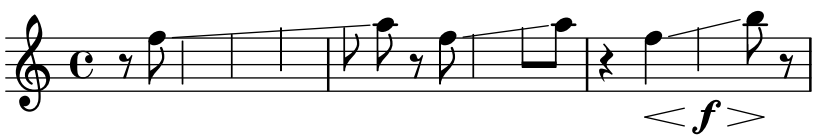
Los pulsos que se saltan en glissandos muy largos se indican a veces mediante marcas de tiempo que consisten en figuras sin cabeza. Estas plicas se pueden usar también para albergar indicaciones expresivas intermedias.

Si las plicas no quedan bien alineadas con el glissando, podría ser necesario recolocarlas ligeramente.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c'' {
  r8 f8\glissando \glissandoSkipOn f4 g a |
  a8\noBeam \glissandoSkipOff a8
    r8 f8\glissando \glissandoSkipOn g4 a8 \glissandoSkipOff a8 |
  r4 f\glissando\< \glissandoSkipOn a4\f\> \glissandoSkipOff b8\! r |
}
```
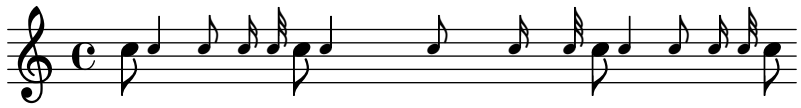


Ajuste del espaciado de las notas de adorno

Se puede ajustar la separación entre las notas de adorno utilizando la propiedad `spacing-increment` de `Score.GraceSpacing`.

```
graceNotes = {
  \grace { c4 c8 c16 c32 }
  c8
}

\relative c'' {
  c8
  \graceNotes
  \override Score.GraceSpacing.spacing-increment = #2.0
  \graceNotes
  \revert Score.GraceSpacing.spacing-increment
  \graceNotes
}
```



Ajustar las posiciones de las ligaduras verticalmente

Usando `\override Slur.positions` es posible fijar la posición vertical de los puntos de inicio y de fin de una ligadura de expresión a valores absolutos (o mejor dicho, forzar al algoritmo de ligaduras de LilyPond para que tenga en cuenta estos valores según se desee). En muchos casos, esto implica bastante ensayo y error hasta que se encuentran unos valores aceptables. Porblablemente habrá probado la instrucción `\offset` solo para comprobar que no funciona con las ligaduras de expresión, sino que emite una advertencia en su lugar.

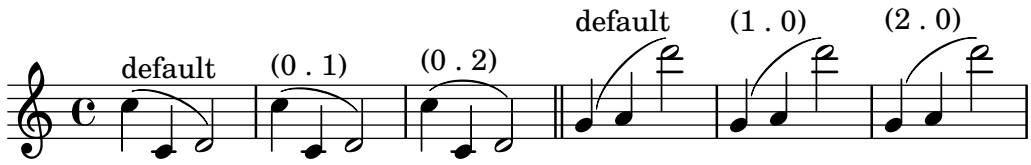
El código de este fragmento nos permite trucar las posiciones de comienzo y final especificando cambios *relativos*, de manera parecida a `\offset`.

La sintaxis es: `\offsetPositions #'(dy1 . dy2)`

```
offsetPositions =
#(define-music-function (offsets) (number-pair?)
  #{
     \once \override Slur.control-points =
       #(lambda (grob)
          (match-let ((((_ . y1) _ _ (_ . y2))
                       (ly:slur::calc-control-points grob))
                      ((off1 . off2) offsets))
            (set! (ly:grob-property grob 'positions)
                  (cons (+ y1 off1) (+ y2 off2)))
            (ly:slur::calc-control-points grob)))
  #})

\relative c'' {
  c4(^"default" c, d2)
  \offsetPositions #'(0 . 1)
  c'4(^"(0 . 1)" c, d2)
  \offsetPositions #'(0 . 2)
  c'4(^"(0 . 2)" c, d2)
  \bar "||"
  g4(^"default" a d'2)
  \offsetPositions #'(1 . 0)
  g,,4(^"(1 . 0)" a d'2)
```

```
  \offsetPositions #'(2 . 0)
  g,,4(^"(2 . 0)" a d'2)
}
```
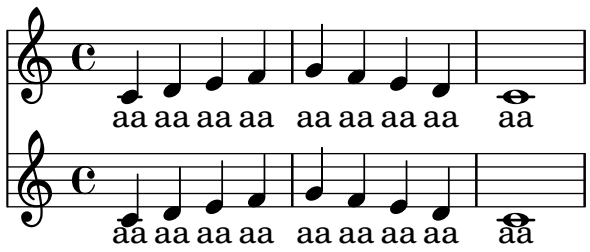


Adjusting vertical spacing of lyrics

This snippet shows how to bring the lyrics line closer to the staff.

```
music = \relative c' { c4 d e f | g4 f e d | c1 }
text = \lyricmode { aa aa aa aa aa aa aa aa aa }

<<
  \new Staff \new Voice = melody \music
  % Default layout:
  \new Lyrics \lyricsto melody \text

  \new Staff \new Voice = melody \music
  % Reducing the minimum space below the staff and above the lyrics.
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  } \lyricsto melody \text
>>
```



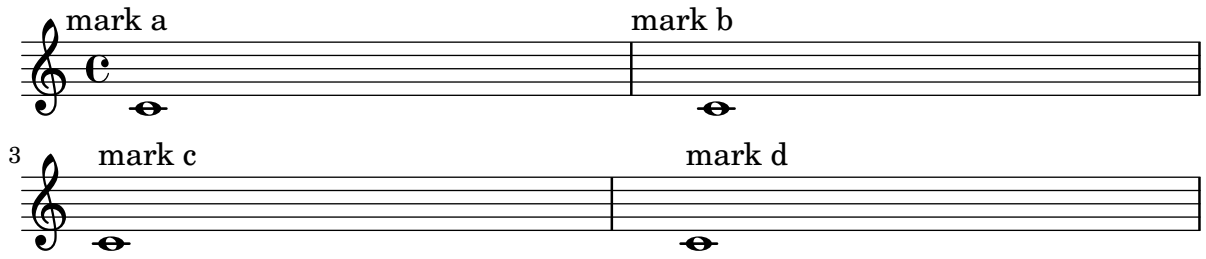
Aligning text marks to notes

By default, `TextMark` objects are aligned to so-called `NonMusicalPaperColumn` grobs, like the left edge of the staff or a bar line. They can be aligned to a note instead by setting the `non-musical` property to `#f`.

```
\layout {
  line-length = 80\mm
}

{
  \textMark "mark a" c'1 |
  \textMark "mark b" c'1 |
  \break
  \override Score.TextMark.non-musical = ##f
  \textMark "mark c" c'1 |
  \textMark "mark d" c'1 |
```

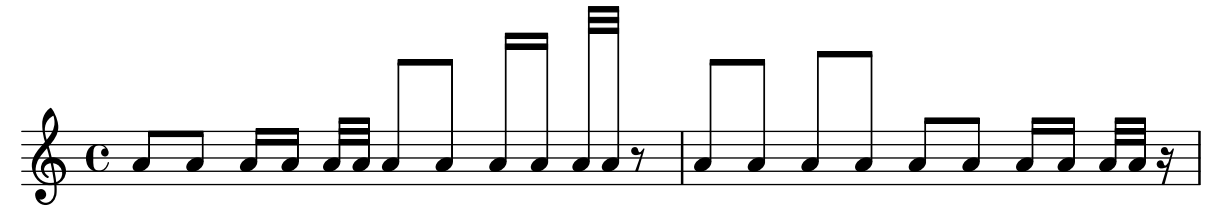
```
}
```



Alterar la longitud de las plicas unidas por una barra

Se puede variar la longitud de las plicas de las figuras unidas por una barra mediante la sobreescritura de la propiedad `beamed-lengths` de los detalles (`details`) del objeto `Stem`. Si se utiliza un solo valor como argumento, la longitud se aplica a todas las plicas. Si se usan varios argumentos, el primero se aplica a las corcheas, el sgundo a las semicorcheas y así sucesivamente. El último argumento también se aplica a todas las figuras que son mmás cortas que la longitud de la figura del último argumento. También se pueden usar argumentos no enteros.

```
\relative c'' {
  \override Stem.details.beamed-lengths = #'(2)
    a8[ a] a16[ a] a32[ a]
    \override Stem.details.beamed-lengths = #'(8 10 12)
    a8[ a] a16[ a] a32[ a] r8 |
  \override Stem.details.beamed-lengths = #'(8)
    a8[ a]
    \override Stem.details.beamed-lengths = #'(8.5)
    a8[ a]
    \revert Stem.details.beamed-lengths
    a8[ a] a16[ a] a32[ a] r16 |
}
```



Numeración de compases alternativa

Fijando la propiedad de contexto `alternativeNumberingStyle`, se dispone de dos métodos alternativos adicionales para la numeración de compases dentro de las repeticiones.

```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|."
}

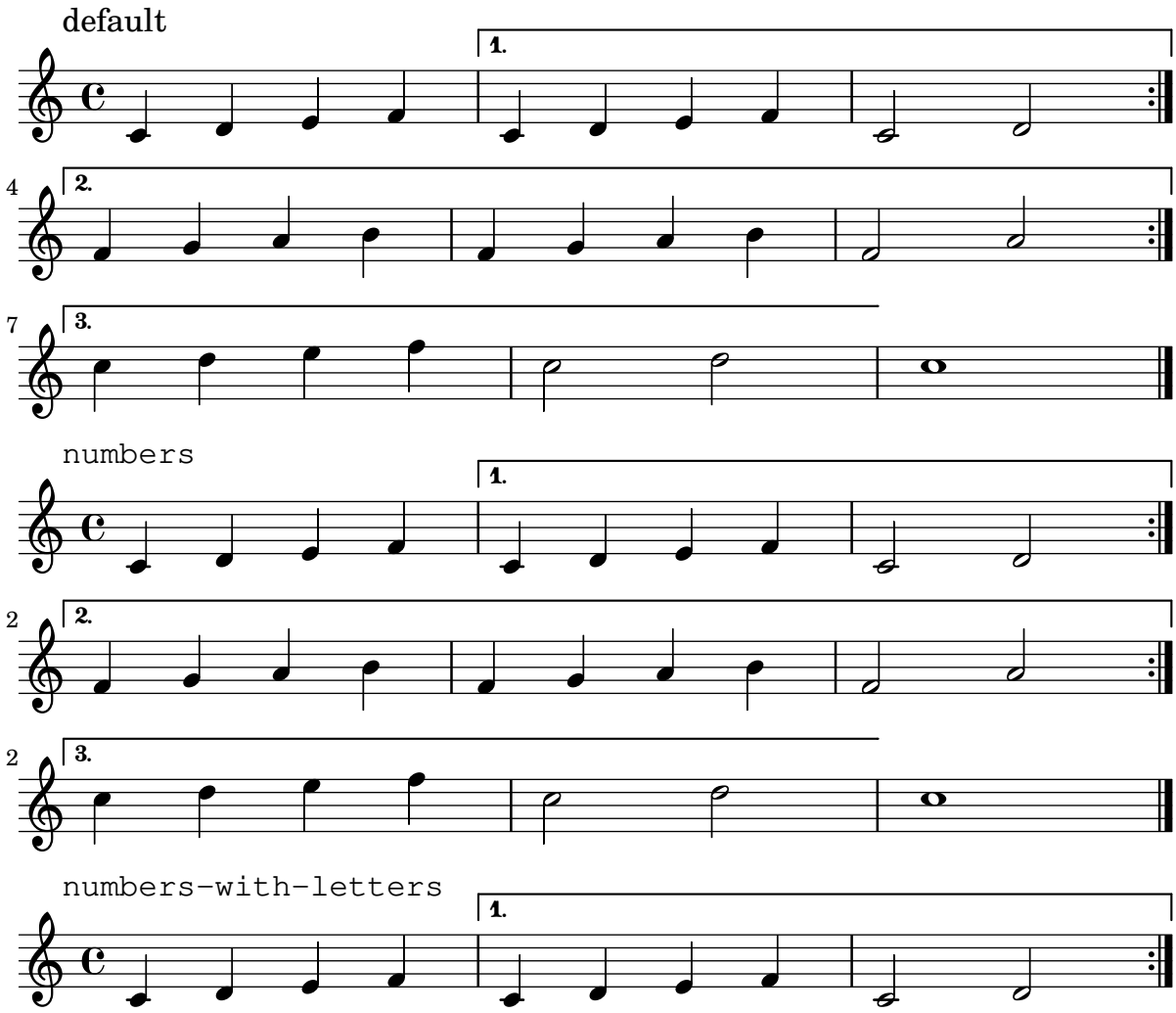
{
```

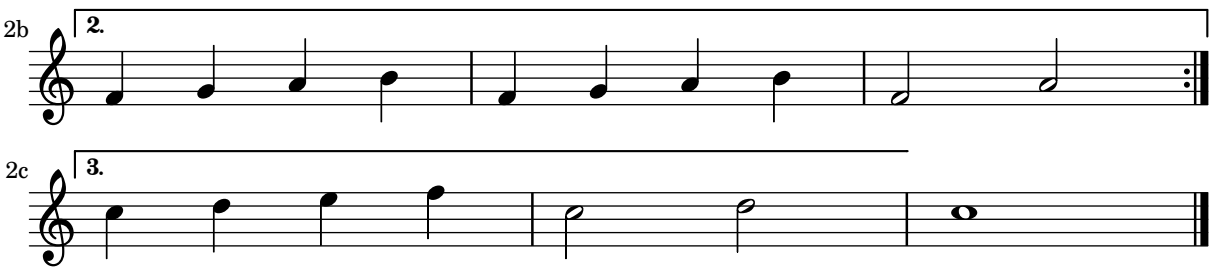
```
  \textMark \markup \large "default"
  \music
}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
  \music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}
```





Corchetes de análisis encima del pentagrama

De forma predeterminada se añaden corchetes de análisis sencillos debajo del pentagrama. El ejemplo siguiente muestra una manera de colocarlos por encima.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```



Corchetes de análisis con etiquetas

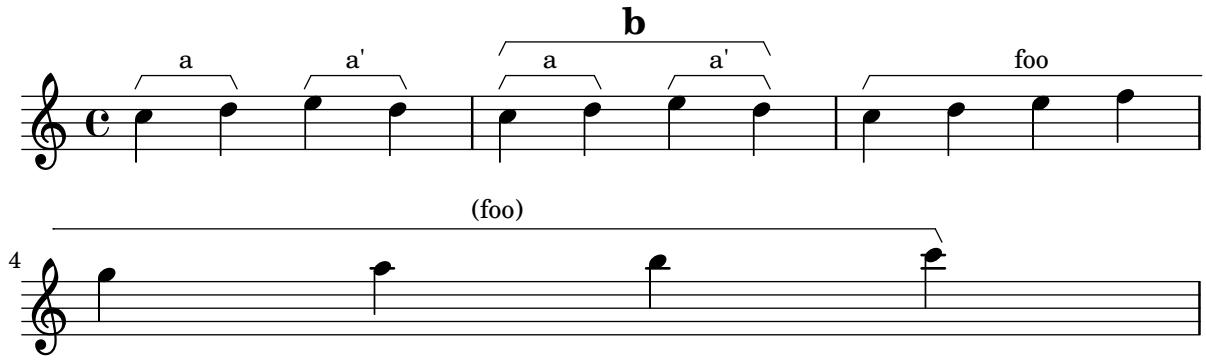
Se pueden añadir indicaciones de marcado de texto a los corchetes de análisis por medio de la propiedad `text` del objeto gráfico `HorizontalBracketText`. Si se quieren añadir textos diferentes al corchetes que comienzan en el mismo tiempo, es necesario usar la instrucción `\tweak`.

El texto del corchete se encierra entre paréntesis después de un salto de línea. El orden vertical de los corchetes anidados se puede controlar con la propiedad `outside-staff-priority`.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
    c''\startGroup d''\stopGroup
    \once\override HorizontalBracketText.text = "a'"
    e''\startGroup d''\stopGroup |
  c''-\tweak outside-staff-priority #801
      \tweak HorizontalBracketText.text
        \markup \bold \huge "b" \startGroup
```

```
     -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```

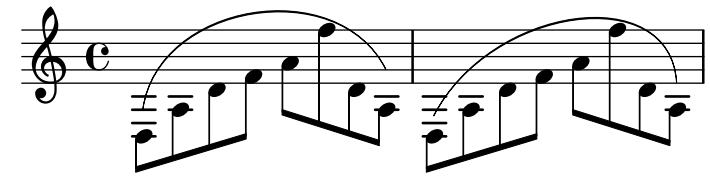


Ligaduras asimétricas

Se puede hacer que una ligadura de expresión sea asimétrica para adaptarse mejor a un patrón asimétrico de notas.

```
slurNotes = { d,8( a' d f a f' d, a) }

\relative c' {
  \stemDown
  \slurUp
  \slurNotes
  \once \override Slur.eccentricity = #3.0
  \slurNotes
}
```



Breaking horizontal alignment of dynamics and textscripts

LilyPond uses `DynamicLineSpanner` grobs to horizontally align successive dynamic objects like hairpins and dynamic text, even if they are positioned on different sides of a staff. This connection cannot be broken, contrary to the vertical alignment (see snippet “Breaking vertical alignment of dynamics and textscripts”).

There are two solutions to circumvent the problem.

- Modify the `shorten-pair` property of the `Hairpin` grob to compensate the offset by which the hairpin was moved.
- Put the two dynamic objects into different voices.

Both solutions are demonstrated in this snippet.

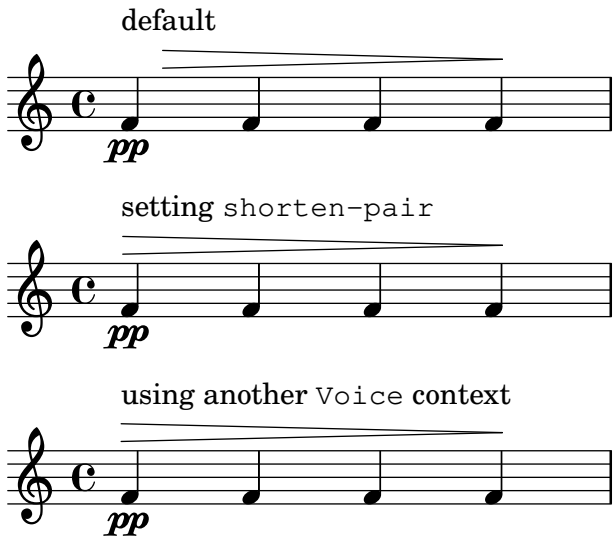
```
{
  <>^"default"
  f'_\pp ^\> f' f' f'\!
}

{
  <>^\markup { setting \typewriter shorten-pair }
  f'_\pp \tweak shorten-pair #'(-3 . 0) ^\> f' f' f'\!
}

{
  <>^\markup { using another \typewriter Voice context }
  << { f'^\> f' f' f'\! }
     \new Voice { s4_\pp } >>
}

\layout {
  line-width = 8\cm
  ragged-right = ##f

  \context {
    \Voice
    \override TextScript.staff-padding = #3.5
  }
}
```



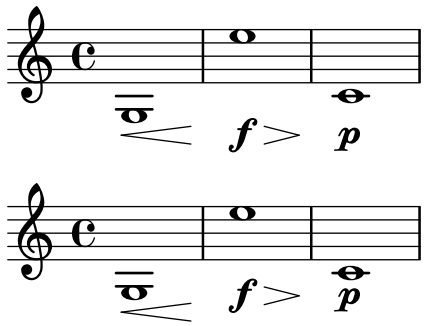
Desalinear indicaciones dinámicas y textuales verticalmente

De forma predeterminada, LilyPond usa ojetos gráficos DynamicLineSpanner para alinear verticalmente objetos de matiz dinámico sucesivos como reguladores y dinamicas textuales. Sin embargo esto no siempre es deseable. Insertando \breakDynamicSpan, que finaliza el objeto extenso de alineamiento de forma prematura, se puede evitar este alineamiento vertical.

Véase también el fragmento de código “Desalinear indicaciones dinámicas y textuales horizontalmente”.

```
{ g1\< |
  e''\f\> |
  c'\p }
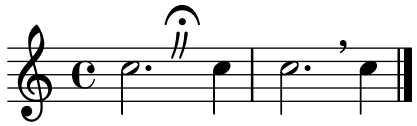
{ g1\< |
  e''\breakDynamicSpan\f\> |
  c'\p }
```



Caesura ("railtracks") with fermata

A caesura is sometimes denoted by a double "railtracks" breath mark with a fermata sign positioned above. This snippet shows an optically pleasing combination of railtracks and fermata.

```
\relative c'' {
  c2.
  % construct the symbol
  \override BreathingSign.text = \markup {
    \override #'(direction . 1)
    \override #'(baseline-skip . 1.8)
    \dir-column {
      \translate #'(0.155 . 0)
        \center-align \musicglyph "scripts.caesura.curved"
      \center-align \musicglyph "scripts.ufermata"
    }
  }
  \breathe c4
  % set the breath mark back to normal
  \revert BreathingSign.text
  c2. \breathe c4
  \bar "|."
}
```

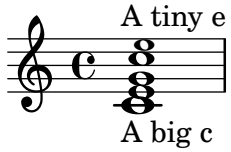


Changing a single note's size in a chord

Individual note heads in a chord can be modified with the `\tweak` command inside a chord, by altering the `font-size` property.

Inside the chord (within the brackets `< >`), before the note to be altered, place the `\tweak` command, followed by `font-size` and define the proper size like `#-2` (a tiny note head).
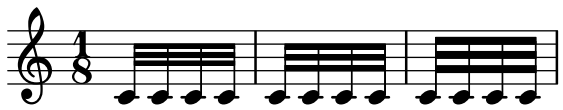
```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Cambiar el grosor y el espaciado de las barras

Para hacer que las barras sean más gruesas o más delgadas, altere la propiedad `beam-thickness` del objeto gráfico `Beam`. Para ajustar el espaciado entre las barras, altere la propiedad `length-fraction`.

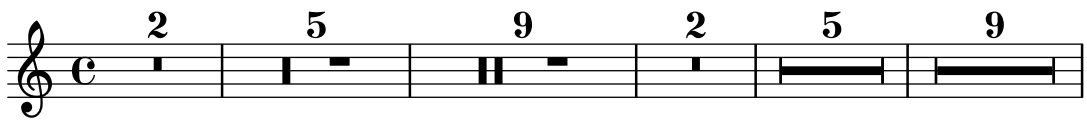
```
\relative f' {
  \time 1/8
  \override Beam.beam-thickness = #0.4
  \override Beam.length-fraction = #0.8
  c32 c c c
  \revert Beam.beam-thickness  % 0.48 is default thickness
  \revert Beam.length-fraction  % 1.0 is default spacing
  c32 c c c
  \override Beam.beam-thickness = #0.6
  \override Beam.length-fraction = #1.3
  c32 c c c
}
```



Cambiar la forma de los silencios multicompás

Si hay diez compases de silencio o menos, se imprime en el pentagrama una serie de silencios de breve y longa (conocidos en alemán como "Kirchenpausen", «silencios eclesiásticos»); en caso contrario se muestra una barra gruesa y larga. Este valor predeterminado de diez se puede cambiar sobreescribiendo la propiedad `expand-limit`:

```
\relative c'' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = 3
    R1*2 | R1*5 | R1*9
  }
}
```



Modificación de propiedades para objetos gráficos individuales
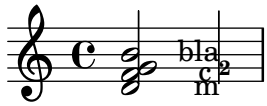
La instrucción `\applyOutput` hace posible el ajuste fino de cualquier objeto de presentación, en cualquier contexto. Requiere una función de Scheme contres argumentos.

En el ejemplo de abajo, la función `mc-squared` se ejecuta para todos los objetos gráficos `NoteHead` (dentro del contexto `Voice` actual) en el punto temporal en curso; la función modifica el `stencil` o sello del objeto gráfico, usando la propiedad `staff-position` para sustituir ciertas alturas con elementos de marcado.

Consulte el 'Manual de extensión del programa' (`https://lilypond.org/doc/v2.24/Documentation/extending/running-a-function-on-all-layout-objects`) para más información.

```
#(define (mc-squared grob grob-origin context)
   (let ((sp (ly:grob-property grob 'staff-position)))
     (ly:grob-set-property!
      grob 'stencil
      (grob-interpret-markup grob
       #{ \markup \lower #0.5
           #(case sp
              ((-5) "m")
              ((-3) "c ")
              ((-2) #{ \markup \teeny \bold 2 #})
              (else "bla")) #}))))

\relative c' {
  <d f g b>2
  \applyOutput Voice.NoteHead #mc-squared
  <d f g b>2
}
```
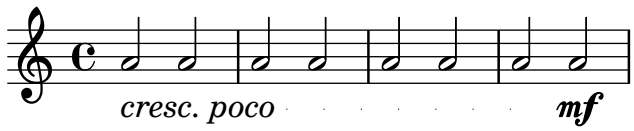


Cambiar el texto y los estilos de objeto de extensión para las indicaciones dinámicas textuales

Se puede modificar el texto empleado para los *crescendos* y *decrescendos* modificando las propiedades de contexto `crescendoText` y `decrescendoText`.

El estilo de la línea de extensión se puede cambiar modificando la propiedad `'style` de `DynamicTextSpanner`. El valor predeterminado es `dashed-line` (línea discontinua), y entre otros valores posibles se encuentran `line` (línea), `dotted-line` (línea de puntos) y `none` (nada):

```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Cambiar la familia de tipografía predeterminada para el texto

Las familias de fuente tipográfica para el texto se pueden sobreescribir.

```
%{
You may have to install additional fonts.

