LilyPond

Le système de gravure musicale

Morceaux choisis

L'équipe de développement de LilyPond

Ce document regroupe un certain nombre d'exemples, trucs et astuces pour LilyPond issus du LilyPond Wiki (`https://wiki.lilypond.community`). Il est dans le domaine public.

Pour connaître la place qu'occupe ce manuel dans la documentation, consultez la page Section "Manuels" dans *Informations générales*.

Si vous ne disposez pas de certains manuels, la documentation complète se trouve sur `https://lilypond.org/`.

Ce document a été placé dans le domaine public ; en France, les auteurs renoncent à tous leurs droits patrimoniaux.

Pour LilyPond version 2.27.4

Note : La génération automatique de ce document ne permet malheureusement pas l'affichage des rubriques dans la langue de Molière ; veuillez nous en excuser.

Table des matières

Préface **1**

notation musicale

1 Pitches **5**

Un ambitus par voix 5
Ajout d'une indication d'octave pour une seule voix 5
Variante fine des notes profilées Aiken 6
Modification de la hauteur de hampes ligaturées 6
Ambitus 7
Ambitus après l'armure 7
Ambitus sur plusieurs voix 8
Profilage des notes selon leur degré dans la gamme 8
Changement automatique de l'orientation de hampe de la note médiane selon la mélodie . 9
Modification du texte des marques d'octaviation 10
Réglage de l'affichage d'un ambitus 10
Modification de l'intervalle des lignes de la portée 11
Transposition arbitraire d'une clef 12
Coloration des notes selon leur hauteur 12
Création d'une séquence de notes de même hauteur 13
Création d'armures personnalisées 13
Orientation de la tête d'un « fa » profilé en cas de fusion 14
Impression forcée d'un bécarre avant une accidentelle 15
Impression forcée de la clef 15
Génération de notes aléatoires 15
Non répétition de l'altération après saut de ligne sur liaison de prolongation 16
Conservation de la taille lors d'un changement de clef 16
Exemple de musique « Makam » 17
Modification de la pente de l'extension d'octaviation 17
Armures inhabituelles 17
Easy play – chiffres en lieu et place des lettres 18
Orchestre, chœur et piano 19
Suppression des bécarres superflus 24
Suppression des bécarres superflus lors d'un changement de tonalité 24
Citation d'une autre voix et transposition 24
Séparation entre altérations annulées et nouvelle armure 25
Transposition et réduction du nombre d'altérations accidentelles 26
Exemple de makam turc 27
Affinage des propriétés d'une clef 28
Utilisation d'`autoChange` avec plus d'une voix 29

2 Rhythms **31**

Coexistence de ligatures et liaisons avec des rythmes comprenant des n-olets 31
Ajout de parties de batterie 31
Ajustement de l'espacement des notes d'ornement 32
Alignement des numéros de mesure 32

Brève alternative, avec deux barres verticales 33
Appoggiature avant une barre de mesure 34
Subdivision de ligatures automatiques 34
Changement automatique des durées 35
Définition de règles de ligature pour la partition 36
Débordement de ligature 37
Ligature au moment d'un saut de ligne 38
Ligature et directions de hampe inversées 39
Modifier l'apparence d'un silence multimesure 39
Spécification du nombre de points d'augmentation d'une note 39
Changement de tempo sans indication sur la partition 40
Modifier l'apparence du chiffre de n-olet 40
Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations` 40
Notation pour psalmodie 42
Métrique décomposée 42
Signes de direction, signes de sous-groupe 43
Contrôle de l'impression des crochets de n-olet 43
Exemple de sonnaille et clochette 44
Création d'une indication métronomique sous forme d'étiquette 45
Dessin à main levée de liaisons de tenue 45
Impression de trémolo avec ligature flottante 46
Plusieurs triolets avec une seule commande `\tuplet` 47
Fixation arbitraire du numéro de repère de départ 47
Génération de crochets personnalisés 48
Rythmique et guitare 49
Indication personnalisée d'une polymétrie complexe 50
Exemple de woodblocks haut et bas 50
Recours à la propriété `transparent` pour rendre des objets invisibles 51
Dessin d'une liaison d'articulation au trait discontinu 52
Contrôle manuel du positionnement des ligatures 53
Fusion de silences multimesures dans une partie polyphonique 53
Modification de la longueur d'un crochet de n-olet 53
Déplacement des notes pointées dans une polyphonie 54
Contrôle de la taille d'un silence multimesure 54
Ajout de texte à un silence multimesure 55
N-olets au chiffrage inhabituel 55
Dénombrer une unique mesure de silence 56
PartCombine et autoBeamOff 57
Exemple pour percussions 57
Saut de ligne au milieu d'un n-olet avec ligature 59
Positionnement des ligatures de notes d'ornement à la hauteur de celles des notes normales . 60
Positionnement des notes d'ornement avec espace flottant 60
Positionnement des silences multimesures 61
Positionnement de points d'orgue en opposition sur une barre de mesure 62
Préservation de l'indication de n-olet lors d'un repère final 63
Imprimer les numéros de mesure à intervalle régulier 63
Impression du numéro des mesures tronquées 64
Inscrire le numéro de mesure dans un cadre ou un cercle 64
Impression du numéro de mesure selon `modulo-bar-number-visible` 65
Changement de la fréquence d'impression du numéro de mesure 66
Impression du métronome et des repères sous la portée 66

Impression de musique aux métriques différentes 66
Afficher le numéro de la première mesure 69
Affichage du crochet de n-olet du côté des têtes de note 69
Redéfinition des réglages de mise en forme par défaut des notes d'ornement 70
Suppression des numéros de mesure d'une partition 70
Styles de silences 71
Annulation des règles de ligature par défaut 72
Barres rythmiques 72
Sauts de notes en mode paroles 73
Sauts de notes en mode paroles (2) 73
Moignons de hampe 73
Ligature à la pulsation 74
Subdivision des ligatures 74
Exemple pour tam-tam 76
Exemple pour tambourin 76
Encadrement sur trois côtés 77
Métrique entre crochets 77
Métrique entre parenthèses 78
Affichage seulement du numérateur d'une métrique (au lieu d'une fraction) 78
Mise en forme des notes d'ornement 79
Personnalisation de la métrique 79
Crochets de style alternatif 80
Utilisation de hampe barrée pour une note normale 81
Liaison de tenue et arpège 81

3 Expressive marks 83

Mise entre parenthèses d'un signe d'interprétation ou d'une note d'un accord 83
Ajout de marques temporelles à un long glissando 83
Ajustement du positionnement vertical d'une liaison 84
Ajustement du galbe des chutes ou sauts 84
Alignement des bornes de soufflet relativement aux `NoteColumns` 85
Brève alternative, avec deux barres verticales 85
Liaison asymétrique 85
Signes de respiration 86
Soufflet de crescendo partiellement interrompu 87
Césure en forme de « voie de chemin de fer » avec point d'orgue 87
Texte centré sous un soufflet 88
Modification du texte et de l'extension de nuances textuelles 89
Modification de l'aspect des liaisons d'articulation 90
Modification de l'indicateur de respiration 90
Spécification du nombre de points d'augmentation d'une note 91
Combinaison de nuance et de texte 91
Combinaison de nuance et de texte (2) 91
Glissando contemporain 92
Contrôle de la visibilité des extensions d'objet après saut de ligne 92
Contrôler la position des doigtés dans un accord 93
Contrôle de l'ordre vertical des articulations et ornements 94
Création d'un grupetto retardé 94
Arpège distribué sur plusieurs voix 95
Arpège distribué sur une partition pour piano 95
Arpège distribué pour un autre contexte que le piano 96

Création de doigtés sur deux caractères 97
Création d'une liaison entre plusieurs voix 97
Création d'extensions de texte 98
Personnalisation des extenseurs de nuance textuelle 99
Glissando par dessus un objet graphique 99
Stylisation des lignes de soufflet 99
Masquage de l'extension des nuances textuelles 100
Alignement de nuances personnalisées comme « più f » 100
Insertion d'une césure 103
Liaison « Laissez vibrer » 104
Terminaison de ligne en flèche 104
Dessin d'une liaison d'articulation au trait discontinu 105
Modification de la signification des raccourcis pour les signes d'articulation 105
Déplacement vertical des liaisons d'articulation 106
Déplacer les extrémités de soufflets 107
Positionnement des arpeggios 107
Positionnement d'une annotation à l'intérieur d'une liaison 107
Différents styles de soufflet 108
Impression de soufflets « al niente » 108
Impression du métronome et des repères sous la portée 109
Soufflets et barres de mesure 109
Ajustement de la longueur d'un soufflet 110
Impression d'une même articulation des deux côtés d'une note ou d'un accord 110
Indications de nuance entre parenthèses en italique 111
Slap ou pizzicato Bartok 111
Using \arpeggioBracket to make divisi more visible 112
Utilisation d'un crochet pour rendre les divisions plus évidentes 112
Remplacement du signe de respiration par une coche 113
Accords et double liaison d'articulation 113
Utilisation de la propriété `whiteout` 114
Articulation baroque en forme de coche 114
Alignement vertical des nuances sur plusieurs notes 115

4 Repeats 116

Modification des barres de mesure par défaut 116
Contrôle de l'apparence des traits de trémolo 116
Trémolo et changement de portée 118
Impression de trémolo avec ligature flottante 118
Répétition en pourcent isolée 119
Compteur de mesures 119
Affichage du numéro de répétition en pourcent 120
Compteur de répétition en pourcent 120
Positionnement des segno et coda (avec saut de ligne) 121
Succession de reprises et style de barre par défaut 122
Diminution de la taille du crochet d'alternative 122
Expansion de répétition de trémolo 122
Crochet de reprise sous les chiffrages d'accord 123
Crochets de *volta* sur plusieurs portées 124
Indication de reprise avec texte grâce à `repeatCommands` 125

5 Simultaneous notes 126

Ajout de voix pour éviter les collisions 126
Modification des indications de parties combinées 126
Modification de la taille d'une note particulière d'un accord 127
Clusters 127
Combinaison de deux parties sur une même portée 128
Impression d'accords complexes 129
Décalage horizontal forcé 129
Recours à la propriété `transparent` pour rendre des objets invisibles 130
Déplacement des notes pointées dans une polyphonie 131
Suppression des avertissements de chevauchement 131
Deux `\partCombine` sur une même portée 131

6 Staff notation 134

Un ambitus par voix 134
Ajout d'une portée supplémentaire 134
Ajout d'une portée supplémentaire après un saut de ligne 135
Ajout d'indicateurs à une portée dédoublée après un saut de ligne 136
Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur 140
Numérotation des mesures et alternatives 142
Ambitus après l'armure 143
Modification des barres de mesure par défaut 144
Modification du nombre de lignes de la portée 144
Modification de la taille d'une portée 145
Impression de papier à musique 145
Création d'armures personnalisées 147
Hampes interportées 148
Indicateur de regroupement et portée unique 149
Affichage d'un système complet lorsqu'une seule portée est active 149
Allongement d'une marque de trille (`TrillSpanner`) 151
Rappel du glissando à l'occasion d'une alternative 152
Liaison de prolongation aplatie 153
Adaptation de la largeur de mesure selon le `MetronomeMark` 156
Glissando par dessus un objet graphique 156
Harmonizing bar line thickness for staves with different sizes 156
Incipit 159
Insertion d'un fragment dans un *markup* au-dessus de la portée 163
Impression de la corde aiguë d'un `TabStaff` en bas 164
Tablature en lettres 165
Saut de ligne et glissando 165
Empâtement de certaines lignes d'une portée 167
Compteur de mesures 167
Présentation à l'ancienne (barres de mesure entre les portées) 168
Modification de la pente de l'extension d'octaviation 168
Imbrications de regroupements de portées 169
Armures inhabituelles 170
Orchestre, chœur et piano 171
Impression de noms d'accords identiques à la basse différente par une oblique et nouvelle basse 176
Impression des paroles dans la portée 178
Citation d'une autre voix 178

Citation d'une autre voix et transposition 180
Retrait de l'accolade à la première ligne d'une pièce pour piano 180
Suppression de la partie interportée des barres de mesure d'un regroupement autre que `ChoirStaff` 181
Masquage de la première ligne si elle est vide 182
Séparation visuelle entre les systèmes 183
Galbe individuel des liaisons d'un accord 184
Barre de mesure en encoche 187
Métrique entre crochets 187
Métrique entre parenthèses 187
Affinage des propriétés d'une clef 188
Deux `\partCombine` sur une même portée 189
Indication de regroupement de portées par un rectangle 191
Utilisation d'`autoChange` avec plus d'une voix 191
Utilisation de lignes de repères dans une partition « à la française » 192
Alignement vertical de `StaffGroups` sans `SystemStartBar` 195
Crochet de reprise sous les chiffrages d'accord 202
Crochets de *volta* sur plusieurs portées 203

7 Editorial annotations 204

Ajout de doigtés à une partition 204
Ajout de liens à des objets 204
Ajout de *markups* à une tablature 206
Impression des doigtés à l'intérieur de la portée 206
Numérotation des mesures et alternatives 207
Crochets d'analyse au-dessus de la portée 208
Crochet d'analyse avec texte 209
Profilage des notes selon leur degré dans la gamme 210
Blanchiment de lignes de portée avec la commande `\whiteout` 211
Modification de la taille d'une note particulière d'un accord 211
Modification de l'aspect des liaisons d'articulation 211
Coloration des notes selon leur hauteur 212
Contrôler la position des doigtés dans un accord 213
Création d'un grupetto retardé 213
Impression de papier à musique 214
Création de doigtés sur deux caractères 216
Direction par défaut des hampes de la ligne médiane 216
Taille de police différente pour `instrumentName` et `shortInstrumentName` 217
Encadrement d'objets 218
Encerclement de notes 219
Encercler divers objets 219
Intégration de PostScript dans un *markup* 220
Génération de notes profilées spéciales 220
Apparence du quadrillage temporel 221
Quadrillage temporel : mise en évidence du rythme et synchronisation des notes 222
Hammer on et pull off 223
Hammer on et pull off avec accords 223
Hammer on et pull off gérés par les voix 224
Empâtement de certaines lignes d'une portée 224
Adjonction d'une croix sur la hampe des notes d'un fragment parlé 224
Compteur de mesures 225

Crochet de mesure ... 226
Positionnement précis des indications de doigté ... 227
Positionnement d'une annotation à l'intérieur d'une liaison ... 228
Impression de texte de droite à gauche ... 228
Ligne de prolongation pour numéro de corde ... 228
Utilisation de la propriété `whiteout` ... 229

8 Text ... 230

Ajout de *markups* à une tablature ... 230
Ajout de la date du jour à une partition ... 230
Ajustement de l'espacement vertical des paroles ... 231
Alignement des noms d'instrument ... 231
Alignement de la première syllabe d'un mélisme ... 233
Alignement de repère sur une note ... 233
Blanchiment de lignes de portée avec la commande `\whiteout` ... 234
Texte centré sous un soufflet ... 234
Modification du texte des marques d'octaviation ... 236
Changement des fontes textuelles par défaut ... 236
Combinaison de nuance et de texte ... 237
Combinaison de nuance et de texte (2) ... 238
Combinaison de deux parties sur une même portée ... 238
Création d'extensions de texte ... 239
Champs d'entête ... 240
Intégration de PostScript dans un *markup* ... 242
Mise en forme individuelle de syllabes ... 242
Séparation de syllables par une liaison ... 242
Alignement des syllabes ... 243
Liste de *markups* ... 243
Ajout de texte à un silence multimesure ... 244
De l'ubiquité des objets *markup* ... 245
Impression du numéro de version ... 246
Piano et paroles entre les portées ... 246
Impression des indications sur toutes les portées d'un système ... 247
Impression de texte de droite à gauche ... 248
Impression des paroles dans la portée ... 248
Indications de nuance entre parenthèses en italique ... 249
Bloc de texte indépendant sur deux colonnes ... 249
Ligne de prolongation pour numéro de corde ... 250
Encadrement sur trois côtés ... 250
UTF-8 ... 251
Ensemble vocal avec alignement des paroles selon le contexte ... 253
Indication de reprise avec texte grâce à `repeatCommands` ... 254

Notation spécialisée

9 Vocal music 257

Un ambitus par voix 257
Ajout d'indicateurs à une portée dédoublée après un saut de ligne 257
Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur 261
Ajustement de l'espacement vertical des paroles 263
Alignement de la première syllabe d'un mélisme 263
Ambitus 264
Ambitus après l'armure 265
Ambitus sur plusieurs voix 265
Exemples de notation ancienne – transcription moderne de musique grégorienne 266
Psalmodie anglicane 266
Agencement de paroles séparées sur une seule ligne 269
Changement de fontes des couplets 270
Notation pour psalmodie 271
Impression forcée de tirets entre syllabes 271
Mise en forme individuelle de syllabes 271
Séparation de syllables par une liaison 272
Modèle pour cantique 272
Alignement des syllabes 274
Adjonction d'une croix sur la hampe des notes d'un fragment parlé 275
Orchestre, chœur et piano 276
Piano, mélodie et paroles 281
Impression des paroles dans la portée 282
Modèle pour chœur SATB, sur quatre portées 282
Paroles, musique et accords 284
Paroles, musique, accords et diagrammes de fret 285
Portée unique et paroles 286
Sauts de notes en mode paroles 286
Sauts de notes en mode paroles (2) 287
Using \arpeggioBracket to make divisi more visible 287
Utilisation d'un crochet pour rendre les divisions plus évidentes 288
Recours aux balises pour produire une partition ancienne et moderne à partir de la même source 288
Positionnement d'une ossia et des paroles 291
Alignement vertical des numéros de couplet sur différentes portées 292
Centrage vertical de paroles communes 293
Ensemble vocal (simple) 294
Ensemble vocal avec réduction pour piano 296
Ensemble vocal avec alignement des paroles selon le contexte 298
Ensemble vocal avec couplet et refrain 300

10 Keyboard and other multi-staff instruments 303

Symboles de registre pour accordéon 303
Modification du texte des indications de pédale 304
Clusters 304
Contrôler la position des doigtés dans un accord 304
Création d'une liaison entre plusieurs voix 305
Accord distribué et problème de hampe – solution 306
Trémolo et changement de portée 307

Affinage des indications de pédale 307
Indication d'un accord à cheval sur deux portées par un crochet 308
Symboles de jazz 308
Liaison « Laissez vibrer » 314
Piano – cannevas simple 315
Piano et paroles entre les portées 315
Piano, mélodie et paroles 316
Retrait de l'accolade à la première ligne d'une pièce pour piano 317
Utilisation d'`autoChange` avec plus d'une voix 318
Ensemble vocal avec réduction pour piano 319

11 Unfretted string instruments 322

Création d'une liaison entre plusieurs voix 322
Harmoniques pointées 322
Slap ou pizzicato Bartok 323
Quatuor à cordes (conducteur) 323
Quatuor à cordes, avec parties séparées 324

12 Fretted string instruments 328

Ajout de doigtés à une partition 328
Ajout de doigtés à des tablatures 328
Ajout de *markups* à une tablature 328
Impression des doigtés à l'intérieur de la portée 329
Barrés et diagrammes de fret automatiques 330
Orientation des diagrammes de fret 330
Changement d'accord et diagramme de fret 331
Glissando d'accords et tablature 332
Accords avec doigté espacé pour `FretBoards` et `TabVoice` 333
Contrôler la position des doigtés dans un accord 333
Personnalisation de diagrammes de fret 334
Personnalisation des diagrammes de fret 335
Doigtés, indications de corde, et doigtés main droite 337
Notation de flamenco 337
Construction et développement de diagrammes de fret 341
Diagrammes de fret alternatifs 348
Harmoniques et tablature 350
Indication d'un glissé de guitare 351
Rythmique et guitare 352
Hammer on et pull off 353
Hammer on et pull off avec accords 353
Hammer on et pull off gérés par les voix 354
Repositionnement d'un diagramme de fret 354
Symboles de jazz 355
Liaison « Laissez vibrer » 361
Impression de la corde aiguë d'un `TabStaff` en bas 362
Tablature en lettres 363
Harmoniques sur corde à vide en tablature 363
Positionnement des doigtés main droite 365
Polyphonie en mode tablature 366
Création de diagrammes de fret prédéfinis pour d'autres instruments 367
Glissando et tablature 369

Hampes et ligatures en mode tablature 370
Ligne de prolongation pour numéro de corde 370

13 Percussion 372

Ajout de parties de batterie 372
Exemple de sonnaille et clochette 373
Indication personnalisée d'une polymétrie complexe 373
Exemple de woodblocks haut et bas 374
Symboles de jazz 375
Baguettes pour percussion 380
Exemple pour percussions 383
Impression de musique aux métriques différentes 385
Exemple pour tam-tam 387
Exemple pour tambourin 387

14 Wind instruments 389

Modification de la taille d'un diagramme pour bois 389
Symboles de doigtés pour instruments à vent 389
Coup de langue à la flûte 390
Ajout de texte à un diagramme de doigté 390
Doigtés pour flûte à bec 391
Liste des différents diagrammes de doigtés pour bois 392
Liste des diagrammes de doigtés pour bois 393

15 Chord notation 395

Ajout d'une basse chiffrée au-dessus ou au-dessous des notes 395
Chiffrages et barres de mesure 395
Ajustement des glyphes d'altération en basse chifrée 396
Modification de la taille d'une note particulière d'un accord 396
Personnalisation du séparateur d'accords 396
Emplacement des altération en basse continue 397
Modèles de chiffrage d'accords 397
Chiffrage d'un maj7 398
Noms d'accord alternatifs 398
Accords avec doigté espacé pour `FretBoards` et `TabVoice` 408
Clusters 409
Contrôler la position des doigtés dans un accord 409
Accord distribué et problème de hampe – solution 410
Personnalisation du style de grille harmonique 411
Personnalisation du symbole *no-chord* 412
Impression d'accords complexes 412
Interruption manuelle des prolongations de certains chiffrages 412
Impression de noms d'accords identiques à la basse différente par une oblique et nouvelle basse 413
Impression des accords si changement 415
Chanson simple 416
Paroles, musique et accords 416
Paroles, musique, accords et diagrammes de fret 417
Mélodie simple et accords 418
Prolongateur commun de basse figurée 419
Crochet de reprise sous les chiffrages d'accord 419

16 Contemporary music 421

Débordement de ligature 421
Soufflet de crescendo partiellement interrompu 422
Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations` 422
Clusters 424
Glissando contemporain 424
Liaison de prolongation aplatie 425
Coup de langue à la flûte 427
Indication personnalisée d'une polymétrie complexe 428
Adjonction d'une croix sur la hampe des notes d'un fragment parlé 429
Armures inhabituelles 429
Impression de musique aux métriques différentes 430
Screech and boink 432
Moignons de hampe 434

17 Ancient notation 435

Ajout d'une basse chiffrée au-dessus ou au-dessous des notes 435
Gravure de musique ancienne 435
Exemples de notation ancienne – transcription moderne de musique grégorienne 438
Métrique ancienne 439
Notation pour psalmodie 439
Guidons 439
Incipit 440
Présentation à l'ancienne (barres de mesure entre les portées) 445
Styles de silences 445
Recours aux balises pour produire une partition ancienne et moderne à partir de la même source 447
Articulation baroque en forme de coche 449

18 World music 450

Improvisation en musique arabe 450
Exemple de musique « Makam » 450
Impression de texte de droite à gauche 450
Exemple de makam turc 451

Autres collections

19 Automatic notation 455

Subdivision de ligatures automatiques 455
Fixation arbitraire du numéro de repère de départ 455
Génération en Scheme de partitions complètes (y compris des parties d'ouvrage) sans utiliser l'analyseur 456
Suppression des bécarres superflus 458
Suppression des bécarres superflus lors d'un changement de tonalité 458
Ensemble vocal avec réduction pour piano 459

20 Breaks 462

Ajout d'une portée supplémentaire après un saut de ligne 462
Positionnement des segno et coda (avec saut de ligne) 463
Masquage de la première ligne si elle est vide 463

21 Connecting notes 466

Coexistence de ligatures et liaisons avec des rythmes comprenant des n-olets 466
Subdivision de ligatures automatiques 466
Modification de l'aspect des liaisons d'articulation 467
Contrôle de l'impression des crochets de n-olet 467
Création d'une liaison entre plusieurs voix 468
Liaison « Laissez vibrer » 469
Contrôle manuel du positionnement des ligatures 469

22 Contexts and engravers 470

Un ambitus par voix 470
Ajout d'une portée supplémentaire 470
Ajout d'une portée supplémentaire après un saut de ligne 471
Chiffrages et barres de mesure 472
Ambitus après l'armure 472
Crochet d'analyse avec texte 473
Changement automatique de l'orientation de hampe de la note médiane selon la mélodie 473
Affectation d'un canal MIDI par voix 474
Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations` 475
Arpège distribué sur plusieurs voix 476
Impression de papier à musique 477
Arpège distribué pour un autre contexte que le piano 479
Création d'armures personnalisées 480
Hampes interportées 481
Définition d'un graveur en Scheme : graveur d'ambitus 481
Affichage d'un système complet lorsqu'une seule portée est active 488
Les graveurs un par un 490
Apparence du quadrillage temporel 493
Quadrillage temporel : mise en évidence du rythme et synchronisation des notes 494
Compteur de mesures 495
Crochet de mesure 496
Présentation à l'ancienne (barres de mesure entre les portées) 497
Imbrications de regroupements de portées 498
Saut de ligne au milieu d'un n-olet avec ligature 499
Impression de noms d'accords identiques à la basse différente par une oblique et nouvelle basse 500
Impression des indications sur toutes les portées d'un système 502
Impression de musique aux métriques différentes 503
Suppression des numéros de mesure d'une partition 505
Indication de regroupement de portées par un rectangle 506
Utilisation de lignes de repères dans une partition « à la française » 506
Recours aux balises pour produire une partition ancienne et moderne à partir de la même source 509
Ensemble vocal avec couplet et refrain 511

Crochet de reprise sous les chiffrages d'accord 513
Crochets de *volta* sur plusieurs portées 514

23 Education 516
Quadrillage temporel : mise en évidence du rythme et synchronisation des notes 516
Empâtement de certaines lignes d'une portée 517

24 Headword 518
Ancient headword 518
Chords headword 521
Editorial headword 523
Expressive headword 524
Figured bass headword 525
Fretted headword 527
Keyboard headword 530
Pitches headword 534
Repeats headword 535
Rhythms headword 537
Simultaneous headword 539
Staff headword 541
Text headword 543
Unfretted headword 545
Vocal headword 548
Wind headword 550

25 MIDI 552
Affectation d'un canal MIDI par voix 552
Changement de tempo sans indication sur la partition 553
Création de nuance particulière pour la sortie MIDI 553
Démonstration des instruments MIDI 554
Réglage de l'égalisation par défaut des instruments MIDI 557

26 Non-music 559
Alignement des noms d'instrument 559
Champs d'entête 560
Liste des différents diagrammes de doigtés pour bois 561

27 Paper and layout 563
Alignement des noms d'instrument 563
Agencement de paroles séparées sur une seule ligne 564
Partitionnement d'un ouvrage 565
Modification de la taille d'une portée 569
Découpe de systèmes 570
Impression de papier à musique 572
Champs d'entête 574
Affichage d'un système complet lorsqu'une seule portée est active 575
Séparation visuelle entre les systèmes 577
Table des matières 578
Alignement vertical de `StaffGroups` sans `SystemStartBar` 580

28 Preparing parts 587
Fixation arbitraire du numéro de repère de départ 587
Dénombrer une unique mesure de silence 587
Quatuor à cordes, avec parties séparées 587

29 Real music 591
Affectation d'un canal MIDI par voix 591
Création d'une séquence de notes de même hauteur 592
Création d'une liaison entre plusieurs voix 592
Trémolo et changement de portée 593
Démonstration des instruments MIDI 593
Harmoniques pointées 597
Indication personnalisée d'une polymétrie complexe 597
Indication d'un accord à cheval sur deux portées par un crochet 598
Insertion d'un fragment dans un *markup* au-dessus de la portée 599
Exemple pour percussions 600
Impression de musique aux métriques différentes 601

30 Really cool 604
Ajout de la date du jour à une partition 604
Blanchiment de lignes de portée avec la commande `\whiteout` 604
Texte centré sous un soufflet 605
Modification des propriétés d'objets particuliers 606
Clusters 607
Coloration des notes selon leur hauteur 607
Création d'une séquence de notes de même hauteur 608
Génération de notes aléatoires 608
Génération en Scheme de partitions complètes (y compris des parties d'ouvrage) sans utiliser l'analyseur 609
Empâtement de certaines lignes d'une portée 611
Armures inhabituelles 612
Impression de musique aux métriques différentes 612

31 Really simple 616
Ajout d'une portée supplémentaire 616
Ajout de parties de batterie 616
Ajout de doigtés à une partition 617
Alignement de repère sur une note 617
Crochets d'analyse au-dessus de la portée 618
Modification de la taille d'une note particulière d'un accord 618
Changement de fontes des couplets 619
Modification de l'aspect des liaisons d'articulation 619
Combinaison de nuance et de texte 620
Combinaison de nuance et de texte (2) 620
Fixation arbitraire du numéro de repère de départ 621
Alignement des syllabes 621
Fusion de silences multimesures dans une partie polyphonique 622
Modification de la longueur d'un crochet de n-olet 622
Impression du numéro de version 623
Piano – cannevas simple 623

Piano et paroles entre les portées 623
Piano, mélodie et paroles 624
Paroles, musique et accords 625
Mélodie simple et accords 626
Portée unique et paroles 627
Portée unique avec quelques notes 627
Sauts de notes en mode paroles 628
Sauts de notes en mode paroles (2) 628
Quatuor à cordes (conducteur) 628
Utilisation de `\tweak` pour retoucher des objets particuliers 630
Ensemble vocal (simple) 630
Crochets de *volta* sur plusieurs portées 632

32 Scheme 633

Ajout d'un doigté supplémentaire avec Scheme 633
Ajout d'indicateurs à une portée dédoublée après un saut de ligne 633
Ajout de liens à des objets 637
Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur 639
Ajout de la date du jour à une partition 641
Ajustement du positionnement vertical d'une liaison 642
Texte centré sous un soufflet 642
Modification des propriétés d'objets particuliers 644
Modèles de chiffrage d'accords 645
Coloration des notes selon leur hauteur 646
Création d'une séquence de notes de même hauteur 647
Création de nuance particulière pour la sortie MIDI 647
Contrôle de la position et du nombre de points d'une barre de reprise 648
Définition d'un graveur en Scheme : graveur d'ambitus 649
Taille de police différente pour `instrumentName` et `shortInstrumentName` 656
Affichage de la généalogie d'un objet 657
Encerclement de notes 659
Encercler divers objets 659
Rappel du glissando à l'occasion d'une alternative 660
Liaison de prolongation aplatie 661
Coup de langue à la flûte 664
Diagrammes de fret alternatifs 664
Génération de notes profilées spéciales 666
Génération de crochets personnalisés 667
Génération de notes aléatoires 668
Génération en Scheme de partitions complètes (y compris des parties d'ouvrage) sans utiliser l'analyseur 668
Répétition en pourcent isolée 671
Easy play – chiffres en lieu et place des lettres 671
Personnalisation de certains types d'articulation 672
Positionnement des notes d'ornement avec espace flottant 674
Impression de noms d'accords identiques à la basse différente par une oblique et nouvelle basse 675
Réglage de l'égalisation par défaut des instruments MIDI 677
Séparation entre altérations annulées et nouvelle armure 678
Indications de nuance entre parenthèses en italique 679
Ligne de prolongation pour numéro de corde 679

Encadrement sur trois côtés ... 680
Transposition et réduction du nombre d'altérations accidentelles ... 681
Deux \partCombine sur une même portée ... 682
Personnalisation de la métrique ... 684
Utilisation de ly:grob-object pour accéder aux *grobs* avec \tweak ... 684
Articulation baroque en forme de coche ... 685

33 Spacing ... 687

Ajustement de l'espacement vertical des paroles ... 687
Impression des doigtés à l'intérieur de la portée ... 687
Rupture d'alignement horizontal des nuances graphiques et textuelles ... 688
Rupture d'alignement vertical des nuances graphiques et textuelles ... 689
Harmonizing bar line thickness for staves with different sizes ... 689
Référencement de page ... 691
Espacement strictement proportionnel des notes ... 692
Alignement vertical des nuances et indications textuelles ... 693
Positionnement d'une ossia et des paroles ... 693

34 Specific notation ... 695

Symboles de registre pour accordéon ... 695
Chiffrages et barres de mesure ... 696
Ajout de parties de batterie ... 696
Ajout de doigtés à des tablatures ... 697
Variante fine des notes profilées Aiken ... 697
Impression des doigtés à l'intérieur de la portée ... 698
Modification du nombre de lignes de la portée ... 698
Notation pour psalmodie ... 699
Modèles de chiffrage d'accords ... 699
Chiffrage d'un maj7 ... 700
Accords avec doigté espacé pour FretBoards et TabVoice ... 701
Clusters ... 701
Glissando contemporain ... 702
Contrôler la position des doigtés dans un accord ... 702
Exemple de sonnaille et clochette ... 703
Impression de papier à musique ... 703
Guidons ... 705
Démonstration des instruments MIDI ... 706
Orientation de la tête d'un « fa » profilé en cas de fusion ... 710
Intégration de PostScript dans un *markup* ... 710
Les graveurs un par un ... 711
Notation de flamenco ... 713
Exemple de woodblocks haut et bas ... 717
Repositionnement d'un diagramme de fret ... 718
Séparation de syllabes par une liaison ... 719
Liaison « Laissez vibrer » ... 719
Exemple pour percussions ... 719
Exemple pour tam-tam ... 720
Exemple pour tambourin ... 721
Métrique entre crochets ... 721
Métrique entre parenthèses ... 721
Recours à une voix supplémentaire pour gérer les sauts ... 722

Liste des diagrammes de doigtés pour bois 723

35 Symbols and glyphs 725

Symboles de registre pour accordéon 725
Ajout d'indicateurs à une portée dédoublée après un saut de ligne 726
Gravure de musique ancienne 730
Signes de respiration 733
Soufflet de crescendo partiellement interrompu 734
Césure en forme de « voie de chemin de fer » avec point d'orgue 734
Guidons 735
Contrôle de la position et du nombre de points d'une barre de reprise 736
Symboles de doigtés pour instruments à vent 737
Séparation de syllables par une liaison 738
Positionnement des segno et coda (avec saut de ligne) 738
Styles de silences 739
Indication de reprise avec texte grâce à `repeatCommands` 740

36 Templates 741

Exemples de notation ancienne – transcription moderne de musique grégorienne 741
Psalmodie anglicane 741
Modèle pour cantique 744
Symboles de jazz 746
Orchestre, chœur et piano 752
Piano – cannevas simple 757
Piano et paroles entre les portées 757
Piano, mélodie et paroles 758
Modèle pour chœur SATB, sur quatre portées 759
Paroles, musique et accords 761
Paroles, musique, accords et diagrammes de fret 762
Mélodie simple et accords 763
Portée unique et paroles 763
Portée unique avec quelques notes 764
Quatuor à cordes (conducteur) 765
Quatuor à cordes, avec parties séparées 766
Ensemble vocal (simple) 768
Ensemble vocal avec réduction pour piano 770
Ensemble vocal avec alignement des paroles selon le contexte 772
Ensemble vocal avec couplet et refrain 774

37 Titles 777

Ajout de la date du jour à une partition 777
Alignement des noms d'instrument 777
Champs d'entête 779
Impression du numéro de version 780

38 Tweaks and overrides 781

Ajout d'une indication d'octave pour une seule voix 781
Ajout de liens à des objets 781
Ajout de *markups* à une tablature 783
Ajout de marques temporelles à un long glissando 784

Ajustement de l'espacement des notes d'ornement 784
Ajustement du positionnement vertical d'une liaison 785
Ajustement de l'espacement vertical des paroles 786
Alignement de repère sur une note 786
Modification de la hauteur de hampes ligaturées 787
Numérotation des mesures et alternatives 787
Crochets d'analyse au-dessus de la portée 789
Crochet d'analyse avec texte 789
Liaison asymétrique 790
Rupture d'alignement horizontal des nuances graphiques et textuelles 790
Rupture d'alignement vertical des nuances graphiques et textuelles 791
Césure en forme de « voie de chemin de fer » avec point d'orgue 792
Modification de la taille d'une note particulière d'un accord 792
Modification de l'épaisseur et de l'écartement des ligatures 793
Modifier l'apparence d'un silence multimesure 793
Modification des propriétés d'objets particuliers 793
Modification du texte et de l'extension de nuances textuelles 794
Changement des fontes textuelles par défaut 794
Modification de la taille d'une portée 795
Changement de tempo sans indication sur la partition 796
Modification du texte des indications de pédale 796
Contrôle de la visibilité des extensions d'objet après saut de ligne 797
Contrôle de l'apparence des traits de trémolo 798
Contrôle de l'ordre vertical des articulations et ornements 799
Contrôle de l'impression des crochets de n-olet 800
Création d'un grupetto retardé 801
Création d'armures personnalisées 801
Création d'extensions de texte 802
Accord distribué et problème de hampe – solution 803
Hampes interportées 803
Guidons 804
Personnalisation de diagrammes de fret 805
Personnalisation des diagrammes de fret 806
Indicateur de regroupement et portée unique 808
Affichage de la généalogie d'un objet 809
Harmoniques pointées 811
Encadrement d'objets 811
Encercler divers objets 811
Personnalisation des extenseurs de nuance textuelle 812
Allongement d'une marque de trille (`TrillSpanner`) 813
Rappel du glissando à l'occasion d'une alternative 813
Affinage des indications de pédale 814
Liaison de prolongation aplatie 815
Impression forcée d'un bécarre avant une accidentelle 817
Décalage horizontal forcé 818
Construction et développement de diagrammes de fret 818
Génération de notes profilées spéciales 825
Génération de crochets personnalisés 826
Glissando par dessus un objet graphique 827
Stylisation des lignes de soufflet 827
Alignement de nuances personnalisées comme « più f » 828
Repositionnement d'un diagramme de fret 831

Insertion d'une césure 832
Conservation de la taille lors d'un changement de clef 833
Terminaison de ligne en flèche 833
Recours à la propriété `transparent` pour rendre des objets invisibles 834
Saut de ligne et glissando 834
Contrôle manuel du positionnement des ligatures 836
Centrage des numéros de mesure 836
Présentation à l'ancienne (barres de mesure entre les portées) 837
Modification de la pente de l'extension d'octaviation 837
Déplacement des notes pointées dans une polyphonie 838
Déplacement vertical des liaisons d'articulation 838
Imbrications de regroupements de portées 839
Personnalisation de certains types d'articulation 840
Affichage du numéro de répétition en pourcent 842
Positionnement des arpeggios 842
Positionnement précis des indications de doigté 842
Positionnement des silences multimesures 843
Positionnement d'une annotation à l'intérieur d'une liaison 844
Inscrire le numéro de mesure dans un cadre ou un cercle 844
Impression du métronome et des repères sous la portée 845
Impression du nom des notes avec ou sans indication d'octave 845
Affichage du crochet de n-olet du côté des têtes de note 846
Espacement strictement proportionnel des notes 847
Retrait de l'accolade à la première ligne d'une pièce pour piano 847
Suppression de la partie interportée des barres de mesure d'un regroupement autre que `ChoirStaff` 848
Masquage de la première ligne si elle est vide 849
Styles de silences 850
Barres rythmiques 851
Séparation entre altérations annulées et nouvelle armure 852
Soufflets et barres de mesure 852
Séparation visuelle entre les systèmes 853
Galbe individuel des liaisons d'un accord 854
Impression d'une même articulation des deux côtés d'une note ou d'un accord 857
Ligne de prolongation pour numéro de corde 858
Suppression des avertissements de chevauchement 858
Métrique entre crochets 858
Métrique entre parenthèses 859
Affichage seulement du numérateur d'une métrique (au lieu d'une fraction) 859
Crochet de n-olet et changement de portée 860
Affinage des propriétés d'une clef 860
Mise en forme des notes d'ornement 862
Crochets de style alternatif 862
Utilisation de `ly:grob-object` pour accéder aux *grobs* avec `\tweak` 864
Utilisation de `\tweak` pour retoucher des objets particuliers 865
Alignement vertical des nuances et indications textuelles 865
Positionnement d'une ossia et des paroles 866
Alignement vertical des numéros de couplet sur différentes portées 867
Prolongateur commun de basse figurée 868

39 Workaround 870
Ajout d'une portée supplémentaire après un saut de ligne 870
Appoggiature avant une barre de mesure 871
Rupture d'alignement horizontal des nuances graphiques et textuelles 871
Rupture d'alignement vertical des nuances graphiques et textuelles 872
Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations` 873
Accord distribué et problème de hampe – solution 874
Impression d'accords complexes 875
Rappel du glissando à l'occasion d'une alternative 876
Adaptation de la largeur de mesure selon le `MetronomeMark` 877
Empâtement de certaines lignes d'une portée 878
Adjonction d'une croix sur la hampe des notes d'un fragment parlé 878
Positionnement des notes d'ornement avec espace flottant 879
Positionnement des segno et coda (avec saut de ligne) 879
Préservation de l'indication de n-olet lors d'un repère final 880
Impression de texte de droite à gauche 881
Indications de nuance entre parenthèses en italique 881
Exemple pour tambourin 882
Transposition et réduction du nombre d'altérations accidentelles 882
Expansion de répétition de trémolo 884
Recours à une voix supplémentaire pour gérer les sauts 885
Alignement vertical des nuances et indications textuelles 886

Préface

Ce document regroupe un certain nombre d'exemples, trucs et astuces pour LilyPond issus du LilyPond Wiki (`https://wiki.lilypond.community`), successeur du LilyPond Snippet Repository (LSR).

Nous profitons de ces lignes pour adresser tous nos remerciements à Sebastiano Vigna pour la maintenance du site LSR et à Jean Abou Samra pour s'occuper du LilyPond Wiki.

Notez bien que cet ouvrage n'est pas une simple copie du LilyPond Wiki. D'une part, certains exemples proviennent directement des sources de LilyPond – plus précisément le répertoire `Documentation/snippets/new/` – et, d'autre part, ce qui provient du Wiki est mis à jour à l'aide de `convert-ly` – LilyPond Wiki repose principalement sur une version stable de LilyPond, alors que ce manuel correspond à la version 2.27.4.

Les extraits sont regroupés par catégorie (`https://wiki.lilypond.community/wiki/Special:Categories`), correspondant dans la mesure du possible au découpage du manuel de notation de LilyPond. Un même extrait peut se voir attribuer plusieurs catégories, et toutes les catégories du Wiki ne sont pas forcément reprises.

Tous les extraits ici présentés voient l'indentation des systèmes réduite à zéro.

Si vous consultez la version HTML de cet ouvrage, un clic sur le nom du fichier ou sur l'image vous affichera le code sous-jacent.

notation musicale
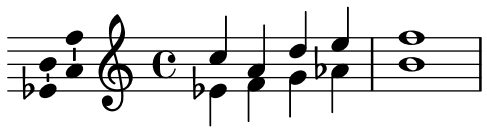
1 Pitches

See also Section “Pitches” dans *Manuel de notation*.

Un ambitus par voix

L’ambitus peut être individualisé par voix. Il faut en pareil cas éviter qu’ils se chevauchent.

```
\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
  }
>>
```



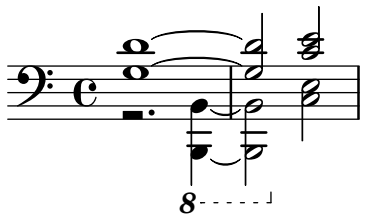
Ajout d’une indication d’octave pour une seule voix

Lorsque plusieurs voix cohabitent sur une même portée, déterminer l’octaviation d’une voix affecte la position des notes de toutes les voix, jusqu’à la fin du crochet d’octaviation. Si l’octaviation ne doit s’appliquer qu’à une seule voix, le Ottava_spanner_engraver devrait être déplacé dans le contexte 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
    }
  >>
}
```
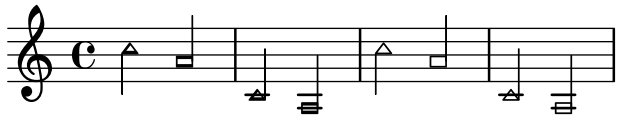


Variante fine des notes profilées Aiken

Les notes profilées Aiken, lorsqu'elles sont blanches, deviennent difficile à distinguer avec des tailles de portée réduites, notamment en présence de ligne supplémentaire. Perdant du blanc sur leur intérieur les fait alors ressembler à des noires.

```
\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
  }
}
```

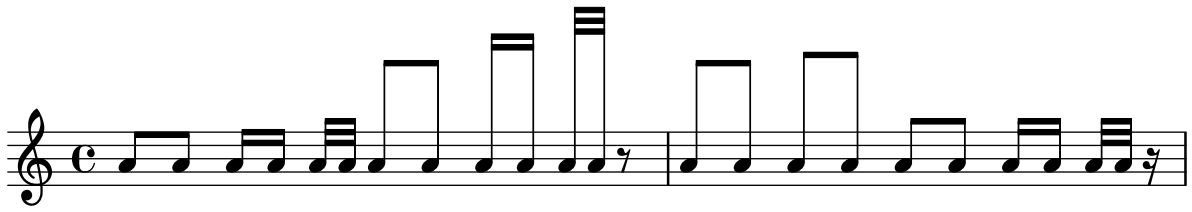


Modification de la hauteur de hampes ligaturées

La hauteur de hampe des notes ligaturées est gérée par la sous-propriété `beamed-lengths` des `details` de l'objet `Stem`. Lorsqu'elle ne comporte qu'une seule valeur, cette hauteur s'appliquera à toutes les hampes. En présence de plusieurs arguments, le premier affectera les croches, le second les doubles croches, et ainsi de suite. Le dernier argument s'appliquera aussi aux notes de plus courte durée. Les arguments peuvent être des valeurs décimales.

```
\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 |
}
```
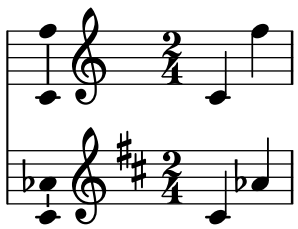


Ambitus

Un *ambitus* indique la tessiture, autrement dit les hauteurs extrêmes d'une voix.

Seules seront affichées les altérations non comprises dans l'armure. Les objets `AmbitusNoteHead` peuvent avoir des lignes supplémentaires.

```
\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'
    }
  }
>>
```
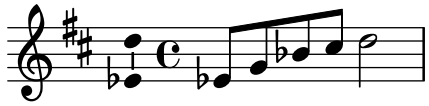


Ambitus après l'armure

L'ambitus se place par défaut à gauche de la clef. La fonction `\ambitusAfter` permet cependant de modifier ce positionnement ; sa syntaxe est `\ambitusAfter` *grob-interface* – voir Section "Graphical Object Interfaces" dans *Référence des propriétés internes* pour une liste des valeurs de *grob-interface* possibles.

L'un des cas d'usage est d'insérer l'ambitus entre l'armure et la métrique.
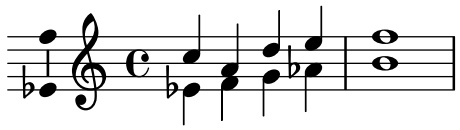
```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



Ambitus sur plusieurs voix

Si plusieurs voix se trouvent sur une même portée, on peut attribuer le graveur `Ambitus_engraver` au contexte `Staff` afin d'obtenir l'ambitus sur toutes les voix cumulées, non d'une seule des voix actives.

```
\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
  }
>>
```



Profilage des notes selon leur degré dans la gamme

La propriété `shapeNoteStyles` permet d'affecter un profil particulier à chaque degré de la gamme – à partir de l'armure ou de la propriété `tonic`. Ses valeurs sont constituées d'une liste de symboles, qu'il s'agisse de formes géométriques (`triangle`, `cross` ou `xcircle`) ou basés sur la tradition des graveurs américains (avec quelques noms de note latins).

LilyPond dispose de deux raccourcis, `\aikenHeads` et `\sacredHarpHeads`, permettant de reproduire d'anciens recueils de chansons américaines.

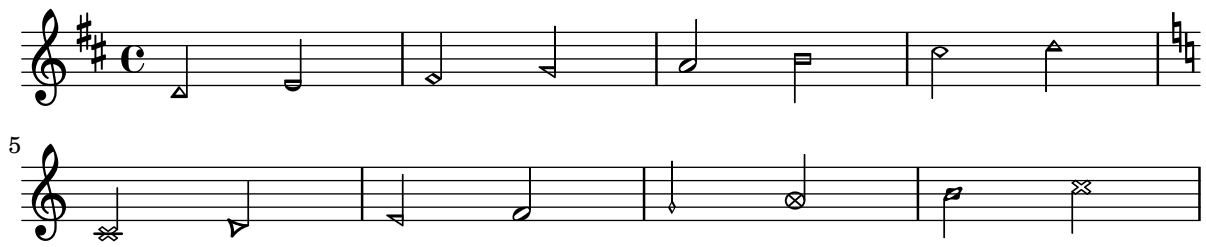
L'exemple suivant montre plusieurs manières de profiler les têtes de note, ainsi que la capacité de transposer tout en respectant la fonction harmonique de chaque note dans la gamme.

```
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
  }
}
```



Changement automatique de l'orientation de hampe de la note médiane selon la mélodie
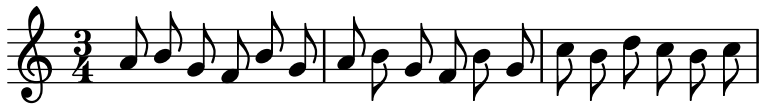
Afin de suivre la ligne mélodique, LilyPond peut inverser l'orientation de hampe de la note médiane, dès lors qu'aura été ajouté au contexte de voix le graveur `Melody_engraver`.

La propriété de contexte `suspendMelodyDecisions` permet, si besoin, de désactiver temporairement ce comportement.

```
\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
  }
}
```

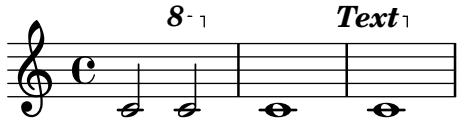


Modification du texte des marques d'octaviation

En interne, la fonction `\ottava` détermine les propriétés `ottavation` (par ex. en `"8va"` ou `"8vb"`) et `middleCPosition`. Vous pouvez modifier le texte d'une marque d'octaviation en définissant `ottavation` après avoir fait appel à `ottava`.

Un texte bref est particulièrement utile lorsque l'octaviation est courte.

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



Réglage de l'affichage d'un ambitus

L'affichage d'un *ambitus* peut s'affiner pour répondre à vos préférences en matière d'esthétique.

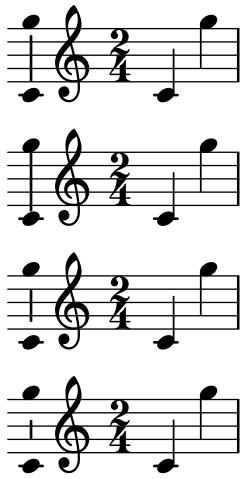
```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\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''
}
```

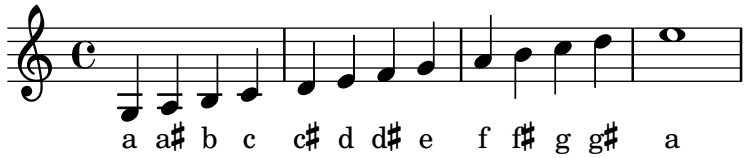


Modification de l'intervalle des lignes de la portée

staffLineLayoutFunction permet de changer le positionnement des notes. Dans cet exemple, la valeur qui lui est attribuée – ly:pitch-semitones – génère une gamme chromatique où l'écart entre chaque ligne et interligne de la portée est ramené au demi ton.

```
scale = \relative c' {
  a4 ais b c
  cis4 d dis e
  f4 fis g gis
  a1
}
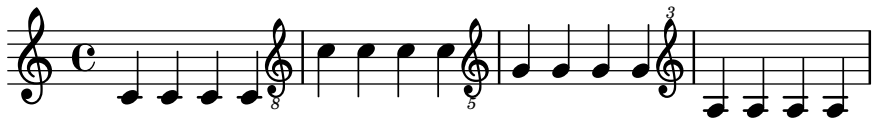
\new Staff \with {
  \remove "Accidental_engraver"
  staffLineLayoutFunction = #ly:pitch-semitones
}
{
  <<
    \scale
    \context NoteNames {
      \set printOctaveNames = ##f
      \scale
    }
  >>
}
```



Transposition arbitraire d'une clef

Les clefs peuvent être transposées d'un intervalle différent de l'octave.

```
\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
}
```



Coloration des notes selon leur hauteur

Les têtes de note peuvent adopter une couleur différente selon leur hauteur ou leur nom ; la fonction utilisée ici fait même la distinction entre enharmoniques.
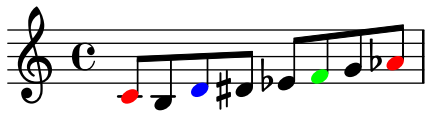
```
% Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (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
  }
}
```
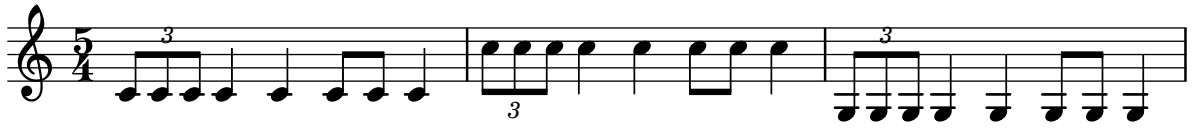


Création d'une séquence de notes de même hauteur

Lorsque la musique comporte à de nombreuses reprises une même séquence sur des hauteurs différentes, la fonction musicale ci-dessous peut s'avérer fort utile. Elle considère une note dont seule la hauteur est utilisée.

Cet exemple reproduit le rythme utilisé tout au long de « Mars », l'une des pièces de l'œuvre de Gustav Holst « Les planètes ».

```
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
}
```
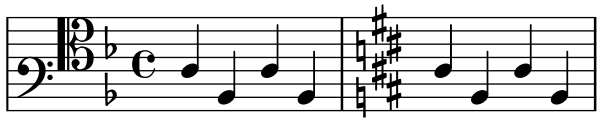


Création d'armures personnalisées

LilyPond accepte les armures personnalisées. Dans cet exemple, il s'agit d'une tonalité de ré mineur et ré majeur auxquelles on rajoute des bémols.

```
\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, |
}
```
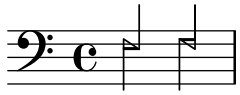


Orientation de la tête d'un « fa » profilé en cas de fusion

Grâce à la propriété `NoteCollision.fa-merge-direction`, l'orientation de la tête des notes profilées « fa » (`fa`, `faThin`, etc.) peut se gérer indépendamment de la direction des hampes lorsque deux voix ayant la même hauteur et des directions de hampes différentes sont fusionnées. Lorsque cette propriété n'est pas activée sera utilisée la variante « descendante ».

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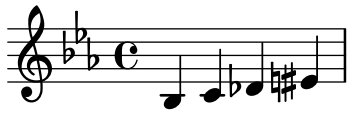
  << { \aikenHeads
       f2
       \override Staff.NoteCollision.fa-merge-direction = #UP
       f2 }
  \\ { \aikenHeads
       f2
       f2 }
  >>
}
```



Impression forcée d'un bécarre avant une accidentelle

L'exemple suivant indique comment forcer l'impression d'un bécarre avant un changement d'altération.
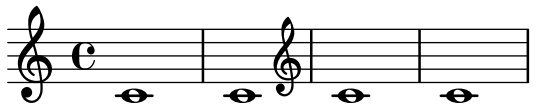
```
\relative c' {
  \key es \major
  bes c des
  \tweak Accidental.restore-first ##t
  eis
}
```



Impression forcée de la clef

Lorsqu'une clef a déjà été imprimée et qu'aucun changement n'est intervenu, LilyPond ignore toute répétition de la commande `\clef` puisque le signe est le même. Il est cependant possible de rappeler la clef à s'imprimer à l'aide de l'instruction `\set Staff.forceClef = ##t`.

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



Génération de notes aléatoires
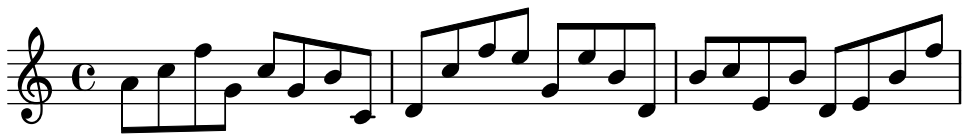
Le fragment de code Scheme ci-dessous génère aléatoirement des notes. Il s'utilise ainsi :

```
\randomNotes n de à dur
```

pour générer *n* notes dont la hauteur varie entre *de* et *à* sur une durée de *dur*.

```
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)))))
```

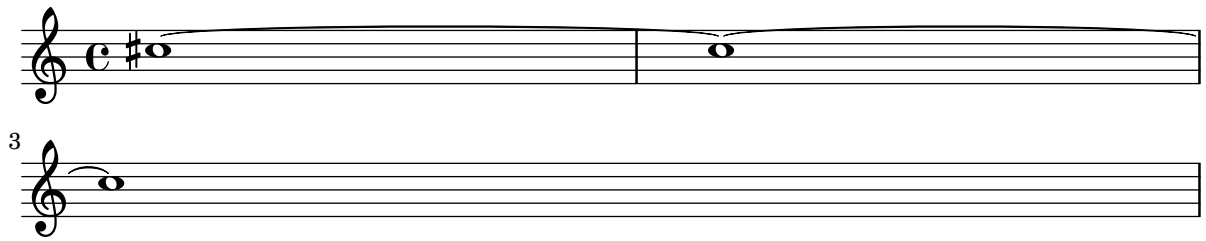
```
\randomNotes 24 c' g'' 8
```



Non répétition de l'altération après saut de ligne sur liaison de prolongation

Cet exemple illustre comment, lorsqu'une note affublée d'une altération accidentelle est prolongée, ne pas répéter cette altération après un saut de ligne.
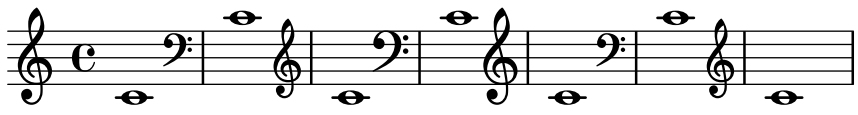
```
\relative c'' {
  \override Accidental.hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```



Conservation de la taille lors d'un changement de clef

Lorsqu'un changement de clef intervient, le nouveau signe s'imprime dans une taille inférieure. On peut y déroger en activant la propriété de contexte `full-size-change`.

```
\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
}
```
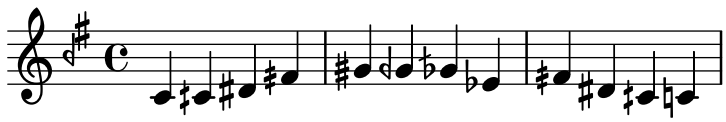


Exemple de musique « Makam »

Le « makam » est une forme de mélodie turque qui utilise des altérations d'un neuvième de ton.

Consultez le fichier d'initialisation `ly/makam.ly` pour plus de détails sur les hauteurs et altérations utilisées (voir le chapitre 5.7.4 - Autres sources d'information du manuel d'initiation pour le localiser).

```
\include "makam.ly"

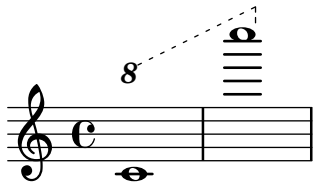
\relative c' {
  \set Staff.keyAlterations = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Modification de la pente de l'extension d'octaviation

Il est possible d'adapter la pente d'une indication d'octaviation.

```
\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
}
```



Armures inhabituelles

La commande `\key` détermine la propriété `keyAlterations` d'un contexte `Staff`.

Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété. Il s'agit en l'occurrence de définir une liste :

```
\set Staff.keyAlterations =
  #`(((octave . pas) . altération) ((octave . pas) . altération) ...)
```
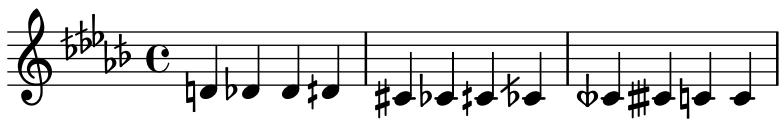
dans laquelle, et pour chaque élément, *octave* spécifie l'octave (0 pour celle allant du do médium au si supérieur), *pas* la note dans cette octave (0 pour do et 6 pour si), et *altération* sera ,SHARP ou ,FLAT ou ,DOUBLE-SHARP, etc. (attention à la virgule en préfixe).

Une formulation abrégée – (*pas* . *altération*) – signifie que l'altération de l'élément en question sera valide quelle que soit l'octave.

En ce qui concerne les gammes microtonales dans lesquelles un « dièse » n'est pas d'un centième, *altération* se réfère à un deux-centième de ton entier.

```
\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 |
}
```



Easy play – chiffres en lieu et place des lettres

En mode « easy play », les têtes de note utilisent la propriété note-names attachée à l'objet NoteHead pour déterminer ce qui apparaîtra dans la tête. Intervenir sur cette propriété permet d'imprimer un chiffre correspondant au degré dans la gamme.

La création d'un graveur dédié permet de traiter toutes les notes.

```
#(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))))))

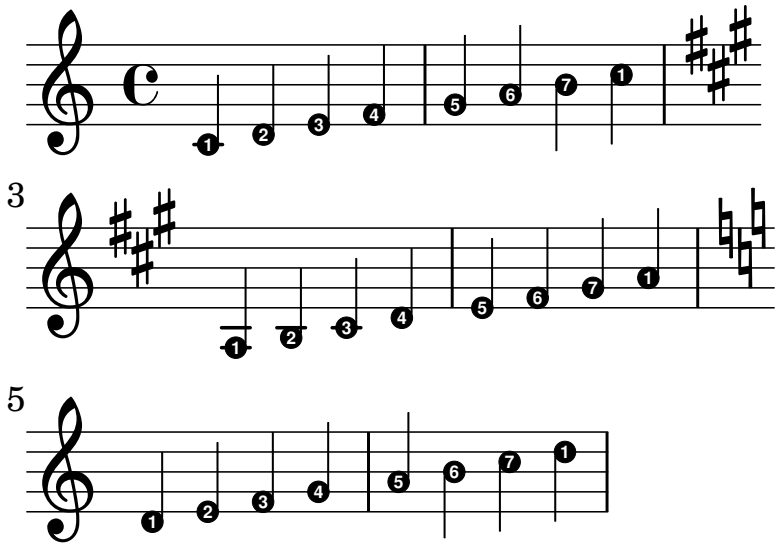
#(set-global-staff-size 30)
```

```
\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
}
```



Orchestre, chœur et piano

Ce canevas illustre l'utilisation de contextes `StaffGroup` pour regrouper les instruments selon leur famille, imbriqués dans un `GrandStaff`, ainsi que le recours à la fonction `\transpose` pour les instruments transpositeurs. Dans tous les identificateurs, la musique est stockée en ut. Les notes peuvent tout aussi bien être saisies en ut ou dans la tonalité particulière de l'instrument avant d'être transposées puis affectées à une variable.

```
#(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 }
        >>

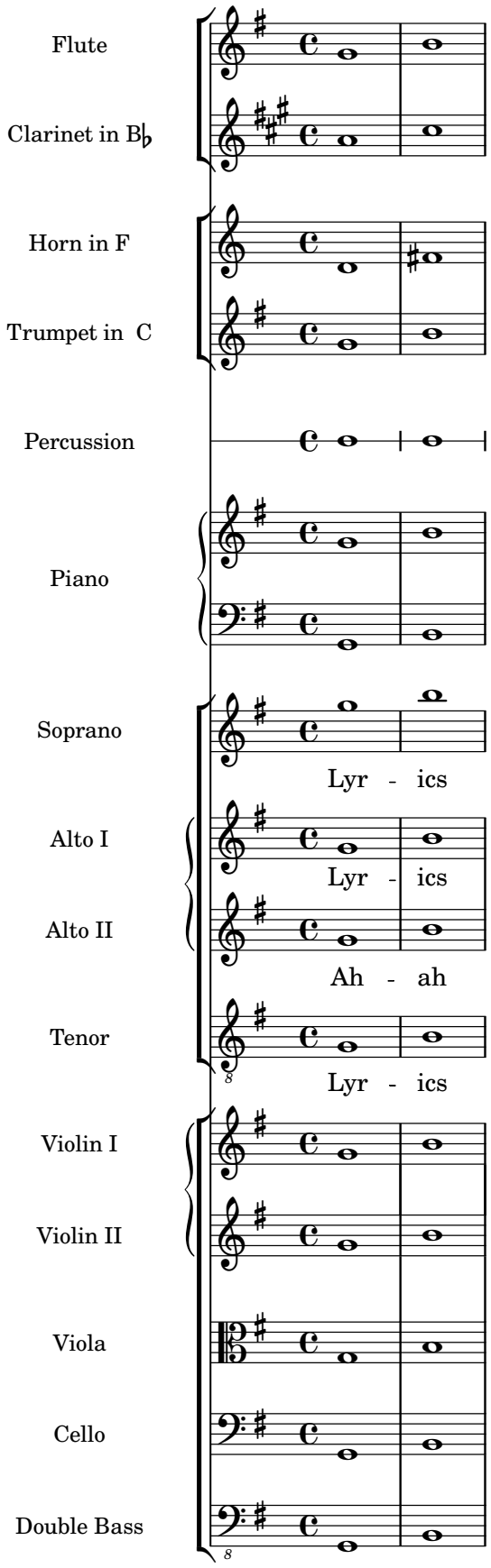
        \new Staff = "Staff_tenor" \with {
          instrumentName = "Tenor"
        }
          \new Voice = "tenor" \tenorMusic
          \new Lyrics \lyricsto "tenor" { \tenorLyrics }
      >>

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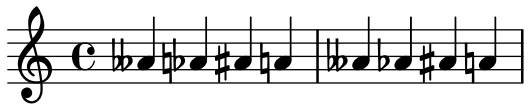
        \new Staff = "Staff_bass" \with {
          instrumentName = "Double Bass"
        }
          \bassMusic
      >>
    >>
  }
}
```


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

Suppression des bécarres superflus

En accord avec les règles traditionnelles de l'écriture musicale, on grave un bécarre avant un dièse ou un bémol si la note était auparavant affublée d'un double-dièse ou double-bémol. Pour adopter un comportement plus contemporain, la propriété `extraNatural` du contexte `Staff` doit se voir attribuer la valeur `#f` (faux).
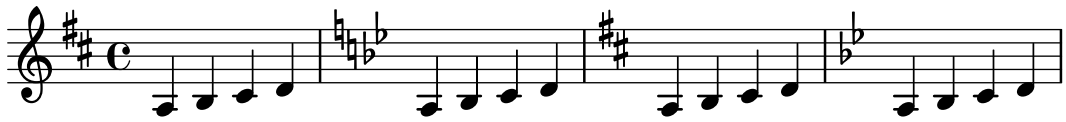
```
\relative c'' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Suppression des bécarres superflus lors d'un changement de tonalité

Après un changement de tonalité, un bécarre est imprimé pour annuler toute altération précédente. Ce comportement s'annule en désactivant la propriété `printKeyCancellation` du contexte `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
}
```



Citation d'une autre voix et transposition

Les citations tiennent compte de la transposition, aussi bien celle de l'instrument d'origine que celle de la partie où elles interviennent. Dans l'exemple suivant, tous les instruments sont en tonalité de concert et seront repris par un instrument en fa. Le destinataire de la citation peut à son tour transposer à l'aide de la commande `\transpose`. En pareil cas, toutes les hauteurs seront transposées, y compris celles de la citation.
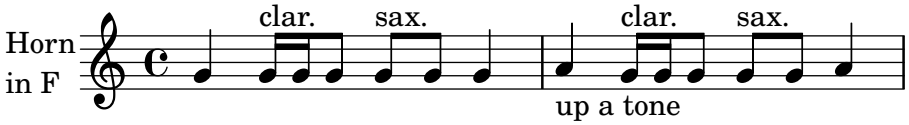
```
\addQuote clarinet {
  \transposition bes
  \*8 { d'16 d' d'8 }
}

\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" >>
}
```



Séparation entre altérations annulées et nouvelle armure

Les altérations supprimées lors d'un changement de tonalité sont par défaut accolées à la nouvelle armure. Ce comportement peut s'adapter grâce à la propriété `break-align-orders` de l'objet `BreakAlignment`.

Si l'on se réfère à la référence des propriétés internes à propos de l'objet Section "break-alignment-interface" dans *Référence des propriétés internes*, l'ordre du deuxième élément est :

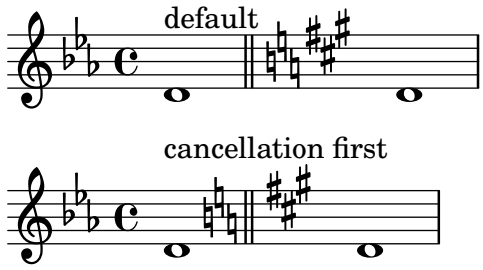
```
ambitus
breathing-sign
clef
cue-clef
cue-end-clef
custos
key-cancellation
key-signature
left-edge
signum-repetitionis
staff-bar
staff-ellipsis
time-signature
```

La consultation de cette liste nous apprend qu'il faut déplacer `key-cancellation` pour le mettre avant `staff-bar`. Ceci se réalise à l'aide de la fonction `\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 }
```



Transposition et réduction du nombre d'altérations accidentelles

Cet exemple, grâce à un peu de code Scheme, donne la priorité aux enharmoniques afin de limiter le nombre d'altérations supplémentaires. La règle applicable est :

- Les altérations doubles sont supprimées
- Si dièse -> Do
- Mi dièse -> Fa
- Do bémol -> Si
- Fa bémol -> Mi

Cette façon de procéder aboutit à plus d'enharmoniques naturelles.

```
#(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)))
```
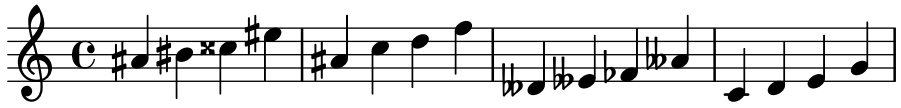
```
     (ly:make-pitch o n (/ a 4))))

#(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 }
}
```



Exemple de makam turc

Ce canevas utilise le début d'un *saz semai* turc bien connu du répertoire aux fins d'illustrer certains éléments de la notation musicale turque.

```
#(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
```

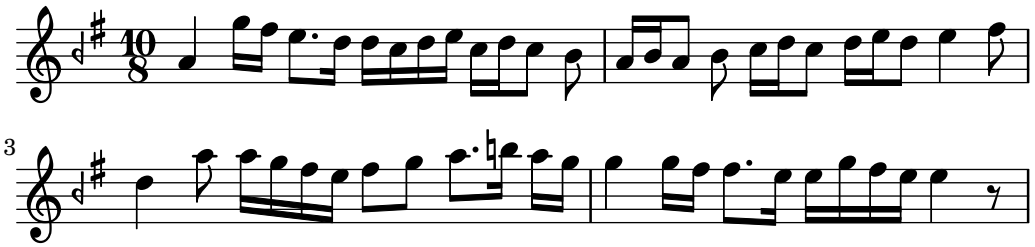
```
  \key a \huseyni
  \time 10/8

  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



Affinage des propriétés d'une clef

Modifier le glyphe, la position de la clef ou son octaviation ne changeront pas la position des notes ; il faut pour y parvenir modifier aussi la position du do médium. La redéfinition préalable de `middleCClefPosition` permet de placer l'armure sur les bonnes lignes. Le positionnement est relatif à la ligne médiane, un nombre positif faisant monter, un nombre négatif abaissant.

Par exemple, la commande `\clef "treble_8"` équivaut à définir les propriétés de contexte `clefGlyph`, `clefPosition` – qui contrôle la position verticale de la clef –, `middleCPosition` et `clefOctavation`. Une nouvelle clef apparaîtra dès lors que l'une de ces propriétés, à l'exception de `middleCPosition`, aura été modifiée.

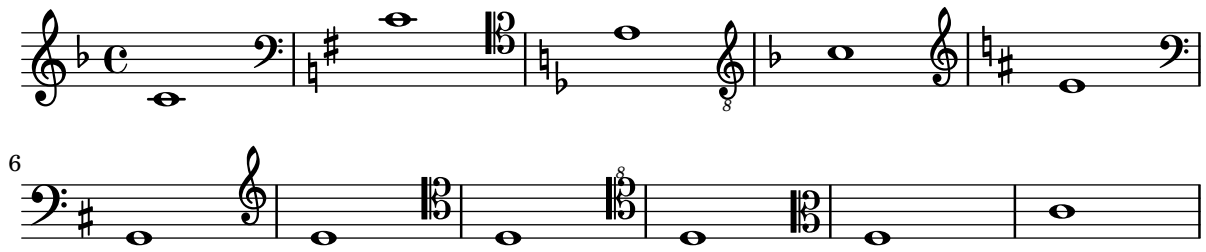
Les exemples qui suivent illustrent les différentes possibilités de définir ces propriétés manuellement. Sur la première ligne, la position relative des notes par rapport aux clefs est préservée, ce qui n'est pas le cas pour la deuxième ligne.

```
{
  % 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
}
```



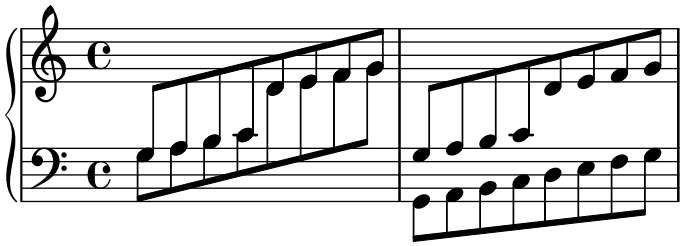
Utilisation d'autoChange avec plus d'une voix

L'instruction \autoChange est opérationnelle y compris en présence de voix multiples.

```
\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 Section “Rhythms” dans *Manuel de notation*.

Coexistence de ligatures et liaisons avec des rythmes comprenant des n-olets

La syntaxe de LilyPond demande parfois un positionnement inhabituel des parenthèses, crochets, etc. qui peuvent s'entrelacer.

Par exemple, le crochet ouvrant une ligature manuelle se place **à la suite** de la note de départ et sa durée, non pas avant. De même, le crochet fermant se place à la fin de la ligature, y compris lorsque la dernière note se trouve incluse dans un n-olet.

Cet extrait illustre la manière de combiner ligatures manuelles, liaisons d'articulation, de prolongation ou de phrasé, avec des n-olets bornées par des accolades.

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



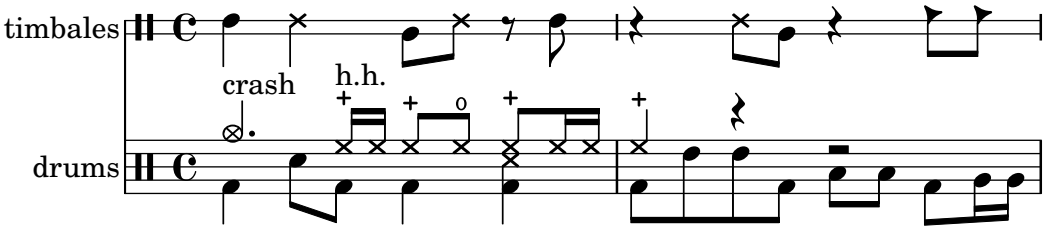
Ajout de parties de batterie

Grâce à la puissance des outils préconfigurés tels que la fonction `\drummode` et le contexte `DrumStaff`, la saisie de parties pour percussions est extrêmement simplifiée : chaque composant d'une batterie trouve sa place sur une portée dédiée (avec une clef particulière) et les têtes de note sont spécifiques à chaque élément. Il est également possible d'affecter un symbole particulier à chaque élément, tout comme de restreindre le nombre de lignes de la portée.

```
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 }
}
```
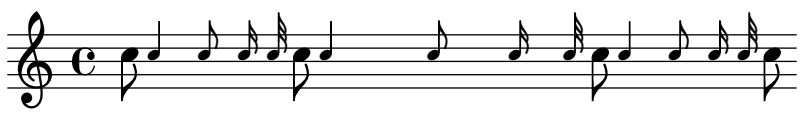


Ajustement de l'espacement des notes d'ornement

La propriété `spacing-increment` de `Score.GraceSpacing` permet d'ajuster l'espacement d'un groupe de notes d'ornement.

```
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
}
```



Alignement des numéros de mesure

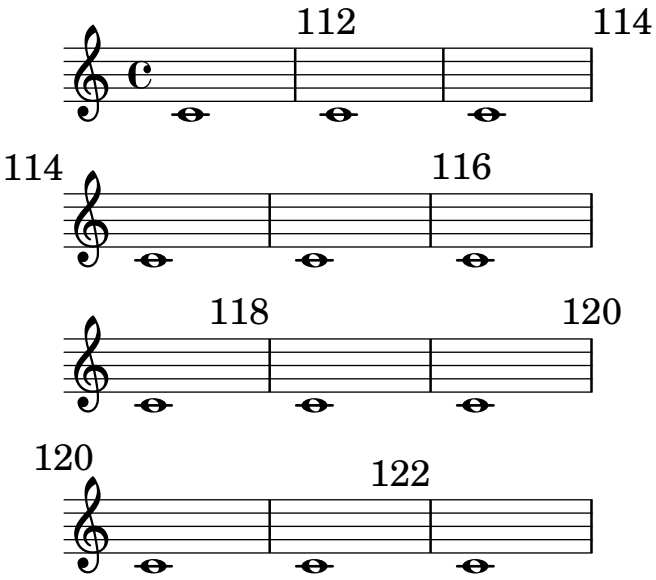
L'alignement des numéros de mesure dépend de l'endroit où ils se produisent : en début de ligne, ils seront alignés par la droite ; ils seront alignés sur la gauche de la barre de mesure dans les autres cas. Ceci peut se modifier à l'aide de la fonction Scheme `break-alignment-list`. Les trois arguments à cette fonction déterminent respectivement l'alignement en fin de ligne, en cours de ligne, et en début de ligne.

```
\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 |
}
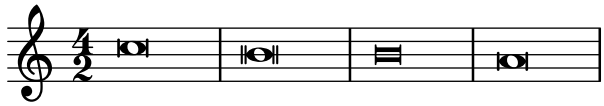
\paper {
  line-width = 70\mm
}
```



Brève alternative, avec deux barres verticales

Voici comment obtenir une brève – aussi appelée note carée – flanquée de deux barres verticales, au lieu d'une comme habituellement.

```
\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
}
```



Appoggiature avant une barre de mesure
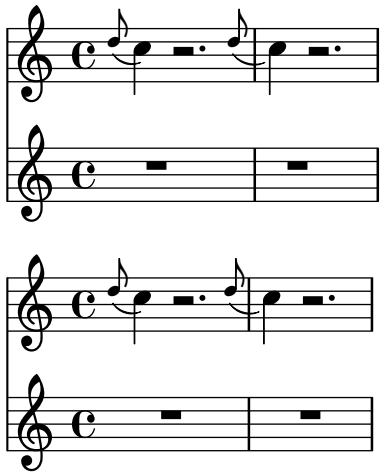
Par défaut, appogiatures et autres notes d'ornement sur le premier temps d'une mesure s'impriment après la barre. Dans le cas d'une ligne unique, elles peuvent toutefois précéder la barre grâce à l'insertion d'une barre invisible puis d'une visible.

Néanmoins, dans le cas d'un système à plusieurs portées, l'ajout d'une barre invisble gène le bonn positionnement des silences de mesure ; ils ne sont plus vraiment centrés, mais légèrement décalés sur la gauche. Une meilleure solution en pareil cas est d'utiliser la commande `\afterGrace` tout en jouant sur le réglage de sa propriété `afterGraceFraction`.

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

afterGraceFraction = 15/16

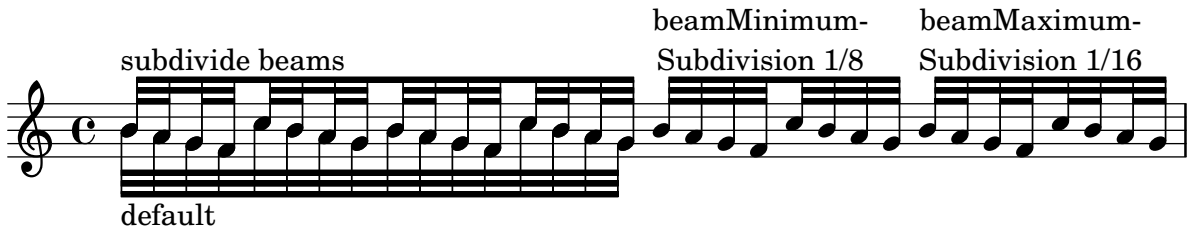
<<
  {
    \appoggiatura d''8 c''4 \afterGrace r2. d''8( |
    c''4) r2.
  }
  { R1 | R1 }
>>
```



Subdivision de ligatures automatiques

Dès lors que la propriété `subdivideBeams` aura été activée, une ligature sera subdivisée autant que possible. Les limites dans la profondeur sont gérées par les propriétés `beamMinimumSubdivision` et `beamMaximumSubdivision`.

```
\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]
  }
}
```



Changement automatique des durées
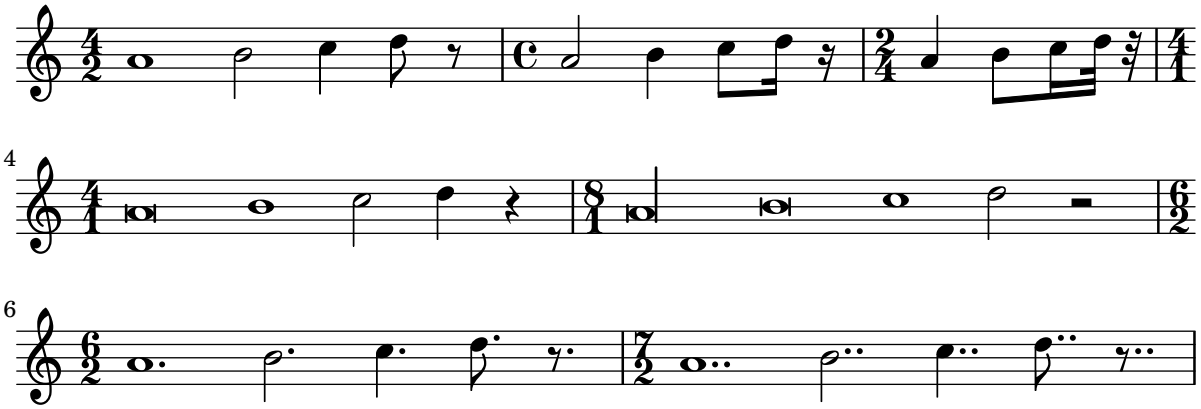
`shiftDurations` permet de modifier la longueur des notes d'un morceau.

Cette instruction prend deux arguments : un coefficient d'échelonnement (une puissance de deux) et un nombre de points d'augmentation (entier positif).

```
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
```
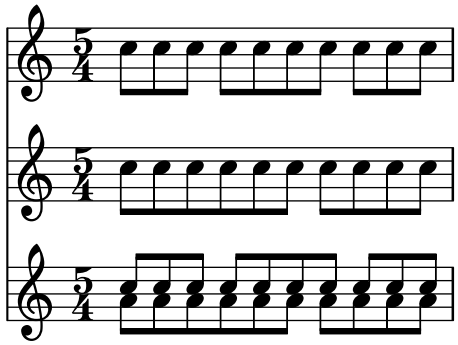
```
  \time 7/2
  \shiftDurations 0 2 \music
}
```



Définition de règles de ligature pour la partition

Les règles de ligature définies au niveau du contexte `Score` s'appliqueront à toutes les portées. Il est toutefois possible de moduler au niveau `Staff` ou `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 a
        }
      >>
    }
  >>
}
```



Débordement de ligature

En combinant `stemLeftBeamCount`, `stemRightBeamCount` et des paires de `[]` attachées à des notes isolées, vous pourrez obtenir des crochets rectilignes aux allures de ligatures.

Pour des crochets rectilignes à droite sur des notes isolées, il suffit d'ajouter une paire d'indicateurs de ligature `[]` et de déterminer `stemLeftBeamCount` à zéro, comme dans l'exemple 1.

Pour des crochets rectiligne à gauche, c'est `stemRightBeamCount` qu'il faudra déterminer (exemple 2).

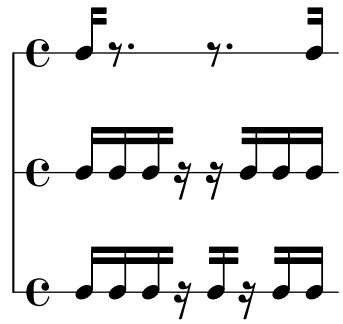
Pour que les barres de ligature débordent sur la droite, `stemRightBeamCount` doit avoir une valeur positive ; pour un débordement à gauche, c'est sur `stemLeftBeamCount` qu'il faut jouer. Tout ceci est illustré par l'exemple 3.

Il est parfois judicieux, lorsqu'une note est encadrée de silences, de l'affubler de crochets rectilignes de part et d'autre. L'exemple 4 montre qu'il suffit d'adjoindre à cette note un `[]`.

Notez bien que `\set stemLeftBeamCount` sera toujours synonyme de `\once \set`. Autrement dit, la détermination des ligatures n'est pas « permanente » ; c'est la raison pour laquelle les crochets du `c'16[]` isolé du dernier exemple n'ont rien à voir avec le `\set` indiqué deux notes auparavant.

```
\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
    }
  >>
}
```



Ligature au moment d'un saut de ligne

Il est normalement impensable qu'un saut de ligne tombe au milieu d'une ligature. LilyPond le permet néanmoins en activant la propriété `Beam.breakable`.

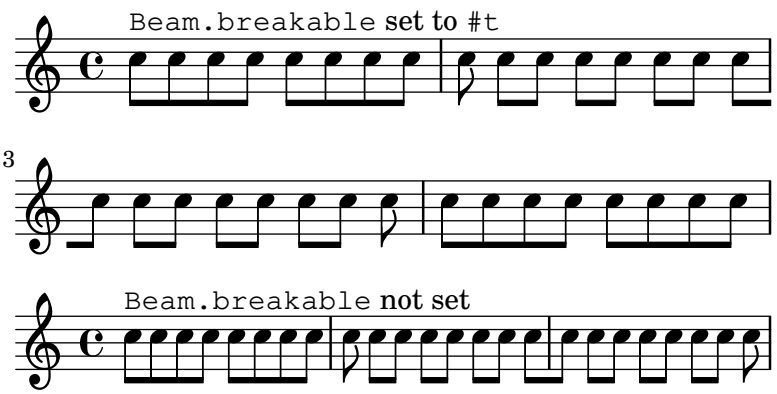
Cette propriété n'affecte pas les sauts manuels insérés par les commandes comme `\break`.
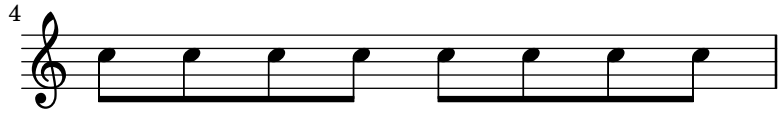
```
music = {
  \*8 c8
  c8 \*7 { c[ c] }  c
  \*8 c8
}

\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
}
```
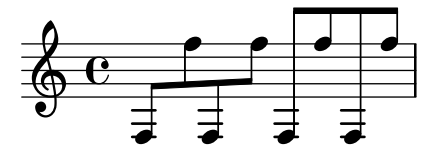




Ligature et directions de hampe inversées

LilyPond insère automatiquement des ligatures coudées – certaines hampes vers le haut, d'autres vers le bas – lorsqu'il détecte un intervalle important entre des têtes de notes. Ce comportement peut être changé par l'intermédiaire de l'objet `auto-knee-gap` – défini par défaut à 5,5 espaces, plus la largeur et la pente de la ligature en question.

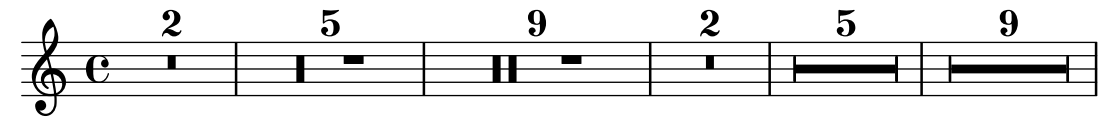
```
{
  f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



Modifier l'apparence d'un silence multimesure

Dans le cas où ce silence dure moins de dix mesures, LilyPond imprime sur la portée des « silences d'église » – *Kirchenpause* en allemand – et qui sont une simple suite de rectangles. Par défaut fixée à 10, la propriété `expand-limit` peut se régler afin d'obtenir un silence unique.
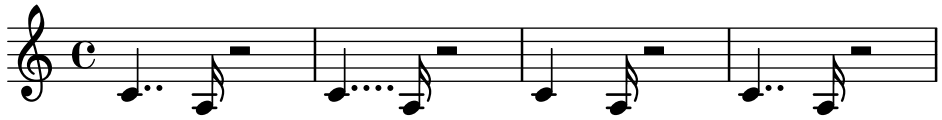
```
\relative c'' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = 3
    R1*2 | R1*5 | R1*9
  }
}
```



Spécification du nombre de points d'augmentation d'une note

Le nombre de points d'augmentation affectés à une note en particulier peut se modifier indépendamment des points placés après la note, à l'aide de la propriété `dot-count` de l'objet `Dots`.
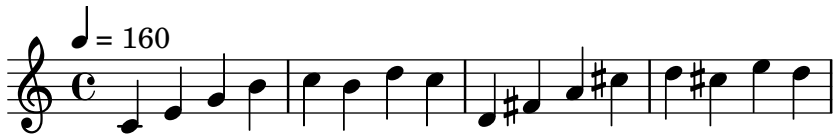
```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = 4
  c4.. a16 r2 |
  \override Dots.dot-count = 0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```



Changement de tempo sans indication sur la partition

Vous pouvez indiquer un changement de tempo pour le fichier MIDI sans pour autant l'imprimer. Il suffit alors de le rendre invisible aux musiciens.

```
\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 { }
}
```

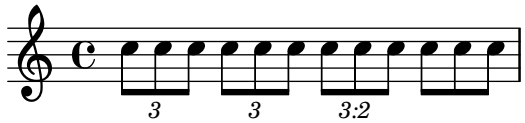


Modifier l'apparence du chiffre de n-olet

LilyPond imprime par défaut le numérateur de la fraction fournie en argument à la commande `\tuplet` du côté du crochet de n-olet.

Il est toutefois possible d'imprimer la fraction entière *num*:*den*, voire de ne rien imprimer du tout.

```
\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 }
}
```



Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations`

Polymétrie sans alignement des mesures

La prise en charge de contextes aux métriques indépendantes demande de supprimer le `Timing_translator` du contexte `Score`, conjointement à la création d'un contexte

TimingStaffGroup disposant du Timing_translator. Ceci fera de Timing un alias de TimingStaffGroup, permettant ainsi aux commandes \time de s'appliquer au sein du TimingStaffGroup où elles apparaissent.

Contrairement à la commande native de LilyPond \enablePerStaffTiming, cette approche requiert la création explicite de contextes TimingStaffGroup mais, par contre, autorise la création de multiples contextes Staff qui suivront de concert la mesure définie dans le TimingStaffGroup qui les englobe.

Échelonnement local des métriques

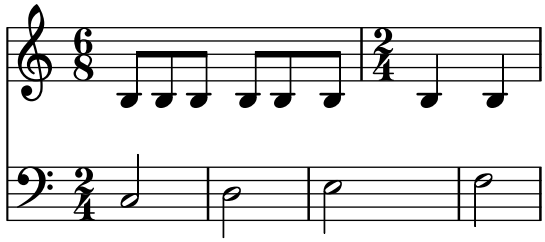
La commande \time, qui ne peut être échelonné, établit une mesure de la longueur désirée dans Timing, autrement dit dans TimingStaffGroup. Dans le code ci-dessous, toutes les portées incluses dans un TimingStaffGroup utilisent une métrique échelonnée, de sorte que, peu importe la métrique, elle tiendra dans la longueur de mesure désirée. Si l'un des contextes inclus n'avait pas de métrique échelonnée, le choix de la métrique devant être fournie à Timing aurait toute son importance.

L'utilisation de la commande \polymetric \time permet de définir les propriétés d'échelonnement de la métrique dans le contexte Timing, et l'utilisation de \scaleDurations échelonnera le mètre et les notes afin qu'elles tiennent dans la mesure.

```
\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
    }
```

```
  >>
>>
```

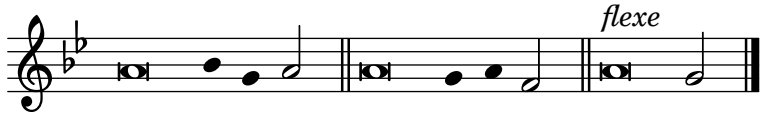


Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```
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
  }
}
```
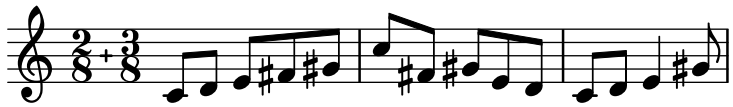


Métrique décomposée

Des métriques telles que « 5/8 » peuvent s'interpréter sous une forme décomposée – « 3/8 + 2/8 » par exemple – qui combine plusieurs métriques inégales.

LilyPond est capable de rendre ce type de notation, plus facile à lire et à interpréter, en imprimant cette métrique composite et en adaptant les règles de ligature automatique en conséquence.

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



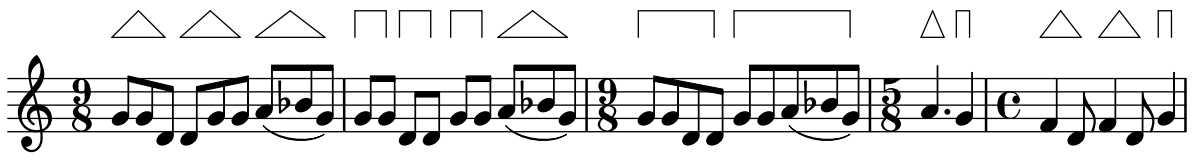
Signes de direction, signes de sous-groupe

Les regroupement par temps au sein d'une mesure sont contrôlés par des propriétés de contexte : `beatStructure` liste la longueur de chaque temps, en unités de `beatBase`. Les valeurs par défaut sont répertoriées dans le fichier `scm/time-signature-settings.scm`. Ces propriétés sont modifiables grâce à la commande `\set`.

Par ailleurs, l'instruction `\time` accepte des règles de pulsation différentes des valeurs par défaut. Dans la mesure où `\time` s'applique au contexte `Timing`, elle ne redéfinira pas les valeurs de `beatStructure` ou `beatBase` lorsqu'elles sont modifiées dans un contexte de niveau inférieur comme `Voice` par exemple.

Si l'on fait appel au `Measure_grouping_engraver`, la fonction `set-time-signature` créera aussi des symboles `MeasureGrouping`. Ces symboles aident à la lecture des œuvres modernes à la rythmique complexe. Dans l'exemple qui suit, la mesure à 9/8 est divisée en 2, 2, 2 et 3, alors que la mesure à 5/8 répond aux règles par défaut contenues dans le fichier `scm/time-signature-settings.scm`. Pour une mesure à 4/4, il faudra explicitement définir `beatBase` en croches afin que le motif irrégulier de la mesure soit correctement rendu.

```
\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"
    }
  }
}
```



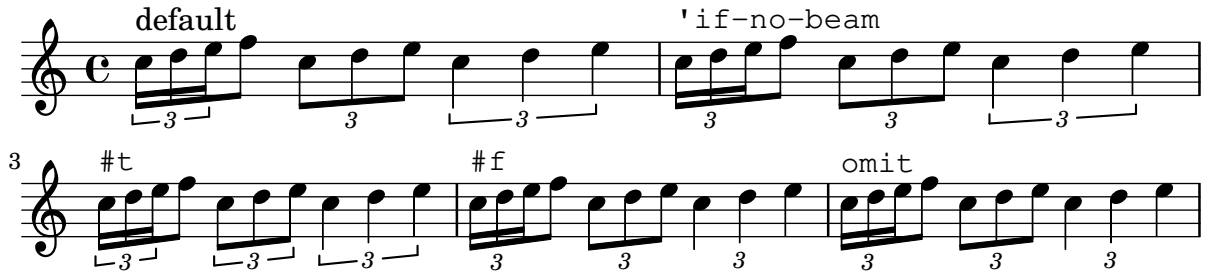
Contrôle de l'impression des crochets de n-olet

Selon la tradition, les crochets indicateurs de n-olet sont toujours imprimés, sauf dans le cas où ils seraient de la même longueur qu'une ligature.

LilyPond permet, au travers de la propriété `bracket-visibility`, de contôler précisément leur affichage : déterminée à `#t`, ils seront toujours imprimés ; `#f` permet de ne jamais les imprimer – donc omettre l'objet `TupletBracket` –, et `if-no-beam` les imprimera en l'absence de ligature (comportement par défaut).

```
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
  }
}
```



Exemple de sonnaille et clochette

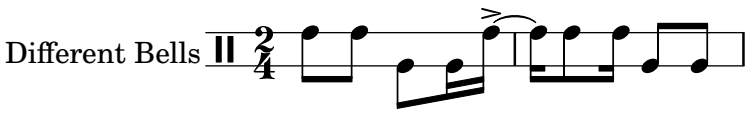
Voici deux différentes cloches, saisies « cb » (*cow bell* pour sonnaille) et « rb » (*ride bell* pour clochette).

```
#(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)
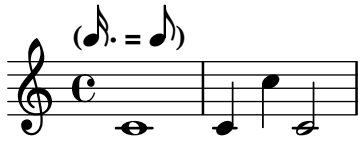
  \time 2/4
  rb8 8 cb8 16 rb16-> ~ |
  16 8 16 cb8 8 |
}
```



Création d'une indication métronomique sous forme d'étiquette

Vous pouvez créer des indications de tempo sous la forme d'étiquettes textuelles – des objets *markup* –, notamment des équivalences. Cependant, elles n'apparaîtront pas dans le fichier 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
}
```



Dessin à main levée de liaisons de tenue

Il est possible de graver manuellement les liaisons de tenue, en modifiant la propriété `staff-position` (un décalage) de l'objet `Tie`. Lorsqu'il y en a plusieurs, elles peuvent être ajustées manuellement à l'aide de la propriété `tie-configuration` de l'objet `TieColumn`, constituée d'une liste de paires. Pour chaque paire, le premier nombre indique la distance à la portée, en espaces de portée, et le second la direction (1 pour haut, −1 pour bas).

Notez bien que LilyPond fait la distinction, au niveau du décalage, entre valeur exacte et valeur inexacte. Dans le cas d'une valeur exacte – autrement dit un entier ou une fraction comme `(/ 4 5)` – celle-ci servira de position verticale brute, ensuite afinée par LilyPond de sorte à éviter les lignes de la portée. Dans le cas d'une valeur inexacte, tel un nombre à virgule flottante, c'est elle qui servira à positionner verticalement, sans ajustement.

```
\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' {
```

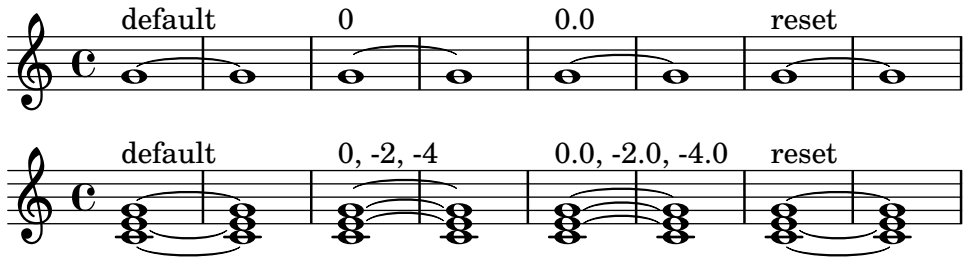
```
  \override TextScript.outside-staff-priority = ##f
  \override TextScript.padding = 0

  <>^"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>
}
```
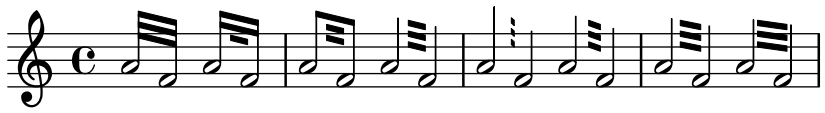


Impression de trémolo avec ligature flottante

Lorsque la durée totale d'un trémolo est inférieure à la noire, égale une blanche, ou bien entre une blanche et une ronde, il est d'usage que toutes les ligatures soient en contact avec les hampes. Certains styles de gravure font cependant apparaître des ligatures détachées, centrées entre les hampes. Pour ce type de trémolo, le nombre de hampes flottantes se gère au travers de la propriété gap-count de l'objet Beam, et l'écart entre ligature et hampe se définit par la propriété 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 }
}
```

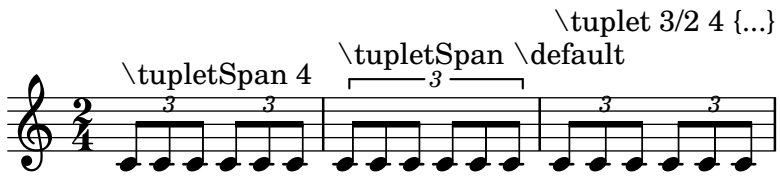


Plusieurs triolets avec une seule commande \tuplet

La propriété tupletSpannerDuration spécifie la longueur voulue de chaque crochet. Avec elle, vous pouvez faire plusieurs n-olets en ne tapant \tuplet qu'une fois, ce qui évite une longue saisie.

Il existe différents moyens de définir tupletSpannerDuration. La commande \tupletSpan lui affecte une durée arbitraire qui sera réinitialisée dès l'intervention d'une durée à \default. Vous pouvez aussi opter pour fournir un argument supplémentaire à la commande \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 }
}
```

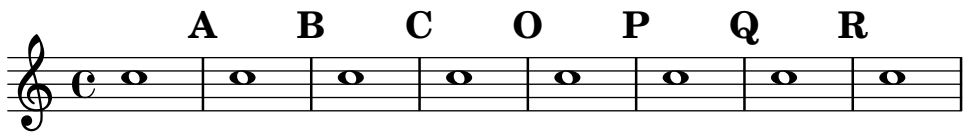


Fixation arbitraire du numéro de repère de départ

Voici comment initialiser arbitrairement une indication automatique de repère, qu'elle soit alphabétique ou numérique :

```
\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
}
```



Génération de crochets personnalisés

Une fonction Scheme personnalisée permet de redéfinir la propriété `stencil` de l'objet `Flag`, de sorte à modifier le glyphe utilisé pour les crochets de croche.

```
#(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
}
```

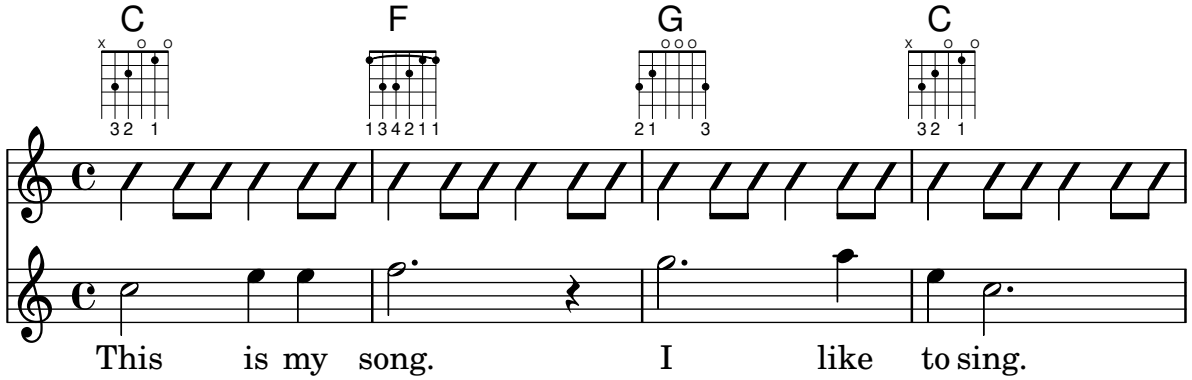


Rythmique et guitare

En matière de notation pour guitare, il arrive que soient indiqués les « coups de gratte » en plus de la mélodie, grilles d'accords et diagrammes de tablature.

```
\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.
  }
>>
```



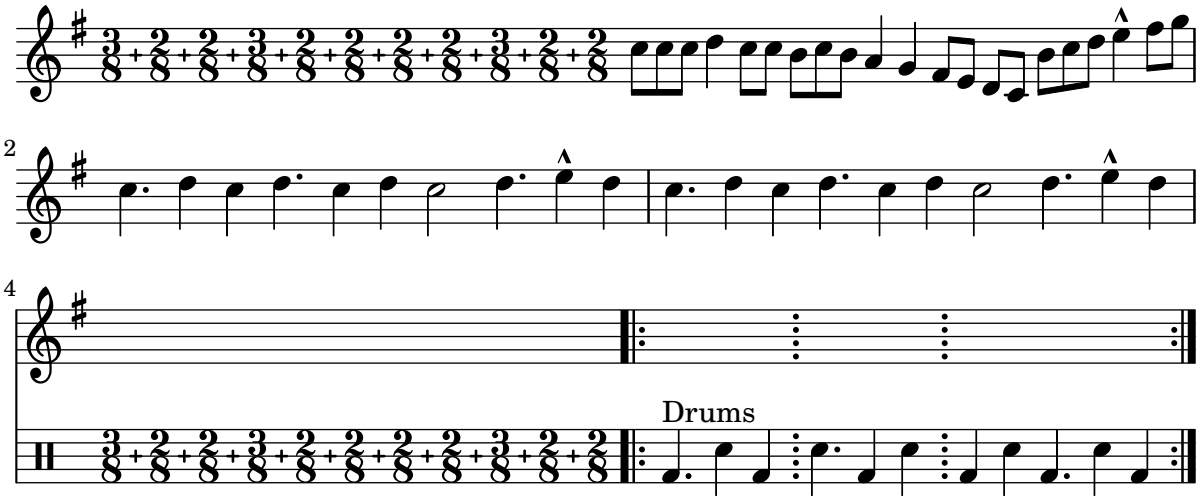
Indication personnalisée d'une polymétrie complexe

Bien que la métrique complexe ne soit pas l'élément primordial de cet exemple, elle permet d'indiquer la pulsation de cette pièce qui, par ailleurs, constitue le canevas d'une chanson traditionnelle des Balkans.

```
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
}
```



Exemple de woodblocks haut et bas

Deux woodblocks se saisissent « wbh » (woodblock haut) et « wbl » (woodblock bas). La hauteur des barres de mesure a été adaptée par une commande `\override` afin qu'elles ne soient pas trop courtes. La position des deux lignes de la portée doit aussi être explicitement définie.

```
% 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)))

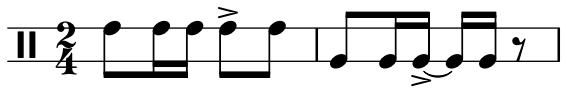
woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

  % 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 |
  }
}
```



Recours à la propriété `transparent` pour rendre des objets invisibles
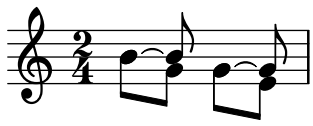
Affecter un objet graphique d'un `\hide` permet d'imprimer un objet « à l'encre sympathique » : l'objet n'est pas affiché bien que tous ses attributs soient préservés :

- l'objet en question occupe donc sa place,
- il est pris en compte lors de collisions, et
- il peut se voir attaché liaisons ou ligatures comme les autres.

Cet exemple illustre la manière de connecter deux voix par une liaison de prolongation. Les liaisons de prolongation ne peuvent normalement intervenir que dans la même voix. Dès lors que la liaison est entamée dans une autre voix et que la première hampe ascendante est masquée transparente dans cette même voix, la liaison semble passer d'une voix à l'autre.

```
\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
}
```
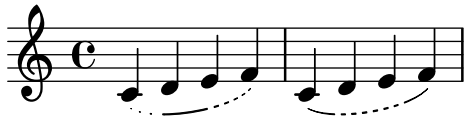


Dessin d'une liaison d'articulation au trait discontinu

Une liaison d'articulation peut être formée de traits discontinus variables. La propriété `dash-definition` se compose d'une liste de segments de liaison. Chaque segment contient une liste de paramètres qui déterminent le comportement du trait pour une section de la liaison.

Les segments de cette liaison se définissent selon le paramètre de Bézier `t` qui est compris entre 0 (l'extrémité gauche de la liaison) et 1 (l'extrémité droite de la liaison). Chaque segment se composera selon la liste (*t-début t-fin segment-style segment-taille*). La portion de liaison allant de *t-début* à *t-fin* aura un trait *segment-style* de longueur *segment-taille*. *segment-taille* est exprimé en espace de portée ; un *segment-style* à 1 donnera un trait plein. *segment-taille* s'exprime en espace de portée.
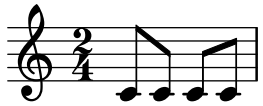
```
\relative c' {
  \once \override
    Slur.dash-definition = #'((  0  0.3  0.1 0.75)
                              (0.3  0.6  1   1   )
                              (0.65 1.0  0.4 0.75))
  c4( d e f)
  \once \override
    Slur.dash-definition = #'((0    0.25  1   1   )
                              (0.3  0.7   0.4 0.75)
                              (0.75 1.0   1   1   ))
  c4( d e f)
}
```



Contrôle manuel du positionnement des ligatures

Le positionnement et la pente des ligatures peuvent être contrôlés manuellement à l'aide d'une adaptation de la propriété `positions` de l'objet `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
}
```
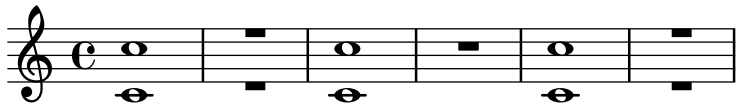


Fusion de silences multimesures dans une partie polyphonique

Dans le cadre d'un portée polyphonique, les silences multimesures sont positionnés différemment selon la voix à laquelle ils appartiennent. Le réglage suivant permet néanmoins de les imprimer sur une même ligne. En l'absence du `\once`, l'adaptation sera effective pour tous les autres silences à venir dans cette voix.

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

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

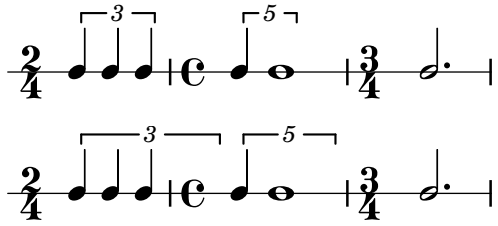


Modification de la longueur d'un crochet de n-olet

Les crochets indiquant un n-olet peuvent être prolongés jusqu'à un élément de rupture ou jusqu'à la note suivante. LilyPond termine un crochet de n-olet sur la droite de sa dernière tête de note par défaut ; un crochet de pleine longueur s'étendra plus avant, soit jusqu'à la note suivante et en traversant tous les éléments non rythmiques, soit sur tout l'espace précédant le prochain élément de notation, que ce soit une clef, une métrique, une armure ou une autre note. L'exemple suivant illustre la manière d'activer ces deux fonctionnalités.

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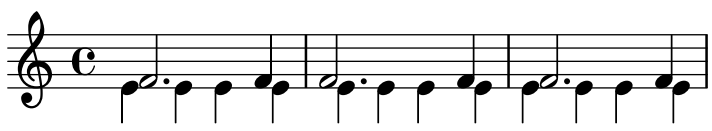
\new RhythmicStaff {
  % Set tuplets to be extendable...
  \set tupletFullLength = ##t
  % ...to cover all items up to the next note
  \set tupletFullLengthNote = ##t
  \time 2/4 \tuplet 3/2 { c4 4 4 }
  % ...or to cover just whitespace.
  \set tupletFullLengthNote = ##f
  \time 4/4 \tuplet 5/4 { 4 1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



Déplacement des notes pointées dans une polyphonie

Une note pointée appartenant à la voix supérieure d'une portée polyphonique sera par défaut décalée vers la droite afin d'éviter les collisions avec les autres voix. Ce comportement peut être modifié par un réglage de la propriété `prefer-dotted-right` de l'objet graphique `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 }
>>
```



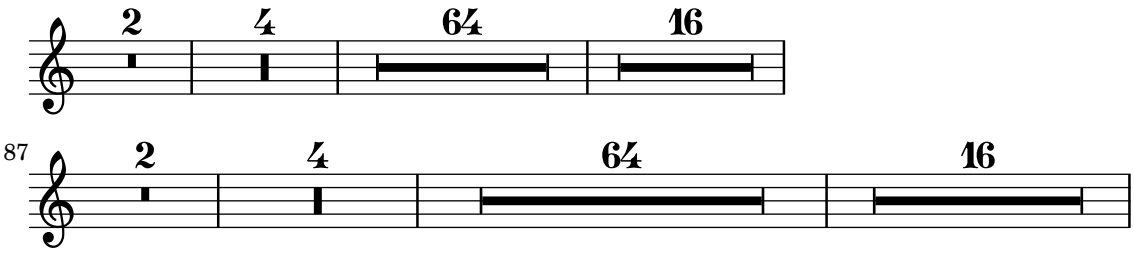
Contrôle de la taille d'un silence multimesure

Les silences multimesure ont une largeur relative à leur durée totale, contrôlée par la propriété `space-increment` de l'objet graphique `MultiMeasureRest`. Sa valeur par défaut est fixée à 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
}

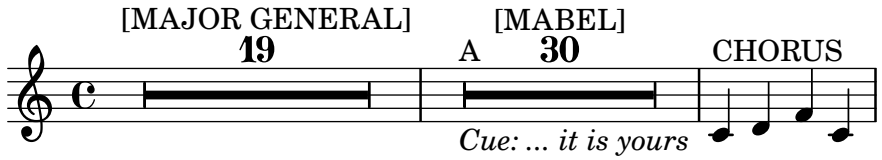
\layout {
  ragged-right = ##t
}
```



Ajout de texte à un silence multimesure

Lorsque du texte est attaché à un silence multimesure, il sera centré dans la mesure, au-dessus ou en dessous de la portée. Afin d'étirer la mesure dans le cas où ce texte est relativement long, il suffit d'insérer un accord vide auquel on attache le texte en question, avant le silence multimesure. Le texte attaché à un silence invisible sera aligné sur la gauche de là où serait positionnée la note dans la mesure. Cependant, si la taille de la mesure est déterminée par la longueur du texte, il apparaîtra comme centré.

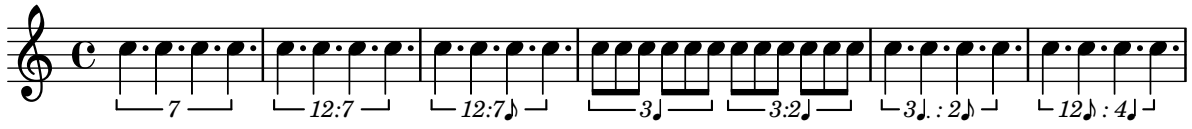
```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```



N-olets au chiffrage inhabituel

LilyPond sait aussi gérer des n-olets dont le chiffrage imprimé ne correspond pas exactement à la fraction de mesure à laquelle ils se réfèrent, tout comme ceux auxquels une valeur de note vient en complément du chiffre.

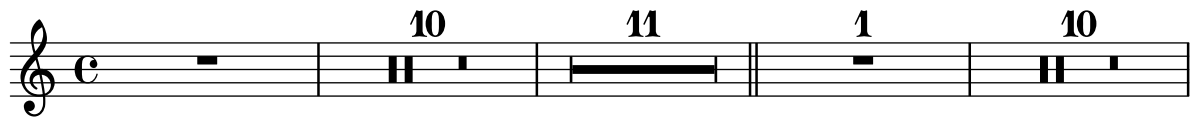
```
\relative c'' {
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override 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. }
}
```

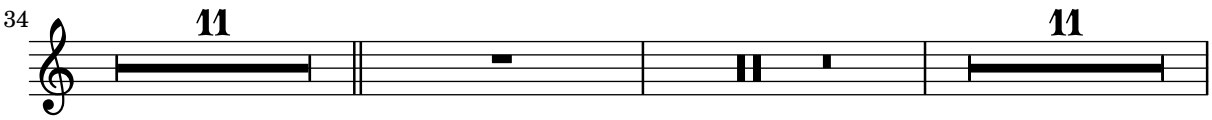


Dénombrer une unique mesure de silence

Les silences multimesures affichent leur longueur sauf s'il n'y a qu'une seule mesure. Ceci peut se modifier en réglant `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 et autoBeamOff

La fonction `\autoBeamOff` dans le cadre d'un `\partCombine` agit de façon bien particulière ; c'est pourquoi il vaut mieux tout d'abord recourir à

```
\set Staff.autobeaming = ##f
```

pour désactiver les ligatures automatiques pour l'ensemble de la portée concernée.
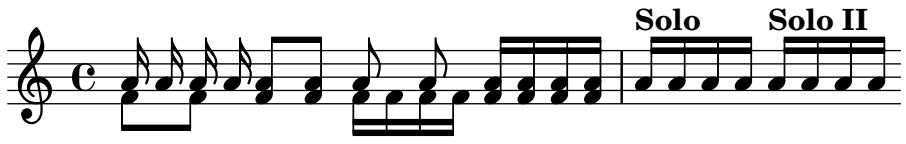
L'instruction `\partCombine` fonctionne de manière interne sur la base de quatre voix : solo hampes montantes, solo hampes descendantes, ensemble et solo.

Lorsque `\autoBeamOff` apparaît dans le premier argument de la combinaison, il s'applique à la voix active à ce moment précis, qu'il s'agisse du solo hampes montantes ou du combiné hampes montantes. Lorsqu'elle est introduite dans le second argument, la commande `\autoBeamOff` s'appliquera au solo hampes descendantes.

Vous devrez donc, afin que `\autoBeamOff` soit pleinement opérationnel dans le cadre d'un `\partCombine`, l'introduire aux **quatre** niveaux.

```
{
  % \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
  }
}
```



Exemple pour percussions

Ce court exemple est tiré de « L'histoire du Soldat » de Stravinsky.

```
#(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
}

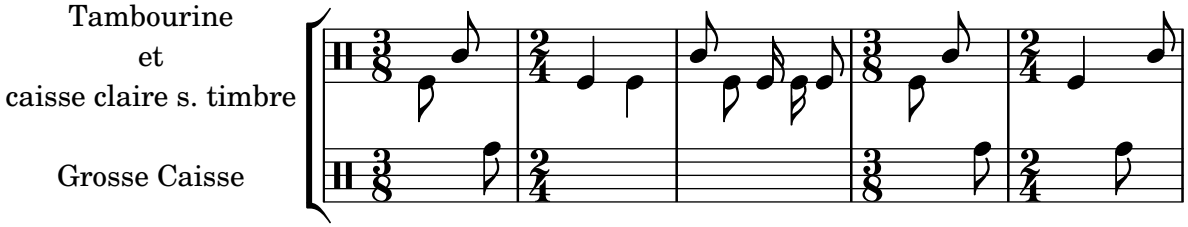
drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
      sn4 \D sn4 |
      \U tamb8 \D sn \U sn16 \D sn \U sn8 |
      \D sn8 \U tamb s |
      \U sn4 s8 \U tamb
    }
  >>
}

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
  >>
```

```
}
```



Saut de ligne au milieu d'un n-olet avec ligature

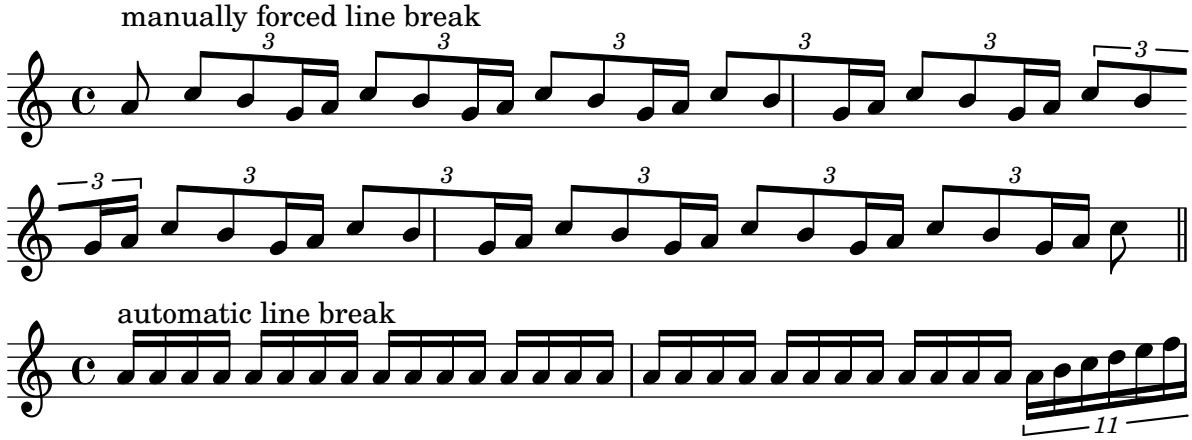
Ces exemples peu académiques démontrent comment il est possible d'insérer un saut de ligne dans un n-olet portant une ligature et qui ne sauraient se découper de manière exacte.
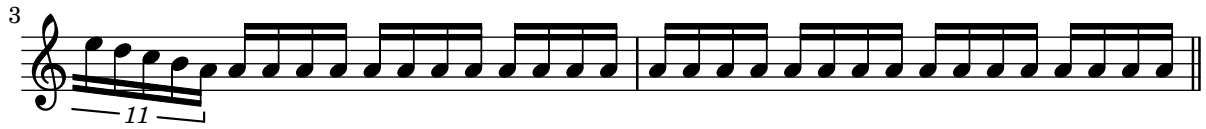
Cette fonctionnalité n'est opérationnelle qu'avec des ligatures manuelles.

```
\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 "||"
}
```
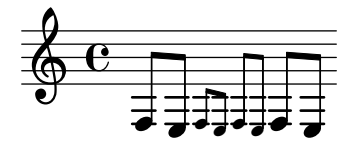




Positionnement des ligatures de notes d'ornement à la hauteur de celles des notes normales

La ligature de notes se trouvant sur des lignes supplémentaires est généralement centrée sur la portée. Les notes d'ornement ayant une hampe raccourcie, leur ligature peut se retrouver en dehors de la portée lorsqu'elles sont sur des lignes supplémentaires. LilyPond permet de rallonger les hampes en pareil cas.

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



Positionnement des notes d'ornement avec espace flottant

Lorsque la propriété `strict-grace-spacing` est activée, l'espacement des notes d'ornement se fera de manière « élastique ». Autrement dit, elles seront décollées de leur note de rattachement : LilyPond commence par espacer les notes normales, puis les ornements sont placés à la gauche de leur note de rattachement.

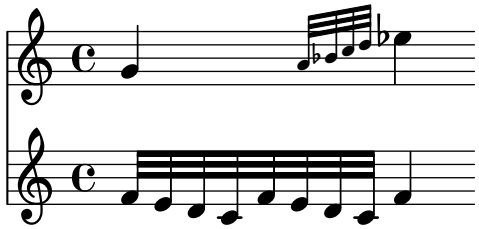
Cependant, en raison du ticket 6876 (`https://gitlab.com/lilypond/lilypond/-/issues/6876`), les altérations accidentelles sont ignorées lorsque cette propriété est activée. Le code ci-dessous propose une solution de contournement à ce problème.

Autre effet indésirable de cette propriété, LilyPond ne vérifie pas qu'il y a suffisamment d'espace pour les notes d'ornement (ceci fait l'objet du ticket 2630 (`https://gitlab.com/lilypond/lilypond/-/issues/2630`)). Il faut donc s'assurer d'avoir suffisamment d'espace disponible en recourant conjointement `\newSpacingSection` et une valeur appropriée pour le `base-shortest-duration` de l'objet graphique `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
  }
}
```



Positionnement des silences multimesures

Si l'on peut positionner verticalement un silence simple en le rattachant à une note, il n'en va pas de même pour un silence multimesure. Néanmoins, et uniquement dans le cadre de musique polyphonique, les silences multimesures sont positionnés différemment selon qu'ils appartiennent à une voix au numéro pair ou impair.

Le positionnement des silences multimesures peut se contrôler comme ici indiqué.

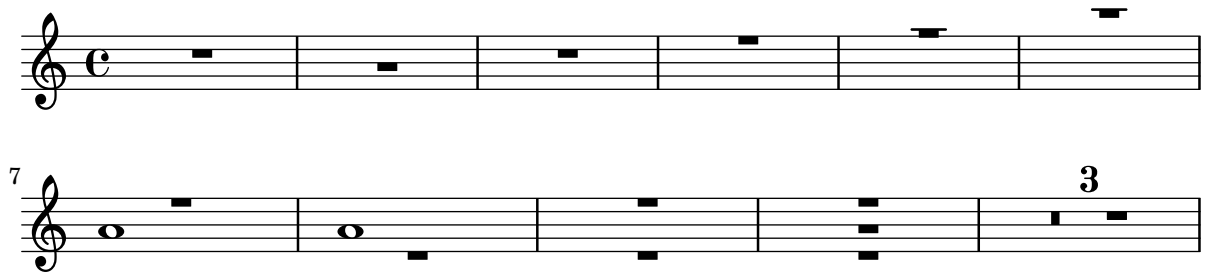
```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line.
  R1
  % They can be moved using an override or tweak.
  \tweak staff-position -2 R1
  \tweak staff-position 0 R1
  \tweak staff-position 2 R1
  \override MultiMeasureRest.staff-position = 3 R1
  \override MultiMeasureRest.staff-position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



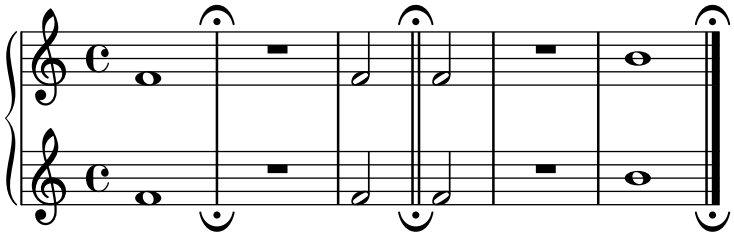
Positionnement de points d'orgue en opposition sur une barre de mesure

Dans l'extrait suivant est construite une commande qui permet d'imprimer un point d'orgue à la fois en surplomb et en dessous d'une barre de mesure. Dans le cas où il n'y aurait normalement pas de barre à cet endroit, sera ajoutée une double barre. Sémantiquement, cette commande constuit une césure plus longue que la normale, ce qui pourrait être considéré comme un usage abusif selon le cas.

```
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
>>
```



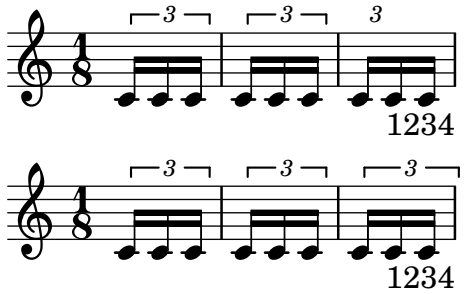
Préservation de l'indication de n-olet lors d'un repère final

En raison de ce qui est rapporté dans le ticket 2362 (`https://gitlab.com/lilypond/lilypond/-/issues/2362`), l'ajout d'une marque-repère en fin de pièce peut entraîner la perte de la dernière indication de n-olet. La désactivation de `TupletBracket.full-length-to-extent` pallie ce problème.

```
\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

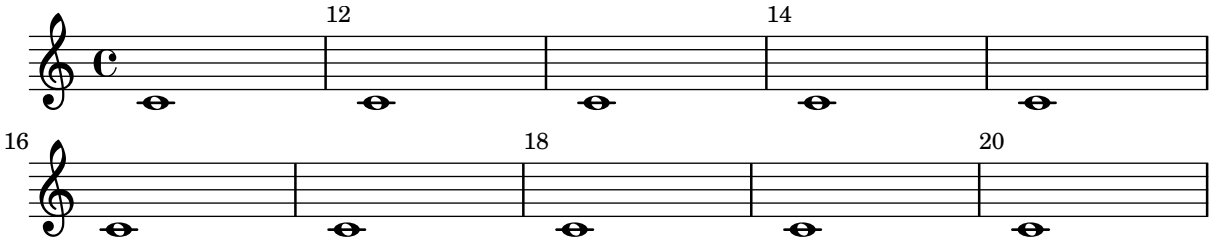
  \time 1/8
  \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' c' }
  \tweak direction #DOWN \textEndMark "1234"
}
```



Imprimer les numéros de mesure à intervalle régulier

Vous pouvez imprimer un numéro de mesure à intervalle régulier plutôt qu'en tête de chaque ligne seulement, en recourant à la propriété `barNumberVisibility`. Voici comment afficher le numéro toutes les deux mesures sauf en fin de ligne.

```
\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
}
```

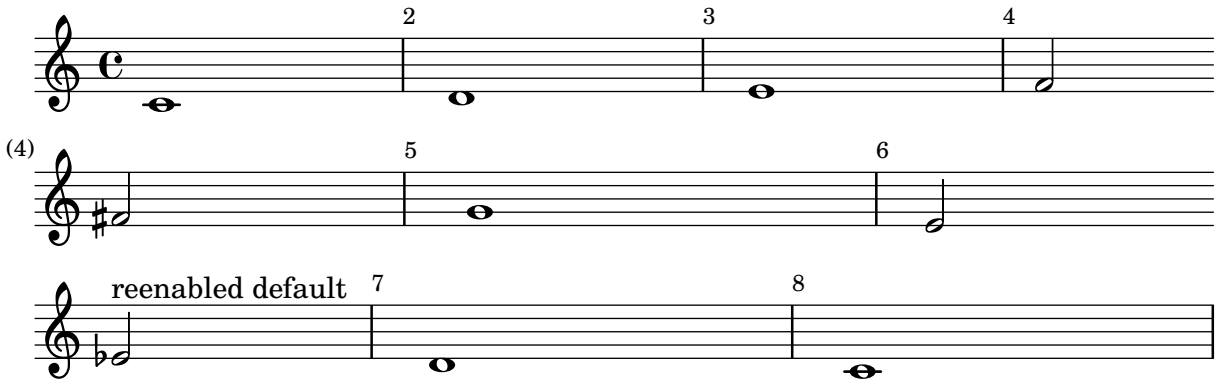


Impression du numéro des mesures tronquées

Le numéro de mesure n'est pas répété en début de ligne pour une mesure tronquée. L'objet BarNumber apparaîtra, entre parenthèses, dès lors que la propriété barNumberVisibility sera affublée de first-bar-number-invisible-save-broken-bars.

```
\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
}
```



Inscrire le numéro de mesure dans un cadre ou un cercle

Les numéros de mesure peuvent être encadrés ou entourés d'un cercle.

```
\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

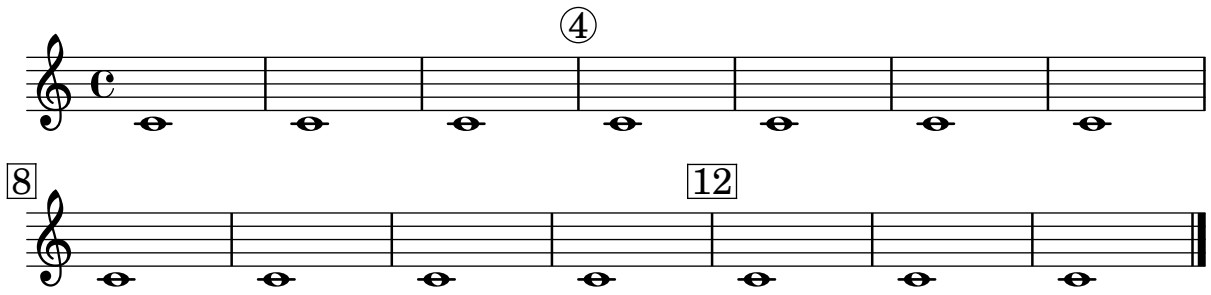
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % 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 "|."
}
```



Impression du numéro de mesure selon `modulo-bar-number-visible`

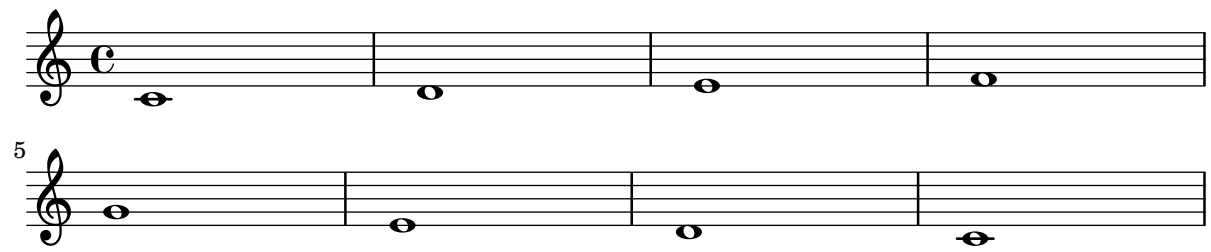
Lorsque le reste de la division du numéro de la mesure courante par le premier argument de `modulo-bar-number-visible` égale le deuxième argument, LilyPond imprime un objet `BarNumber`.

Ceci permet d'imprimer le numéro de mesure à un intervalle donné, par exemple :

- `(modulo-bar-number-visible 3 2)` → affichage à 2, 5, 8...
- `(modulo-bar-number-visible 4 2)` → affichage à 2, 6, 10...
- `(modulo-bar-number-visible 3 1)` → affichage à 3, 5, 7...
- `(modulo-bar-number-visible 5 2)` → affichage à 2, 7, 12...

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

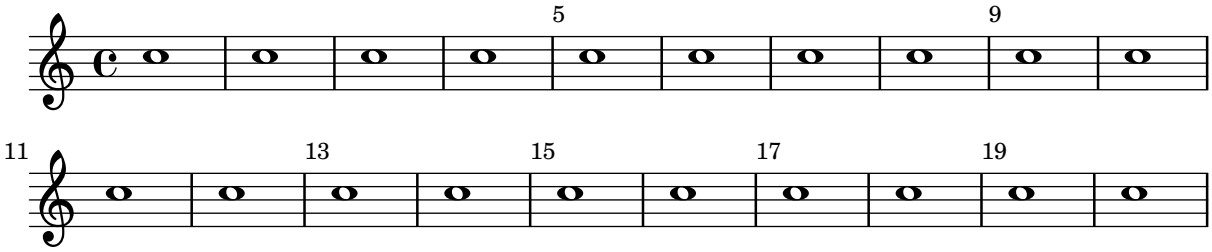
\relative c' {
  c1 | d | e | f \break
  g1 | e | d | c
}
```



Changement de la fréquence d'impression du numéro de mesure

La fonction de contexte `set-bar-number-visibility` permet de modifier la fréquence à laquelle les numéros de mesures s'impriment.

```
\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
}
```



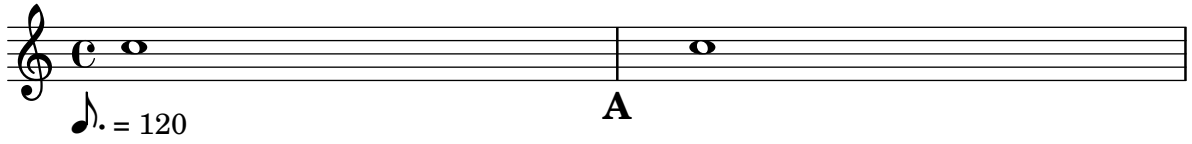
Impression du métronome et des repères sous la portée

Les indications de tempo et les marques de repère s'impriment par défaut au-dessus de la portée. Le fait de régler en conséquence la propriété `direction` des objets `MetronomeMark` et `RehearsalMark` les placera au-dessous de la portée.

```
\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
}
```



Impression de musique aux métriques différentes

Bien qu'ayant des métriques bien différentes, les deux parties ci-dessous présentées restent synchrones.

Les barres de mesure ne peuvent plus être gérées au niveau du contexte `Score` ; le `Timing_translator` doit être déplacé du contexte `Score` au contexte `Staff` afin de permettre des barres de mesure individualisées.

Le Bar_number_engraver devra lui aussi être déplacé, puisqu'il dépend de propriétés attachées au Timing_translator, afin de numéroter les mesures. L'utilisation d'un bloc \with dans la portée concernée permettra un affichage des numéros de mesure.

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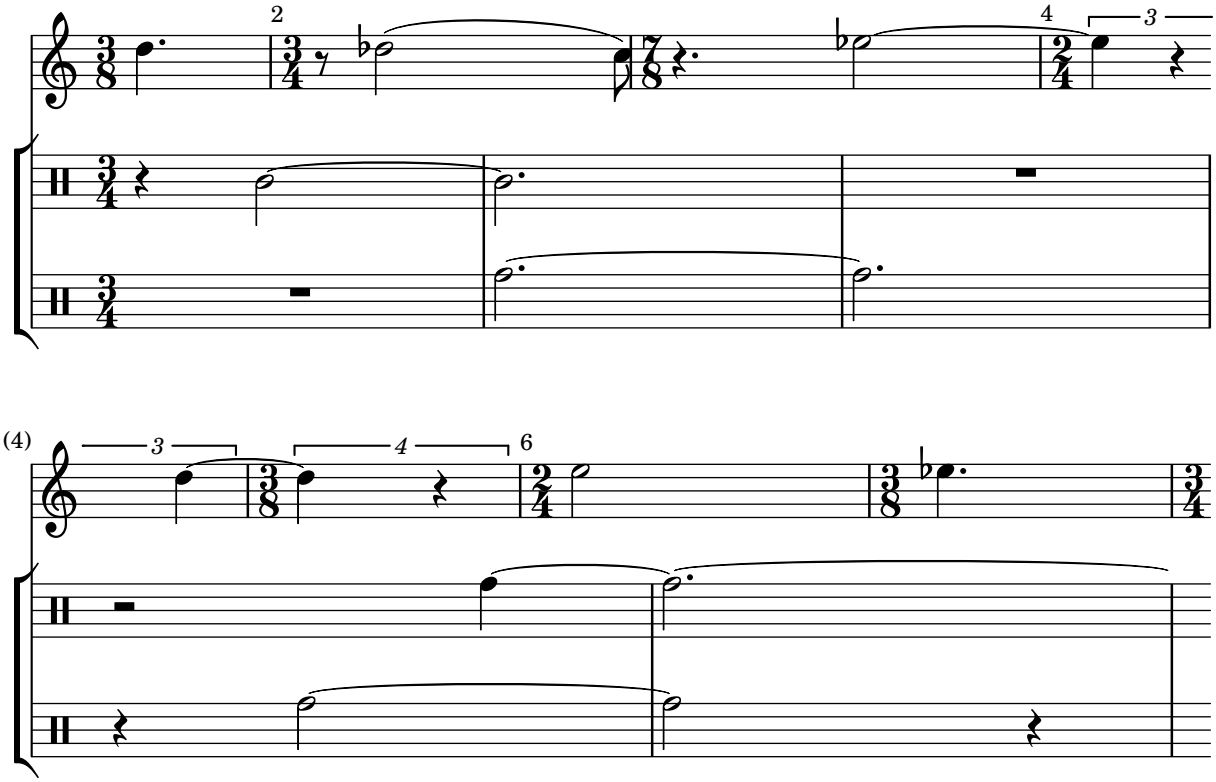
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"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

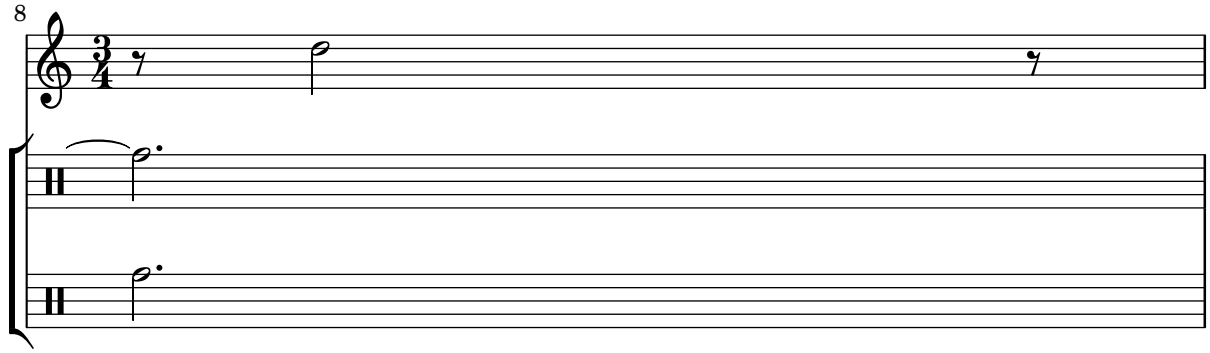
Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. 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
  >>
}
```



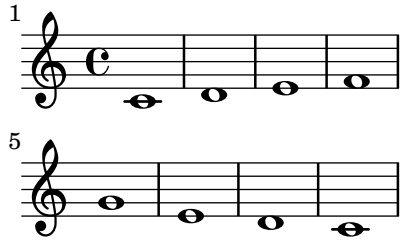


Afficher le numéro de la première mesure

Par défaut, LilyPond n'affiche pas le premier numéro de mesure s'il est inférieur à 2. Le fait de définir `barNumberVisibility` à `all-bar-numbers-visible` permettra d'imprimer n'importe quel numéro pour la première mesure.

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

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



Affichage du crochet de n-olet du côté des têtes de note
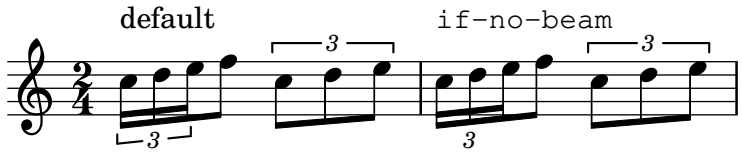
Quelle que soit l'option choisie pour contrôler la visibilité d'une indication de n-olet, elle s'affichera ou sera masquée indépendamment de son positionnement (côté hampe ou tête de note). Toutefois, lorsqu'elle se place du côté des têtes de note, certains auteurs recommandent de toujours afficher un crochet. L'option `visible-over-note-heads` permet d'y parvenir.

```
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
  }
}
```



Redéfinition des réglages de mise en forme par défaut des notes d'ornement

Les réglages par défaut des notes d'ornement sont stockés dans les variables suivantes :
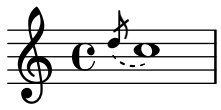
```
startGraceMusic
stopGraceMusic
startAcciaccaturaMusic
stopAcciaccaturaMusic
startAppoggiaturaMusic
stopAppoggiaturaMusic
```

Ces variables sont définies dans le fichier `ly/grace-init.ly`. Amender leur définition permet d'en varier les effets.

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

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

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



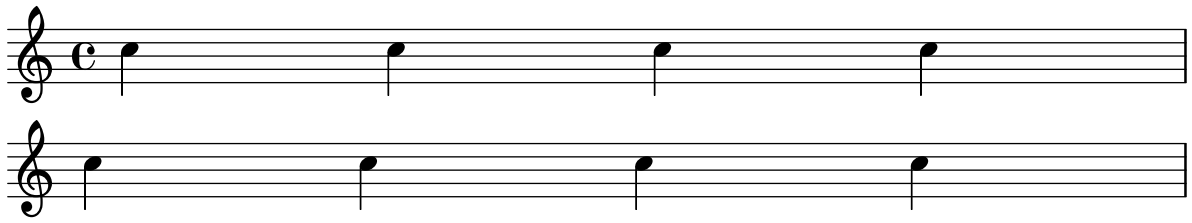
Suppression des numéros de mesure d'une partition

Désactiver le graveur concerné – `Bar_number_engraver` – donnera une partition – contexte `Score` – sans numéros de mesure.

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

```
  }
}

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



Styles de silences

Les silences peuvent être gravés selon différents styles.

```
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

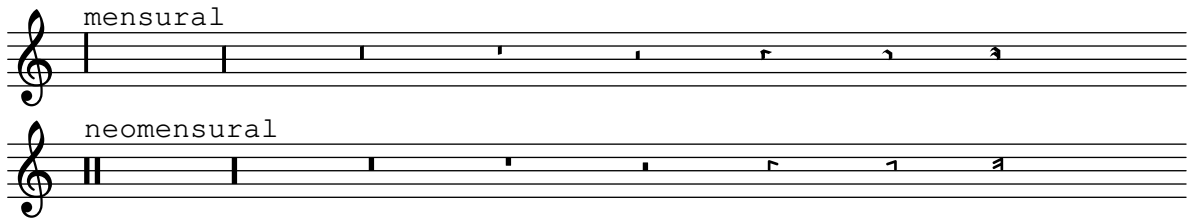
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

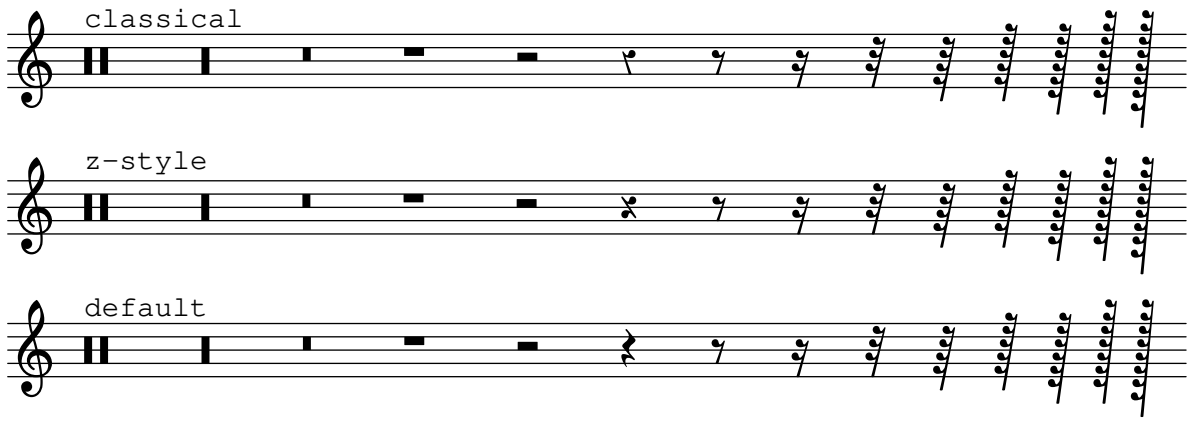
  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \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
}
```





Annulation des règles de ligature par défaut

Pour obtenir des ligatures en groupes de 3-4-3-2 croches, dans une mesure à 12/8, il faudra préalablement annuler les réglages par défaut relatifs à 12/8, puis ajouter les règles adaptées :

```
\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
}
```



Barres rythmiques

Il arrive, dans une feuille de chant « simple », que les notes soient remplacées par une « pulsation » et que la structure de la chanson soit indiquée par les accords au-dessus des mesures. Ceci peut être utile lorsque l'on crée ou retranscrit la structure d'une chanson, ainsi que pour donner au guitariste et musiciens de jazz une pseudo partition.

```
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 |
```

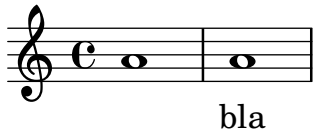
```
  \stopPat
  f'4 e' d' c'
}
```



Sauts de notes en mode paroles

La syntaxe s qui permet de « faire un saut dans le temps » n'est disponible qu'en mode notes et en mode accords. Dans les autres situations, comme en mode paroles par exemple, la commande \skip produit les mêmes effets.

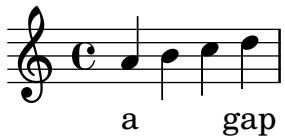
```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip1 bla1 }
>>
```



Sauts de notes en mode paroles (2)

Sachant qu'un s au sein d'un bloc \lyricmode est interprété non comme de l'espace mais comme une syllabe, l'utilisation de paires de guillemets informatiques ("") ou un caractère souligné simple (_) permet de « sauter » une note.

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



Moignons de hampe

Certaines conventions en matière de notation autorisent les ligatures à enjamber des silences. Dans certains cas, des moignons de hampe accrochés à la ligature offrent une meilleure visibilité du rythme ; certaines éditions modernes vont même alors jusqu'à omettre le silence.

Cet exemple illustre la progression : notation traditionnelle, ligature enjambant le silence, silence surplombé d'un moignon et enfin seule une hampe tronquée. Les moignons s'obtiennent par amendement de la propriété stemlet-length de l'objet Stem, alors que les silences sont masqués à l'aide d'un \hide.
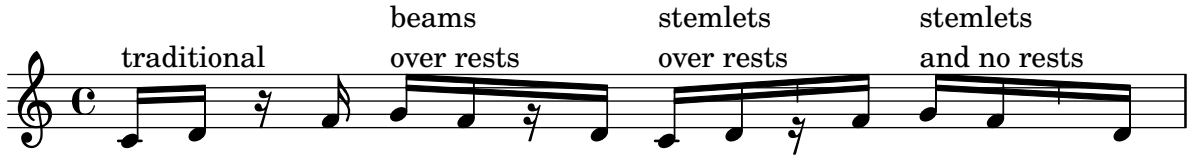
Les *markups* ajoutés au code ci-dessous mettent en exergue les différentes notations.

```
\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
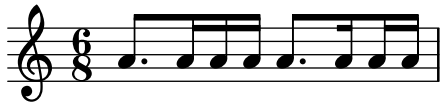
  c'16[^\markup \column { "stemlets" "over rests" } d' r f']
  g'16[^\markup \column { "stemlets" "and no rests" } f'
  \once \hide Rest
  r16 d']
}
```



Ligature à la pulsation

Une sous-ligature tronquée peut pointer en direction de la pulsation à laquelle elle se rattache. Dans l'exemple suivant, la première ligature évite toute troncature (comportement par défaut), alors que la deuxième respecte rigoureusement la pulsation.

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



Subdivision des ligatures

Les ligatures d'une succession de notes de durée inférieure ou égale à la double croche ne sont pas subdivisées par défaut. Autrement dit, tous les traits de ligature (deux ou plus) seront continus. Ce comportement peut être modifié afin de diviser la ligature en sous-groupes grâce à la propriété `subdivideBeams`. Lorsqu'elle est activée, un certain nombre de traits de ligature entre deux hampes est supprimé à des intervalles correspondant à la durée de sous-groupe souhaitée.

Les propriétés `beamMinimumSubdivision` et `beamMaximumSubdivision` permettent de configurer les limites de la subdivision automatique des ligatures : l'intervalle rythmique minimum auquel subdiviser et le nombre de tronçons supprimés selon cet intervalle. Leur valeur par défaut est respectivement de `0` et `+inf.0`, ce qui demandera à LilyPond de subdiviser le plus possible.

Certains cas particulier doivent être pris en considération.

- Lorsque le nombre fourni en numérateur à `beamMaximumSubdivision` n'est pas une puissance de 2, la plus petite durée pour subdiviser sera de `beamMaximumSubdivision` divisé par une puissance de 2 tout en restant supérieur ou égal à `beamMinimumSubdivision`.
- Dès lors que `beamMaximumSubdivision` est inférieur à `beamMinimumSubdivision`, la profondeur de subdivision se limite à `beamMaximumSubdivision` mais pas la fréquence ou l'intervalle, ce qui peut dévier de la valeur métrique à laquelle on est en droit de s'attendre.

Lorsque la propriété `respectIncompletBeams` est activée, les subdivisions incomplètes à plus de deux hampes sont considérées comme « extension » du sous-groupe précédent, autrement dit, le groupe de subdivision précédente sera étendu pour couvrir aussi la subdivision incomplète. Lorsqu'elle est désactivée, ce qui est le cas par défaut, une nouvelle subdivision est alors entamée.

```
\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

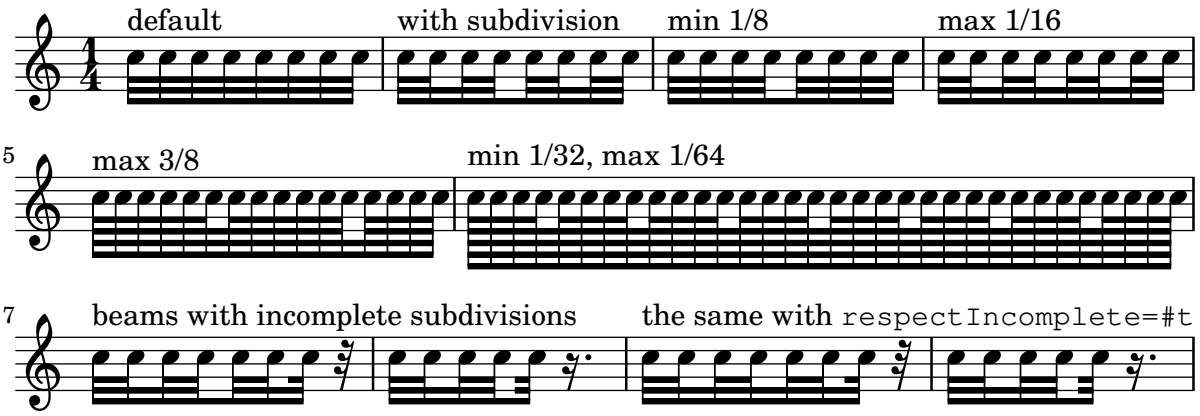
  <>^"max 3/8"
  \once \set beamMaximumSubdivision = #3/8
  \*16 c64

  <>^"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 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 r32
  % No visual change since we have only two stems in the
  % incomplete subgroup.
```

```
  c32 c c c c r16.
}
```



Exemple pour tam-tam

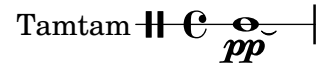
Un simple exemple pour tam-tam, saisi « tt ».

```
#(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
}
```



Exemple pour tambourin

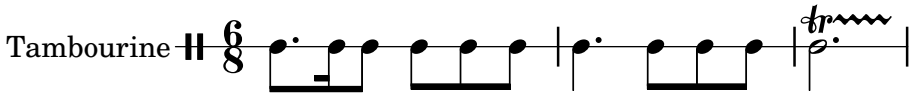
Voici une partie de tambourin, saisi par « tamb ».

```
#(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 |
}
```



Encadrement sur trois côtés

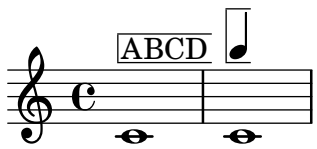
Voici comment construire une commande de *markup* chargée d'agrémenter du texte ou autre annotation, d'une bordure sur trois côtés.