Red Hat Fedora: dejavu-fonts-all

Debian GNU/Linux, Ubuntu: fonts-dejavu-core
                          fonts-dejavu-extra
%}

\paper {
  %{
     run
         lilypond -dshow-available-fonts
     to show all fonts available in the process log.
  %}
  property-defaults.fonts.serif = "DejaVu Serif"
  property-defaults.fonts.sans = "DejaVu Sans"
  property-defaults.fonts.typewriter = "DejaVu Sans Mono"
}

{
  g'''4^\markup {
    DejaVu Serif: \bold bold
                  \italic italic
                  \italic \bold { bold italic }
  }
  g4_\markup {
    \override #'(font-family . sans) {
      DejaVu Sans: \bold bold
                   \italic italic
                   \italic \bold { bold italic }
    }
  }
  g''2^\markup {
    \override #'(font-family . typewriter) {
      DejaVu Sans Mono: \bold bold
                        \italic italic
                        \italic \bold { bold italic }
    }
  }
}
```



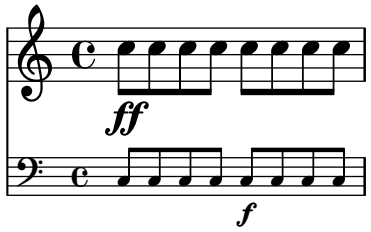
Modificar el tamaño de la pauta

La manera más sencilla de redimensionar los pentagramas es usar

```
#(set-global-staff-size tamaño)
```

El tamaño de una pauta individual se puede cambiar usando las propiedades `staff-space` y de `fontSize`.
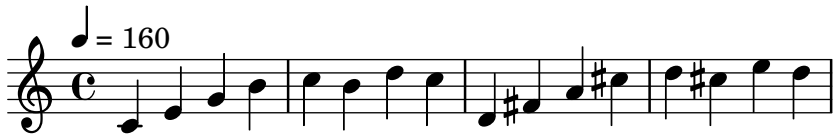
```
<<
  \new Staff \relative c'' {
    \dynamicDown c8\ff c c c c c c c
  }
  \new Staff \with {
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
  } \relative c {
    \clef bass c8 c c c c\f c c c
  }
>>
```



Cambiar el tempo sin indicación metronómica

Para cambiar el tempo en la salida MIDI sin imprimir nada, hacemos invisible la indicación metronómica:

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```
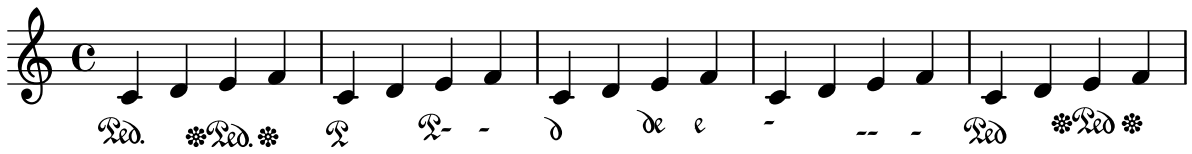


Modificar el texto de las indicaciones de pedal

Se puede usar la propiedad de contexto `Staff.pedalSustainStrings` para fijar el texto de las indicaciones de pisar pedal y levantar pedal. Observe que las únicas cadenas válidas son las que están en la lista de glifos de pedal: los valores que aparecen en este fragmento de código son una relación exhaustiva.

```
sustainNotes = { c4\sustainOn d e\sustainOff\sustainOn f\sustainOff }

\relative c' {
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("P" "P-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("d" "de" "e")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("M" "M-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("Ped" "*Ped" "*")
  \sustainNotes
}

\layout {
  ragged-right = ##f
}
```



Controlar la visibilidad de los objetos de extensión después de un salto de línea

La visibilidad de los objetos de extensión que acaban en la primera nota después de un salto de línea está controlada por la función de callback de `after-line-breaking` `ly:spanner::kill-zero-spanned-time`.

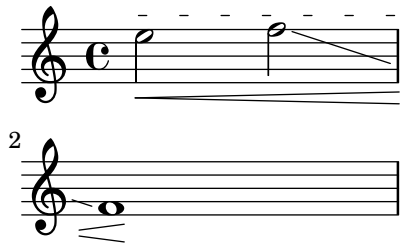
Para los objetos como los glissandos y los reguladores, el comportamiento predeterminado es ocultar el objeto de extensión después del salto; la inhabilitación de la función de callback hace que el objeto de extensión roto por la izquierda pueda mostrarse.

De forma inversa, los objetos de extensión que son visibles normalmente, como los objetos de extensión de texto, se pueden ocultar habilitando la función de callback.

```
\paper {
  line-width = 50\mm
}

\relative c'' {
  \override Hairpin.to-barline = ##f
  \override Glissando.breakable = ##t
```

```
  % show hairpin
  \override Hairpin.after-line-breaking = ##t
  % hide text span
  \override TextSpanner.after-line-breaking =
    #ly:spanner::kill-zero-spanned-time
  e2\<\startTextSpan
  % show glissando
  \override Glissando.after-line-breaking = ##t
  f2\glissando
  \break
  f,1\!\stopTextSpan
}
```



Controlar el aspecto de las barras de trémolo

Usando diversas propiedades del objeto gráfico `StemTremolo` es posible controlar la apariencia de las barras de trémolo.

- La propiedad `slope` fija la inclinación de las barras de trémolo.
- La propiedad `shape` determina si las barras de trémolo tienen forma de rectángulo (valor `rectangle`) o de barras de corchea pequeñas (valor `beam-like`).
- La propiedad `style` fija la inclinación y la forma dependiendo de si la nota tiene corchete, barra o solo plica. Esto existe en contraste con las dos propiedades anteriores, que cambian la inclinación y la forma incondicionalmente. Hay dos estilos definidos.
  - `default`: las barras para los corchetes de plica hacia abajo son más largos y más inclinados que las barras para los corchetes de plica hacia arriba; las barras de trémolo sobre notas barradas tienen una forma rectangular y son paralelas a la barra de corchea.
  - `constant`: todas las barras de trémolo tienen forma de barra de corchea y tienen la misma inclinación excepto para los corchetes de plica hacia abajo.

```
music = {
  a''4:32 a':
  e''8: \noBeam e':
  a'':[ a':]
  f':[ g':]
  d':[ d':]
}

\new Staff {
  <>^\markup "default"
  \music
}

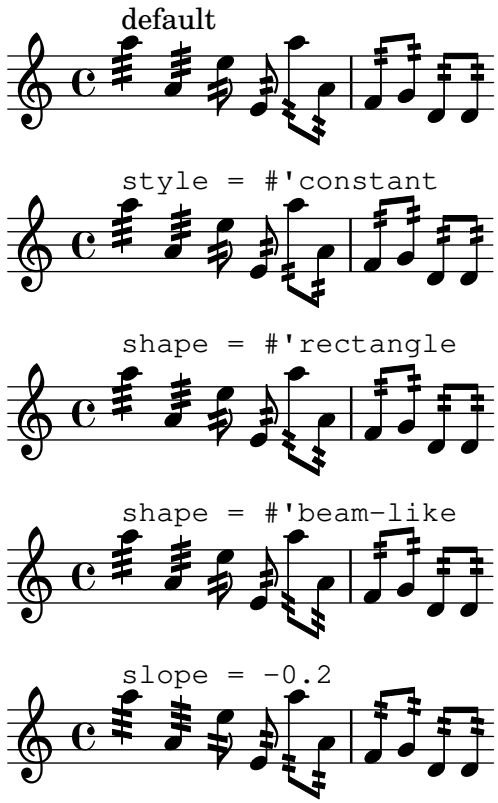
\new Staff {
  <>^\markup \typewriter "style = #'constant"
```

```
  \override StemTremolo.style = #'constant
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'rectangle"
  \override StemTremolo.shape = #'rectangle
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'beam-like"
  \override StemTremolo.shape = #'beam-like
  \music
}

\new Staff {
  <>^\markup \typewriter "slope = -0.2"
  \override StemTremolo.slope = -0.2
  \music
}
```



Controlar la ordenación vertical de las inscripciones

El orden vertical que ocupan las inscripciones gráficas está controlado con la propiedad `script-priority`. Cuanto más bajo es este número, más cerca de la nota se colocará. En este ejemplo, el `TextScript` (el símbolo del *sostenido*) tiene primero la prioridad más baja, por lo que se sitúa en la posición más baja en el primer ejemplo. En el segundo, el `Script` (el símbolo del *semitrino*) es el que la tiene más baja, por lo que se sitúa en la parte interior. Cuando dos

objetos tienen la misma prioridad, el orden en que se introducen determina cuál será el que aparece en primer lugar.

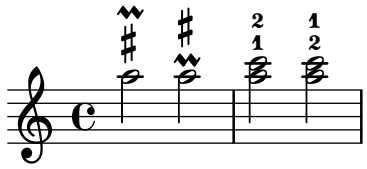
Observe que para los objetos gráficos `Fingering`, `StringNumber` y `StrokeFinger`, si se usan dentro de un acorde, el orden vertical también viene determinado por la posición vertical de la cabeza de la nota asociada, que se añade a (o, dependiendo de la dirección, se sustrae de) el valor `script-priority` del objeto gráfico. Esto nos asegura que para las digitaciones que están por encima de un acorde, la nota inferior está asociada con la digitación inferior (y viceversa para la otra dirección); no importa si escribimos las notas del acorde de arriba a abajo o de abajo a arriba.

De manera predeterminada, las inscripciones de carácter menos técnico se sitúan más cerca de la cabeza de la nota; el orden es más o menos: articulación, armónico, digitación, digitación de la mano derecha, número de cuerda, calderón, indicaciones de arco, e inscripción de texto.

```
\relative c''' {
  \once \override TextScript.script-priority = -100
  a2^\prall^\markup { \sharp }

  \once \override Script.script-priority = -100
  a2^\prall^\markup { \sharp }

  \set fingeringOrientations = #'(up)
  <c-2 a-1>2
  <a-1 c\tweak script-priority -100 -2>2
}
```



Controlar la visibilidad de los corchetes de grupo especial

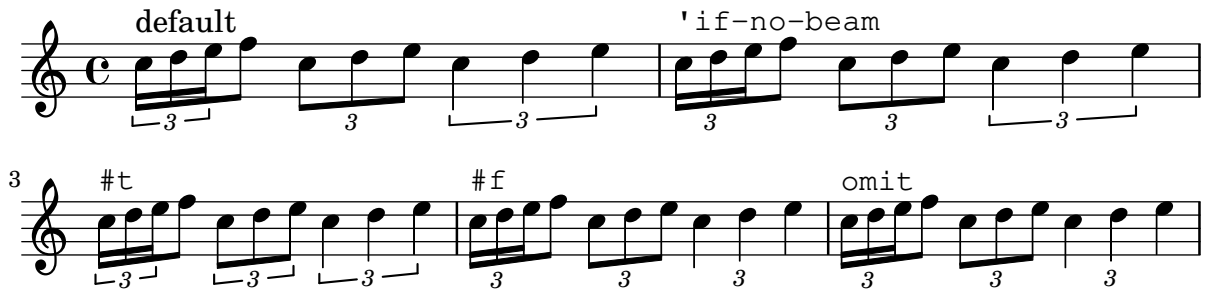
El comportamiento predeterminado de la visibilidad de los corchetes de grupo de valoración especial es imprimir el corchete a no ser que haya una barra de la misma longitud que el grupo especial.

Para controlar la visibilidad de los corchetes de grupo, establezca la propiedad `bracket-visibility` a `#t` (imprimir el corchete siempre), `#if-no-beam` (imprimir el corchete solamente si no hay barra), o `#f` (no imprimir nunca el corchete). Este último equivale de hecho a omitir el objeto `TupletBracket` de la salida impresa.

```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e } f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

\new Voice {
  \relative c' {
    \override Score.TextMark.non-musical = ##f
    \textMark "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    \textMark \markup \typewriter "'if-no-beam" \music
    \override TupletBracket.bracket-visibility = ##t
```
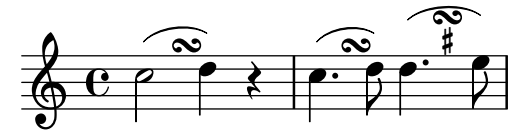
```
    \textMark \markup \typewriter "#t" \music
    \override TupletBracket.bracket-visibility = ##f
    \textMark \markup \typewriter "#f" \music
    \omit TupletBracket
    \textMark \markup \typewriter "omit" \music
  }
}
```



Crear un grupeto de anticipación

La creación de un grupeto circular de anticipación entre dos notas, donde la nota inferior del grupeto utiliza una alteración, requiere varias sobreescrituras de propiedades. La propiedad `outside-staff-priority` se debe establecer al valor `#f`, pues en caso contrario tendría prioridad sobre la propiedad `avoid-slur`. Cambiando el primer argumento de `\after` (que es una duración) puede ajustarse la posición horizontal.

```
\relative c'' {
  \after 2*2/3 \turn c2( d4) r |
  \after 4 \turn c4.( d8)
  \after 4
  {
    \once \set suggestAccidentals = ##t
    \once \override AccidentalSuggestion.outside-staff-priority = ##f
    \once \override AccidentalSuggestion.avoid-slur = #'inside
    \once \override AccidentalSuggestion.font-size = -3
    \once \override AccidentalSuggestion.script-priority = -1
    \once \hideNotes
    cis8\turn \noBeam
  }
  d4.( e8)
}
```
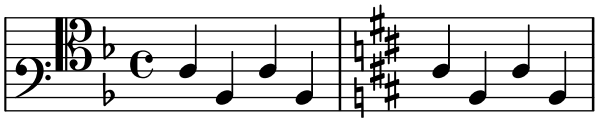


Creación de armaduras de clave personalizadas

LilyPond contempla armaduras de tonalidad personalizadas. En este ejemplo se muestra una armadura de Re menor y Re mayor con un rango ampliado de bemoles mostrados.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil =
    #(lambda (grob)
        (grob-interpret-markup grob
          #{ \markup\combine
               \musicglyph "clefs.C"
               \translate #'(-3 . -2)
                 \musicglyph "clefs.F"
          #}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}

{
  \key d\minor f bes, f bes, |
  \key d\major fis b, fis b, |
}
```



Crear elementos de extensión textuales

Las instrucciones \startTextSpan y \stopTextSpan permiten la creación de elementos de extensión textuales tan fácilmente como indicaciones de pedal u octavaciones. Sobreescribimos ciertas propiedades del objeto TextSpanner para modificar su salida.

```
\paper { ragged-right = ##f }

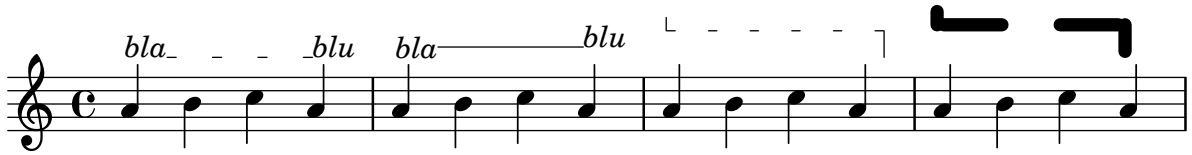
\relative c'' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.bound-details.left.text =
    \markup { \draw-line #'(0 . 1) }
```

```
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . -2) }
  \once \override TextSpanner.bound-details.right.padding = #-2
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.dash-period = #10
  \override TextSpanner.dash-fraction = #0.5
  \override TextSpanner.thickness = #10
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}
```



Solución de problemas de acordes y barras que cruzan el pentagrama

A veces es mejor usar plicas que parten del 'otro' pentagrama para crear acordes de pentagrama cruzado y así engañar al detector de colisiones de plicas de LilyPond. En el siguiente fragmento de código, si se hubieran usado las plicas que provienen del pentagrama inferior, habría sido necesario usar explícitamente

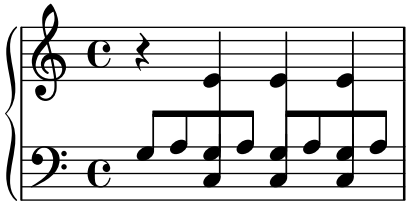
```
\override Staff.Beam.collision-voice-only = ##t
```

de forma que LilyPond no mueva las plicas.

```
\new PianoStaff <<
  \new Staff = up \relative c' <<
    { r4
      \override Stem.cross-staff = ##t
      \override Stem.length = #19 % this is in half-spaces,
          % so it makes stems 9.5 staffspaces long
      \override Stem.Y-offset = #-6 % stems are normally lengthened
          % upwards, so here we must lower the stem by the amount
          % equal to the lengthening - in this case (19 - 7) / 2
          % (7 is default stem length)
      e e e }
    { s4
      \change Staff = "bottom"
      \override NoteColumn.ignore-collision = ##t
      c, c c
    }
  >>

  \new Staff = bottom \relative c' {
    \clef bass
    \voiceOne
    g8 a g a g a g a
```

```
  }
>>
```



Plicas de pentagrama cruzado

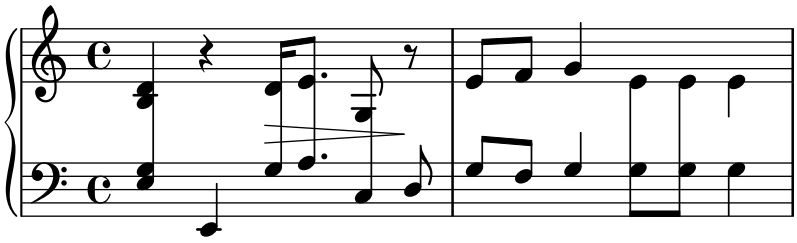
Este fragmento de código muestra el uso del grabador `Span_stem_engraver` y de `\crossStaff` para conectar automáticamente plicas de un pentagrama a otro.

No es necesario especificar las longitudes de las plicas porque la distancia variable entre las cabezas de las notas y los pentagramas se calcula automáticamente. Sin embargo, es importante aplicar `\crossStaff` a la voz o pentagrama correctos (esto es, en el lado opuesto de donde está o estaría posicionada la barra de corcheas) para conseguir el efecto deseado

```
\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

\new PianoStaff <<
  \new Staff {
    <b d'>4 r d'16\> e'8. g8 r\! |
    e'8 f' g'4
      \voiceTwo
      % Down to lower staff
      \crossStaff { e'8 e'8 } e'4 |
  }

  \new Staff {
    \clef bass
    \voiceOne
    % Up to upper staff
    \crossStaff { <e g>4 e, g16 a8. c8 } d |
    g8 f g4 \voiceTwo g8 g g4 |
  }
>>
```



Custos

Se pueden tipografiar «custos» en diferentes estilos.

```
\layout {
  ragged-right = ##t
}

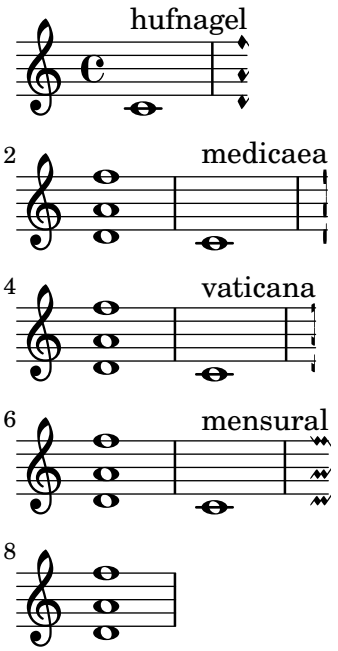
\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

    \override Staff.Custos.style = #'hufnagel
    c1^"hufnagel" \break
    <d a' f'>1

    \override Staff.Custos.style = #'medicaea
    c1^"medicaea" \break
    <d a' f'>1

    \override Staff.Custos.style = #'vaticana
    c1^"vaticana" \break
    <d a' f'>1

    \override Staff.Custos.style = #'mensural
    c1^"mensural" \break
    <d a' f'>1
  }
}
```



Personalizar los diagramas de posiciones

Se pueden modificar las propiedades de los diagramas de posiciones de acordes estableciendo la propiedad `fret-diagram-details`. Para los diagramas de posiciones de `FretBoard`, se aplican los overrides (sobreescrituras) al objeto `FretBoards.FretBoard`. Como `Voice`, `FretBoards` es un contexto del nivel inferior, y por tanto se puede omitir su nombre en la sobreescritura de propiedades.

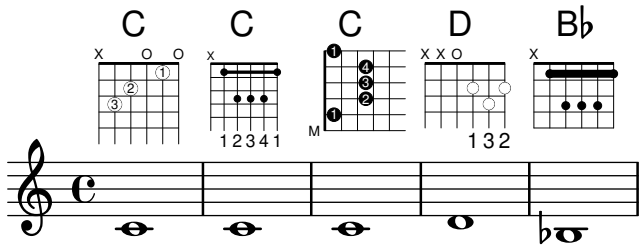
```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        "x;1-1-(;3-2;3-3;3-4;1-1-);"

% shorthand
oo = #(define-music-function
       (grob-path value)
       (list? scheme?)
       #{ \once \override $grob-path = #value #})

<<
  \new ChordNames {
    \chordmode { c1 | c | c | d | bes }
  }
  \new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard.size = 1.2
    \override FretBoard.fret-diagram-details.finger-code = #'in-dot
    \override FretBoard.fret-diagram-details.dot-color = #'white
    \chordmode {
      c
      \oo FretBoard.size #1.0
      \oo FretBoard.fret-diagram-details.barre-type #'straight
      \oo FretBoard.fret-diagram-details.dot-color #'black
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      c'
      \oo FretBoard.fret-diagram-details.barre-type #'none
      \oo FretBoard.fret-diagram-details.number-type #'arabic
      \oo FretBoard.fret-diagram-details.orientation #'landscape
      \oo FretBoard.fret-diagram-details.mute-string "M"
      \oo FretBoard.fret-diagram-details.label-dir #LEFT
      \oo FretBoard.fret-diagram-details.dot-color #'black
      c'
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      \oo FretBoard.fret-diagram-details.dot-radius #0.35
      \oo FretBoard.fret-diagram-details.dot-position #0.5
      \oo FretBoard.fret-diagram-details.fret-count #3
      d
      \oo FretBoard.fret-diagram-details.barre-type #'straight
      \oo FretBoard.fret-diagram-details.finger-code #'none
      \oo FretBoard.fret-diagram-details.dot-radius #0.25
      \oo FretBoard.fret-diagram-details.dot-color #'black
      \oo FretBoard.fret-diagram-details.string-overhang #0.
      \oo FretBoard.fret-diagram-details.barre-thickness #2.
      bes
    }
  }
  \new Voice {
    c'1 | c' | c' | d' | bes
  }
```

>>



Personalizar diagramas de posiciones de marcado

Se pueden modificar las propiedades de los diagramas de posiciones estableciendo el valor de la propiedad `fret-diagram-details`. Para los diagramas de posiciones de marcado, se pueden aplicar overrides (sobreescrituras) al objeto `Voice.TextScript` o directamente al elemento de marcado.

```
<<
  \chords { c1 | c | c | d }

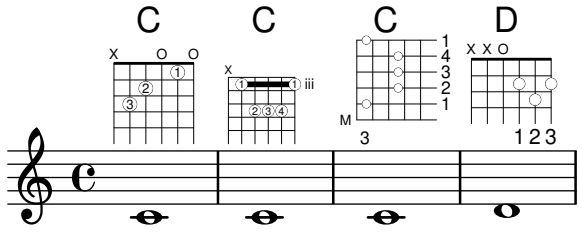
  \new Voice = "mel" {
    \textLengthOn
    % Set global properties of fret diagram
    \override TextScript.size = 1.2
    \override TextScript.fret-diagram-details.finger-code = #'in-dot
    \override TextScript.fret-diagram-details.dot-color = #'white

    %% C major for guitar, no barre, using defaults
       % terse style
    c'1^\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

    %% C major for guitar, barred on third fret
       % verbose style
       % size 1.0
       % roman fret label, finger labels below string, straight barre
    c'1^\markup {
      % standard size
      \override #'(size . 1.0) {
        \override #'(fret-diagram-details . (
                    (number-type . roman-lower)
                    (finger-code . in-dot)
                    (barre-type . straight))) {
          \fret-diagram-verbose #'((mute 6)
                                   (place-fret 5 3 1)
                                   (place-fret 4 5 2)
                                   (place-fret 3 5 3)
                                   (place-fret 2 5 4)
                                   (place-fret 1 3 1)
                                   (barre 5 1 3))
        }
      }
    }
```

```
    %% C major for guitar, barred on third fret
       % verbose style
       % landscape orientation, arabic numbers, M for mute string
       % no barre, fret label down or left, small mute label font
    c'1^\markup {
      \override #'(fret-diagram-details . (
                    (finger-code . below-string)
                    (number-type . arabic)
                    (label-dir . -1)
                    (mute-string . "M")
                    (orientation . landscape)
                    (barre-type . none)
                    (xo-font-magnification . 0.4)
                    (xo-padding . 0.3))) {
        \fret-diagram-verbose #'((mute 6)
                                 (place-fret 5 3 1)
                                 (place-fret 4 5 2)
                                 (place-fret 3 5 3)
                                 (place-fret 2 5 4)
                                 (place-fret 1 3 1)
                                 (barre 5 1 3))
      }
    }

    %% simple D chord
       % terse style
       % larger dots, centered dots, fewer frets
       % label below string
    d'1^\markup {
      \override #'(fret-diagram-details . (
                    (finger-code . below-string)
                    (dot-radius . 0.35)
                    (dot-position . 0.5)
                    (fret-count . 3))) {
        \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```



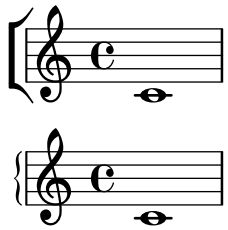
Mostrar corchete o llave en grupos de un solo pentagrama

Si hay un solo pentagrama en un contexto `ChoirStaff` o `StaffGroup`, el comportamiento predeterminado es que no se imprima el corchete en la barra inicial. Esto se puede cambiar estable-

ciendo la propiedad `collapse-height` a un valor que sea menor que el número de líneas en la pauta.

Observe que en contextos como `PianoStaff` y `GrandStaff` en que los sistemas empiezan con una llave en lugar de un corchete, se debe establecer el valor de una propiedad distinta, como se ve en el segundo sistema del ejemplo.