```
% 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)))
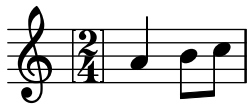

\relative c' {
  c1^\markup { \NWS-box ABCD }
  c1^\markup { \NWS-box \note {4} #1.0 }
}
```



Métrique entre crochets

La métrique peut être mise entre crochets.
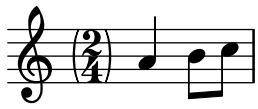
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Métrique entre parenthèses

Une métrique peut être mise entre parenthèses.
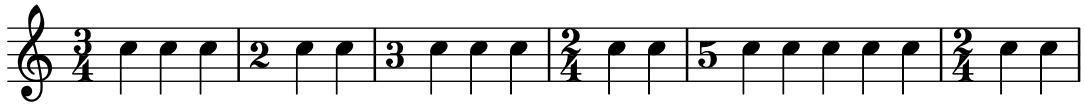
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Affichage seulement du numérateur d'une métrique (au lieu d'une fraction)

La métrique est parfois indiquée non pas par une fraction (par ex. 7/4) mais simplement par son numérateur (le chiffre 7 dans ce cas). L'instruction `\override Staff.TimeSignature.style = #'single-number` permet de déroger au style par défaut de manière permanente – un `\revert Staff.TimeSignature.style` annulera ces modifications. Lorsque cette métrique sous la forme d'un seul chiffre ne se présente qu'une seule fois, il suffit de recourir à un `\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
}
```
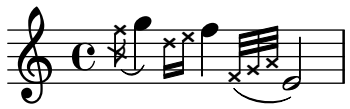


Mise en forme des notes d'ornement

Il est possible de changer globalement l'apparence des notes d'ornement dans un morceau, au moyen des fonctions `add-grace-property` et `remove-grace-property`.

Ici, par exemple, on ôte la définition de l'orientation (propriété `direction`) des objets `Stem` pour toutes les petites notes, afin que les hampes ne soient pas toujours orientées vers le haut, et on leur préfère des têtes en forme de croix.

```
\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
    }
  }
}
```
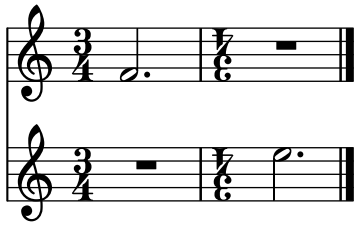


Personnalisation de la métrique

Il est possible de définir de nouveaux styles de métrique. La métrique de la deuxième mesure se présente sens dessus dessous sur les deux portées.

```
#(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''
  }
>>
```



Crochets de style alternatif

Une dérogation à la propriété `stencil` de l'objet `Flag` permet aux croches et notes de durée inférieure d'adopter une autre forme de crochet. Sont disponibles les variantes `modern-straight-flag`, `old-straight-flag` et `flat-flag`. Un `\revert` permet de retrouver l'allure par défaut.

Des crochets empilés, autrement dit à l'espacement resserré, s'obtiennent à l'aide de l'instruction `\flagStyleStacked`, qui s'annule par un `\flagStyleDefault`.

Une dérogation au stencil de `Flag` ne modifie en rien le positionnement vertical individuel des crochets. Ceci s'observe avec des crochets rectilignes : LilyPond n'ajuste pas dynamiquement l'écart entre les crochets les uns par rapport aux autres de la même manière que pour les ligatures. L'une des solutions pour harmoniser l'apparence consiste à remplacer les crochets par des demi-ligatures comme indiqué dans la deuxième portée . ceci ne peut toutefois se faire automatiquement. Dans le code de cet extrait, ces demi-ligatures se mentionnent à l'aide du préfixe `@`, comme par exemple `@c8`.

Il est important de noter que des demi-ligatures *ne sont pas* des objets `Flag`, ce qui signifie que des modifications apportées aux objets `Flag` n'auront aucun effet sur elles (il faut utiliser des propriétés de `Beam`), et les propriétés de leur objet `Stem` associé adoptera le même comportement qu'avec des ligatures.

```
"@" =
#(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

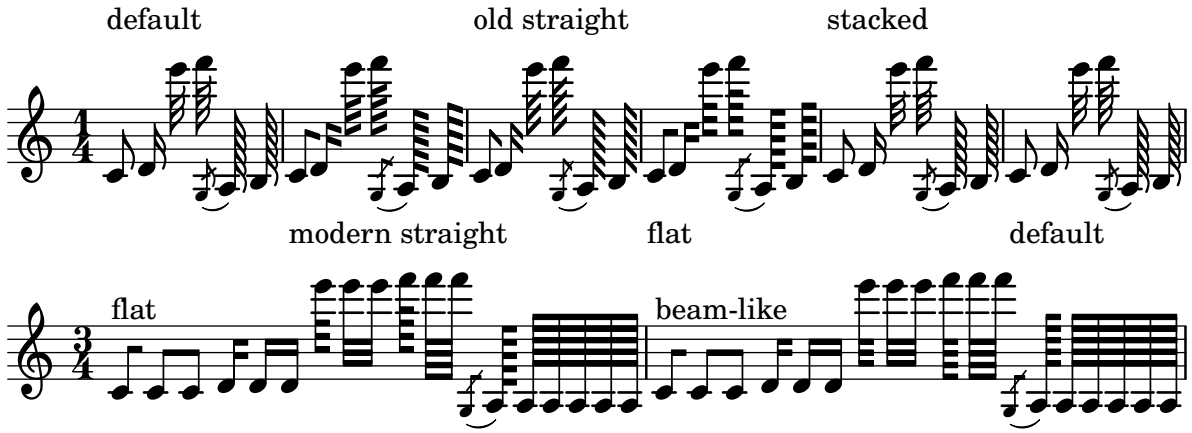
  \flagStyleStacked
    <>^"stacked" \testnotes
  \flagStyleDefault
    <>_"default" \testnotes
}

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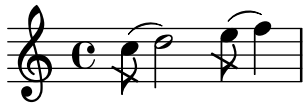
\layout {
  indent = 0
  \context {
    \Score
    \override NonMusicalPaperColumn.line-break-permission = ##f
  }
}
```



Utilisation de hampe barrée pour une note normale

Le trait que l'on trouve sur les hampes des acciaccatures peut être appliqué dans d'autres situations.

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

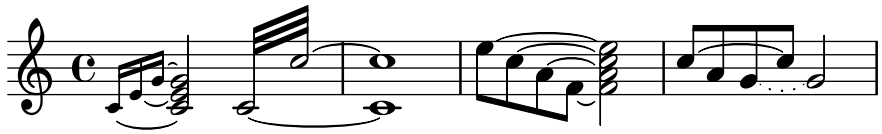


Liaison de tenue et arpège

Les liaisons de tenue servent parfois à rendre un accord arpégé. Dans ce cas, les notes liées ne sont pas toutes consécutives. Il faut alors assigner à la propriété `tieWaitForNote` la valeur `#t` (*true* pour « vrai »). Cette même méthode peut servir, par exemple, à lier un trémolo à un accord.

```
\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
}
```
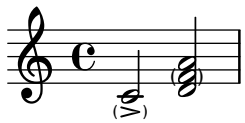


3 Expressive marks

See also Section “Expressive marks” dans *Manuel de notation*.

Mise entre parenthèses d’un signe d’interprétation ou d’une note d’un accord

La fonction \parenthesize, qui permet de mettre un objet entre parenthèses, a ceci de particulier qu’elle est associée à un objet graphique Parentheses.

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



Ajout de marques temporelles à un long glissando
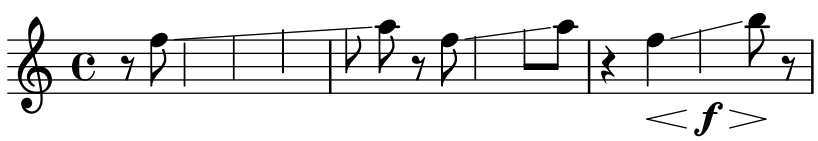
Lorsqu’un glissando s’étend dans la durée, on trouve parfois des indications temporelles, matérialisées par des hampes sans tête de note. De telles hampes permettent aussi d’indiquer des éléments intermédiaires.

L’alignement des hampes avec la ligne de glissando peut requérir quelques aménagements.

```
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 |
}
```



Ajustement du positionnement vertical d'une liaison

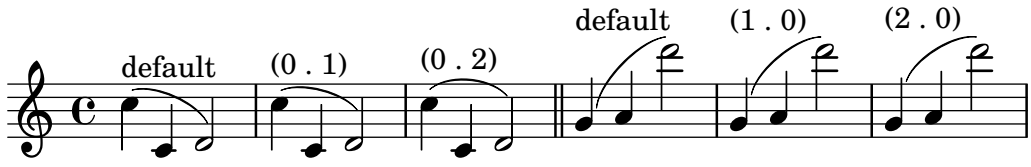
Il est possible, à l'aide d'un `\override Slur.positions`, de définir le positionnement vertical des extrémités d'une liaison à des valeurs absolues (ou, plutôt, forcer l'algoritme de LilyPond à considérer ces valeurs comme étant préférables). Dans de nombreux cas, ceci demande de procéder à tâtons pour obtenir une valeur acceptable, d'autant que la commande `\offset` est inefficace pour les liaisons et affiche un avertissement.

Le code ci-dessous parmet d'ajuster le positionnement du début et de la fin d'une liaison en spécifiant des modifications *relatives* comme le ferait un `\offset`.

Syntaxe : `\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)
}
```



Ajustement du galbe des chutes ou sauts

La propriété `shortest-duration-space` peut devoir être retouchée pour ajuster l'apparence des chutes ou sauts.

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

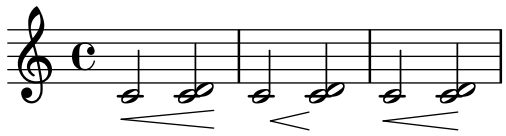
```
}
```



Alignement des bornes de soufflet relativement aux `NoteColumns`

Les terminaisons des soufflets peuvent s'aligner sur la gauche, au centre ou sur la droite des *grobs* `NoteColumn` à l'aide d'une dérogation à la propriété `endpoint-alignments` – `LEFT`, `CENTER` ou `RIGHT` – sous forme de paire représentant les extrémités gauche et droite du soufflet. Les `endpoint-alignments` devraient être des directions (soit −1, 0 ou 1), les autres valeurs entraînant l'émission d'un avertissement. L'extrémité droite d'un soufflet se terminant sur un silence ne sera pas affectée et s'alignera toujours sur le bord gauche de ce silence.

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



Brève alternative, avec deux barres verticales

Voici comment obtenir une brève – aussi appelée note carée – flanquée de deux barres verticales, au lieu d'une comme habituellement.

```
\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
}
```

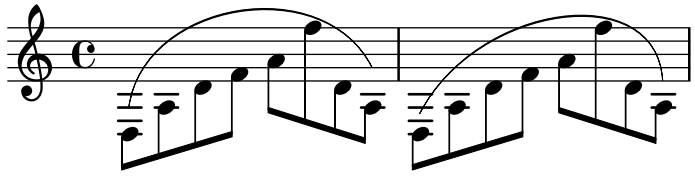


Liaison asymétrique

Une liaison peut adopter une courbe asymétrique afin de s'adapter au mieux à la ligne mélodique.

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

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



Signes de respiration

Les indications de respiration sont disponibles sous différentes formes : virgule (par défaut), trait, en V ou en « voie de chemin de fer » (césure).

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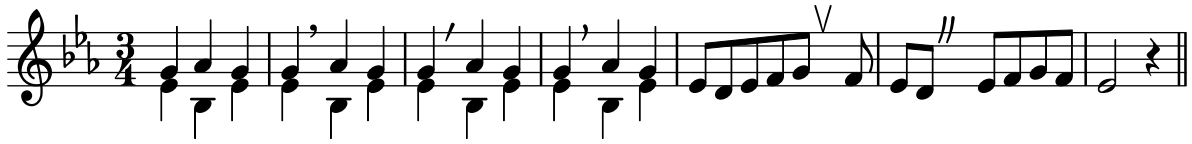
  % rvarcomma and lvarcomma are variations of the default rcomma
  % and lcomma
  % N.B.: must use Staff context here, since we start a Voice below
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % 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 "||"
}
```



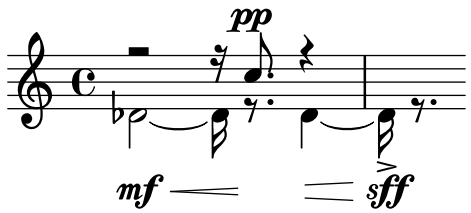
Soufflet de crescendo partiellement interrompu

Une portion d'un soufflet de crescendo peut être rendue invisible. Il suffit pour cela de dessiner un rectangle par dessus ce tronçon, ce qui aura pour effet de le rendre invisible. Ce rectangle est défini en tant que *markup* textuel.

La commande de *markup* `with-dimensions` indique à LilyPond de ne prendre en considération que l'extrémité inférieure du rectangle lors de son positionnement par rapport au soufflet. L'ajustement de la propriété `staff-padding` permet d'éviter au rectangle de venir s'intercaler entre le soufflet et la portée.

Le soufflet doit se trouver à un niveau inférieur à celui du *markup* afin que le dessin du rectangle puisse effectivement le recouvrir.

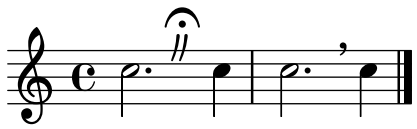
```
\relative c' {
  <<
    {
      \dynamicUp
      r2 r16 c'8.\pp r4
    }
    \\
    {
      \override DynamicLineSpanner.layer = #0
      des,2\mf\< ~
      \override TextScript.layer = #2
      \once\override TextScript.staff-padding = #6
      \once\override TextScript.vertical-skylines = #'()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



Césure en forme de « voie de chemin de fer » avec point d'orgue

Une césure peut parfois s'indiquer par une double respiration surmontée d'un point d'orgue. Le code ci-dessous permet d'obtenir la combinaison répondant à cette esthétique.

```
\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 "|."
}
```



Texte centré sous un soufflet

La fonction comprise dans l'extrait suivant permet d'ajouter du texte – comme « molto » ou « poco » – à un soufflet de (de)crescendo. Le texte ajouté viendra se positionner selon le positionnement du soufflet. Le soufflet s'aligne avec un objet DynamicText.

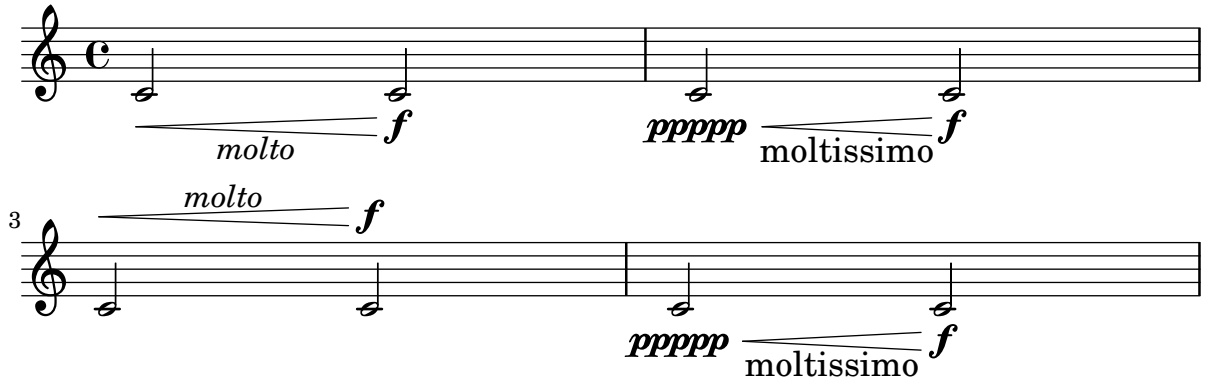
Cet exemple présente aussi comment, à l'aide de code Scheme, influencer la manière dont un objet est normalement imprimé.

```
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
}
```
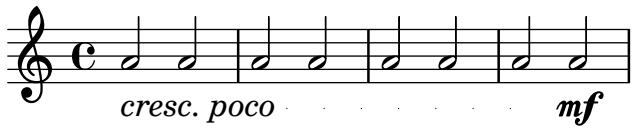


Modification du texte et de l'extension de nuances textuelles

Le texte par défaut des crescendos et decrescendos se change en modifiant les propriétés de contexte `crescendoText` et `decrescendoText`.

L'aspect de la ligne d'extension est fonction de la propriété `style` du `DynamicTextSpanner`. Sa valeur par défaut est `dashed-line`, mais d'autres valeurs sont disponibles, comme `line`, `dotted-line` et `none`.

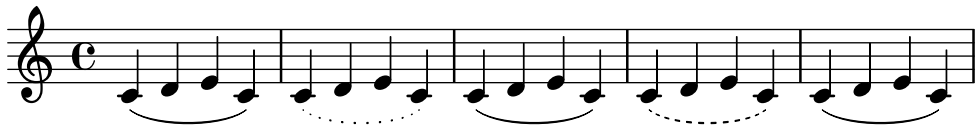
```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Modification de l'aspect des liaisons d'articulation

Une liaison d'articulation peut se présenter sous la forme d'un trait continu ou discontinu, voire en pointillé.

```
\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)
}
```



Modification de l'indicateur de respiration

On peut choisir le glyphe imprimé par cette commande, en modifiant la propriété `text` de l'objet `BreathingSign`, pour lui affecter n'importe quelle indication textuelle.
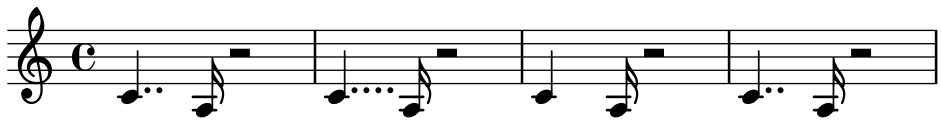
```
\relative c'' {
  c2
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  \breathe
  d2
}
```



Spécification du nombre de points d'augmentation d'une note

Le nombre de points d'augmentation affectés à une note en particulier peut se modifier indépendamment des points placés après la note, à l'aide de la propriété `dot-count` de l'objet `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 |
}
```
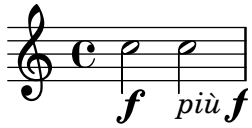


Combinaison de nuance et de texte

Certaines indications de nuance peuvent requérir un complément textuel, comme « più *f* » ou « *p* subito ». Elles se réalisent aisément à l'aide d'un bloc `\markup`. Le résultat se comportera comme un objet `TextScript`.

Voir aussi `Combinaison de nuance et de texte (2)`.

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

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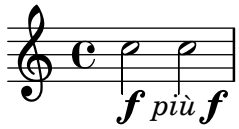


Combinaison de nuance et de texte (2)

Certaines indications de nuance peuvent requérir un complément textuel, comme « più *f* » ou « *p* subito ». Elles se réalisent à l'aide de la fonction Scheme `make-dynamic-script`. Les objets alors créés se comportent somme des objets graphiques `DynamicText`.

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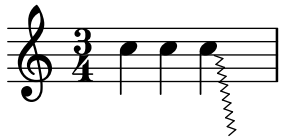
\score {
  \relative c'' {
    c2\f c\piuF
  }
}
```



Glissando contemporain

De nos jours, il peut arriver que la note d'arrivée d'un glissando soit absente de la partition. Pour ce faire, il vous faudra utiliser une cadence et « masquer » la note d'arrivée.

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



Contrôle de la visibilité des extensions d'objet après saut de ligne

La visibilité des extensions qui se terminent sur la première note après un saut de ligne est contrôlée par un appel de `after-line-breaking` à la fonction `ly:spanner::kill-zero-spanned-time`.

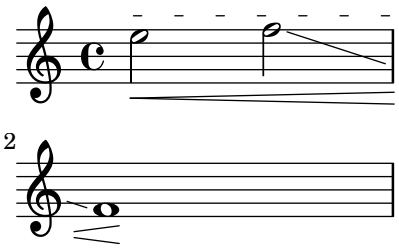
Pour des objets tels qu'un glissando ou un soufflet, le comportement par défaut est de ne pas être reportés après un saut de ligne. L'extension sera donc reprise dès lors que l'appel aura été désactivé.

Il en va inversement pour les extensions qui, par défaut pour les textes notamment, sont reportées après un saut de ligne ; il faudra alors activer l'appel pour empêcher leur report.

```
\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
}
```

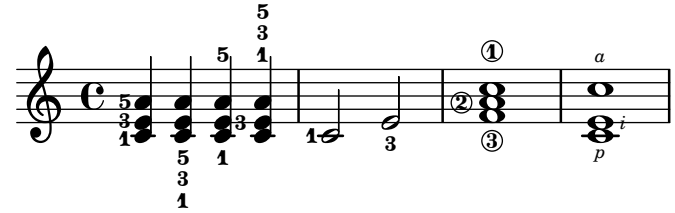


Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise à l'aide de la propriété `fingeringOrientation`. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord <. . .>, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue à l'aide des propriétés `stringNumberOrientation` et `strokeFingerOrientation`.

Ces propriétés peuvent prendre une liste de trois valeurs. Elles contrôlent si les informations doivent être placées au-dessus (en présence d'un `up`) ou au-dessous (en présence d'un `down`), à gauche (en présence d'un `left` ou à droite (en présence d'un `right`). Par contre, si un positionnement n'est pas mentionné, aucun doigté n'y figurera. LilyPond tient compte de ces contraintes et trouvera le meilleur emplacement pour le doigté des notes de l'accord qui suit. Bien entendu, `left` et `right` sont exclusifs l'un de l'autre – les doigtés ne peuvent se placer que d'un seul côté.

```
\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 >
}
```



Contrôle de l'ordre vertical des articulations et ornements

Les symboles s'ordonnent verticalement suivant la propriété `script-priority`. Plus sa valeur numérique est faible, plus le symbole sera proche de la note. Dans l'exemple suivant, l'objet `TextScript` – le dièse – a d'abord la propriété la plus basse et se voit donc placé au plus près de la note ; ensuite, c'est l'objet `Script` – le mordant – qui a la propriété la plus basse, et se place alors sous le dièse. Lorsque deux objets ont la même priorité, c'est l'ordre dans lequel ils sont indiqués qui détermine lequel sera placé en premier.

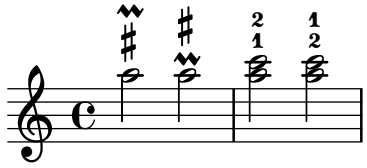
Il est à noter que pour les objets `Fingering`, `StringNumber` et `StrokeFinger`, lorsqu'ils apparaissent dans un accord, leur ordonnancement vertical est aussi déterminé par le positionnement vertical de leur tête de note d'attachement qui sera ajoutée (ou soustraite selon la direction) de la valeur de `script-priority` du *grob*. Ceci permet de s'assurer que, pour des doigtés au-dessous d'un accord, la note la plus basse soit associée au doigté le plus bas, et inversement. L'ordre dans lequel les hauteurs sont saisies dans l'accord est ici sans importance.

Par défaut, les scripts les moins techniques sont inscrits au plus près de la tête de note. L'ordre de base est articulation, flageolet, doigté, doigté main droite, numéro de corde, point d'orgue, archet et script textuel.

```
\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
}
```
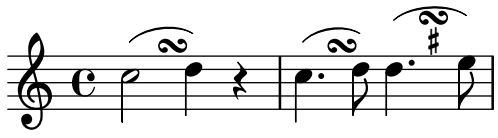


Création d'un grupetto retardé

Obtenir un *gruppetto* retardé et dans lequel la note la plus basse est altérée requiert quelques surcharges. La propriété `outside-staff-priority` doit être désactivée (`#f`) pour éviter qu'elle prenne le pas sur la propriété `avoid-slur`. L'ajustement du premier argument de `\after` (il s'agit d'une durée) permet de jouer sur le positionnement 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
```
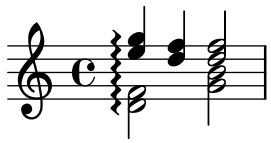
```
  }
  d4.( e8)
}
```



Arpège distribué sur plusieurs voix

Affecter le graveur Span_arpeggio_engraver au contexte de la portée (Staff) permet de distribuer un arpège sur plusieurs voix.

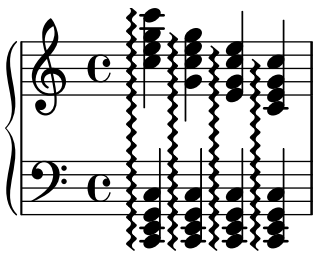
```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Arpège distribué sur une partition pour piano

Dans une double portée pour piano (PianoStaff), un arpège peut s'étendre sur les deux portées grâce à la propriété 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
  }
>>
```



Arpège distribué pour un autre contexte que le piano

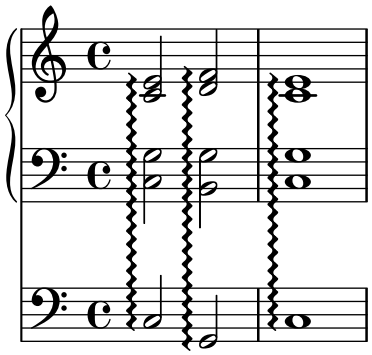
Il est possible de distribuer un arpège sur plusieurs portées d'un système autre que le GrandStaff et ses semblables (PianoStaff, +ChoirStaff et StaffGroup) dès lors que vous incluez le Span_arpeggio_engraver au contexte Score.

```
<<
  \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"
```

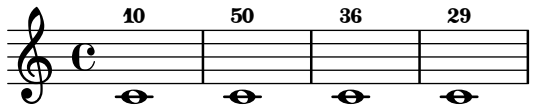
```
  }
}
```



Création de doigtés sur deux caractères

Il est tout à fait possible de noter un doigté supérieur à 5.

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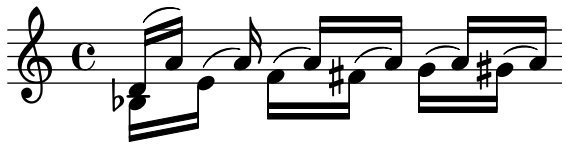


Création d'une liaison entre plusieurs voix

Dans certaines situations, il peut être nécessaire de lier des notes appartenant à des voix différentes. La solution consiste à ajouter une note invisible dans l'une des voix, à l'aide de la commande `\hideNotes`.

Voici par exemple la mesure 235 de la chaconne de la seconde partita pour violon 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)
```

```
    }
  >>
}
```



Création d'extensions de texte

Les commandes \startTextSpan et \stopTextSpan permettent d'ajouter une ligne de prolongation aux indications textuelles, à l'instar des indications de pédale ou d'octaviation. Jouer sur les propriétés de l'objet TextSpanner permet d'en modifier le rendu.

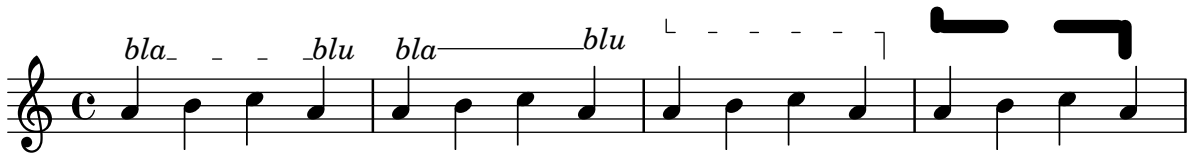
```
\paper { ragged-right = ##f }

\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
}
```
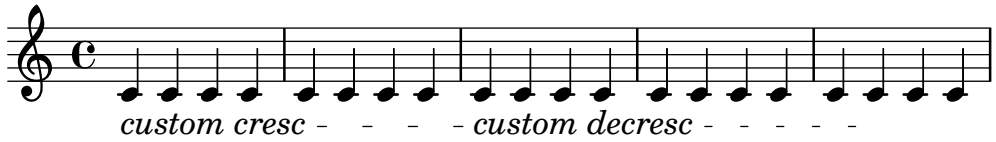


Personnalisation des extenseurs de nuance textuelle

Il s'agit de fonctions postfix pour personnaliser l'extension des crescendos textuels. L'extension devrait débuter sur la première note de la mesure. Il faut utiliser `-\mycresc` – comme une articulation – sous peine que le départ de l'extension n'apparaisse qu'à la note suivante.

```
% 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 par dessus un objet graphique

Un glissando peut sauter un objet `NoteColumn`.

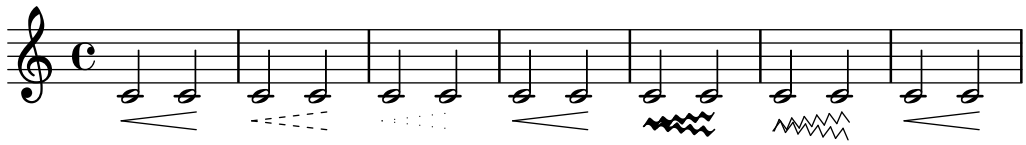
```
\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}
```



Stylisation des lignes de soufflet

Les lignes d'un soufflet peuvent adopter tous les styles permis par la `line-interface` : `dashed-line` (discontinu), `dotted-line` (pointillé), `line` (continu), `trill` (ondulé) ou `zigzag`.

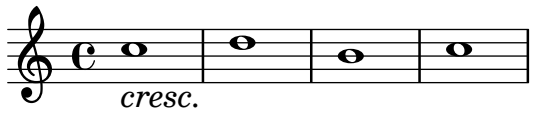
```
\relative c' {
  c2\< c\!
  \override Hairpin.style = #'dashed-line
  c2\< c\!
  \override Hairpin.style = #'dotted-line
  c2\< c\!
  \override Hairpin.style = #'line
  c2\< c\!
  \override Hairpin.style = #'trill
  c2\< c\!
  \override Hairpin.style = #'zigzag
  c2\< c\!
  \revert Hairpin.style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



Masquage de l'extension des nuances textuelles

Les crescendos et decrescendos indiqués textuellement – tels que *cresc.* ou *dim.* – sont suivis de pointillés qui montrent leur étendue. On peut empêcher l'impression de ces pointillés.

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



Alignement de nuances personnalisées comme « più f »

Certaines indications de nuance requièrent un complément textuel, comme « sempre **pp** ». Dans la mesure où les nuances sont habituellement centrées sous la note, le *pp* se trouvera repoussé loin après la note à laquelle il s'applique.

Différentes approches permettent de correctement aligner horizontalement ce « sempre **pp** » comme s'il ne s'agissait que d'un simple *pp* :

- Un simple `\once \override DynamicText.X-offset = #-9.2` avant la note considérée de telle sorte que la nuance soit décalée manuellement à la bonne place. Inconvénient : il faut le faire manuellement à chaque fois qu'intervient ce *markup*.
- L'intégration d'un rembourrage (`#:hspace 7.1`) à la définition de cette nuance personnalisée afin qu'une fois centrée par LilyPond elle soit correctement alignée. Inconvénient : le rembourrage occupera exactement cet espace et ne permettra à aucun autre *markup* ou nuance d'apparaître à cet endroit.
- L'application d'un décalage au script de nuance `\once \override ... .X-offset = ...`. Inconvénient : là aussi il faut le faire à chaque fois.

- L'attribution arbitraire d'une dimension à 0 du texte additionnel à l'aide d'un `#:with-dimensions '(0 . 0) '(0 . 0)`. Inconvénient : LilyPond considère que « sempre » n'occupe pas d'espace, et donc pourra mettre à cet endroit d'autres éléments, ce qui pourrait générer des collisions qui ne seront pas détectées par les routines *ad hoc*. D'autre part, il semble persister un espacement, ce qui laisse l'impression d'un alignement différent en l'absence de texte additionnel.
- L'ajout, explicite, du décalage directement dans la fonction Scheme du script de nuance.
- La définition d'un alignement explicite au sein du script de nuance. Ceci ne sera suivi d'effet, par défaut, qu'en jouant sur `X-offset`. Inconvénient : il faut définir `DynamicText.X-offset`, ce qui s'appliquera à toutes les nuances textuelles. Par ailleurs, l'alignement sera réalisé sur le bord droit du texte additionnel, non sur le milieu de *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")))

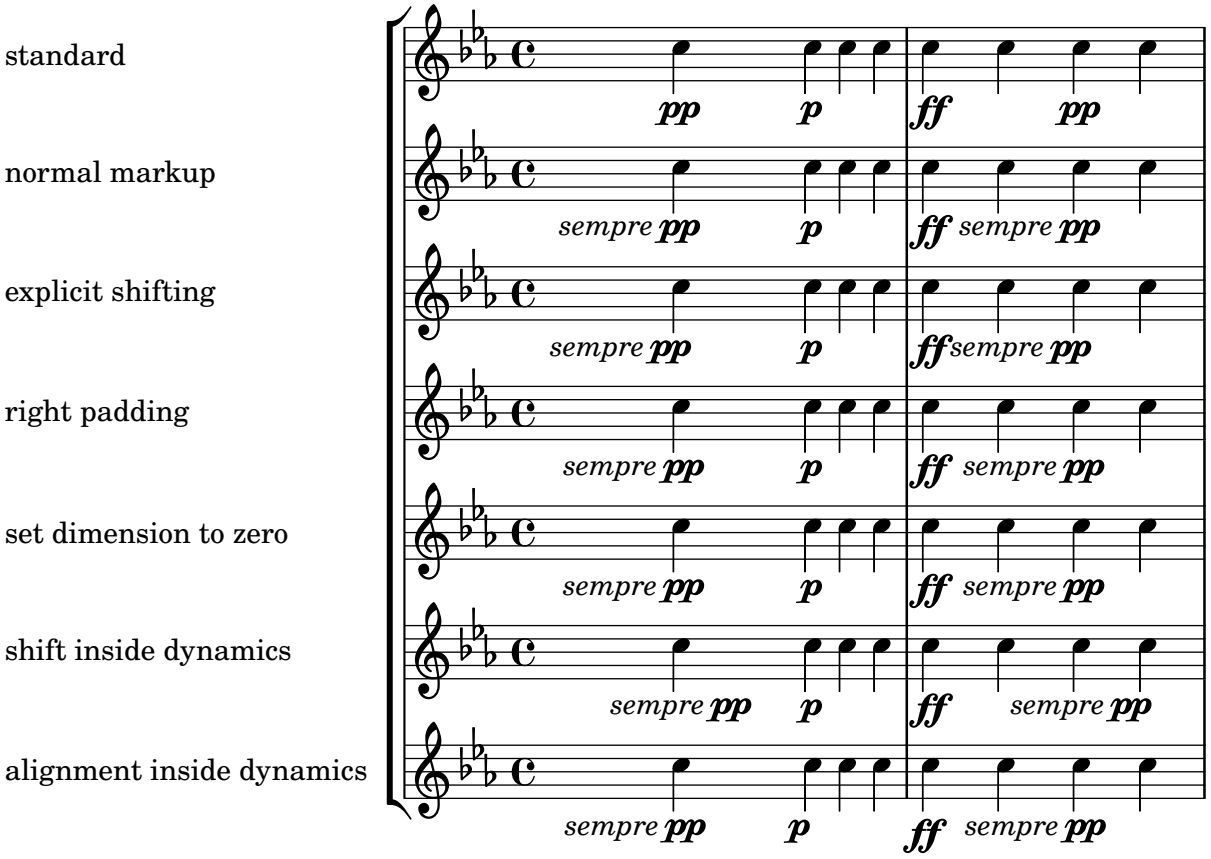
% Solution 5: Dynamic with explicit shifting inside the scheme function
semppG =
#(make-dynamic-script
  (markup #:hspace 0
          #:translate '(-18.85 . 0)
          #:line (#:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

% 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 }
```
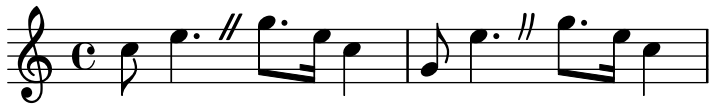


Insertion d'une césure

Une surcharge de la propriété `text` de l'objet `BreathingSign` permet de créer une marque de césure.

LilyPond dispose également d'une variante courbée.

```
\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
}
```
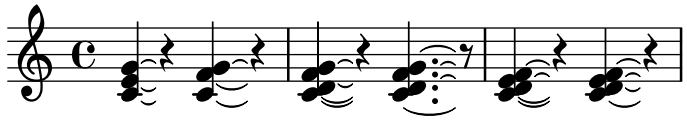


Liaison « Laissez vibrer »

Les indications de « laisser vibrer » ont une taille fixe. Leur formatage est accessible au travers de la propriété `tie-configuration`.

```
\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
}
```



Terminaison de ligne en flèche

Les extensions de texte, tout comme les indications sous forme de ligne tel un glissando, peuvent se voir pourvues d'une extrémité en flèche.

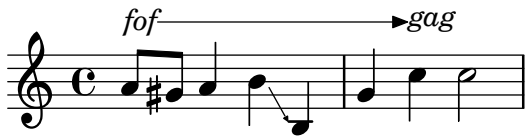
```
\relative c'' {
  \override TextSpanner.bound-padding = #1.0
  \override TextSpanner.style = #'line
  \override TextSpanner.bound-details.right.arrow = ##t
  \override TextSpanner.bound-details.left.text = #"fof"
  \override TextSpanner.bound-details.right.text = #"gag"
  \override TextSpanner.bound-details.right.padding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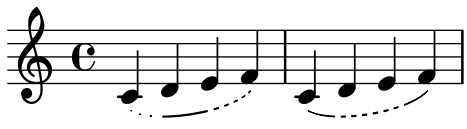


Dessin d'une liaison d'articulation au trait discontinu

Une liaison d'articulation peut être formée de traits discontinus variables. La propriété `dash-definition` se compose d'une liste de segments de liaison. Chaque segment contient une liste de paramètres qui déterminent le comportement du trait pour une section de la liaison.

Les segments de cette liaison se définissent selon le paramètre de Bézier `t` qui est compris entre 0 (l'extrémité gauche de la liaison) et 1 (l'extrémité droite de la liaison). Chaque segment se composera selon la liste (*t-début t-fin segment-style segment-taille*). La portion de liaison allant de *t-début* à *t-fin* aura un trait *segment-style* de longueur *segment-taille*. *segment-taille* est exprimé en espace de portée ; un *segment-style* à 1 donnera un trait plein. *segment-taille* s'exprime en espace de portée.

```
\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)
}
```



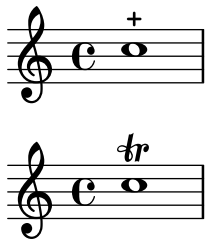
Modification de la signification des raccourcis pour les signes d'articulation

Les raccourcis sont répertoriés dans le fichier `ly/script-init.ly`, dans lequel on retrouve les variables `dashHat`, `dashPlus`, `dashDash`, `dashBang`, `dashLarger`, `dashDot` et `dashUnderscore` ainsi que leur valeur par défaut. Ces valeurs peuvent être modifiées selon vos besoins. Il suffit par exemple, pour affecter au raccourci `-+` (`dashPlus`) le symbole du trille en lieu et place du `+` (caractère plus), d'assigner la valeur `\trill` à la variable `dashPlus` :

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

dashPlus = \trill

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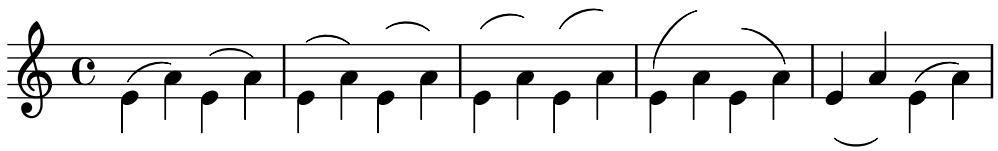


Déplacement vertical des liaisons d'articulation

Le positionnement vertical d'une liaison se contrôle par la propriété `positions` de l'objet `Slur`. Cette propriété dispose de deux paramètres : le premier affecte l'extrémité gauche de la liaison, le second son extrémité droite. La valeur des paramètres n'aura aucune influence sur le galbe de la liaison. LilyPond ne s'en servira que pour adapter le positionnement de la courbe. Des valeurs positives décalent la liaison vers le haut et s'utilisent pour des hampes descendantes. Des valeurs négatives entraînent un décalage vers le bas.

Voir aussi l'exemple « Ajustement du positionnement vertical d'une liaison ».

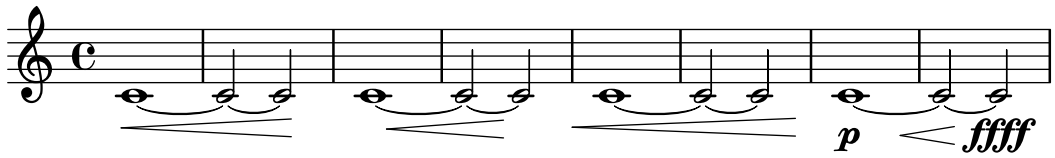
```
\relative c' {
  \stemDown
  e4( a)
  \override Slur.positions = #'(1 . 1)
  e4( a)
  \override Slur.positions = #'(2 . 2)
  e4( a)
  \override Slur.positions = #'(3 . 3)
  e4( a)
  \override Slur.positions = #'(4 . 4)
  e4( a)
  \override Slur.positions = #'(5 . 5)
  e4( a)
  \override Slur.positions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



Déplacer les extrémités de soufflets

Les terminaisons des soufflets peuvent être décalées en jouant sur la propriété `shorten-pair` de l'objet`Hairpin`. Les valeurs positives déplacent vers l'intérieur, les négatives vers l'extérieur. Contrairement à la propriété `minimum-length`, cette propriété n'affecte que l'apparence du soufflet sans ajuster l'espacement horizontal (y compris avec les nuances textuelles). Cette méthode est donc appropriée aux situations dans lesquelles un soufflet requiert un ajustement fin dans l'espace qui lui est alloué.

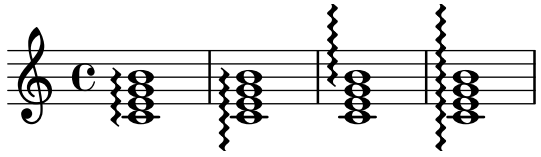
```
{
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(2 . 2)
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(-2 . -2)
  c'1~\<
  c'2~ c'\!
  c'1~\p-\tweak shorten-pair #'(2 . 0)\<
  c'2~ c'\ffff
}
```



Positionnement des arpeggios

L'ajustement de la taille d'une indication d'arpeggio s'effectue au travers du positionnement de ses extrémités haute ou basse.

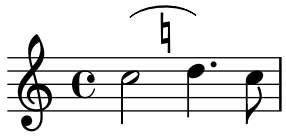
```
\relative c' {
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 0)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(0 . 5)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 5)
  <c e g b>1\arpeggio
}
```



Positionnement d'une annotation à l'intérieur d'une liaison

Lorsqu'une annotation doit s'incrire à l'intérieur d'une liaison, la propriété `outside-staff-priority` doit être désactivée.

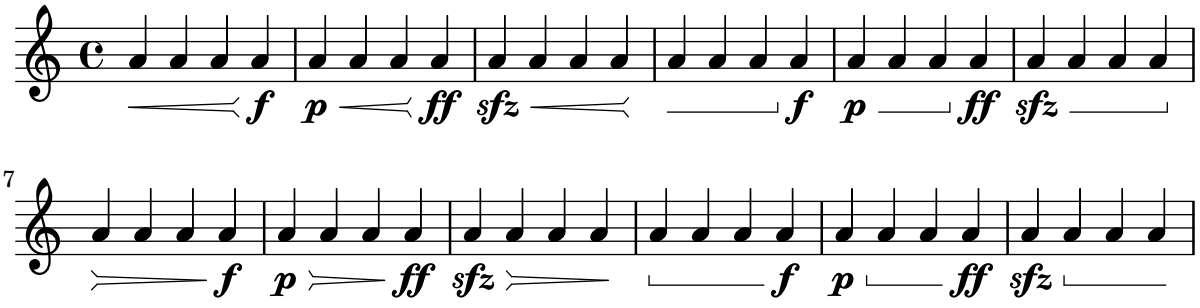
```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```



Différents styles de soufflet

Les soufflets de nuance peuvent adopter des styles différents.
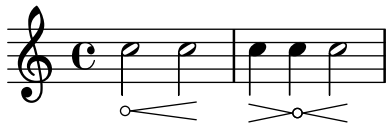
```
\relative c'' {
  \override Hairpin.stencil = #flared-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #flared-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
}
```



Impression de soufflets « al niente »

Des crescendos ou decrescendos *al niente* peuvent être indiqués de manière graphique, en assignant *vrai* (`#t`) à la propriété `circled-tip`, ce qui affiche un cercle à leur extrémité.

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



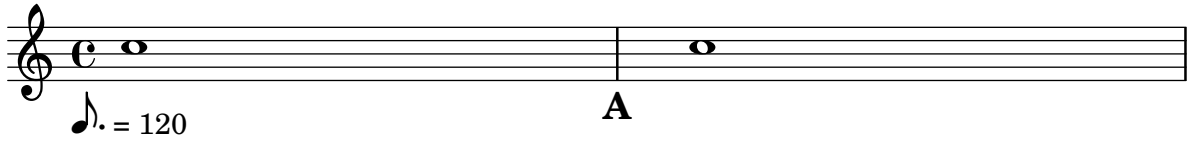
Impression du métronome et des repères sous la portée

Les indications de tempo et les marques de repère s'impriment par défaut au-dessus de la portée. Le fait de régler en conséquence la propriété `direction` des objets `MetronomeMark` et `RehearsalMark` les placera au-dessous de la portée.

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

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

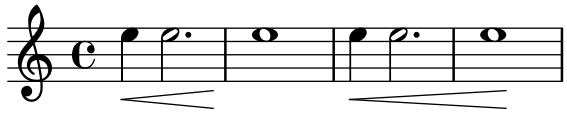
  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Soufflets et barres de mesure

En principe, un soufflet – (de)crescendo imprimé sous forme graphique – commence au bord gauche de la note de départ, et se termine au bord droit de la note d'arrivée. Cependant, si la note d'arrivée est sur un premier temps, le soufflet s'arrêtera au niveau de la barre de mesure qui la précède. Ce comportement peut être annulé en assignant *faux* (`#f`) à la propriété `to-barline`.

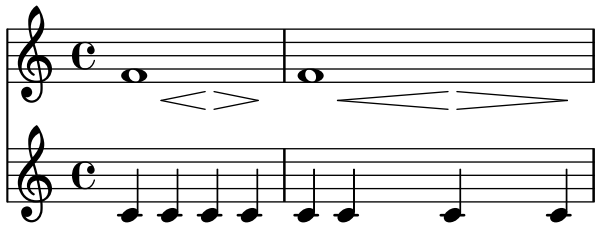
```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



Ajustement de la longueur d'un soufflet

Si un soufflet est trop court, il suffit d'ajuster la propriété `minimum-length` de l'objet `Hairpin` pour l'allonger.

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



Impression d'une même articulation des deux côtés d'une note ou d'un accord

LilyPond ne permet pas, par défaut, qu'une même articulation (accent, flageolet, point d'orgue, etc.) se retrouve à la fois au-dessus et au-dessous d'une note. Par exemple, `c4_\fermata^\fermata` ne donnera qu'un seul point d'orgue en dessous du do ; celui du dessus sera tout bonnement ignoré.

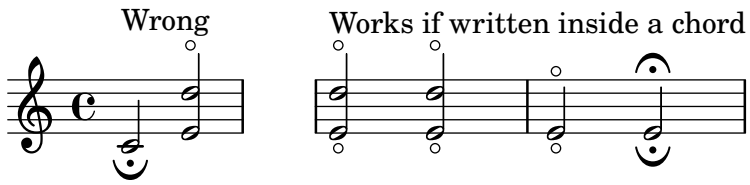
On peut néanmoins accoler des scripts, tels des doigtés, à l'intérieur d'un accord ; il peut donc y avoir autant d'articulations que de besoin, ce qui, par voie de conséquence, permet de s'affranchir de la présence des hampes et de positionner l'articulation relativement à la tête de note comme dans le cas du flageolet ci-dessous. L'imitation du traitement d'un script externe à un accord requérant un `add-stem-support` demande de libeller la note comme étant un accord et d'ajouter les articulations au sein de la construction `<...>`.

Un simple amendement permettra de rectifier le positionnement habituel en surplomb : `<c-\tweak direction #DOWN-\fermata^\fermata>`

```
\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
}
```
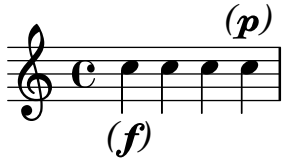


Indications de nuance entre parenthèses en italique

Le moyen le plus simple pour ajouter des parenthèses à une indication de nuance consiste à utiliser la commande \parenthesize et et d'ajuster la taille de la fonte, comme par exemple { c'4\tweak Parentheses.font-size 2 \parenthesize \f }. Toutefois, ces parenthèses se présentent droit alors que les indications de nuance apparaissent normalement en italique. Le code ci-dessous offre une solution pour présenter les parenthèses elles aussi en italique.

```
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 )
       }
    #}))

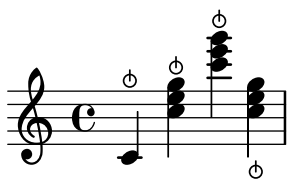
\relative c'' {
  c4\paren\f c c \dynamicUp c\paren\p
}
```



Slap ou pizzicato Bartok

Un *slap*, aussi appelé *pizzicato à la Bartók*, est un pizzicato où, au lieu de tirer la corde de côté comme à l'accoutumée, « on la pince en la soulevant plus fortement, et à la verticale, en la lâchant violemment. Celle-ci frappe alors la touche, et produit à la fois la note et le son percussif » (Wikipedia). Il s'indique par un cercle flanqué d'un trait vertical.

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

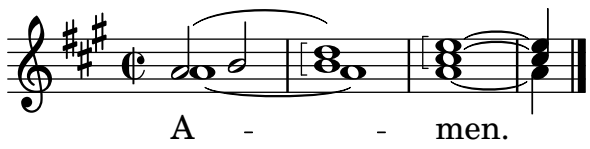


Using \arpeggioBracket to make divisi more visible

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 "|."
      }
    >>
  }
}
```



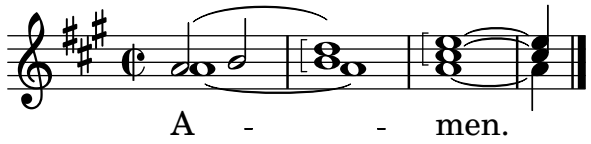
Utilisation d'un crochet pour rendre les divisions plus évidentes

La commande nonArpeggiato permet de mettre en évidence les divisions d'un pupitre en l'absence de hampe, comme on le voit régulièrement dans les partitions pour chœur.

```
\include "english.ly"

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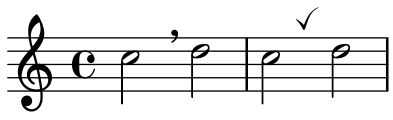
```
          <cs e>\nonArpeggiato ~
          <cs e>4
          \fine
        }
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower" {
        \voiceTwo
        a1 ~
        a
        a ~
        a4
        \fine
      }
    >>
  }
}
```



Remplacement du signe de respiration par une coche

Les musiques vocales ou pour vents utilisent souvent une coche en tant que signe de respiration. Ceci indique une respiration qui enlève une fraction à la note précédente plutôt qu'une véritable pause comme le fait un signe sous forme de virgule. La coche peut être remontée un peu afin de l'isoler de la portée.

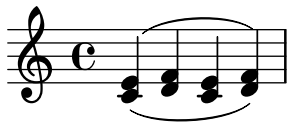
```
\relative c'' {
  c2
  \breathe
  d2
  \override BreathingSign.Y-offset = #2.6
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.tickmark" }
  c2
  \breathe
  d2
}
```



Accords et double liaison d'articulation

Certains compositeurs utilisent deux liaisons lorsqu'ils veulent lier des accords. Dans LilyPond, il faut pour cela activer la propriété de contexte `doubleSlurs`.

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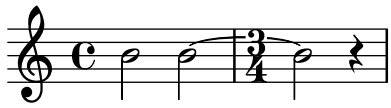


Utilisation de la propriété whiteout

Tout objet graphique peut s'imprimer sur un fond blanc afin de masquer une partie des objets qu'il recouvre. Ceci trouve toute son utilité pour améliorer certaines collisions, notamment dans des situations où un repositionnement d'objets est irréaliste. Il faut alors explicitement définir la propriété `layer` afin de contrôler quels objets seront masqués par le fond blanc.

Dans l'exemple ci-dessous, la liaison est en collision avec la métrique ; la situation est améliorée dès lors que la portion de liaison qui traverse la métrique est masquée par l'affectation de la propriété `whiteout` à l'objet `TimeSignature`. Pour ce faire, `TimeSignature` est déplacé sur un calque au-dessus de celui de `Tie` – il reste sur le calque par défaut (1) –, puis le `StaffSymbol` est placé sur un calque supérieur à celui de `TimeSignature` pour ne pas être masqué.

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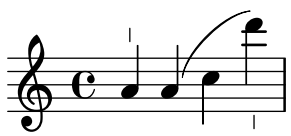


Articulation baroque en forme de coche

On trouve régulièrement, en musique baroque, cette courte ligne verticale. Sa signification peut varier, mais elle indique le plus souvent une note plus « appuyée ». Voici comment générer ce signe particulier.

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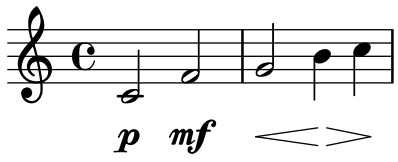
\relative c' {
  a'4^\upline a( c d')_\upline
}
```



Alignement vertical des nuances sur plusieurs notes

Des nuances qui commencent, finissent ou se produisent sur une même note auront le même alignement vertical. L'augmentation de la propriété `staff-padding` de l'objet `DynamicLineSpanner` permet d'aligner différentes nuances affectées à différentes notes.

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



4 Repeats

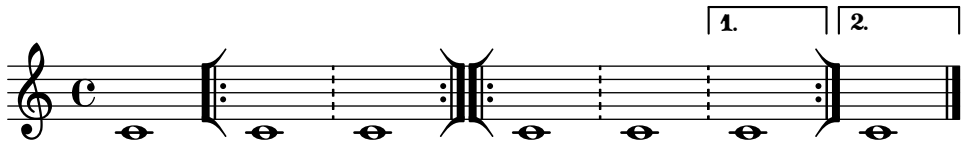
See also Section “Repeats” dans *Manuel de notation*.

Modification des barres de mesure par défaut

Les barres de mesure par défaut peuvent se modifier grâce à leur redéfinition au sein d'un contexte `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 "|."
}
```



Contrôle de l'apparence des traits de trémolo

Différentes propriétés de l'objet graphique `StemTremolo` permettent de gérer l'apparence des traits de trémolo.

- La propriété `slope` définit la pente des traits.
- La propriété `shape` détermine si les traits doivent ressembler à des rectangles (valeur `rectangle`) ou bien à des petites ligatures (valeur `beam-like`).
- La propriété `style` gère à la fois la pente et la forme selon que la note est affublée d'un crochet, est ligaturée ou n'a qu'une simple hampe. Ceci diffère des deux autres propriétés qui, elles, modifient l'apparence de façon inconditionnelle. Deux styles sont définis.
  - `default` : les traits en présence de hampe descendante et de crochet sont plus longs et plus pentus que pour des hampes avec crochet ascendantes ; les traits sur des notes ligaturées prennent l'apparence de rectangles et sont parallèles à la ligature.
  - `constant` : tous les traits ont l'apparence de petites ligatures et ont la même pente, hormis lorsque sur des hampes descendantes crochetées.

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

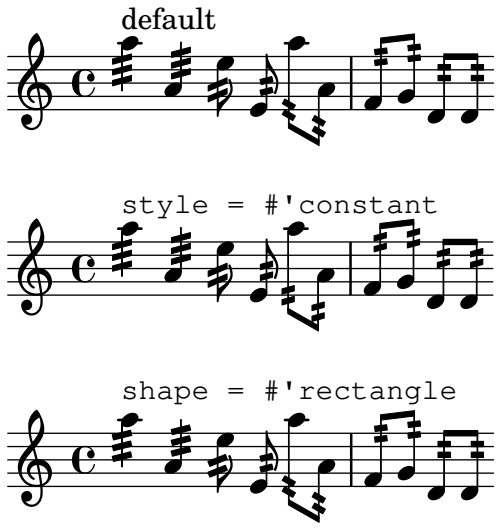
\new Staff {
  <>^\markup "default"
  \music
}

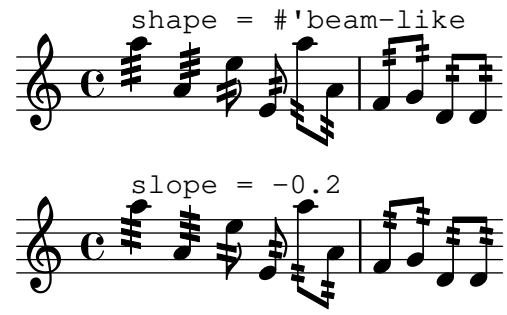
\new Staff {
  <>^\markup \typewriter "style = #'constant"
  \override StemTremolo.style = #'constant
  \music
}

\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
}
```

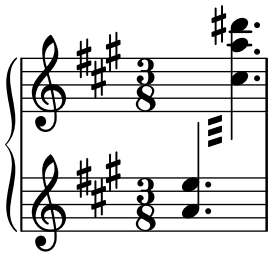




Trémolo et changement de portée

Dans la mesure où \repeat tremolo requiert deux arguments musicaux pour un trémolo d'accords, la note ou l'accord de la portée opposée doit être encadré par des accolades et se voir adjoindre la commande \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
      }
    }
  }
>>
```
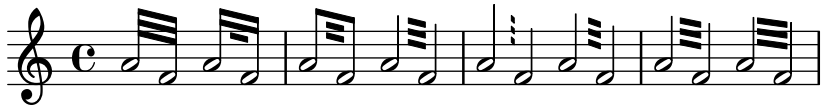


Impression de trémolo avec ligature flottante

Lorsque la durée totale d'un trémolo est inférieure à la noire, égale une blanche, ou bien entre une blanche et une ronde, il est d'usage que toutes les ligatures soient en contact avec les hampes. Certains styles de gravure font cependant apparaître des ligatures détachées, centrées entre les hampes. Pour ce type de trémolo, le nombre de hampes flottantes se gère au travers de la propriété gap-count de l'objet Beam, et l'écart entre ligature et hampe se définit par la propriété 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 }
}
```
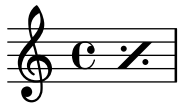


Répétition en pourcent isolée

Des symboles de pourcentage isolés peuvent aussi être obtenus au moyen d'un silence multimesure dont on modifie l'aspect :

```
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
}
```

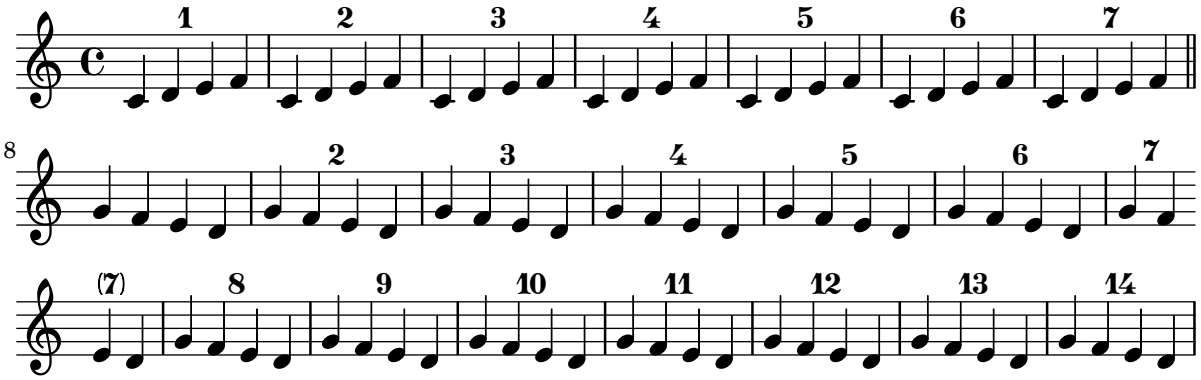


Compteur de mesures

Le code ci-dessous permet, de façon détournée, d'afficher un compteur de mesures à l'aide d'une répétition en pourcent rendue transparente.

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

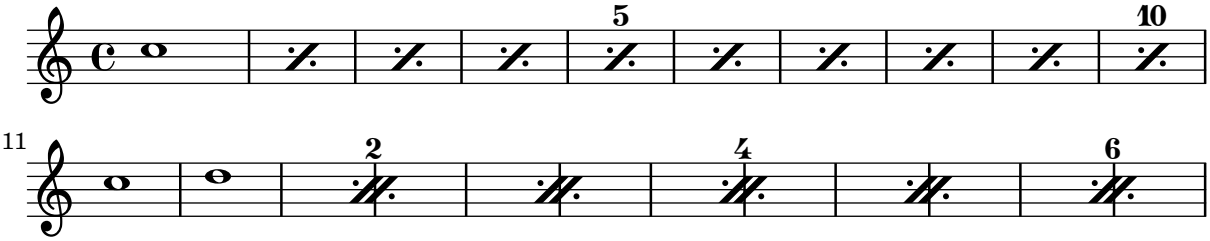
```
\new Staff {
  \startMeasureCount
  \*7 { c'4 d' e' f' }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \*5 { g'4 f' e' d' }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \*7 { g'4 f' e' d' }
  \stopMeasureCount
}
```



Affichage du numéro de répétition en pourcent

Le numéro de mesure répétée sera imprimé à intervalle régulier si vous déterminez la propriété de contexte `repeatCountVisibility`.

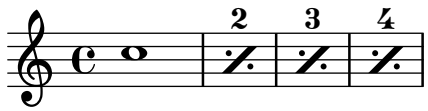
```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \%10 c1 \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \%6 { c1 d1 }
}
```



Compteur de répétition en pourcent

Les répétitions de plus de deux mesures sont surmontées d'un compteur, si l'on active la propriété de contexte `countPercentRepeats`.
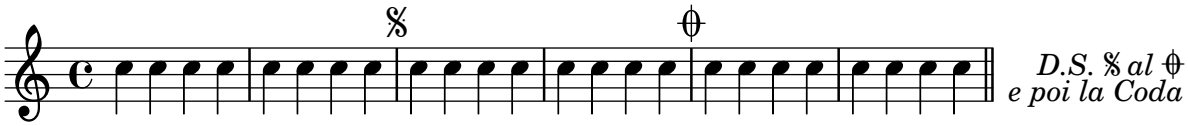
```
\relative c'' {
  \set countPercentRepeats = ##t
  \%4 c1
}
```
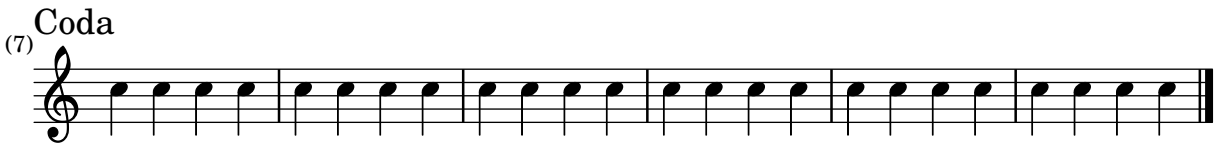


Positionnement des segno et coda (avec saut de ligne)

Le code ci-dessous permet d'adjoindre à un signe *segno* un texte *D.S. al Coda*, là où se trouverait normalement un bout de portée. La *coda* entamera une nouvelle ligne. Une variante, indiquée ici même, permet de laisser la *coda* sur la même ligne.

```
\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
}
```
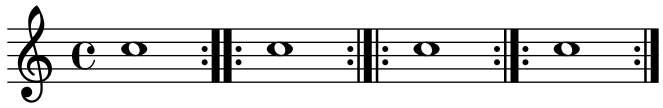




Succession de reprises et style de barre par défaut

LilyPond dispose de trois différents styles de barre pour indiquer une succession de reprises. Vous devez opter pour un style par défaut, à l'aide de la propriété de contexte `doubleRepeatBarType`.
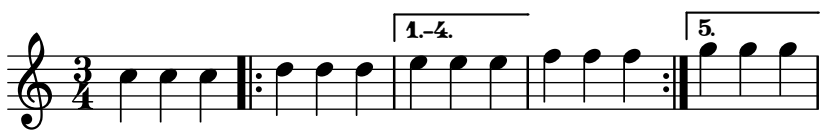
```
\relative c'' {
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":..:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":|.|:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":|.:"
  \repeat volta 2 { c1 }
}
```



Diminution de la taille du crochet d'alternative

Les crochets indiquant les fins alternatives s'étalent tout au long de celles-ci. On peut les raccourcir en jouant sur la propriété `VoltaBracket.musical-length`. Dans l'exemple suivant, le crochet ne s'étend que sur une mesure à 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 } } }
}
```



Expansion de répétition de trémolo

Actuellement, saisir *note*:*durée* est en quelque sorte un raccourci de `\repeat tremolo`, mais n'est pas expansé par une directive `\unfoldRepeats` (ceci fait l'objet du ticket 6145 (`https://`
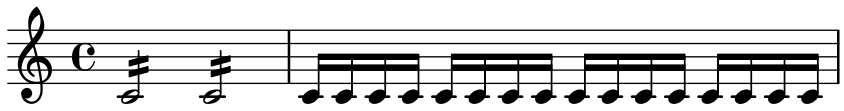
gitlab.com/lilypond/lilypond/-/issues/6145)). La fonction ci-dessous présentée constitue un contournement à cette faiblesse.

```
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
}
```

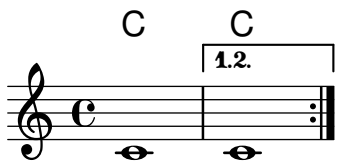


Crochet de reprise sous les chiffrages d'accord

L'ajout du Volta_engraver à la bonne portée permet d'imprimer les crochets de reprise entre les chiffrages et la portée.

```
\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
```

```
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```

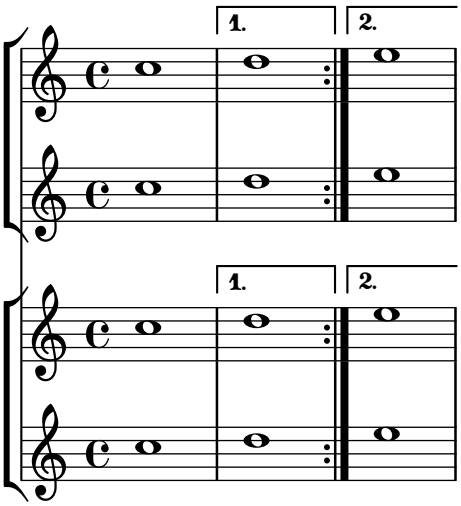


Crochets de *volta* sur plusieurs portées

L'ajout du `Volta_engraver` à une portée particulière permet que les crochets de *volta* s'affichent sur d'autres portées que la première d'une partition.

```
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
  >>
>>
```
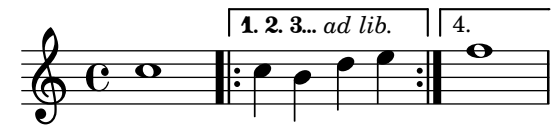


Indication de reprise avec texte grâce à `repeatCommands`

La commande `\repeat volta` permet d'indiquer facilement des reprises. Il est certains cas où l'adjonction d'un texte sous forme de `\markup` nécessite cependant de recourir à la propriété de contexte `repeatCommands`.

Dans la mesure où l'argument de `repeatCommands` est constitué d'une liste, le plus simple est de définir le texte dans une variable qui sera ensuite incorporée dans la liste en respectant la syntaxe Scheme `#`(volta` *`texteIdentificateur`*`) ...`). Les commandes de début et de fin de reprise peuvent s'ajouter séparément à la liste des éléments.

```
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
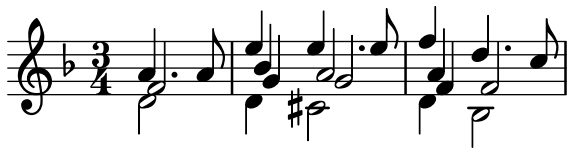
See also Section “Simultaneous notes” dans *Manuel de notation*.

Ajout de voix pour éviter les collisions

Dans certains cas de musique polyphonique complexe, une voix supplémentaire peut permettre d’éviter les risques de collision. Lorsque quatre voix parallèles ne suffisent pas, la fonction Scheme `context-spec-music` permet d’ajouter encore d’autres voix.

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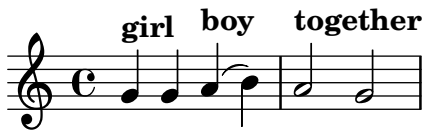
\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    \new Voice  {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \new Voice {
      \voiceTwo
      d,2
      d4 cis2
      d4 bes2
    }
    \new Voice {
      \voiceThree
      f'2
      bes4 a2
      a4 s2
    }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



Modification des indications de parties combinées

Lorsque vous regroupez automatiquement des parties, vous pouvez modifier le texte qui sera affiché pour les solos et pour les parties à l’unisson :

```
\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
    }
>>
```

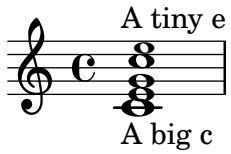


Modification de la taille d'une note particulière d'un accord

La commande \tweak permet de modifier la propriété font-size d'une note particulière d'un accord.

Il s'agit de placer, à l'intérieur même de l'accord (dans la construction < >) et avant la note considérée, l'instruction \tweak suivie de font-size et de définir la taille voulue – comme #-2 pour une petite tête.

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

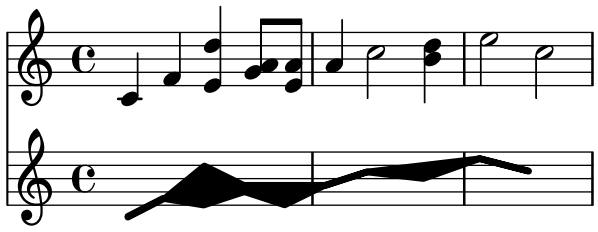


Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

```
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
>>
```



Combinaison de deux parties sur une même portée

L'outil de combinaison de parties (la commande \partCombine) permet d'avoir deux parties différentes sur une même portée. LilyPond ajoute automatiquement des indications textuelles, telles que « solo » ou « a2 ». Si votre intention n'est que de fusionner les parties, sans ajouter de texte, assignez faux à la propriété printPartCombineTexts.

Dans le cas de partitions vocales, et plus particulièrement d'hymnes, ces « solo/a2 » ne sont d'aucune utilité, aussi vaut-il mieux les désactiver. Dans le cas où il y aurait alternance entre *solo* et *tutti*, il vaut mieux faire appel à de la musique polyphonique standard.

Voici trois moyens d'imprimer deux parties sur une même portée : en polyphonie normale, avec \partCombine sans indication supplémentaire, et avec \partCombine commentée.

```
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
}
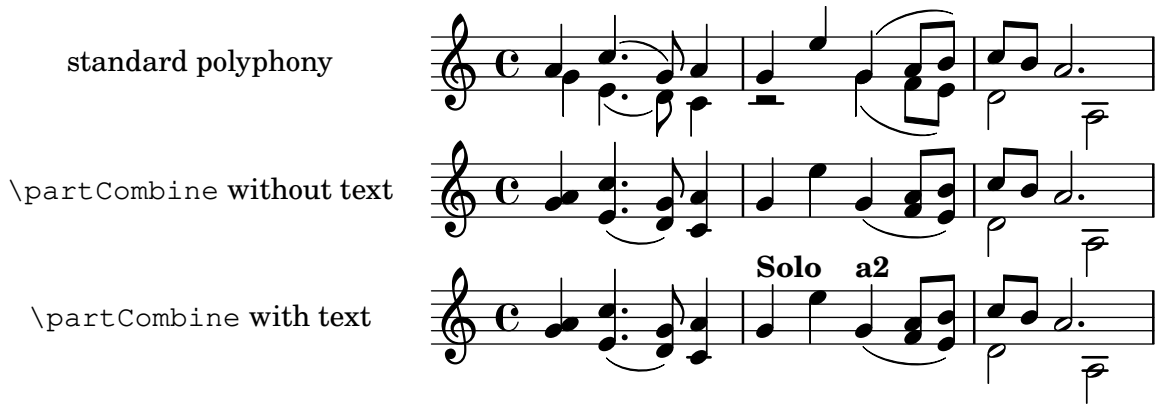
\score {
  <<
    \new Staff \with {
      instrumentName = "standard polyphony"
    } << \musicUp \\ \musicDown >>

    \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
    }
  }
}
```
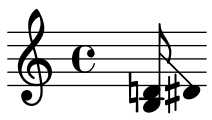


Impression d'accords complexes

Voici comment obtenir l'impression d'un accord au sein duquel une même note est jouée deux fois avec des altérations différentes.

```
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)
}

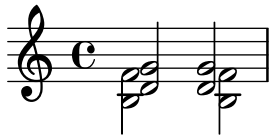

\relative c' {
  << { \fixA <b d!>8 } \\ { \voiceThree \fixB dis } >> s
}
```



Décalage horizontal forcé

Quand LilyPond est dépassé, la propriété `force-hshift` de l'objet `NoteColumn` et des silences à hauteur déterminée peuvent s'avérer utiles pour dicter au programme les choix de placement. On travaille ici en espace de portée.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
    <b f'>2
    \once \override NoteColumn.force-hshift = 1.7
    <b f'>2
  }
>>
```



Recours à la propriété transparent pour rendre des objets invisibles
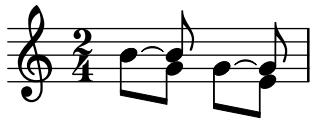
Affecter un objet graphique d'un \hide permet d'imprimer un objet « à l'encre sympathique » : l'objet n'est pas affiché bien que tous ses attributs soient préservés :

- l'objet en question occupe donc sa place,
- il est pris en compte lors de collisions, et
- il peut se voir attaché liaisons ou ligatures comme les autres.

Cet exemple illustre la manière de connecter deux voix par une liaison de prolongation. Les liaisons de prolongation ne peuvent normalement intervenir que dans la même voix. Dès lors que la liaison est entamée dans une autre voix et que la première hampe ascendante est masquée transparente dans cette même voix, la liaison semble passer d'une voix à l'autre.

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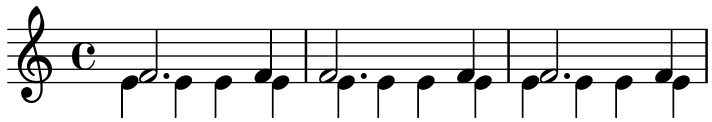
\paper {
  line-width = 40\mm
  ragged-right = ##f
}
```



Déplacement des notes pointées dans une polyphonie

Une note pointée appartenant à la voix supérieure d'une portée polyphonique sera par défaut décalée vers la droite afin d'éviter les collisions avec les autres voix. Ce comportement peut être modifié par un réglage de la propriété `prefer-dotted-right` de l'objet graphique `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 }
>>
```
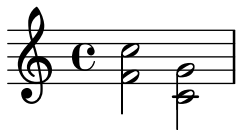


Suppression des avertissements de chevauchement

If notes from two voices with stems in the same direction are placed at the same position, and both voices have no shift or the same shift specified, the error message « warning: ignoring too many clashing note columns » appears when compiling the LilyPond file. This message can be suppressed by setting the `ignore-collision` property of the `NoteColumn` object to `#t`. Please note that this does not just suppress warnings but stops LilyPond trying to resolve collisions at all and so may have unintended results unless used with care.

```
ignore = \override NoteColumn.ignore-collision = ##t

\relative c' {
  \new Staff <<
    \new Voice { \ignore \stemDown f2 g }
    \new Voice { c2 \stemDown c, }
  >>
}
```



Deux \partCombine sur une même portée

La fonction `\partCombine` prend deux expressions musicales contenant chacune une partie et les distribue dans quatre contextes `Voice` nommés « two », « one », « solo » et « chords » selon le moment et la manière dont ces parties fusionnent en une voix commune.

Des variantes de \partCombine – \partCombineUp et \partCombineDown – permettent de fusionner des voix aux hampes ascendantes et descendantes. Les combiner alors pour présenter quatre parties sur une seule portée requiert cependant des réglages spécifiques, comme indiqué ci-dessous.

```
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.

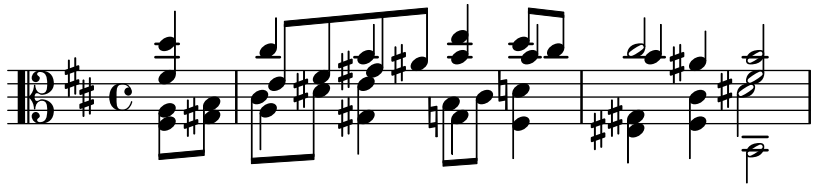
The context is called 'Down'; internally, the function calls
`\\partCombineDown`."
  #{
    \new VoiceBox = "Down" <<
      \set VoiceBox.soloText = #"Solo III"
      \set VoiceBox.soloIIText = #"Solo IV"
      \context Voice ="one" { \voiceFour }
      \context Voice ="two" { \voiceTwo }
      \context 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"
  }
}
```
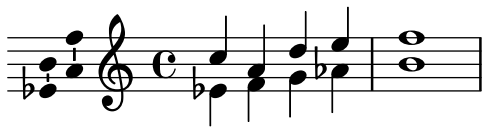


6 Staff notation

See also Section “Staff notation” dans *Manuel de notation*.

Un ambitus par voix

L’ambitus peut être individualisé par voix. Il faut en pareil cas éviter qu’ils se chevauchent.

```
\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
  }
>>
```

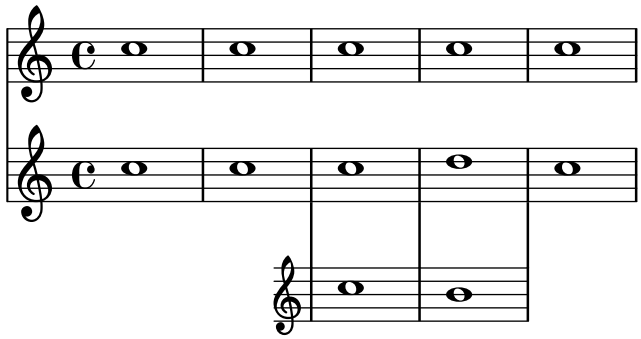


Ajout d’une portée supplémentaire

Vous pouvez ajouter une nouvelle portée, éventuellement de manière temporaire, après le début d’un morceau.

```
\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
      }
    }
  >>
```

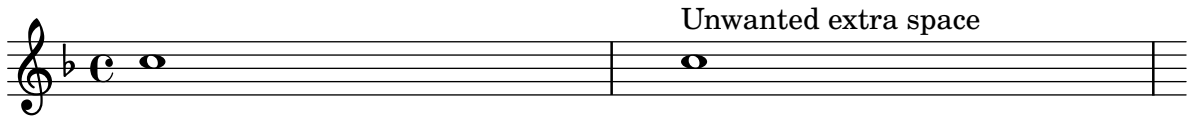
```
}
```

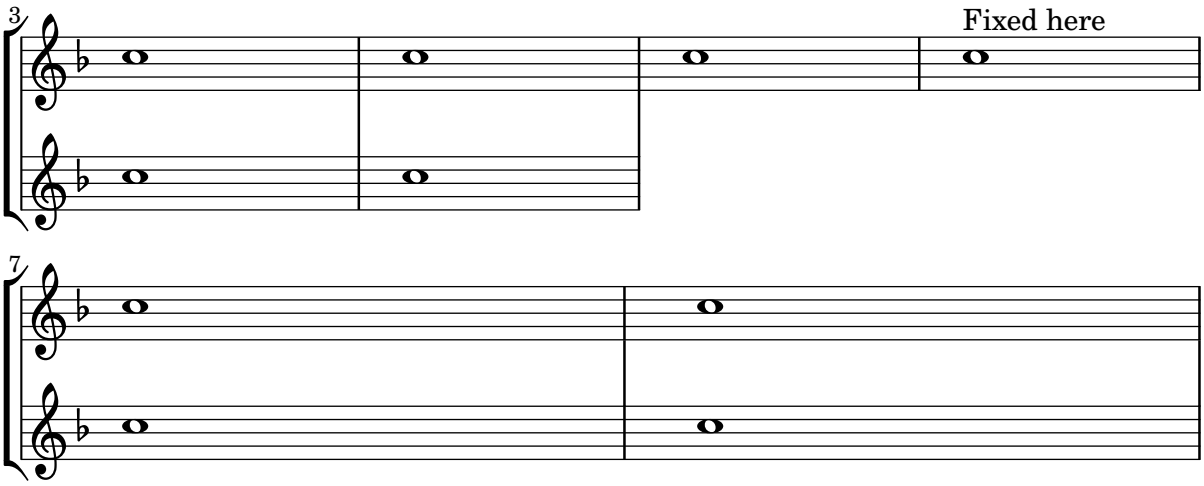


Ajout d'une portée supplémentaire après un saut de ligne

Lorsqu'une nouvelle portée vient s'ajouter après un saut de ligne, LilyPond préserve un espace juste avant le saut de ligne – pour un éventuel changement d'armure qui, quoi qu'il en soit, n'est pas imprimé. L'astuce consiste alors, comme indiqué dans l'exemple suivant, à ajuster le `explicitKeySignatureVisibility` de l'objet graphique `Staff`.

```
\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
       }
    >>
  }
}
```





Ajout d'indicateurs à une portée dédoublée après un saut de ligne

Dans cet extrait sont définies les commandes \splitStaffBarLine, \convUpStaffBarLine et \convDownStaffBarLine. Ces commandes ajoutent une double flèche après la dernière barre de mesure d'une portée, indiquant par là que ses différentes voix disposeront de leur propre portée à la ligne suivante, ou bien se regrouperont.

Il est à noter que, dans l'implémentations ici proposée, sont dessinées des flêches sans dimension dans la marge droite. Ceci ne devrait pas poser de problème pour une impression normale. Il sera cependant nécessaire d'agrandir horizontalement la boîte englobante, comme indiqué ci-après, afin d'éviter le rognage lors d'une sortie sous forme d'image.

```
#(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"
}
```

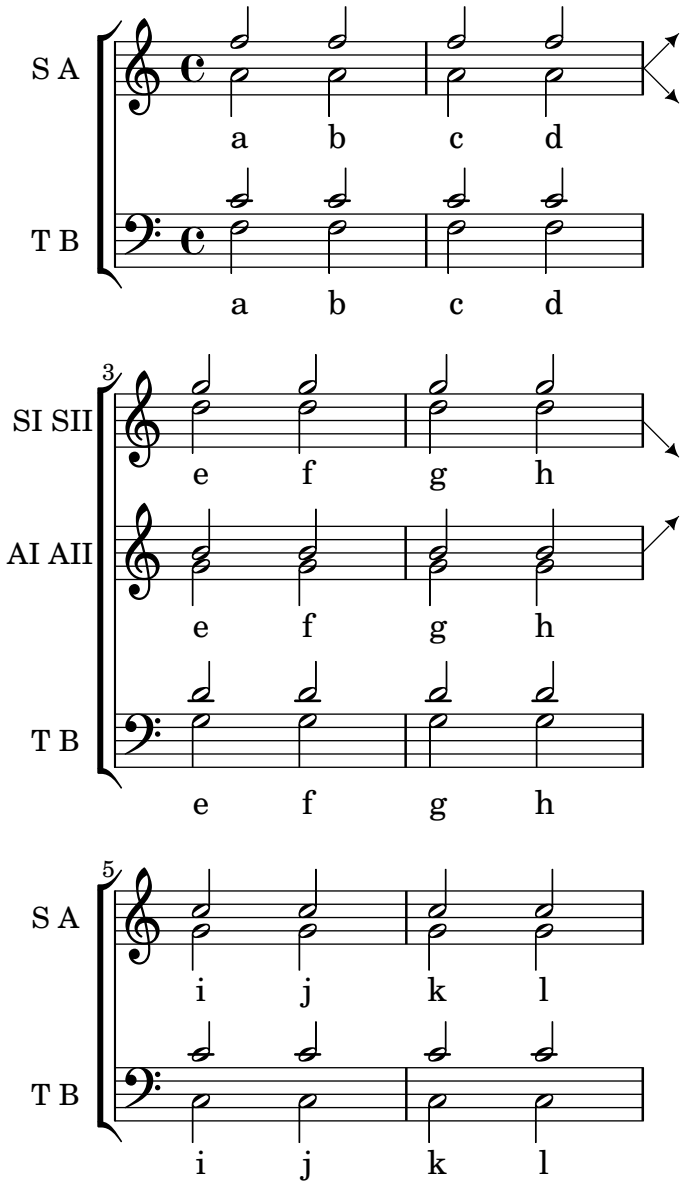
```
sI = {
  \voiceOne
  \*4 f''2
  \separateSopranos
  \*4 g''2
  \convSopranos
  \*4 c''2
}
sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \*4 d''2
}
aI = {
  \voiceTwo
  \*4 a'2
  \voiceOne
  \*4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \*4 g'2
}
aII = {
  s1*2
  \voiceTwo
  \*4 g'2
}
ten = {
  \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
      }
    }
}
```



Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur

Lorsque le nombre d'instruments cités dans la réduction pour piano se multiplie, vous pourriez avoir intérêt à créer votre propre fonction pour gérer ces repères. La fonction musicale `\cueWhile` prend quatre arguments : la musique d'où provient la citation, telle que définie par `\addQuote`, le nom qui sera mentionné en regard de cette citation, son positionnement – `UP` ou `DOWN` selon qu'il sera attribué à `\voiceOne` et placé au-dessus ou `\voiceTwo` et placé en dessous – et enfin la musique du piano qui interviendra en parallèle. Le nom de l'instrument en question viendra s'aligner sur la gauche de la citation. Bien que vous puissiez effectuer plusieurs citations, elle ne peuvent être simultanées.

```
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 }
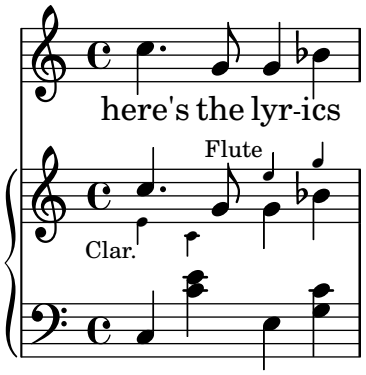
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

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
      }
    >>
  >>
```

```
}
```



Numérotation des mesures et alternatives

Un réglage de la propriété de contexte `alternativeNumberingStyle` permet d'accéder à deux méthodes alternatives de gestion de la numérotation des mesures en cas de reprises.
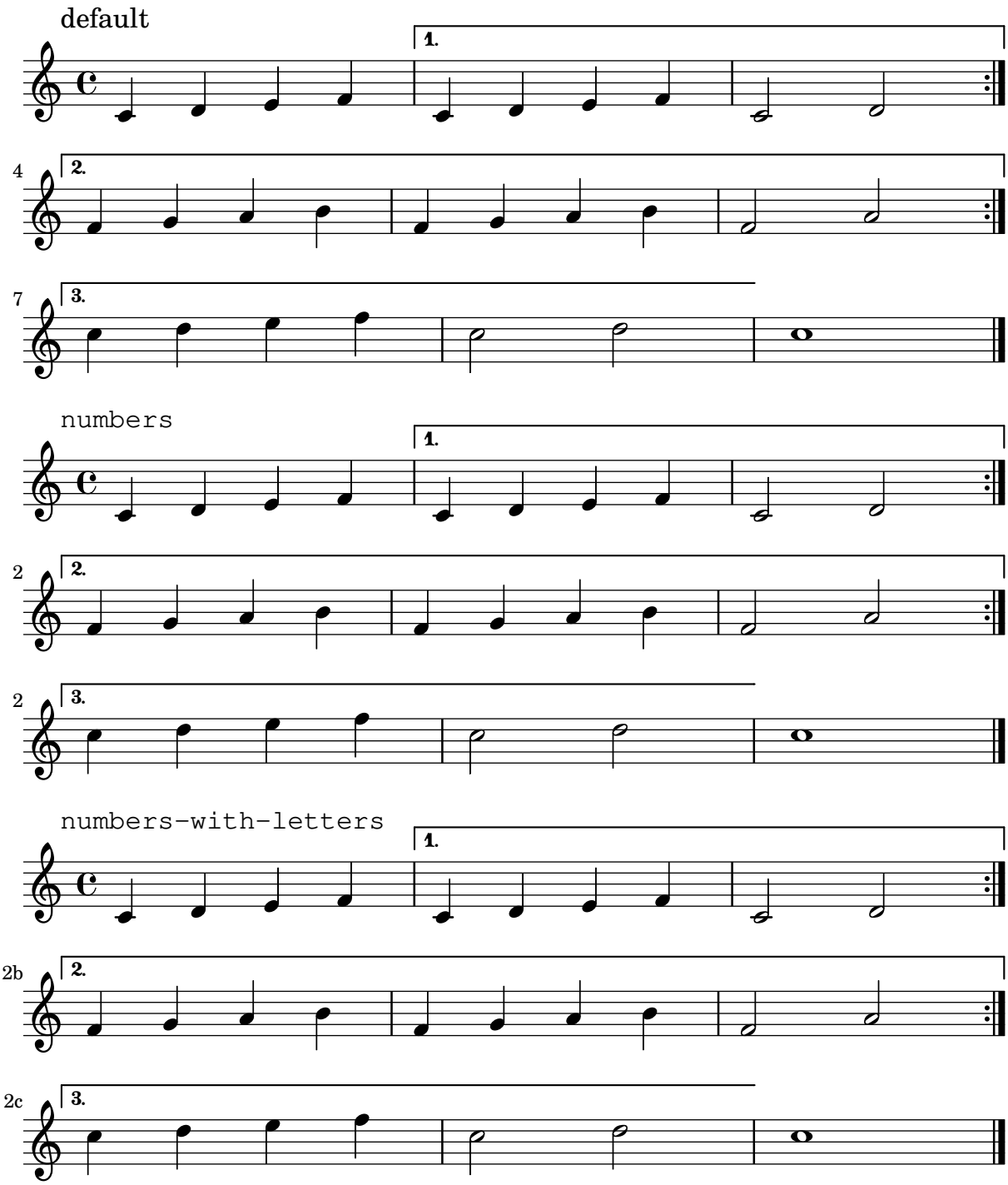
```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|."
}

{
  \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
  }
}
```
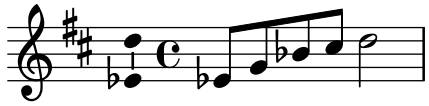


Ambitus après l'armure

L'ambitus se place par défaut à gauche de la clef. La fonction `\ambitusAfter` permet cependant de modifier ce positionnement ; sa syntaxe est `\ambitusAfter` *`grob-interface`* – voir Section "Graphical Object Interfaces" dans *Référence des propriétés internes* pour une liste des valeurs de *grob-interface* possibles.

L'un des cas d'usage est d'insérer l'ambitus entre l'armure et la métrique.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```

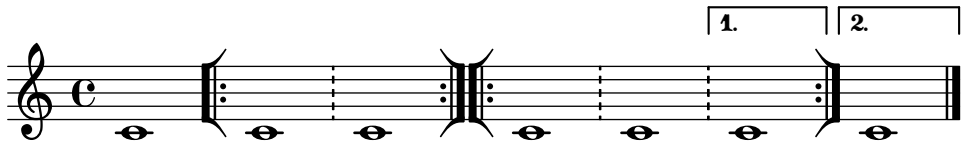


Modification des barres de mesure par défaut

Les barres de mesure par défaut peuvent se modifier grâce à leur redéfinition au sein d'un contexte 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 "|."
}
```



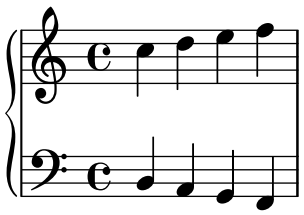
Modification du nombre de lignes de la portée

Le nombre de lignes d'une portée se modifie par adaptation de la propriété line-count du 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
    }
  >>
}
```



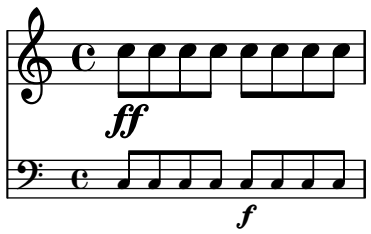
Modification de la taille d'une portée

Bien que le meilleur moyen de définir la taille des portées consiste à utiliser

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

Une portée en particulier peut se redimensionner à l'aide des propriétés `staff-space` et `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
  }
>>
```



Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le `Bar_number_engraver` du contexte `Score`, et les `Time_signature_engraver`, `Clef_engraver` et `Bar_engraver` du contexte `Staff`.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

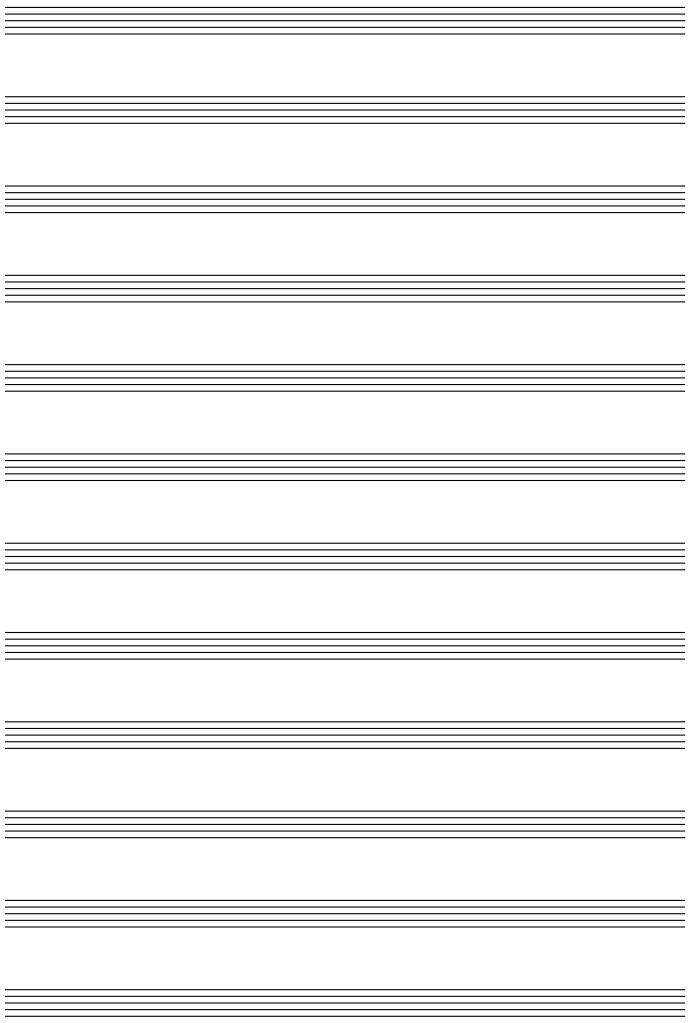
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % 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
  }
  %}
}
```



Création d'armures personnalisées
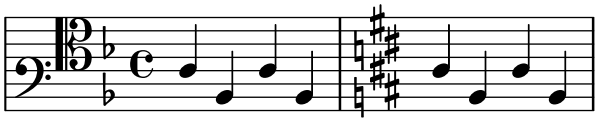
LilyPond accepte les armures personnalisées. Dans cet exemple, il s'agit d'une tonalité de ré mineur et ré majeur auxquelles on rajoute des bémols.

```
\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, |
}
```



Hampes interportées

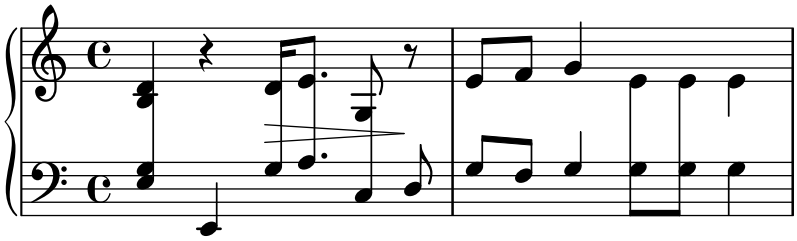
L'exemple ci-dessous illustre l'utilisation du `Span_stem_engraver` et de la commande `\crossStaff` afin de connecter des hampes entre les portées.

Nul n'est besoin de spécifier la taille des hampes ; le graveur calcule automatiquement la distance relative des têtes de note avec les portées. Il est toutefois important d'appliquer `\crossStaff` dans le contxte de voix ou de portée approprié – autrement dit en opposition à l'endroit où se trouvera une hampe) pour obtenir l'effet attendu.

```
\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 |
  }
>>
```
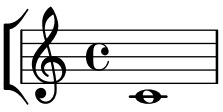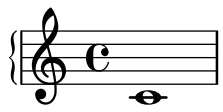


Indicateur de regroupement et portée unique

Lorsque, dans des regroupements de type `ChoirStaff` ou `StaffGroup`, une seule portée est active, aucune indication n'est donnée en début de ligne. Affecter à la propriété `collapse-height` un nombre de lignes inférieur à celui de la portée permet de modifier ce comportement par défaut.

Notez bien que dans le cas des `PianoStaff` et `GrandStaff`, pour lesquels le délimiteur de système est une accolade et non un crochet, il ne s'agit pas de la même propriété – voir le deuxième système de l'exemple.

```
\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
    }
  >>
}
```





Affichage d'un système complet lorsqu'une seule portée est active

Dans la plupart des conducteurs d'orchestre, il est d'usage qu'un instrument ou un groupe d'instruments ne joue pas pendant un moment ; leurs portées disparaissent alors pendant ce temps – les anglophones appellent cela une « partition à la française » *(Frenched score)*. LilyPond dispose de cette fonctionnalité à l'aide de la commande `\RemoveEmptyStaves`.

Il est préférable, lorsque l'un d'entre eux reprend, d'imprimer **toutes les portées du groupe en question** grâce au `Keep_alive_together_engraver` placé au niveau du regroupement, qu'il s'agisse d'un `GrandStaff` ou d'un `StaffGroup`.

Dans l'exemple suivant, les violons ne jouent pas lors du deuxième système. Bien que seul le premier violon redémarre à la dernière mesure du troisième système, la portée de second violon apparaît elle aussi.

```
\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 }
    }
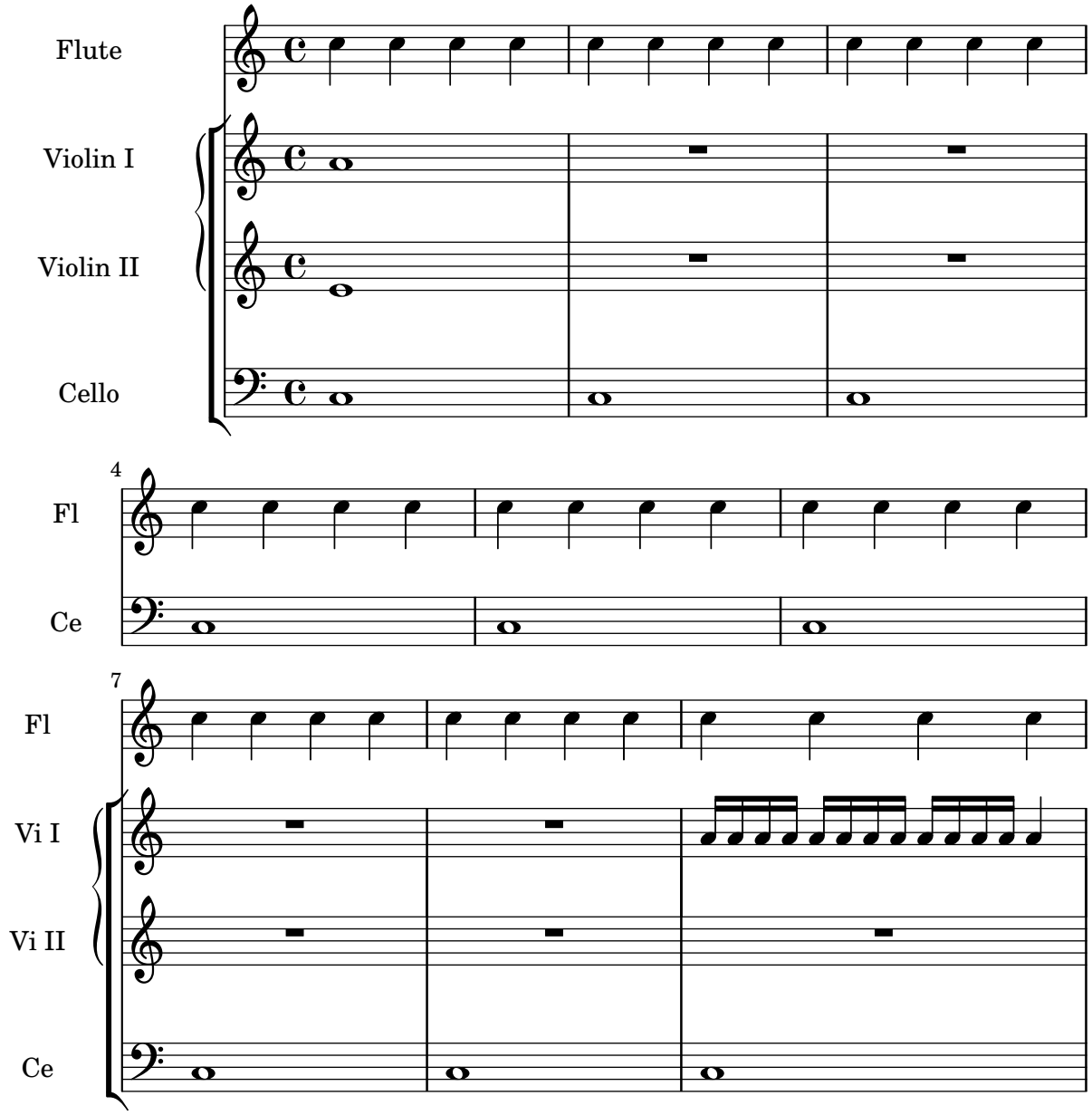
    \new StaffGroup = "StaffGroup_Strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
          shortInstrumentName = "Vi I"
        } \relative c'' {
          a1 | R1*7 | \*12 a16 a4 |
        }
        \new Staff = "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
  }
}
```



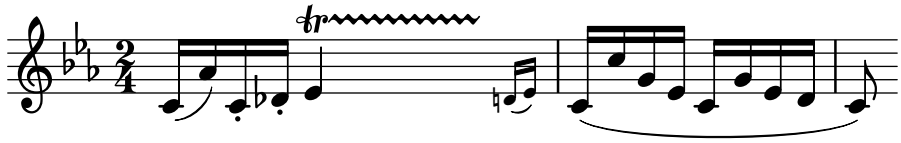
Allongement d'une marque de trille (`TrillSpanner`)

La propriété `minimum-length` d'une extension de trille – objet `TrillSpanner` – n'est effective qu'après un appel explicite à la procédure `set-spacing-rods`.

Pour ce faire, la propriété `set-spacing-rods` doit être définie à `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)
}
```



Rappel du glissando à l'occasion d'une alternative

Un glissando qui se prolonge sur plusieurs sections `\alternative` peut se rappeler à l'aide d'une note d'ornement supplémentaire et masquée, à laquelle sera attaché le départ du glissando, ce dans chaque bloc `\alternative`. Cette note d'ornement devrait avoir la même hauteur que la note où commençait le glissando originel. Ceci est géré par une fonction musicale qui prendra en argument la hauteur de la note d'ornement.

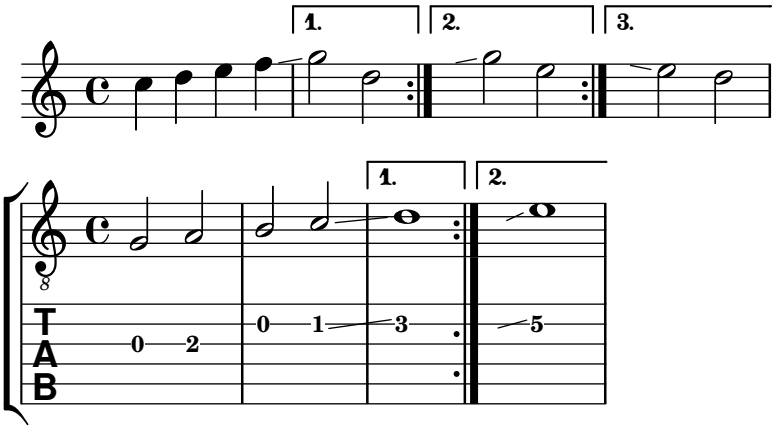
Dans le cadre d'une musique polyphonique, il ne faudra pas oublier d'ajouter une note d'ornement dans toutes les autres voix afin de préserver la synchronisation.

```
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 }
    >>
  >>
}
```



Liaison de prolongation aplatie

La fonction `flared-tie` ici présentée permet de dessiner une liaison de prolongation constituée de lignes droites. Elle est sensée remplacer la fonction par défaut de dessin des liaisons de tenue – elle remplace l'argument `stencil` de l'objet `Tie`.

L'argument de cette fonction `flared-tie` est une liste de paires de coordonnées qui spécifient des points additionnels entre l'entame et la terminaison afin de déterminer l'étendue des lignes de la liaison. L'entame et la terminaison sont identiques à ceux de la liaison originelle. La valeur des abscisses et ordonnées sont des multiples de largeur et de la hauteur de la boîte englobante de la liaison originelle (tout en tenant compte de son positionnement) ; par voie de conséquence, le premier point a pour coordonnée (0,0), et le dernier point (1,0).

La fonction `flare-tie` définit un raccourci pour l'obtention d'une liaison aplatie. Des ajustements plus poussés sont possibles, en apportant des dérogations à `Tie.details.height-limit` ou à l'aide de la fonction `\shape`. La définition personnalisée peut aussi se modifier à la volée.

```
#(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

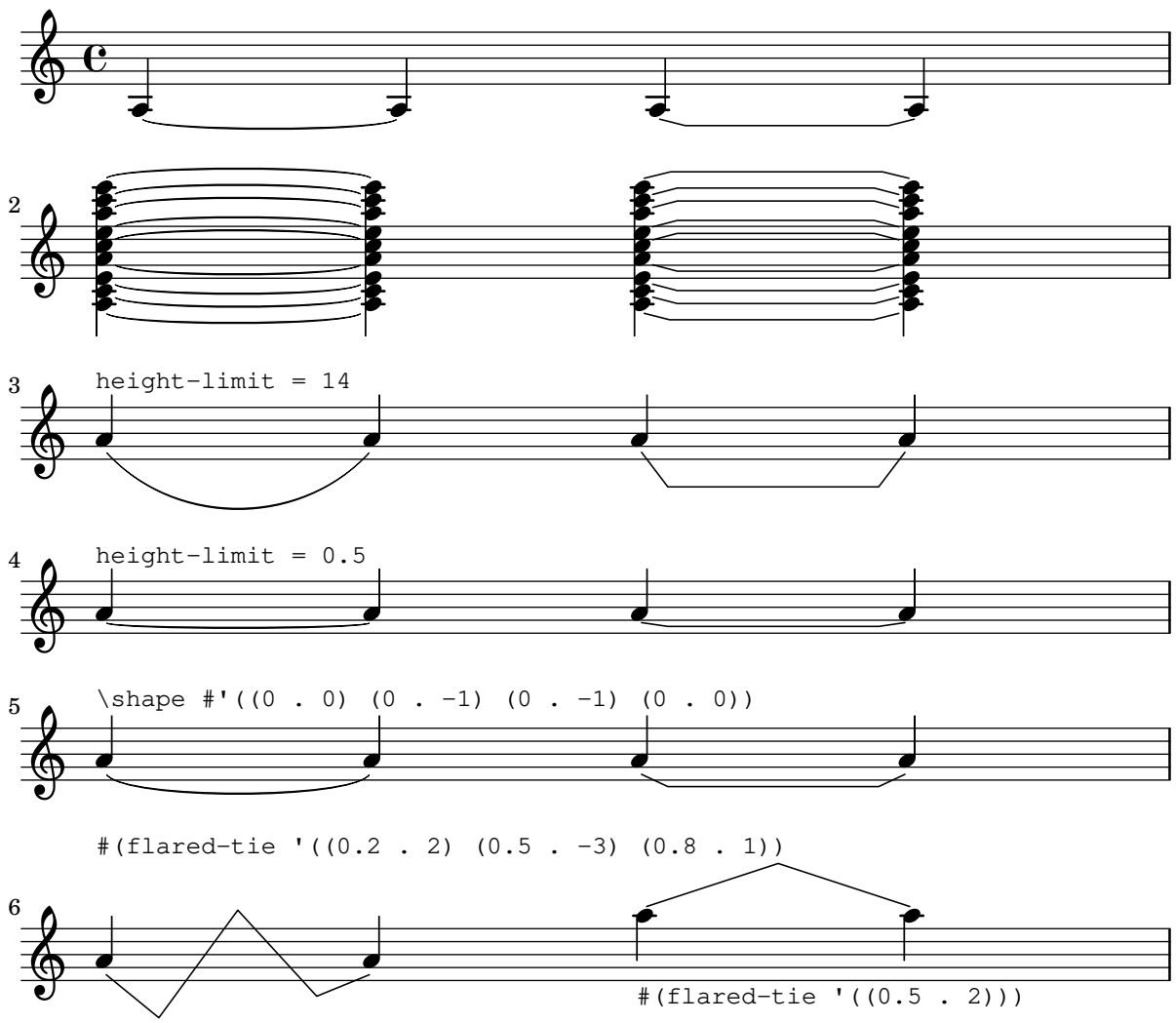
  <>^\markup \small \typewriter "height-limit = 14"
  \override Tie.details.height-limit = 14
  a'4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <>^\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
}
```



Adaptation de la largeur de mesure selon le `MetronomeMark`

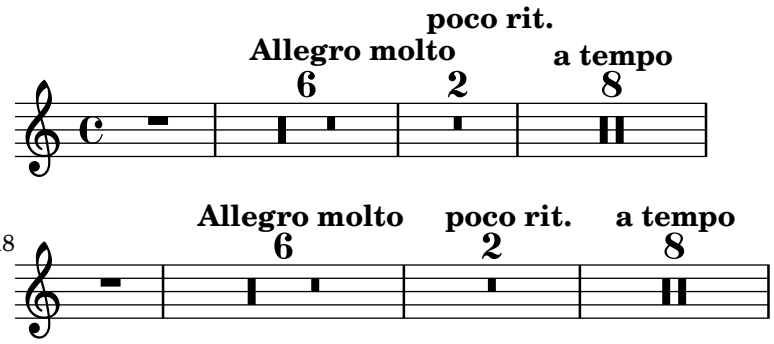
Par défaut, les indications métronomiques n'influencent en rien l'espacement horizontal.

Une simple dérogation, comme dans la deuxième partie de cet exemple, fournit la solution.

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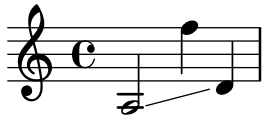
\layout {
  ragged-right = ##t
}
```



Glissando par dessus un objet graphique

Un glissando peut sauter un objet `NoteColumn`.

```
\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`.

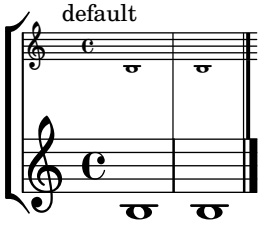
To fix this, multiple workarounds are available, as demonstrated below.

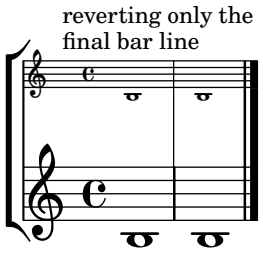
```
\markuplist {
  % First row.
  \fill-line {
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny "default"
          b1 b \bar "|."
        }
        \new Staff { b b }
      >>
    }
    \score {
      \new StaffGroup <<
        \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 }
      >>
    }
  }
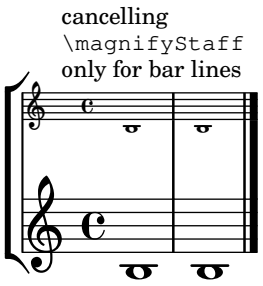
  \vspace #2
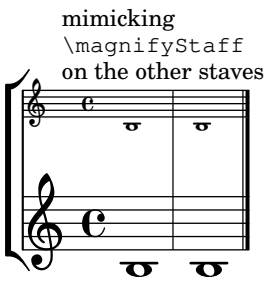
  % Second row.
  \fill-line {
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny \column { "mimicking"
                                            \typewriter "\\magnifyStaff"
```

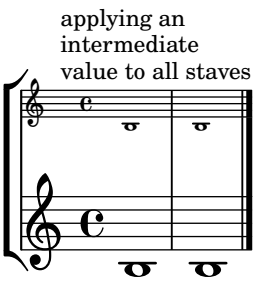
```
                                          "on the other staves" }
          b1 b \bar "|." }
        \new Staff \with { #(scale-props 'magnifyStaff 1/2 #t
                                         '((BarLine thick-thickness)
                                           (BarLine hair-thickness)
                                           (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 }
      >>
    }
    ""
  }
}
```











Incipit

Dans le cadre de la transcription de musique mensurale, un « incipit » en début de pièce permet d'indiquer tonalité et tempo originaux. Si les musiciens sont aujourd'hui habitués aux barres de mesure qui mettent en évidence la structure rythmique, elles n'étaient pas courantes à l'époque – le mètre variait souvent après quelques notes. Dans une forme de compromis, les barres de mesures sont régulièrement imprimées entre les portées.

```
% 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.
}
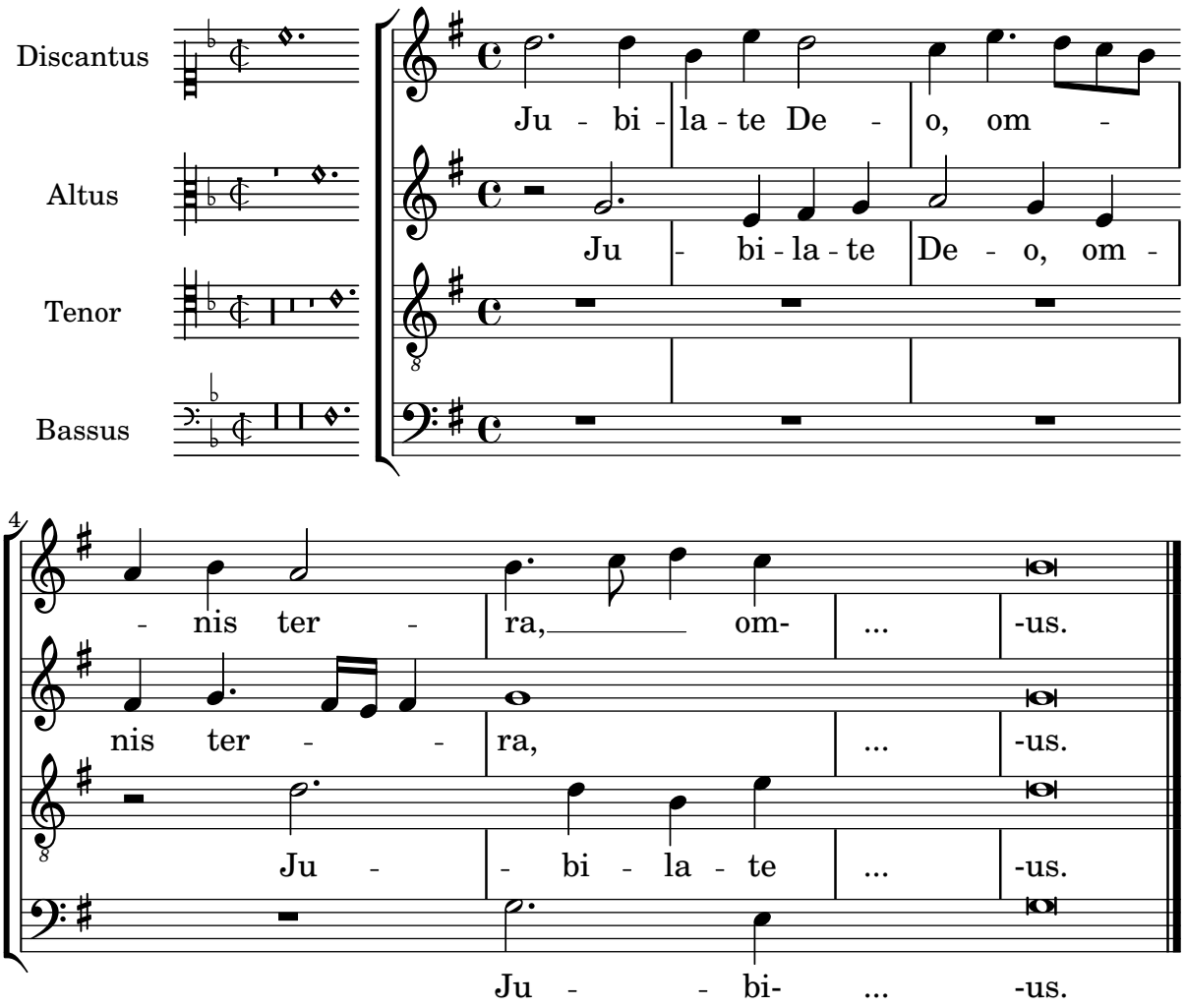
bassusIncipit = \new PetrucciStaff {
  % The original print shows the b flat
  % for the f major key signature twice.
  \override Staff.KeySignature.flat-positions = #'((-7 . 6))
  \clef "mensural-f"
  \key f\major
  \time 2/2
  \tweak Y-offset #1 r\longa \tweak Y-offset #1 r\longa
  f1.
}

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
  }
}
```



Insertion d'un fragment dans un *markup* au-dessus de la portée

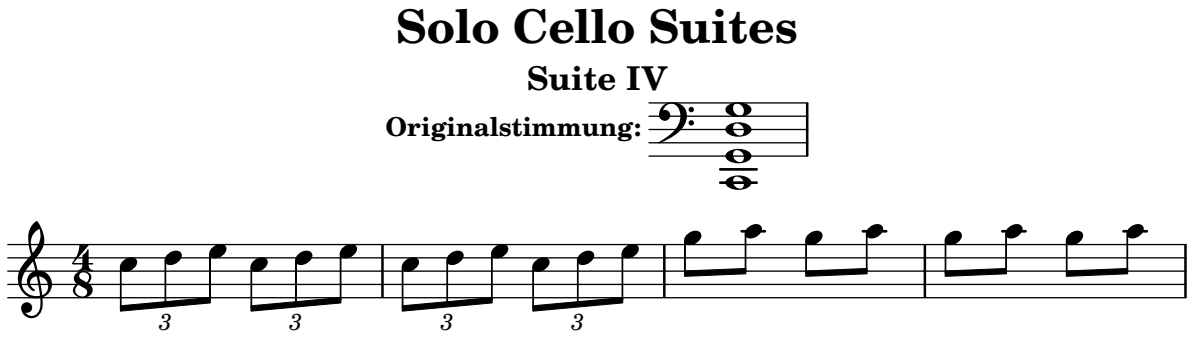
La commande \markup est polyvalente. Dans cet exemple, elle contient un bloc \score plutôt que du texte ou une marque.

```
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
}
```



Impression de la corde aiguë d'un `TabStaff` en bas

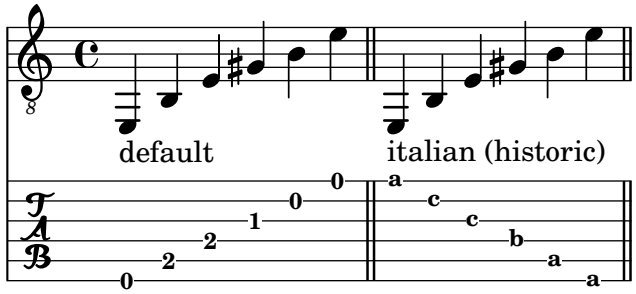
Il est d'usage, en matière de tablature, d'imprimer la première corde en haut. Il est toutefois possible de la positionner en bas grâce à une désactivation de la propriété de contexte `stringOneTopmost`. Définir ce réglage au sein d'un bloc `\layout` l'appliquera à l'intégralité du contexte.

```
%\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
  }
>>
```

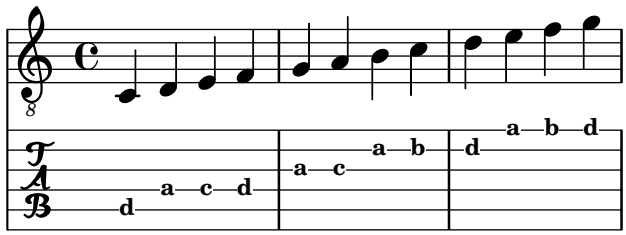


Tablature en lettres

Une tablature peut comporter des lettres en lieu et place des chiffres.

```
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
  }
>>
```



Saut de ligne et glissando

Normalement, LilyPond refuse de sauter automatiquement à la ligne à l'endroit où un glissando se prolonge sur la mesure qui suit. Ce comportement est modifiable par l'affectation de la valeur `#t` à la propriété `Glissando.breakable`. L'activation de la propriété `after-line-breaking` permet la continuation de l'indication de glissando après le saut de ligne.

La propriété breakable n'affecte pas les sauts manuels insérés par les commandes comme \break.

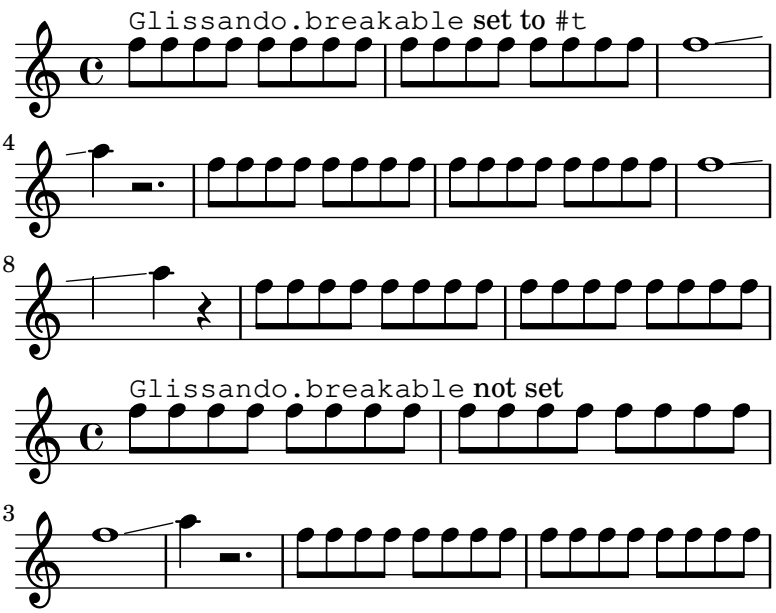
```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
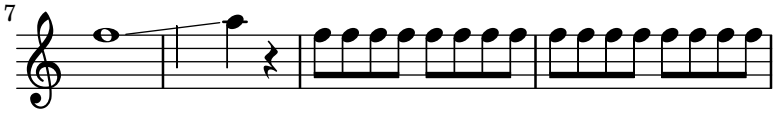
music = {
  \*16 f8 |
  f1\glissando |
  a4 r2. |
  \*16 f8 |
  f1\glissando \once\glissandoSkipOn |
  a2 a4 r4 |
  \*16 f8
}

\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
}
```
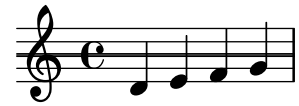




Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane, ou bien pour mettre en exergue la ligne portant la clef de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doit être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```
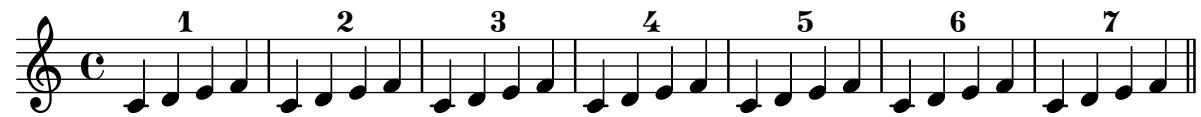


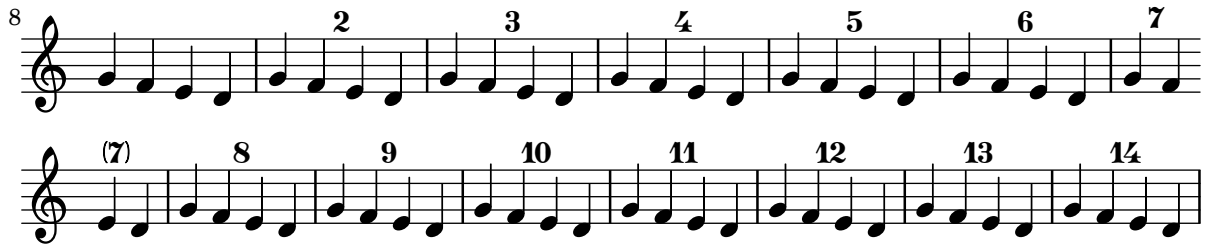
Compteur de mesures

Le code ci-dessous permet, de façon détournée, d'afficher un compteur de mesures à l'aide d'une répétition en pourcent rendue transparente.

```
\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
}
```





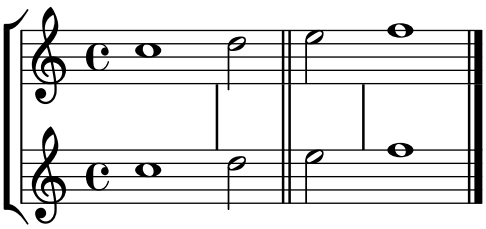
Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat il faudra définir `measureBarType` à `"-span|"` et utiliser un regroupement de portées permettant l'extension des barres entre les portées, tel un `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```

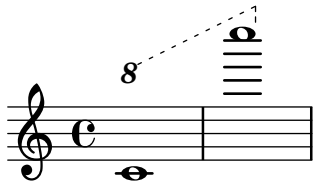


Modification de la pente de l'extension d'octaviation

Il est possible d'adapter la pente d'une indication d'octaviation.

```
\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
}
```

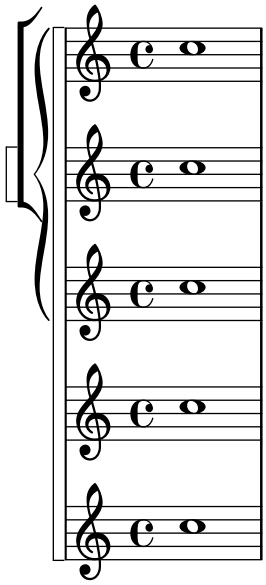


Imbrications de regroupements de portées

La propriété de contexte systemStartDelimiterHierarchy permet de créer des regroupements imbriqués complexes. Dans le cadre d'un StaffGroup, elle prend en argument la liste alphabétique des sous-groupes à hiérarchiser. Chaque sous-groupe peut être affublé d'un délimiteur particulier. Chacun des regroupements intermédiaires doit être borné par des parenthèses. Bien que des éléments de la liste puissent être omis, le premier délimiteur embrassera toujours l'intégralité des portées. Vous disposez des quatre délimiteurs SystemStartBar, SystemStartBracket, SystemStartBrace et 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 }
>>
```



Armures inhabituelles

La commande `\key` détermine la propriété `keyAlterations` d'un contexte `Staff`.

Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété. Il s'agit en l'occurrence de définir une liste :

```
\set Staff.keyAlterations =
  #`(((octave . pas) . altération) ((octave . pas) . altération) ...)
```
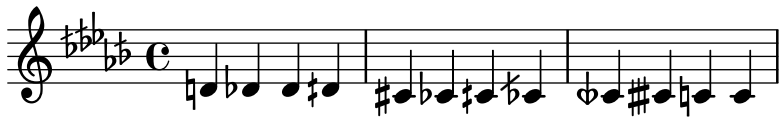
dans laquelle, et pour chaque élément, *octave* spécifie l'octave (0 pour celle allant du do médium au si supérieur), *pas* la note dans cette octave (0 pour do et 6 pour si), et *altération* sera `,SHARP` ou `,FLAT` ou `,DOUBLE-SHARP`, etc. (attention à la virgule en préfixe).

Une formulation abrégée – `(`*pas* `.` *altération*`)` – signifie que l'altération de l'élément en question sera valide quelle que soit l'octave.

En ce qui concerne les gammes microtonales dans lesquelles un « dièse » n'est pas d'un centième, *altération* se réfère à un deux-centième de ton entier.

```
\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 |
}
```



Orchestre, chœur et piano

Ce canevas illustre l'utilisation de contextes `StaffGroup` pour regrouper les instruments selon leur famille, imbriqués dans un `GrandStaff`, ainsi que le recours à la fonction `\transpose` pour les instruments transpositeurs. Dans tous les identificateurs, la musique est stockée en ut. Les notes peuvent tout aussi bien être saisies en ut ou dans la tonalité particulière de l'instrument avant d'être transposées puis affectées à une variable.

```
#(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 }
    >>

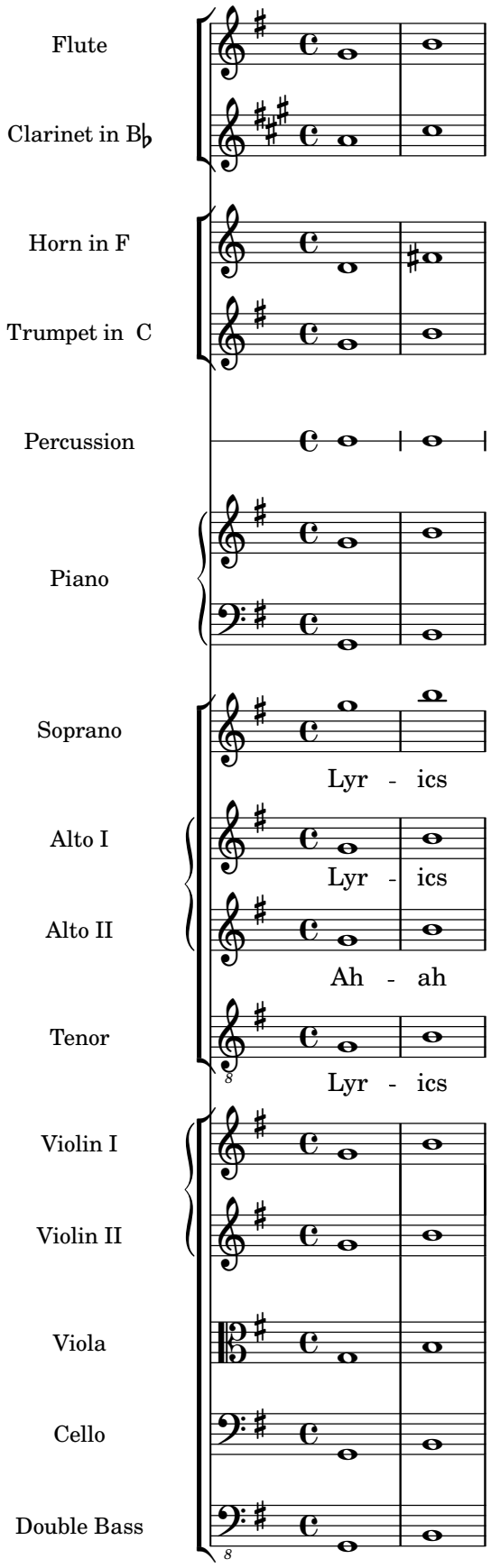
    \new StaffGroup = "StaffGroup_strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "Staff_violinI" \with {
          instrumentName = "Violin I"
        }
          \violinIMusic
        \new Staff = "Staff_violinII" \with {
          instrumentName = "Violin II"
        }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

Impression de noms d'accords identiques à la basse différente par une oblique et nouvelle basse

Voici comment imprimer une suite d'accords nommés (objets ChordNames) dont seule la basse diffère, indiquée par une oblique et cette nouvelle basse, à l'aide du graveur Scheme ici défini. Son comportement peut se gérer finement à l'aide de la propriété de contexte chordChanges.

```
#(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

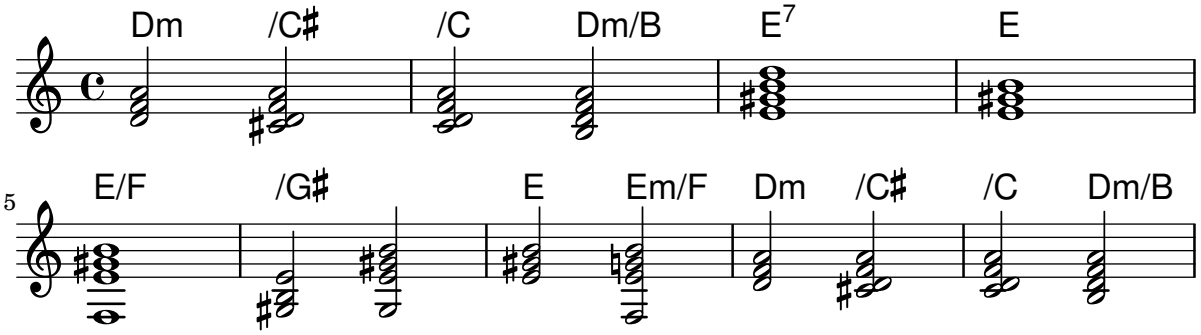
  d:m/c
  \set chordChanges = ##f
  d:m/b

  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
}
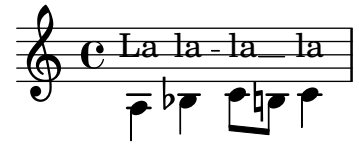
<<
  \new ChordNames
    \with { \consists #Bass_changes_equal_root_engraver }
    \myChords
  \new Staff \myChords
>>
```



Impression des paroles dans la portée

Des paroles peuvent venir s'inscrire dans la portée même. Ces paroles sont décalées par la dérogation `\override LyricText.extra-offset = #'(0 . dimension)` ; des commandes similaires s'occuperont des extenseurs et des tirets. Le décalage optimal ne peut s'obtenir qu'en procédant à tâtons.

```
<<
  \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 }
>>
```



Citation d'une autre voix

Les types d'événements pris en charge pour la citation sont déterminés par la propriété de contexte `quotedEventTypes`. Par défaut, sa valeur est fixée à `(note-event rest-event`

`tie-event beam-event tuplet-span-event`), ce qui signifie que seuls les notes, silences, liaisons, ligatures et n-olets seront mentionnés par `\quoteDuring`.

Dans l'exemple suivant, le quart de soupir n'est pas reproduit puisque `rest-event` n'est pas mentionné parmi les éléments de `quotedEventTypes`.

Pour connaître la liste des types d'événements, reportez-vous à la rubrique *Music classes* de la référence des propriétés internes.
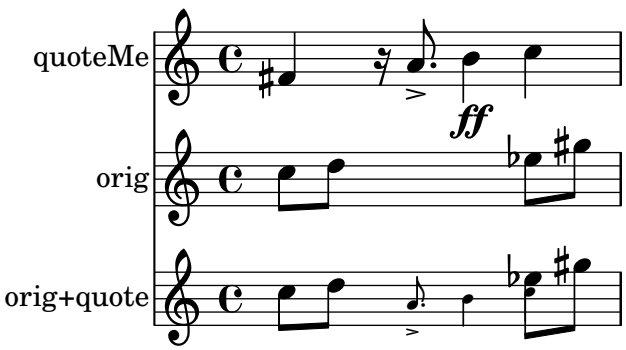
```
quoteMe = \relative c' {
  fis4 r16 a8.-> b4\ff c
}
\addQuote quoteMe \quoteMe

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. }
    }
  >>
>>
```



Citation d'une autre voix et transposition
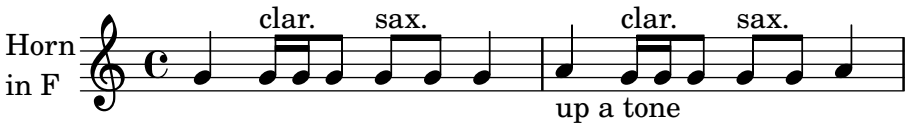
Les citations tiennent compte de la transposition, aussi bien celle de l'instrument d'origine que celle de la partie où elles interviennent. Dans l'exemple suivant, tous les instruments sont en tonalité de concert et seront repris par un instrument en fa. Le destinataire de la citation peut à son tour transposer à l'aide de la commande `\transpose`. En pareil cas, toutes les hauteurs seront transposées, y compris celles de la citation.

```
\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" >>
}
```



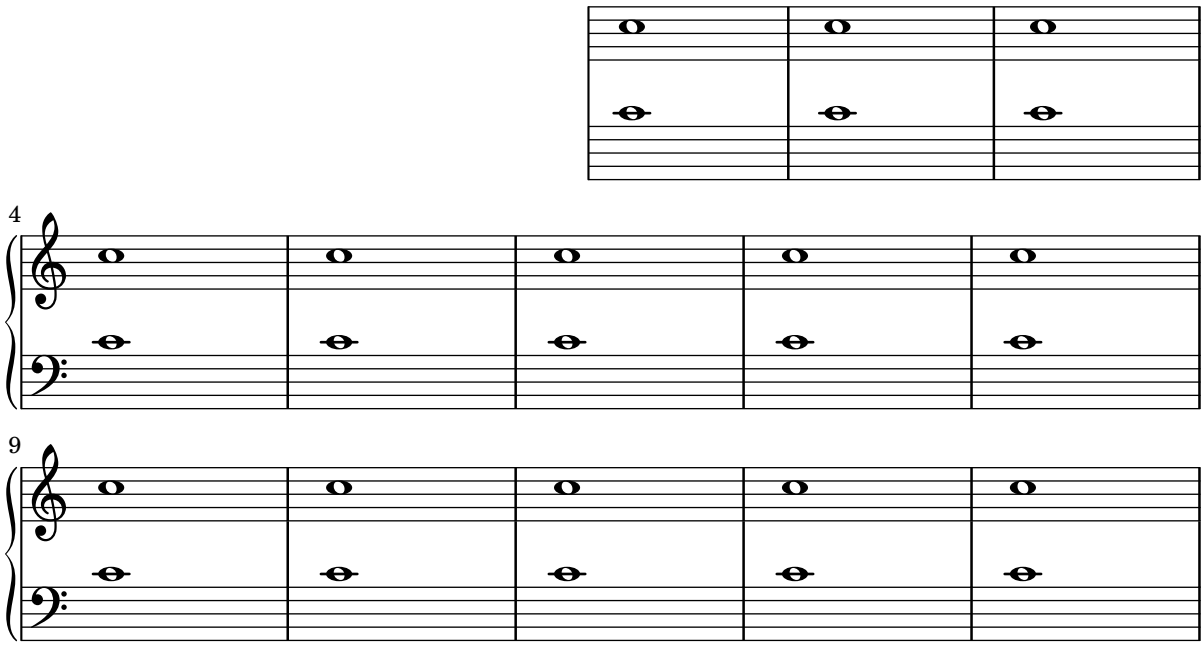
Retrait de l'accolade à la première ligne d'une pièce pour piano

Dans cet extrait est supprimée la première accolade d'un `PianoStaff` ou d'un `GrandStaff`, ainsi que les clefs. Ceci peut s'avérer utile pour couper et coller l'image générée dans de la musique préexistante.

Est utilisée la fonction `\alterBroken` afin de masquer l'accolade du début.

```
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
    }
  }
}
```

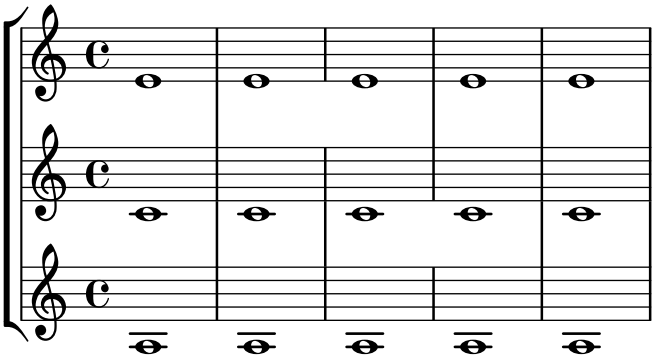


Suppression de la partie interportée des barres de mesure d'un regroupement autre que ChoirStaff

Les barres de mesure des regroupements StaffGroup, PianoStaff et GrandStaff sont par défaut d'un seul tenant. La portion entre les portées – l'objet SpanBar – peut néanmoins être supprimée, portée par portée.

```
\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
    }
  >>
}
```



Masquage de la première ligne si elle est vide

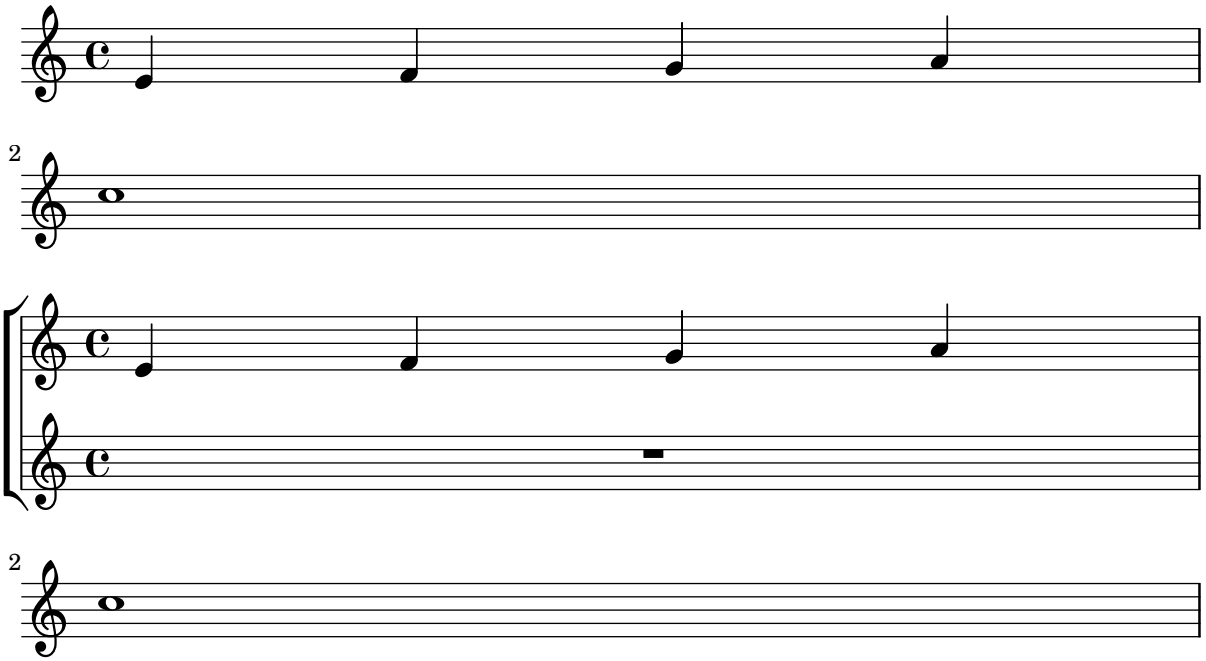
Par défaut, le premier système comportera absolument toutes les portées. Si vous préférez masquer les portées vides y compris pour le premier système, vous devrez activer la propriété `remove-first` de l'objet `VerticalAxisGroup`. Mentionnée dans un bloc `\layout`, cette commande agira de manière globale. Pour qu'elle ne soit effective que pour une portée particulière, vous devrez également spécifier le contexte (`Staff` pour qu'il ne concerne que la portée en cours) en préfixe de la propriété.

La première ligne inférieure du deuxième `StaffGroup` est bien présente, pour la simple raison que le réglage en question ne s'applique qu'à la portée dans laquelle il a été inscrit.

```
\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
  }
>>
```



Séparation visuelle entre les systèmes

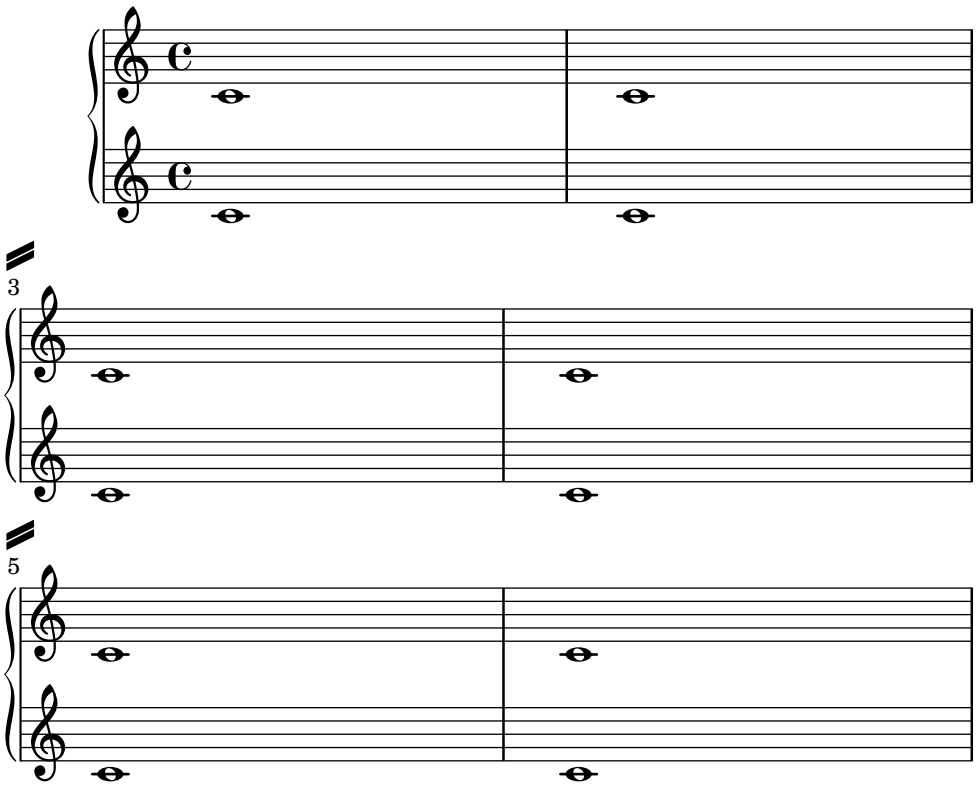
La séparation entre deux systèmes consécutifs peut être mise en évidence par n'importe quel *markup*. LilyPond dispose à cet effet d'une double oblique inversée : `\slashSeparator`.

```
#(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
    >>
  }
}
```



Galbe individuel des liaisons d'un accord

Le galbe des liaisons de prolongation d'un accord peuvent adopter des galbes différents comme illustré ci-dessous.

Dans la première ligne, on voit que toutes les notes de l'accord bénéficient d'une liaison.

Modifier ces liaisons à l'aide de `\shape` est sans effet dans la mesure où `TieColumn` les positionne de son propre chef, ignorant plus ou moins ce que contient `\shape`. On pourrait s'en sortir en affectant `#t` à `positioning-done`, mais `positioning-done` est une propriété interne et l'activer revient à dire : tout le positionnement est terminé, il n'y a plus rien à faire. La deuxième ligne illustre le fait que le positionnement n'est pas terminé : toutes les liaisons sont par dessous, et leur épaisseur n'est pas optimale.

Le fait d'indiquer explicitement le positionnement des liaisons permet d'utiliser `\shape` pour chacune des liaisons (4e ligne) et fonctionne même dans le cas d'un saut de ligne et avec la propriété `tieWaitForNote`.

```
{
  \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
}

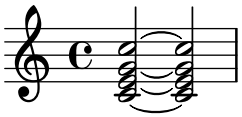
{
  \textMark \markup {
    It also works with the \typewriter tieWaitForNote property. }
  \set tieWaitForNote = ##t
  c'4-\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
  e'-\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
  g'-\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
  c''-\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
  \once \override TieColumn.positioning-done = ##t
```

```
  <c' e' g' c''>1
}

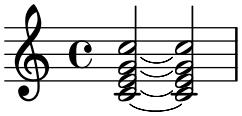
\layout {
  indent = 0
  \context {
    \Score
    \override TextMark.padding = #4
    \override TextMark.break-align-symbols = #'(left-edge)
  }
}

\paper {
  score-system-spacing.padding = 3
}
```

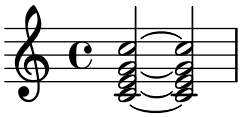
Chords can be tied note by note.



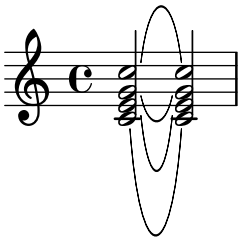
Modifying those ties with `\shape` does not succeed, because `TieColumn` positions them on its own behalf, ignoring `\shape` input more or less. You may circumvent this by setting `positioning-done` to `#t` – alas, `positioning-done` is an internal property, and setting it to `#t` means: all positioning 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.



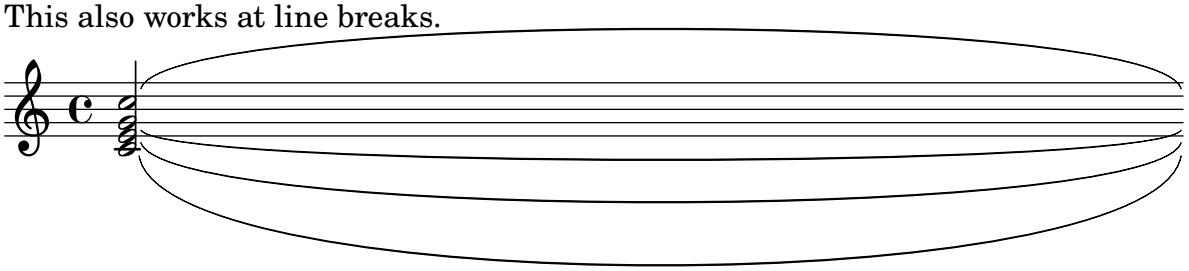
To fix that, enter ties with explicit direction modifiers.

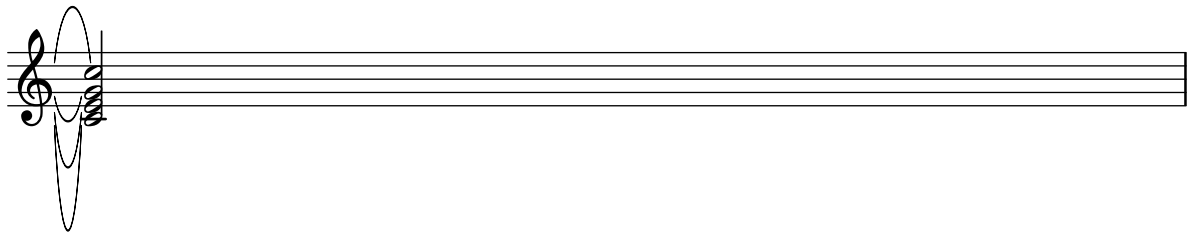


Now you can use `\shape` for each tie as usual.

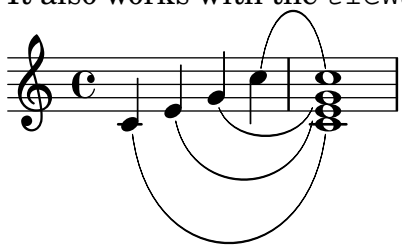


This also works at line breaks.
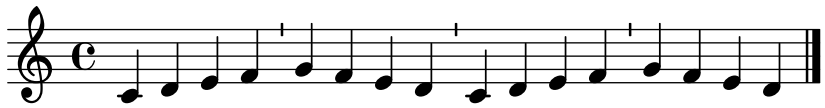




It also works with the `tieWaitForNote` property.



Barre de mesure en encoche

Lorsque les barres de mesure ne sont là que dans un but de coordination et non pour accentuer le rythme, il arrive souvent qu'elles se présentent sous la forme d'une simple encoche.
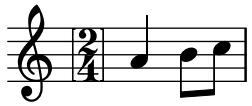
```
\relative c' {
  \set Score.measureBarType = #"'"
  c4 d e f
  g4 f e d
  c4 d e f
  g4 f e d
  \bar "|."
}
```



Métrique entre crochets

La métrique peut être mise entre crochets.
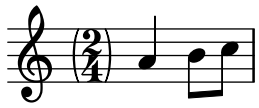
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Métrique entre parenthèses

Une métrique peut être mise entre parenthèses.

```
\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
}
```



Affinage des propriétés d'une clef

Modifier le glyphe, la position de la clef ou son octaviation ne changeront pas la position des notes ; il faut pour y parvenir modifier aussi la position du do médium. La redéfinition préalable de `middleCClefPosition` permet de placer l'armure sur les bonnes lignes. Le positionnement est relatif à la ligne médiane, un nombre positif faisant monter, un nombre négatif abaissant.

Par exemple, la commande `\clef "treble_8"` équivaut à définir les propriétés de contexte `clefGlyph`, `clefPosition` – qui contrôle la position verticale de la clef –, `middleCPosition` et `clefOctavation`. Une nouvelle clef apparaîtra dès lors que l'une de ces propriétés, à l'exception de `middleCPosition`, aura été modifiée.

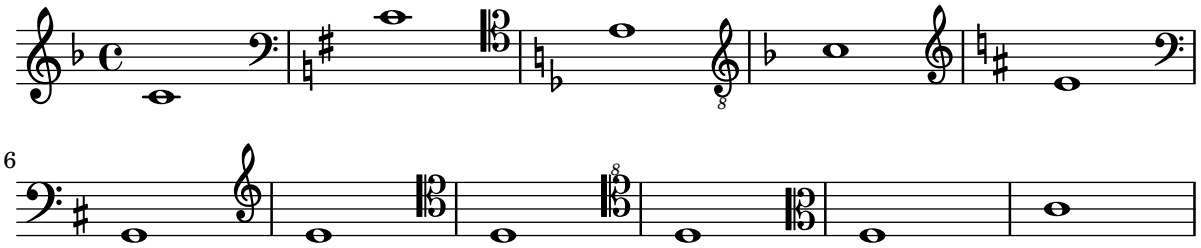
Les exemples qui suivent illustrent les différentes possibilités de définir ces propriétés manuellement. Sur la première ligne, la position relative des notes par rapport aux clefs est préservée, ce qui n'est pas le cas pour la deuxième ligne.

```
{
  % 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
}
```



Deux \partCombine sur une même portée

La fonction \partCombine prend deux expressions musicales contenant chacune une partie et les distribue dans quatre contextes Voice nommés « two », « one », « solo » et « chords » selon le moment et la manière dont ces parties fusionnent en une voix commune.

Des variantes de \partCombine – \partCombineUp et \partCombineDown – permettent de fusionner des voix aux hampes ascendantes et descendantes. Les combiner alors pour présenter quatre parties sur une seule portée requiert cependant des réglages spécifiques, comme indiqué ci-dessous.

```
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.