```
\score {
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
```



Imprimir el árbol genealógico de un grob

Al trabajar con los callbacks de un grob, puede ser de mucha ayuda entender el árbol genealógico de un grob. La mayor parte de los grobs tienen padres que influyen en el posicionamiento del grob. los padres X e Y influyen en las posiciones horizontal y vertical del grob, respectivamente. Además, cada pade puede tener padres a su vez.

Por desgracia, existen varios aspectos de la genealogía de un grob que pueden llevar a confusión:

- Los tipos de padre que tiene un grob pueden depender del contexto.
- Para ciertos grobs, los padres X e Y son el mismo.
- Un *ancestro* concreto puede estar relacionado con un grob de más de una manera.
- El concepto de *generaciones* es engañoso.

Por ejemplo, el grob `System` puede ser tanto un padre (sobre el lado Y) como un abuelo (dos veces en el lado X) de un grob `VerticalAlignment`.

El macro definido en este fragmento de código imprime en la consola una representación textual de la genealogía de un grob. Por ejemplo, la llamada

```
{
  \once \override NoteHead.before-line-breaking = #display-ancestry
```

```
  c
}
```

genera la siguiente salida.

```
----------------------------------
NoteHead
X,Y: NoteColumn
     X: PaperColumn
        X,Y: System
     Y: VerticalAxisGroup
        X: NonMusicalPaperColumn
           X,Y: System
        Y: VerticalAlignment
           X: NonMusicalPaperColumn
              X,Y: System
           Y: System
```

Como consecuencia, tenemos que ejecutar el código del fragmento de código por nosotros mismos, porque el archivo de salida no muestra los datos que nos interesan.
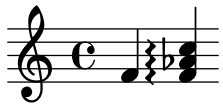
```
#(define (get-ancestry grob)
   (if (not (null? (ly:grob-parent grob X)))
       (list (grob::name grob)
             (get-ancestry (ly:grob-parent grob X))
             (get-ancestry (ly:grob-parent grob Y)))
       (grob::name grob)))

#(define (format-ancestry lst padding)
   (string-append
    (symbol->string (car lst)) "\n"
    (let ((X-ancestry (if (list? (cadr lst))
                          (format-ancestry (cadr lst) (+ padding 3))
                          (symbol->string (cadr lst))))
          (Y-ancestry (if (list? (caddr lst))
                          (format-ancestry (caddr lst) (+ padding 3))
                          (symbol->string (caddr lst)))))
      (if (equal? X-ancestry Y-ancestry)
          (string-append (format #f "~&")
                         (make-string padding #\space)
                         "X,Y: "
                         (if (list? (cadr lst))
                             (format-ancestry (cadr lst) (+ padding 5))
                             (symbol->string (cadr lst))))
          (string-append (format #f "~&")
                         (make-string padding #\space)
                         "X: " X-ancestry "\n"
                         (make-string padding #\space)
                         "Y: " Y-ancestry (format #f "~&")))))
    (format #f "~&")))

#(define (display-ancestry grob)
   (format (current-output-port)
           "~2&~a~2%~a~&"
           (make-string 36 #\-)
```

```
          (if (ly:grob? grob)
              (format-ancestry (get-ancestry grob) 0)
              (format #f "~a is not a grob" grob))))
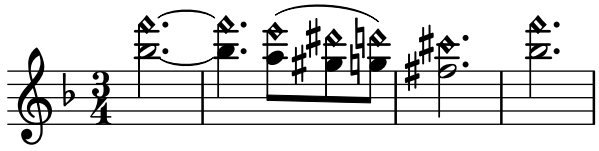
\relative c' {
  \once \override NoteHead.before-line-breaking = #display-ancestry
  f4
  \once \override Accidental.before-line-breaking = #display-ancestry
  \once \override Arpeggio.before-line-breaking = #display-ancestry
  <f as c>4\arpeggio
}
```



Armónicos con puntillo

Los armónicos artificiales que usan la instrucción \harmonic no tienen puntillo. Para sobreescribir este comportamiento, fije la propiedad de contexto harmonicDots.

```
\relative c''' {
  \time 3/4
  \key f \major
  \set harmonicDots = ##t
  <bes f'\harmonic>2. ~
  <bes f'\harmonic>4. <a e'\harmonic>8( <gis dis'\harmonic> <g d'\harmonic>)
  <fis cis'\harmonic>2.
  <bes f'\harmonic>2.
}
```



Rodear los objetos gráficos con rectángulos

La propiedad stencil se puede sobreescribir para trazar un rectángulo que encierre objetos gráficos arbitrarios, bien sea usando \override o bien \tweak.

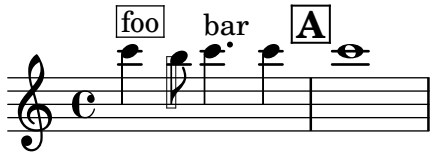
```
\relative c'' {
  \once \override TextScript.stencil =
    #(make-stencil-boxer 0.1 0.3 ly:text-interface::print)
  c'4^"foo"

  \tweak Stem.stencil
    #(make-stencil-boxer 0.05 0.25 ly:stem::print)
  b8

  c4.^"bar" c4

  \override Score.RehearsalMark.stencil =
    #(make-stencil-boxer 0.15 0.3 ly:text-interface::print)
  \mark \default
```

```
  c1
}
```

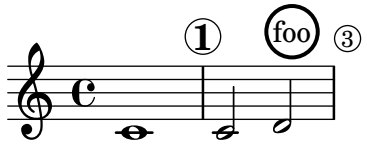


Rodear diversos objetos con una circunferencia

La instrucción \circle traza circunferencias alrededor de los objetos de marcado creados con \markup. Para otros objetos pueden requerirse trucos específicos, como se muestra aquí para el caso de las letras de ensayo y los números de compás.

```
\relative c' {
  c1
  \set Score.rehearsalMarkFormatter =
    #(lambda (mark context)
             (make-circle-markup (format-mark-numbers mark context)))
  \mark \default

  c2 d^\markup {
    \override #'(thickness . 3) {
      \circle foo
    }
  }
  \override Score.BarNumber.break-visibility = #all-visible
  \override Score.BarNumber.stencil =
    #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
}
```
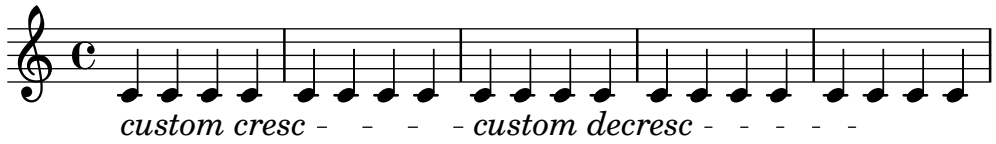


Dynamics spanner with custom text

Postfix functions for custom crescendo text spanners. The spanners should start on the first note of the measure. One has to use -\mycresc, otherwise the spanner start will rather be assigned to the next note.

```
% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.
mycresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'CrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))
mydecresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'DecrescendoEvent
               'span-direction START
               'span-type 'text
```

```
                'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4 c4\! c4 c4
}
```

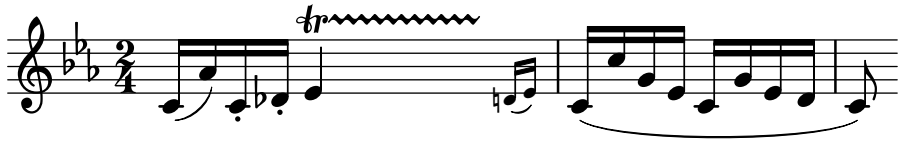


Extending a trill spanner

For `TrillSpanner` grobs, the `minimum-length` property becomes effective only if the `set-spacing-rods` procedure is called explicitly.

To do this, the `springs-and-rods` property should be set to `ly:spanner::set-spacing-rods`.

```
\relative c' {
  \key c\minor
  \time 2/4
  c16( as') c,-. des-.
  \once\override TrillSpanner.minimum-length = #15
  \once\override TrillSpanner.springs-and-rods = #ly:spanner::set-spacing-rods
  \afterGrace es4\startTrillSpan { d16[(\stopTrillSpan es)] }
  c( c' g es c g' es d
  c8)
}
```



Extender glissandos sobre repeticiones

Se puede simular un glissando que se extiende hasta el interior de varios bloques `\alternative` de primera y segunda vez mediante la adición de una nota de adorno oculta con un glissando al comienzo de cada bloque `\alternative`. La nota de adorno debe estar a la misma altura que la nota que da inicio al primer glissando. Esto se implementa aquí con una función musical que toma como argumento la altura de la nota de adorno.

Observe que en música polifónica la nota de adorno debe coincidir con las notas de adorno correspondientes en todas las otras voces.

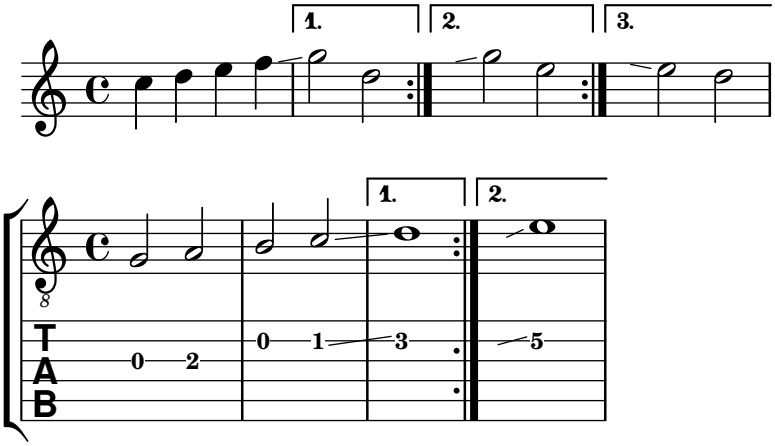
```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
```

```
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music =  \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}

\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff  <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
}
```
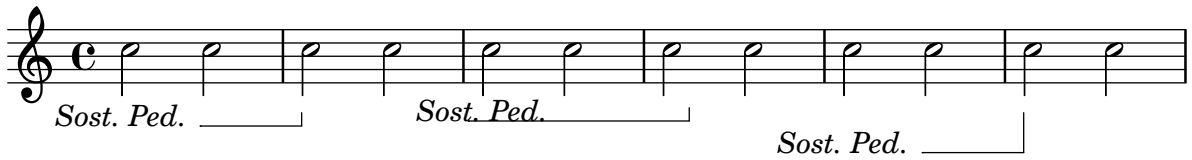


Ajuste fino de las líneas de pedal

Se puede alterar el aspecto de las líneas de pedal de varias formas.

```
\paper {
  ragged-right = ##f
}

\relative c'' {
  c2\sostenutoOn c
  c2\sostenutoOff c
  c2\tweak shorten-pair #'(-7 . -2) \sostenutoOn c
  c2\sostenutoOff c
  c2\tweak edge-height #'(0 . 3) \sostenutoOn c
  c2\sostenutoOff c
}
```



Ligaduras planas

Este fragmento de código aporta una función `flared-tie` para trazar una ligadura formada por líneas rectas. Su objetivo es sustituir a la función predeterminada de trazado de la ligadura (esto es, servir como un argumento sustitutorio para la propiedad `stencil` del objeto gráfico `Tie`).

El argumento de `flared-tie` es una lista de pare de coordenadas que especifican puntos adicionales entre el primero y el último para abarcar todas las líneas de la ligadura. Los puntos primero y último son idénticos a los puntos de inicio y final de la ligadura, respectivamente. Los valores de las coordenadas X e Y son múltiplos de la longitud y la altura del rectángulo circundante (teniendo también en cuenta la dirección de la ligadura); por tanto, el primer punto tiene las coordenadas (0,0), y el último (1,0).

La función `flare-tie` define una forma abreviada de especificar una ligadura plana. Es posible efectuar un trucaje adicional de la forma mediante la sobreescritura de `Tie.details.height-limit` o con la función `\shape`.

También es posible cambiar la definición personalizada sobre la marcha.

```
#(define ((flared-tie coords) grob)
   (define (pair-to-list pair)
     (list (car pair) (cdr pair)))

   (define (normalize-coords goods x y dir)
     (map
      (lambda (coord)
        (cons (* x (car coord)) (* y dir (cdr coord))))
      goods))

   (define (my-c-p-s points thick)
     (make-connected-path-stencil points thick 1.0 1.0 #f #f))

   ;; Calling `ly:tie::print` and assigning its return value to a
   ;; variable in this outer `let` triggers LilyPond to position the
   ;; tie, allowing us to extract its extents.  We only proceed,
```

```
   ;; however, if the tie doesn't get discarded (for whatever reason).
   (let ((sten (ly:tie::print grob)))
     (if (grob::is-live? grob)
         (let* ((layout (ly:grob-layout grob))
                (line-thickness (ly:output-def-lookup layout
                                                      'line-thickness))
                (thickness (ly:grob-property grob 'thickness 0.1))
                (used-thick (* line-thickness thickness))
                (dir (ly:grob-property grob 'direction))
                (xex (ly:stencil-extent sten X))
                (yex (ly:stencil-extent sten Y))
                (lenx (interval-length xex))
                (leny (interval-length yex))
                (xtrans (car xex))
                (ytrans (if (> dir 0)(car yex) (cdr yex)))
                ;; Add last point.
                (coord-list (append coords '((1.0 . 0.0))))
                (uplist
                 (map pair-to-list
                      (normalize-coords coord-list lenx (* leny 2) dir))))
           (ly:stencil-translate
            (my-c-p-s uplist used-thick)
            (cons xtrans ytrans)))
         '())))

% Define a default tie shape consisting of three straight lines.
#(define flare-tie
   (flared-tie '((0.1 . 0.3) (0.9 . 0.3))))

\relative c' {
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <a c e a c e a c e>~ q
  \once \override Tie.stencil = #flare-tie
  q~ q\break

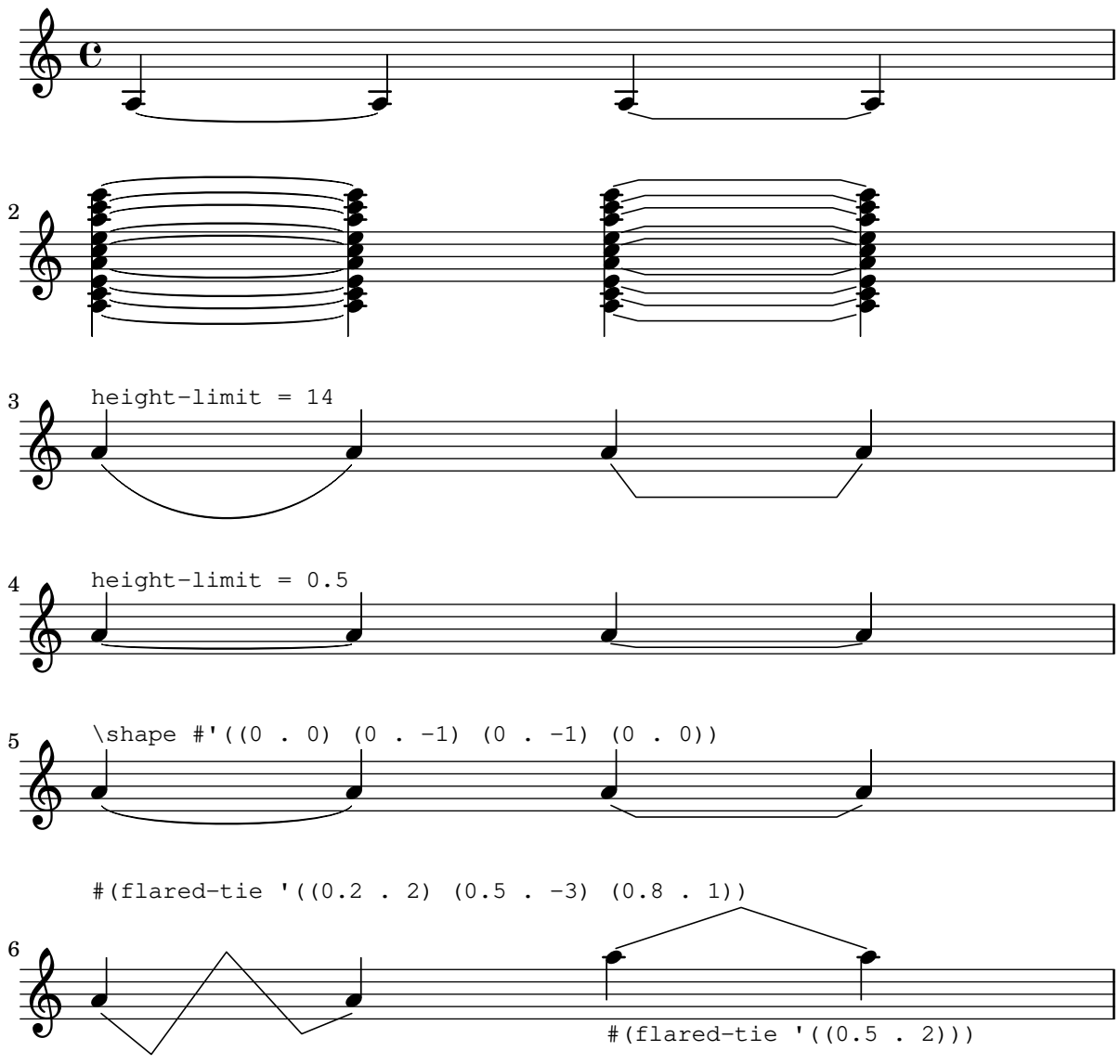
  <>^\markup \small \typewriter "height-limit = 14"
  \override Tie.details.height-limit = 14
  a'4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <>^\markup \small \typewriter "height-limit = 0.5"
  \override Tie.details.height-limit = 0.5
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  \revert Tie.details.height-limit
```

```
  <>^\markup \small \typewriter
            "\\shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0))"
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a
  \once \override Tie.stencil = #flare-tie
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a \break
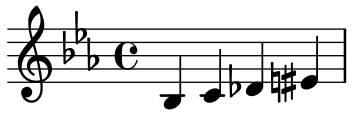
  <>^\markup \small \typewriter
            "#(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1))"
  \once \override Tie.stencil =
        #(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))
  a4~ a
  <>_\markup \small \typewriter
            "#(flared-tie '((0.5 . 2)))"
  \once \override Tie.stencil = #(flared-tie '((0.5 . 2)))
  a'4~ a
}
```



Forzar un becuadro de cancelación antes de una alteración accidental

El ejemplo siguiente muestra cómo forzar un becuadro antes de una alteración accidental.

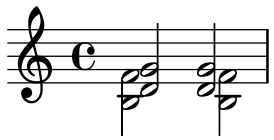
```
\relative c' {
  \key es \major
  bes c des
  \tweak Accidental.restore-first ##t
  eis
}
```



Forzar el desplazamiento horizontal de las notas

Cuando el motor de tipografiado no es capaz de todo, se puede usar la sintaxis siguiente para sobreescribir las decisiones de tipografía. Las unidades de medida que se usan aquí son espacios de pentagrama.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
    <b f'>2
    \once \override NoteColumn.force-hshift = 1.7
    <b f'>2
  }
>>
```



Diagramas de posiciones de acorde, explicados y desarrollados

Este fragmento de código presenta muchas posibilidades para obtener diagramas de posiciones de acorde y cómo trucarlos.

```
<<
  \chords {
    a1 a \bar "||" \break
    \repeat unfold 3 {
      c c c d d \bar "||" \break
    }
  }

  \new Voice {
    % Set global properties of fret diagram
    \override TextScript.size = 1.2
    \override TextScript.fret-diagram-details
                .finger-code = #'below-string
```

```
\override TextScript.fret-diagram-details
            .dot-color = #'black

% 1
%
% A chord for ukulele.
a'1^\markup
  \override #'(fret-diagram-details
               . ((string-count . 4)
                  (dot-color . white)
                  (finger-code . in-dot)))
  \fret-diagram "4-2-2;3-1-1;2-o;1-o;"

% 2
%
% A chord for ukulele, with formatting defined in definition
% string: 1.2 * size, 4 strings, 4 frets, fingerings below,
% string dot radius .35 of fret spacing, dot position 0.55 of
% fret spacing.
a'1^\markup
  \override #'(fret-diagram-details
               . ((dot-color . white)
                  (open-string . "o")))
  \fret-diagram
    "s:1.2;w:4;h:3;f:2;d:0.35;p:0.55;4-2-2;3-1-1;2-o;1-o;"


%%
%% These chords will be in normal orientation
%%


% 3
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-lower)
                  (finger-code . below-string)
                  (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 5 1 3))

% 4
```

```
%
% C major for guitar, barred on third fret: double barre used
% to test barre function, verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . arabic)
                  (dot-label-font-mag . 0.9)
                  (finger-code . in-dot)
                  (fret-label-font-mag . 0.6)
                  (fret-label-vertical-offset . 0)
                  (label-dir . -1)
                  (mute-string . "M")
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 4 2 5)
                           (barre 5 1 3))

% 5
%
% C major for guitar, with capo on third fret: verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-upper)
                  (dot-label-font-mag . 0.9)
                  (finger-code . none)
                  (fret-label-vertical-offset . 0.5)
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (capo 3)
                           (open 5)
                           (place-fret 4 5 1)
                           (place-fret 3 5 2)
                           (place-fret 2 5 3)
                           (open 1))

% 6
%
% Simple D chord.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
```

```
                  (dot-radius . 0.35)
                  (string-thickness-factor . 0.3)
                  (dot-position . 0.5)
                  (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 7
%
% Simple D chord, large top fret thickness.
d'1^\markup
  \override #'(fret-diagram-details
              . ((finger-code . below-string)
                 (dot-radius . 0.35)
                 (dot-position . 0.5)
                 (top-fret-thickness . 7)
                 (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"


%%
%% These chords will be in landscape orientation
%%
\override TextScript.fret-diagram-details
            .orientation = #'landscape


% 8
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight
% barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
              . ((number-type . roman-lower)
                 (finger-code . below-string)
                 (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 5 1 3))

% 9
%
% C major for guitar, barred on third fret: Double barre
% used to test barre function, verbose style.
c'1^\markup
  % 110% of default size
```

```
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . arabic)
                  (dot-label-font-mag . 0.9)
                  (finger-code . in-dot)
                  (fret-label-font-mag . 0.6)
                  (fret-label-vertical-offset . 0)
                  (label-dir . -1)
                  (mute-string . "M")
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 4 2 5)
                           (barre 5 1 3))

% 10
%
% C major for guitar, with capo on third fret: verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-upper)
                  (dot-label-font-mag . 0.9)
                  (finger-code . none)
                  (fret-label-vertical-offset . 0.5)
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (capo 3)
                           (open 5)
                           (place-fret 4 5 1)
                           (place-fret 3 5 2)
                           (place-fret 2 5 3)
                           (open 1))

% 11
%
% Simple D chord.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
                  (dot-radius . 0.35)
                  (dot-position . 0.5)
                  (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
```

```
% 12
%
% Simple D chord, large top fret thickness.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
                  (dot-radius . 0.35)
                  (dot-position . 0.5)
                  (top-fret-thickness . 7)
                  (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"


%%
%% These chords will be in opposing-landscape orientation.
%%
\override TextScript.fret-diagram-details
            .orientation = #'opposing-landscape


% 13
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight
% barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-lower)
                  (finger-code . below-string)
                  (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 5 1 3))

% 14
%
% C major for guitar, barred on third fret: double barre
% used to test barre function, verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . arabic)
                  (dot-label-font-mag . 0.9)
                  (finger-code . in-dot)
                  (fret-label-font-mag . 0.6)
```

```
                  (fret-label-vertical-offset . 0)
                  (label-dir . -1)
                  (mute-string . "M")
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (place-fret 5 3 1)
                           (place-fret 4 5 2)
                           (place-fret 3 5 3)
                           (place-fret 2 5 4)
                           (place-fret 1 3 1)
                           (barre 4 2 5)
                           (barre 5 1 3))

% 15
%
% C major for guitar, with capo on third fret: verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
               . ((number-type . roman-upper)
                  (dot-label-font-mag . 0.9)
                  (finger-code . none)
                  (fret-label-vertical-offset . 0.5)
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3)))
  \fret-diagram-verbose #'((mute 6)
                           (capo 3)
                           (open 5)
                           (place-fret 4 5 1)
                           (place-fret 3 5 2)
                           (place-fret 2 5 3)
                           (open 1))

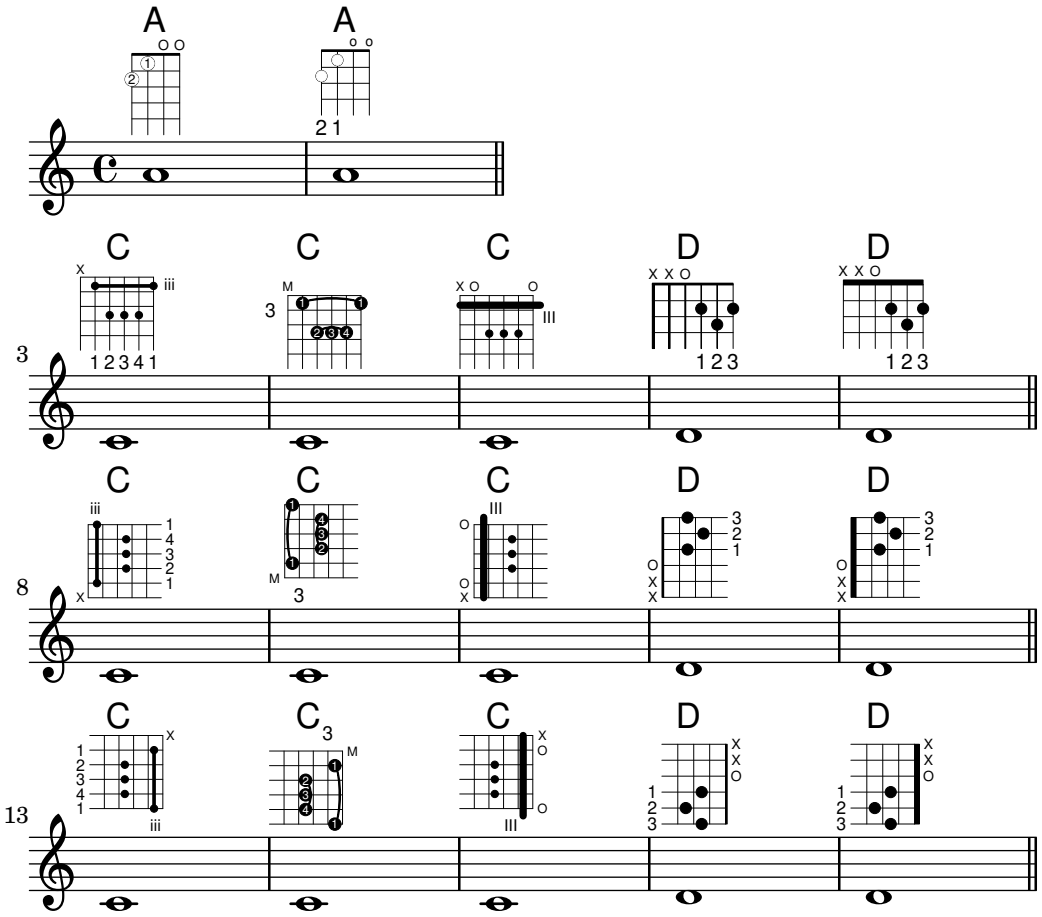
% 16
%
% Simple D chord.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
                  (dot-radius . 0.35)
                  (dot-position . 0.5)
                  (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 17
%
% Simple D chord, large top fret thickness.
d'1^\markup
  \override #'(fret-diagram-details
               . ((finger-code . below-string)
```

```
                        (dot-radius . 0.35)
                        (dot-position . 0.5)
                        (top-fret-thickness . 7)
                        (fret-count . 3)))
      \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
>>


\paper {
  ragged-right = ##t
  system-system-spacing.basic-distance = 20
}

\layout {
  \context {
    \Score
    \override SpacingSpanner.spacing-increment = 3
  }
}
```



Generate special note head shapes

When a note head with a special shape cannot easily be generated with graphic markup, a drawing specification for `ly:make-stencil` can be used to generate the shape. This snippet gives an example for a parallelogram-shaped note head.

Unfortunately, the available commands in a drawing specification are currently not documented (this is tracked in Issue #6874 (https://gitlab.com/lilypond/lilypond/-/issues/6874)); in any case, the used path sub-command has the following signature, quite similar to the make-path-stencil Scheme function.
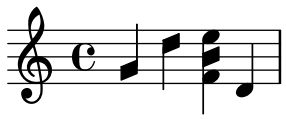
(path *thickness command-list line-cap-style line-join-style fill*)

The commands in *command-list* resemble PostScript drawing commands but with arguments after the command name.

```
parallelogram =
  #(ly:make-stencil
    '(path 0.1
           (rmoveto 0 0.25
            lineto 1.2 0.75
            lineto 1.2 -0.25
            lineto 0 -0.75
            lineto 0 0.25)
           round
           round
           #t)
    (cons -0.05 1.25)
    (cons -.75 .75))

myNoteHeads = \override NoteHead.stencil = \parallelogram
normalNoteHeads = \revert NoteHead.stencil

\relative c'' {
  \myNoteHeads
  g4 d'
  \normalNoteHeads
  <f, \tweak stencil \parallelogram b e>4 d
}
```



Generar corchetes personalizados

La propiedad stencil del grob Flag (el objeto gráfico corchete) se puede fijar a una función de Scheme personalizada que genere el glifo del corchete.

```
#(define-public (weight-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (log (- (ly:grob-property stem-grob 'duration-log) 2))
          (is-up? (eqv? (ly:grob-property stem-grob 'direction) UP))
          (yext (if is-up? (cons (* log -0.8) 0) (cons 0 (* log 0.8))))
          (flag-stencil (make-filled-box-stencil '(-0.4 . 0.4) yext))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stroke-stencil (if (equal? stroke-style "grace")
                              (make-line-stencil 0.2 -0.9 -0.4 0.9 -0.4)
                              empty-stencil)))
     (ly:stencil-add flag-stencil stroke-stencil)))
```

```
% Create a flag stencil by looking up the glyph from the font
#(define (inverted-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (dir (if (eqv? (ly:grob-property stem-grob 'direction) UP) "d" "u"))
          (flag (retrieve-glyph-flag "" dir "" grob))
          (line-thickness (ly:staff-symbol-line-thickness grob))
          (stem-thickness (ly:grob-property stem-grob 'thickness))
          (stem-width (* line-thickness stem-thickness))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stencil (if (null? stroke-style)
                       flag
                       (add-stroke-glyph flag stem-grob dir stroke-style "")))
          (rotated-flag (ly:stencil-rotate-absolute stencil 180 0 0)))
     (ly:stencil-translate rotated-flag (cons (- (/ stem-width 2)) 0))))

snippetexamplenotes =
{
  \autoBeamOff c'8 d'16 c'32 d'64 \acciaccatura {c'8} d'64
}

{
  \time 1/4
  <>^"Normal flags"
  \snippetexamplenotes

  <>_"Custom flag: inverted"
  \override Flag.stencil = #inverted-flag
  \snippetexamplenotes

  <>^"Custom flag: weight"
  \override Flag.stencil = #weight-flag
  \snippetexamplenotes
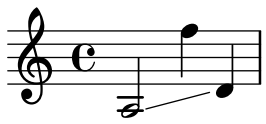
  <>_"Revert to normal"
  \revert Flag.stencil
  \snippetexamplenotes
}
```



Glissando por debajo de un objeto gráfico

Los objetos gráficos de columna de nota (los grobs `NoteColumn`) pueden ser sobrepasados por los glissandos.

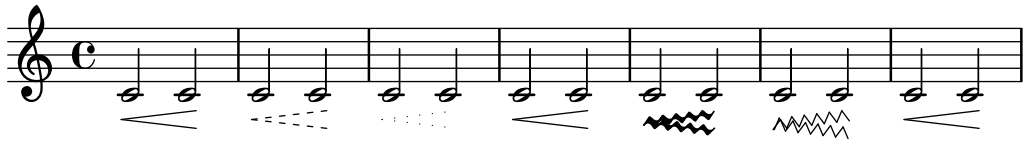
```
\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}
```



Reguladores con distintos estilos de línea

Los reguladores pueden imprimirse en uno cualquiera de los estilos de `line-interface`: `dashed-line` (discontinuo), `dotted-line` (punteado), `line` (línea continua), `trill` (trino) o zigzag (zig-zag).

```
\relative c' {
  c2\< c\!
  \override Hairpin.style = #'dashed-line
  c2\< c\!
  \override Hairpin.style = #'dotted-line
  c2\< c\!
  \override Hairpin.style = #'line
  c2\< c\!
  \override Hairpin.style = #'trill
  c2\< c\!
  \override Hairpin.style = #'zigzag
  c2\< c\!
  \revert Hairpin.style
  c2\< c\!
}
```



Horizontally aligning custom dynamics like “più f”

Some dynamic expressions involve additional text, like “sempre **pp**”. Since dynamics are usually centered under the note, the \pp would be displayed way after the note it applies to.

To correctly align the “sempre **pp**” horizontally so that it is aligned as if it were only the \pp, there are several approaches:

- Simply use `\once \override DynamicText.X-offset = #-9.2` before the note with the dynamics to manually shift it to the correct position. Drawback: This has to be done manually each time you use that dynamic markup...
- Add some padding (`#:hspace 7.1`) into the definition of your custom dynamic mark so that after LilyPond center-aligns it, it is already correctly aligned. Drawback: The padding really takes up that space and does not allow any other markup or dynamics to be shown in that position.
- Shift the dynamic script `\once \override ... .X-offset = ....` Drawback: `\once \override` is needed for every invocation!

- Set the dimensions of the additional text to 0 (using `#:with-dimensions '(0 . 0) '(0 . 0)`). Drawback: For LilyPond, "sempre" has no extent now. This means it might put other stuff there, causing collisions (which are not detected by LilyPond's collision detection algorithm!). There also seems to be some spacing, so it is not exactly the same alignment as without the additional text.
- Add an explicit shift directly inside the scheme function for the dynamic script.
- Set an explicit alignment inside the dynamic script. By default, this won't have any effect, only if one sets `X-offset`! Drawback: One needs to set `DynamicText.X-offset`, which will apply to all dynamic texts! Also, it is aligned at the right edge of the additional text, not at the center of \pp.

```
\paper {
  ragged-right = ##f
  indent = 5\cm
}

% Solution 1: Using a simple markup with a particular halign value
% Drawback: It's a markup, not a dynamic command, so \dynamicDown
%           etc. will have no effect
semppMarkup = \markup { \halign #1.4 \italic "sempre" \dynamic "pp" }

% Solution 2: Using a dynamic script & shifting with
%             \once \override ...X-offset = ..
% Drawback: \once \override needed for every invocation
semppK =
#(make-dynamic-script
  (markup #:line
          (#:normal-text
           #:italic "sempre"
           #:dynamic "pp")))

% Solution 3: Padding the dynamic script so the center-alignment
%             puts it at the correct position
% Drawback: the padding really reserves the space, nothing else can be there
semppT =
#(make-dynamic-script
  (markup #:line
          (#:normal-text
           #:italic "sempre"
           #:dynamic "pp"
           #:hspace 7.1)))

% Solution 4: Dynamic, setting the dimensions of the additional text to 0
% Drawback: To lilypond "sempre" has no extent, so it might put
%           other stuff there => collisions
% Drawback: Also, there seems to be some spacing, so it's not exactly the
%           same alignment as without the additional text
semppM =
#(make-dynamic-script
  (markup #:line
          (#:with-dimensions '(0 . 0) '(0 . 0)
                             #:right-align
```

```
                              #:normal-text
                              #:italic "sempre"
                              #:dynamic "pp")))

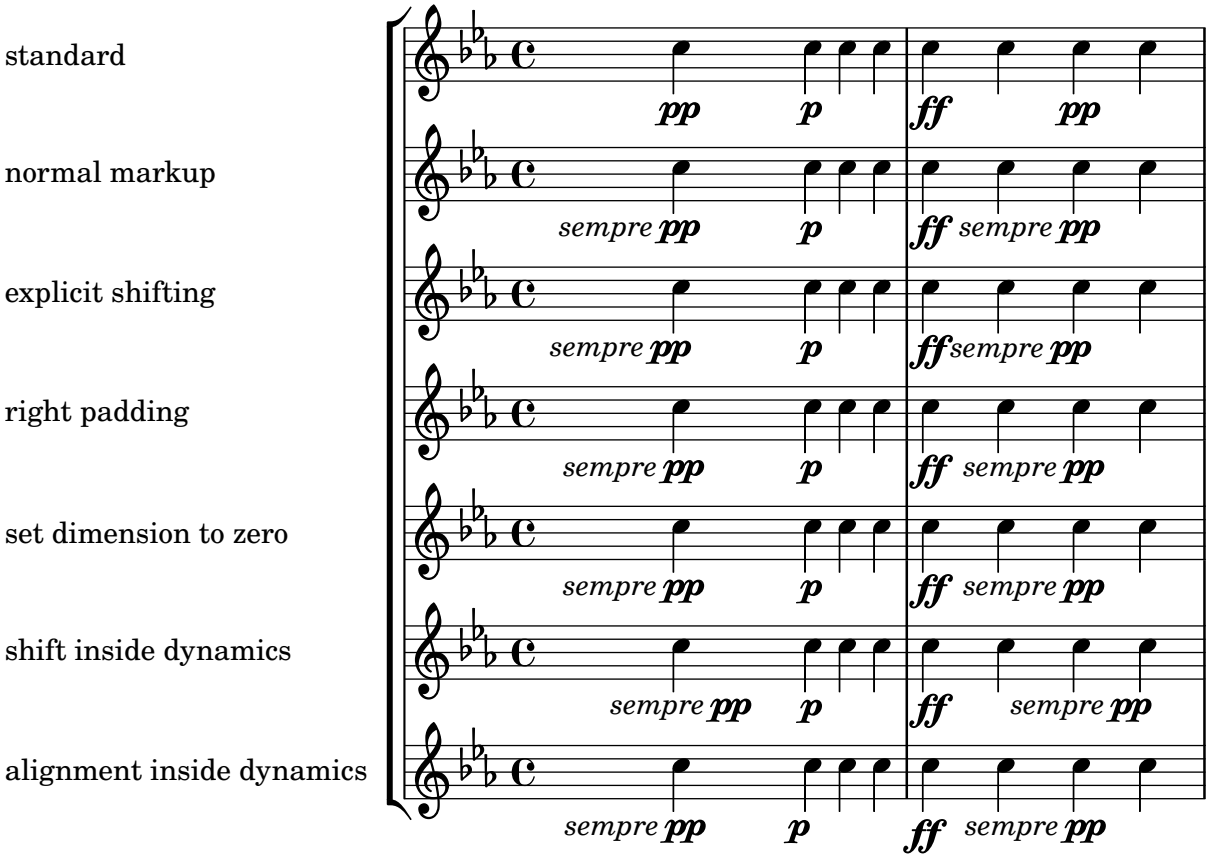
% Solution 5: Dynamic with explicit shifting inside the scheme function
semppG =
#(make-dynamic-script
  (markup #:hspace 0
          #:translate '(-18.85 . 0)
          #:line (#:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

% Solution 6: Dynamic with explicit alignment. This has only effect
%             if one sets X-offset!
% Drawback: One needs to set DynamicText.X-offset!
% Drawback: Aligned at the right edge of the additional text,
%           not at the center of pp
semppMII =
#(make-dynamic-script
  (markup #:line (#:right-align
                  #:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

\new StaffGroup <<
  \new Staff \with { instrumentName = "standard" }
    \relative c'' {
      \key es \major
      c4\pp c\p c c | c\ff c c\pp c
    }
  \new Staff \with {instrumentName = "normal markup" }
    \relative c'' {
      \key es \major
      c4-\semppMarkup c\p c c | c\ff c c-\semppMarkup c
    }
  \new Staff \with { instrumentName = "explicit shifting" }
    \relative c'' {
      \key es \major
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c\p c c
      c4\ff c
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c
    }
  \new Staff \with { instrumentName = "right padding" }
    \relative c'' {
      \key es \major
      c4\semppT c\p c c | c\ff c c\semppT c
    }
  \new Staff \with { instrumentName = "set dimension to zero" }
    \relative c'' {
```

```
      \key es \major
      c4\semppM c\p c c | c\ff c c\semppM c
    }
  \new Staff \with { instrumentName = "shift inside dynamics" }
    \relative c'' {
      \key es \major
      c4\semppG c\p c c | c\ff c c\semppG c
    }
  \new Staff \with { instrumentName = "alignment inside dynamics" }
    \relative c'' {
      \key es \major
      \override DynamicText.X-offset = #-1
      c4\semppMII c\p c c | c\ff c c\semppMII c
    }
>>

\layout { \override Staff.InstrumentName.self-alignment-X = #LEFT }
```



Cómo cambiar la posición de un diagrama de posiciones

Si queremos mover un diagrama de posiciones de acorde, por ejemplo, para evitar la colisión, o situarlo entre dos notas, tenemos varias posibilidades.

1) Modificar el valor de las propiedades `padding` (relleno) o `extra-offset` (desplazamiento adicional) como se muestra en la primera línea.

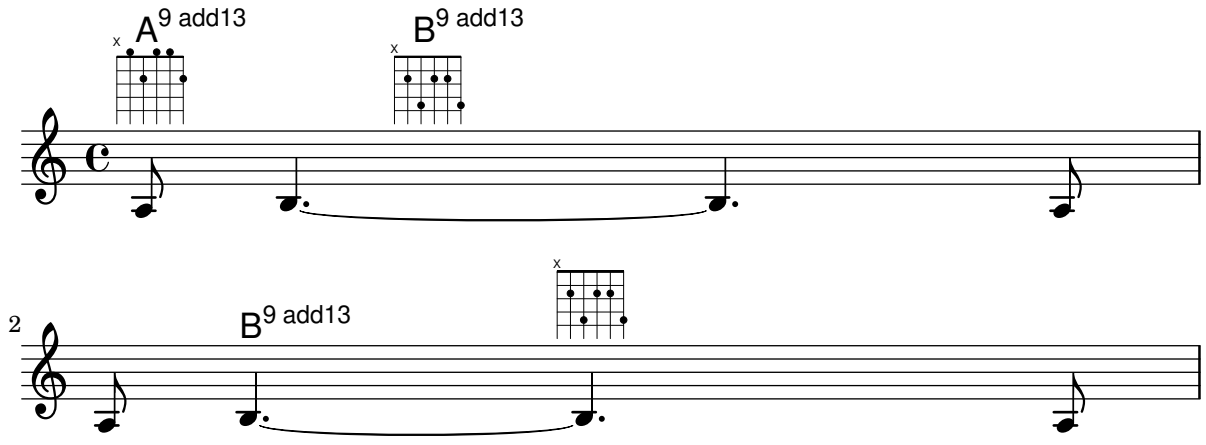
2) Podemos añadir una voz invisible y adjuntar los diagramas de posiciones a las notas invisibles de esa voz (como se ve en la segunda línea).

Si tenemos que mover el diagrama según una posición rítmica dentro del compás (en el ejemplo, la tercera parte del compás) es mejor el segundo ejemplo, porque el diagrama se alinea con el tercer pulso por sí solo.

```
harmonies = \chordmode
{
  a8:13
  \once \override ChordNames.ChordName.extra-offset = #'(10 . 0)
  b8:13 s4. |
  s2 b2:13
}

\score {
  <<
    \new ChordNames \harmonies
    \new Staff {
      % Method 1.
      a8^\markup \fret-diagram "6-x;5-0;4-2;3-0;2-0;1-2;"
      \once \override TextScript.extra-offset = #'(10 . 0)
      b4.~^\markup \fret-diagram "6-x;5-2;4-4;3-2;2-2;1-4;"
        b4. a8 | \break

      % Method 2.
      <<
        { a8 b4.~ b4. a8 }
        { s2 s2^\markup \fret-diagram  "6-x;5-2;4-4;3-2;2-2;1-4;" }
      >> |
    }
  >>
}
```
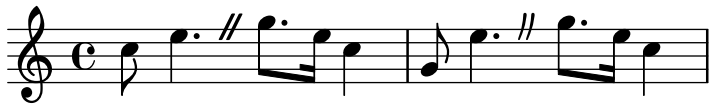


Insertar una cesura

Las marcas de cesura se pueden crear sobreescribiendo la propiedad `text` del objeto `BreathingSign`.

También está disponible una marca de cesura curva.

```
\relative c'' {
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4
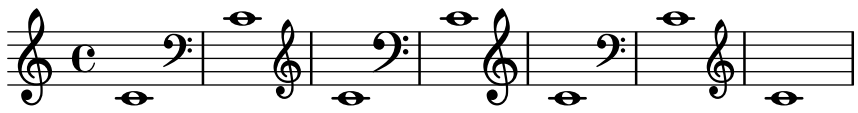
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```



Mantener el tamaño del símbolo en los cambios de clave

Cuando se produce un cambio de clave, el símbolo de clave se imprime a un tamaño menor que la clave inicial. Esto se puede ajustar estableciendo el valor de la propiedad de contexto `full-size-change` a `#t`.

```
\relative c' {
  \clef "treble"
  c1
  \clef "bass"
  c1
  \clef "treble"
  c1
  \override Staff.Clef.full-size-change = ##t
  \clef "bass"
  c1
  \clef "treble"
  c1
  \revert Staff.Clef.full-size-change
  \clef "bass"
  c1
  \clef "treble"
  c1
}
```



Puntas de flecha para las líneas

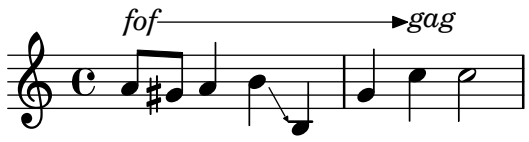
Se pueden aplicar puntas de flecha a los elementos de extensión de texto y de línea (como los glissandos).

```
\relative c'' {
  \override TextSpanner.bound-padding = #1.0
  \override TextSpanner.style = #'line
  \override TextSpanner.bound-details.right.arrow = ##t
  \override TextSpanner.bound-details.left.text = #"fof"
  \override TextSpanner.bound-details.right.text = #"gag"
  \override TextSpanner.bound-details.right.padding = #0.6

  \override TextSpanner.bound-details.right.stencil-align-dir-y = #CENTER
  \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER

  \override Glissando.bound-details.right.arrow = ##t
  \override Glissando.arrow-length = #0.5
  \override Glissando.arrow-width = #0.25

  a8\startTextSpan gis a4 b\glissando b,
  g'4 c\stopTextSpan c2
}
```



Making an object invisible using \hide
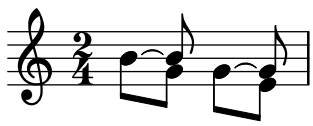
Applying \hide to a grob causes objects of this type to be printed with “invisible ink”. They are not printed, but all of their other behavior is retained:

- the objects still take up space,
- they take part in collision resolution, and
- slurs, ties, and beams can be attached to them as usual.

This snippet demonstrates how to connect different voices using ties. Normally, ties only connect two notes in the same voice. By introducing a tie in a different voice, and blanking the first up-stem in that voice, the tie appears to cross voices.

```
\relative {
  \time 2/4
  <<
    {
      \once \hide Stem
      \once \override Stem.length = #8
      b'8 ~ 8\noBeam
      \once \hide Stem
      \once \override Stem.length = #8
      g8 ~ 8\noBeam
    }
    \\
    {
      b8 g g e
    }
  >>
}
```

```
\paper {
  line-width = 40\mm
  ragged-right = ##f
}
```



Hacer que los glissandos se puedan dividir en el salto de línea

Normalmente, LilyPond rehúsa insertar un salto de línea automático en el punto en que un glissando cruza la barra de compás. Este comportamiento se puede cambiar estableciendo el valor de la propiedad `Glissando.breakable` a `#t`. También, estableciendo el valor de la propiedad `after-line-breaking` a `#t` hace que la línea del glissando continúe después del salto.

La propiedad `breakable` no afecta a los saltos manuales insertados con instrucciones como `\break`.
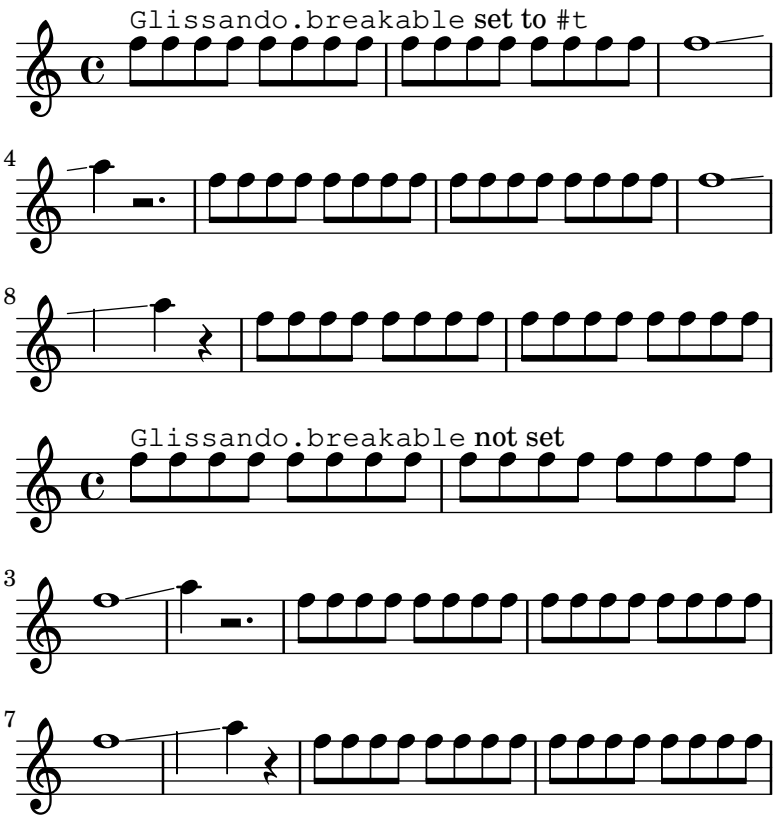
```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

music = {
  \*16 f8 |
  f1\glissando |
  a4 r2. |
  \*16 f8 |
  f1\glissando \once\glissandoSkipOn |
  a2 a4 r4 |
  \*16 f8
}

\relative c'' {
  <>^\markup { \typewriter Glissando.breakable
               set to \typewriter "#t" }
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  \music
}

\relative c'' {
  <>^\markup { \typewriter Glissando.breakable not set }
  \music
}

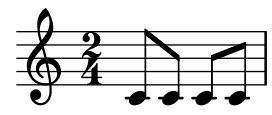
\paper {
  line-width = 100\mm
}
```



Controlar manualmente las posiciones de las barras

Se pueden controlar manualmente las posiciones de las barras de corchea sobreescribiendo el valor del parámetro `positions` del objeto gráfico `Beam`.

```
\relative c' {
  \time 2/4
  % from upper staff-line (position 2) to center (position 0)
  \override Beam.positions = #'(2 . 0)
  c8 c
  % from center to one above center (position 1)
  \override Beam.positions = #'(0 . 1)
  c8 c
}
```



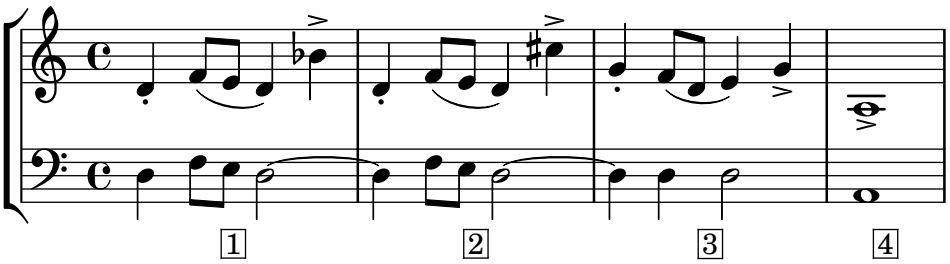
Número del compás centrado entre las barras

Para las partituras de bandas sonoras, es una convención común centrar los números de compás entre las líneas divisorias. Esto se consigue fijando la propiedad de contexto `centerBarNumbers` a `#t`. Cuando se usa esta sobreescritura, el tipo de objeto gráfico del número de compás es `CenteredBarNumber` y no `BarNumber`.

Este ejemplo muestra un número de ajustes: los números de compás centrados están encerrados en un ractángulo y se sitúan debajo de los pentagramas.