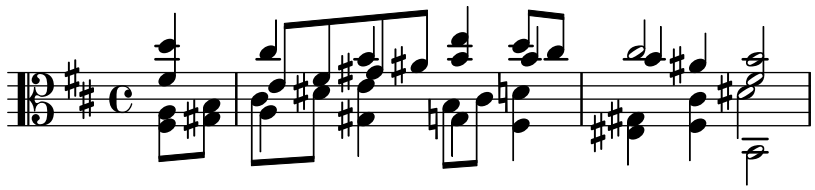
The context is called 'Down'; internally, the function calls
`\\partCombineDown`."
  #{
    \new VoiceBox = "Down" <<
      \set VoiceBox.soloText = #"Solo III"
      \set VoiceBox.soloIIText = #"Solo IV"
      \context Voice ="one" { \voiceFour }
      \context Voice ="two" { \voiceTwo }
      \context 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"
```

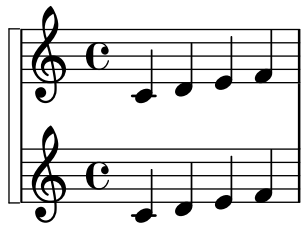
```
  }
}
```



Indication de regroupement de portées par un rectangle

Un regroupement de portées sera indiqué par un simple rectangle – `SystemStartSquare` – en début de ligne dès lors que vous le mentionnerez explicitement au sein d'un contexte `StaffGroup` ou `ChoirStaff`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



Utilisation d'autoChange avec plus d'une voix

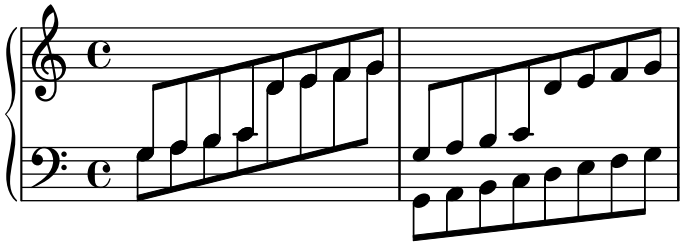
L'instruction \autoChange est opérationnelle y compris en présence de voix multiples.

```
\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
    }
  >>
}
```



Utilisation de lignes de repères dans une partition « à la française »

Le recours à des contextes MarkLine (tel qu'indiqué dans Inscrire les repères ailleurs que sur la première portée (https://wiki.lilypond.community/wiki/Placing_rehearsal_marks_other_than_above_the_top_staff)) dans une partition « à la française » peut s'avérer problématique lorsque toutes les portées entre deux contextes Markline sont absentes d'un système. La présence du Keep_alive_together_engraver au sein de chacun des StaffGroup permet de maintenir actif le MarkLine uniquement lorsque les autres portées du regroupement sont elles-mêmes actives.

```
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
```

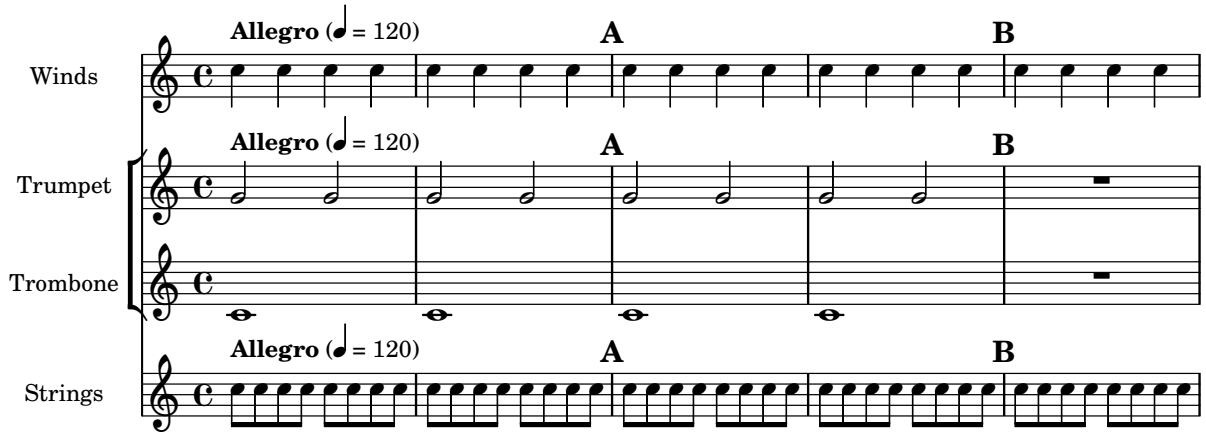
```
  R1*17
}
strings = \*240 c''8
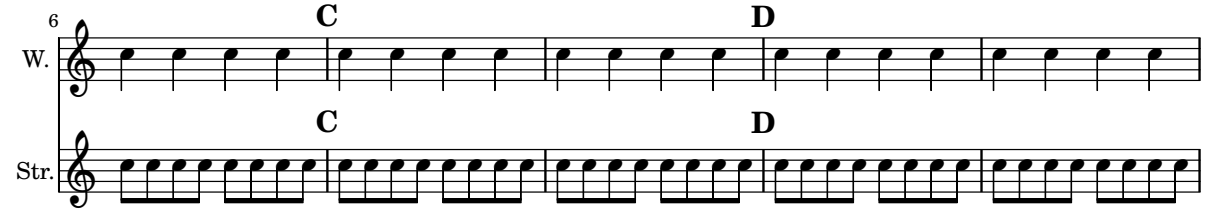
#(set-global-staff-size 16)
\paper {
  systems-per-page = 5
  ragged-last-bottom = ##f
  tagline = ##f
}
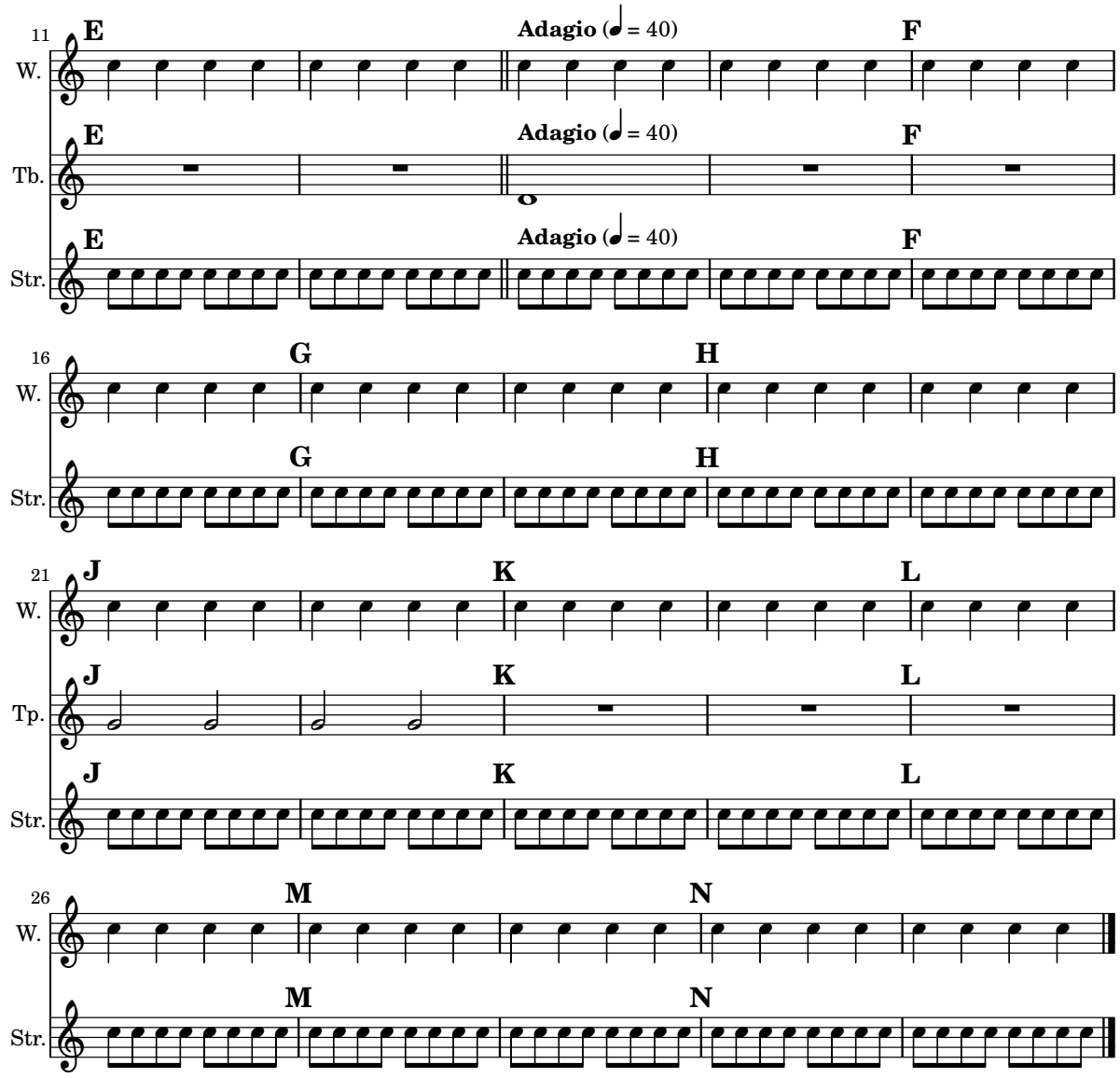
\layout {
  indent = 16\mm
  short-indent = 5\mm
  \context {
    \name MarkLine
    \type Engraver_group
    \consists Apply_output_engraver
    \consists Axis_group_engraver
    \consists Mark_engraver
    \consists Metronome_mark_engraver
    \consists Staff_collecting_engraver
    \override VerticalAxisGroup.remove-empty = ##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 }
    >>
  >>
}
```







Alignement vertical de `StaffGroups` sans `SystemStartBar`

Cet extrait indique comment obtenir des regroupements `StaffGroup` alignés verticalement, disposant chacun de leur indication de groupe `SystemStartBar`, sans qu'il ne soient connectés entre eux.

Il est à noter que ceci ne fonctionne correctement que lorsque la musique tient sur un seul système.

```
#(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
  >>
>>
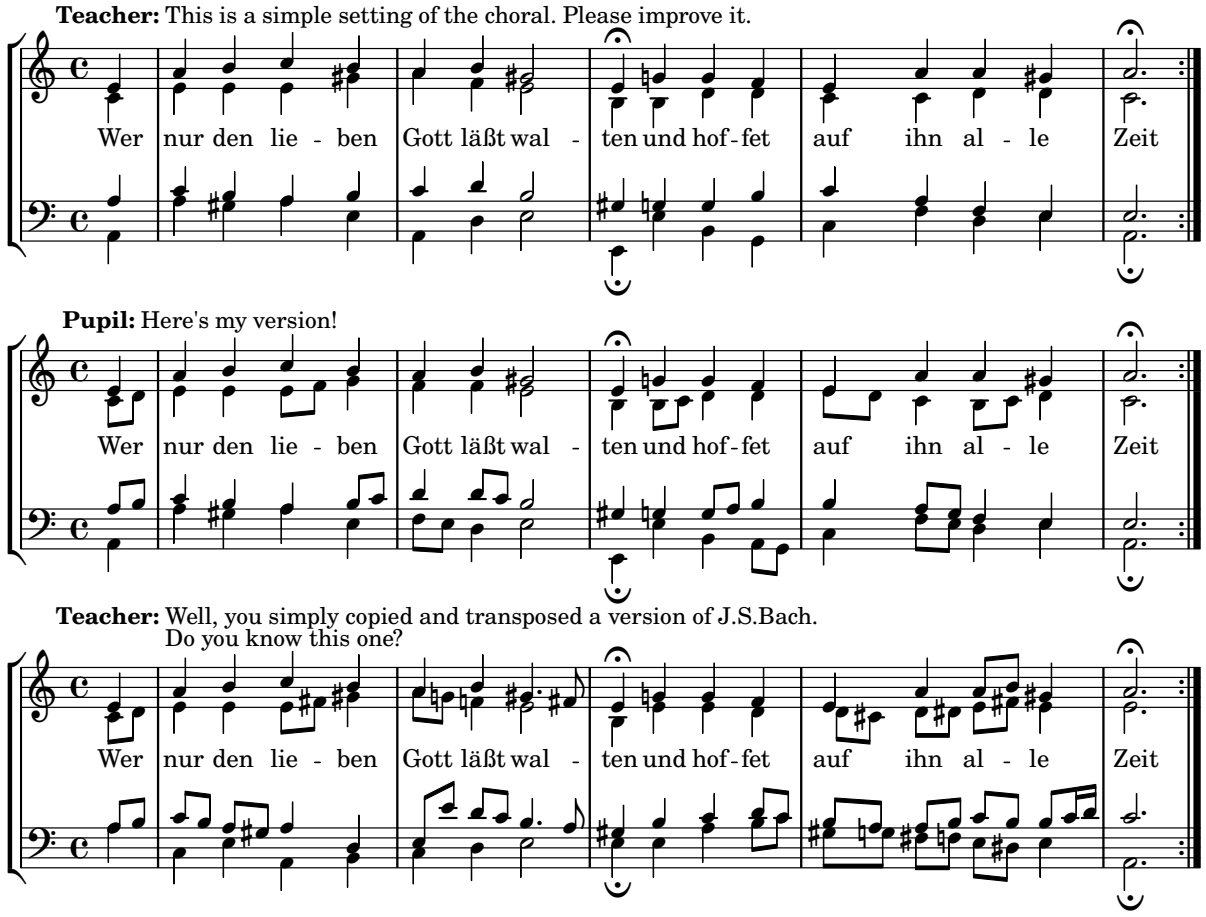

% Score
```

```
\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
    }
  }
}
```

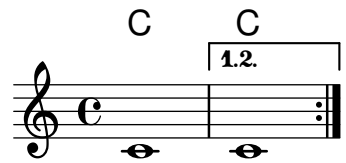
Exercise: Improve the given choral



Crochet de reprise sous les chiffrages d'accord

L'ajout du `Volta_engraver` à la bonne portée permet d'imprimer les crochets de reprise entre les chiffrages et la portée.

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

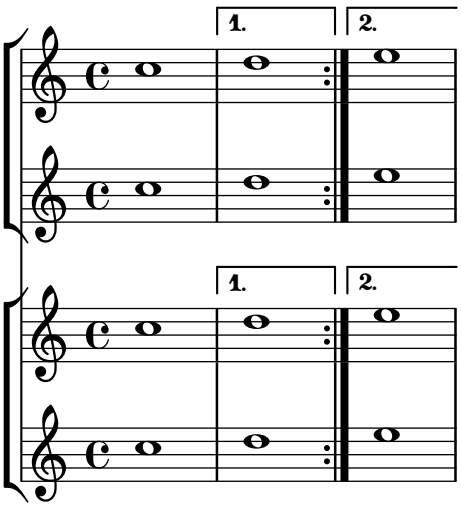


Crochets de *volta* sur plusieurs portées

L'ajout du `Volta_engraver` à une portée particulière permet que les crochets de *volta* s'affichent sur d'autres portées que la première d'une partition.

```
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
  >>
>>
```
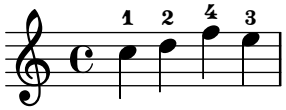


7 Editorial annotations

See also Section “Editorial annotations” dans *Manuel de notation*.

Ajout de doigtés à une partition

Les instructions de doigtés se saisissent selon une syntaxe très simple.

```
\relative c'' {
  c4-1 d-2 f-4 e-3
}
```



Ajout de liens à des objets

La fonction `add-link`, telle que définie ci-dessous, permet d’ajouter un lien au stencil d’un objet graphique. Elle s’emploie au sein d’un `\override` ou d’un `\tweak`.

À noter que le fonctionnemment du `point-and-click` est perturbé sur les objets ainsi liés.

Cette fonction n’est opérationnelle que pour une sortie PDF.

La coloration des objets liés s’obtient par une commande séparée.

```
#(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
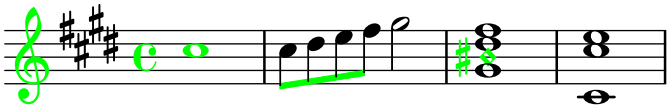
   \once \override Staff.Clef.color = #green
   \once \override Staff.Clef.after-line-breaking =
     #(add-link urlI)

   \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>
}
```



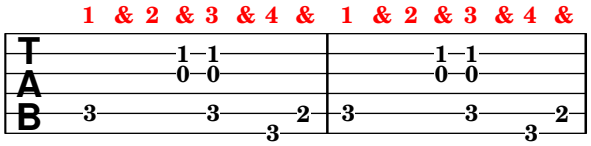
Ajout de *markups* à une tablature

Par défaut, les *markups* n'apparaissent pas dans une tablature.

Il suffit, pour les voir apparaître, de rétablir la propriété `stencil` de l'objet `TextScript` dans le contexte `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^"&" }
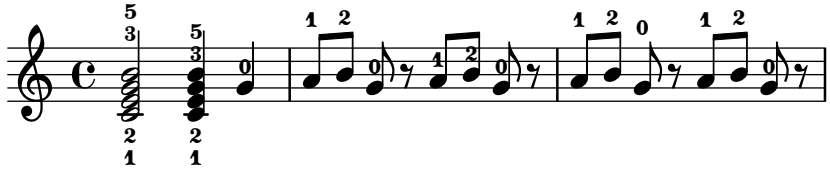
\score {
 \new TabStaff {
   \*2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```



Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Il est néanmoins possible d'annuler ce comportement. Une attention particulière doit toutefois être portée dans les cas où doigté et hampe vont dans la même direction : les indications de doigté n'évitent les hampe qu'en présence de ligature. Ce réglage peut s'adapter pour éviter toutes les hampes ou aucune d'elles. L'exemple suivant illustre ces deux options, ainsi que la manière de revenir au comportement par défaut.

```
\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
}
```



Numérotation des mesures et alternatives

Un réglage de la propriété de contexte `alternativeNumberingStyle` permet d'accéder à deux méthodes alternatives de gestion de la numérotation des mesures en cas de reprises.

```
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 "|."
}
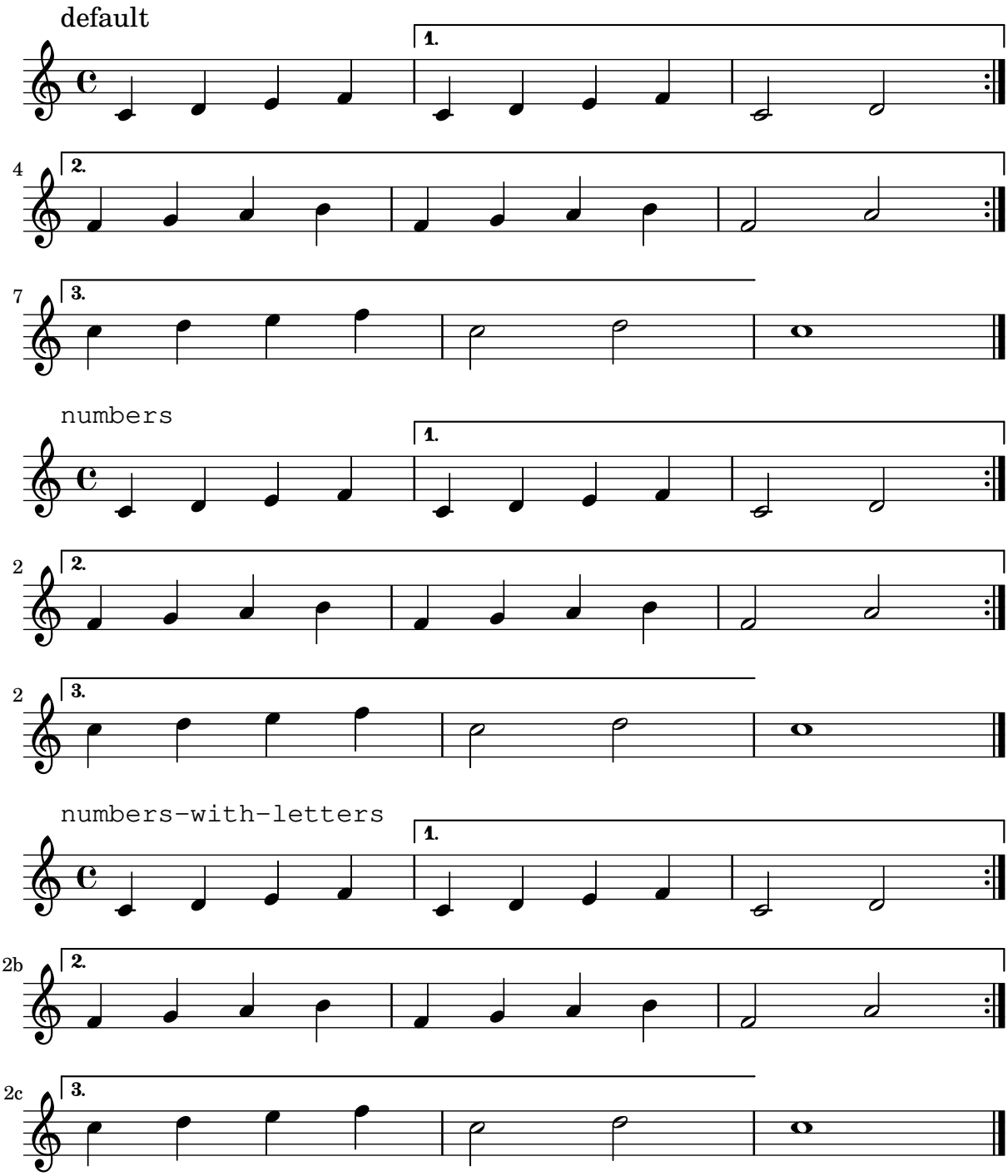
{
  \textMark \markup \large "default"
  \music
}

{
  \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
  }
}
```



Crochets d'analyse au-dessus de la portée

Les crochets d'analyse viennent par défaut se positionner au-dessous de la portée. L'exemple suivant vous indique comment les faire apparaître en surplomb de la portée.

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

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```



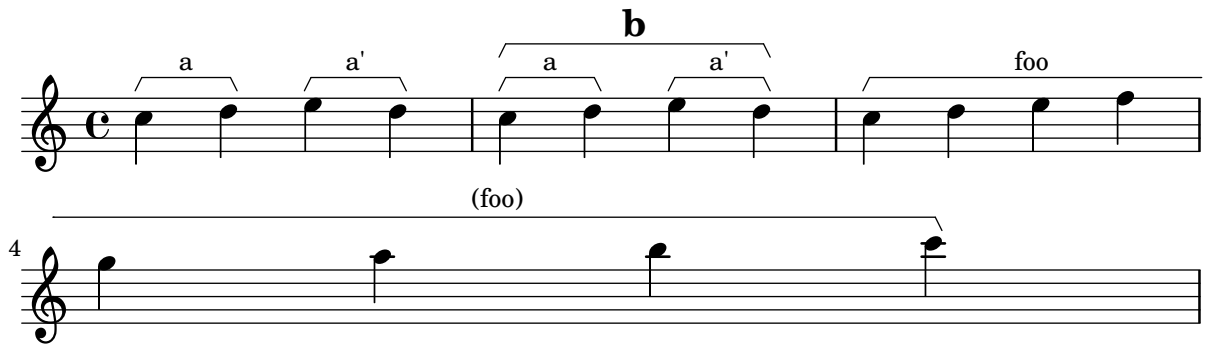
Crochet d'analyse avec texte

Un *markup* textuel peut venir s'ajouter aux crochets d'analyse grâce à la propriété `text` de l'objet graphique `HorizontalBracketText`. Plusieurs crochets présents en un même moment requièrent d'utiliser la commande `\tweak`.

Le texte ajouté sera répété, entre parenthèse, après un saut de ligne. L'ordre vertical des crochets imbriqués se gère à l'aide de la propriété `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
}
```



Profilage des notes selon leur degré dans la gamme

La propriété `shapeNoteStyles` permet d'affecter un profil particulier à chaque degré de la gamme – à partir de l'armure ou de la propriété `tonic`. Ses valeurs sont constituées d'une liste de symboles, qu'il s'agisse de formes géométriques (`triangle`, `cross` ou `xcircle`) ou basés sur la tradition des graveurs américains (avec quelques noms de note latins).

LilyPond dispose de deux raccourcis, `\aikenHeads` et `\sacredHarpHeads`, permettant de reproduire d'anciens recueils de chansons américaines.

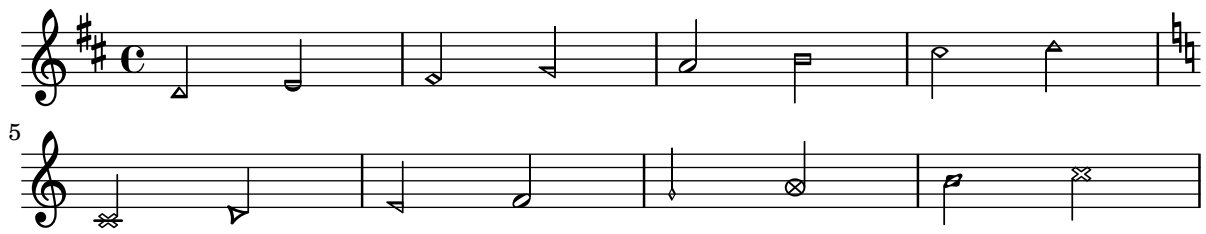
L'exemple suivant montre plusieurs manières de profiler les têtes de note, ainsi que la capacité de transposer tout en respectant la fonction harmonique de chaque note dans la gamme.

```
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
  }
}
```

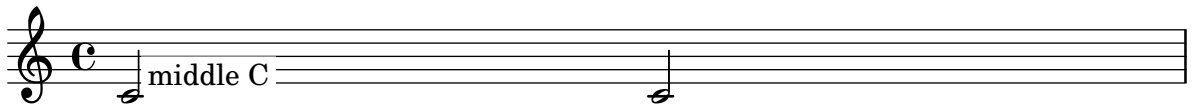


Blanchiment de lignes de portée avec la commande `\whiteout`

L'instruction `\whiteout` intercale un rectangle blanc contenant un *markup*. Dans la mesure où les lignes de la portée sont à un niveau inférieur à la plupart des autres objets graphiques, ce cache ne devrait pas masquer d'autres éléments.

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

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```

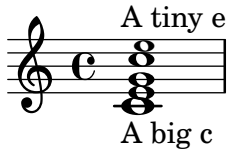


Modification de la taille d'une note particulière d'un accord

La commande `\tweak` permet de modifier la propriété `font-size` d'une note particulière d'un accord.

Il s'agit de placer, à l'intérieur même de l'accord (dans la construction `< >`) et avant la note considérée, l'instruction `\tweak` suivie de `font-size` et de définir la taille voulue – comme `#-2` pour une petite tête.

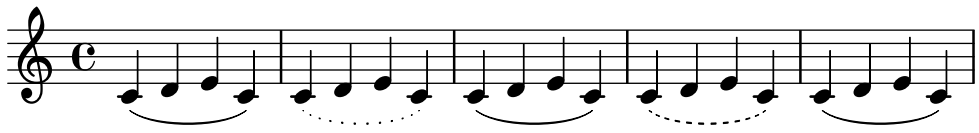
```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Modification de l'aspect des liaisons d'articulation

Une liaison d'articulation peut se présenter sous la forme d'un trait continu ou discontinu, voire en pointillé.

```
\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)
}
```



Coloration des notes selon leur hauteur

Les têtes de note peuvent adopter une couleur différente selon leur hauteur ou leur nom ; la fonction utilisée ici fait même la distinction entre enharmoniques.
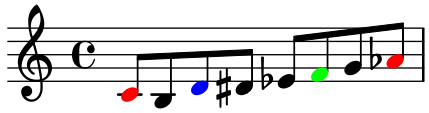
```
% Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (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
  }
}
```

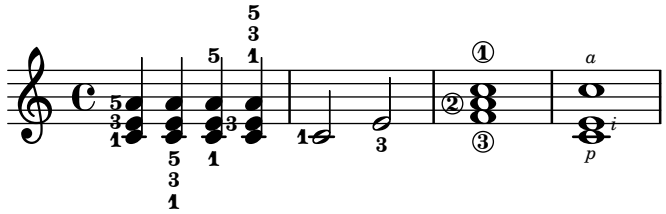


Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise à l'aide de la propriété `fingeringOrientation`. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `<...>`, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue à l'aide des propriétés `stringNumberOrientation` et `strokeFingerOrientation`.

Ces propriétés peuvent prendre une liste de trois valeurs. Elles contrôlent si les informations doivent être placées au-dessus (en présence d'un `up`) ou au-dessous (en présence d'un `down`), à gauche (en présence d'un `left` ou à droite (en présence d'un `right`). Par contre, si un positionnement n'est pas mentionné, aucun doigté n'y figurera. LilyPond tient compte de ces contraintes et trouvera le meilleur emplacement pour le doigté des notes de l'accord qui suit. Bien entendu, `left` et `right` sont exclusifs l'un de l'autre – les doigtés ne peuvent se placer que d'un seul côté.

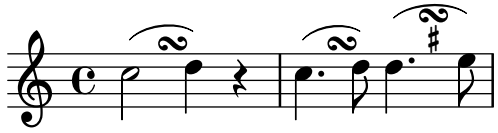
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \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 >
}
```



Création d'un grupetto retardé

Obtenir un *gruppetto* retardé et dans lequel la note la plus basse est altérée requiert quelques surcharges. La propriété `outside-staff-priority` doit être désactivée (`#f`) pour éviter qu'elle prenne le pas sur la propriété `avoid-slur`. L'ajustement du premier argument de `\after` (il s'agit d'une durée) permet de jouer sur le positionnement 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)
}
```



Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le Bar_number_engraver du contexte Score, et les Time_signature_engraver, Clef_engraver et Bar_engraver du contexte Staff.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

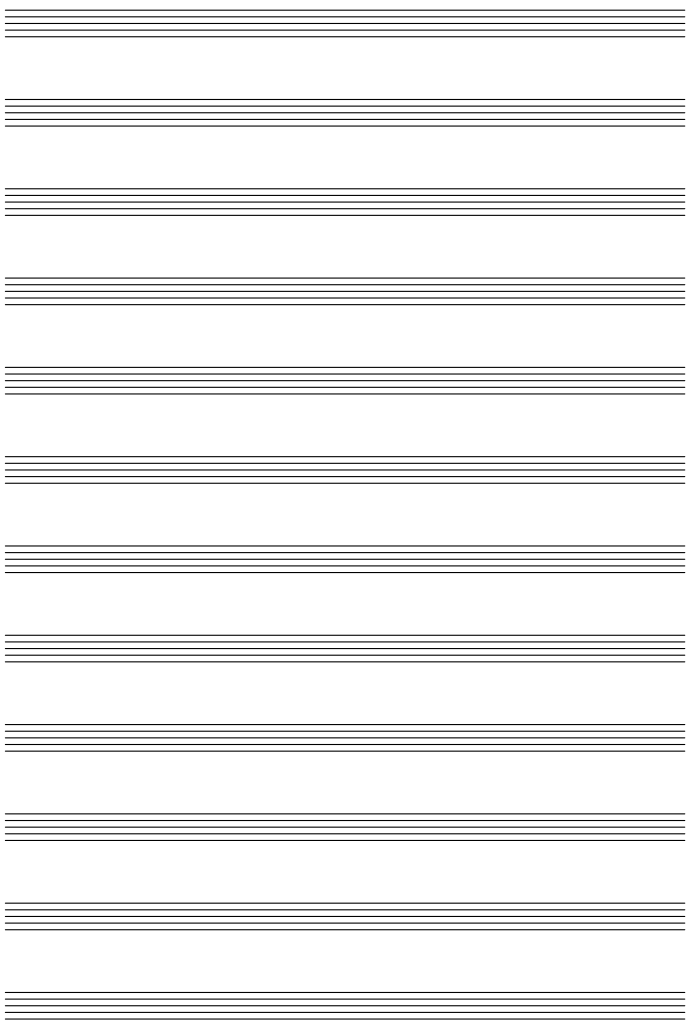
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % 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
   }
  %}

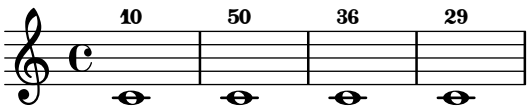
  % uncomment these lines for "A4" size
  %{
  \paper {
    #(set-paper-size "a4")
    ragged-last-bottom = ##f
    line-width = 180\mm
    left-margin = 15\mm
    bottom-margin = 10\mm
    top-margin = 10\mm
    tagline = ##f
  }
  %}
}
```



Création de doigtés sur deux caractères

Il est tout à fait possible de noter un doigté supérieur à 5.
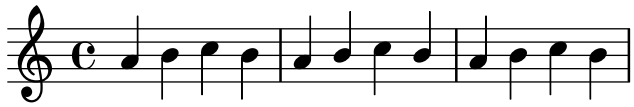
```
\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}
```



Direction par défaut des hampes de la ligne médiane

La direction des hampes des notes placées sur la ligne médiane de la portée est gérée par la propriété `neutral-direction` de l'objet `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
}
```



Taille de police différente pour instrumentName et shortInstrumentName

Une simple dérogation à un contexte particulier permet de différencier la taille des variables *instrumentName* et *shortInstrumentName*.

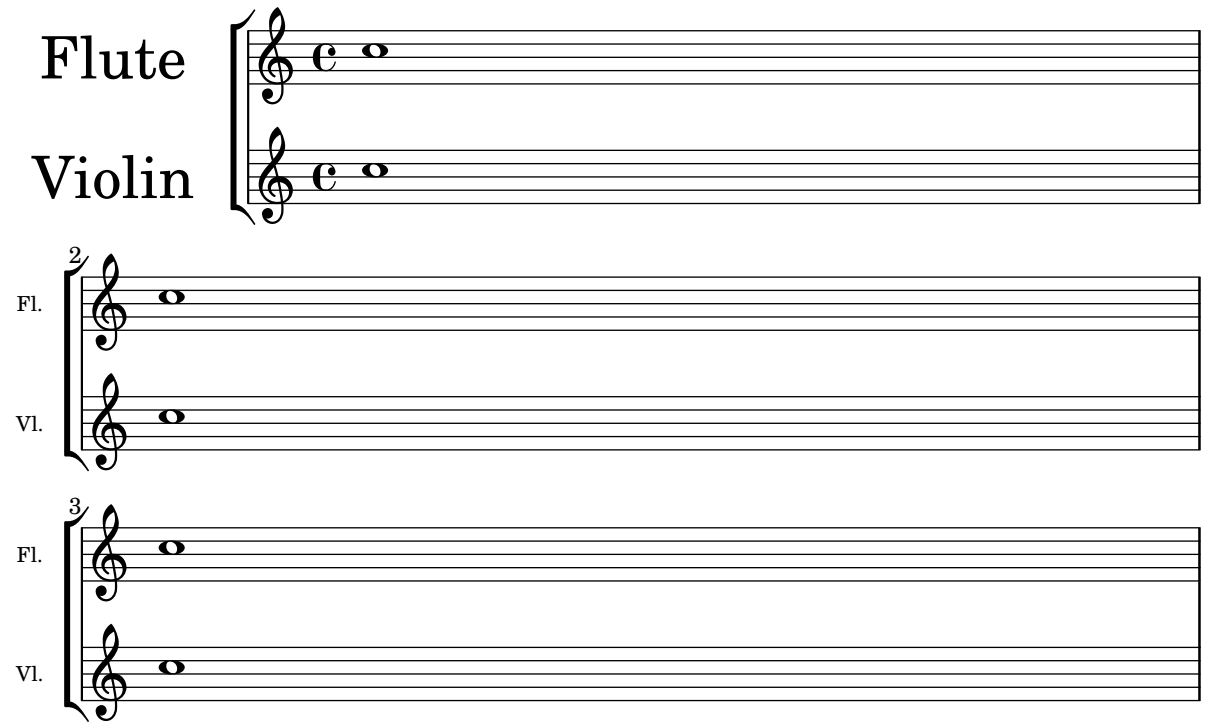
```
InstrumentNameFontSize =
#(define-music-function (font-size-pair) (pair?)
   "Set the font size of `InstrumentName` grobs.

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'' }
>>
```



Encadrement d'objets

La propriété `stencil` peut se modifier pour obtenir l'encadrement de n'importe quel objet, que ce soit à l'aide d'un `\override` ou d'un `\tweak`.

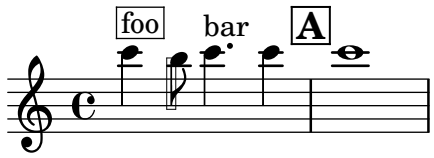
```
\relative c'' {
  \once \override TextScript.stencil =
    #(make-stencil-boxer 0.1 0.3 ly:text-interface::print)
  c'4^"foo"

  \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
```
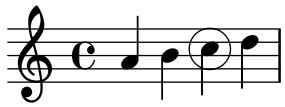
```
}
```



Encerclement de notes

Un cercle peut venir entourer une note à l'aide d'une fonction Scheme dédiée qui, temporairement, viendra en modifier la propriété `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'' }
```

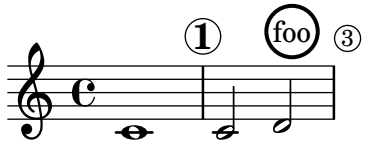


Encercler divers objets

La commande `\circle` permet de dessiner un cercle autour d'objets `\markup`. D'autres objets nécessitent de faire appel à des techniques spécifiques. Cet exemple illustre deux manières de procéder : pour les repères, et pour les numéros de mesure.

```
\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)
}
```



Intégration de PostScript dans un *markup*

Du code PostScript peut directement être intégré dans un bloc `\markup`.

En règle générale, il est préférable d'utiliser la commande de *markup* `\polygon` qui est fonctionnelle avec tous les moteurs de rendu 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
}
```



Génération de notes profilées spéciales

Lorsque le profil spécifique d'une tête de note ne peut être obtenu facilement à l'aide d'un *markup* graphique, il est possible d'utiliser des instructions de dessin pour `ly:make-stencil` afin de générer ce profil. Cet exemple propose une tête de note profilée en parallélogramme.

Malheureusement, les commandes disponibles pour spécifier le dessin ne sont pas documentées à ce jour – c'est l'objet du ticket #6874 (`https://gitlab.com/lilypond/lilypond/-/issues/6874`) – mais la sous-commande `path` a la signature suivante, très proche de la fonction Scheme `make-path-stencil`.
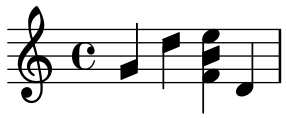
```
(path épaisseur liste-commandes style-ligne-principale style-jonction-ligne remplissage
```

Les commandes dans *liste-commandes* ressemblent aux instructions de dessin en PostScript, à ceci près que les arguments suivent ici le nom de la commande.

```
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
}
```
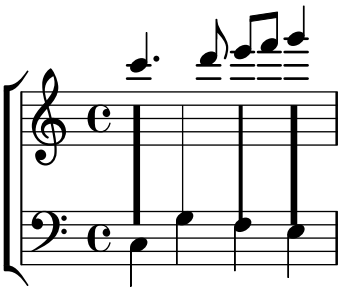


Apparence du quadrillage temporel

Modifier certaines des propriétés du quadrillage temporel aura pour effet d'en changer l'apparence.

```
\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
  }
}
```



Quadrillage temporel : mise en évidence du rythme et synchronisation des notes

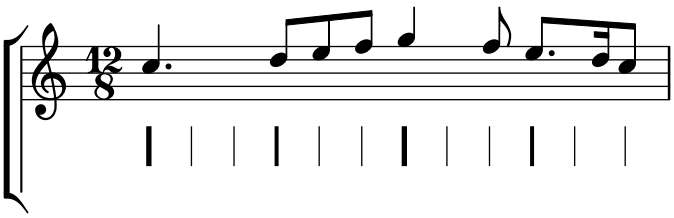
Des lignes verticales entre les portées peuvent s'ajouter dans le but d'indiquer la synchronisation entre des notes. Dans le cas de musique monophonique, on peut toutefois créer une deuxième portée, invisible, et raccourcir les lignes comme ici.

```
\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
```

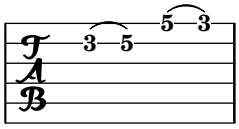
```
    \override NoteColumn.X-offset = #-0.5
  }
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #1/8
    % set line length and positioning:
    % two staff spaces above center line on hidden staff
    % to four spaces below center line on visible staff
    \override GridPoint.Y-extent = #'(2 . -4)
  }
}
```



Hammer on et pull off

Hammer-on et *pull-off* peuvent s'indiquer par des liaisons.

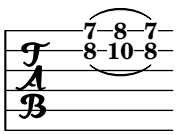
```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Hammer on et pull off avec accords

Dans le cadre de notes en accord, les *hammer-on* et *pull-off* sont indiqués par un arc simple. Vous obtiendrez néanmoins un arc double en réglant la propriété `doubleSlurs` sur `#t`.

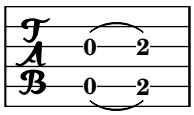
```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



Hammer on et pull off gérés par les voix

L'arc des *hammer-on* et *pull-off* est ascendant dans les voix une et trois, et descendant dans les voix deux et quatre.
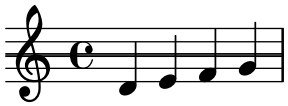
```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane, ou bien pour mettre en exergue la ligne portant la clef de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doit être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```
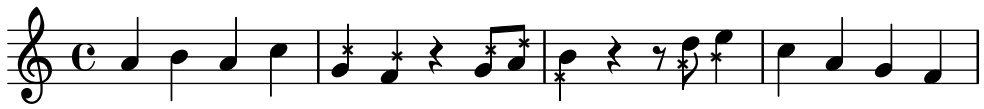


Adjonction d'une croix sur la hampe des notes d'un fragment parlé

Voici comment ajouter une croix aux hampes. Le début du fragment parlé est stipulé par une commande `\speakOn`, et la fin par une commande `\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
  }
}
```

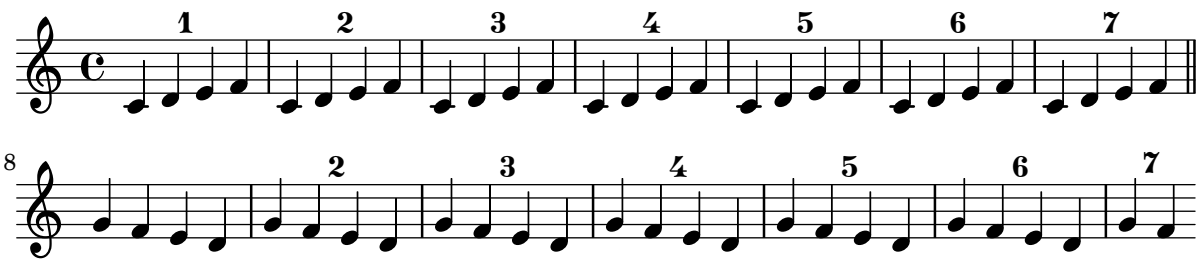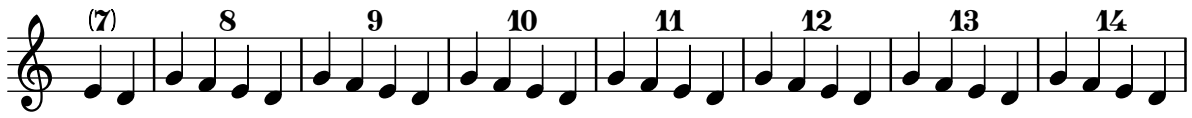


Compteur de mesures

Le code ci-dessous permet, de façon détournée, d'afficher un compteur de mesures à l'aide d'une répétition en pourcent rendue transparente.

```
\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
}
```



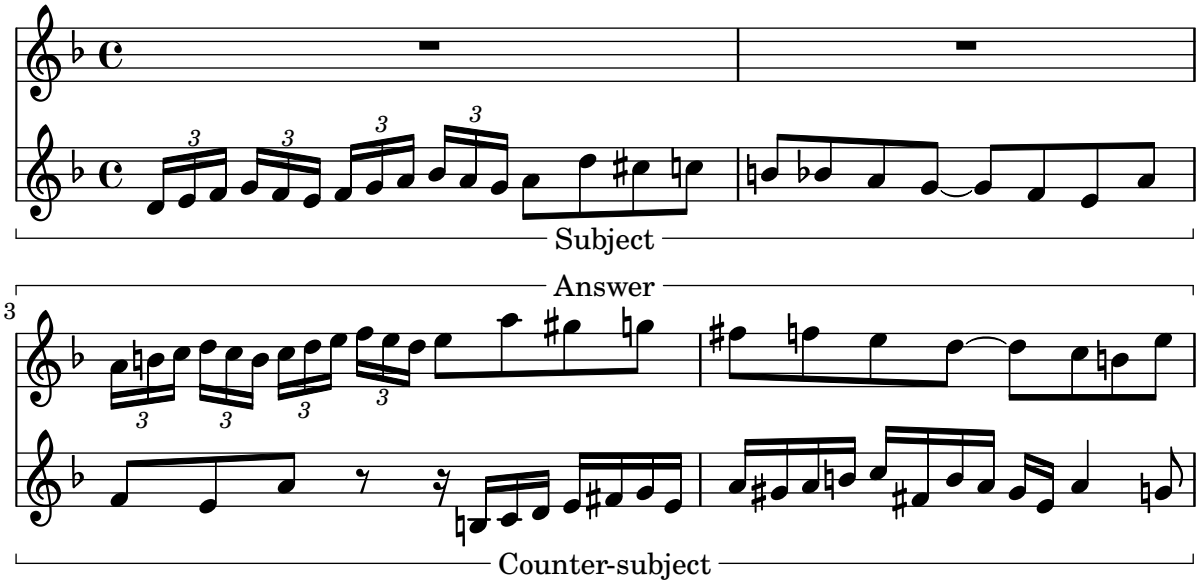


Crochet de mesure

Les bandeaux sur mesure sont un moyen alternatif d'imprimer un crochet annoté. Contrairement aux crochets horizontaux, ces bandeaux s'étendent entre deux barres de mesure plutôt qu'entre deux notes. Le texte est centré sur l'empan du crochet.

```
\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
  }
>>
```



Positionnement précis des indications de doigté

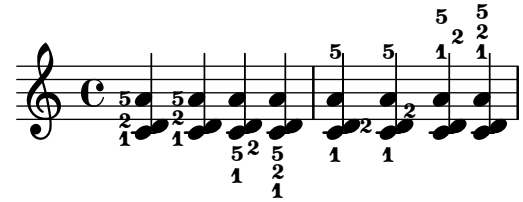
Les options par défaut en matière de positionnement des doigtés d'un accord donnent généralement de bons résultats. Il se peut néanmoins qu'un ajustement soit nécessaire dans certains cas particuliers, notamment en présence d'un intervalle de seconde. L'astuce ici présentée permet d'obtenir un meilleur rendu.

```
\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>
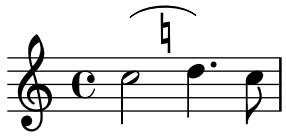
    \set fingeringOrientations = #'(up)
    <c-1 d-\tweak extra-offset #'(0 . 1.1)-2
     a'-\tweak extra-offset #'(0 . 1)-5>
    <c-1 d-\tweak extra-offset #'(-1.2 . 1.5)-2
     a'-\tweak extra-offset #'(0 . 1.4)-5> |
  }
}
```



Positionnement d'une annotation à l'intérieur d'une liaison

Lorsqu'une annotation doit s'incrire à l'intérieur d'une liaison, la propriété `outside-staff-priority` doit être désactivée.

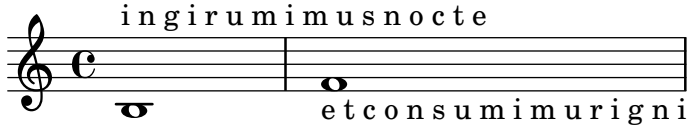
```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```



Impression de texte de droite à gauche

Du texte, inclus dans un objet *markup*, peut s'imprimer de droite à gauche, comme illustré ci-dessous.

```
{
  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 }
  }
}
```
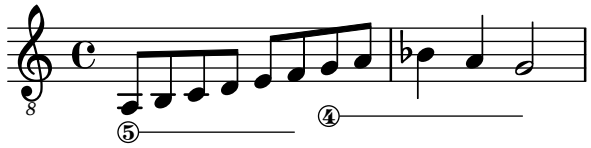


Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
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
```
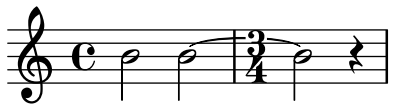
```
}
```



Utilisation de la propriété `whiteout`

Tout objet graphique peut s'imprimer sur un fond blanc afin de masquer une partie des objets qu'il recouvre. Ceci trouve toute son utilité pour améliorer certaines collisions, notamment dans des situations où un repositionnement d'objets est irréaliste. Il faut alors explicitement définir la propriété `layer` afin de contrôler quels objets seront masqués par le fond blanc.

Dans l'exemple ci-dessous, la liaison est en collision avec la métrique ; la situation est améliorée dès lors que la portion de liaison qui traverse la métrique est masquée par l'affectation de la propriété `whiteout` à l'objet `TimeSignature`. Pour ce faire, `TimeSignature` est déplacé sur un calque au-dessus de celui de `Tie` – il reste sur le calque par défaut (1) –, puis le `StaffSymbol` est placé sur un calque supérieur à celui de `TimeSignature` pour ne pas être masqué.

```
{
  \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 Section “Text” dans *Manuel de notation*.

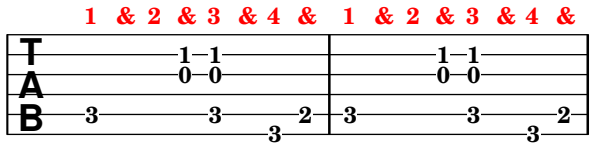
Ajout de *markups* à une tablature

Par défaut, les *markups* n’apparaissent pas dans une tablature.

Il suffit, pour les voir apparaître, de rétablir la propriété `stencil` de l’objet `TextScript` dans le contexte `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
    }
  }
}
```



Ajout de la date du jour à une partition

Avec un peu de code Scheme, voici comment ajouter facilement la date du jour à votre partition.

```
\paper { tagline = ##f }
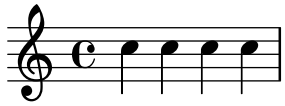
% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))

% 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
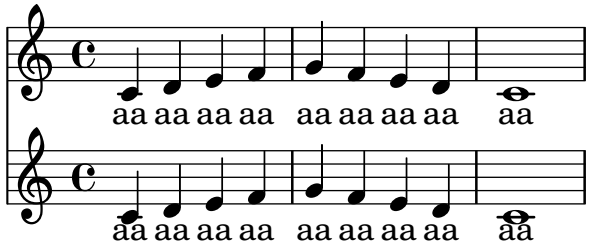
Ajustement de l'espacement vertical des paroles

Le code ci-dessous indique comment rapprocher de la portée à laquelle elle est affectée, une ligne de paroles.

```
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
>>
```



Alignement des noms d'instrument

L'alignement horizontal des noms d'instrument se gère à l'aide de la propriété `self-alignment-X` de l'objet `InstrumentName` (habitullement attaché à un contexte `Staff`). Les variables `indent` et `short-indent`, attachées au bloc `\layout`, déterminent l'espace alloué à l'alignement des noms d'instrument, respectivement dans leurs formes développée et abrégée.

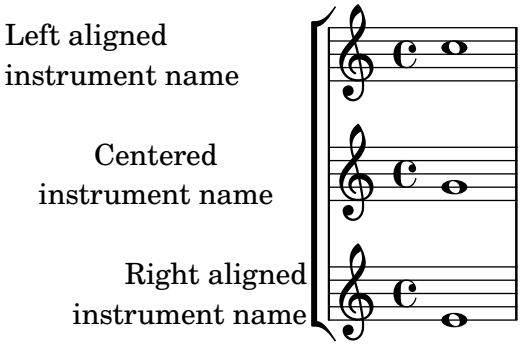
```
\paper {
  left-margin = 3\cm
}

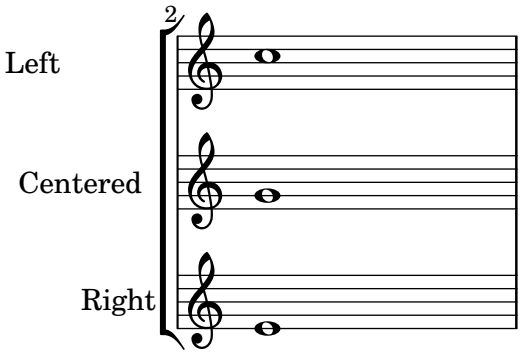
\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                            "instrument name" }
    shortInstrumentName = "Left"
  } {
    c''1 \break c''1
  }

  \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
}
```
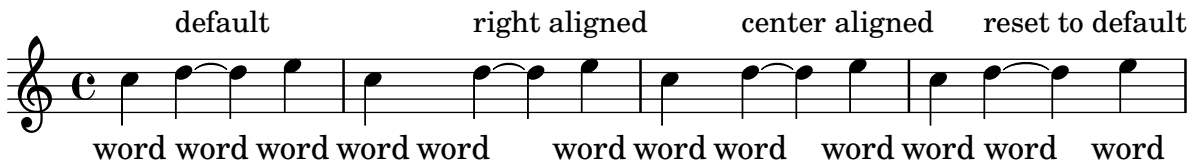




Alignement de la première syllabe d'un mélisme

Par défaut, une syllabe qui entame un mélisme est alignée sur la note par la gauche. Cet alignement peut se modifier à l'aide de la propriété `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
}
```

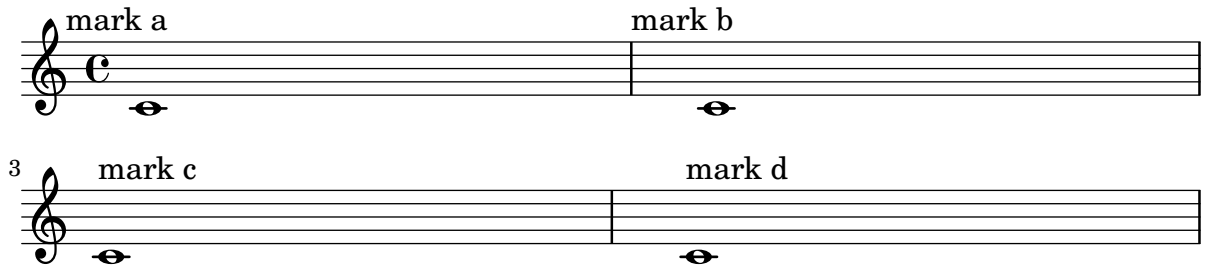


Alignement de repère sur une note

Les objets `TextMark` s'alignent par défaut sur des objets graphiques `NonMusicalPaperColumn` tels que le coin gauche de la portée ou une barre de mesure. Ils peuvent aussi venir s'aligner dès lors qu'aura été désactivée la propriété `non-musical`.

```
\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 |
}
```



Blanchiment de lignes de portée avec la commande `\whiteout`

L'instruction `\whiteout` intercale un rectangle blanc contenant un *markup*. Dans la mesure où les lignes de la portée sont à un niveau inférieur à la plupart des autres objets graphiques, ce cache ne devrait pas masquer d'autres éléments.

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

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```



Texte centré sous un soufflet

La fonction comprise dans l'extrait suivant permet d'ajouter du texte – comme « molto » ou « poco » – à un soufflet de (de)crescendo. Le texte ajouté viendra se positionner selon le positionnement du soufflet. Le soufflet s'aligne avec un objet `DynamicText`.

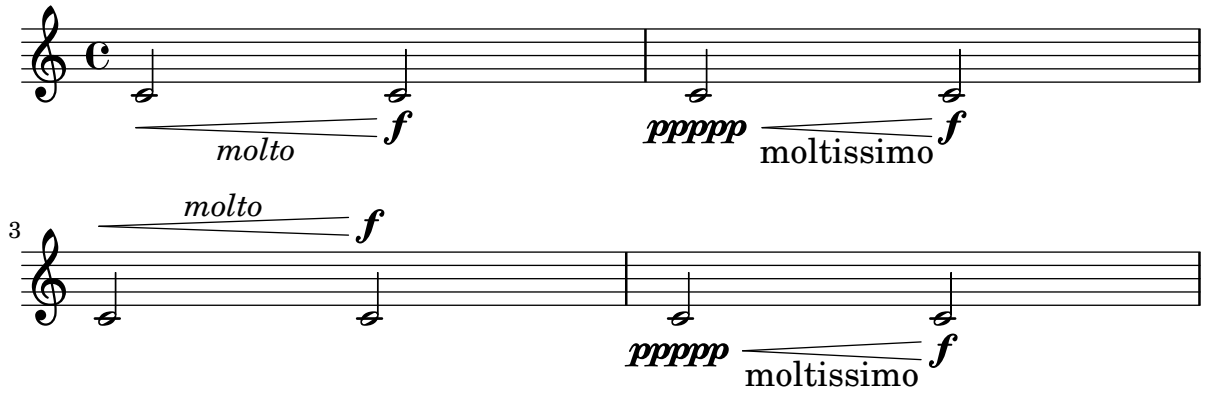
Cet exemple présente aussi comment, à l'aide de code Scheme, influencer la manière dont un objet est normalement imprimé.

```
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
}
```

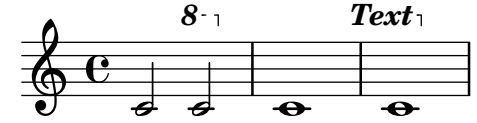


Modification du texte des marques d'octaviation

En interne, la fonction `\ottava` détermine les propriétés `ottavation` (par ex. en `"8va"` ou `"8vb"`) et `middleCPosition`. Vous pouvez modifier le texte d'une marque d'octaviation en définissant `ottavation` après avoir fait appel à `ottava`.

Un texte bref est particulièrement utile lorsque l'octaviation est courte.

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



Changement des fontes textuelles par défaut

Les familles de fontes par défaut pour le texte peuvent être changées.

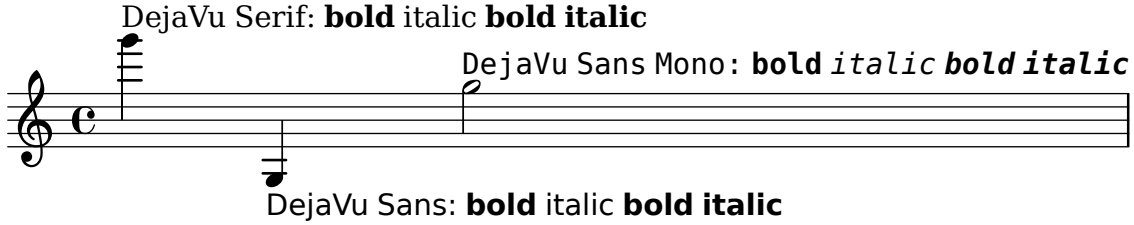
```
%{
You may have to install additional fonts.

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 }
    }
  }
}
```
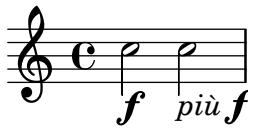


Combinaison de nuance et de texte

Certaines indications de nuance peuvent requérir un complément textuel, comme « più *f* » ou « *p* subito ». Elles se réalisent aisément à l'aide d'un bloc `\markup`. Le résultat se comportera comme un objet `TextScript`.

Voir aussi `Combinaison de nuance et de texte (2)`.

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

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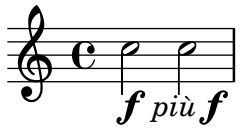


Combinaison de nuance et de texte (2)

Certaines indications de nuance peuvent requérir un complément textuel, comme « più *f* » ou « *p* subito ». Elles se réalisent à l'aide de la fonction Scheme `make-dynamic-script`. Les objets alors créés se comportent somme des objets graphiques `DynamicText`.

```
piuF = #(make-dynamic-script
          #{ \markup { \normal-text \italic più \dynamic f } #})

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



Combinaison de deux parties sur une même portée

L'outil de combinaison de parties (la commande `\partCombine`) permet d'avoir deux parties différentes sur une même portée. LilyPond ajoute automatiquement des indications textuelles, telles que « solo » ou « a2 ». Si votre intention n'est que de fusionner les parties, sans ajouter de texte, assignez faux à la propriété `printPartCombineTexts`.

Dans le cas de partitions vocales, et plus particulièrement d'hymnes, ces « solo/a2 » ne sont d'aucune utilité, aussi vaut-il mieux les désactiver. Dans le cas où il y aurait alternance entre *solo* et *tutti*, il vaut mieux faire appel à de la musique polyphonique standard.

Voici trois moyens d'imprimer deux parties sur une même portée : en polyphonie normale, avec `\partCombine` sans indication supplémentaire, et avec `\partCombine` commentée.

```
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) |
```
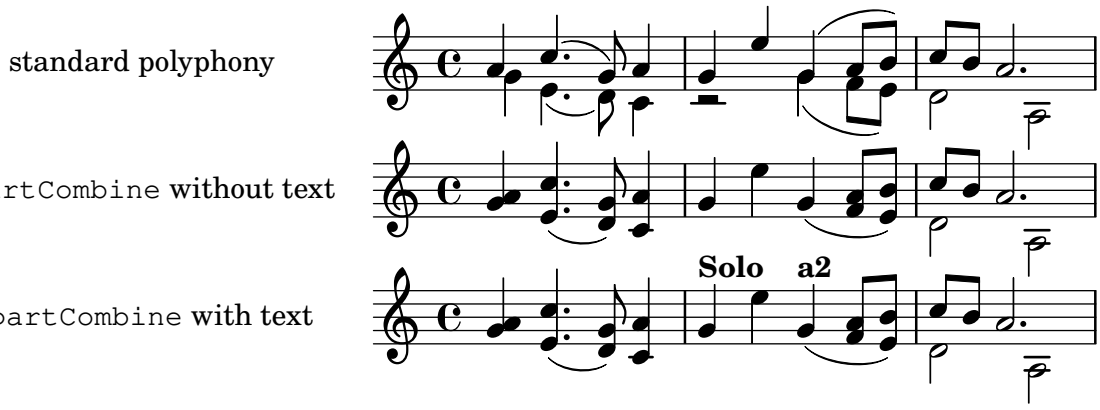
```
  d2 \stemDown a
}

\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
    }
  }
}
```



Création d'extensions de texte

Les commandes \startTextSpan et \stopTextSpan permettent d'ajouter une ligne de prolongation aux indications textuelles, à l'instar des indications de pédale ou d'octaviation. Jouer sur les propriétés de l'objet TextSpanner permet d'en modifier le rendu.

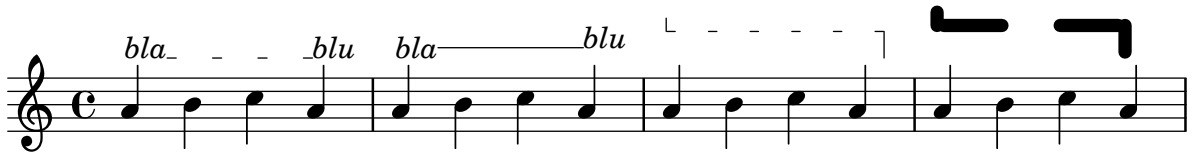
```
\paper { ragged-right = ##f }

\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
}
```



Champs d'entête

Voici la liste de tous les champs d'entête définis par défaut dans LilyPond. Grâce à l'activation de `print-all-headers`, beaucoup plus de champs s'affichent, indiquant la hiérachie des blocs `\header`.

```
\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
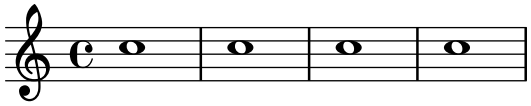
localtitle

localsubtitle

localpoet — **localinstrument** — localcomposer

localmeter — localarranger

localpiece — localopus



copyright

tagline

Intégration de PostScript dans un *markup*

Du code PostScript peut directement être intégré dans un bloc \markup.

En règle générale, il est préférable d'utiliser la commande de *markup* \polygon qui est fonctionnelle avec tous les moteurs de rendu 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
}
```
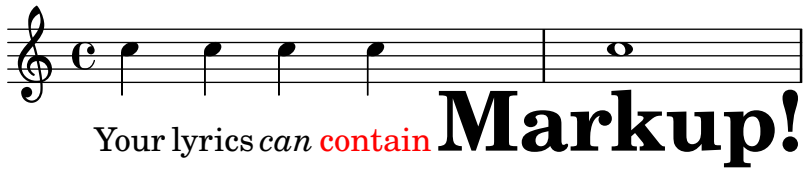


Mise en forme individuelle de syllabes

La commande \markup permet d'individualiser la mise en forme de certaines syllabes d'un bloc \lyricmode.

```
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! }
}

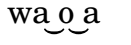
<<
  \new Voice = "melody" \mel
  \new Lyrics \lyricsto "melody" \lyr
>>
```



Séparation de syllables par une liaison

Des mots monosyllabiques peuvent s'attacher par une liaison à l'aide d'un tilde.

```
\lyrics {
  wa~o~a
}
```



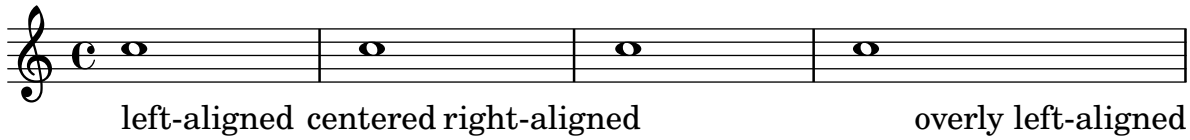
Alignement des syllabes

L'alignement horizontal des paroles peut se gérer à l'aide de la propriété `self-alignment-X` de l'objet `LyricText`. Les valeurs `-1` ou `#LEFT` produiront un alignement par la gauche, les valeurs `0` ou `#CENTER` un alignement centré, et les valeurs `1` ou `#RIGHT` un alignement par la droite. D'autres valeurs numériques sont permises – attention à ne pas oublier le préfixe Scheme '`#`' en cas de nombre négatif.

```
\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"
}
```



Liste de *markups*

Un texte susceptible de se répartir sur plusieurs page se saisit avec la commande `\markuplist`. La commande de *markup* `\paragraph` ici définie introduit un alinéa avant d'appeler `\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.
      }
    }
  }
}
```

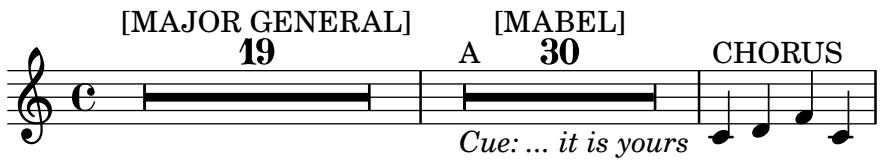
> Il y avait en Westphalie, dans le château de M. le baron de Thunder-ten-tronckh, un jeune garçon à qui la nature avait donné les mœurs les plus douces. Sa physionomie annonçait son âme. Il avait le jugement assez droit, avec l'esprit le plus simple ; c'est, je crois, pour cette raison qu'on le nommait 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.

Ajout de texte à un silence multimesure

Lorsque du texte est attaché à un silence multimesure, il sera centré dans la mesure, au-dessus ou en dessous de la portée. Afin d'étirer la mesure dans le cas où ce texte est relativement long, il suffit d'insérer un accord vide auquel on attache le texte en question, avant le silence

multimesure. Le texte attaché à un silence invisible sera aligné sur la gauche de là où serait positionnée la note dans la mesure. Cependant, si la taille de la mesure est déterminée par la longueur du texte, il apparaîtra comme centré.

```
\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
  }
}
```



De l'ubiquité des objets *markup*

Les objets textuels se saisissent soit en tant que simples chaînes de caractères bornées par des guillemets informatiques, soit dans des blocs `\markup` qui, eux, acceptent tout un panel de mises en forme avancée et améliorations graphiques.

En tant que tels, les blocs *markup* peuvent s'utiliser :

- pour tout objet `TextScript` (attaché aux notes par `-`, `^` ou `_`),
- pour tout `TextMark` introduit par les mots-clés `\textMark` ou `\textEndMark`, ou bien tout autre objet similaire comme `MetronomeMark` introduit par `\tempo`,
- comme bloc de *markup* indépendant, placé au niveau supérieur en dehors de tout bloc `\score`,
- pour toute définition au sein du bloc `\header` (titre, sous-titre, compositeur. . .) ou dans certaines variables définies dans le bloc `\paper` telle que `evenHeaderMarkup` pour les numéros de page.

`\markup` peut aussi servir pour les paroles, les noms d'accord et les nuances. En fait, `\markup` peut servir à personnaliser l'apparence de pratiquement n'importe quel objet, comme l'illustre cet exemple qui recourt à différentes méthodes.

```
\paper {
  paper-width = 8\cm
  paper-height = 8\cm
}

\header {
  title = \markup "Title"
  tagline = \markup "(tagline)"
}

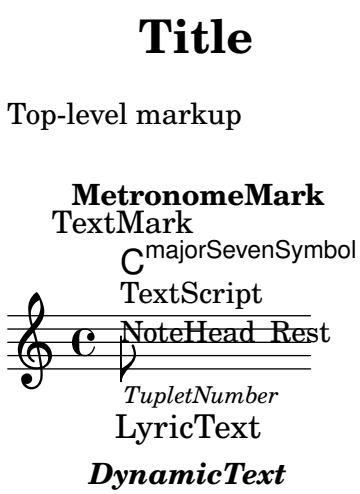
\markup "Top-level markup"
```

```
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 }
>>
```



Impression du numéro de version

Il est possible déjouter le numéro de la version de LilyPond utilisée dans un *markup*.

```
\markup { Processed with LilyPond version #(lilypond-version) }
```

Processed with LilyPond version 2.27.4

Piano et paroles entre les portées

Lorsque la mélodie est doublée au piano, cela ne nécessite pas forcément une portée spécifique. Les paroles peuvent s'insérer entre les deux portées de la partition pour piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

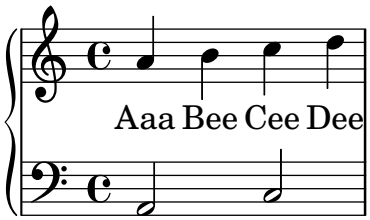
  a4 b c d
}

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 { }
}
```

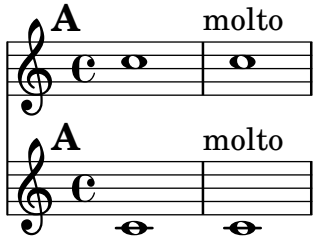


Impression des indications sur toutes les portées d'un système

Bien que ces indications textuelles et repères ne soient habituellement imprimées qu'au niveau de la portée supérieure d'un système, leur affectation peut être répercutée à chacune des portées.

```
\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
```

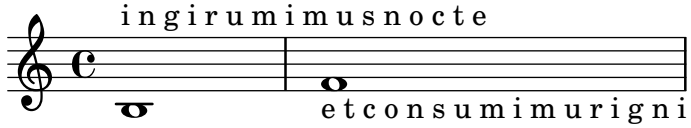
```
    }
    \context {
      \Staff
      \consists Mark_engraver
      \consists Text_mark_engraver
      \consists Staff_collecting_engraver
    }
  }
}
```



Impression de texte de droite à gauche

Du texte, inclus dans un objet *markup*, peut s'imprimer de droite à gauche, comme illustré ci-dessous.
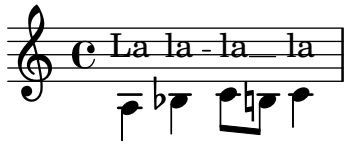
```
{
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```



Impression des paroles dans la portée

Des paroles peuvent venir s'inscrire dans la portée même. Ces paroles sont décalées par la dérogation `\override LyricText.extra-offset = #'(0 . dimension)` ; des commandes similaires s'occuperont des extenseurs et des tirets. Le décalage optimal ne peut s'obtenir qu'en procédant à tâtons.

```
<<
  \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 }
>>
```
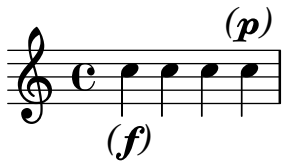


Indications de nuance entre parenthèses en italique

Le moyen le plus simple pour ajouter des parenthèses à une indication de nuance consiste à utiliser la commande \parenthesize et et d'ajuster la taille de la fonte, comme par exemple { c'4\tweak Parentheses.font-size 2 \parenthesize \f }. Toutefois, ces parenthèses se présentent droit alors que les indications de nuance apparaissent normalement en italique. Le code ci-dessous offre une solution pour présenter les parenthèses elles aussi en italique.

```
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
}
```



Bloc de texte indépendant sur deux colonnes

L'utilisation de la commande \markup permet de distribuer un bloc de texte indépendant sur plusieurs colonnes.

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { O sacrum convivium }
      \line { 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 { in which Christ is received, }
      \line { the memory of His Passion is renewed, }
```

```
      \line { the mind is filled with grace, }
      \line { and a pledge of future glory is given to us. }
      \line { Amen. }
    }
    \hspace #1
  }
}
```

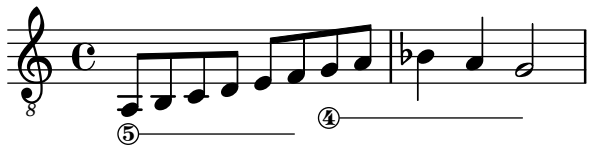
O sacrum convivium	*O sacred feast*
in quo Christus sumitur,	*in which Christ is received,*
recolitur memoria passionis ejus,	*the memory of His Passion is renewed,*
mens impletur gratia,	*the mind is filled with grace,*
futurae gloriae nobis pignus datur.	*and a pledge of future glory is given to us.*
Amen.	*Amen.*

Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
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
}
```



Encadrement sur trois côtés

Voici comment construire une commande de *markup* chargée d'agrémenter du texte ou autre annotation, d'une bordure sur trois côtés.

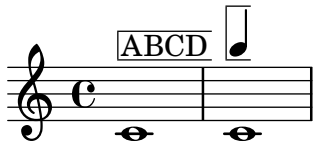
```
% New command to add a three-sided box, with sides north, west, and south.
% Based on the `box-stencil` command defined in `scm/stencil.scm`.
% Note that ";;" is used to comment a line in Scheme.
#(define-public (NWS-box-stencil stencil thickness padding)
   "Add a box around STENCIL, producing a 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

Différents éléments textuels peuvent s'ajouter à la notation, comme des titres ou des paroles, dès lors qu'ils sont encodés en UTF-8, grâce à un moteur de rendu reposant sur Pango. Selon les fontes disponibles devraient apparaître quatre lignes de texte : en bulgare (en cyrillique), en hébreux, en japonais et en portugais.

```
%{
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 {
           .
}

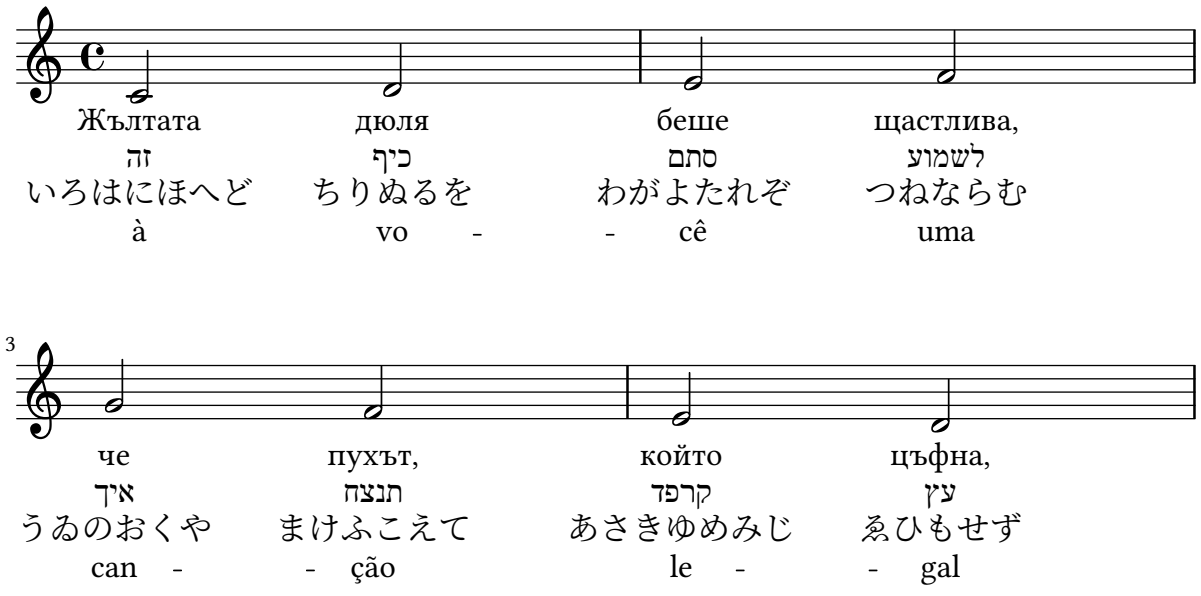
japanese = \lyricmode {




}

% "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 }
```



Ensemble vocal avec alignement des paroles selon le contexte

Ce canevas ressemble beaucoup à celui pour chœur à quatre voix mixtes. La différence réside dans le fait que les paroles sont positionnées en ayant recours à `alignAboveContext` et `alignBelowContext`.

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
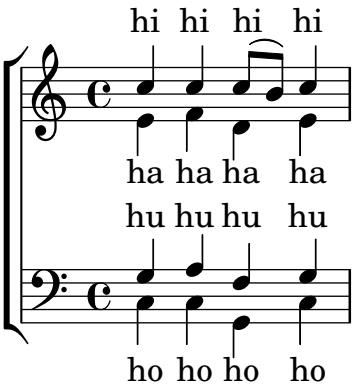
altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

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
  >>
}
```
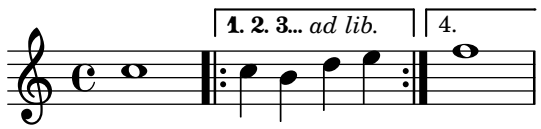


Indication de reprise avec texte grâce à repeatCommands

La commande \repeat volta permet d'indiquer facilement des reprises. Il est certains cas où l'adjonction d'un texte sous forme de \markup nécessite cependant de recourir à la propriété de contexte repeatCommands.

Dans la mesure où l'argument de repeatCommands est constitué d'une liste, le plus simple est de définir le texte dans une variable qui sera ensuite incorporée dans la liste en respectant la syntaxe Scheme #`(volta *texteIdentificateur*) ...). Les commandes de début et de fin de reprise peuvent s'ajouter séparément à la liste des éléments.