```
\layout {
  \context {
    \Score
    centerBarNumbers = ##t
    barNumberVisibility = #all-bar-numbers-visible
    \override CenteredBarNumber.stencil
      = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
    \override CenteredBarNumberLineSpanner.direction = #DOWN
  }
}

\new StaffGroup <<
  \new Staff \relative c' {
    d4-. f8( e d4) bes'-> |
    d,-. f8( e d4) cis'-> |
    g-. f8( d e4) g-> |
    a,1-> |
  }
  \new Staff \relative c {
    \clef bass
    d4 f8 e d2~ |
    4 f8 e d2~ |
    4 4 2 |
    a1 |
  }
>>
```



Disposición *Mensurstriche* (líneas divisorias entre pentagramas)

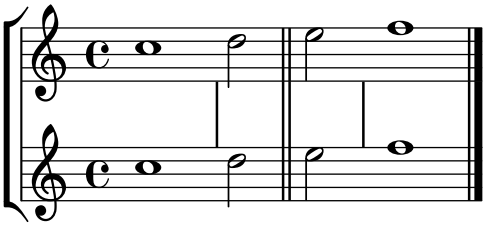
La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir fijando `measureBarType` al valor `"-span|"` y usando un contexto de agrupamiento que admita barras extendidas, como `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
```

```
}

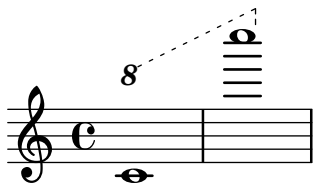
\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



Modificación de la inclinación de la línea de octava

Es posible cambiar la inclinación de la línea de octava alta o baja.
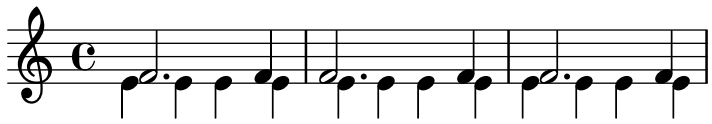
```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0)
                (attach-dir . ,LEFT)
                (padding . 0)
                (stencil-align-dir-y . ,CENTER)))
       (right . ((Y . 5.0) ; Change the number here
                 (padding . 0)
                 (attach-dir . ,RIGHT)
                 (text . ,(make-draw-dashed-line-markup
                           (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
     #ly:horizontal-line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
     #ly:horizontal-line-spanner::calc-right-bound-info
  \ottava 1
  c1
  c'''1
}
```



Desplazar las notas con puntillo en polifonía

Cuando se puede una nota en la voz superior para evitar la colisión con una nota de otra voz, el comportamiento predeterminado es desplazar la nota superior a la derecha. Se puede cambiar modificando la propiedad `prefer-dotted-right` del objeto gráfico `NoteCollision`.

```
\new Staff \relative c' <<
  {
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##f
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##t
    f2. f4
  }
  \\
  { e4 e e e e e e e e e e }
>>
```
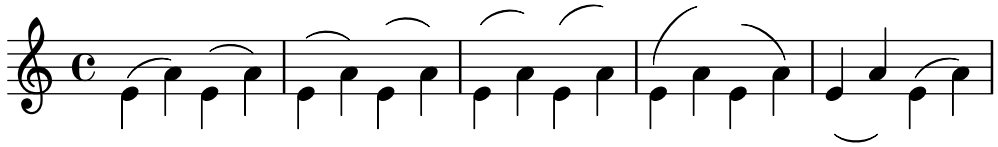


Desplazar ligaduras de expresión verticalmente

Se puede ajustar la posición vertical de una ligadura de expresión utilizando la propiedad `positions` del objeto `Slur`. La propiedad tiene dos parámetros, refiriéndose el primero al extremo izquierdo de la ligadura y el segundo al derecho. Los valores de los parámetros no se utilizan por parte de LilyPond para producir un desplazamiento exacto de la ligadura: más bien selecciona la colocación que mejor aspecto tiene, teniendo en cuenta los valores de los parámetros. Los valores positivos desplazan la ligadura hacia arriba, y son adecuados a notas que tienen las plicas hacia abajo. Los valores negativos desplazan las ligaduras bajas aún más hacia abajo.

Véase también el fragmento de código “Ajustar las posiciones de las ligaduras verticalmente”.

```
\relative c' {
  \stemDown
  e4( a)
  \override Slur.positions = #'(1 . 1)
  e4( a)
  \override Slur.positions = #'(2 . 2)
  e4( a)
  \override Slur.positions = #'(3 . 3)
  e4( a)
  \override Slur.positions = #'(4 . 4)
  e4( a)
  \override Slur.positions = #'(5 . 5)
  e4( a)
  \override Slur.positions = #'(0 . 5)
  e4( a)
  \override Slur.positions = #'(5 . 0)
  e4( a)
  \stemUp
  \override Slur.positions = #'(-5 . -5)
  e4( a)
  \stemDown
  \revert Slur.positions
  e4( a)
}
```

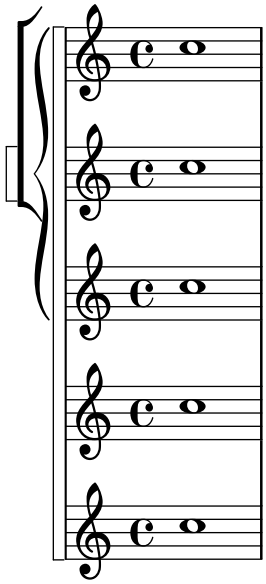


Anidado de grupos de pentagramas

Se puede utilizar la propiedad systemStartDelimiterHierarchy para crear grupos de pentagramas anidados de forma más compleja. La propiedad systemStartDelimiterHierarchy del contexto StaffGroup toma una lista alfabética del número de pentagramas producidos. Se puede proporcionar antes de cada pentagrama un delimitador de comienzo de sistema. Se debe encerrar entre corchetes y admite tantos pentagramas como encierren las llaves. Se pueden omitir los elementos de la lista, pero el primer corchete siempre abarca todos los pentagramas. Las posibilidades son SystemStartBar, SystemStartBracket, SystemStartBrace y SystemStartSquare.

```
\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = 4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare
          (SystemStartBrace
            (SystemStartBracket a
              (SystemStartSquare b))
            c)
          d)

  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Modificación de articulaciones por tipo

A veces queremos modificar un solo timpo de articulación. Aunque siempre es posible usar la instrucción \tweak, podría hacerse tedioso hacerlo para todos y cada uno de los símbolos de una partitura completa. El ejemplo muestra cómo trucar articulaciones con una lista de ajustes personalizados. Un caso sería la creación de una hoja de estilos.

```
#(define (custom-script-tweaks ls)
   (lambda (grob)
     (let* ((type (ly:event-property (ly:grob-property grob 'cause)
                                     'articulation-type))
            (tweaks (assoc-ref ls type)))
       (when tweaks
         (for-each
          (lambda (x) (ly:grob-set-property! grob (car x) (cdr x)))
          tweaks)))))

customScripts =
#(define-music-function (settings) (list?)
   #{
     \override Script.before-line-breaking =
       #(custom-script-tweaks settings)
   #})
revertCustomScripts = \revert Script.before-line-breaking


% Example

% Predefine two sets of desired tweaks.
#(define my-settings-1
   '((accent . ((font-size . 0)
                (color . (1 0 0))))
     (segno . ((font-size . 0)
               (color . (1 0 0))))
     (staccato . ((color . (1 0 0))
                  (padding . 0.5)))
     (staccatissimo . ((padding . 1)
                       (color . (1 0 0))))
     (tenuto . ((color . (1 0 0))
                (rotation . (45 0 0))
                (padding . 2)
                (font-size . 10)))
    ))

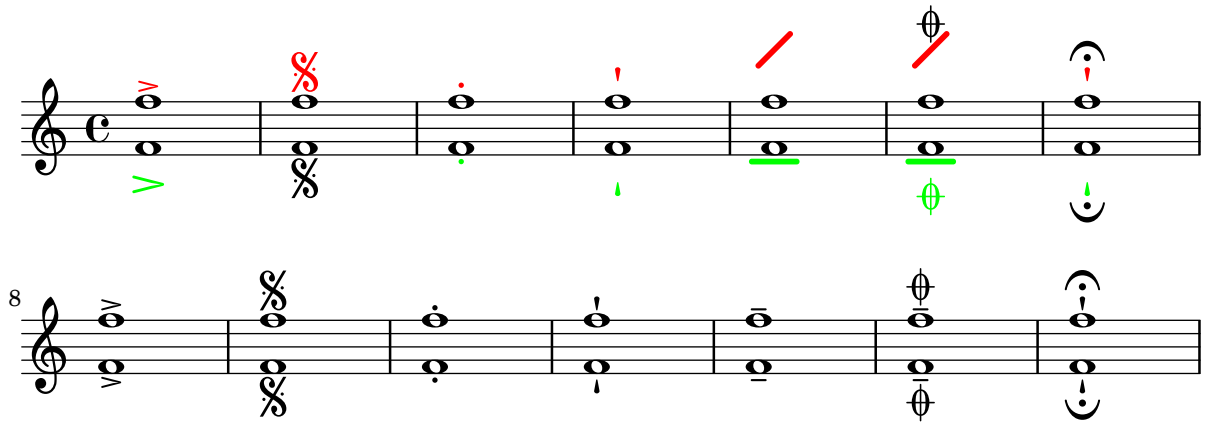
#(define my-settings-2
  '((accent . ((font-size . 4)
               (color . (0 1 0))
               (padding . 1.5)))
    (coda . ((color . (0 1 0))
             (padding . 1)))
    (staccato . ((color . (0 1 0))))
    (staccatissimo . ((padding . 2)
                      (color . (0 1 0))))
```

```
    (tenuto . ((color . (0 1 0))
               (font-size . 10)))
    ))

music = { f1-> | f\segno | f-. | f-! | f-- | f--\coda | f-!\fermata | }

block = {
  \music
  \break
  \revertCustomScripts \music
}

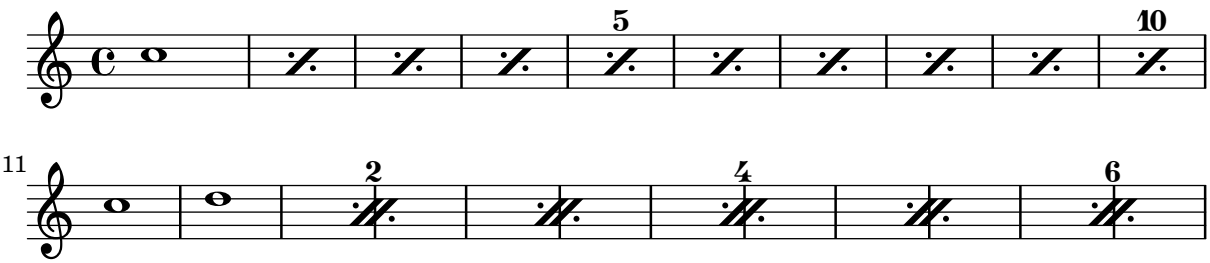
\new Staff <<
  \new Voice \with { \customScripts #my-settings-1 }
    \relative c''{ \voiceOne \block }
  \new Voice \with { \customScripts #my-settings-2 }
    \relative c' { \voiceTwo \block }
>>
```



Visibilidad del contador de repeticiones de tipo porcentaje

Se pueden mostrar los contadores de las repeticiones del tipo porcentaje a intervalos regulares mediante el establecimiento de la propiedad de contexto `repeatCountVisibility`.

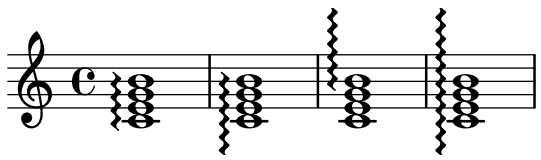
```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \%10 c1 \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \%6 { c1 d1 }
}
```



Posicionar símbolos de arpegio

Si necesitamos alargar o acortar un símbolo de arpegio, podemos modificar independientemente los extremos superior e inferior.

```
\relative c' {
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 0)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(0 . 5)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 5)
  <c e g b>1\arpeggio
}
```



Posicionamiento preciso de indicaciones de digitación

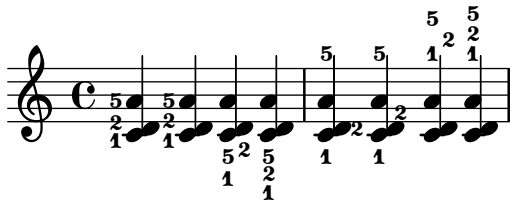
El posicionado semiautomático de las digitaciones dentro de un acorde funciona bien en casi todas las situaciones. Si una de las indicaciones precisa colocarse de forma más exacta, pueden usarse los trucos siguientes. Esto es especialmente útil para corregir el posicionado cuando están presentes intervalos de segunda.

```
\score {
  \relative c' {
    \set fingeringOrientations = #'(left)
    <c-1 d-2 a'-5>4
    <c-1 d-\tweak extra-offset #'(0 . 0.2)-2 a'-5>

    \set fingeringOrientations = #'(down)
    <c-1 d-2 a'-5>
    <c-\tweak extra-offset #'(0 . -1.1)-1
     d-\tweak extra-offset #'(-1.2 . -1.8)-2 a'-5> |

    \set fingeringOrientations = #'(down right up)
    <c-1 d-\tweak extra-offset #'(-0.3 . 0)-2 a'-5>4
    <c-1 d-\tweak extra-offset #'(-1 . 1.2)-2 a'-5>

    \set fingeringOrientations = #'(up)
    <c-1 d-\tweak extra-offset #'(0 . 1.1)-2
     a'-\tweak extra-offset #'(0 . 1)-5>
    <c-1 d-\tweak extra-offset #'(-1.2 . 1.5)-2
     a'-\tweak extra-offset #'(0 . 1.4)-5> |
  }
}
```



Posicionar los silencios multicompás

A diferencia de los silencios normales, no existe una instrucción predefinida para modificar la posición predeterminada de un símbolo de silencio multicompás sobre el pentagrama, adjuntándolo a una nota, independientemente de cuál sea su forma. Sin embargo, en la música polifónica los silencios multicompás de las voces de numeración par e impar están separados verticalmente.

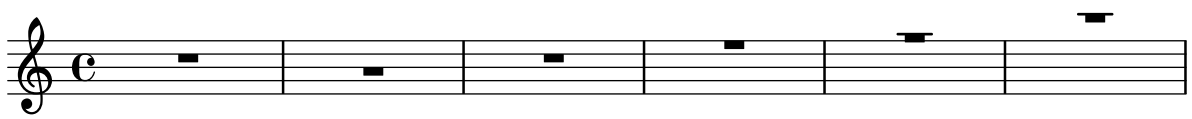
Este fragmento de código muestra la forma de controlar la colocación de los silencios multicompás.

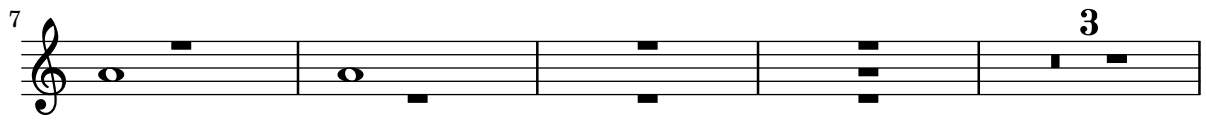
```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line.
  R1
  % They can be moved using an override or tweak.
  \tweak staff-position -2 R1
  \tweak staff-position 0 R1
  \tweak staff-position 2 R1
  \override MultiMeasureRest.staff-position = 3 R1
  \override MultiMeasureRest.staff-position = 6 R1
  \revert MultiMeasureRest.staff-position
  \break

  % Odd-numbered voices are under the top line.
  << { R1 } \\ { a1 } >>
  % Even-numbered voices are under the bottom line.
  << { a1 } \\ { R1 } >>
  % Multi-measure rests in both voices remain separate.
  << { R1 } \\ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override or tweak.
  << { R1 } \\ { R1 } \\ { \tweak staff-position -2 R1 } >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed.
  \compressMMRests
  <<
    \revert MultiMeasureRest.direction
    { R1*3 } \\
    \revert MultiMeasureRest.direction
    { R1*3 }
  >>
}
```
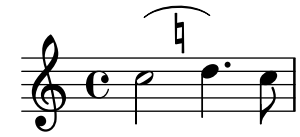




Situar los elementos de marcado de texto por dentro de las ligaduras

Los elementos de marcado de texto deben tener la propiedad `outside-staff-priority` establecida a `#f` para que se impriman por dentro de las ligaduras de expresión.

```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```



Imprimir números de compás dentro de rectángulos o circunferencias

Los números de compás también se pueden imprimir dentro de rectángulos o de circunferencias.

```
\relative c' {
  % Center bar numbers except at the beginning of a staff.
  \override Score.BarNumber.self-alignment-X =
    #(break-alignment-list CENTER CENTER 0.3)

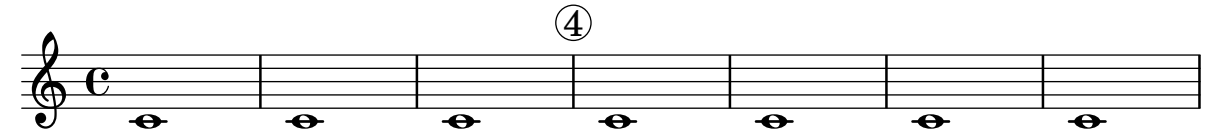
  % Prevent bar numbers at the end of a line and permit them elsewhere.
  \override Score.BarNumber.break-visibility = #end-of-line-invisible

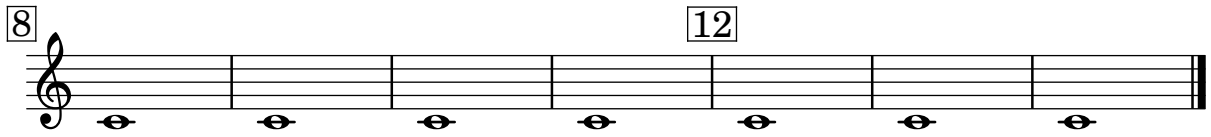
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2.
  \override Score.BarNumber.font-size = 2

  % Draw a circle round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \*7 c1 \break

  % Draw a box round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \*7 c1 \bar "|."
}
```





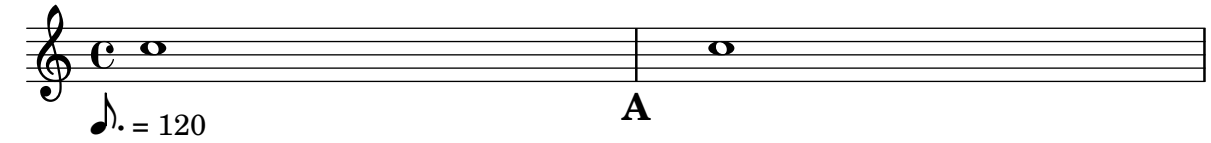
Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama

De forma predeterminada, las indicaciones metronómicas y las letras de ensayo se imprimen encima del pentagrama. Para colocarlas debajo del pentagrama, simplemente ajustamos adecuadamente la propiedad `direction` de `MetronomeMark` o de `RehearsalMark`.

```
\layout {
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Impresión de los nombres de las notas con o sin indicación de la octava
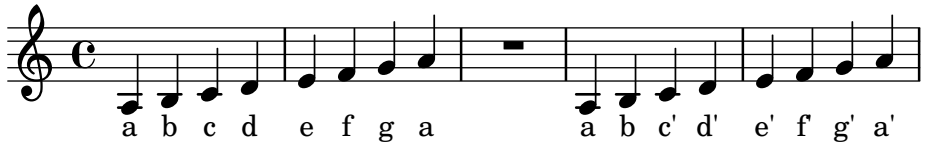
Se puede usar el contexto `NoteNames` para imprimir el valor textual de las notas. La propiedad `printOctaveNames` activa o desactiva la representación de la octava de las notas.

```
scale = \relative c' {
  a4 b c d
  e4 f g a
}

\new Staff {
  <<
    \scale
    \context NoteNames {
      \set printOctaveNames = ##f
      \scale
    }
  >>
  R1
  <<
```

```
    \scale
    \context NoteNames {
      \set printOctaveNames = ##t
      \scale
    }
  >>
}

\layout {
  \context {
    \NoteNames
    % Allow vertical overlapping of different `NoteNames` contexts
    % to make them appear as if they were a single line.
    \override VerticalAxisGroup
              .nonstaff-nonstaff-spacing
              .minimum-distance = ##f
  }
}
```



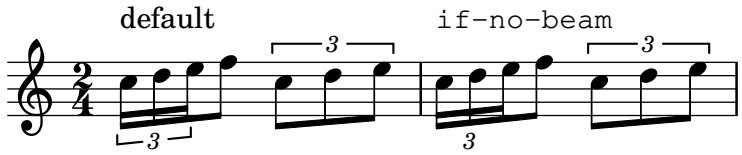
Imrpesión de corchetes de tresillo en el lado de la cabeza de la nota

Cualquiera que sea la opción elegida para el control de la visibilidad del corchete de los tresillos, mostrará u ocultará el corchete independientemenre de su ubicación (en el lado de la plica o en el lado de la cabeza). Sin embargo, cuando se coloca el corchete del lado de la cabeza, algunos autores recomiendan imprimir siempre el corchete de tresillo. Se puede usar la opción `visible-over-note-heads` para obtener esto.

```
music = \relative c'' {
  \tupletNeutral \tuplet 3/2 { c16[ d e } f8]
  \tupletUp \tuplet 3/2 { c8 d e }
}

\new Voice {
  \relative c' {
    \override TextScript.staff-padding = #2.5

    \time 2/4
    \override TupletBracket.visible-over-note-heads = ##t
    \override Score.TextMark.non-musical = ##f
    <>^\markup "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    <>^\markup \typewriter "if-no-beam" \music
  }
}
```

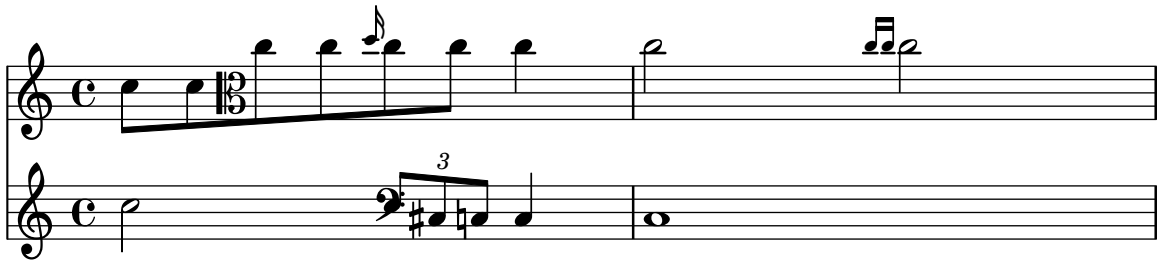


Espaciado de las notas estrictamente proporcional

Si está establecida al valor `#t` la propiedad `strict-note-spacing` del objeto gráfico `SpacingSpanner`, el espaciado de las notas no queda influido por los compases o claves que pueda haber dentro de un sistema. En lugar de ello, se colocan justo antes de la nota que tiene lugar en el mismo momento temporal. Esto puede producir colisiones.

```
\relative c'' <<
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  \set Score.proportionalNotationDuration = #1/16

  \new Staff {
    c8[ c \clef alto c c \grace { d16 } c8 c] c4
    c2 \grace { c16[ c16] } c2
  }
  \new Staff {
    c2 \tuplet 3/2 { c8 \clef bass cis,, c } c4
    c1
  }
>>
```



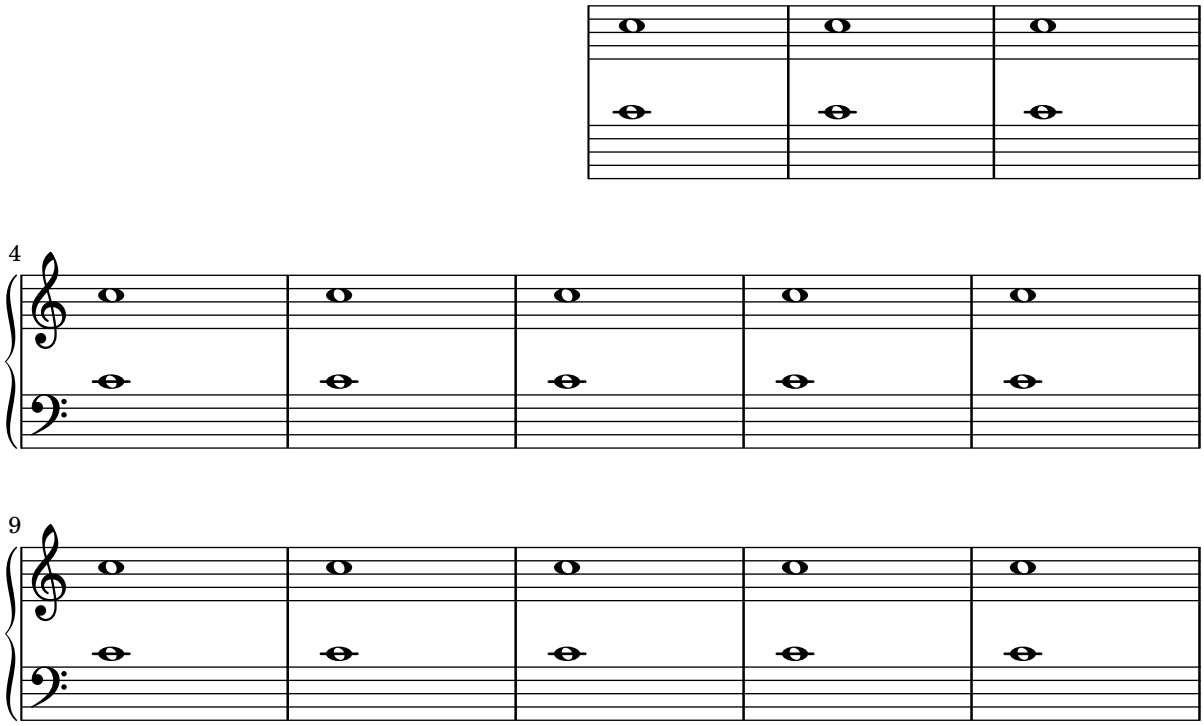
Quitar la llave en el primer sistema de una partitura de piano

Este fragmento elimina la primera llave de un `PianoStaff` o un `GrandStaff`, junto con las claves. Puede ser útil cuando se está cortando y pegando la imagen de la partitura editada dentro de otra música existente.

El código emplea `\alterBroken` para ocultar el delimitador en forma de llave que aparece al principio.

```
someMusic =  {
  \once \omit Staff.Clef
  \once \omit Staff.TimeSignature
  \*3 c1 \break
  \*5 c1 \break
  \*5 c1
}

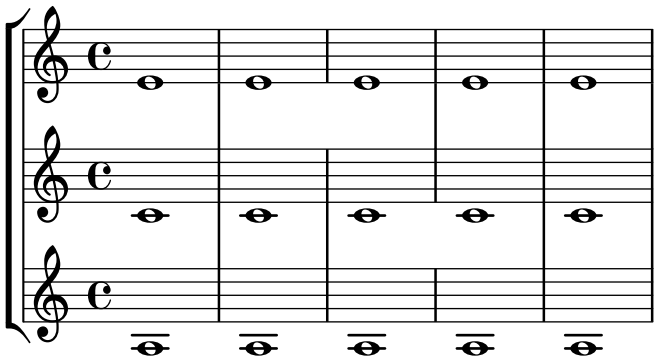
\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c'' \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
    indent=75\mm
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}
```



Quitar las barras de compás entre los pentagramas de un StaffGroup, PianoStaff o GrandStaff

De forma predeterminada, las líneas divisorias en los contextos `StaffGroup`, `PianoStaff` o `GrandStaff` se conectan entre los pentagramas mediante una línea. Se puede alterar este comportamiento pentagrama a pentagrama.

```
\relative c' {
  \new StaffGroup <<
    \new Staff {
      e1 | e
      \once \override Staff.BarLine.allow-span-bar = ##f
      e1 | e | e
    }
    \new Staff {
      c1 | c | c
      \once \override Staff.BarLine.allow-span-bar = ##f
      c1 | c
    }
    \new Staff {
      a1 | a | a | a | a
    }
  >>
}
```



Quitar la primera línea vacía

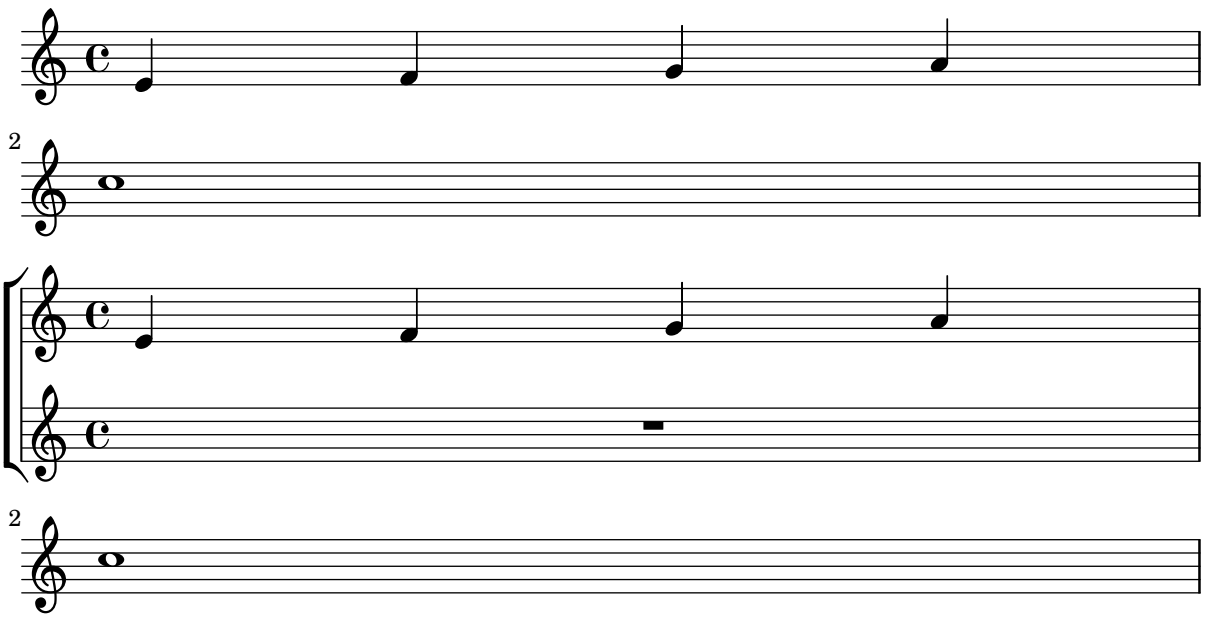
Para eliminar el primer pentagrama vacío de una aprtitura, establecemos la propiedad `remove-first` del objeto gráfico `VerticalAxisGroup` al valor `#t`. Esto se puede hacer globalmente dentro del bloque `\layout`, o localmente dentro del pentagrama concreto que se quiere suprimir. En este último caso, tenemos que especificar el contexto (`Staff` se aplica solo al pentagrama actual) delante de la propiedad.

El pentagrama inferior del segundo grupo no se elimina, porque el ajuste solo se aplica al pentagrama concreto dentro del que se escribe.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup.remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
```

```
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>

\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
>>
```



Estilos de silencios

Los silencios se pueden imprimir en distintos estilos.

```
restsA = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
  s64 s128 s256 s512 s1024 s1024
}
restsB = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
  r64 r128 r256 r512 r1024 s1024
}
```

```
\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

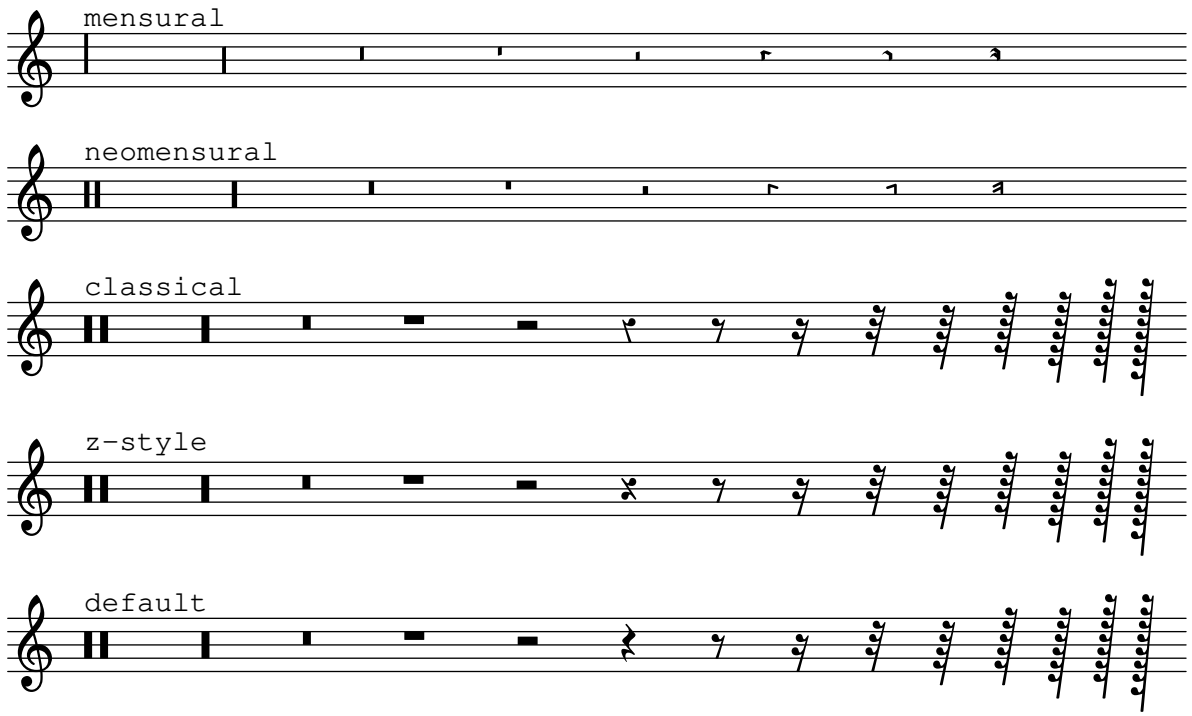
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>^\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>^\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>^\markup \typewriter { default } \restsB \bar "" \break
}
```



Barras rítmicas

En las hojas guía de acordes o *lead-sheets* "sencillas", a veces no se imprime realmente ninguna nota. En su lugar se hace una notación que tiene solamente "patrones rítmicos" y acordes encima de los compases para representar la estructura de la canción. Tal funcionalidad puede ser útil al crear o transcribir la estructura de una canción, o si se quieren compartir las hojas guía con guitarristas o músicos de jazz.

```
startPat = {
  \improvisationOn
  \omit Stem
}
stopPat = {
  \improvisationOff
  \undo \omit Stem
}

\new Voice \with {
  \consists Pitch_squash_engraver
} {
  c'4 d' e' f' |
  \startPat
  4 4 4 4 |
  \stopPat
  f'4 e' d' c'
}
```



Separar las cancelaciones de tonalidad de los cambios de armadura

De forma predeterminada, las alteraciones accidentales que se usan para las cancelaciones en las armaduras se colocan adyacentes a las que se usan para los cambios de tonalidad. Este comportamiento se puede cambiar sobreescribiendo la propiedad `break-align-orders` del objeto gráfico `BreakAlignment`.

Si busca la definición de `break-alignment-interface` en la Referencia de funcionamiento interno (`https://lilypond.org/doc/v2.24/Documentation/internals/break_002dalignment_002dinterface`) de LilyPond, obtendrá la siguiente lista de los símbolos de break-align disponibles:

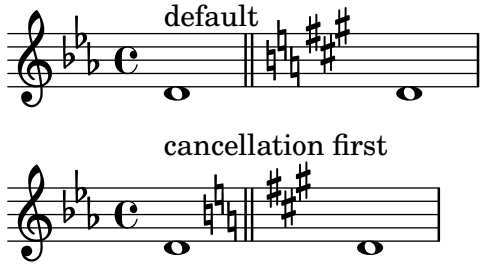
```
ambitus
breathing-sign
clef
cue-clef
cue-end-clef
custos
key-cancellation
key-signature
left-edge
signum-repetitionis
staff-bar
staff-ellipsis
time-signature
```

A partir de esta lista, vemos que lo que queremos hacer es trasladar la `key-cancellation` de forma que esté antes del `staff-bar`. Esto se hace con la función `\breakAlignInsert`.

```
music = { \key es \major d'1 \bar "||"
          \key a \major d'1 }

{ <>^\markup "default"
  \music }

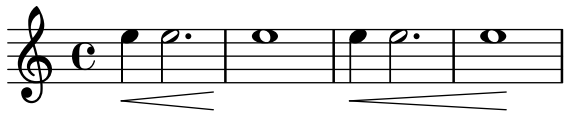
{ <>^\markup "cancellation first"
  \breakAlignInsert key-cancellation before staff-bar
  \music }
```



Establecer el comportamiento de los reguladores en las barras de compás

Si la nota que da fin a un regulador cae sobre la primera parte de un compás, el regulador se detiene en la línea divisoria inmediatamente precedente. Se puede controlar este comportamiento sobreescribiendo la propiedad `to-barline`.

```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



Fijar un separador entre los sistemas

Se pueden insertar separadores sistema entre los sistemas de una página. Se puede usar cualquier elemento de marcado, pero `\slashSeparator` está disponible como una elección predeterminada adecuada.

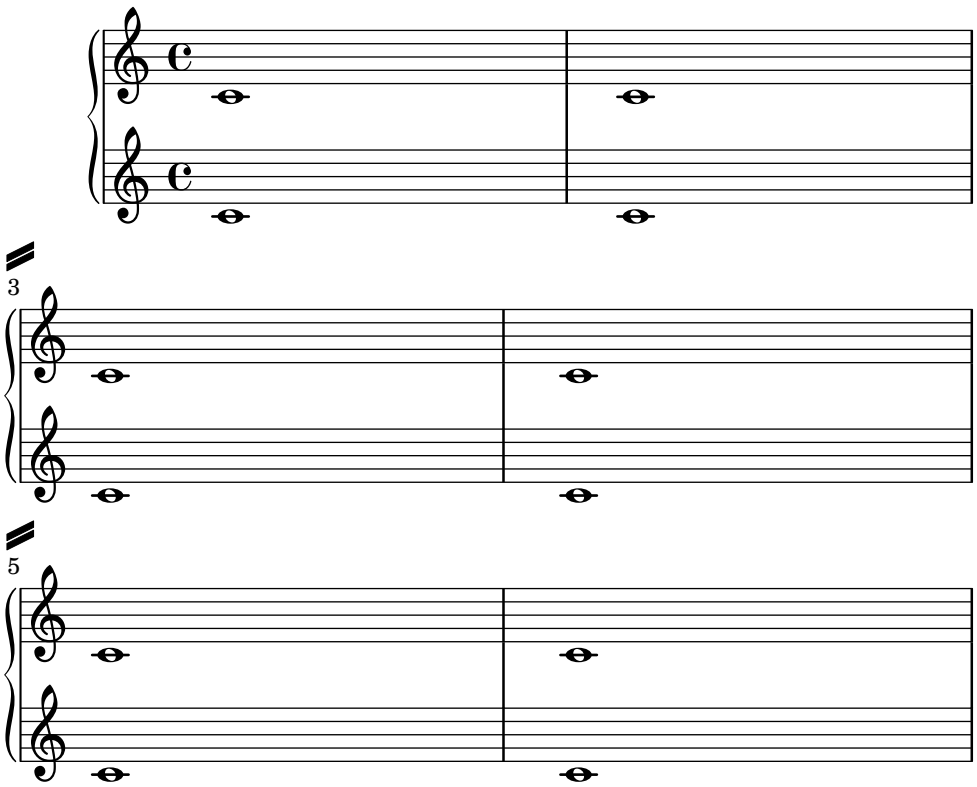
```
#(set-default-paper-size "a5")

\paper {
  system-separator-markup = \slashSeparator
  tagline = ##f
}

notes = \relative c' {
  c1 | c \break
  c1 | c \break
  c1 | c
```

```
}

\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}
```



Dar forma a ligaduras de unión individuales entre acordes

Para dar forma a las ligaduras de unión individuales entre acordes use el método que se muestra más abajo.

```
{
  \textMark "Chords can be tied note by note."
  <c'~ e'~ g'~ c''~>2 q
}

{
  \textMark \markup \override #'(baseline-skip . 3) \wordwrap {
    Modifying those ties with \typewriter "\\shape" does not succeed,
    because \typewriter TieColumn positions them on its own behalf,
    ignoring \typewriter "\\shape" input more or less.  You may
    circumvent this by setting \typewriter positioning-done to
    \typewriter "#t" -- alas, \typewriter positioning-done is an
    internal property, and setting it to \typewriter "#t" means: all
    positioning is done, don't do anything further.  The next example
```

```
    demonstrates a case where the positioning is not finished: all tie
    directions are down, and the thickness is not accurate.
  }
  <c'~ e'~ g'~ c''~>2
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark "To fix that, enter ties with explicit direction modifiers."
  <c'_~ e'_~ g'_~ c''^~>2
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark \markup {
    Now you can use \typewriter "\\shape" for each tie as usual. }
  <c'-\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
   e'-\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
   g'-\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
   c''-\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
  >2
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark "This also works at line breaks."
  <c'-\shape #'(((0 . 0) (0 . -10) (0 . -10) (0 . 0))
                ((0 . 0) (0 . -10) (0 . -10) (0 . 0))) _~
   e'-\shape #'(((0 . 0) (0 . -5) (0 . -5) (0 . 0))
                ((0 . 0) (0 . -5) (0 . -5) (0 . 0))) _~
   g'-\shape #'(((0 . 0) (0 . -2) (0 . -2) (0 . 0))
                ((0 . 0) (0 . -2) (0 . -2) (0 . 0))) _~
   c''-\shape #'(((0 . 0) (0 . 5) (0 . 5) (0 . 0))
                 ((0 . 0) (0 . 5) (0 . 5) (0 . 0))) ^~
  >2
  \break
  \once \override TieColumn.positioning-done = ##t
  q
}

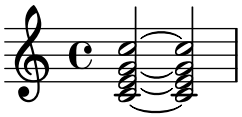
{
  \textMark \markup {
    It also works with the \typewriter tieWaitForNote property. }
  \set tieWaitForNote = ##t
  c'4-\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
  e'-\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
  g'-\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
  c''-\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
  \once \override TieColumn.positioning-done = ##t
```

```
  <c' e' g' c''>1
}

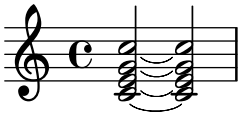
\layout {
  indent = 0
  \context {
    \Score
    \override TextMark.padding = #4
    \override TextMark.break-align-symbols = #'(left-edge)
  }
}

\paper {
  score-system-spacing.padding = 3
}
```

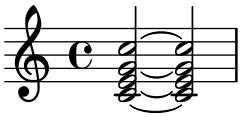
Chords can be tied note by note.



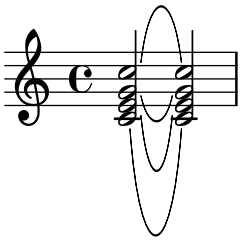
Modifying those ties with `\shape` does not succeed, because `TieColumn` positions them on its own behalf, ignoring `\shape` input more or less. You may circumvent this by setting `positioning-done` to `#t` – alas, `positioning-done` is an internal property, and setting it to `#t` means: all positioning is done, don't do anything further. The next example demonstrates a case where the positioning is not finished: all tie directions are down, and the thickness is not accurate.



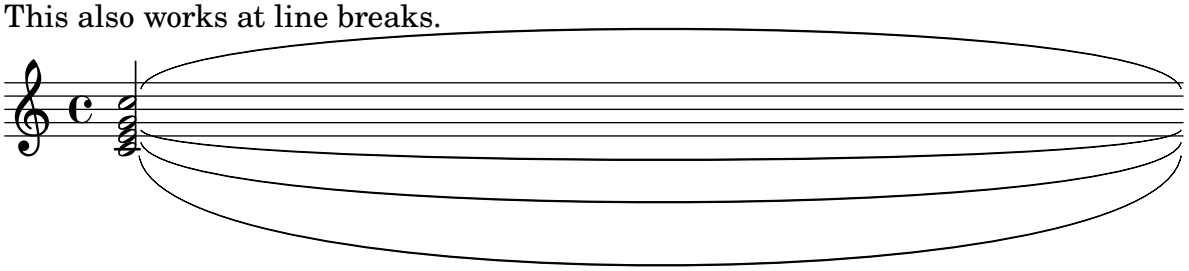
To fix that, enter ties with explicit direction modifiers.

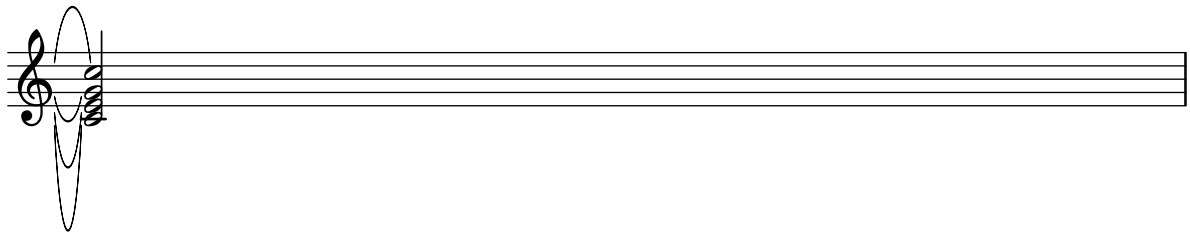


Now you can use `\shape` for each tie as usual.

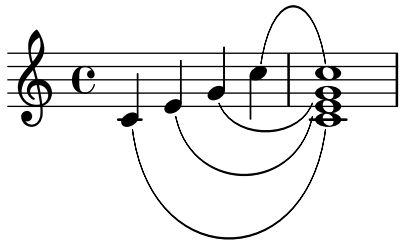


This also works at line breaks.





It also works with the `tieWaitForNote` property.



Imprimir la misma articulación encima y debajo de la misma nota o acorde

De forma predeterminada, LilyPond no permite poner la misma articulación (un acento, un calderón, un círculo de armónico, etc.) encima y debajo de la nota. Por ejemplo, `c4_\fermata^\fermata` imprime solamente el calderón inferior. El calderón superior sencillamente se ignora.

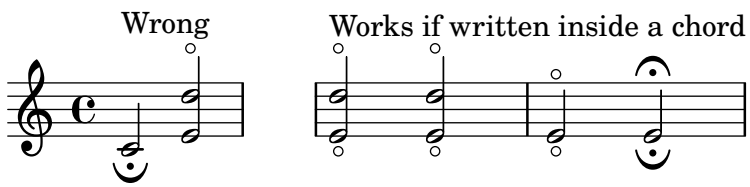
Sin embargo, se pueden adosar inscripciones (de igual forma que las digitaciones) dentro de un acorde, lo que significa que es posible tener tantas articulaciones como se desee. Este enfoque tiene la ventaja de que ignora la plica y posiciona la articulación de forma relativa a la cabeza de la nota. Puede verse esto en el caso de los flageolets (indicaciones de armónico) que aparecen en el fragmento de código. Para recrear el comportamiento de las inscripciones fuera del acorde, se requiere `add-stem-support`.

La solución consiste en escribir la nota como un acorde y añadir las articulaciones dentro de los paréntesis en ángulo `<...>`, usando los modificadores de dirección `^` y `_` según corresponda.

```
\relative c' {
  <>^"Wrong"
  c2_\fermata^\fermata % The second fermata is ignored!
  <e d'>2^\flageolet_\flageolet

  \stopStaff s1 \startStaff

  <>^"Works if written inside a chord"
  <e_\flageolet d'^\flageolet>2
  <e_\flageolet d'^\flageolet>2
  <e_\flageolet^\flageolet>2
  <e_\fermata^\fermata>2
}
```
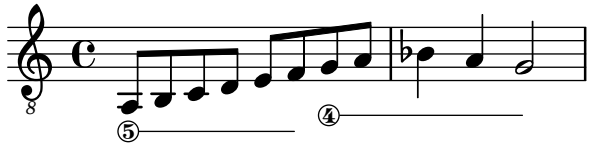


Líneas de extensión para números de cuerda

Hacemos una línea extensora para las indicaciones de número de cuerda, motrando que una serie de notas se han de tocar sobre la misma cuerda.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})

\relative c {
  \clef "treble_8"
  \textSpannerDown
  \stringNumberSpanner "5" a8\startTextSpan b c d
    e f\stopTextSpan \stringNumberSpanner "4" g\startTextSpan a |
  bes4 a g2\stopTextSpan
}
```
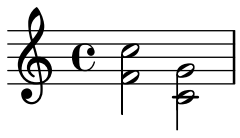


Evitar las advertencias sobre columnas de notas que chocan

Si se colocan sobre la misma posición notas de dos voces distintas con las plicas en la misma dirección, y ninguna de las voces tiene un desplazamiento o ambas tienen el mismo desplazamiento, aparece el mensaje de error "advertencia: demasiadas columnas de notas en colisión; se ignora" aparece al compilar el archivo de LilyPond. Este mensaje se puede evitar fijando la propiedad `ignore-collision` del objeto `NoteColumn` al valor `#t`. Observe que esto no elimina solamente las advertencias, sino que hace que LilyPond deje de intentar resolver las colisiones en absoluto, por lo que pueden obtenerse resultados distintos de los esperados si no se usa con prudencia.

```
ignore = \override NoteColumn.ignore-collision = ##t
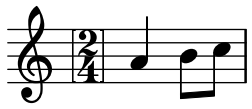
\relative c' {
  \new Staff <<
    \new Voice { \ignore \stemDown f2 g }
    \new Voice { c2 \stemDown c, }
  >>
}
```



Indicación de compás entre corchetes

La indicación de compás se puede encerrar entre corchetes.
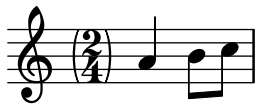
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Compás entre paréntesis

Se puede encerrar la indicación de compás entre paréntesis.
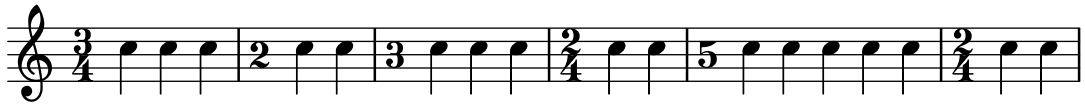
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Indicación de compás imprimiendo solo el numerador (en lugar de la fracción)

A veces, la indicación de compás no debe imprimir la fracción completa (p.ej. 7/4), sino solo el numerador (el dígito 7 en este caso). Esto se puede hacer fácilmente utilizando `\override Staff.TimeSignature.style = #'single-number` para cambiar el estilo permanentemente. Usando `\revert Staff.TimeSignature.style`, se puede revertir el cambio. Para aplicar el estilo de un número único a una sola indicación de compás, use `\tweak`.