```
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))
}
```



Notation spécialisée
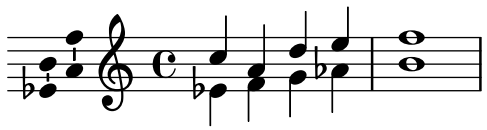
9 Vocal music

See also Section “Vocal music” dans *Manuel de notation*.

Un ambitus par voix

L’ambitus peut être individualisé par voix. Il faut en pareil cas éviter qu’ils se chevauchent.

```
\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
  }
>>
```



Ajout d’indicateurs à une portée dédoublée après un saut de ligne

Dans cet extrait sont définies les commandes \splitStaffBarLine, \convUpStaffBarLine et \convDownStaffBarLine. Ces commandes ajoutent une double flèche après la dernière barre de mesure d’une portée, indiquant par là que ses différentes voix disposeront de leur propre portée à la ligne suivante, ou bien se regrouperont.

Il est à noter que, dans l’implémentations ici proposée, sont dessinées des flêches sans dimension dans la marge droite. Ceci ne devrait pas poser de problème pour une impression normale. Il sera cependant nécessaire d’agrandir horizontalement la boîte englobante, comme indiqué ci-après, afin d’éviter le rognage lors d’une sortie sous forme d’image.

```
#(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
```

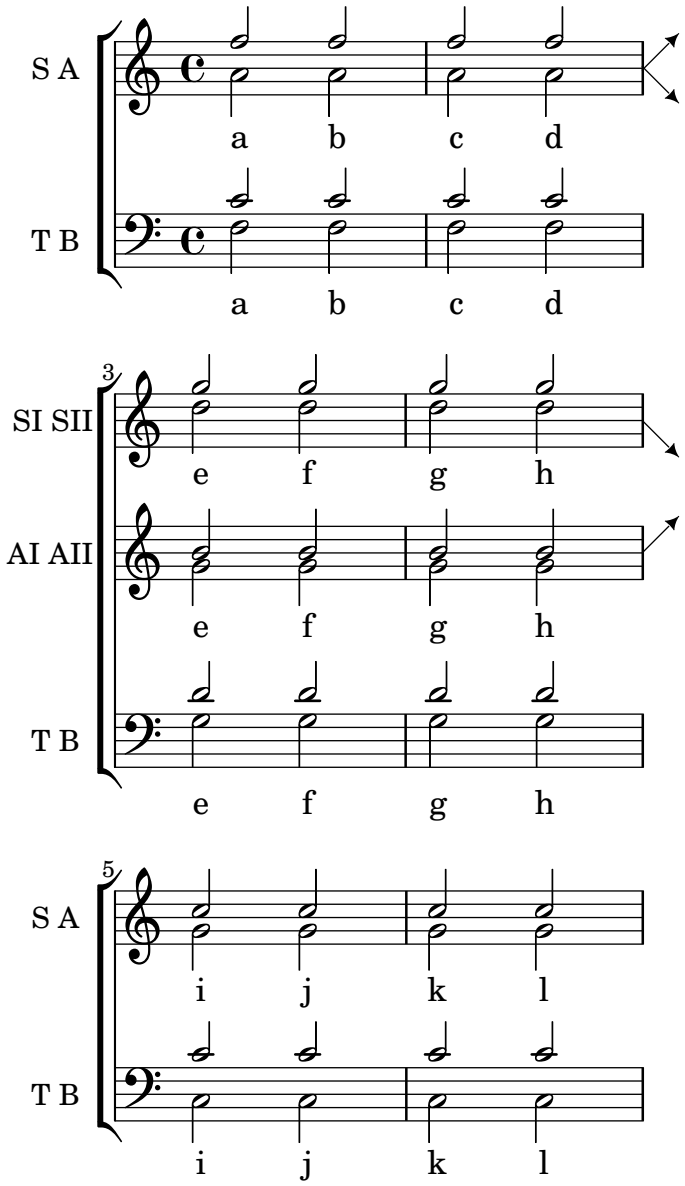
```
  \*4 f2
  \*4 g2
  \*4 c2
}

\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
      }
    }
}
```



Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur

Lorsque le nombre d'instruments cités dans la réduction pour piano se multiplie, vous pourriez avoir intérêt à créer votre propre fonction pour gérer ces repères. La fonction musicale `\cueWhile` prend quatre arguments : la musique d'où provient la citation, telle que définie par `\addQuote`, le nom qui sera mentionné en regard de cette citation, son positionnement – `UP` ou `DOWN` selon qu'il sera attribué à `\voiceOne` et placé au-dessus ou `\voiceTwo` et placé en dessous – et enfin la musique du piano qui interviendra en parallèle. Le nom de l'instrument en question viendra s'aligner sur la gauche de la citation. Bien que vous puissiez effectuer plusieurs citations, elle ne peuvent être simultanées.

```
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 }
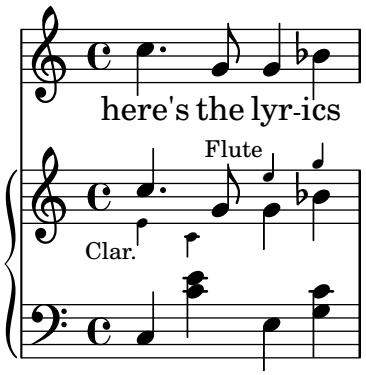
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

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
      }
    >>
  >>
```
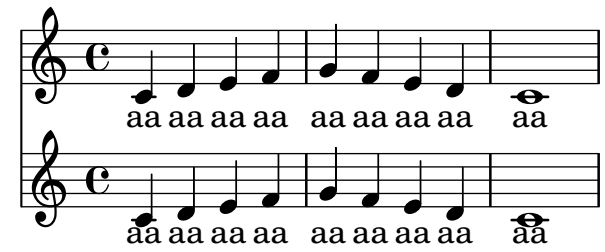
```
}
```



Ajustement de l'espacement vertical des paroles

Le code ci-dessous indique comment rapprocher de la portée à laquelle elle est affectée, une ligne de paroles.

```
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
>>
```
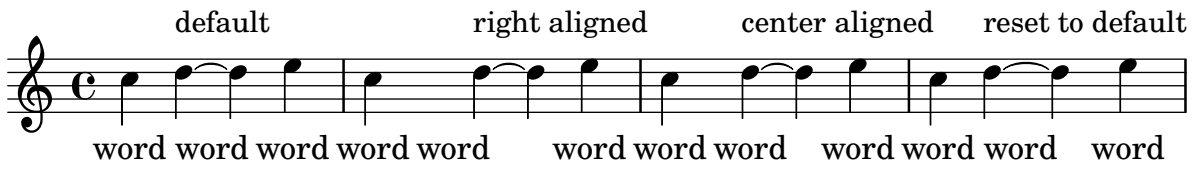


Alignement de la première syllabe d'un mélisme

Par défaut, une syllabe qui entame un mélisme est alignée sur la note par la gauche. Cet alignement peut se modifier à l'aide de la propriété `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
}
```
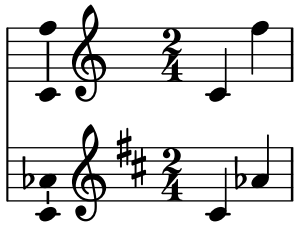


Ambitus

Un *ambitus* indique la tessiture, autrement dit les hauteurs extrêmes d'une voix.

Seules seront affichées les altérations non comprises dans l'armure. Les objets `AmbitusNoteHead` peuvent avoir des lignes supplémentaires.

```
\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'
    }
  }
>>
```
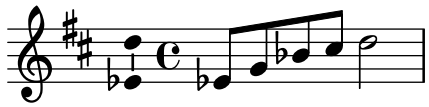


Ambitus après l'armure

L'ambitus se place par défaut à gauche de la clef. La fonction `\ambitusAfter` permet cependant de modifier ce positionnement ; sa syntaxe est `\ambitusAfter` *`grob-interface`* – voir Section "Graphical Object Interfaces" dans *Référence des propriétés internes* pour une liste des valeurs de *grob-interface* possibles.

L'un des cas d'usage est d'insérer l'ambitus entre l'armure et la métrique.
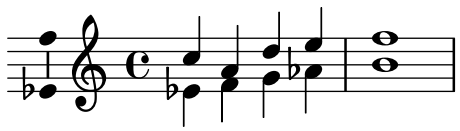
```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



Ambitus sur plusieurs voix

Si plusieurs voix se trouvent sur une même portée, on peut attribuer le graveur `Ambitus_engraver` au contexte `Staff` afin d'obtenir l'ambitus sur toutes les voix cumulées, non d'une seule des voix actives.

```
\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
  }
>>
```



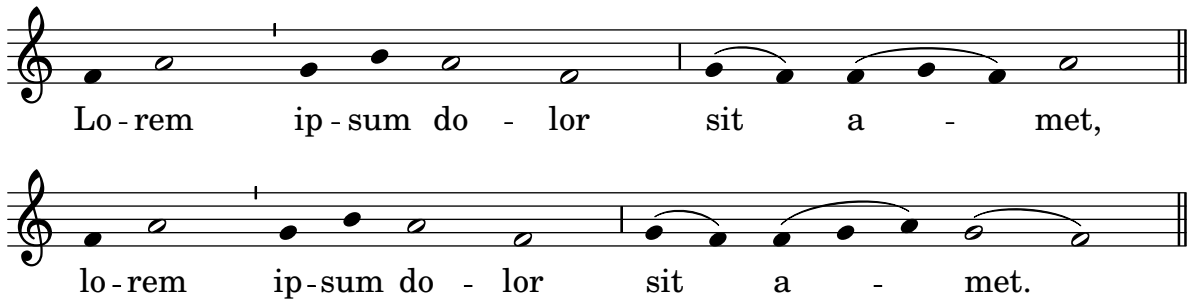
Exemples de notation ancienne – transcription moderne de musique grégorienne

Voici comment vous pourriez transcrire du grégorien. Pour mémoire, il n'y a en grégorien ni découpage en mesure, ni hampe ; seules sont utilisées des têtes de note blanches ou noires, ainsi que des signes spécifiques permettant d'indiquer des silences de différentes durées.

```
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
  >>
}
```



Psalmodie anglicane

Cet exemple illustre la manière de présenter un cantique tel qu'on le trouve dans l'église anglicane. Vous noterez comment sont ajoutés les couplets indépendamment de la musique. Dans le but de vous montrer plusieurs styles, comparez le code des deux couplets.

```
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
        >>
      >>

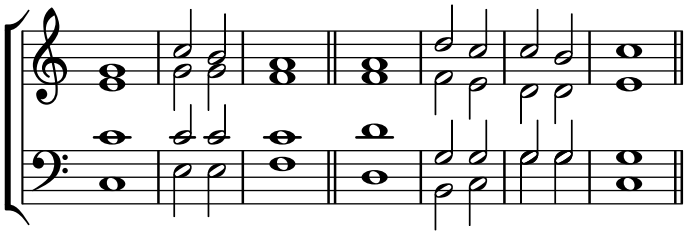
      \new Staff <<
        \clef "bass"
        \global
        \new Voice = "Tenor" <<
          \voiceOne
          \TenorMusic
        >>
        \new Voice = "Bass" <<
          \voiceTwo
          \BassMusic
        >>
      >>
    >>
```

```
    \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 }
            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 in the pro- }
    \line { vocation * and as in the \bold day of tempt- \tick }
    \line { -ation \tick 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.

Agencement de paroles séparées sur une seule ligne

On peut vouloir positionner les paroles de différents intervenants sur une même ligne, notamment pour un dialogue par bribes. Cet extrait montre comment y parvenir en ajustant la propriété nonstaff-nonstaff-spacing de l'objet graphique VerticalAxisGroup.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
              .nonstaff-nonstaff-spacing
              .minimum-distance = ##f
  }
}

aliceSings = \markup { \smallCaps "Alice" }
eveSings = \markup { \smallCaps "Eve" }

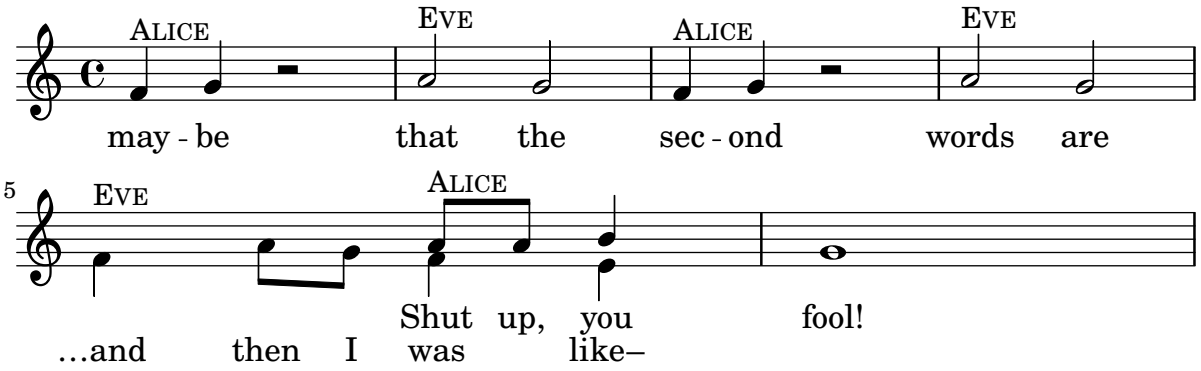
<<
  \new Staff <<
    \new Voice = "alice" {
      f'4^\aliceSings g' r2 |
      s1 |
      f'4^\aliceSings g' r2 |
      s1 | \break
      % ...

      \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--
  }
>>
```



Changement de fontes des couplets
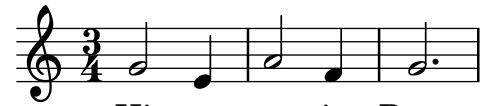
Chaque couplet peut disposer de sa propre fonte, y compris son numéro.

```
%{
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.

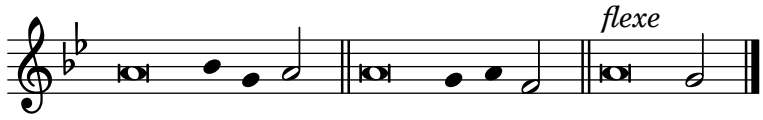
2. Oh, ché-ri, je t'aime

Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

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

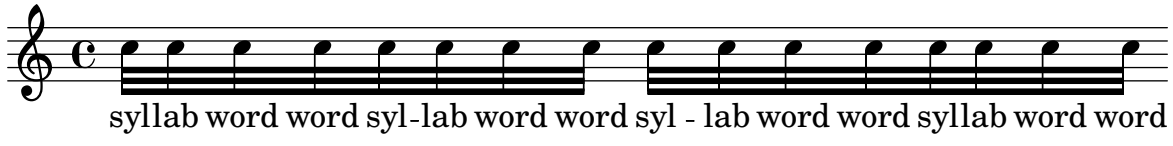
\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}
```



Impression forcée de tirets entre syllabes

LilyPond n'imprimera de tiret entre deux syllabes que s'il juge qu'il y a suffisamment d'espace. Ce comportement peut être modifié grâce à la propriété `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
}
```

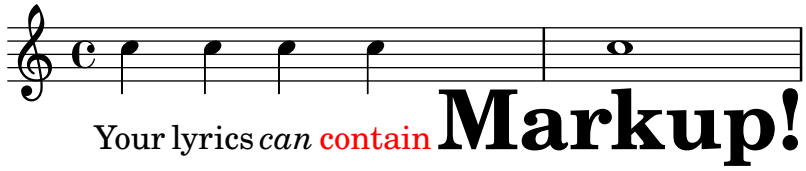


Mise en forme individuelle de syllabes

La commande `\markup` permet d'individualiser la mise en forme de certaines syllabes d'un bloc `\lyricmode`.

```
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! }
}

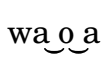
<<
  \new Voice = "melody" \mel
  \new Lyrics \lyricsto "melody" \lyr
>>
```



Séparation de syllables par une liaison

Des mots monosyllabiques peuvent s'attacher par une liaison à l'aide d'un tilde.

```
\lyrics {
  wa~o~a
}
```



Modèle pour cantique

Le code ci-dessous illustre la manière d'agencer un cantique liturgique dans lequel chaque ligne débute et se termine par une mesure incomplète. Vous noterez par ailleurs l'affichage des paroles indépendamment de la musique.

```
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

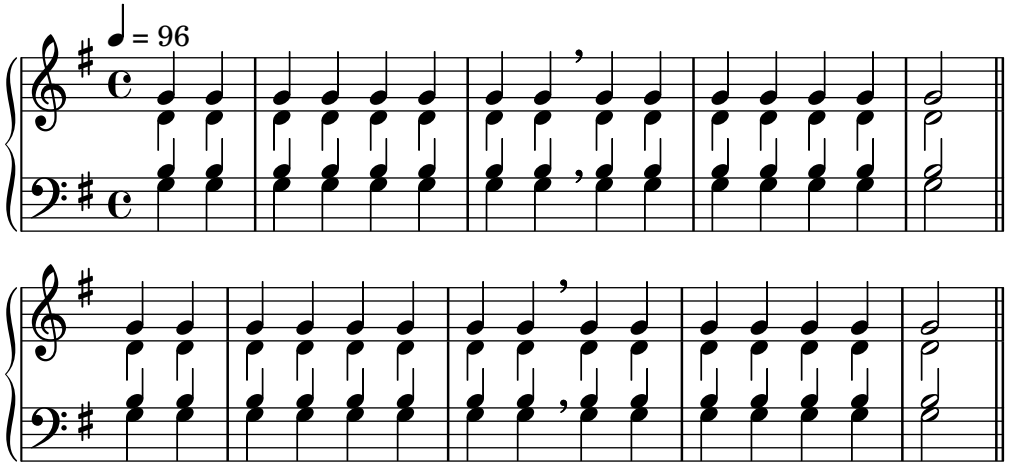
    \new Staff <<  % Start Staff = LH
      \global
      \clef "bass"
      \new Voice = "Tenor" <<  % Start Voice = "Tenor"
        \Timeline
        \voiceOne
        \TenorMusic
      >>  % End Voice = "Tenor"
      \new Voice = "Bass" <<  % Start Voice = "Bass"
        \Timeline
        \voiceTwo
        \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
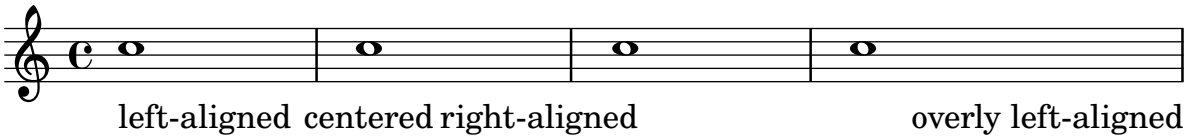
Alignement des syllabes

L'alignement horizontal des paroles peut se gérer à l'aide de la propriété `self-alignment-X` de l'objet `LyricText`. Les valeurs `-1` ou `#LEFT` produiront un alignement par la gauche, les valeurs `0` ou `#CENTER` un alignement centré, et les valeurs `1` ou `#RIGHT` un alignement par la droite. D'autres valeurs numériques sont permises – attention à ne pas oublier le préfixe Scheme '`#`' en cas de nombre négatif.

```
\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"
}
```
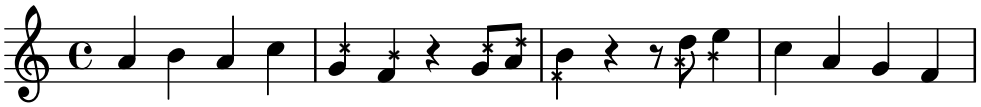


Adjonction d'une croix sur la hampe des notes d'un fragment parlé

Voici comment ajouter une croix aux hampes. Le début du fragment parlé est stipulé par une commande \speakOn, et la fin par une commande \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
  }
}
```



Orchestre, chœur et piano

Ce canevas illustre l'utilisation de contextes `StaffGroup` pour regrouper les instruments selon leur famille, imbriqués dans un `GrandStaff`, ainsi que le recours à la fonction `\transpose` pour les instruments transpositeurs. Dans tous les identificateurs, la musique est stockée en ut. Les notes peuvent tout aussi bien être saisies en ut ou dans la tonalité particulière de l'instrument avant d'être transposées puis affectées à une variable.

```
#(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 }
    >>

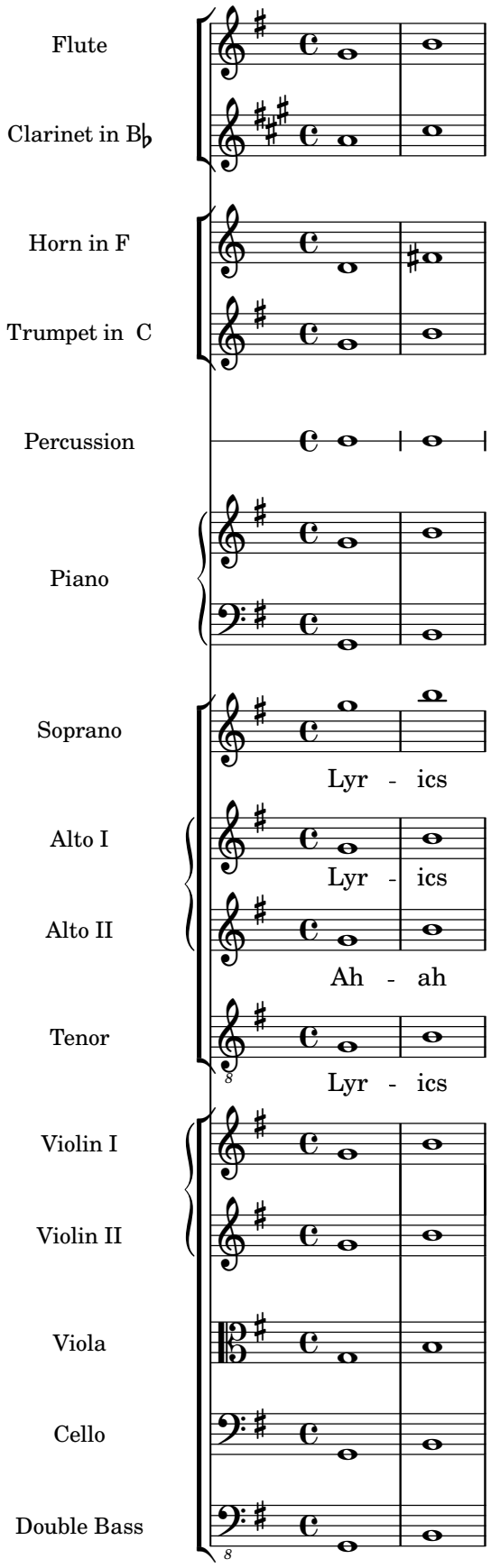
    \new StaffGroup = "StaffGroup_strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "Staff_violinI" \with {
          instrumentName = "Violin I"
        }
          \violinIMusic
        \new Staff = "Staff_violinII" \with {
          instrumentName = "Violin II"
        }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

Piano, mélodie et paroles

Il s'agit du format classique pour le chant : une portée pour la mélodie et les paroles au-dessus de l'accompagnement au piano.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

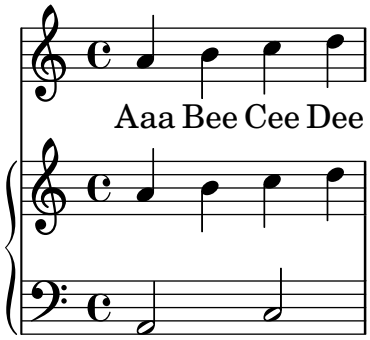
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}
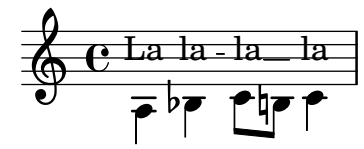
\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```



Impression des paroles dans la portée

Des paroles peuvent venir s'inscrire dans la portée même. Ces paroles sont décalées par la dérogation `\override LyricText.extra-offset = #'(0 . dimension)` ; des commandes similaires s'occuperont des extenseurs et des tirets. Le décalage optimal ne peut s'obtenir qu'en procédant à tâtons.

```
<<
  \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 }
>>
```
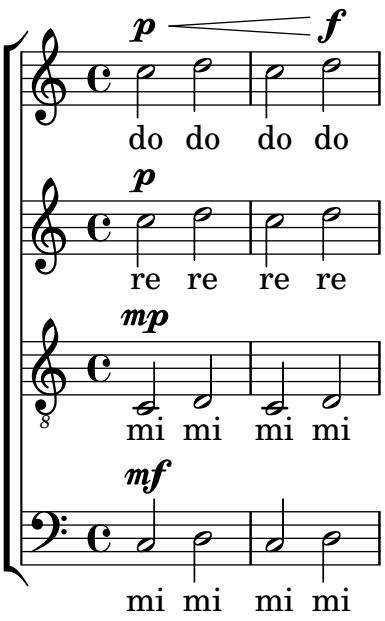


Modèle pour chœur SATB, sur quatre portées

Modèle pour chœur à quatre voix mixtes, chaque pupitre ayant sa propre portée.

```
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
    >>
  >>
}
```



Paroles, musique et accords

Ce canevas comporte tous les éléments d'une chanson : la mélodie, les paroles, les accords.

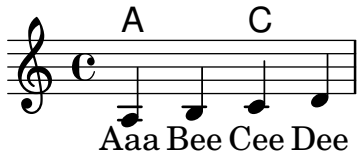
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  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 { }
}
```



Paroles, musique, accords et diagrammes de fret

Ce canevas comporte, en plus de la mélodie, des paroles et des accords, les diagrammes de fret.

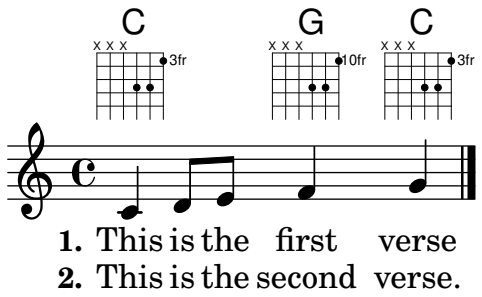
```
verseI = \lyricmode {
  \stanza "1."
  This is the first verse
}

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 { }
}
```



Portée unique et paroles
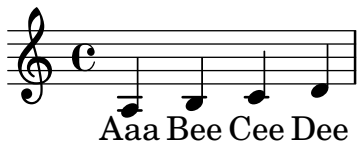
Ce canevas comporte une simple ligne mélodique agrémentée de paroles. Recopiez-le, ajoutez-y d'autres notes et paroles. Les ligatures automatiques sont ici désactivées comme il est d'usage en matière de musique vocale. Pour activer la fonction de ligature automatique, modifiez ou commentez la ligne en question.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

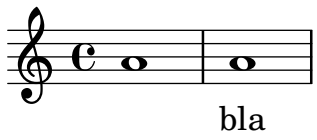
\score{
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Sauts de notes en mode paroles

La syntaxe `s` qui permet de « faire un saut dans le temps » n'est disponible qu'en mode notes et en mode accords. Dans les autres situations, comme en mode paroles par exemple, la commande `\skip` produit les mêmes effets.
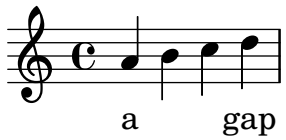
```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip1 bla1 }
>>
```



Sauts de notes en mode paroles (2)

Sachant qu'un s au sein d'un bloc \lyricmode est interprété non comme de l'espace mais comme une syllabe, l'utilisation de paires de guillemets informatiques ("") ou un caractère souligné simple (_) permet de « sauter » une note.

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



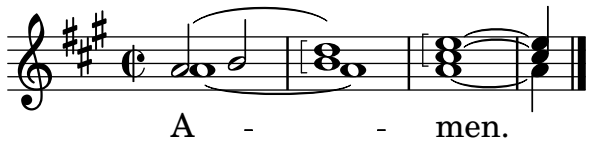
Using \arpeggioBracket to make divisi more visible

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 "|."
      }
```

```
    >>
  }
}
```

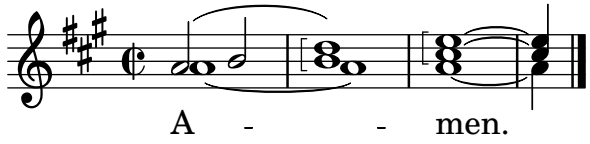


Utilisation d'un crochet pour rendre les divisions plus évidentes

La commande `nonArpeggiato` permet de mettre en évidence les divisions d'un pupitre en l'absence de hampe, comme on le voit régulièrement dans les partitions pour chœur.

```
\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
      }
    >>
  }
}
```



Recours aux balises pour produire une partition ancienne et moderne à partir de la même source

Grâce aux balises (*tags*), il est possible d'utiliser une même source pour produire une partition de musique mensurale et moderne. Dans cet exemple est créée la fonction `menrest` qui permettra

de positionner les silence comme dans la version originale, tout en respectant leur position sur une portée standard.

Les balises permettent aussi de gérer d'autres différenciations selon les besoins, comme des « mesures de silence » (R1, R\breve, etc.) en notation moderne mais des silences normaux (r1, r\breve, etc.) en notation ancienne. L'action de convertir de la musique mensurale en version moderne est communément appelée « transcription ».

L'appel à `c4.\Be c8 c\Am` revient au même que `c4.[ c8 c]`. Néanmoins, il évite les avertissements si le début intervient sur une note dépourvue de ligature mais pourrait s'en voir affublée de par l'utilisation du `Completion_heads_engraver`.

[La ligne légèrement plus courte dans la portée mensurale permet d'éviter que le glyphe de custode ne soit raccourci lors de la génération des images. On peut l'éviter à l'aide d'un `\with-true-dimensions` comme ici.]

```
\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

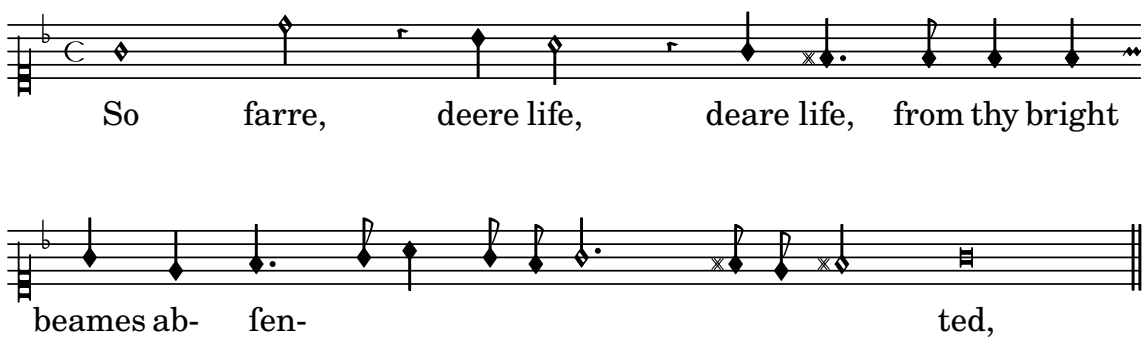
Music = \relative c'' {
  \key f \major
  g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
  g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}
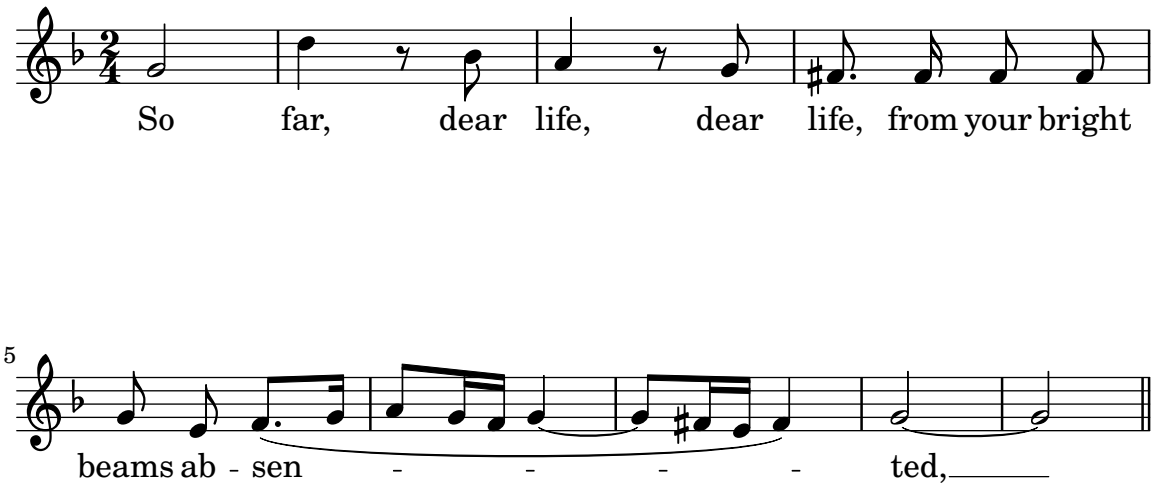
MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}
```

```
\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
    >>
  }
}
```



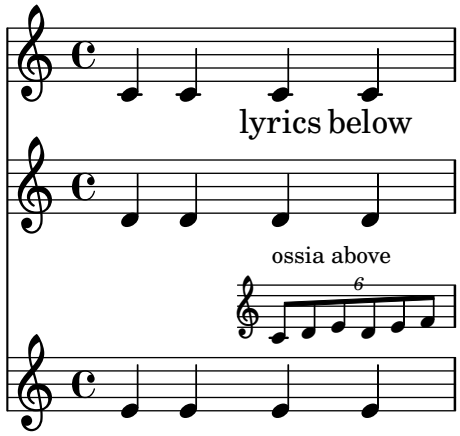


Positionnement d'une ossia et des paroles

Cet exemple illustre la manière de positionner une portée d'ossia et des paroles à l'aide des propriétés de contexte `alignBelowContext` et `alignAboveContext`.

```
\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]
        }
      }
    >>
  }
>>
```



Alignement vertical des numéros de couplet sur différentes portées

Il peut arriver que les numéros de couplet ne s'alignent pas verticalement s'ils sont attachés à différentes portées. Une adaptation de la propriété `self-alignment-X` de l'objet graphique `LyricText` permet d'y remédier.

```
\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
  }
>>

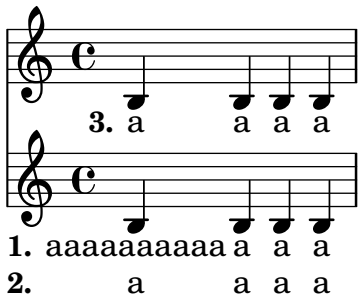
\markup \vspace #1
\markup {
  using \typewriter "self-alignment-X = #LEFT" }

<<
  \new Staff { b b b b }
  \new Lyrics \lyricmode {
    \stanza "3."
    a a a a
  }

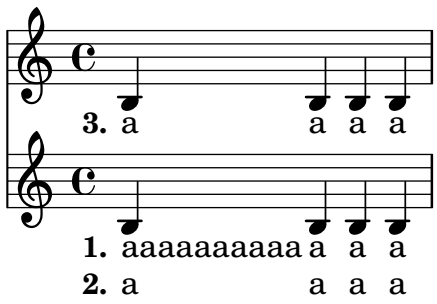
  \new Staff { b b b b }
```

```
  \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`



Centrage vertical de paroles communes

Il arrive, dans certaines partitions vocales, que quelques paroles soient communes à tous les couplets. Cette partie commune peut peut se présenter verticalement centrée, comme dans l'exemple suivant.

```
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.
}
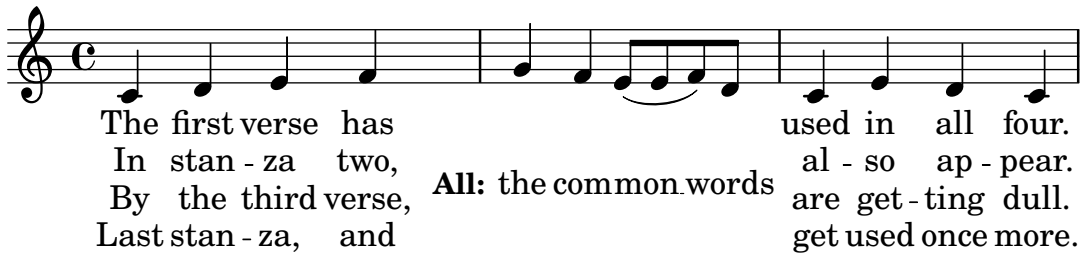
lyricsB = \lyricmode { In stan -- za two,   \skipFour al -- so ap -- pear. }

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
  >>
}
```



Ensemble vocal (simple)

Ce fichier constitue un canevas standard de partition pour chœur à quatre voix mixtes. Lorsque les ensembles s'étoffent, il est judicieux de recourir à une section spécifique incluse dans chacune des parties, tout particulièrement pour gérer la métrique et la tonalité qui, la plupart du temps, sont communes à tous les pupitres. Comme il est d'usage pour les hymnes, les quatre voix sont réparties sur deux portées.

```
\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
}
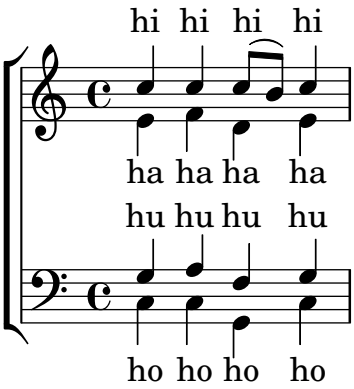
altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

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
  >>
}
```



Ensemble vocal avec réduction pour piano

Ce canevas ajoute une réduction pour piano à une partition standard pour chœur à quatre voix mixtes. Ceci illustre l'un des avantages de LilyPond : une expression musicale peut être réutilisée sans effort. Toute modification apportée à l'une des voix, mettons `tenorMusique`, est automatiquement reportée dans la réduction pour 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
}

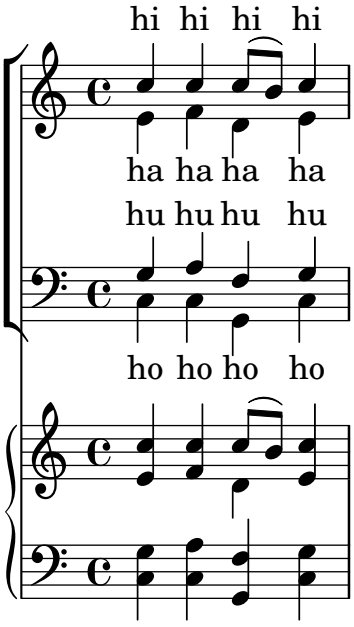
\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \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 >>
      >>
    >>
  >>
}
```



Ensemble vocal avec alignement des paroles selon le contexte

Ce canevas ressemble beaucoup à celui pour chœur à quatre voix mixtes. La différence réside dans le fait que les paroles sont positionnées en ayant recours à `alignAboveContext` et `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
}
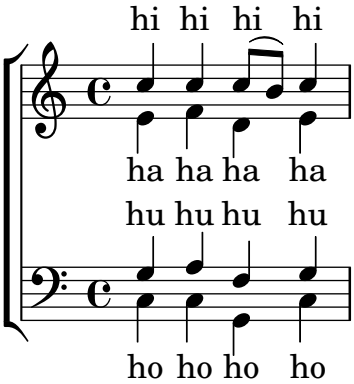
tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

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
  >>
}
```



Ensemble vocal avec couplet et refrain

Ce canevas illustre la manière d'agencer une œuvre vocale où le couplet est chanté en solo et le refrain à deux voix. Vous noterez le recours aux silences invisibles dans la variable `\global` : ils permettent de positionner les changements de métrique et autres éléments communs à toutes les parties, ce pour l'intégralité du morceau.

```
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 |
}
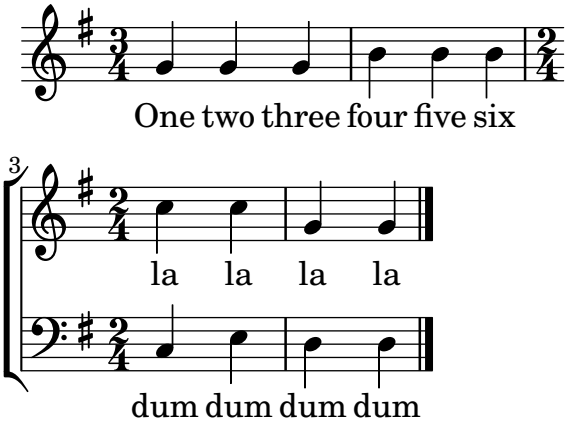
\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

    \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 Section “Keyboard and other multi-staff instruments” dans *Manuel de notation*.

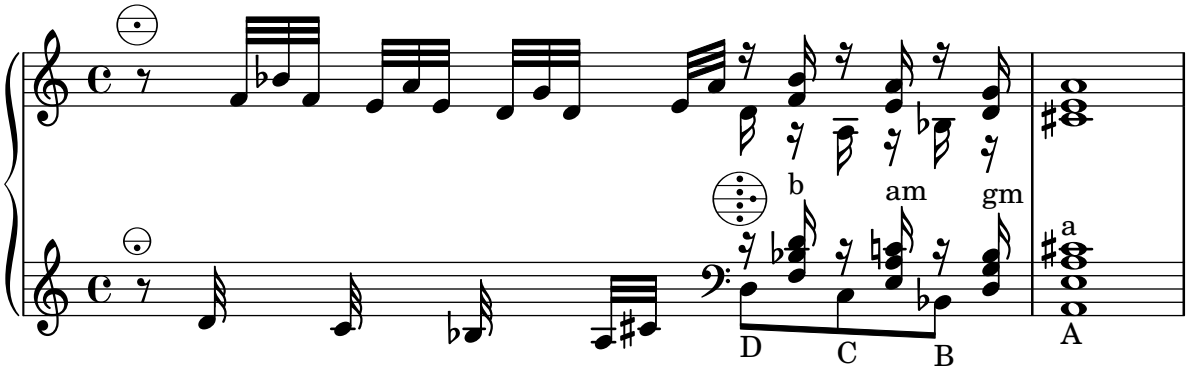
Symboles de registre pour accordéon

Les symboles spécifiques aux registres d’accordéon sont disponible tant sous forme de `\markup` que d’événements musicaux indépendants – un changement de registre a tendance à intervenir entre des événements musicaux réels. Les registres de basse ne sont pas très standardisés. Les différentes commandes disponibles sont regroupées à l’annexe Section “Registres d’accordéon” dans *Manuel de notation*.

```
#(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" }
    >>
  }
>>
```
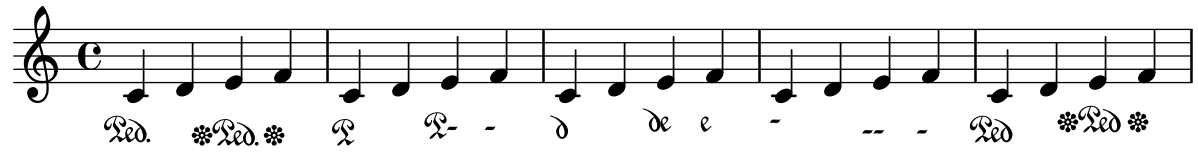


Modification du texte des indications de pédale

La propriété de contexte `Staff.pedalSustainStrings` permet de définir le texte affiché pour les instructions de pédale Les seuls caractères autorisés sont les glyphes particuliers de pédale ici présentés.

```
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
}
```

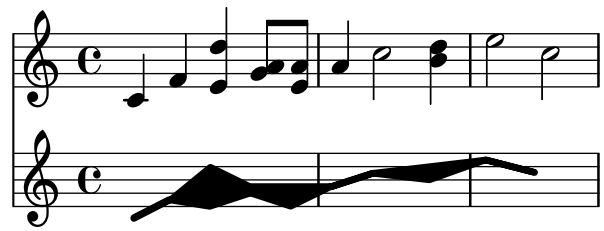


Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

```
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
>>
```



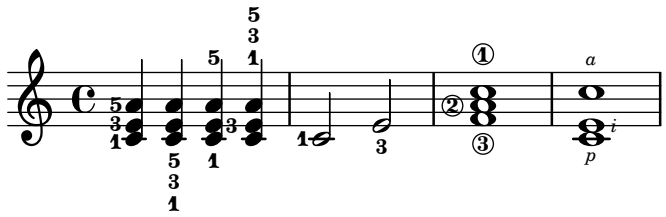
Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise à l'aide de la propriété `fingeringOrientation`. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser

une syntaxe d'accord <...>, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue à l'aide des propriétés stringNumberOrientation et strokeFingerOrientation.

Ces propriétés peuvent prendre une liste de trois valeurs. Elles contrôlent si les informations doivent être placées au-dessus (en présence d'un up) ou au-dessous (en présence d'un down), à gauche (en présence d'un left ou à droite (en présence d'un right). Par contre, si un positionnement n'est pas mentionné, aucun doigté n'y figurera. LilyPond tient compte de ces contraintes et trouvera le meilleur emplacement pour le doigté des notes de l'accord qui suit. Bien entendu, left et right sont exclusifs l'un de l'autre – les doigtés ne peuvent se placer que d'un seul côté.

```
\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 >
}
```
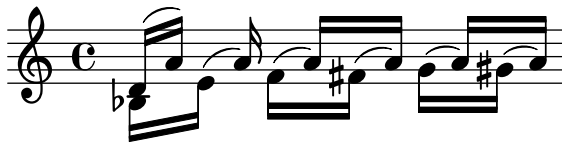


Création d'une liaison entre plusieurs voix

Dans certaines situations, il peut être nécessaire de lier des notes appartenant à des voix différentes. La solution consiste à ajouter une note invisible dans l'une des voix, à l'aide de la commande \hideNotes.

Voici par exemple la mesure 235 de la chaconne de la seconde partita pour violon 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)
    }
  >>
}
```



Accord distribué et problème de hampe – solution

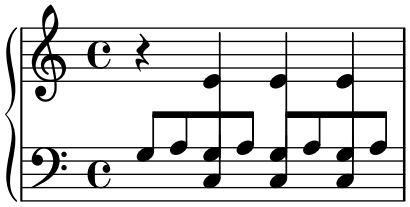
Il est parfois préférable d'utiliser les hampes de l'autre portée pour créer des accords distribués, afin de tricher avec le détecteur de collision des ligatures de LilyPond. Dans l'exemple suivant, le fait de partir des hampes de la portée inférieure aurait rendu nécessaire l'adaptation du détecteur de collision des ligatures, par une clause

```
\override Staff.Beam.collision-voice-only = ##t
```

afin que LilyPond ne déplace pas les ligatures.

```
\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
    }
  >>

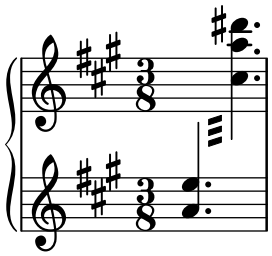
  \new Staff = bottom \relative c' {
    \clef bass
    \voiceOne
    g8 a g a g a g a
  }
>>
```



Trémolo et changement de portée

Dans la mesure où `\repeat tremolo` requiert deux arguments musicaux pour un trémolo d'accords, la note ou l'accord de la portée opposée doit être encadré par des accolades et se voir adjoindre la commande `\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
      }
    }
  }
>>
```
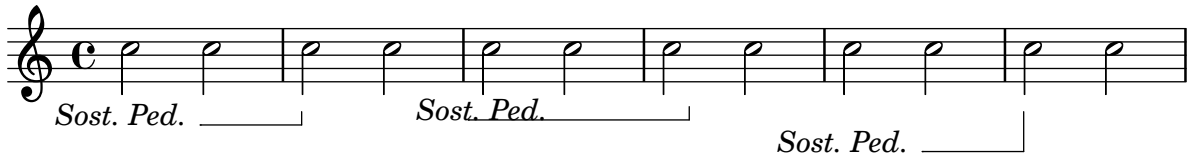


Affinage des indications de pédale

Les crochets d'indication de pédale peuvent se modifier de différentes manières.

```
\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
}
```



Indication d'un accord à cheval sur deux portées par un crochet

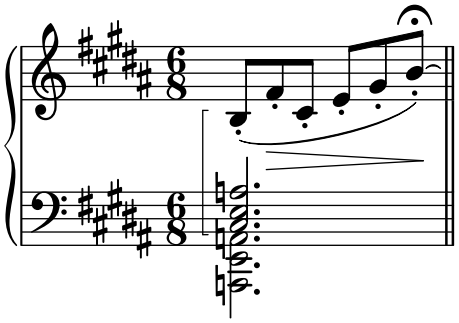
Un crochet « non-arpeggiato » peut indiquer que des notes réparties sur deux portées différentes doivent être jouées par la même main. Le contexte `PianoStaff` doit accepter ces crochets « distribués ».

Est ici reproduite la mesure 65 du prélude de Debussy *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
  }
>>
```



Symboles de jazz

Bien que compliqué de prime abord, voici un canevas tout à fait indiqué pour les ensembles de jazz. Vous noterez que tous les instruments sont notés en ut (`\key c \major`), la tonalité de concert. Les notes seront automatiquement transposées dès lors qu'elles seront inscrites dans une section `\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
  >>
}

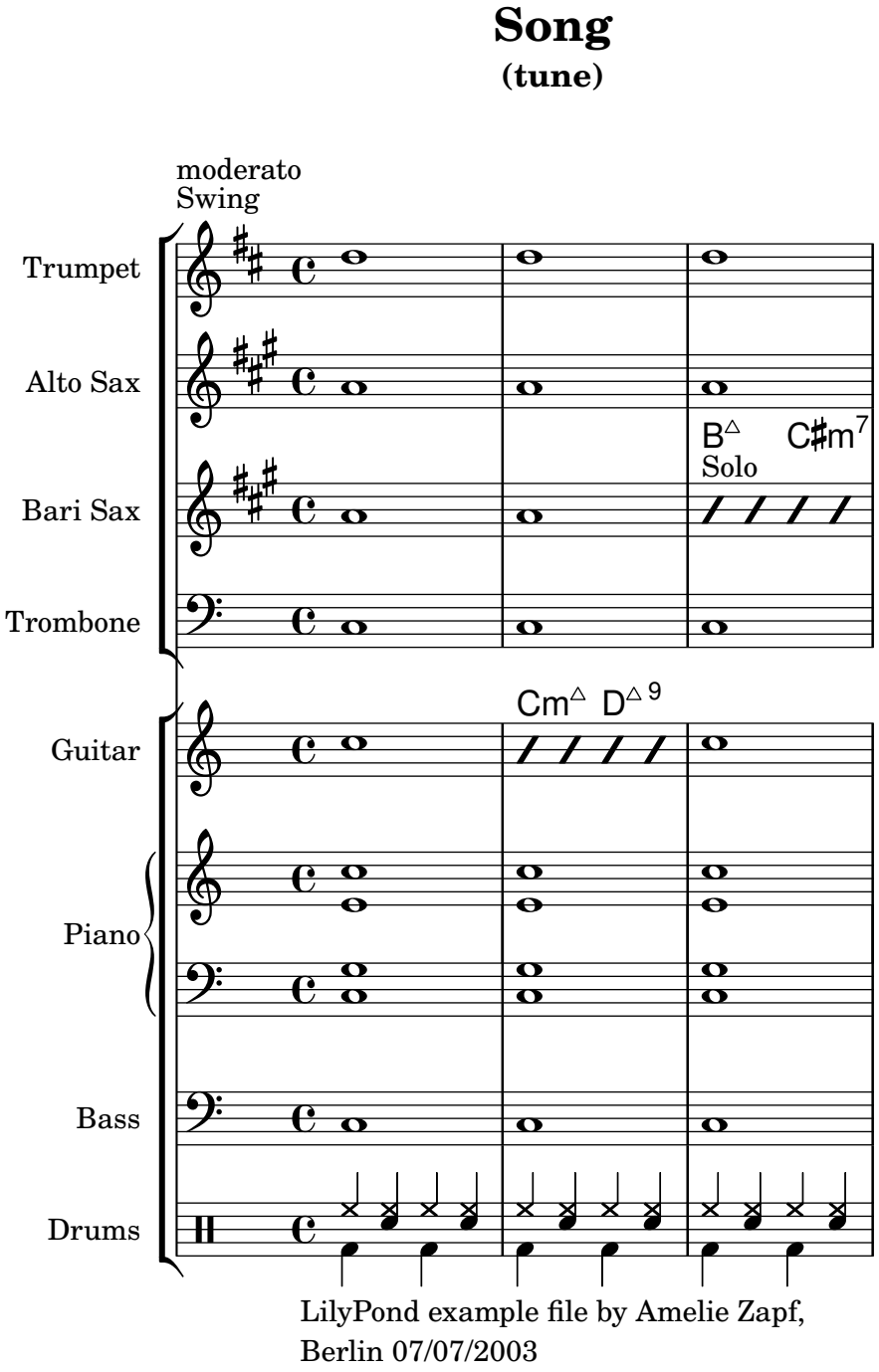
%%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%%%

\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 { }
 }
}
```
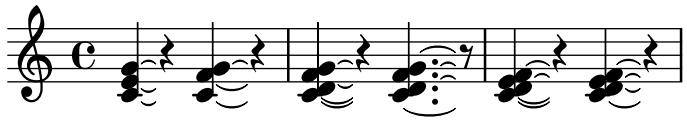


Liaison « Laissez vibrer »

Les indications de « laisser vibrer » ont une taille fixe. Leur formatage est accessible au travers de la propriété `tie-configuration`.

```
\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
}
```



Piano – cannevas simple

Voici une simple partition pour piano avec quelques notes.
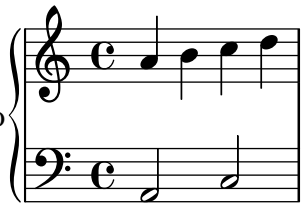
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  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 { }
}
```



Piano et paroles entre les portées

Lorsque la mélodie est doublée au piano, cela ne nécessite pas forcément une portée spécifique. Les paroles peuvent s'insérer entre les deux portées de la partition pour piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

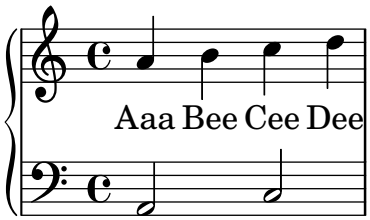
  a4 b c d
}

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 { }
}
```



Piano, mélodie et paroles

Il s'agit du format classique pour le chant : une portée pour la mélodie et les paroles au-dessus de l'accompagnement au piano.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}
```

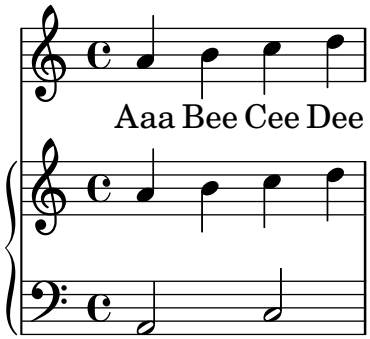
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  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 { }
}
```



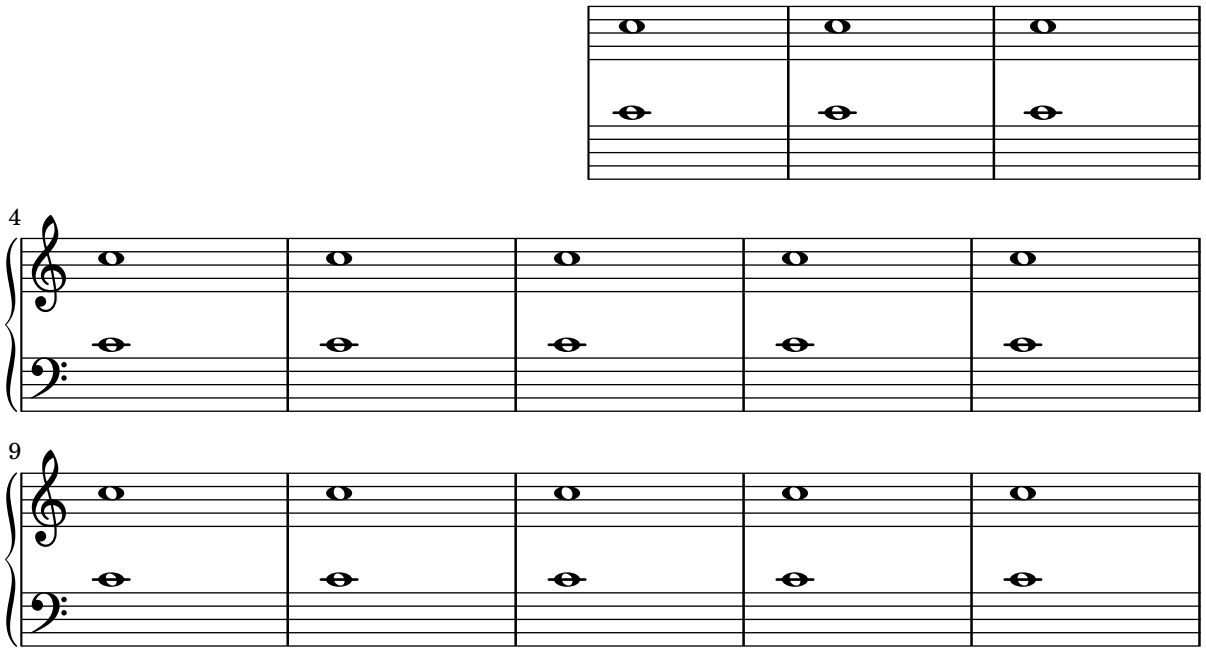
Retrait de l'accolade à la première ligne d'une pièce pour piano

Dans cet extrait est supprimée la première accolade d'un `PianoStaff` ou d'un `GrandStaff`, ainsi que les clefs. Ceci peut s'avérer utile pour couper et coller l'image générée dans de la musique préexistante.

Est utilisée la fonction `\alterBroken` afin de masquer l'accolade du début.

```
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
    }
  }
}
```



Utilisation d'autoChange avec plus d'une voix

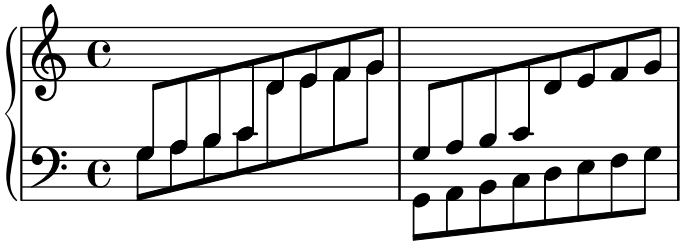
L'instruction \autoChange est opérationnelle y compris en présence de voix multiples.

```
\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
    }
  >>
}
```



Ensemble vocal avec réduction pour piano

Ce canevas ajoute une réduction pour piano à une partition standard pour chœur à quatre voix mixtes. Ceci illustre l'un des avantages de LilyPond : une expression musicale peut être réutilisée sans effort. Toute modification apportée à l'une des voix, mettons `tenorMusique`, est automatiquement reportée dans la réduction pour 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
}

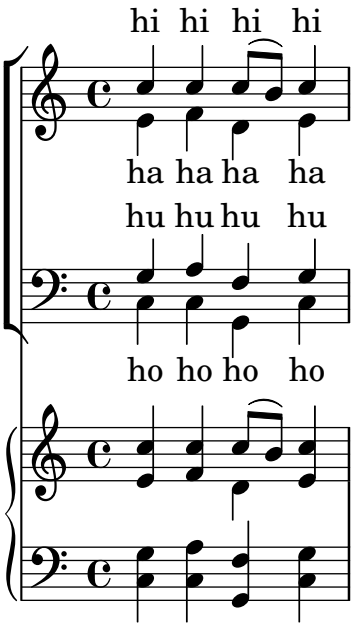
bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\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
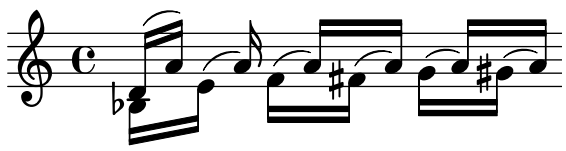
See also Section “Unfretted string instruments” dans *Manuel de notation*.

Création d’une liaison entre plusieurs voix

Dans certaines situations, il peut être nécessaire de lier des notes appartenant à des voix différentes. La solution consiste à ajouter une note invisible dans l’une des voix, à l’aide de la commande `\hideNotes`.

Voici par exemple la mesure 235 de la chaconne de la seconde partita pour violon solo, BWV 1004, de Bach.
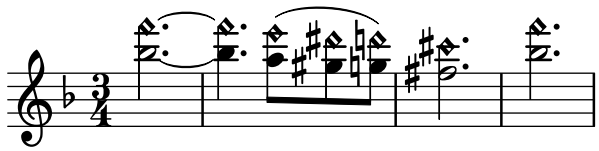
```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```



Harmoniques pointées

Les notes harmoniques artificielles, obtenues grâce à `\harmonic`, ne sont pas pointées. Ce comportement peut être modifié en activant la propriété de contexte `harmonicDots`.

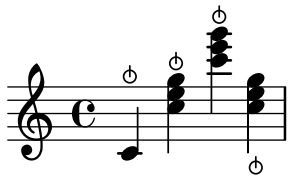
```
\relative c''' {
  \time 3/4
  \key f \major
  \set harmonicDots = ##t
  <bes f'\harmonic>2. ~
  <bes f'\harmonic>4. <a e'\harmonic>8( <gis dis'\harmonic> <g d'\harmonic>)
  <fis cis'\harmonic>2.
  <bes f'\harmonic>2.
}
```



Slap ou pizzicato Bartok

Un *slap*, aussi appelé *pizzicato à la Bartók*, est un pizzicato où, au lieu de tirer la corde de côté comme à l'accoutumée, « on la pince en la soulevant plus fortement, et à la verticale, en la lâchant violemment. Celle-ci frappe alors la touche, et produit à la fois la note et le son percussif » (Wikipedia). Il s'indique par un cercle flanqué d'un trait vertical.

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



Quatuor à cordes (conducteur)

Voici un canevas pour quatuor à cordes. Notez l'utilisation de la variable `\global` pour traiter la métrique et la tonalité.

Voir aussi `Quatuor à cordes, avec parties séparées`.

```
global= {
  \time 4/4
  \key c \major
}

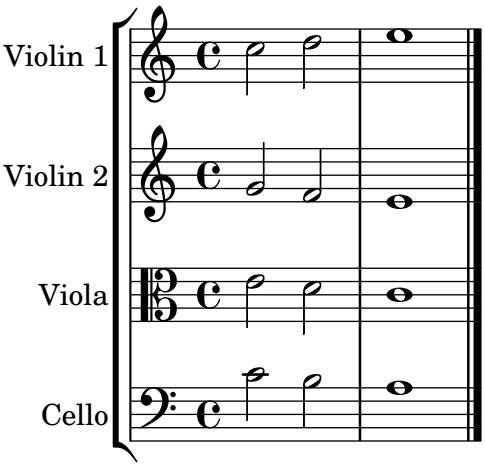
violinOne = \new Voice \relative c'' {
  c2 d
  e1
  \bar "|."
}

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 { }
}
```



Quatuor à cordes, avec parties séparées

Grâce à ce canevas, vous pouvez obtenir une partition d'excellente facture pour quatuor à cordes mais aussi, si le besoin s'en faisait sentir, une partie séparée par instrument. Par ailleurs, cet exemple illustre l'utilisation de la fonction `\tag` dans le but d'extraire des parties séparées.

Pour des raisons techniques, un même fichier source ne peut générer ici de multiples fichiers. Il vous faudra donc découper ce canevas en plusieurs fichiers séparés ; leur nom respectif est indiqué en commentaire.

`piece.ly` comporte tout ce qui a trait à la musique. Les autres fichiers – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` et `vlc.ly` – vous permettront d'obtenir un conducteur et les parties selon le pupitre.

N'oubliez pas de supprimer les commentaires superflus des fichiers individualisés !

```
% 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 { }
}

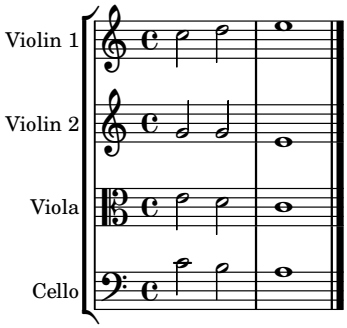
% vla.ly
% (This is the Viola part file.)

\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
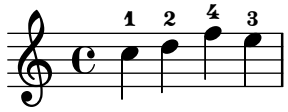
12 Fretted string instruments

See also Section “Fretted string instruments” dans *Manuel de notation*.

Ajout de doigtés à une partition

Les instructions de doigtés se saisissent selon une syntaxe très simple.

```
\relative c'' {
  c4-1 d-2 f-4 e-3
}
```

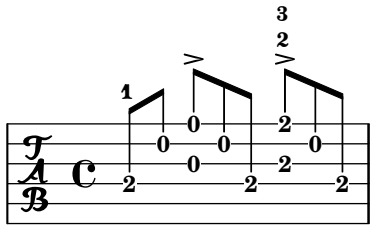


Ajout de doigtés à des tablatures

L’ajout de doigtés à des tablatures s’obtient en conjuguant des `\markup` et des `\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]
  }
}
```



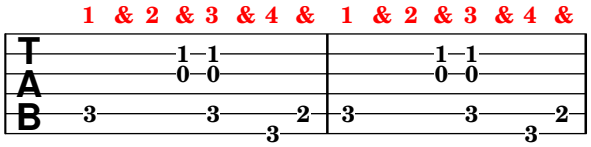
Ajout de *markups* à une tablature

Par défaut, les *markups* n’apparaissent pas dans une tablature.

Il suffit, pour les voir apparaître, de rétablir la propriété `stencil` de l’objet `TextScript` dans le contexte `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^"&" }
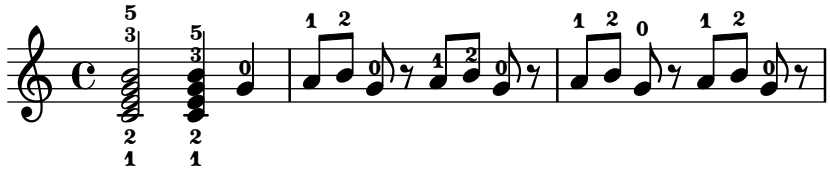
\score {
 \new TabStaff {
   \*2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```



Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Il est néanmoins possible d'annuler ce comportement. Une attention particulière doit toutefois être portée dans les cas où doigté et hampe vont dans la même direction : les indications de doigté n'évitent les hampe qu'en présence de ligature. Ce réglage peut s'adapter pour éviter toutes les hampes ou aucune d'elles. L'exemple suivant illustre ces deux options, ainsi que la manière de revenir au comportement par défaut.

```
\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
}
```

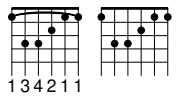


Barrés et diagrammes de fret automatiques

L'utilisation de diagrammes de fret automatiques permet d'imprimer une indication de barré dès lors qu'un même doigt s'applique à plusieurs cordes.

Lorsqu'aucune indication de doigt n'est fournie dans l'accord qui sera rendu sous forme de diagramme, aucune indication de barré n'apparaîtra puisqu'il n'y a aucun moyen d'identifier où les barrés devraient venir se positionner.

```
\new FretBoards {
  <f,-1 c-3 f-4 a-2 c'-1 f'-1>1
  <f, c f a c' f'>1
}
```



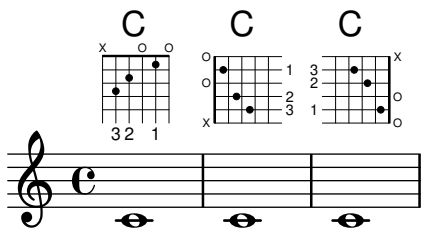
Orientation des diagrammes de fret

Les diagrammes de fret peuvent s'orienter de trois manières différentes. Ils s'aligneront par défaut sur la corde du haut ou le sommet du fret.

```
\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'
  }
>>
```



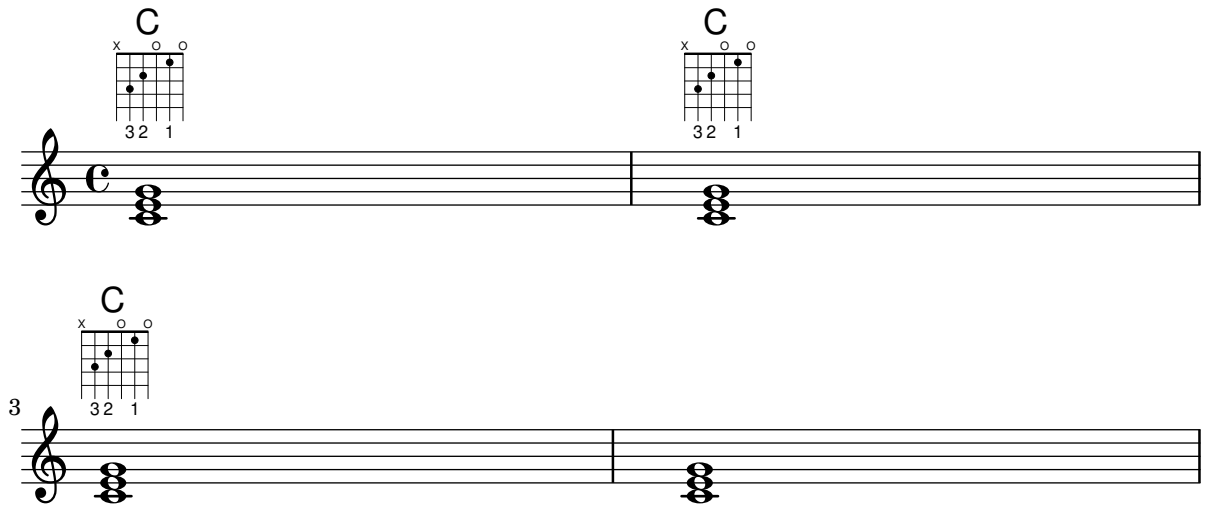
Changement d'accord et diagramme de fret

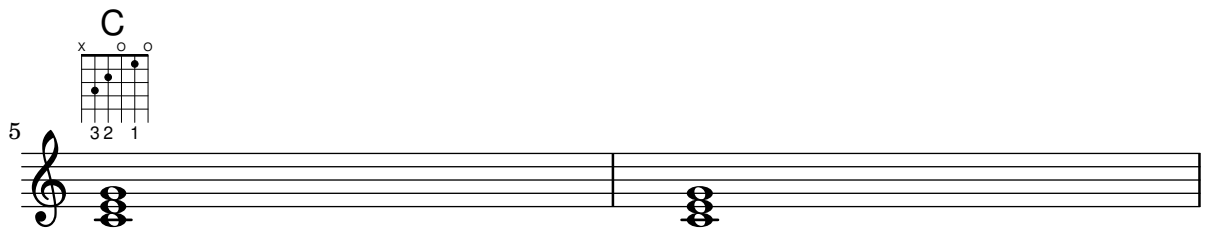
Vous pouvez opter pour n'imprimer les diagrammes de fret qu'à l'occasion d'un changement d'accord ou de saut de ligne.

```
\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 }
>>
```





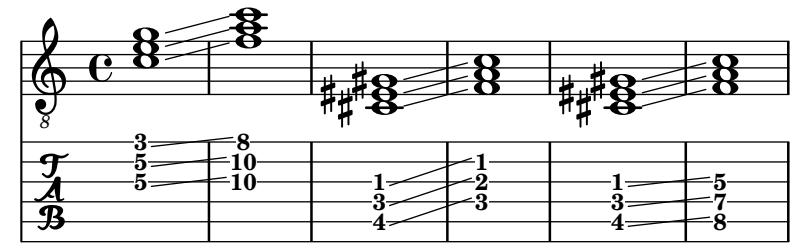
Glissando d'accords et tablature

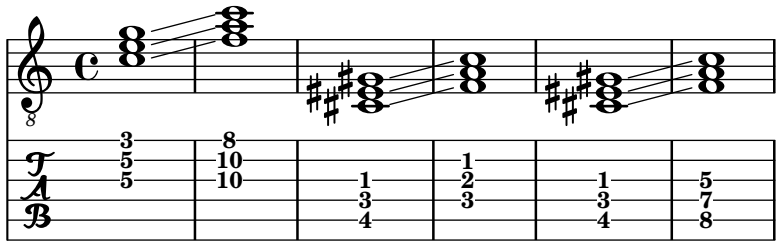
Un glissando sur des accords s'indique dans un `TabStaff` de la même manière que dans un `Staff`, à ceci près que nous pourrions avoir besoin des numéros de corde afin de déterminer correctement les frets d'arrivée.

```
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
    }
  >>
}
```





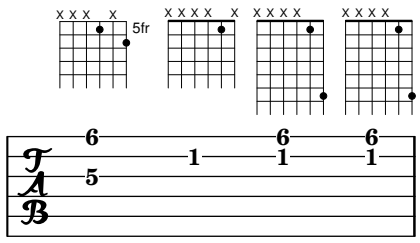
Accords avec doigté espacé pour FretBoards et TabVoice

Il peut arriver que le doigté d'un accord soit assez étendu. Sauf mention contraire, la valeur de la propriété de contexte maximumFretStretch est cependant fixée à 4, ce qui peut générer un avertissement « Pas de corde pour la hauteur... » et la note est omise. On peut régler maximumFretStretch sur une valeur appropriée ou assigner explicitement leur numéro de corde à toutes les notes d'un accord afin de palier ce problème.

```
% 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
>>
```



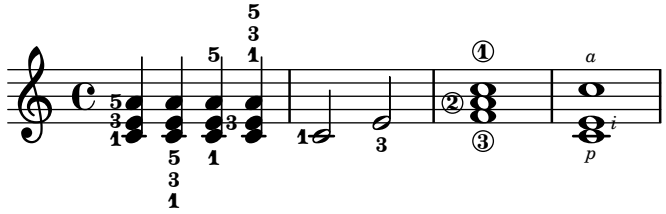
Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise à l'aide de la propriété fingeringOrientation. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord <...>, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue à l'aide des propriétés stringNumberOrientation et strokeFingerOrientation.

Ces propriétés peuvent prendre une liste de trois valeurs. Elles contrôlent si les informations doivent être placées au-dessus (en présence d'un up) ou au-dessous (en présence d'un down), à gauche (en présence d'un left ou à droite (en présence d'un right). Par contre, si un positionnement n'est pas mentionné, aucun doigté n'y figurera. LilyPond tient compte de ces contraintes

et trouvera le meilleur emplacement pour le doigté des notes de l'accord qui suit. Bien entendu, `left` et `right` sont exclusifs l'un de l'autre – les doigtés ne peuvent se placer que d'un seul côté.

```
\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 >
}
```



Personnalisation de diagrammes de fret

Les propriétés d'un diagramme de fret peuvent se modifier à l'aide de la propriété `'fret-diagram-details`. En matière de diagramme de fret, les adaptations s'appliquent à l'objet `FretBoards.FretBoard`. Un `FretBoards` est comparable à un `Voice` : il s'agit d'un contexte du plus bas niveau, et il n'est donc pas primordial de l'instancier de manière explicite pour adapter ses propriétés.

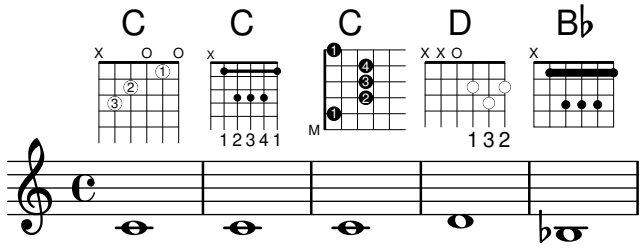
```
\include "predefined-guitar-fretboards.ly"

\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
  }
>>
```



Personnalisation des diagrammes de fret

Les propriétés d'un diagramme de fret sont modifiables grâce à la propriété `fret-diagram-details`. Lorsqu'ils sont générés sous forme de `\markup`, rien n'empêche de modifier les diagrammes en jouant sur les réglages de l'objet `Voice.TextScript` ou bien directement sur le *markup*.

```
<<
  \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

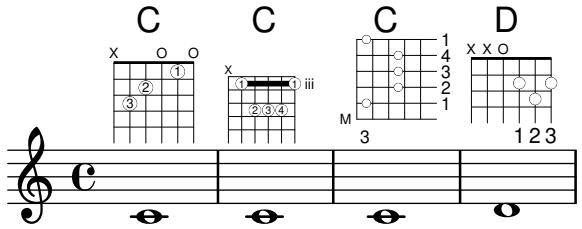
    %% C major for guitar, no barre, using defaults
       % terse style
    c'1^\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

    %% 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;"
      }
    }
  }
>>
```

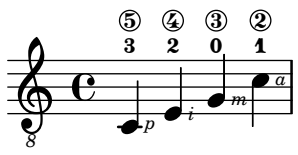


Doigtés, indications de corde, et doigtés main droite

L'exemple suivant illustre comment combiner des doigtés pour la main gauche, des indications de corde et des doigtés pour la main droite.

```
#(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
}
```



Notation de flamenco

La guitare flamenco fait appel à des éléments de notation particuliers :

- le *golpe* : coup frappé sur la caisse de résonance avec l'ongle de l'annulaire,
- une flèche pour indiquer le sens des butés (*strokes*),
- les doigtés : « p » pouce, « i » index, « m » majeur, « a » annulaire et « x » auriculaire,
- les *rasgueados* sur trois ou quatre doigts : grattage des cordes en étendant les doigts rapidement les uns après les autres dans un mouvement continu, se terminant sur l'index,
- les *abanicos* : séries de butés du pouce, index et auriculaire. Il existe aussi un *abanico 2* pour lequel l'index et l'annulaire remplacent l'auriculaire.
- *alza púa* : jeu rapide du pouce.