```
\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-number
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-number style only for the next time signature
  \tweak style #'single-number \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```
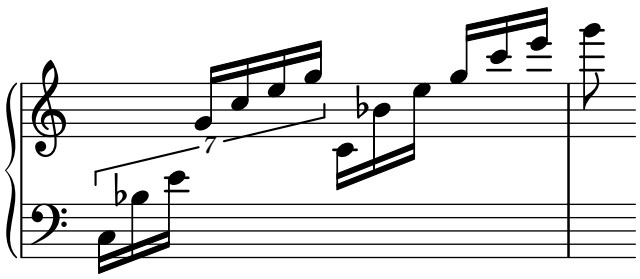


Corchete de tresillo y cambio de pentagrama

Este fragmento muestra cómo preparar un tresillo que se inicia en un pentagrama inferior y termina en el superior.

```
aigues = \relative c' {
  \time 6/8
  s4. \stemDown c16[ bes' e] \stemUp g c e \stemDown |
  g8
}

basses = \relative c {
  \time 3/4
  \clef F
  \tweak positions #'(4.5 . 8.5)
    \tweak edge-height #'(1 . -1)
    \tuplet 7/6 { c16[ bes' e] \change Staff = md \stemUp g[ c e g] } s4. |
  s8
}

\new PianoStaff \with { \omit TimeSignature }
<<
  \new Staff = md \aigues
  \new Staff = mg \basses
>>
```



Trucaje de las propiedades de clave

La modificación del glifo de la clave, su posición o su octavación, no cambian 'per se' la posición de las siguientes notas del pentagrama. Para conseguir armaduras de tonalidad sobre las líneas del pentagrama adecuadas, también debe especificarse `middleCClefPosition`, con valores positivos o negativos que mueven el "Do central" hacia arriba o hacia abajo, respectivamente, en relación con la línea central del pentagrama (usualmente la tercera).

Por ejemplo, la instrucción `\clef "treble_8"` equivale a un ajuste de las propiedades de contexto `clefGlyph`, `clefPosition` (que controla la posición vertical de la clave sobre el pentagrama), `middleCPosition` y `clefTransposition`. Se imprime una clave cada vez que se modifica cualquiera de las propiedades excepto `middleCPosition`.

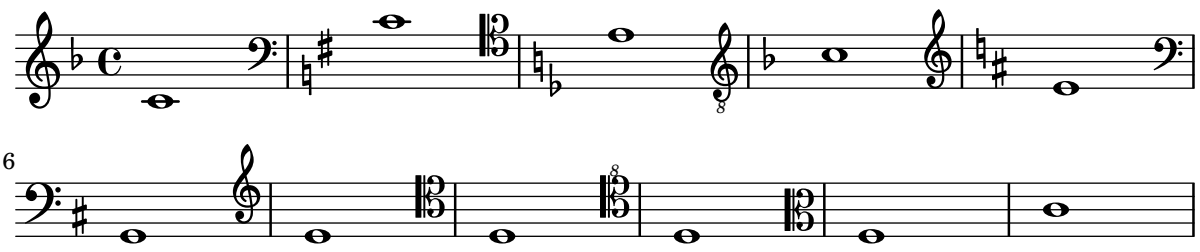
Los siguientes ejemplos muestran las posibilidades cuando se ajustan estas propiedades manualmente. En la primera línea, los cambios manuales preservan el posicionamiento relativo estándar de las claves y las notas, pero no lo hacen en la segunda línea.

```
{
  % The default treble clef.
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  \set Staff.middleCPosition = 6
  \set Staff.middleCClefPosition = 6
  \key g \major
  c'1
  % The baritone clef.
  \set Staff.clefGlyph = "clefs.C"
  \set Staff.clefPosition = 4
  \set Staff.middleCPosition = 4
  \set Staff.middleCClefPosition = 4
  \key f \major
  c'1
  % The standard choral tenor clef.
  \set Staff.clefGlyph = "clefs.G"
  \set Staff.clefPosition = -2
  \set Staff.clefTransposition = -7
  \set Staff.middleCPosition = 1
  \set Staff.middleCClefPosition = 1
  \key f \major
  c'1
  % A non-standard clef.
  \set Staff.clefPosition = 0
  \set Staff.clefTransposition = 0
  \set Staff.middleCPosition = -4
  \set Staff.middleCClefPosition = -4
  \key g \major
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
  % and clefs.
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  c'1
  \set Staff.clefGlyph = "clefs.G"
  c'1
  \set Staff.clefGlyph = "clefs.C"
  c'1
  \set Staff.clefTransposition = 7
  c'1
  \set Staff.clefTransposition = 0
  \set Staff.clefPosition = 0
  c'1

  % Return to the normal clef.
  \set Staff.middleCPosition = 0
```

```
  c'1
}
```
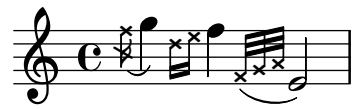


Trucar la disposición de las notas de adorno dentro de la música

La disposición de las expresiones de adorno se puede cambiar usando las funciones `add-grace-property` y `remove-grace-property`.

El ejemplo siguiente borra la definición de la propiedad `direction` (dirección) de los objetos gráficos `Stem` (plicas) para esta nota de adorno, de manera que las plicas no siemmpre apuntan hacia arriba, y cambia la forma predeterminada de las cabezas a aspas.

```
\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
       \acciaccatura { f16 } g4
       \grace { d16 e } f4
       \appoggiatura { f,32 g a } e2
    }
  }
}
```



Uso de estilos alternativos para los corchetes

Se pueden imprimir formas alternativos del corchete o gancho de las corcheas y figuras menores, mediante la sobreescritura de la propiedad `stencil` del objeto `Flag`. LilyPond porporciona las siguientes funciones: `modern-straight-flag`, `old-straight-flag` y `flat-flag`. Utilice `\revert` para recuperar la forma predeterminada.

Para obtener corchetes más compactos o apilados verticalmente, llame a la instrucción `\flagStyleStacked`, que se puede anular después con `\flagStyleDefault`.

La sobreescritura de la forma de `Flag` no modifica la manera en que los corchetes se posicionan verticalmente. Esto se nota especialmente en los corchetes rectos: LilyPond no ajusta dinámicamente los huecos verticales entre los corchetes individuales de la misma forma que con las barras de corchea. Una posible solución para armonizar la apariencia es sustituir los corchetes planos por medias barras, como se ve en el segundo pentagrama; sin embargo, esto no se puede hacer automáticamente. En el código de este fragmento, las medias barras se escriben con el prefijo `@`, por ejemplo `@c8`.

Tenga en cuenta que las medias barras *no* son objetos gráficos del tipo `Flag`. Esto significa, particularmente, que la modificación de las propiedades de `Flag` no van a tener ningún efecto

sobre ellos (tendríamos que usar en su lugar las propiedades del objeto Beam), y las propiedades para su objeto Stem asociado se comportarán también como barras.

```
"@" =
#(define-music-function (music) (ly:music?)
  #{ \set stemLeftBeamCount = 0 $music [] #})

testnotes = {
  \autoBeamOff
  c8 d16 e''32 f64 \acciaccatura { g,,,8 } a128 b
}

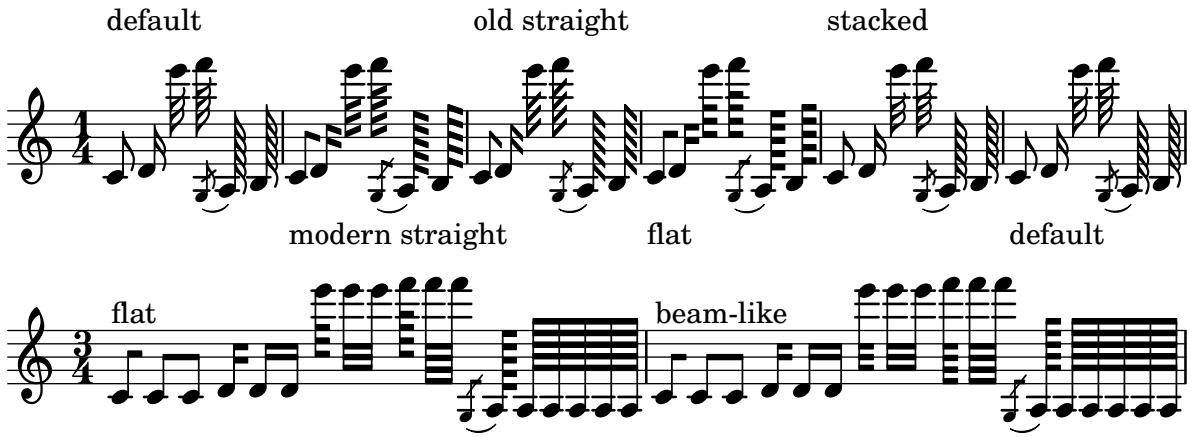
\relative c' {
  \override TextScript.staff-padding = 6
  \time 1/4
    <>^"default" \testnotes
  \override Flag.stencil = #modern-straight-flag
    <>_"modern straight" \testnotes
  \override Flag.stencil = #old-straight-flag
    <>^"old straight" \testnotes
  \override Flag.stencil = #flat-flag
    <>_"flat" \testnotes
  \revert Flag.stencil

  \flagStyleStacked
    <>^"stacked" \testnotes
  \flagStyleDefault
    <>_"default" \testnotes
}

\relative c' {
  \time 3/4
  \override Flag.stencil = #flat-flag

  <>^"flat" c8 c[ c] d16 d[ d] e''32 e[ e] f64 f[ f]
    \acciaccatura { g,,,8 } a128 a[ a a a a]
  <>^"beam-like" @c8 c[ c] @d16 d[ d] @e''32 e[ e] @f64 f[ f]
    \acciaccatura { g,,,8 } @a128 a[ a a a a]
}

\layout {
  indent = 0
  \context {
    \Score
    \override NonMusicalPaperColumn.line-break-permission = ##f
  }
}
```



Utilizar ly:grob-object para acceder a los grobs con \tweak

Se puede acceder "lateralmente" a algunos grobs desde dentro de la función de callback de otro grob. Éstos se encuentran relacionados normalmente como "layout objects" (objetos de presentación) en la sección "Internal properties" (propiedades internas) de un interface de grob. Se usa la función `ly:grob-object` para acceder a estos grobs.

Se presentan más abajo como ejemplo algunas formas de addecer a grobs desde dentro de una función de callback de `NoteHead`, pero la técnica no se limita a las cabezas de nota. Sin embargo, la función de callback de `NoteHead` es especialmente importante, porque es la función de callback implícita que utiliza la instrucción `\tweak`.

La salida de consola de la función de ejemplo que se muestra abajo (`display-grobs`) es como sigue.

```
--------------------
#<Grob Accidental >
()
#<Grob Stem >
```

Probablemente no es tan útil, pero muestra que efectivamente se puede acceder a los objetos gráficos.

```
#(define (notehead-get-accidental notehead)
   ;; notehead is grob
   (ly:grob-object notehead 'accidental-grob))

#(define (notehead-get-arpeggio notehead)
   ;; notehead is grob
   (let ((notecolumn (notehead-get-notecolumn notehead)))
     (ly:grob-object notecolumn 'arpeggio)))
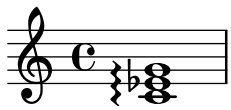
#(define (notehead-get-notecolumn notehead)
   ;; notehead is grob
   (ly:grob-parent notehead X))

#(define (notehead-get-stem notehead)
   ;; notehead is grob
   (let ((notecolumn (notehead-get-notecolumn notehead)))
     (ly:grob-object notecolumn 'stem)))

#(define (display-grobs notehead)
   ;; notehead is grob
```

```
   (let ((accidental (notehead-get-accidental notehead))
         (arpeggio (notehead-get-arpeggio notehead))
         (stem (notehead-get-stem notehead)))
     (format (current-error-port) "~2&~a\n" (make-string 20 #\-))
     (for-each
      (lambda (x) (format (current-error-port) "~a\n" x))
      (list accidental arpeggio stem))))

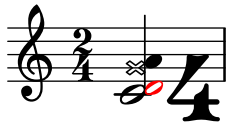
\relative c' {
  %% display grobs for each note head:
  %\override NoteHead.before-line-breaking = #display-grobs
  <c
  %% or just for one:
  \tweak before-line-breaking #display-grobs
  es
  g>1\arpeggio
}
```



Uso de la instrucción \tweak para retocar objetos gráficos individuales

Con la instrucción de trucar `\tweak` todos los objetos gráficos se pueden retocar directamente. Aquí presentamos ejemplos de los tipos de retoque disponibles.

```
\relative c' {
  \time 2/4
  \set fingeringOrientations = #'(right)
  <
    \tweak font-size #3 c
    \tweak color #red d-\tweak font-size #8 -4
    \tweak style #'cross g
    \tweak duration-log #2 a
  >2
}
```



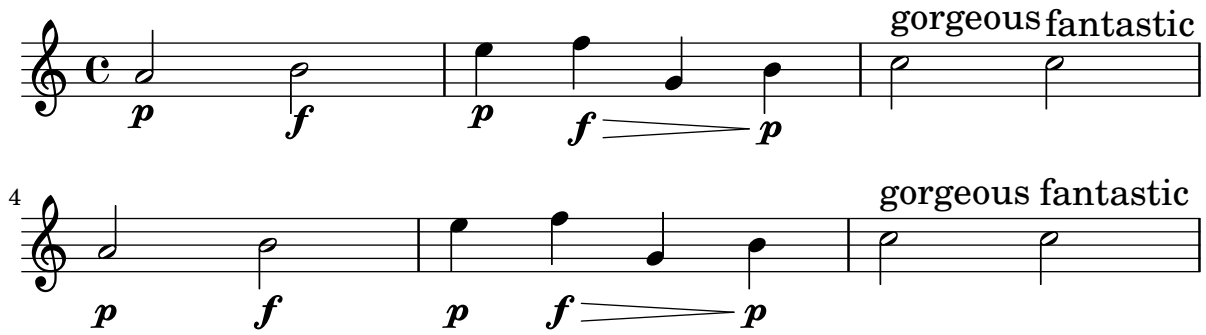
Indicaciones dinámicas y textuales alineadas verticalmente

Para todos los objetos `DynamicLineSpanner` (esto es, reguladores y textos de dinámica), la distancia vertical mínima entre su línea de referencia y el pentagrama viene dada por el valor dentro de la propiedad `staff-padding`, a no ser que otros elementos de notación los fuercen a situarse en una posición más alejada. El establecimiento de esta propiedad a un valor suficientemente grande, permite alinear las indicaciones de matiz dinámico.

Se usa una idea similar junto a \textLengthOn para alinear las inscripciones de texto a lo largo de su línea de base.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = 3
  \textLengthOn
  \override TextScript.staff-padding = 1
  \music
}
```



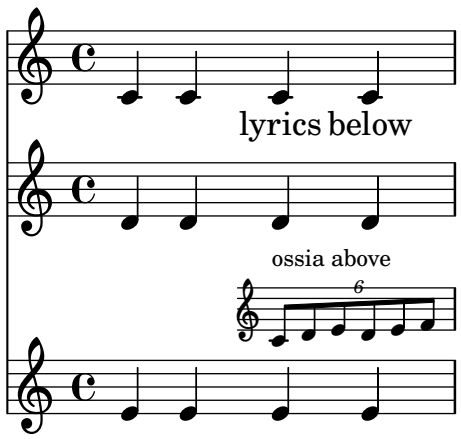
Alineación vertical de la letra y los compases de ossia

Este fragmento de código muestra el uso de las propiedades de contexto `alignBelowContext` y `alignAboveContext` para controlar la posición de la letra y los compases de ossia.

```
\relative c' <<
  \new Staff = "1" { c4 c c c }
  \new Staff = "2" { d4 d d d }
  \new Staff = "3" { e4 e e e }

  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = "1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = "3"
        fontSize = -2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
        \override VerticalAxisGroup.staff-staff-spacing =
           #'((minimum-distance . 0)
              (basic-distance . 0)
              (padding . 1))
      } {
        \tuplet 6/4 {
          \override TextScript.padding = 2
```

```
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



Alineación vertical de los números de estrofa de diferentes pentagramas

Puede ocurrir que los números de estrofa no estén alineados verticalmente si los versos están adjuntos a distinto pentagrama. Para corregirlo, seobreescriba la propiedad `self-alignment-X` del objeto gráfico `LyricText`.

```
\markup { default behavior }

<<
  \new Staff { b b b b }
  \lyrics {
    \stanza "3."
    a a a a
  }

  \new Staff { b b b b }
  \lyrics {
    \stanza "1."
    aaaaaaaaaa a a a
  }
  \lyrics {
    \stanza "2."
    a a a a
  }
>>

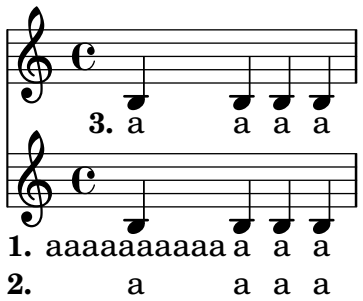
\markup \vspace #1
\markup {
  using \typewriter "self-alignment-X = #LEFT" }

<<
  \new Staff { b b b b }
```

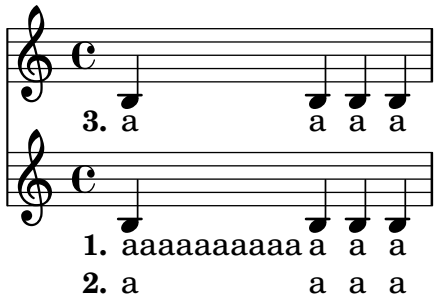
```
  \new Lyrics \lyricmode {
    \stanza "3."
    a a a a
  }

  \new Staff { b b b b }
  \new Lyrics \lyricmode {
    \stanza "1."
    \once \override LyricText.self-alignment-X = #LEFT
    aaaaaaaaaa a a a
  }
  \new Lyrics \lyricmode {
    \stanza "2."
    a a a a
  }
>>
```

default behavior



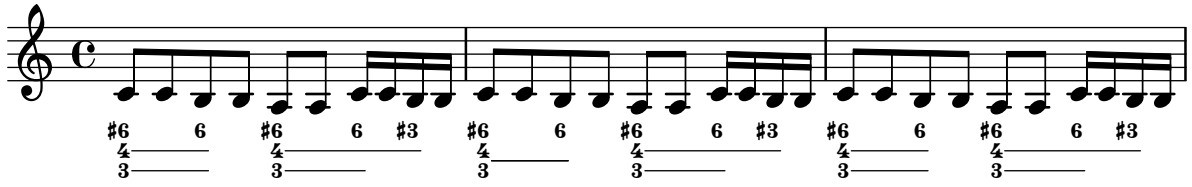
using `self-alignment-X = #LEFT`



Centrado vertical de las líneas de bajo cifrado emparejadas

Allí donde se utilicen líneas extensoras para el bajo cifrado mediante el establecimiento de `useBassFigureExtenders` a `#t`, las parejas de líneas extensoras congruentes se centran verticalmente si el valor de `figuredBassCenterContinuations` tiene el valor `#t`.

```
<<
  \relative c' {
    \repeat unfold 3 {
      c8 c b b a a c16 c b b
    }
  }
  \figures {
    \set useBassFigureExtenders = ##t
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
    \set figuredBassCenterContinuations = ##t
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
    \set figuredBassCenterContinuations = ##f
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
  }
>>
```

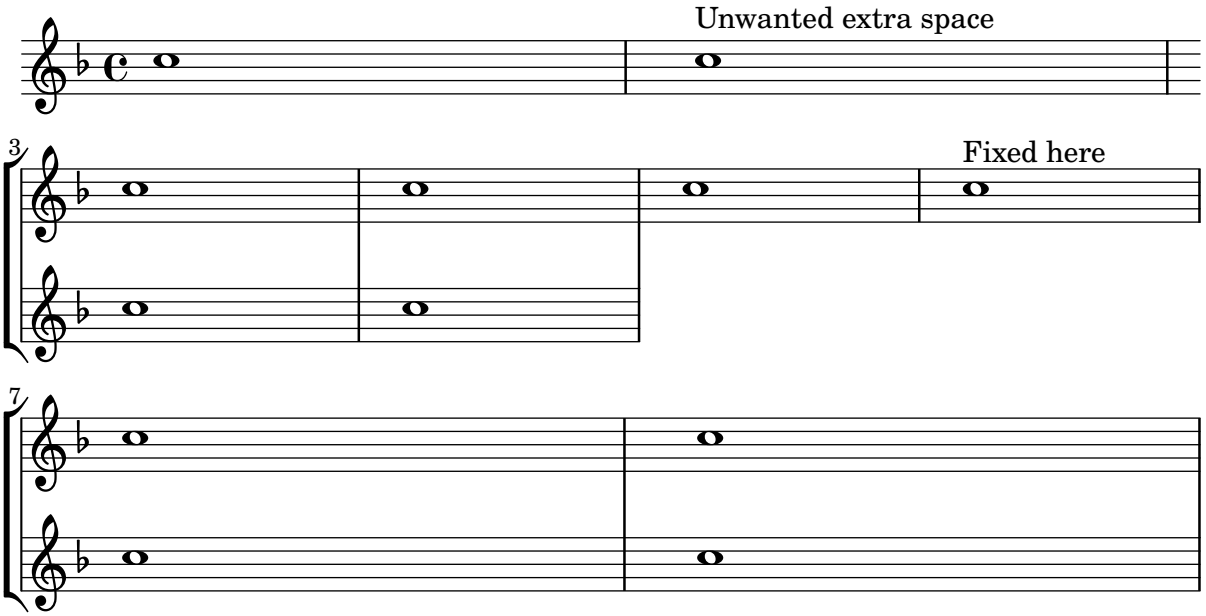


39 Workaround

Añadir un pentagrama adicional en un salto de línea

Al añadir un pentagrama nuevo en un salto de línea, por desgracia se añade un espacio adicional al final de la línea antes del salto (reservado para hacer sitio a un cambio de armadura que de todas formas nunca se imprime). La solución alternativa es establecer la propiedad `Staff.explicitKeySignatureVisibility` del objeto gráfico `Staff` como se muestra en el ejemplo.

```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
       \new Staff {
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
    c1 | c^"Fixed here" \break
    << { c1 | c }
       \new Staff {
         \once \set Staff.explicitKeySignatureVisibility =
           #end-of-line-invisible
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
  }
}
```



Apoyatura o nota de adorno antes de la línea divisoria

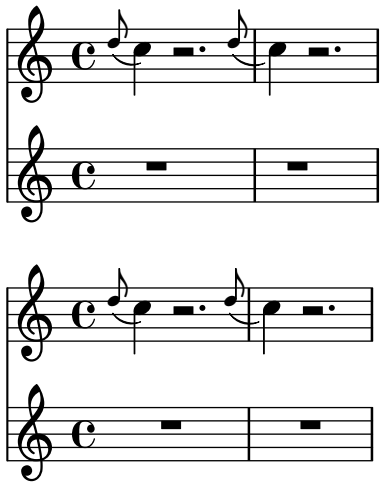
De manera predeterminada, las apoyaturas y las notas de adorno que están en el primer pulso de un compás se imprimen después de la línea divisoria. En pentagramas sueltos, una posible solución para que se impriman antes de la línea divisoria es añadir una línea invisible y después la línea visible.

Sin embargo, en sistemas de varios pentagramas, añadir una barra de compás invisible distorsiona el posicionamiento de los silencios de compás completo en las otras pautas; ya no estarían centrados sino desplazados ligeramente hacia la izquierda. Una solución mejor para tales situaciones es usar la instrucción `\afterGrace` command junto a la configuración apropiada de `afterGraceFraction`.

```
<<
  {
    \appoggiatura d''8 c''4 r2. |
    \appoggiatura { \bar "" d''8 \bar "|" } |
    c''4 r2.
  }
  { R1 | R1 }
>>

afterGraceFraction = 15/16

<<
  {
    \appoggiatura d''8 c''4 \afterGrace r2. d''8( |
    c''4) r2.
  }
  { R1 | R1 }
>>
```



Breaking horizontal alignment of dynamics and textscripts

LilyPond uses `DynamicLineSpanner` grobs to horizontally align successive dynamic objects like hairpins and dynamic text, even if they are positioned on different sides of a staff. This connection cannot be broken, contrary to the vertical alignment (see snippet “Breaking vertical alignment of dynamics and textscripts”).

There are two solutions to circumvent the problem.

- Modify the shorten-pair property of the Hairpin grob to compensate the offset by which the hairpin was moved.
- Put the two dynamic objects into different voices.

Both solutions are demonstrated in this snippet.