La plupart de ces symboles utilisent une flèche en plus des doigtés. Dans le cas d'un *abanico*, les têtes de note ne sont imprimées que pour le premier accord.

Le début du code ci-dessous répertorie le paramétrage de ces différents symboles, que vous pouvez copier dans un fichier `flamenco.ly` pour inclusion dans vos propres compositions.

```
%%%%%%  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
}

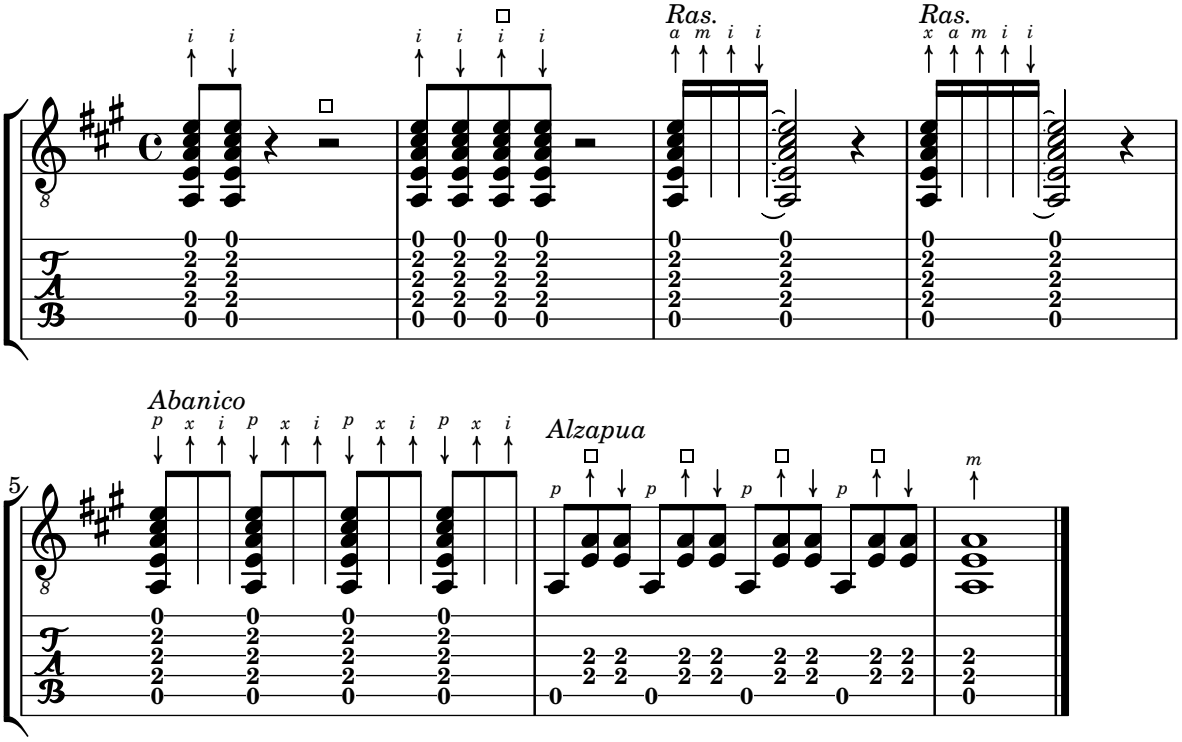
%%%%%%  Cut here ----- End of `flamenco.ly`.


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
  }
}
```



Construction et développement de diagrammes de fret

Voici différentes manières d'obtenir et de personnaliser des diagrammes de fret :

```
<<
  \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))
```

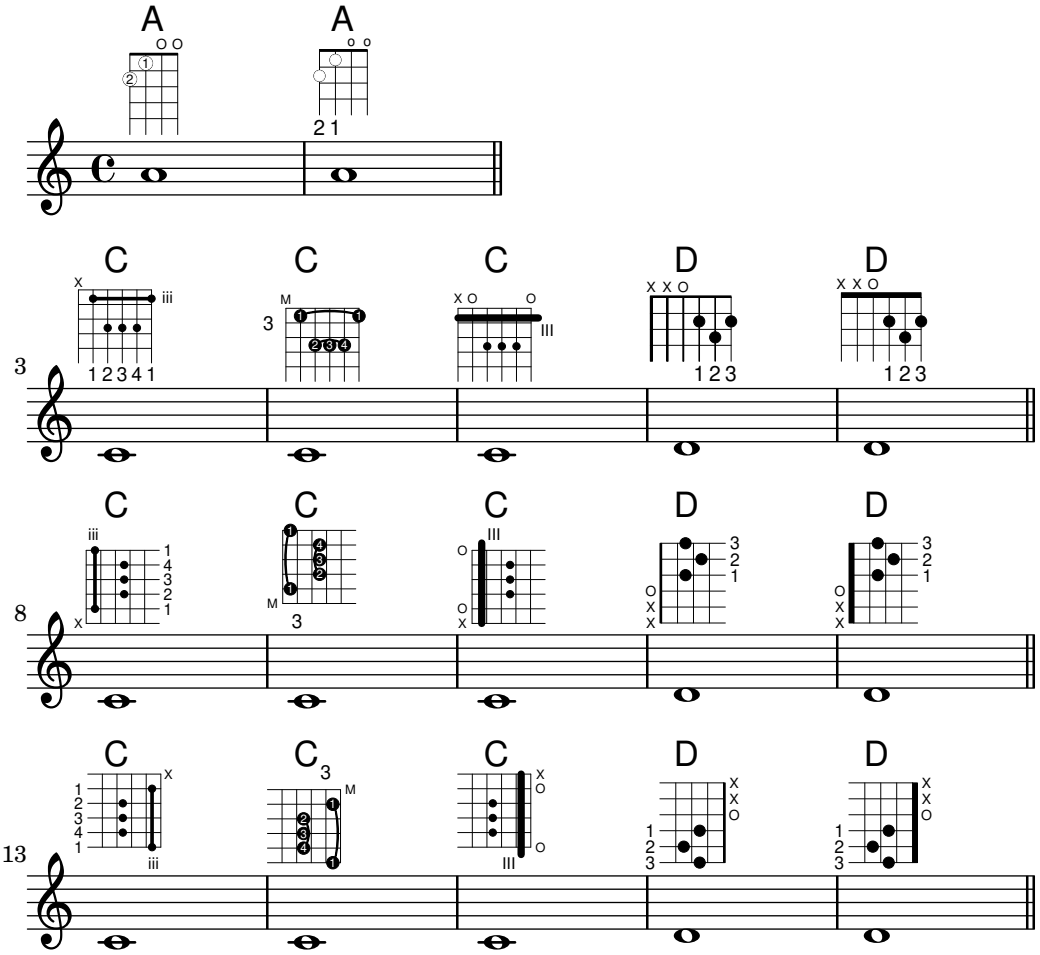
```
    % 15
    %
    % C major for guitar, with capo on third fret: verbose style.
    c'1^\markup
      % 110% of default size
      \override #'(size . 1.1)
      \override #'(fret-diagram-details
                   . ((number-type . roman-upper)
                      (dot-label-font-mag . 0.9)
                      (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
  }
}
```



Diagrammes de fret alternatifs

Vous pouvez tout à fait créer des tables de diagrammes de fret supplémentaires, notamment dans l'optique d'un fret alternatif pour un accord donné. Avant de pouvoir utiliser un diagramme alternatif, vous devrez avoir alimenté une table à cet effet. Les différents diagrammes seront ajoutés à cette table.

Il peut aussi bien s'agir d'une table vide, que de la recopie d'une table existante. La table servant de base pour les diagrammes prédéfinis est sélectionnée par la propriété `\predefinedDiagramTable`.

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

% Make a blank new fretboard table.
#(define custom-fretboard-table-one
   (make-fretboard-table))

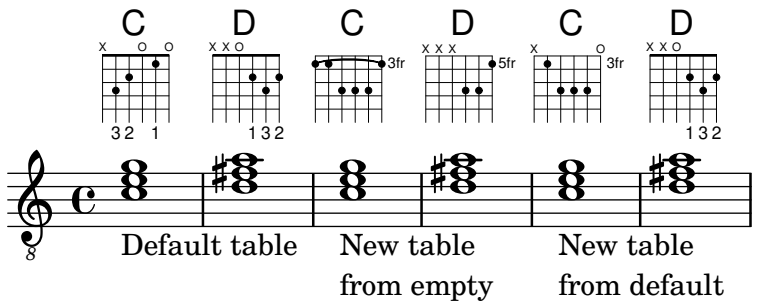
% Make a new fretboard table as a copy of `default-fret-table`.
#(define custom-fretboard-table-two
```

```
   (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 |
      }
    >>
  }
>>
```



Harmoniques et tablature

Le code suivant indique comment imprimer des harmoniques dan une tablature.

```
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" }
}

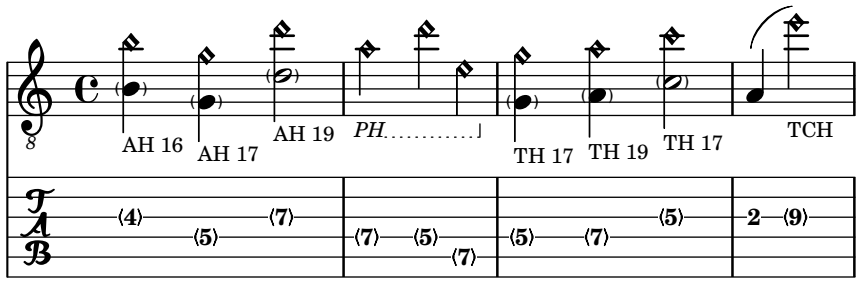
frettedStrings = {
  % artificial harmonics (AH)
  \harmonicByFret 4 g4\3
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 g2\3
```

```
  % 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
      }
    }
  >>
}
```



Indication d'un glissé de guitare

Contrairement au glissando, un « slide » peut aller d'un point non précisé du manche jusqu'à un fret spécifique. Ceci peut s'indiquer à l'aide d'une note d'ornement masquée précédant la note effectivement jouée, comme dans l'exemple suivant.

```
% Hide fret number: useful to draw slide into/from a casual point of
% the fretboard.

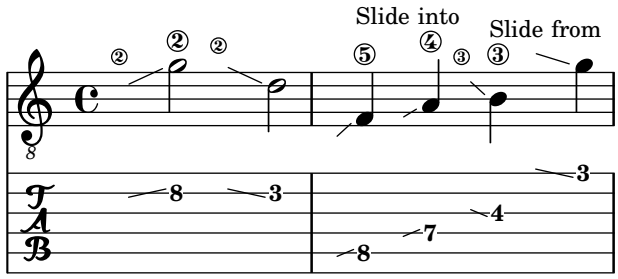
hideFretNumber = {
  \once \hide TabNoteHead
  \once \hide NoteHead
  \once \omit Stem
```

```
  \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
    }
  >>
}
```



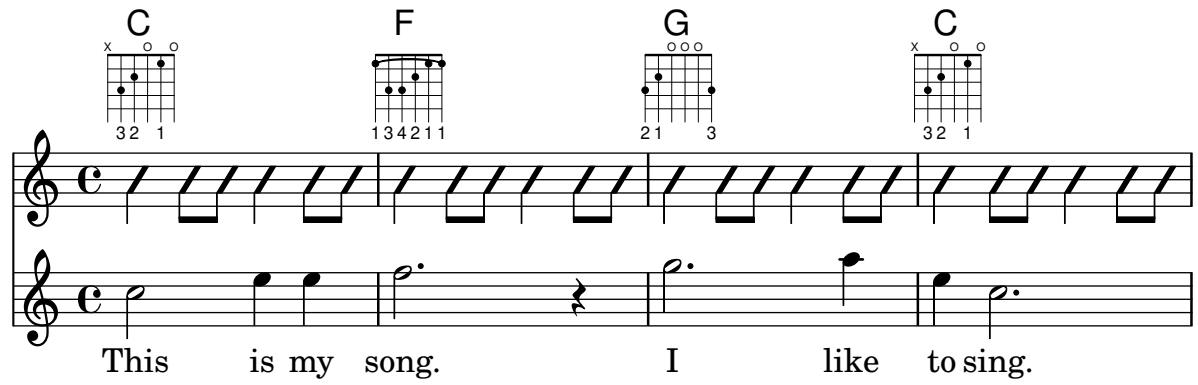
Rythmique et guitare

En matière de notation pour guitare, il arrive que soient indiqués les « coups de gratte » en plus de la mélodie, grilles d'accords et diagrammes de tablature.

```
\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"
```

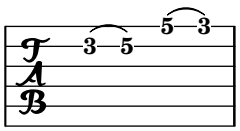
```
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
  \new Voice = "melody" \relative c'' {
    c2 e4 e4
    f2. r4
    g2. a4
    e4 c2.
  }
  \new Lyrics \lyricsto "melody" {
    This is my song.
    I like to sing.
  }
>>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



Hammer on et pull off

Hammer-on et *pull-off* peuvent s'indiquer par des liaisons.

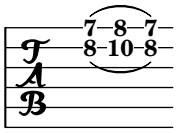
```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Hammer on et pull off avec accords

Dans le cadre de notes en accord, les *hammer-on* et *pull-off* sont indiqués par un arc simple. Vous obtiendrez néanmoins un arc double en réglant la propriété `doubleSlurs` sur `#t`.

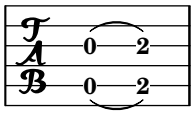
```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



Hammer on et pull off gérés par les voix

L'arc des *hammer-on* et *pull-off* est ascendant dans les voix une et trois, et descendant dans les voix deux et quatre.

```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Repositionnement d'un diagramme de fret

Différents moyens permettent de repositionner un diagramme de fret pour éviter des collisions ou le placer entre deux notes :

1. La modification des valeurs de `padding` ou de `extra-offset` comme pour la première ligne ;
2. L'adjonction d'une voix invisible dans laquelle les diagrammes sont attachés à des notes invisibles comme pour la deuxième ligne.

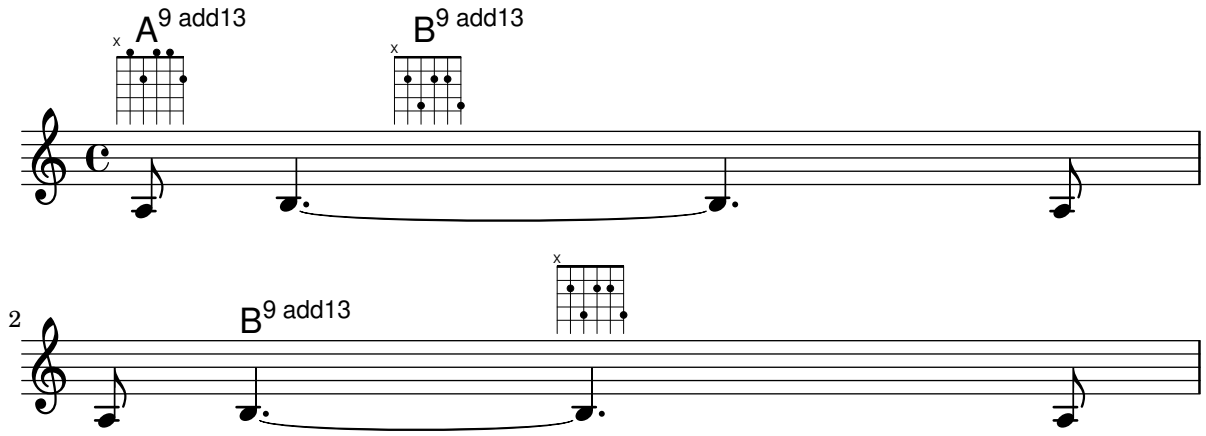
Lorsque le diagramme doit correspondre à une position rythmique dans la mesure, comme au troisième temps de la deuxième mesure, la seconde méthode est plus appropriée puisque le diagramme sera aligné sur le temps.

```
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;" }
      >> |
    }
  >>
}
```



Symboles de jazz

Bien que compliqué de prime abord, voici un canevas tout à fait indiqué pour les ensembles de jazz. Vous noterez que tous les instruments sont notés en ut (`\key c \major`), la tonalité de concert. Les notes seront automatiquement transposées dès lors qu'elles seront inscrites dans une section `\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
  >>
}

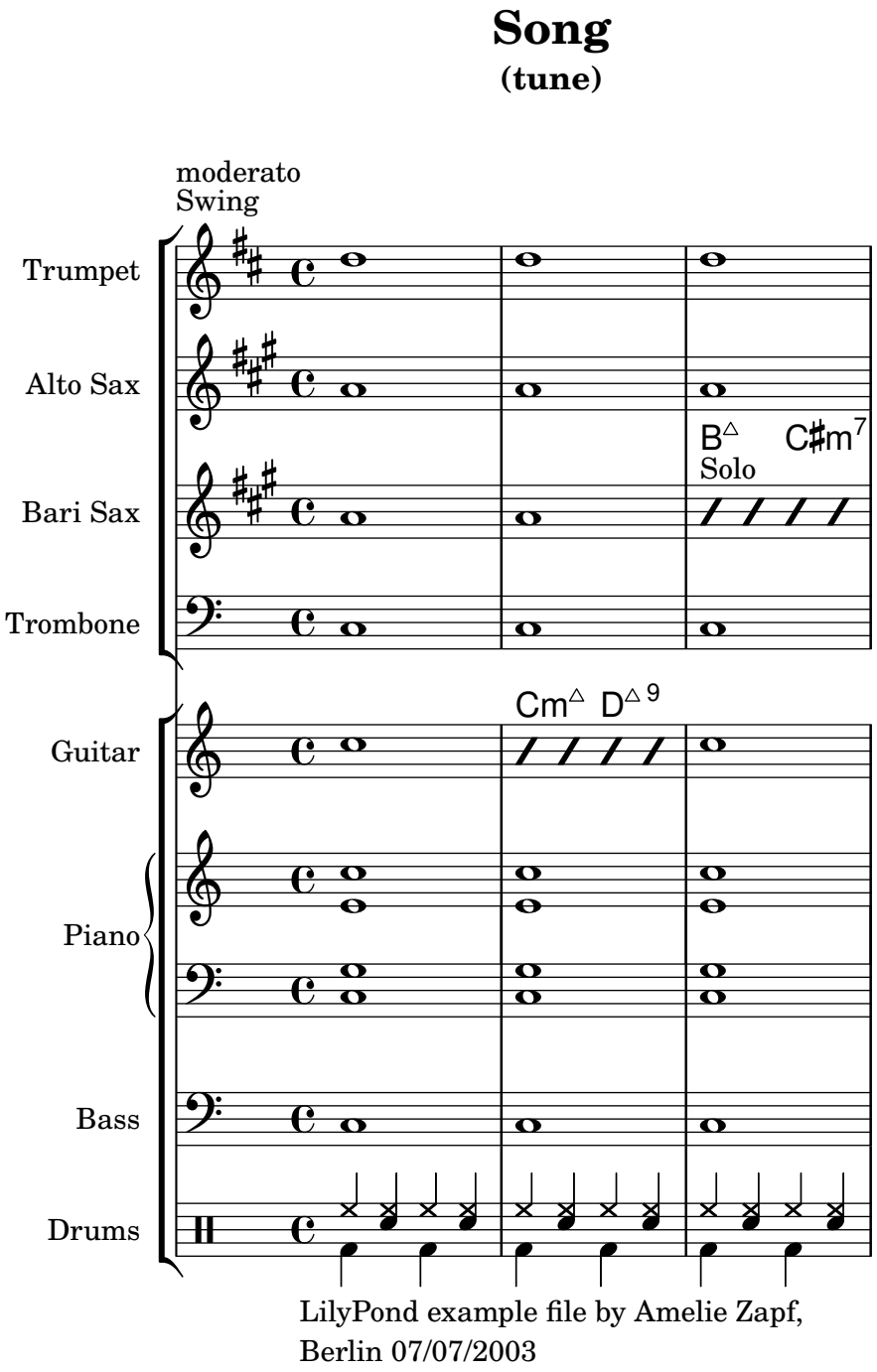
%%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%%%

\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 { }
  }
}
```
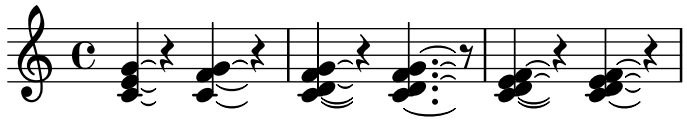


Liaison « Laissez vibrer »

Les indications de « laisser vibrer » ont une taille fixe. Leur formatage est accessible au travers de la propriété `tie-configuration`.

```
\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
}
```



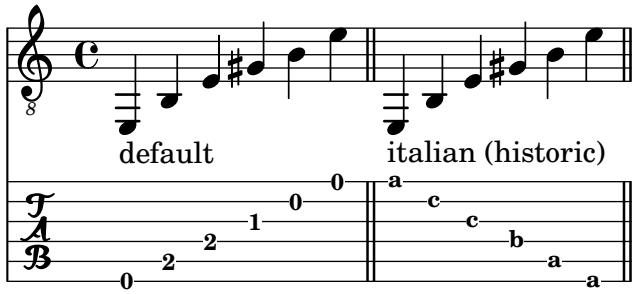
Impression de la corde aiguë d'un TabStaff en bas

Il est d'usage, en matière de tablature, d'imprimer la première corde en haut. Il est toutefois possible de la positionner en bas grâce à une désactivation de la propriété de contexte stringOneTopmost. Définir ce réglage au sein d'un bloc \layout l'appliquera à l'intégralité du contexte.

```
%\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
  }
>>
```

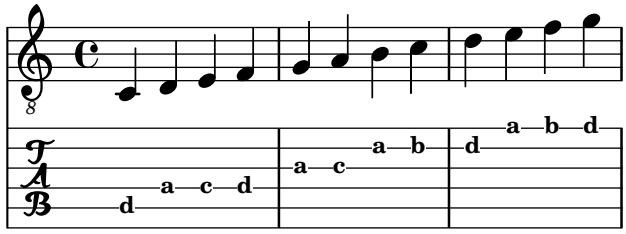


Tablature en lettres

Une tablature peut comporter des lettres en lieu et place des chiffres.

```
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
  }
>>
```



Harmoniques sur corde à vide en tablature

Voici comment obtenir des harmoniques sur corde à vide (harmoniques naturelles) dans une tablature.

```
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

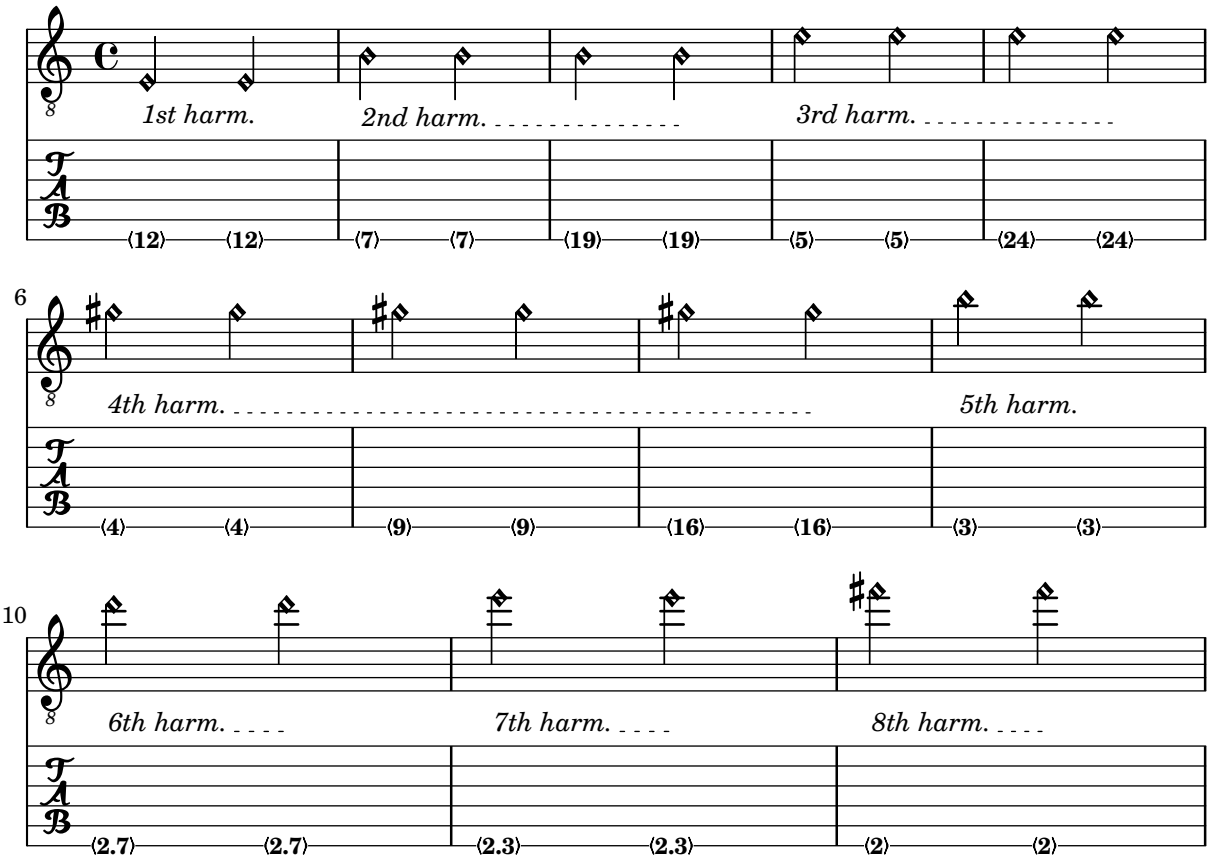
% sixth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "6th harm. "
\harmonicByFret 2.7 e,\6\startTextSpan
\harmonicByRatio #1/7 e,\6\stopTextSpan

% 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
      }
    }
  >>
}
```



Positionnement des doigtés main droite

Le positionnement des doigtés main droite, grâce à une propriété spécifique, peut se contrôler finement, comme l'indique l'exemple suivant.

```
#(define RH rightHandFinger)

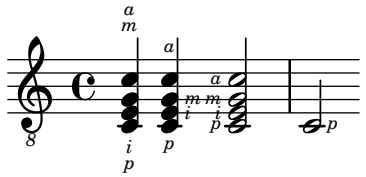
\relative c {
  \clef "treble_8"

  \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
}
```



Polyphonie en mode tablature

Une section polyphonique s'obtient dans un TabStaff de la même manière que dans une portée normale.

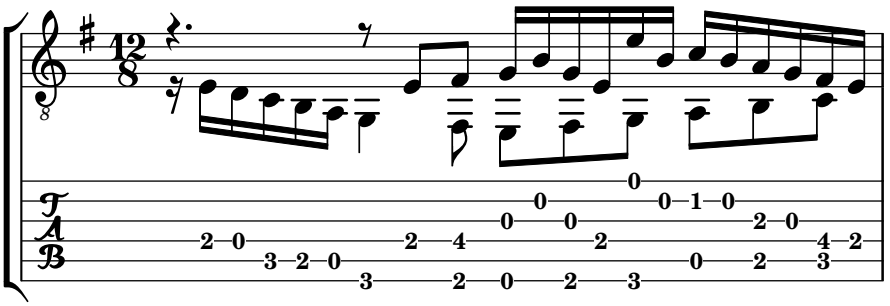
```
upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

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
```

```
    >>
  >>
}
```



Création de diagrammes de fret prédéfinis pour d'autres instruments

La liste des diagrammes standards prédéfinis pour la guitare peut être augmentée d'autres définitions spécifiques à d'autres instruments. Voici comment définir un nouvel accordage ainsi que quelques diagrammes prédéfinis pour le « cuatro vénézuélien ».

Cet exemple illustre aussi la manière d'ajouter des doigtés aux accords ; ils serviront de référence pour la boucle d'accord et seront indiqués dans les diagrammes et le `TabStaff`, mais pas dans la musique.

Ces diagrammes ne peuvent pas être transposés, dans la mesure où ils contiennent des informations sur les cordes. Ceci est amené à évoluer.

```
% 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
      }
    }

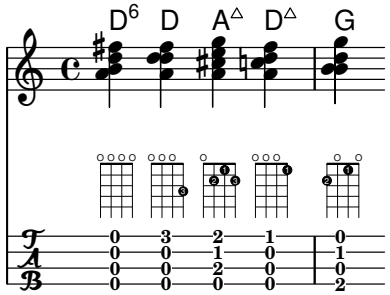
    \new FretBoards {
      \set Staff.stringTunings = #cuatroTuning
%      \override FretBoard
%        #'(fret-diagram-details string-count) = 4
      \override FretBoard.fret-diagram-details.finger-code = #'in-dot
      \primeros
    }

    \new TabStaff \relative c'' {
      \set TabStaff.stringTunings = #cuatroTuning
      \primeros
    }

  >>

  \layout {
```

```
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration =
        \musicLength 16
    }
  }
  \midi { }
}
```



Glissando et tablature

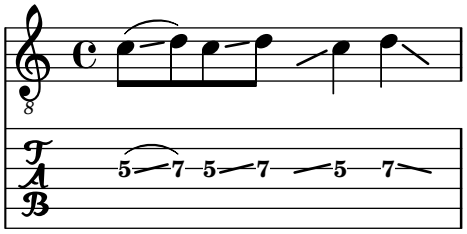
Un glissando s'indique dans un TabStaff tout comme dans un Staff.

```
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
    }
  }
```

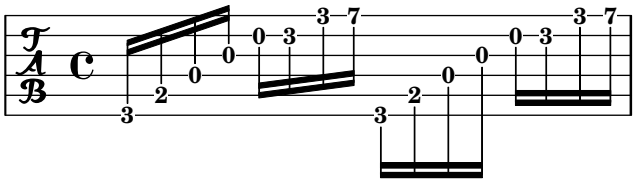
```
}
```



Hampes et ligatures en mode tablature

La direction des hampes se gère dans les tablatures tout comme en notation traditionnelle. Les ligatures peuvent être mises à l'horizontale comme le montre cet exemple.

```
\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
  }
}
```
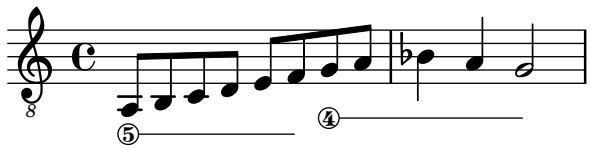


Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
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

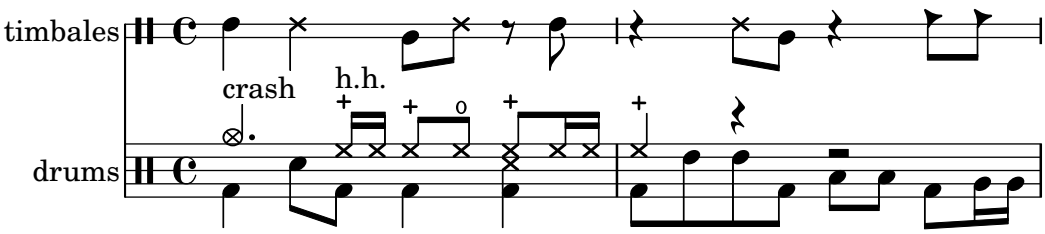
See also Section “Percussion” dans *Manuel de notation*.

Ajout de parties de batterie

Grâce à la puissance des outils préconfigurés tels que la fonction `\drummode` et le contexte `DrumStaff`, la saisie de parties pour percussions est extrêmement simplifiée : chaque composant d'une batterie trouve sa place sur une portée dédiée (avec une clef particulière) et les têtes de note sont spécifiques à chaque élément. Il est également possible d'affecter un symbole particulier à chaque élément, tout comme de restreindre le nombre de lignes de la portée.

```
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 }
}
```



Exemple de sonnaille et clochette

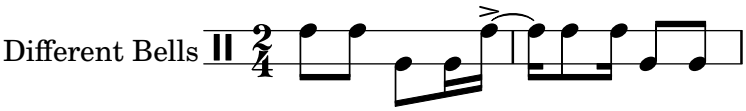
Voici deux différentes cloches, saisies « cb » (*cow bell* pour sonnaille) et « rb » (*ride bell* pour clochette).

```
#(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 |
}
```



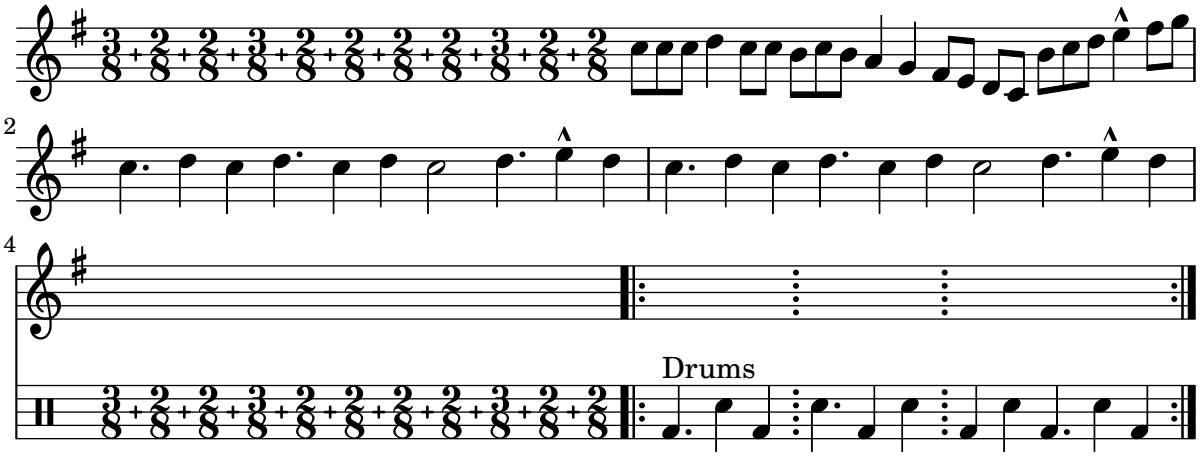
Indication personnalisée d'une polymétrie complexe

Bien que la métrique complexe ne soit pas l'élément primordial de cet exemple, elle permet d'indiquer la pulsation de cette pièce qui, par ailleurs, constitue le canevas d'une chanson traditionnelle des Balkans.

```
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
}
```



Exemple de woodblocks haut et bas

Deux woodblocks se saisissent « wbh » (woodblock haut) et « wbl » (woodblock bas). La hauteur des barres de mesure a été adaptée par une commande \override afin qu'elles ne soient pas trop courtes. La position des deux lignes de la portée doit aussi être explicitement définie.

```
% 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)))

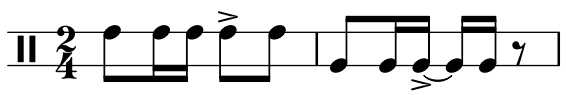
woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

  % 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 |
  }
}
```



Symboles de jazz

Bien que compliqué de prime abord, voici un canevas tout à fait indiqué pour les ensembles de jazz. Vous noterez que tous les instruments sont notés en ut (`\key c \major`), la tonalité de concert. Les notes seront automatiquement transposées dès lors qu'elles seront inscrites dans une section `\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 |
}

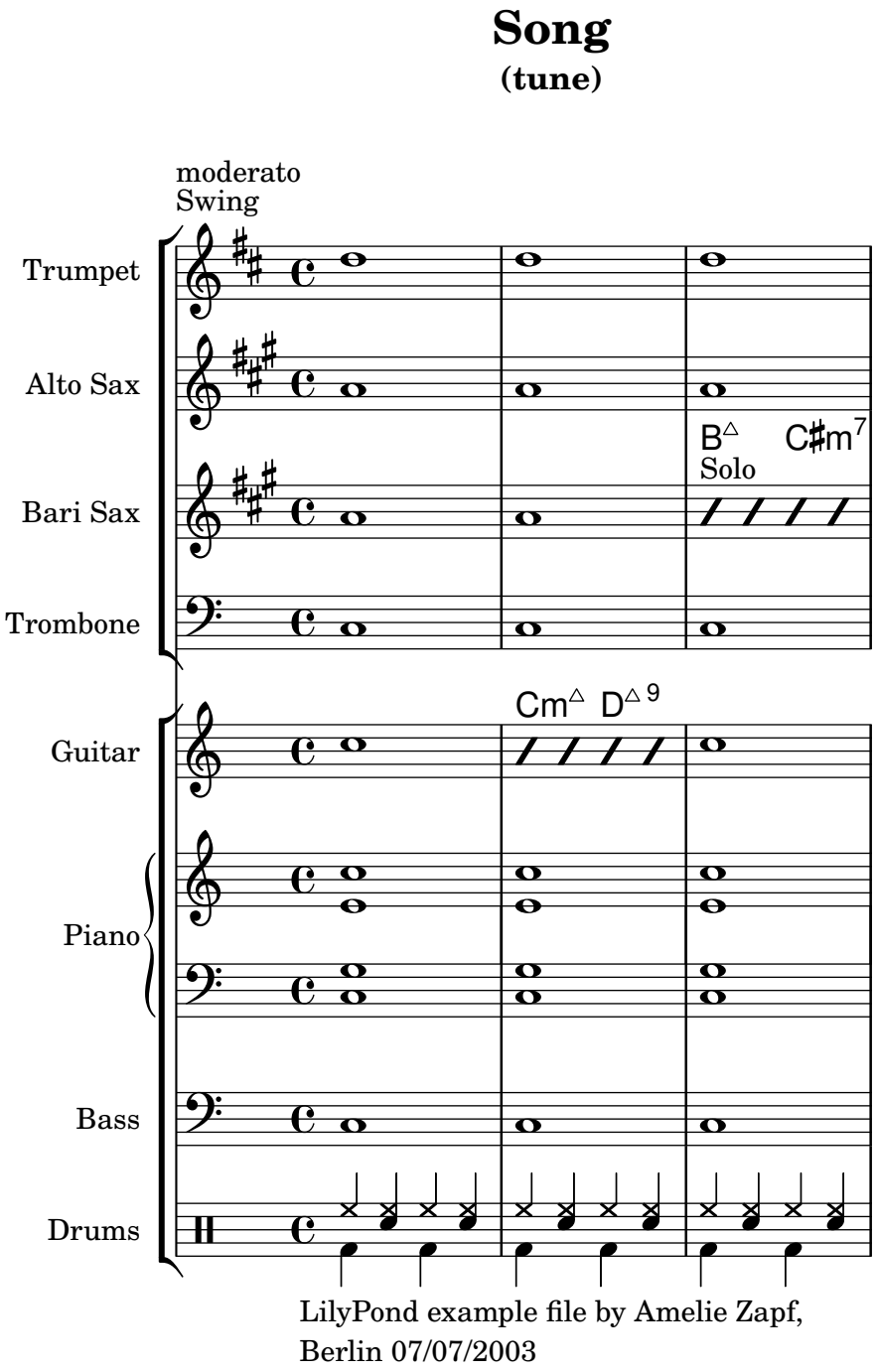
drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%%% 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 { }
  }
}
```



Baguettes pour percussion

La représentation graphique des instruments à percussion n'est pas prise en charge nativement. De tels symboles peuvent cependant être inclus, soit à l'aide de fichiers EPS indépendants, soit à l'aide d'un *markup* contenant le code PostScript correspondant comme dans cet exemple.

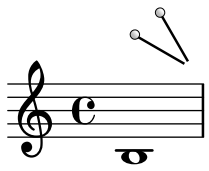
```
stick = \markup \with-dimensions #'(0.80 . 5.2) #'(0.85 . 5.2) {
  \postscript "
    0 6 translate
    0.8 -0.8 scale
    0 0 0 setrgbcolor
    [] 0 setdash
    1 setlinewidth
```

```
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
}
```



Exemple pour percussions

Ce court exemple est tiré de « L'histoire du Soldat » de Stravinsky.

```
#(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.
```

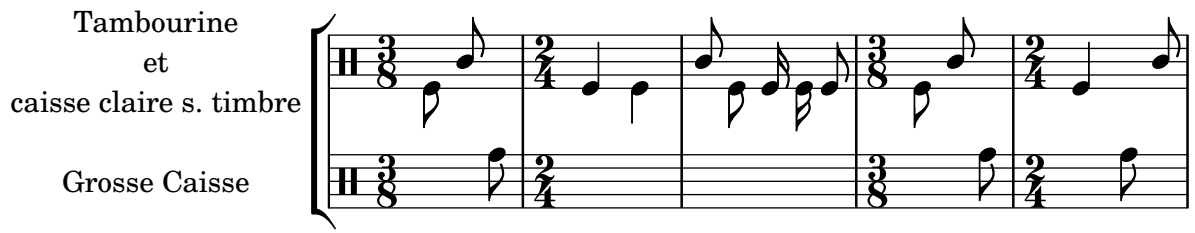
```
  \time 2/4 s2
}

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
  >>
}
```



Impression de musique aux métriques différentes

Bien qu'ayant des métriques bien différentes, les deux parties ci-dessous présentées restent synchrones.

Les barres de mesure ne peuvent plus être gérées au niveau du contexte `Score` ; le `Timing_translator` doit être déplacé du contexte `Score` au contexte `Staff` afin de permettre des barres de mesure individualisées.

Le `Bar_number_engraver` devra lui aussi être déplacé, puisqu'il dépend de propriétés attachées au `Timing_translator`, afin de numéroter les mesures. L'utilisation d'un bloc `\with` dans la portée concernée permettra un affichage des numéros de mesure.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}

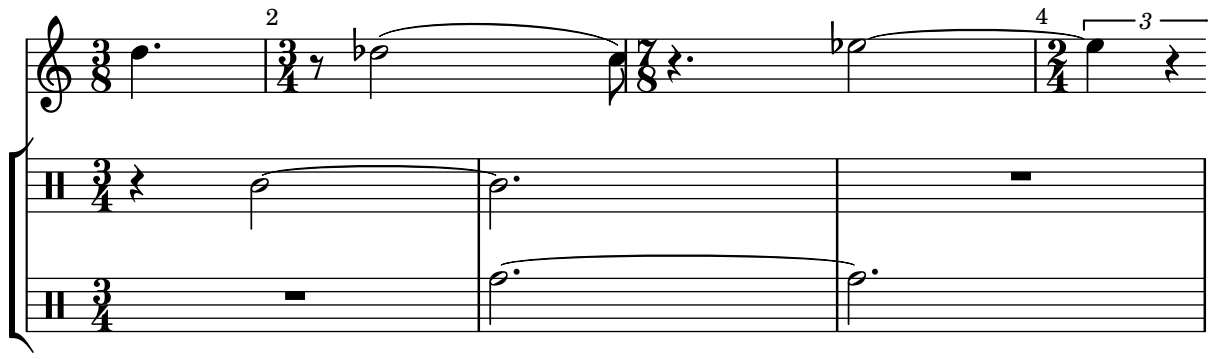
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

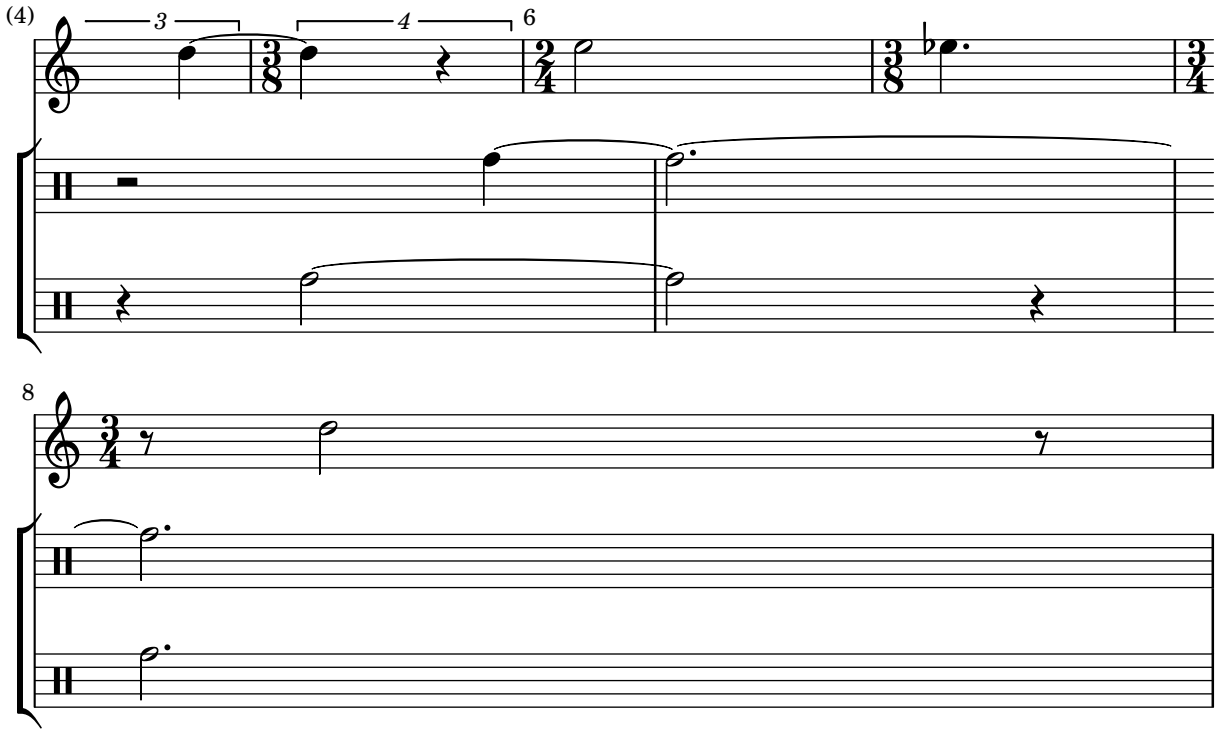
Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. 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
  >>
}
```





Exemple pour tam-tam

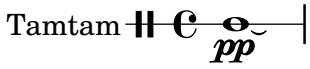
Un simple exemple pour tam-tam, saisi « tt ».

```
#(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 

Exemple pour tambourin

Voici une partie de tambourin, saisi par « tamb ».

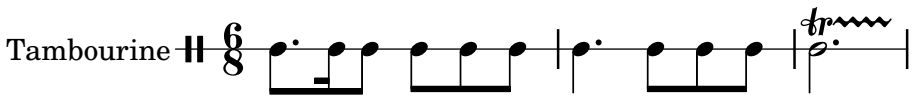
```
#(define mydrums '((tambourine default #f 0)))

\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

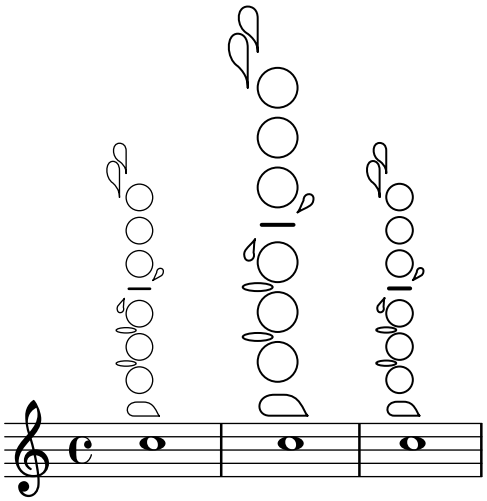
See also Section “Wind instruments” dans *Manuel de notation*.

Modification de la taille d’un diagramme pour bois

La taille et l’épaisseur des diagrammes de doigté pour bois est modifiable à souhait.

```
\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 #'()
}
```



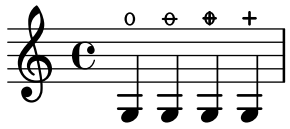
Symboles de doigtés pour instruments à vent

Des symboles spécifiques peuvent être obtenus en combinant les glyphes disponibles, ce qui est tout à fait indiqué en matière d’instrument à vent.

```
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
}
```
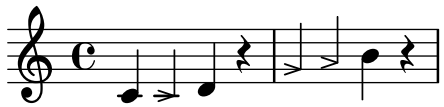


Coup de langue à la flûte

L'indication d'une technique particulière d'articulation, telle le « coup de langue » des flûtistes, s'obtient en remplaçant la tête de note par un glyphe approprié. Pour ce faire, on peut dessiner une tête de note en forme d'accent à l'aide d'un \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
}
```

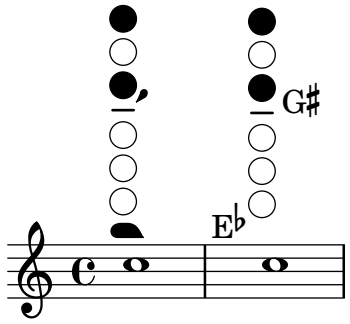


Ajout de texte à un diagramme de doigté

Dans certains cas, vous pouvez opter pour l'affichage textuel d'une clé située à côté d'un trou plutôt que sa représentation graphique.

```
\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)))
}
```



Doigtés pour flûte à bec

Cet exemple illustre la manière de créer et afficher des indications de doigté pour instrument à vent.

```
% 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

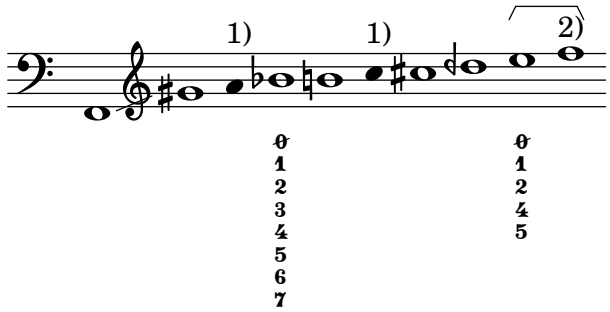
  c''4^\markup "1)"

  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
}
```
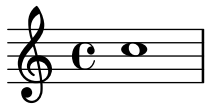


Liste des différents diagrammes de doigtés pour bois

Le code suivant permet d'obtenir une liste de toutes les possibilités en matière de doigtés pour bois, tels qu'ils sont définis dans le fichier `scm/define-woodwind-diagrams.scm`. Cette liste sera produite dans le fichier de journalisation, mais pas sous forme de musique. Pour un affichage en console, supprimez la partie `(current-error-port)` des commandes.

```
#(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}
```



Liste des diagrammes de doigtés pour bois

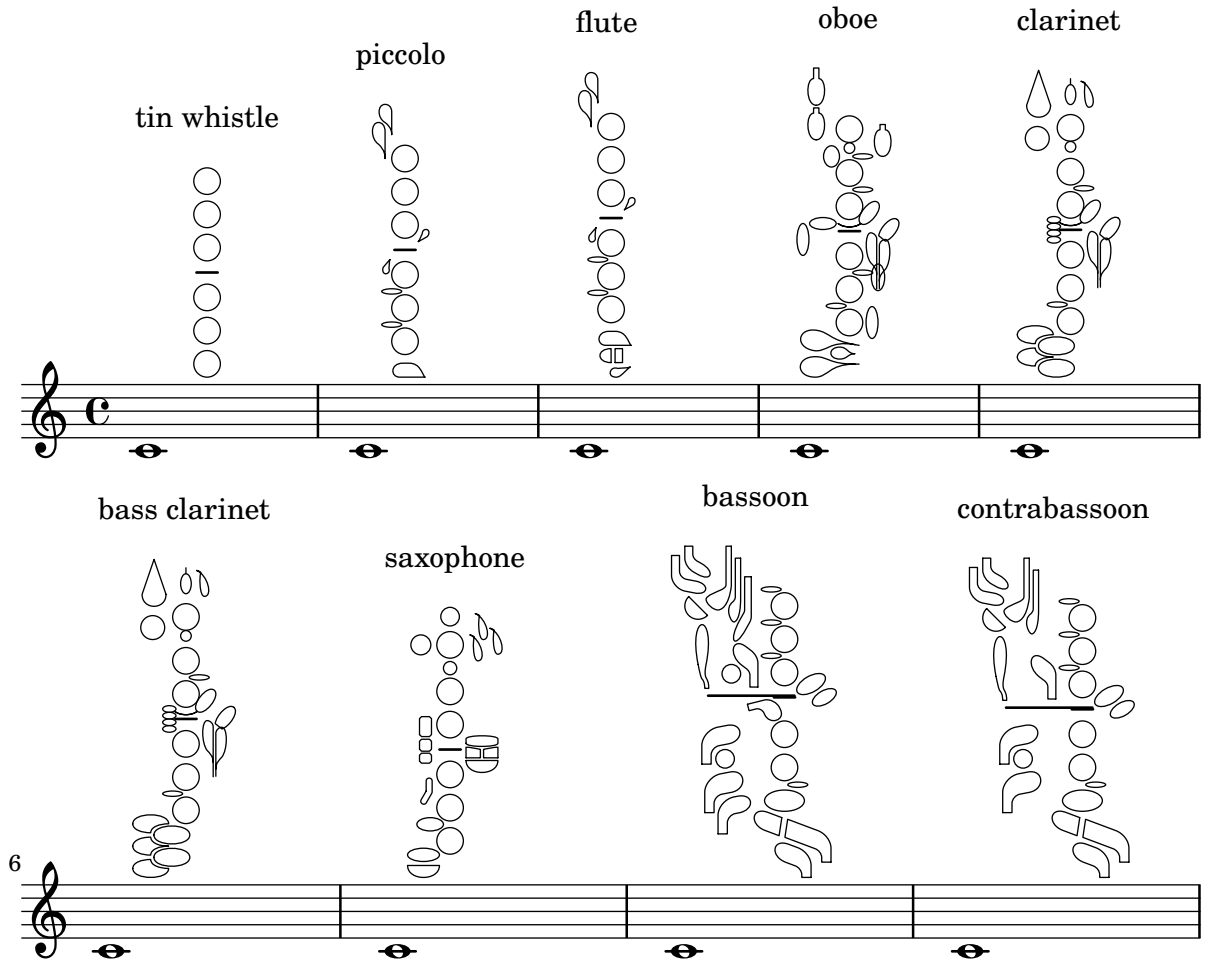
Voici les différents instruments à vent de la section des bois pour lesquels LilyPond peut, à ce jour, afficher des doigtés.

```
\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 Section “Chord notation” dans *Manuel de notation*.

Ajout d’une basse chiffrée au-dessus ou au-dessous des notes
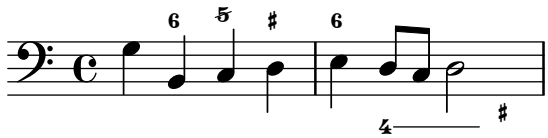
Une ligne de basse chiffrée peut se positionner au-dessus ou en dessous d’une partie de basse à l’aide des commandes `\bassFigureStaffAlignmentDown` et `\bassFigureStaffAlignmentUp`. Une modification temporaire du positionnement s’obtient en faisant précéder la commande d’un `\once`.

La commande `\bassFigureStaffAlignmentNeutral` réinitialise le positionnement des chiffrages à sa valeur par défaut.

```
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
  >>
}
```

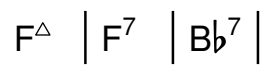


Chiffrages et barres de mesure

L’ajout du graveur `Bar_engraver` à un contexte `ChordNames` permet d’imprimer les barres de mesure entre les chiffrages.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```



Ajustement des glyphes d'altération en basse chifrée

En basse chiffrée, des glyphes spécifiques sont utilisés par défaut pour les `6\\`, `7\\` et `9\\`. D'autres glyphes ont été conçus pour les `2\+`, `4\+` et `5\+` ; ils sont utilisés par défaut dès lors qu'un signe plus est ajouté au chiffre.

Pour modifier ce comportement, il faut passer une liste associative à `figuredBassPlusStrokedAlist` et basculer le glyphe en question sur `#f` (ou ne pas le mentionner).

```
#(set-global-staff-size 26)

\figures {
  \set figuredBassPlusDirection = #RIGHT
  <6\\> <7\\> <9\\> r
  <2\+> <4\+> <5\+> r

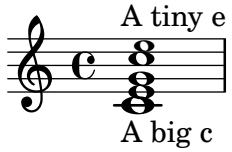
  \set figuredBassPlusStrokedAlist =
    #'((2 . "figbass.twoplus")
       ;; (4 . "figbass.fourplus")
       ;; (5 . "figbass.fiveplus")
       (6 . "figbass.sixstroked")
       ;; (7 . "figbass.sevenstroked")
       ;; (9 . "figbass.ninestroked")
      )
  <6\\> <7\\> <9\\> r
  <2\+> <4\+> <5\+> r
}
```

Modification de la taille d'une note particulière d'un accord

La commande `\tweak` permet de modifier la propriété `font-size` d'une note particulière d'un accord.

Il s'agit de placer, à l'intérieur même de l'accord (dans la construction `< >`) et avant la note considérée, l'instruction `\tweak` suivie de `font-size` et de définir la taille voulue – comme `#-2` pour une petite tête.

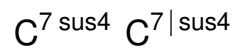
```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Personnalisation du séparateur d'accords

Le séparateur de termes d'un chiffrage peut adopter n'importe quelle forme à l'aide d'un *markup*.

```
\chords {
  c:7sus4
  \set chordNameSeparator = \markup { \typewriter | }
  c:7sus4
}
```



Emplacement des altération en basse continue

On peut choisir d'imprimer les altérations et signes plus aussi bien avant qu'après les chiffres, en réglant les propriétés `figuredBassAlterationDirection` et `figuredBassPlusDirection`.

Lorsqu'un chiffre est suivi d'un signe plus, certains chiffrages utiliseront des glyphes spécifiques.

```
#(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
}
```



Modèles de chiffrage d'accords

La propriété `chordNameExceptions` enregistre une liste de chiffrages personnalisés afin de traiter des cas non prévus ou incorrectement gérés.

Par défaut, LilyPond utilise le jeu de chiffrages issus de l'ouvrage de Klaus Ignatzek « Die Jazzmethode für Klavier 1 » ; l'algorithme de conversion des accords en chiffrages est inclus dans le fichier `scm/chord-ignatzek-names.scm`. Des règles additionnelles pour gérer des chiffrages exceptionnels sont enregistrées en tant que variables `ignatzekExceptions` dans le fichier `ly/chord-modifiers-init.ly`.

Le code ci-dessous modifie ces exceptions en trois étapes.

1. Saisie de musique en accords avec *markup* associé. Par convention, la base – autrement dit la note la plus basse – de chaque accord aura une hauteur de do (`c` pour les anglophones).
2. Appel de la fonction Scheme `sequential-music-to-chord-exceptions` pour créer une liste de nouvelles exceptions, puis l'ajouter à celles déjà répertoriées. Dans la mesure où `ignatzekExceptions` est défini avec le second paramètre de cette fonction fixé à `#t`, afin d'ignorer la base de chaque accord, il faut faire de même.
3. Enregistrement de la nouvelle liste d'exceptions.

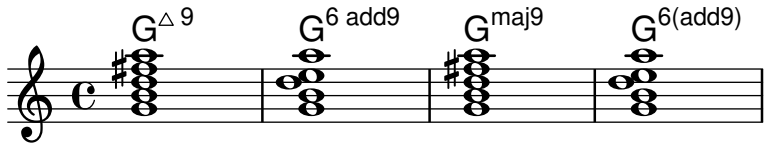
```
% Step 1: Define music with chords and markup for maj9 and 6(add9).
chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

% 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
>>

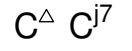
\layout {
  line-width = 10\cm
  ragged-right = ##f
}
```



Chiffrage d'un maj7

La représentation d'un accord de septième majeure se gère par la propriété de contexte `majorSevenSymbol`.

```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```



Noms d'accord alternatifs

Les noms d'accord sont générés à partir d'une liste de hauteurs. Les fonctions qui construisent ces noms peuvent être personnalisées.

Voici les accords suivant la nomenclature d'Ignatzek (pp. 17-18, 1995) qu'utilise LilyPond par défaut depuis la version 1.7.20, comparée à une alternative jazz et à la notation d'Harald Banter (1987). Une police plus petite est utilisée pour cette dernière dans la mesure où elle tend à être plus verbeuse.

Ceci est un reflet du mécanisme utilisé à l'origine par LilyPond, jusque dans les version 1.7. N'ayant pas fait l'objet d'une maintenance suivie, certaines fonctionnalités ont été perdues – notamment au niveau des listes d'exception – et des bogues ont fait leur apparition.

```
%%%% 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
    }
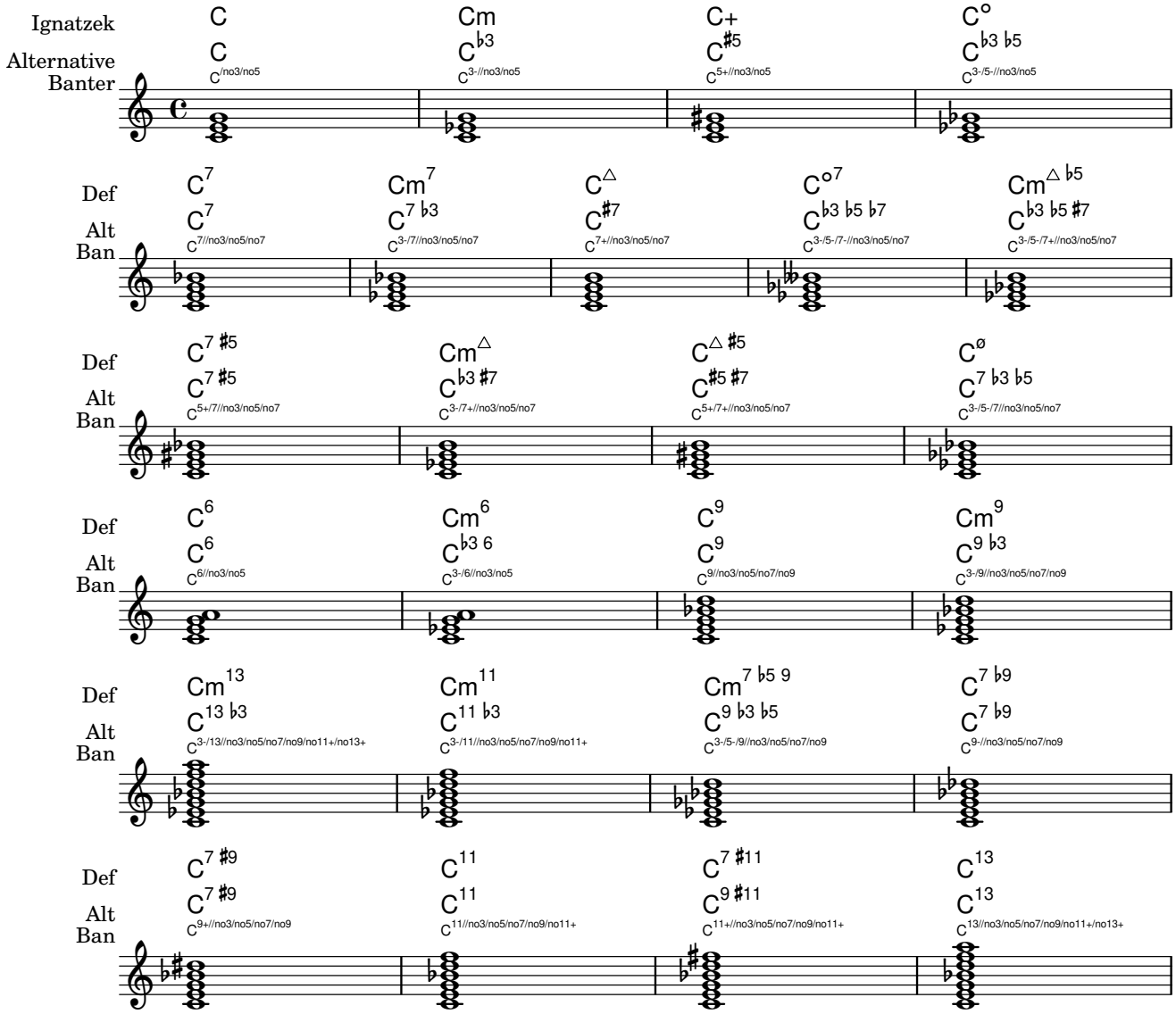
    \new ChordNames {
      \set chordNameFunction = #jazz-chordnames
      \set majorSevenSymbol = \whiteTriangleMarkup
      \set chordNameSeparator = "/"
      \set chordNameExceptionsFull = \efullJazzAlt
      \set chordNameExceptionsPartial = \epartialJazzAlt
      \set instrumentName = "Alternative"
      \set shortInstrumentName = "Alt"
      \chs
    }

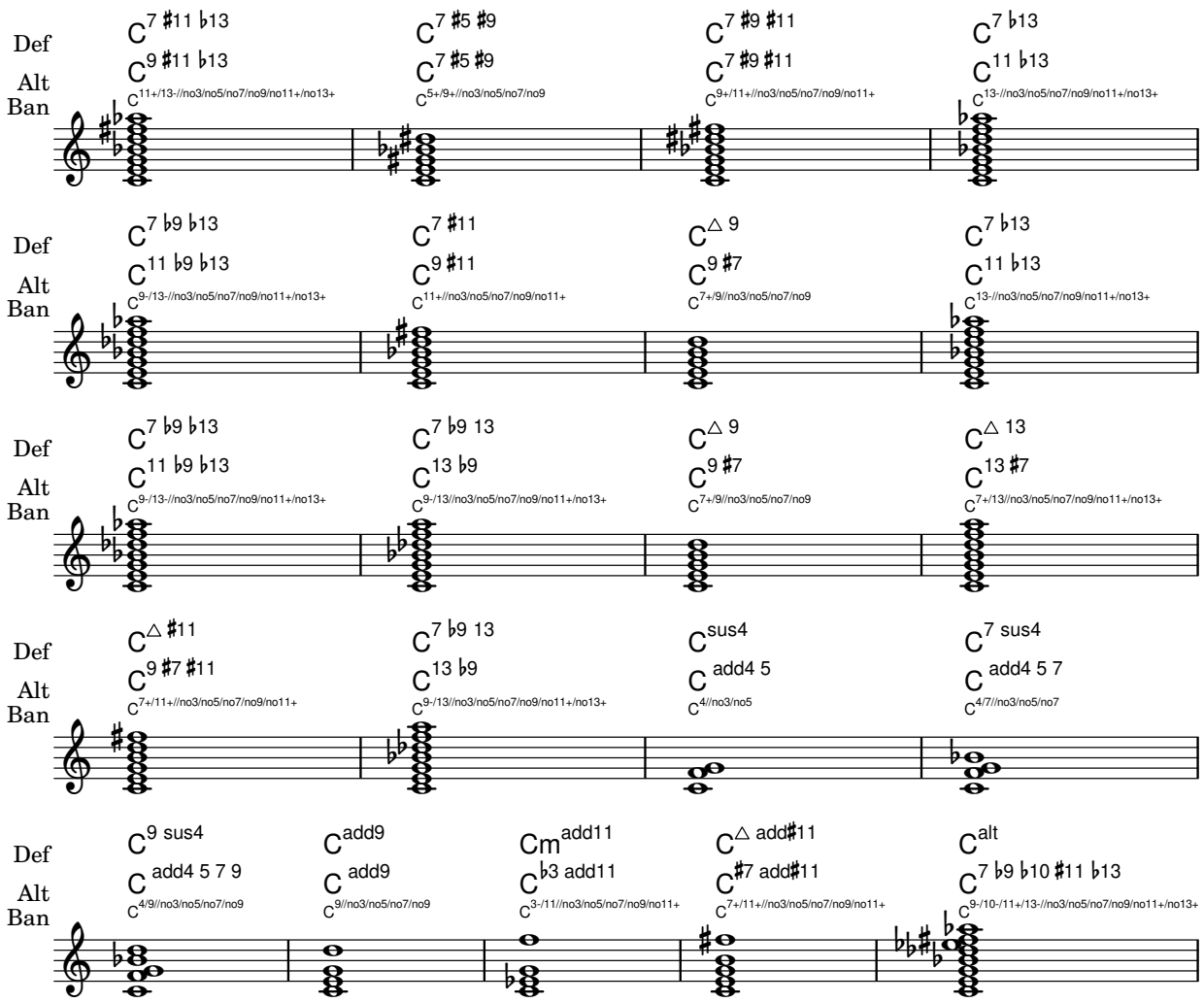
    %% This is the Banter (1987) style.  It gives exceedingly
    %% verbose (wide) names, making the output file take up to 4 pages.

    \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"
    }
  }
}
```





Accords avec doigté espacé pour FretBoards et TabVoice

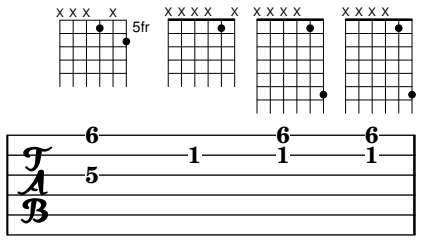
Il peut arriver que le doigté d'un accord soit assez étendu. Sauf mention contraire, la valeur de la propriété de contexte maximumFretStretch est cependant fixée à 4, ce qui peut générer un avertissement « Pas de corde pour la hauteur… » et la note est omise. On peut régler maximumFretStretch sur une valeur appropriée ou assigner explicitement leur numéro de corde à toutes les notes d'un accord afin de palier ce problème.

```
% 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
>>
```

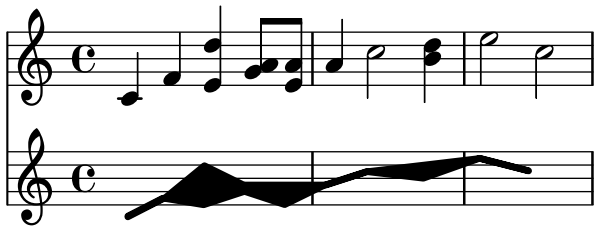


Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

```
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
>>
```



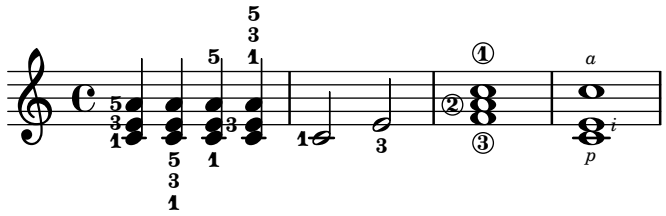
Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise à l'aide de la propriété `fingeringOrientation`. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `<...>`, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue à l'aide des propriétés `stringNumberOrientation` et `strokeFingerOrientation`.

Ces propriétés peuvent prendre une liste de trois valeurs. Elles contrôlent si les informations doivent être placées au-dessus (en présence d'un `up`) ou au-dessous (en présence d'un `down`), à gauche (en présence d'un `left` ou à droite (en présence d'un `right`). Par contre, si un positionnement n'est pas mentionné, aucun doigté n'y figurera. LilyPond tient compte de ces contraintes et trouvera le meilleur emplacement pour le doigté des notes de l'accord qui suit. Bien entendu, `left` et `right` sont exclusifs l'un de l'autre – les doigtés ne peuvent se placer que d'un seul côté.

```
\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 >
}
```



Accord distribué et problème de hampe – solution

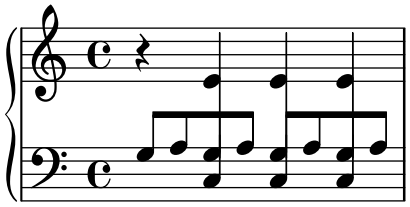
Il est parfois préférable d'utiliser les hampes de l'autre portée pour créer des accords distribués, afin de tricher avec le détecteur de collision des ligatures de LilyPond. Dans l'exemple suivant, le fait de partir des hampes de la portée inférieure aurait rendu nécessaire l'adaptation du détecteur de collision des ligatures, par une clause

```
\override Staff.Beam.collision-voice-only = ##t
```

afin que LilyPond ne déplace pas les ligatures.

```
\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
  }
>>
```



Personnalisation du style de grille harmonique

Il est possible de personnaliser la division des cases à l'aide des propiétés de `ChordSquare` `measure-division-lines-alist` et `measure-division-chord-placement-alist`, qui sont toutes deux des listes associatives. Leurs clés sont des divisions de la mesure, autrement dit des listes de fractions de la mesure que chaque accord, silence ou saut représente. Plus précisément, cette liste de divisions de la mesure est constituée de nombres positifs exacts dont l'addition fait 1, comme par exemple `'(1/2 1/4 1/4)`. L'exigence d'exactitude signifie par exemple que `1/2` est valide, contrairement à `0.5`.

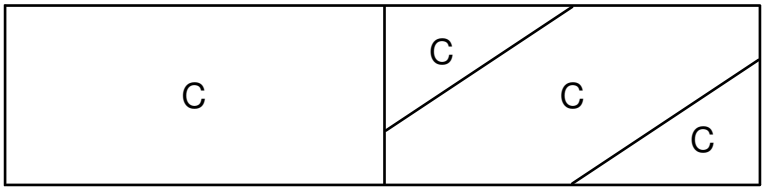
Les valeurs de `measure-division-lines-alist` sont des listes de lignes, représentes sous la forme `(x1 y1 x2 y2)`. La ligne débute au point `(x1 . y1)` et se temine en `(x2 . y2)`. Les coordonnées sont comprises dans l'intervalle $[-1, 1]$ relativement à l'étendue de la case.

Les valeurs de `measure-division-chord-placement-alist` sont des listes de paires `(x . y)` indiquant le positionnement des accords respectifs.

L'exemple ci-dessous illustre le cas particulier d'une grille dont le style spécifie le découpage des mesures en trois parts égales.

```
\paper {
  line-width = 10\cm
  ragged-right = ##f
}

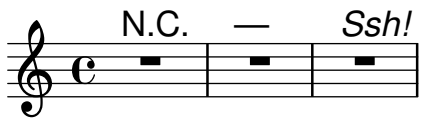
\new ChordGrid \with {
  \override ChordSquare.measure-division-lines-alist =
    #'(((1) . ())
       ((1/3 1/3 1/3) . ((-1 -0.4 0 1) (0 -1 1 0.4))))
  \override ChordSquare.measure-division-chord-placement-alist =
    #'(((1) . ((0 . 0)))
       ((1/3 1/3 1/3) . ((-0.7 . 0.5) (0 . 0) (0.7 . -0.5))))
}
\chordmode {
  \time 3/4
  c2.
  c4 c4 c4
}
```



Personnalisation du symbole *no-chord*

Par défaut, les silences apparaissant dans un contexte `ChordNames` déclenchent l'impression d'un texte « N.C. ». Ce *markup* peut être personnalisé grâce à la propriété de contexte `noChordSymbol`.

```
<<
  \chords {
    R1
    \set noChordSymbol = "---"
    R1
    \set noChordSymbol = \markup \italic "Ssh!"
    R1
  }
  {
    R1*3
  }
>>
```
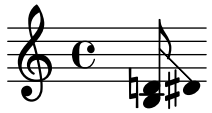


Impression d'accords complexes

Voici comment obtenir l'impression d'un accord au sein duquel une même note est jouée deux fois avec des altérations différentes.

```
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
}
```

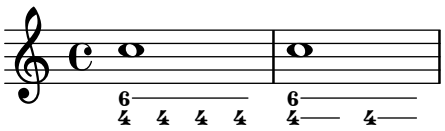


Interruption manuelle des prolongations de certains chiffrages

La basse chiffrée utilise fréquemment des lignes pour indiquer la prolongation jusqu'à un certain point. LilyPond tente alors de placer le plus de prolongateurs possible. L'interruption d'un prolongateur particulier s'obtient en affectant d'un `\!` le chiffre qui doit être réimprimé.

```
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
>>
```



Impression de noms d'accords identiques à la basse différente par une oblique et nouvelle basse

Voici comment imprimer une suite d'accords nommés (objets ChordNames) dont seule la basse diffère, indiquée par une oblique et cette nouvelle basse, à l'aide du graveur Scheme ici défini. Son comportement peut se gérer finement à l'aide de la propriété de contexte chordChanges.

```
#(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

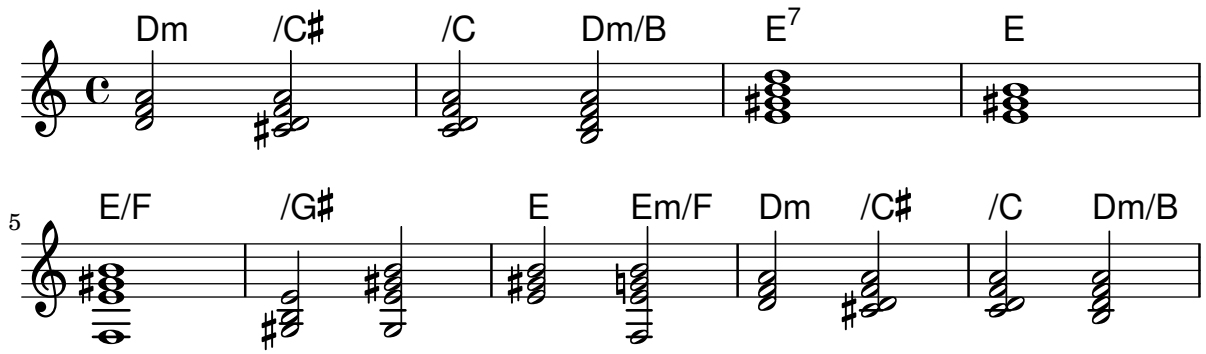
  \set chordChanges = ##t
  d2:m d:m/cis

  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
>>
```

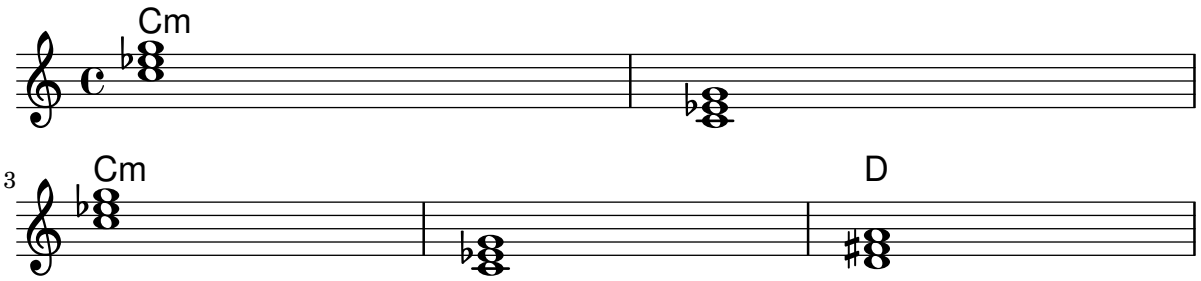


Impression des accords si changement

Tout accord saisi est par défaut imprimé. Ce comportement est amendable : vous pouvez faire ressortir les chiffrages d'accords s'ils ne sont imprimés qu'aux changements d'accord ou en début de ligne.

```
harmonies = \chordmode {
  c'1:m c:m \break
  c'1:m c:m d
}

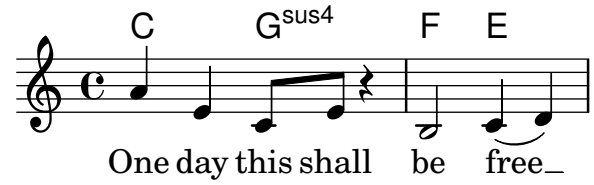
<<
  \new ChordNames {
    \set chordChanges = ##t
    \harmonies
  }
  \new Staff {
    \harmonies
  }
>>
```



Chanson simple

Assembler des noms d'accords, une mélodie et des paroles permet d'obtenir la partition d'une chanson.

```
<<
  \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 __ }
>>
```



Paroles, musique et accords

Ce canevas comporte tous les éléments d'une chanson : la mélodie, les paroles, les accords.