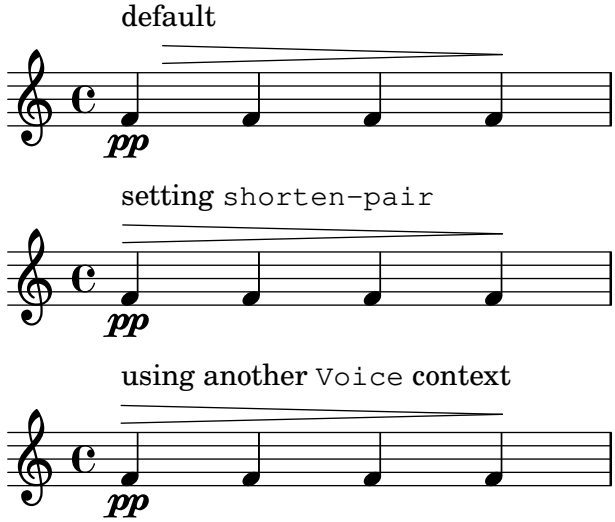
```
{
  <>^"default"
  f'_\pp ^\> f' f' f'\!
}

{
  <>^\markup { setting \typewriter shorten-pair }
  f'_\pp \tweak shorten-pair #'(-3 . 0) ^\> f' f' f'\!
}

{
  <>^\markup { using another \typewriter Voice context }
  << { f'^\> f' f' f'\! }
     \new Voice { s4_\pp } >>
}

\layout {
  line-width = 8\cm
  ragged-right = ##f

  \context {
    \Voice
    \override TextScript.staff-padding = #3.5
  }
}
```



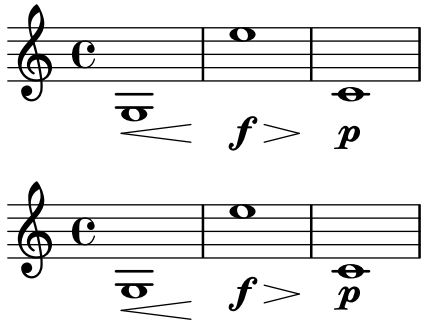
Desalinear indicaciones dinámicas y textuales verticalmente

De forma predeterminada, LilyPond usa ojetos gráficos DynamicLineSpanner para alinear verticalmente objetos de matiz dinámico sucesivos como reguladores y dinamicas textuales. Sin embargo esto no siempre es deseable. Insertando \breakDynamicSpan, que finaliza el objeto extenso de alineamiento de forma prematura, se puede evitar este alineamiento vertical.

Véase también el fragmento de código “Desalinear indicaciones dinámicas y textuales horizontalmente”.

```
{ g1\< |
  e''\f\> |
  c'\p }

{ g1\< |
  e''\breakDynamicSpan\f\> |
  c'\p }
```



Modificar el compás dentro de una sección polimétrica utilizando `\scaleDurations`

Polirritmia flexible con compases no alineados

Para poder crear explícitamente contextos con compases independientes, suprima el traductor `Timing_translator` del contexto `Score` y defina un contexto `TimingStaffGroup` que contenga el traductor `Timing_translator`. Esto hace que `Timing` sea un alias de `TimingStaffGroup`, con la mira puesta en las instrucciones `\time` hacia el `TimingStaffGroup` contenedor.

A diferencia de la instrucción incorporada `\enablePerStaffTiming` de LilyPond, este enfoque requiere la creación explícita de contextos `TimingStaffGroup`; a cambio, permite crear varios contextos `Staff` que de forma conjunta siguen al compás definido en su `TimingStaffGroup` contenedor.

Indicaciones de compás escalados localmente

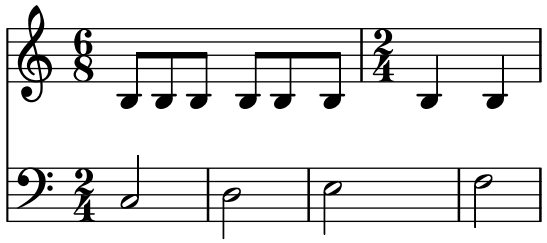
Use la instrucción no escalable `\time` para establecer un compás de la longitud deseada en `Timing`, más conocido como `TimingStaffGroup`. En este fragmento de código, todos los pentagramas que están por debajo de `TimingStaffGroup` usan una indicación de compás escalada, de forma que cualquier compás que establezcamos con la longitud deseada es tan bueno como cualquier otro. Si hubiese un contexto contenedor que no usara una indicación de compás escalada, la elección del compás que se establezca en `Timing` tendría importancia en dicho contexto.

Use la instrucción `\polymetric \time` para establecer propiedades métricas escalablesen los contextos que están por debajo de `Timing`, y use la instrucción `\scaleDurations` para escalar tanto el compás local como las notas que han de llenar el compás.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \accepts TimingStaffGroup
  }
  \context {
```

```
    \StaffGroup
    \name TimingStaffGroup
    \alias StaffGroup
    \consists "Timing_translator"
  }
}

<<
  \new TimingStaffGroup <<
    \new Staff {
      \scaleDurations 8/5 {
        \time 6/5 % to set measure length in Timing
        \context Staff \polymetric \time 6/8
        b8 b b b b b
        \time 4/5 % to set measure length in Timing
        \context Staff \polymetric \time 2/4
        b4 b
      }
    }
  >>
  \new TimingStaffGroup <<
    \new Staff {
      \clef bass
      \time 2/4
      c2 d e f
    }
  >>
>>
```



Solución de problemas de acordes y barras que cruzan el pentagrama

A veces es mejor usar plicas que parten del 'otro' pentagrama para crear acordes de pentagrama cruzado y así engañar al detector de colisiones de plicas de LilyPond. En el siguiente fragmento de código, si se hubieran usado las plicas que provienen del pentagrama inferior, habría sido necesario usar explícitamente

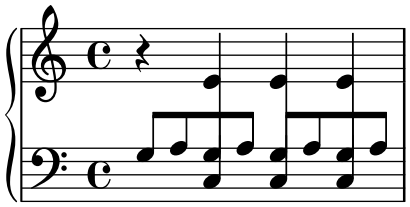
```
\override Staff.Beam.collision-voice-only = ##t
```

de forma que LilyPond no mueva las plicas.

```
\new PianoStaff <<
  \new Staff = up \relative c' <<
    { r4
      \override Stem.cross-staff = ##t
      \override Stem.length = #19 % this is in half-spaces,
          % so it makes stems 9.5 staffspaces long
```

```
      \override Stem.Y-offset = #-6 % stems are normally lengthened
          % upwards, so here we must lower the stem by the amount
          % equal to the lengthening - in this case (19 - 7) / 2
          % (7 is default stem length)
      e e e }
    { s4
      \change Staff = "bottom"
      \override NoteColumn.ignore-collision = ##t
      c, c c
    }
  >>

  \new Staff = bottom \relative c' {
    \clef bass
    \voiceOne
    g8 a g a g a g a
  }
>>
```
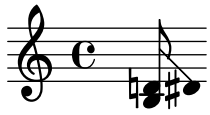


Impresión de acordes complejos

He aquí una forma de imprimir un acorde en el que suena la misma nota dos veces con distintas alteraciones.

```
fixA = {
  \once \override Stem.length = #12
}

fixB = {
  \once \override NoteHead.X-offset = #1.7
  \once \override Stem.length = #7
  \once \override Stem.rotation = #'(45 0 0)
  \once \override Stem.extra-offset = #'(-0.1 . -0.2)
  \once \override Flag.style = #'no-flag
  \once \override Accidental.extra-offset = #'(4 . -.1)
}


\relative c' {
  << { \fixA <b d!>8 } \\ { \voiceThree \fixB dis } >> s
}
```



Extender glissandos sobre repeticiones

Se puede simular un glissando que se extiende hasta el interior de varios bloques `\alternative` de primera y segunda vez mediante la adición de una nota de adorno oculta con un glissando al comienzo de cada bloque `\alternative`. La nota de adorno debe estar a la misma altura que la nota que da inicio al primer glissando. Esto se implementa aquí con una función musical que toma como argumento la altura de la nota de adorno.

Observe que en música polifónica la nota de adorno debe coincidir con las notas de adorno correspondientes en todas las otras voces.

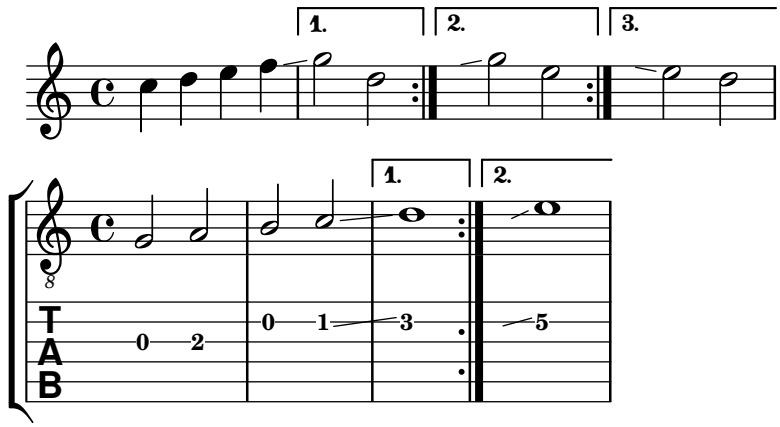
```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music =  \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}

\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff  <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
```

```
}
```



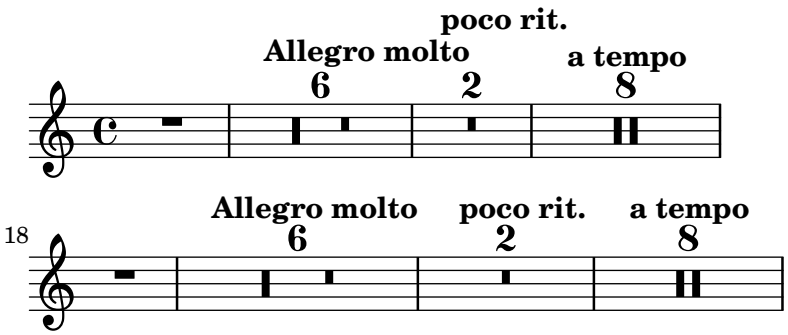
Forcing measure width to adapt to a metronome mark's width

By default, metronome marks do not influence horizontal spacing. This can be solved through a simple override, as shown in the second half of the example.

```
example = {
  R1
  \tempo "Allegro molto" R1*6
  \tempo "poco rit." R1*2
  \tempo "a tempo" R1*8 \break
}

{
  \compressMMRests {
    \example
    \override Score.MetronomeMark.extra-spacing-width = #'(-3 . 0)
    \example
  }
}
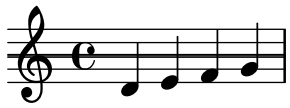
\layout {
  ragged-right = ##t
}
```



Hacer unas líneas del pentagrama más gruesas que las otras

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```
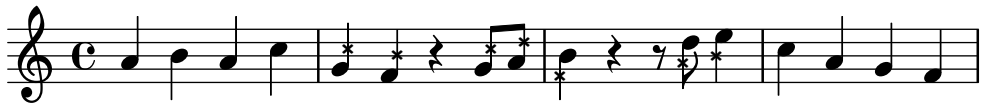


Marcar notas de las partes habladas con unas aspas en la plica

Este ejemplo muestra cómo añadir aspas sobre las plicas. Marcamos el comienzo de una sección hablada con la instrucción \speakOn, y el final con \speakOff.

```
speakOn = \override Stem.stencil =
  #(lambda (grob)
     (let* ((x-parent (ly:grob-parent grob X))
            (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
       (if is-rest?
           empty-stencil
           (ly:stencil-combine-at-edge
            (ly:stem::print grob)
            Y
            (- (ly:grob-property grob 'direction))
            (grob-interpret-markup
     grob
             (markup #:center-align #:fontsize -4
                     #:musicglyph "noteheads.s2cross"))
            -1.7))))

speakOff = \revert Stem.stencil

\new Staff {
  \relative c'' {
    a4 b a c
    \speakOn
    g4 f r g8 a
    b4 r r8 d e4
    \speakOff
    c4 a g f
  }
}
```



Posicionamiento de las notas de adorno con espacio flotante

Al establecer la propiedad strict-grace-spacing hacemos que las columnas musicales para las notas de adorno sean 'flotantes', es decir, desacopladas de las notas que no son de adorno: primero se aplica el espaciado de las notas normales, y luego se ponen las columnas musicales de las notas de adorno a la izquierda de las columnas musicales de las notas principales.

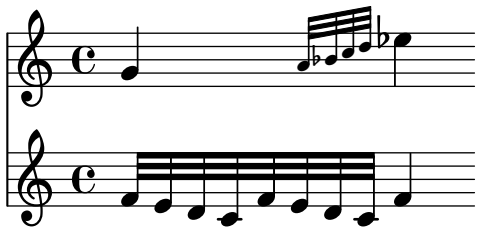
Sin embargo, a causa del bug número 6876 (https://gitlab.com/lilypond/lilypond/-/issues/6876), las alteraciones accidentales se ignoran si esta propiedad está establecida. Este fragmento de código ofrece una solución para este problema.

Otro desafortunado efecto colateral de esta propiedad es que LilyPond no comprueba si hay espacio horizontal suficiente para las notas de adorno (este problema recibe seguimiento como bug número 2630 (https://gitlab.com/lilypond/lilypond/-/issues/2630)). Tenemos que asegurarnos de que ha espacio suficiente, por ejemplo, por medio del uso de \newSpacingSection junto con un valor adecuado para la duración base-shortest-duration del objeto gráfico SpacingSpanner.

```
shiftedGrace =
#(define-music-function (offset music) (number? ly:music?)
   #{
     \override NoteHead.X-offset = #(- offset 0.85)
     \override Stem.X-offset = #offset
     \grace { $music }
     \revert NoteHead.X-offset
     \revert Stem.X-offset
   #})

\relative c'' <<
  { g4 \shiftedGrace #-1.3 a32 \shiftedGrace #-0.5 { bes c d } es4 }
  { f,32 e d c f e d c f4 }
>>

\layout {
  \context {
    \Score
    \override SpacingSpanner.strict-grace-spacing = ##t
  }
}
```

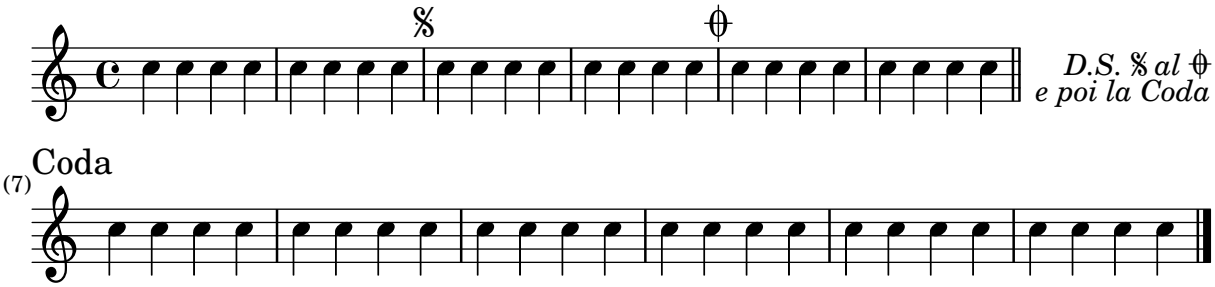


Posicionar los símbolos de repetición segno y coda (con saltos de línea)

Si queremos colocar un símbolo de segno de salida y añadir texto como “D.S. al Coda” junto a él donde normalmente están las líneas del pentagrama, cpodemos usar este código. La coda continúa en una línea nueva. Hay una variante documentada en el fragmento de código, donde la coda se mantiene en la misma línea.

```
\relative c'' {
  c4 c c c | c c c c |
  \repeat segno 2 {
    c4 c c c | c c c c |
    \alternative {
      \volta 1 {
```

```
        c4 c c c | c c c c |
        % If you don't use \break at Coda, use \noBreak here
        % and after \bar "" below.
        \noBreak
        \section % double bar line
        \cadenzaOn % pause bar count
        \stopStaff % remove staff lines
        % Increasing the unfold counter will expand the staff-free space
        \repeat unfold 4 {
          s1
          \bar ""
        }
        % Place JumpScript where the staff would normally be.
        \once \override Score.JumpScript.outside-staff-priority = ##f
        \once \override Score.JumpScript.Y-offset = 0
        \startStaff % resume bar count
        \cadenzaOff % show staff lines again
      }
    }
  }
  \sectionLabel "Coda"
  % Show Coda on a new line
  \break
  \*6 { c4 c c c }
  \fine
}
```



(7)

Evitar que una marca de ensayo final suprima el corchete de un grupo de valoración especial

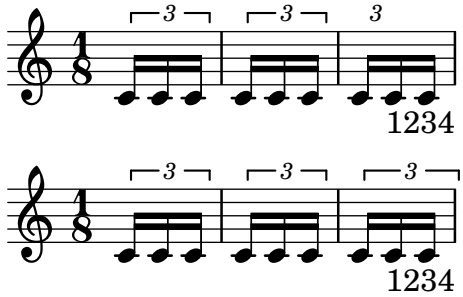
A causa del bug número 2362 (`https://gitlab.com/lilypond/lilypond/-/issues/2362`), al añadir una marca de ensayo `mark` al final puede ocurrir que se pierda una indicación de grupo especial colocado al final. Se puede solventar fijando `TupletBracket.full-length-to-extent` a `#f`.

```
\new Staff {
  \set tupletFullLength = ##t
  \time 1/8
  \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' }
  \tweak direction #DOWN \textEndMark "1234"
}

\new Staff {
  \set tupletFullLength = ##t
```

```
  \override TupletBracket.full-length-to-extent = ##f

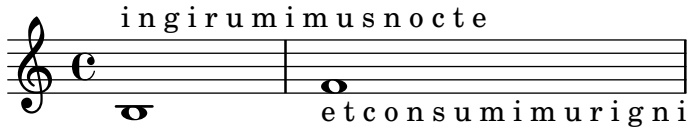
  \time 1/8
  \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' }
  \tweak direction #DOWN \textEndMark "1234"
}
```



Impresión de texto de derecha a izquierda

Es posible imprimir texto de derecha a izquierda en un elemento de marcado, como se muestra aquí.

```
{
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```
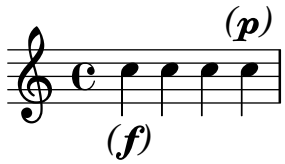


Slanted parentheses around dynamics

The easiest way to parenthesize dynamics is to use the normal \parenthesize command and adjust the font size, e.g., { c'4\tweak Parentheses.font-size 2 \parenthesize \f }. However, this results in upright parentheses, while dynamic marks are normally slanted. This snippet gives a method to make the parentheses slanted as well.

```
paren =
#(define-event-function (dyn) (ly:event?)
   (make-dynamic-script
    #{ \markup \concat {
         \normal-text \italic \fontsize #2 (
         \pad-x #0.2 #(ly:music-property dyn 'text)
         \normal-text \italic \fontsize #2 )
       }
    #}))

\relative c'' {
  c4\paren\f c c \dynamicUp c\paren\p
```

```
}
```



Ejemlo de pandereta

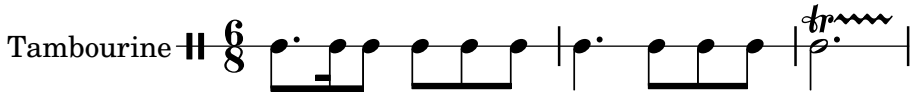
A tambourine example, entered with 'tamb'.

```
#(define mydrums '((tambourine default #f 0)))

\new DrumStaff \with { instrumentName = #"Tambourine" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  \time 6/8
  tamb8. 16 8 8 8 8 |
  tamb4. 8 8 8 |
  % The trick with the scaled duration and the shorter rest
  % is neccessary for the correct ending of the trill-span!
  tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
}
```



Transposing pitches with minimum accidentals ("smart" transpose)

This example uses some Scheme code to enforce enharmonic modifications for notes in order to have the minimum number of accidentals. In this case, the following rules apply:

- double accidentals should be removed
- b sharp → c
- e sharp → f
- c flat → b
- f flat → e

In this manner, the most natural enharmonic notes are chosen.

```
#(define (naturalize-pitch p)
   (let ((o (ly:pitch-octave p))
         ;; `ly:pitch-alteration` returns quarter tone steps.
         (a (* 4 (ly:pitch-alteration p)))
         (n (ly:pitch-notename p)))
     (cond
      ((and (> a 1)
            (or (eqv? n 6) (eqv? n 2)))
       (set! a (- a 2))
```
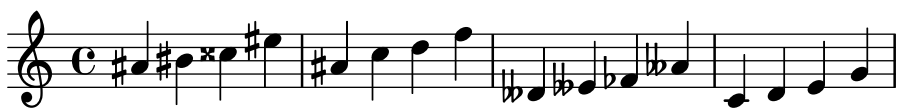
```
       (set! n (+ n 1)))
      ((and (< a -1)
            (or (eqv? n 0) (eqv? n 3)))
       (set! a (+ a 2))
       (set! n (- n 1))))
     (cond
      ((> a 2)
       (set! a (- a 4))
       (set! n (+ n 1)))
      ((< a -2)
       (set! a (+ a 4))
       (set! n (- n 1))))
     (when (< n 0)
       (set! o (- o 1))
       (set! n (+ n 7)))
     (when (> n 6)
       (set! o (+ o 1))
       (set! n (- n 7)))
     (ly:make-pitch o n (/ a 4))))

#(define (naturalize music)
   (let ((es (ly:music-property music 'elements))
         (e (ly:music-property music 'element))
         (p (ly:music-property music 'pitch)))
     (when (pair? es)
       (ly:music-set-property! music 'elements
                               (map naturalize es)))
     (when (ly:music? e)
       (ly:music-set-property! music 'element
                               (naturalize e)))
     (when (ly:pitch? p)
       (set! p (naturalize-pitch p))
       (ly:music-set-property! music 'pitch p))
     music))

naturalizeMusic =
#(define-music-function (m) (ly:music?)
   (naturalize m))

music = \relative c' { c4 d e g }

\new Staff {
  \transpose c ais { \music }
  \naturalizeMusic \transpose c ais { \music }
  \transpose c deses { \music }
  \naturalizeMusic \transpose c deses { \music }
}
```



Unfolding tremolo repeats
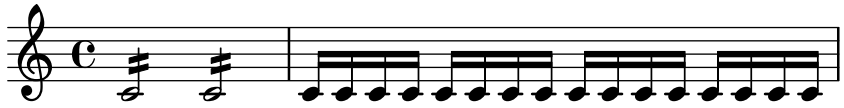
Currently, *note*:*duration*, which is more or less a shortcut for \repeat tremolo, is not unfolded by \unfoldRepeats (this is tracked in Issue #6145 (https://gitlab.com/lilypond/lilypond/-/issues/6145)). The function given in this snippet provides a workaround.

```
fixTremolos =
#(define-music-function (music) (ly:music?)
   (music-map
    (lambda (m)
      (let ((event (any (lambda (a)
                          (and (music-is-of-type? a 'tremolo-event)
                               a))
                        (ly:music-property m 'articulations))))
        (if event
            (let* ((total-tremolo-duration (ly:music-property m
                                                              'duration))
                   (tremolo-type (ly:music-property event
                                                    'tremolo-type))
                   (one-tremolo-note-duration
                    (ly:make-duration (ly:intlog2 tremolo-type)))
                   (tremolo-note-count
                    (/ tremolo-type (expt 2 (ly:duration-log
                                             total-tremolo-duration)))))
              (set! (ly:music-property m 'duration)
                    one-tremolo-note-duration)
              (set! (ly:music-property m 'articulations)
                    (delete! event (ly:music-property m 'articulations)))
              (make-music 'TremoloRepeatedMusic
                          'repeat-count tremolo-note-count
                          'element m))
            m)))
    music))

unfoldRepeats = \unfoldRepeats #'() \fixTremolos \etc

music = { \repeat tremolo 8 c'16 c'2:16 }

{
  \music
  \unfoldRepeats \music
}
```



Uso de una voz adicional para los saltos de línea y de página

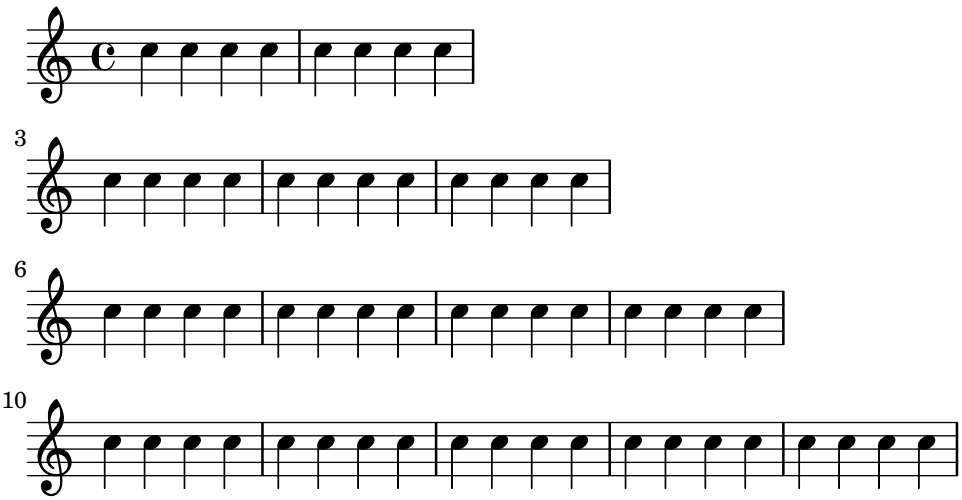
Con frecuencia es más fácil gestionar la información de los saltos de línea y de página manteniéndola separada de la música, por medio de la introducción de una vos adicional que contiene solamente silencios de separacion junto a las instrucciones \break, \pageBreak y otras informaciones de disposición.

Este patrón se hace especialmente útil cuando se está sobreescribiendo la propiedad `line-break-system-details` y las otras útiles pero largas propiedades del objeto gráfico `NonMusicalPaperColumn`.

```
music = \relative c'' { c4 c c c }

\score {
  \new Staff <<
    \new Voice {
      s1*2 \break
      s1*3 \break
      s1*4 \break
      s1*5 \break
    }
    \new Voice {
      \*2 \music
      \*3 \music
      \*4 \music
      \*5 \music
    }
  >>
}

\paper {
  indent = 0
  line-width = 140\mm
  ragged-right = ##t
}
```



Indicaciones dinámicas y textuales alineadas verticalmente

Para todos los objetos `DynamicLineSpanner` (esto es, reguladores y textos de dinámica), la distancia vertical mínima entre su línea de referencia y el pentagrama viene dada por el valor dentro de la propiedad `staff-padding`, a no ser que otros elementos de notación los fuercen a situarse en una posición más alejada. El establecimiento de esta propiedad a un valor suficientemente grande, permite alinear las indicaciones de matiz dinámico.

Se usa una idea similar junto a \textLengthOn para alinear las inscripciones de texto a lo largo de su línea de base.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = 3
  \textLengthOn
  \override TextScript.staff-padding = 1
  \music
}
```