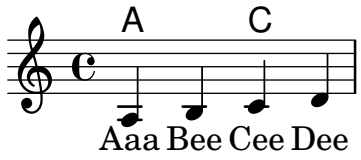
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  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 { }
}
```



Paroles, musique, accords et diagrammes de fret

Ce canevas comporte, en plus de la mélodie, des paroles et des accords, les diagrammes de fret.

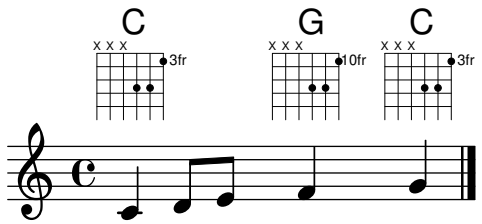
```
verseI = \lyricmode {
  \stanza "1."
  This is the first verse
}

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.

Mélodie simple et accords

Vous avez besoin de la partition d'une mélodie avec les accords ? N'allez pas plus loin !
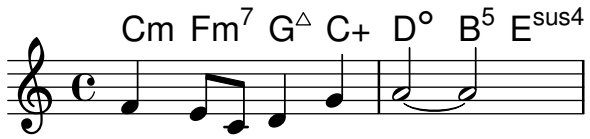
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  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
```

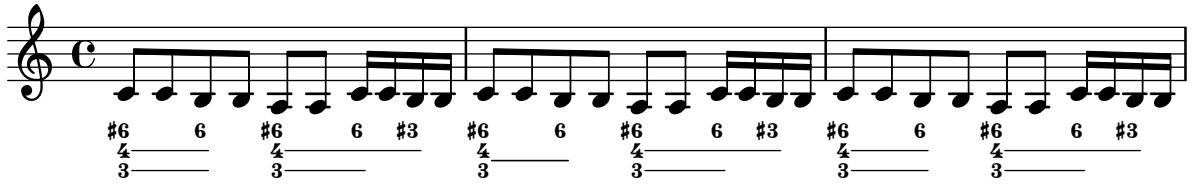
```
  >>
  \layout{ }
  \midi { }
}
```



Prolongateur commun de basse figurée

L'activation de la propriété `useBassFigureExtenders` permet d'afficher des lignes de prolongation pour les chiffres qui se répètent. Deux chiffres prolongés sur la même durée se verront affublés d'un unique prolongateur, verticalement centré entre eux, dès lors que la propriété `figuredBassCenterContinuations` aura elle aussi été activée.

```
<<
  \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
  }
>>
```

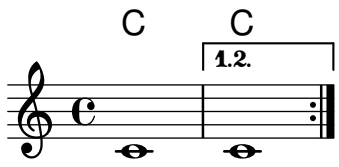


Crochet de reprise sous les chiffrages d'accord

L'ajout du `Volta_engraver` à la bonne portée permet d'imprimer les crochets de reprise entre les chiffrages et la portée.

```
\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 Section “Contemporary music” dans *Manuel de notation*.

Débordement de ligature

En combinant `stemLeftBeamCount`, `stemRightBeamCount` et des paires de `[]` attachées à des notes isolées, vous pourrez obtenir des crochets rectilignes aux allures de ligatures.

Pour des crochets rectilignes à droite sur des notes isolées, il suffit d'ajouter une paire d'indicateurs de ligature `[]` et de déterminer `stemLeftBeamCount` à zéro, comme dans l'exemple 1.

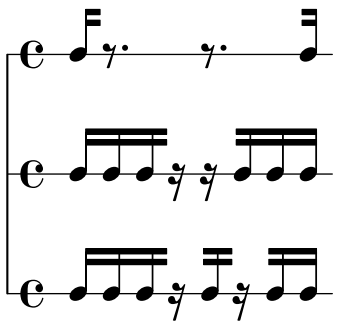
Pour des crochets rectiligne à gauche, c'est `stemRightBeamCount` qu'il faudra déterminer (exemple 2).

Pour que les barres de ligature débordent sur la droite, `stemRightBeamCount` doit avoir une valeur positive ; pour un débordement à gauche, c'est sur `stemLeftBeamCount` qu'il faut jouer. Tout ceci est illustré par l'exemple 3.

Il est parfois judicieux, lorsqu'une note est encadrée de silences, de l'affubler de crochets rectilignes de part et d'autre. L'exemple 4 montre qu'il suffit d'adjoindre à cette note un `[]`.

Notez bien que `\set stemLeftBeamCount` sera toujours synonyme de `\once \set`. Autrement dit, la détermination des ligatures n'est pas « permanente » ; c'est la raison pour laquelle les crochets du `c'16[]` isolé du dernier exemple n'ont rien à voir avec le `\set` indiqué deux notes auparavant.

```
\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
    }
  >>
}
```



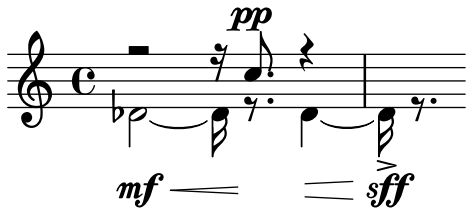
Soufflet de crescendo partiellement interrompu

Une portion d'un soufflet de crescendo peut être rendue invisible. Il suffit pour cela de dessiner un rectangle par dessus ce tronçon, ce qui aura pour effet de le rendre invisible. Ce rectangle est défini en tant que *markup* textuel.

La commande de *markup* `with-dimensions` indique à LilyPond de ne prendre en considération que l'extrémité inférieure du rectangle lors de son positionnement par rapport au soufflet. L'ajustement de la propriété `staff-padding` permet d'éviter au rectangle de venir s'intercaler entre le soufflet et la portée.

Le soufflet doit se trouver à un niveau inférieur à celui du *markup* afin que le dessin du rectangle puisse effectivement le recouvrir.

```
\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.
    }
  >>
}
```



Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations`

Polymétrie sans alignement des mesures

La prise en charge de contextes aux métriques indépendantes demande de supprimer le `Timing_translator` du contexte `Score`, conjointement à la création d'un contexte `TimingStaffGroup` disposant du `Timing_translator`. Ceci fera de `Timing` un alias de `TimingStaffGroup`, permettant ainsi aux commandes `\time` de s'appliquer au sein du `TimingStaffGroup` où elles apparaissent.

Contrairement à la commande native de LilyPond `\enablePerStaffTiming`, cette approche requiert la création explicite de contextes `TimingStaffGroup` mais, par contre, autorise la création de multiples contextes `Staff` qui suivront de concert la mesure définie dans le `TimingStaffGroup` qui les englobe.

Échelonnement local des métriques

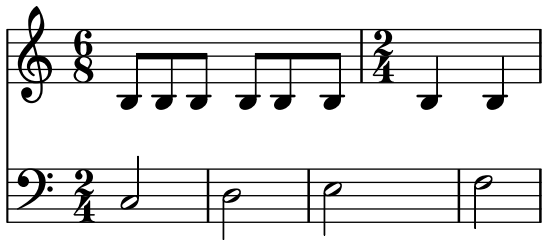
La commande `\time`, qui ne peut être échelonné, établit une mesure de la longueur désirée dans `Timing`, autrement dit dans `TimingStaffGroup`. Dans le code ci-dessous, toutes les portées incluses dans un `TimingStaffGroup` utilisent une métrique échelonnée, de sorte que, peu importe la métrique, elle tiendra dans la longueur de mesure désirée. Si l'un des contextes inclus n'avait pas de métrique échelonnée, le choix de la métrique devant être fournie à `Timing` aurait toute son importance.

L'utilisation de la commande `\polymetric \time` permet de définir les propriétés d'échelonnement de la métrique dans le contexte `Timing`, et l'utilisation de `\scaleDurations` échelonnera le mètre et les notes afin qu'elles tiennent dans la mesure.

```
\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
    }
  >>
>>
```

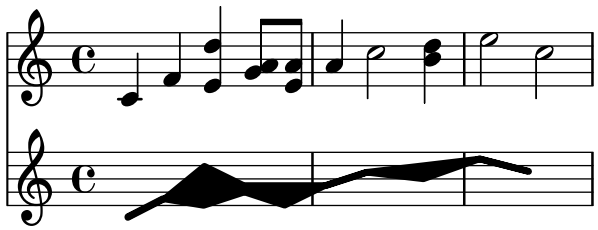


Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

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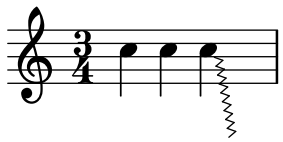
<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Glissando contemporain

De nos jours, il peut arriver que la note d'arrivée d'un glissando soit absente de la partition. Pour ce faire, il vous faudra utiliser une cadence et « masquer » la note d'arrivée.

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



Liaison de prolongation aplatie

La fonction `flared-tie` ici présentée permet de dessiner une liaison de prolongation constituée de lignes droites. Elle est sensée remplacer la fonction par défaut de dessin des liaisons de tenue – elle remplace l'argument `stencil` de l'objet `Tie`.

L'argument de cette fonction `flared-tie` est une liste de paires de coordonnées qui spécifient des points additionnels entre l'entame et la terminaison afin de déterminer l'étendue des lignes de la liaison. L'entame et la terminaison sont identiques à ceux de la liaison originelle. La valeur des abscisses et ordonnées sont des multiples de largeur et de la hauteur de la boîte englobante de la liaison originelle (tout en tenant compte de son positionnement) ; par voie de conséquence, le premier point a pour coordonnée (0,0), et le dernier point (1,0).

La fonction `flare-tie` définit un raccourci pour l'obtention d'une liaison aplatie. Des ajustements plus poussés sont possibles, en apportant des dérogations à `Tie.details.height-limit` ou à l'aide de la fonction `\shape`. La définition personnalisée peut aussi se modifier à la volée.

```
#(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

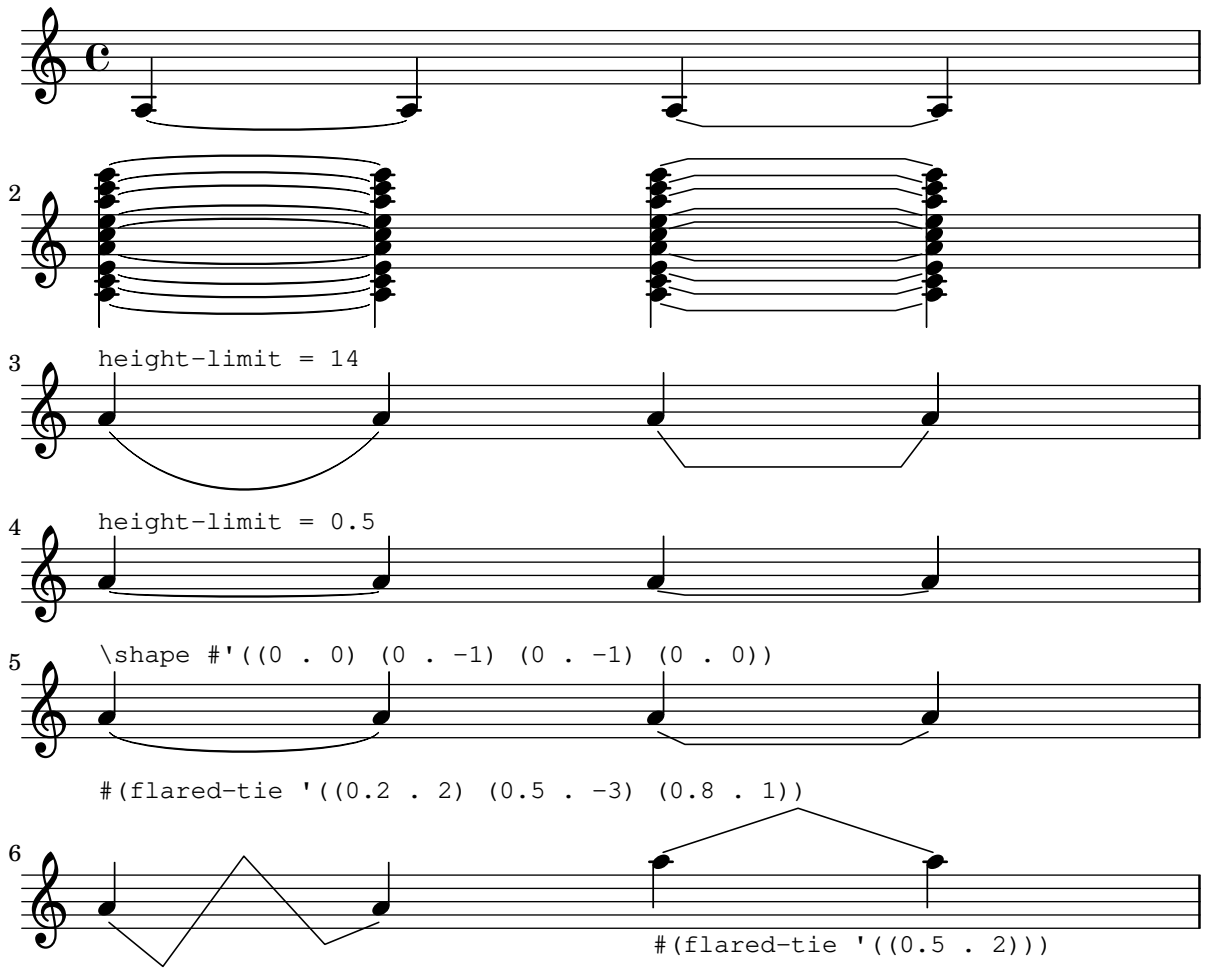
  <>^\markup \small \typewriter "height-limit = 0.5"
  \override Tie.details.height-limit = 0.5
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  \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
}
```
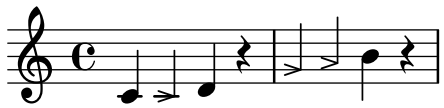


Coup de langue à la flûte

L'indication d'une technique particulière d'articulation, telle le « coup de langue » des flûtistes, s'obtient en remplaçant la tête de note par un glyphe approprié. Pour ce faire, on peut dessiner une tête de note en forme d'accent à l'aide d'un \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
}
```



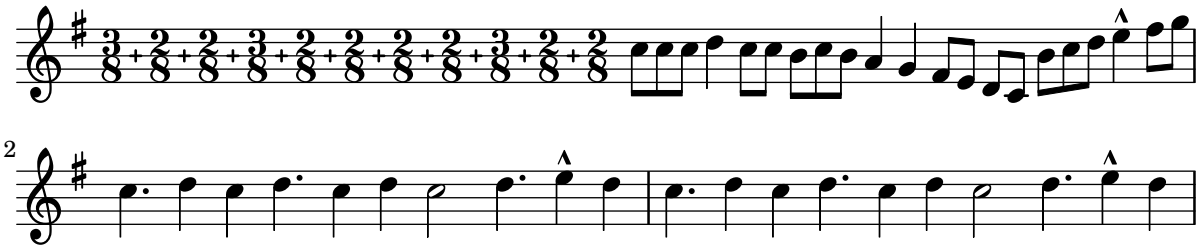
Indication personnalisée d'une polymétrie complexe

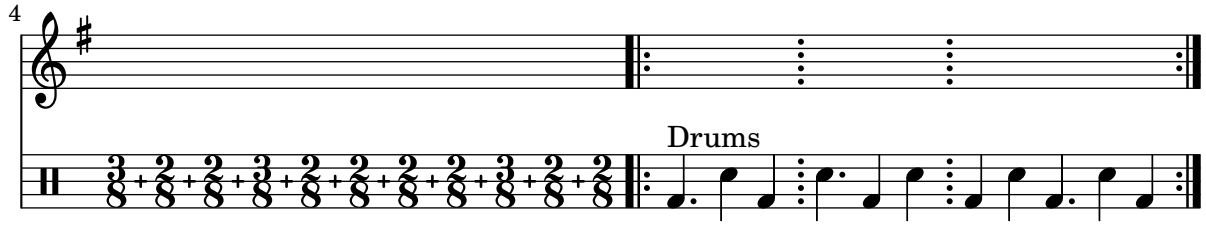
Bien que la métrique complexe ne soit pas l'élément primordial de cet exemple, elle permet d'indiquer la pulsation de cette pièce qui, par ailleurs, constitue le canevas d'une chanson traditionnelle des Balkans.

```
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
}
```
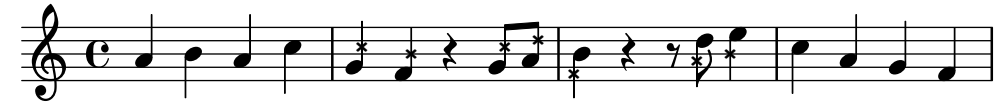




Adjonction d'une croix sur la hampe des notes d'un fragment parlé

Voici comment ajouter une croix aux hampes. Le début du fragment parlé est stipulé par une commande \speakOn, et la fin par une commande \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
  }
}
```



Armures inhabituelles

La commande \key détermine la propriété keyAlterations d'un contexte Staff.

Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété. Il s'agit en l'occurrence de définir une liste :

```
\set Staff.keyAlterations =
  #`(((octave . pas) . altération) ((octave . pas) . altération) ...)
```
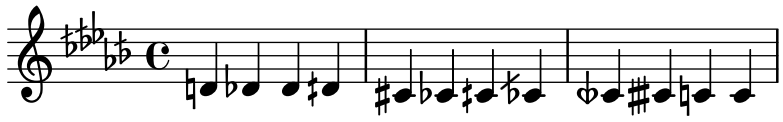
dans laquelle, et pour chaque élément, *octave* spécifie l'octave (0 pour celle allant du do médium au si supérieur), *pas* la note dans cette octave (0 pour do et 6 pour si), et *altération* sera ,SHARP ou ,FLAT ou ,DOUBLE-SHARP, etc. (attention à la virgule en préfixe).

Une formulation abrégée – (*pas . altération*) – signifie que l'altération de l'élément en question sera valide quelle que soit l'octave.

En ce qui concerne les gammes microtonales dans lesquelles un « dièse » n'est pas d'un centième, *altération* se réfère à un deux-centième de ton entier.

```
\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 |
}
```



Impression de musique aux métriques différentes

Bien qu'ayant des métriques bien différentes, les deux parties ci-dessous présentées restent synchrones.

Les barres de mesure ne peuvent plus être gérées au niveau du contexte Score ; le Timing_translator doit être déplacé du contexte Score au contexte Staff afin de permettre des barres de mesure individualisées.

Le Bar_number_engraver devra lui aussi être déplacé, puisqu'il dépend de propriétés attachées au Timing_translator, afin de numéroter les mesures. L'utilisation d'un bloc \with dans la portée concernée permettra un affichage des numéros de mesure.

```
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
```
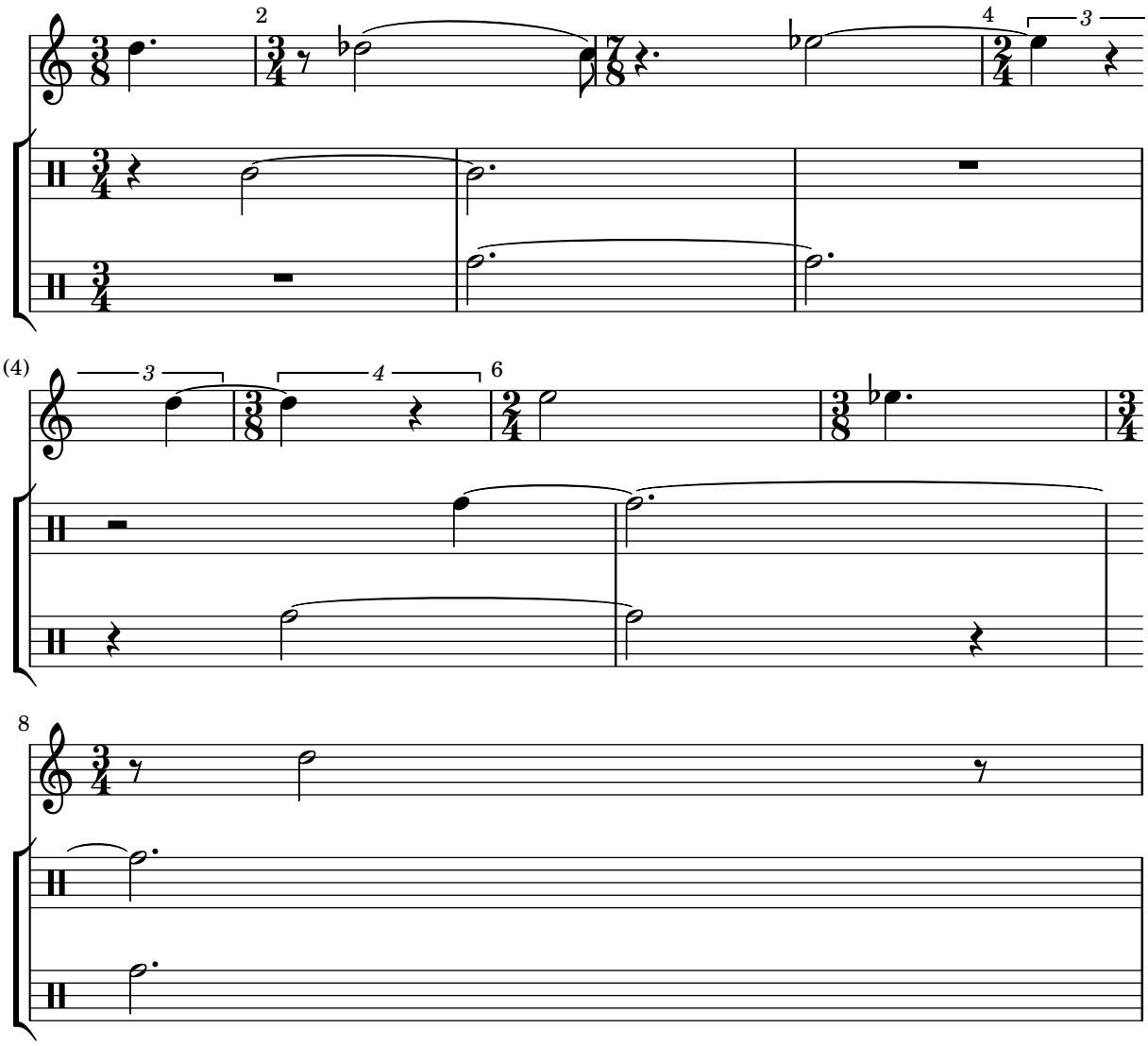
```
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

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
  >>
}
```



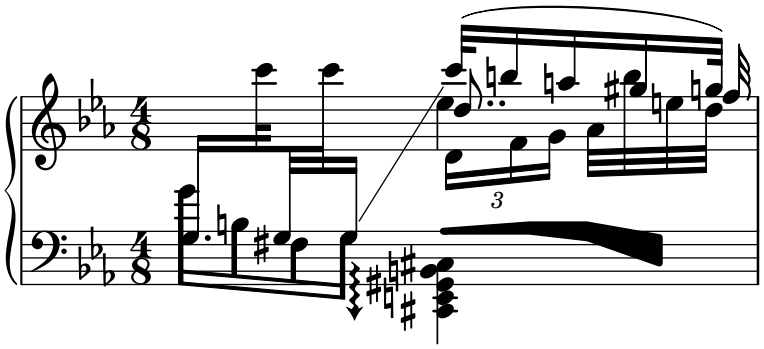
Screech and boink

Notation complexe et aléatoire.

```
\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"
```

```
    }
  }
}
```



Moignons de hampe

Certaines conventions en matière de notation autorisent les ligatures à enjamber des silences. Dans certains cas, des moignons de hampe accrochés à la ligature offrent une meilleure visibilité du rythme ; certaines éditions modernes vont même alors jusqu'à omettre le silence.

Cet exemple illustre la progression : notation traditionnelle, ligature enjambant le silence, silence surplombé d'un moignon et enfin seule une hampe tronquée. Les moignons s'obtiennent par amendement de la propriété `stemlet-length` de l'objet `Stem`, alors que les silences sont masqués à l'aide d'un `\hide`.
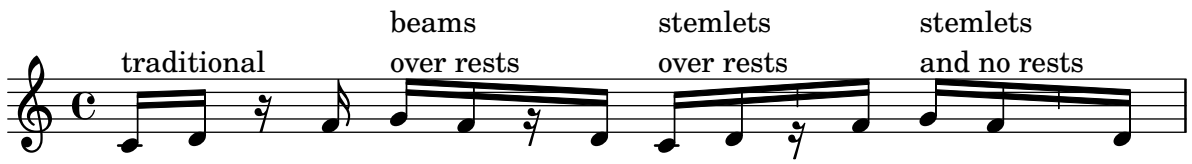
Les *markups* ajoutés au code ci-dessous mettent en exergue les différentes notations.

```
\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']
}
```



17 Ancient notation

See also Section “Ancient notation” dans *Manuel de notation*.

Ajout d’une basse chiffrée au-dessus ou au-dessous des notes
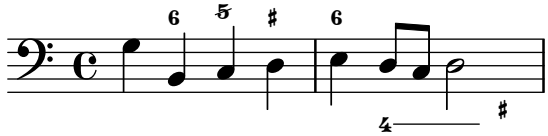
Une ligne de basse chiffrée peut se positionner au-dessus ou en dessous d’une partie de basse à l’aide des commandes `\bassFigureStaffAlignmentDown` et `\bassFigureStaffAlignmentUp`. Une modification temporaire du positionnement s’obtient en faisant précéder la commande d’un `\once`.

La commande `\bassFigureStaffAlignmentNeutral` réinitialise le positionnement des chiffrages à sa valeur par défaut.

```
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
  >>
}
```



Gravure de musique ancienne

Voici de nombreux symboles contenus dans la fonte Emmentaler et que LilyPond utilise en matière de musique ancienne.

```
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

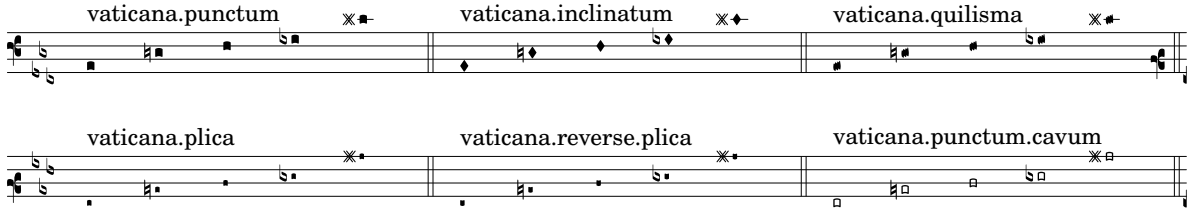
    \override NoteHead.style = #'hufnagel.punctum
    <>^"hufnagel.punctum" \m

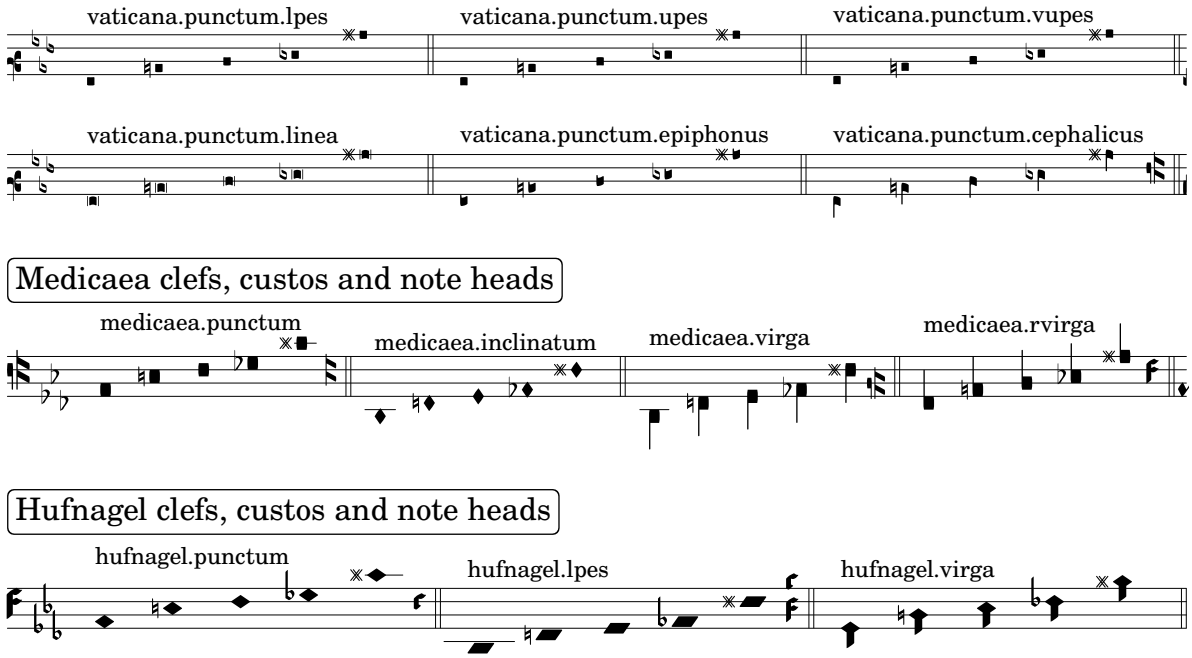
    \clef "hufnagel-do2"
    \override NoteHead.style = #'hufnagel.lpes
    <>^"hufnagel.lpes" \m

    \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





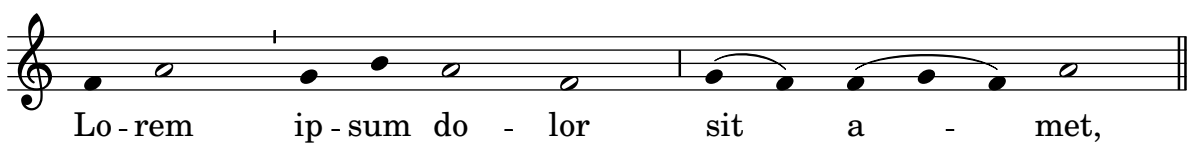
Exemples de notation ancienne – transcription moderne de musique grégorienne

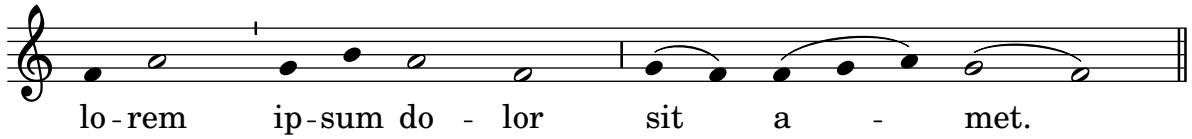
Voici comment vous pourriez transcrire du grégorien. Pour mémoire, il n'y a en grégorien ni découpage en mesure, ni hampe ; seules sont utilisées des têtes de note blanches ou noires, ainsi que des signes spécifiques permettant d'indiquer des silences de différentes durées.

```
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
  >>
}
```

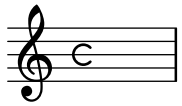




Métrique ancienne

La métrique peut s'imprimer dans un style ancien.

```
{
  \override Staff.TimeSignature.style = #'neomensural
  s1
}
```

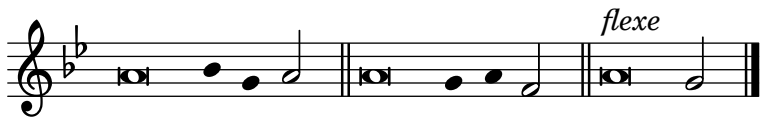


Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```
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
  }
}
```



Guidons

Les guidons peuvent adopter différents styles.

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

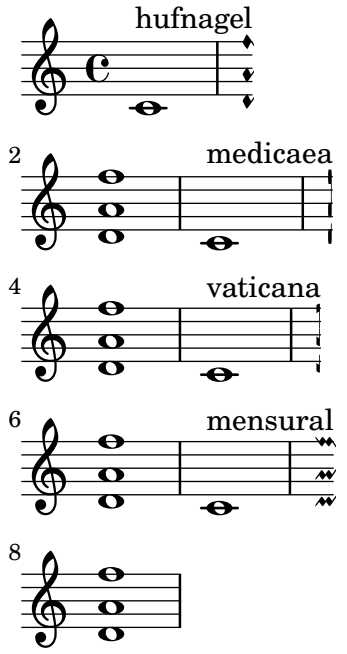
\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4
```

```
    \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
  }
}
```



Incipit

Dans le cadre de la transcription de musique mensurale, un « incipit » en début de pièce permet d'indiquer tonalité et tempo originaux. Si les musiciens sont aujourd'hui habitués aux barres de mesure qui mettent en évidence la structure rythmique, elles n'étaient pas courantes à l'époque – le mètre variait souvent après quelques notes. Dans une forme de compromis, les barres de mesures sont régulièrement imprimées entre les portées.

```
% 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.
}

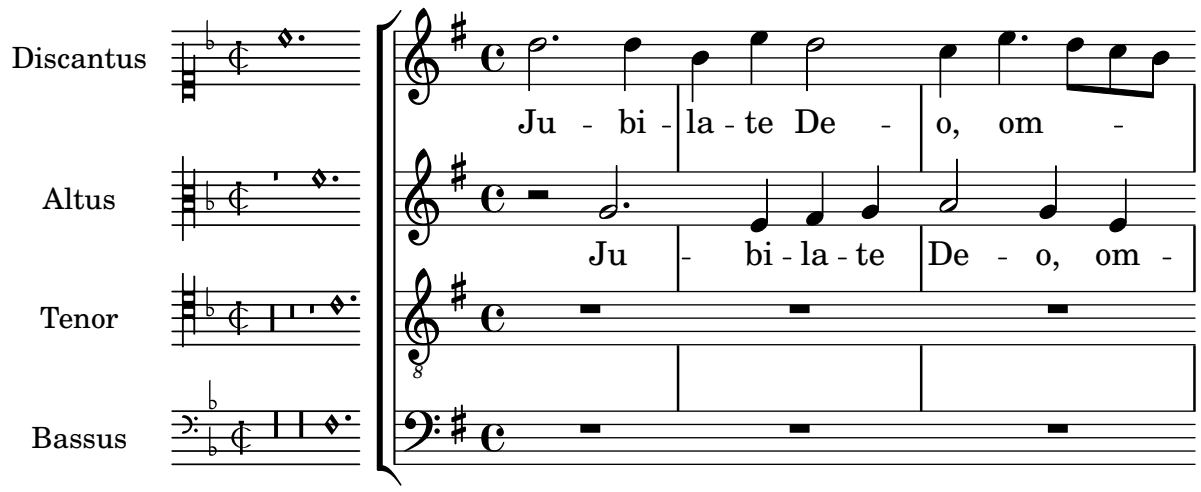
bassusIncipit = \new PetrucciStaff {
  % The original print shows the b flat
  % for the f major key signature twice.
  \override Staff.KeySignature.flat-positions = #'((-7 . 6))
  \clef "mensural-f"
  \key f\major
  \time 2/2
  \tweak Y-offset #1 r\longa \tweak Y-offset #1 r\longa
  f1.
}
```

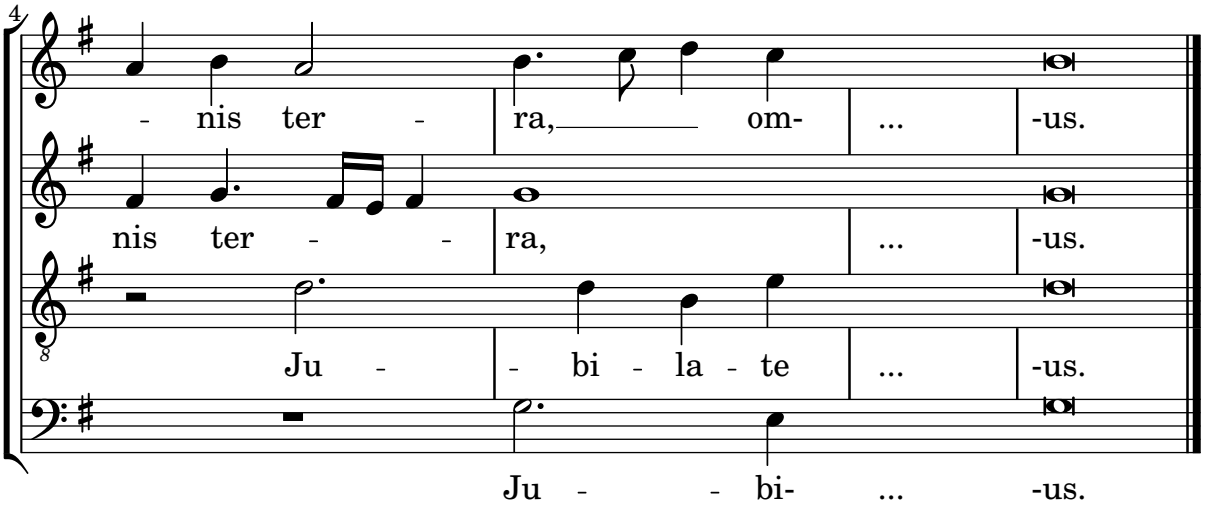
```
bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \hide NoteHead
    e1 |
    g\breve |
  }
}

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
  }
}
```





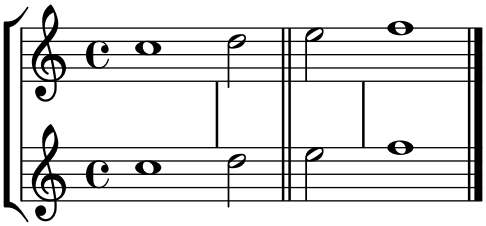
Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat il faudra définir `measureBarType` à `"-span|"` et utiliser un regroupement de portées permettant l'extension des barres entre les portées, tel un `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



Styles de silences

Les silences peuvent être gravés selon différents styles.

```
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

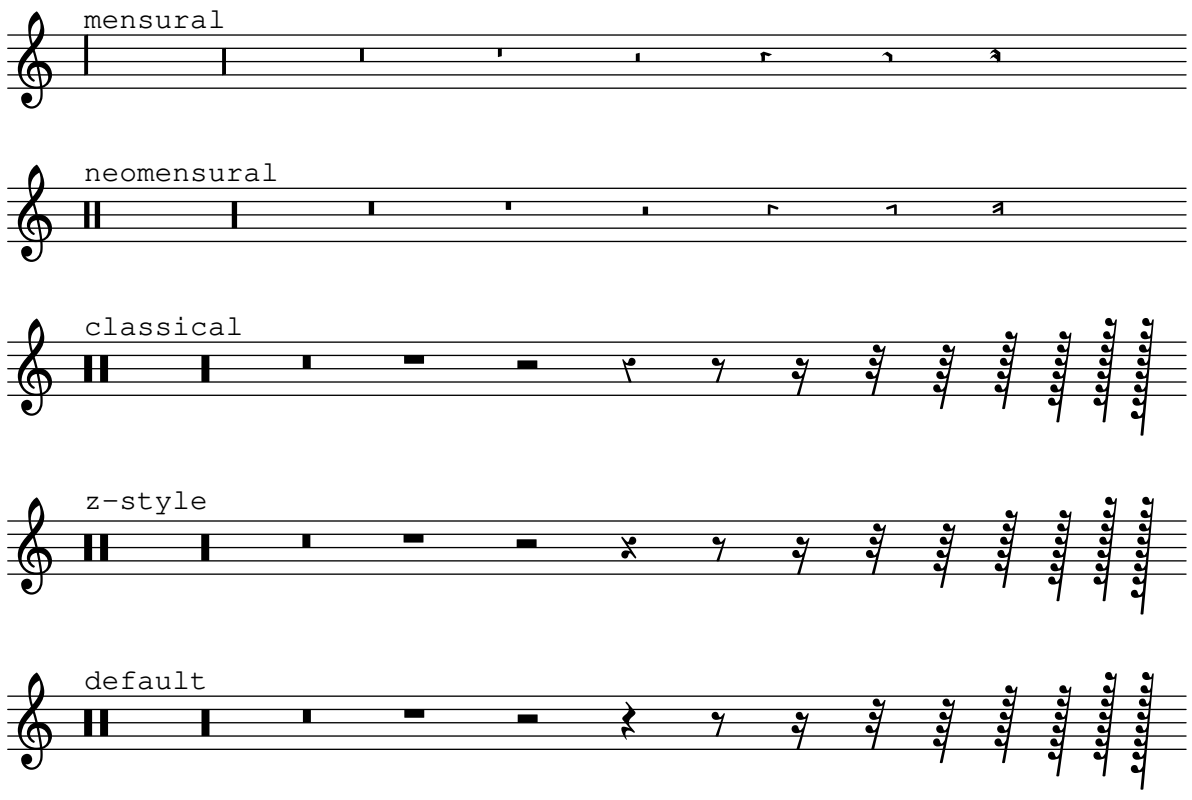
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

  \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
}
```



Recours aux balises pour produire une partition ancienne et moderne à partir de la même source

Grâce aux balises (*tags*), il est possible d'utiliser une même source pour produire une partition de musique mensurale et moderne. Dans cet exemple est créée la fonction `menrest` qui permettra de positionner les silence comme dans la version originale, tout en respectant leur position sur une portée standard.

Les balises permettent aussi de gérer d'autres différenciations selon les besoins, comme des « mesures de silence » (R1, R\breve, etc.) en notation moderne mais des silences normaux (r1, r\breve, etc.) en notation ancienne. L'action de convertir de la musique mensurale en version moderne est communément appelée « transcription ».

L'appel à `c4.\Be c8 c\Am` revient au même que `c4.[ c8 c]`. Néanmoins, il évite les avertissements si le début intervient sur une note dépourvue de ligature mais pourrait s'en voir affublée de par l'utilisation du `Completion_heads_engraver`.

[La ligne légèrement plus courte dans la portée mensurale permet d'éviter que le glyphe de custode ne soit raccourci lors de la génération des images. On peut l'éviter à l'aide d'un `\with-true-dimensions` comme ici.]

```
\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
}

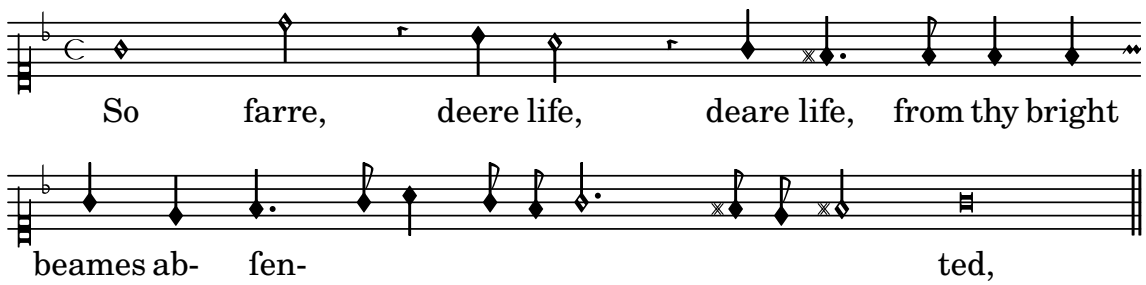
MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
```

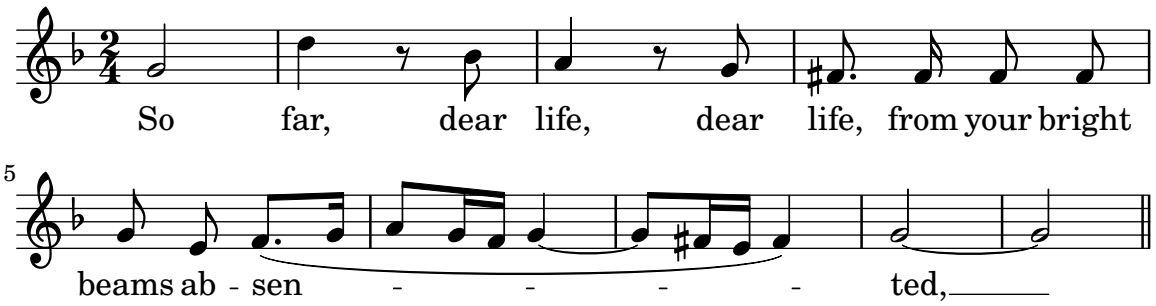
```
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}

\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
    >>
  }
}
```





Articulation baroque en forme de coche

On trouve régulièrement, en musique baroque, cette courte ligne verticale. Sa signification peut varier, mais elle indique le plus souvent une note plus « appuyée ». Voici comment générer ce signe particulier.

```
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

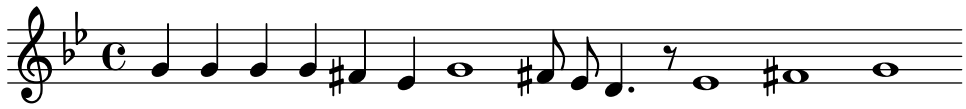
See also Section “World music” dans *Manuel de notation*.

Improvisation en musique arabe

Lorsque les improvisations ou *taqasim* sont temporairement libres, la métrique peut ne pas apparaître, auquel cas on utilisera un `\cadenzaOn`. Les altérations accidentelles devront alors être répétées en raison de l’absence de barre de mesure. Voici comment pourrait débuter une improvisation de *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
}
```
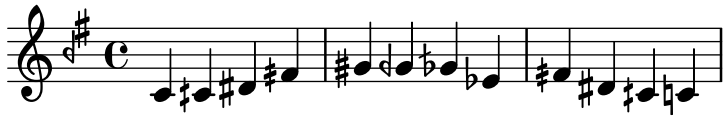


Exemple de musique « Makam »

Le « makam » est une forme de mélodie turque qui utilise des altérations d’un neuvième de ton.

Consultez le fichier d’initialisation `ly/makam.ly` pour plus de détails sur les hauteurs et altérations utilisées (voir le chapitre 5.7.4 - Autres sources d’information du manuel d’initiation pour le localiser).

```
\include "makam.ly"

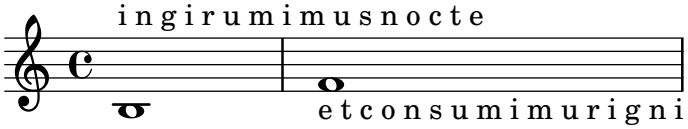
\relative c' {
  \set Staff.keyAlterations = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Impression de texte de droite à gauche

Du texte, inclus dans un objet *markup*, peut s’imprimer de droite à gauche, comme illustré ci-dessous.

```
{
  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 }
  }
}
```



Exemple de makam turc

Ce canevas utilise le début d'un *saz semai* turc bien connu du répertoire aux fins d'illustrer certains éléments de la notation musicale turque.

```
#(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
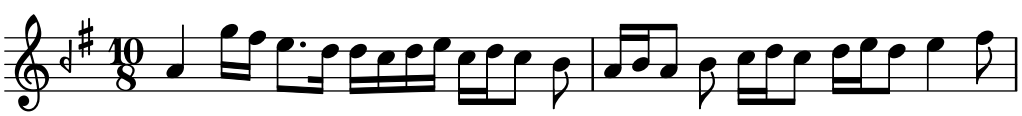
  \key a \huseyni
  \time 10/8
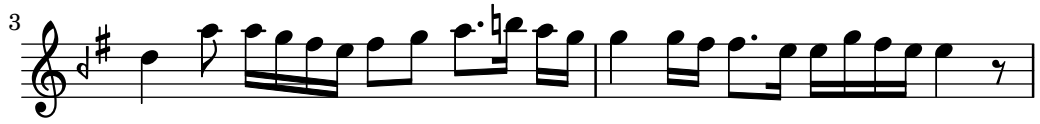
  a'4 g'16[ fb] e8.[ d16] d[ c d e] c[ d c8] bfc |
  a16[ bfc a8] bfc c16[ d c8] d16[ e d8] e4 fb8 |
  d4 a'8 a16[ g fb e] fb8[ g] a8.[ b16] a16[ g] |
  g4 g16[ fb] fb8.[ e16] e[ g fb e] e4 r8 |
}

\layout {
  indent = 0
}
```

Hüseyni Saz Semaisi

Lavtacı Andon




3

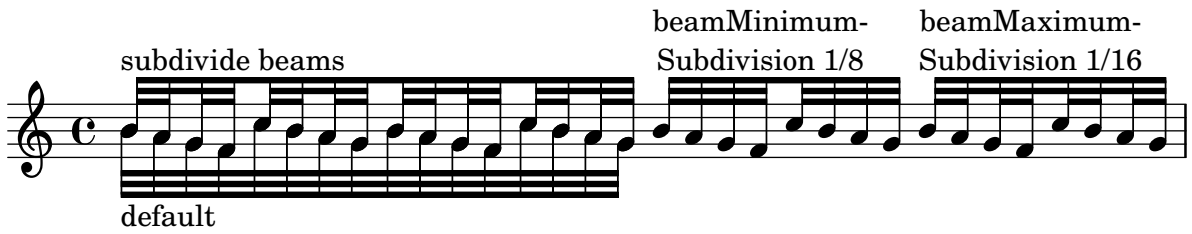
Autres collections

19 Automatic notation

Subdivision de ligatures automatiques

Dès lors que la propriété subdivideBeams aura été activée, une ligature sera subdivisée autant que possible. Les limites dans la profondeur sont gérées par les propriétés beamMinimumSubdivision et beamMaximumSubdivision.

```
\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]
  }
}
```



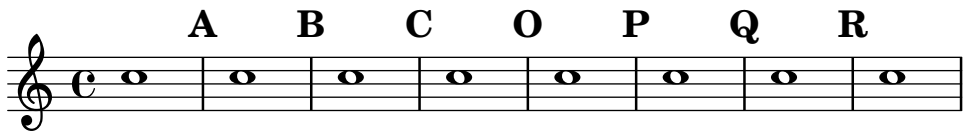
Fixation arbitraire du numéro de repère de départ

Voici comment initialiser arbitrairement une indication automatique de repère, qu'elle soit alphabétique ou numérique :

```
\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
}
```



Génération en Scheme de partitions complètes (y compris des parties d'ouvrage) sans utiliser l'analyseur

Une partition LilyPond, de manière interne, n'est rien d'autre qu'une expression Scheme générée par l'analyseur syntaxique de LilyPond. Il est donc possible, à l'aide de Scheme, de générer automatiquement une partition sans fichier source. Une expression musicale en Scheme sera transformée en partition par un appel à

```
(scorify-music music)
```

Ceci aura pour effet de générer un objet `score` auquel sera appliqué un bloc `layout` comportant la fonction

```
(let* ((layout (ly:output-def-clone $defaultlayout)))
   ; modification de la mise en forme, puis assignation :
   (ly:score-add-output-def! score layout))
```

Il suffit alors de transmettre ce `score` à LilyPond pour qu'il le grave. Les trois fonctions – `(add-score score)`, `(add-text text)` et `(add-music music)` – définies dans le code ci-dessous permettent de transmettre à LilyPond, aux fins de les graver, une partition complète, un *markup* ou simplement de la musique.

Cet exemple permet aussi de graver les pièces contenues dans un bloc `\book { ... }` ainsi que des partitions de niveau supérieur. Chaque partition destinée à être gravée est alors ajoutée à la liste des partitions de niveau supérieur ; le `toplevel-book-handler` – fonction Scheme appelée pour traiter un *book* dès que le bloc `\book { ... }` est clôturé – s'adapte pour prendre en charge tous les éléments `score` jusque là collectés dans l'ouvrage.

Attention : Pour des raisons techniques, seul le premier `\book` sera rendu puisque les autres commandes `\book` génèrent des fichiers additionnels.

```
#(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 }
```
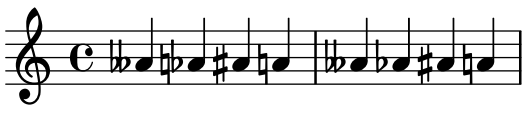
Score with a do



Suppression des bécarres superflus

En accord avec les règles traditionnelles de l'écriture musicale, on grave un bécarre avant un dièse ou un bémol si la note était auparavant affublée d'un double-dièse ou double-bémol. Pour adopter un comportement plus contemporain, la propriété `extraNatural` du contexte `Staff` doit se voir attribuer la valeur `#f` (faux).
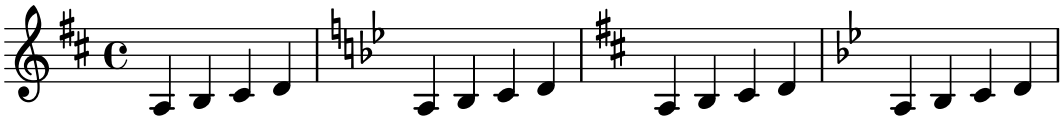
```
\relative c'' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Suppression des bécarres superflus lors d'un changement de tonalité

Après un changement de tonalité, un bécarre est imprimé pour annuler toute altération précédente. Ce comportement s'annule en désactivant la propriété `printKeyCancellation` du contexte `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
}
```



Ensemble vocal avec réduction pour piano

Ce canevas ajoute une réduction pour piano à une partition standard pour chœur à quatre voix mixtes. Ceci illustre l'un des avantages de LilyPond : une expression musicale peut être réutilisée sans effort. Toute modification apportée à l'une des voix, mettons `tenorMusique`, est automatiquement reportée dans la réduction pour 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
}

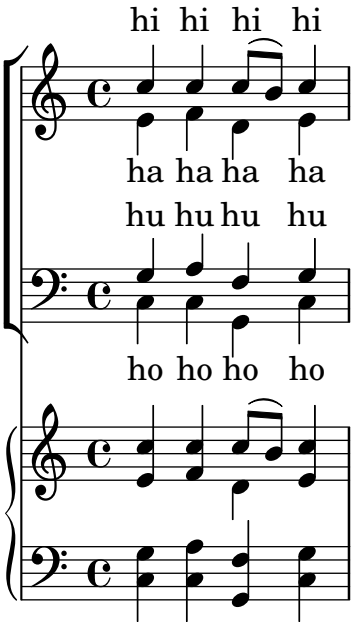
\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \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 >>
      >>
    >>
  >>
}
```

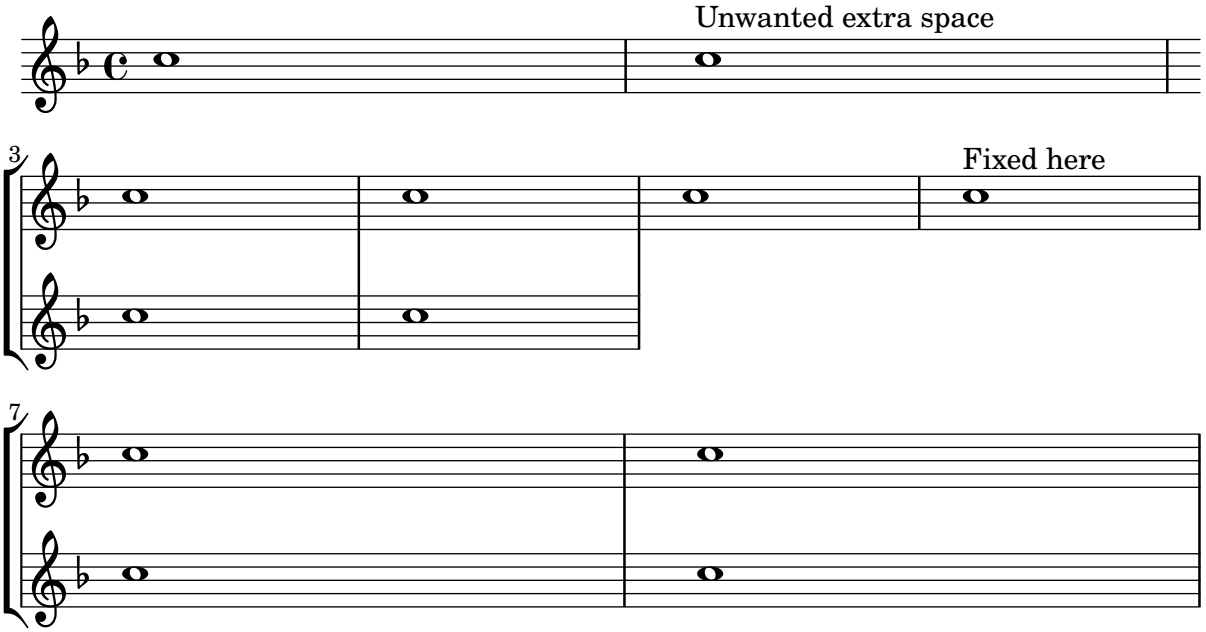


20 Breaks

Ajout d'une portée supplémentaire après un saut de ligne

Lorsqu'une nouvelle portée vient s'ajouter après un saut de ligne, LilyPond préserve un espace juste avant le saut de ligne – pour un éventuel changement d'armure qui, quoi qu'il en soit, n'est pas imprimé. L'astuce consiste alors, comme indiqué dans l'exemple suivant, à ajuster le `explicitKeySignatureVisibility` de l'objet graphique `Staff`.

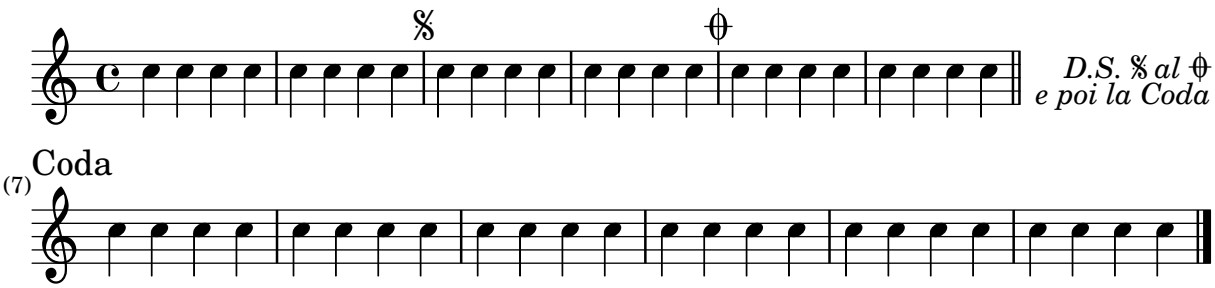
```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
       \new Staff {
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
    c1 | c^"Fixed here" \break
    << { c1 | c }
       \new Staff {
         \once \set Staff.explicitKeySignatureVisibility =
           #end-of-line-invisible
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
  }
}
```



Positionnement des segno et coda (avec saut de ligne)

Le code ci-dessous permet d'adjoindre à un signe *segno* un texte *D.S. al Coda*, là où se trouverait normalement un bout de portée. La *coda* entamera une nouvelle ligne. Une variante, indiquée ici même, permet de laisser la *coda* sur la même ligne.

```
\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
}
```



Masquage de la première ligne si elle est vide

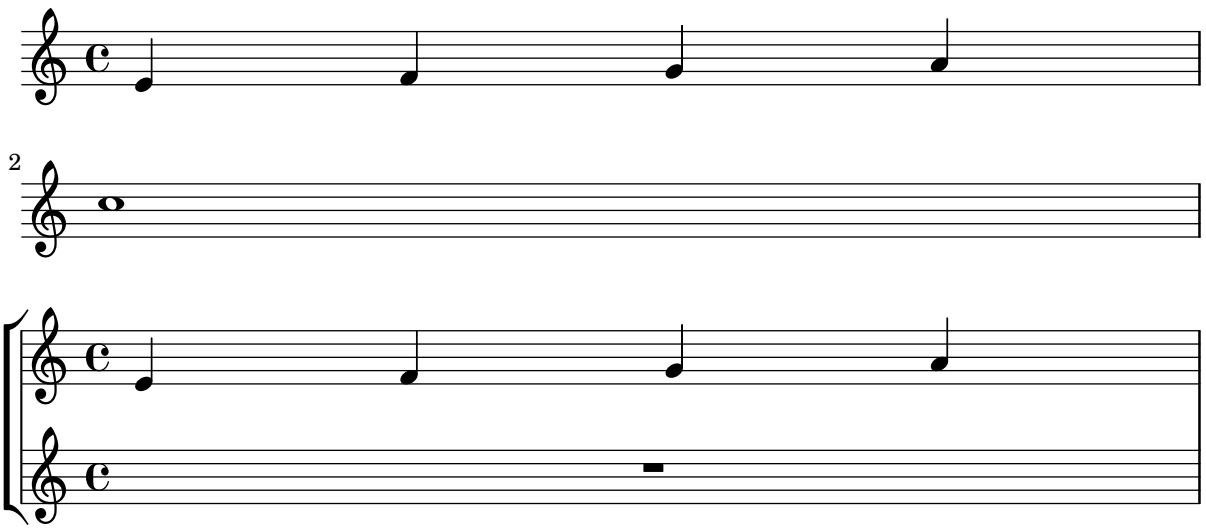
Par défaut, le premier système comportera absolument toutes les portées. Si vous préférez masquer les portées vides y compris pour le premier système, vous devrez activer la propriété `remove-first` de l'objet `VerticalAxisGroup`. Mentionnée dans un bloc `\layout`, cette commande agira de manière globale. Pour qu'elle ne soit effective que pour une portée particulière,

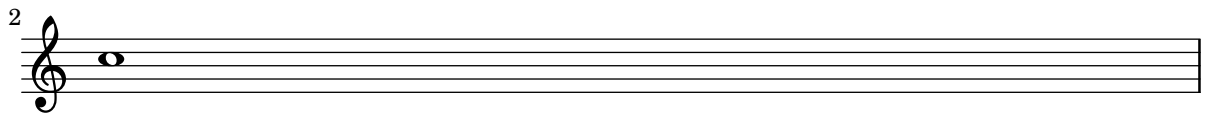
vous devrez également spécifier le contexte (`Staff` pour qu'il ne concerne que la portée en cours) en préfixe de la propriété.

La première ligne inférieure du deuxième `StaffGroup` est bien présente, pour la simple raison que le réglage en question ne s'applique qu'à la portée dans laquelle il a été inscrit.

```
\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
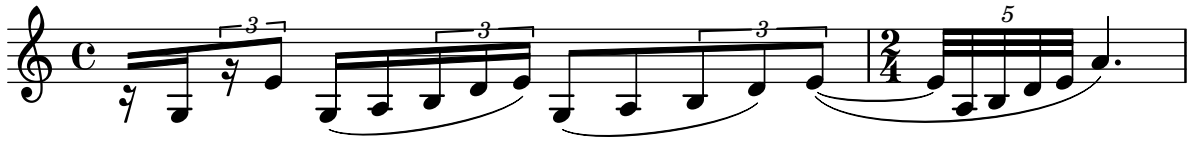
Coexistence de ligatures et liaisons avec des rythmes comprenant des n-olets

La syntaxe de LilyPond demande parfois un positionnement inhabituel des parenthèses, crochets, etc. qui peuvent s'entrelacer.

Par exemple, le crochet ouvrant une ligature manuelle se place **à la suite** de la note de départ et sa durée, non pas avant. De même, le crochet fermant se place à la fin de la ligature, y compris lorsque la dernière note se trouve incluse dans un n-olet.

Cet extrait illustre la manière de combiner ligatures manuelles, liaisons d'articulation, de prolongation ou de phrasé, avec des n-olets bornées par des accolades.

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

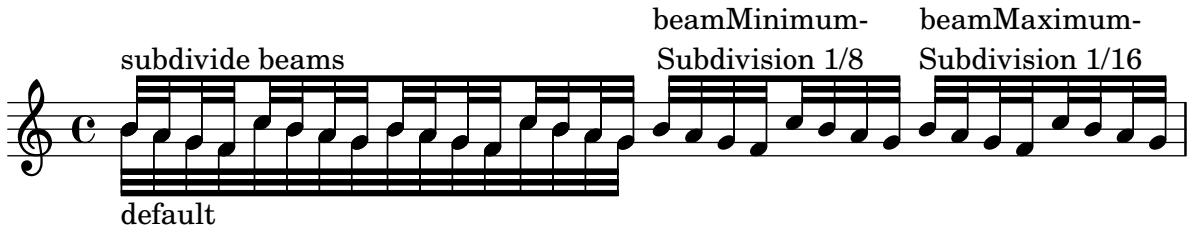


Subdivision de ligatures automatiques

Dès lors que la propriété `subdivideBeams` aura été activée, une ligature sera subdivisée autant que possible. Les limites dans la profondeur sont gérées par les propriétés `beamMinimumSubdivision` et `beamMaximumSubdivision`.

```
\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]
  }
}
```



Modification de l'aspect des liaisons d'articulation

Une liaison d'articulation peut se présenter sous la forme d'un trait continu ou discontinu, voire en pointillé.

```
\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)
}
```



Contrôle de l'impression des crochets de n-olet

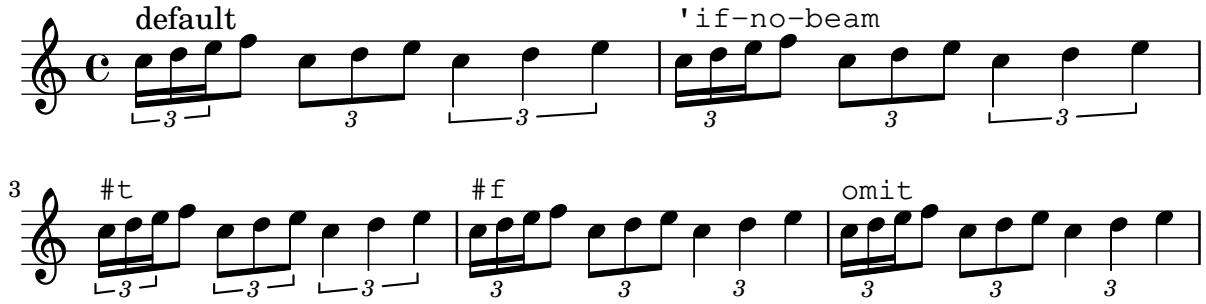
Selon la tradition, les crochets indicateurs de n-olet sont toujours imprimés, sauf dans le cas où ils seraient de la même longueur qu'une ligature.

LilyPond permet, au travers de la propriété `bracket-visibility`, de contôler précisément leur affichage : déterminée à `#t`, ils seront toujours imprimés ; `#f` permet de ne jamais les imprimer – donc omettre l'objet `TupletBracket` –, et `if-no-beam` les imprimera en l'absence de ligature (comportement par défaut).

```
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
  }
}
```
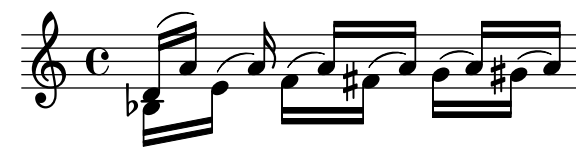


Création d'une liaison entre plusieurs voix

Dans certaines situations, il peut être nécessaire de lier des notes appartenant à des voix différentes. La solution consiste à ajouter une note invisible dans l'une des voix, à l'aide de la commande \hideNotes.

Voici par exemple la mesure 235 de la chaconne de la seconde partita pour violon 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)
    }
  >>
}
```
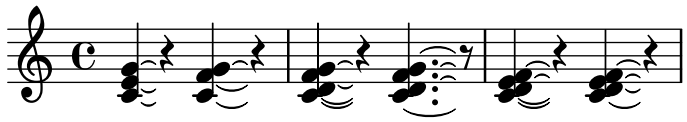


Liaison « Laissez vibrer »

Les indications de « laisser vibrer » ont une taille fixe. Leur formatage est accessible au travers de la propriété `tie-configuration`.

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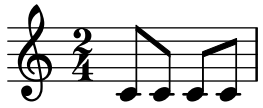
  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
     = #`((-7 . ,DOWN)
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



Contrôle manuel du positionnement des ligatures

Le positionnement et la pente des ligatures peuvent être contrôlés manuellement à l'aide d'une adaptation de la propriété `positions` de l'objet `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
}
```
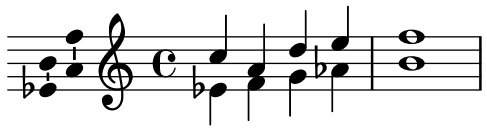


22 Contexts and engravers

See also Section “Changing defaults” dans *Manuel de notation* and Section “Contexts and engravers” dans *Manuel d'initiation*.

Un ambitus par voix

L'ambitus peut être individualisé par voix. Il faut en pareil cas éviter qu'ils se chevauchent.

```
\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
  }
>>
```

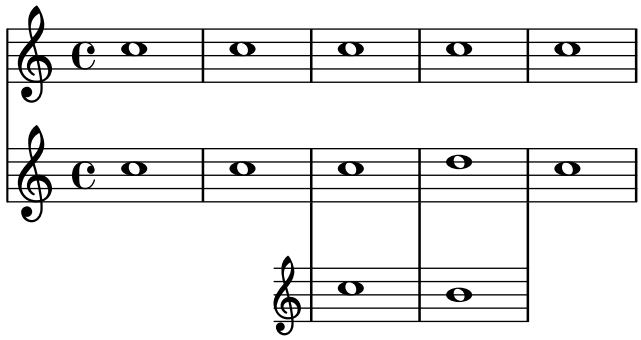


Ajout d'une portée supplémentaire

Vous pouvez ajouter une nouvelle portée, éventuellement de manière temporaire, après le début d'un morceau.

```
\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
      }
    }
```

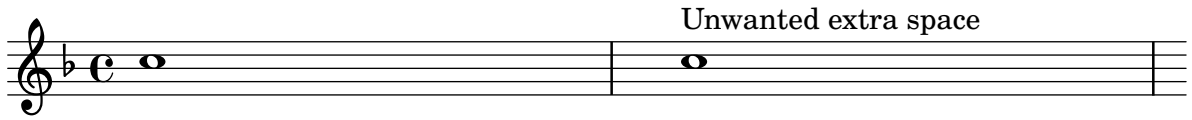
```
  >>
}
```

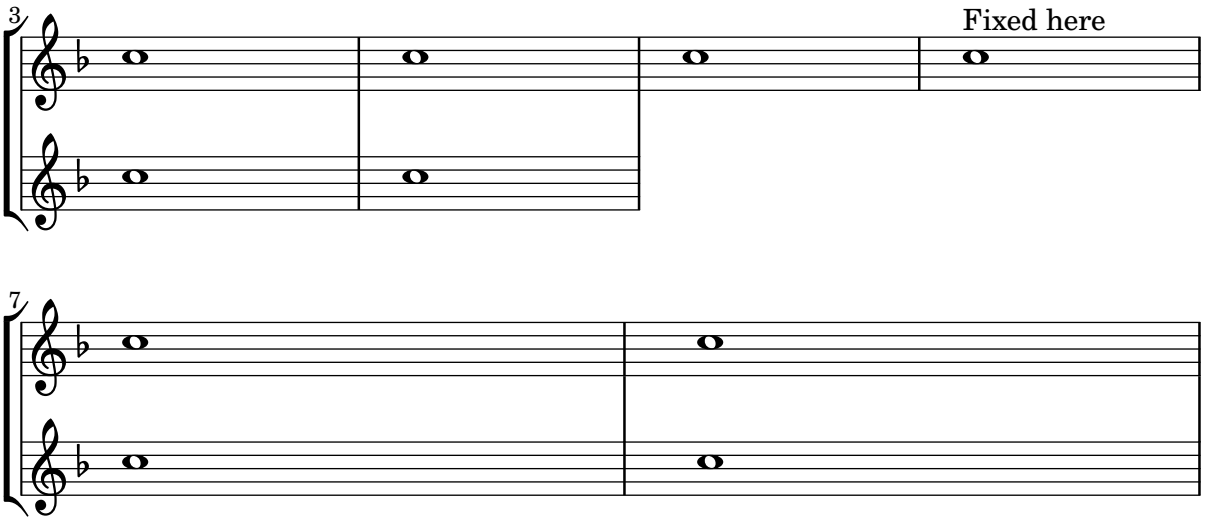


Ajout d'une portée supplémentaire après un saut de ligne

Lorsqu'une nouvelle portée vient s'ajouter après un saut de ligne, LilyPond préserve un espace juste avant le saut de ligne – pour un éventuel changement d'armure qui, quoi qu'il en soit, n'est pas imprimé. L'astuce consiste alors, comme indiqué dans l'exemple suivant, à ajuster le `explicitKeySignatureVisibility` de l'objet graphique `Staff`.

```
\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
       }
    >>
  }
}
```



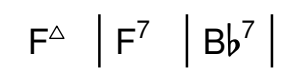


Chiffrages et barres de mesure

L'ajout du graveur `Bar_engraver` à un contexte `ChordNames` permet d'imprimer les barres de mesure entre les chiffrages.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```
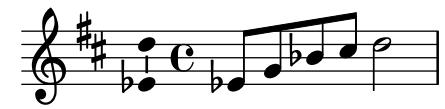


Ambitus après l'armure

L'ambitus se place par défaut à gauche de la clef. La fonction `\ambitusAfter` permet cependant de modifier ce positionnement ; sa syntaxe est `\ambitusAfter` *grob-interface* – voir Section "Graphical Object Interfaces" dans *Référence des propriétés internes* pour une liste des valeurs de *grob-interface* possibles.

L'un des cas d'usage est d'insérer l'ambitus entre l'armure et la métrique.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



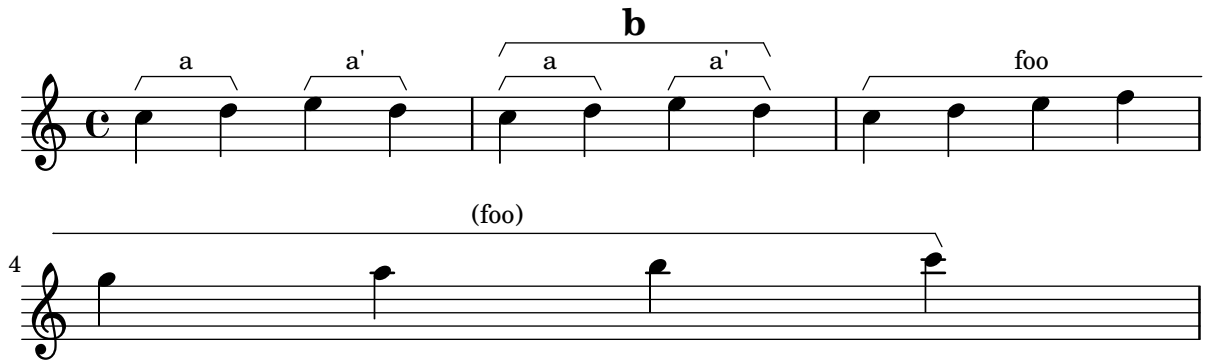
Crochet d'analyse avec texte

Un *markup* textuel peut venir s'ajouter aux crochets d'analyse grâce à la propriété `text` de l'objet graphique `HorizontalBracketText`. Plusieurs crochets présents en un même moment requièrent d'utiliser la commande `\tweak`.

Le texte ajouté sera répété, entre parenthèse, après un saut de ligne. L'ordre vertical des crochets imbriqués se gère à l'aide de la propriété `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
}
```
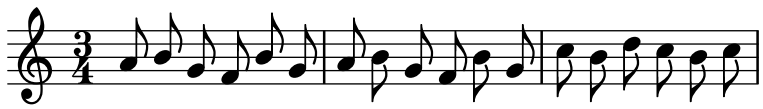


Changement automatique de l'orientation de hampe de la note médiane selon la mélodie

Afin de suivre la ligne mélodique, LilyPond peut inverser l'orientation de hampe de la note médiane, dès lors qu'aura été ajouté au contexte de voix le graveur `Melody_engraver`.

La propriété de contexte `suspendMelodyDecisions` permet, si besoin, de désactiver temporairement ce comportement.

```
\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
  }
}
```
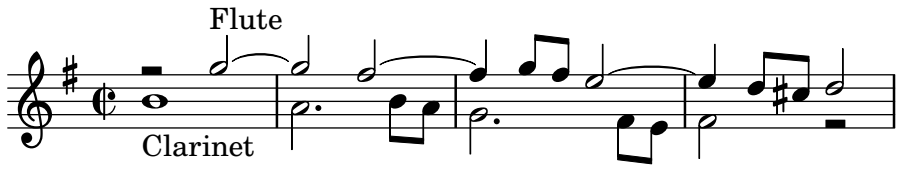


Affectation d'un canal MIDI par voix

Lorsque LilyPond génère un fichier MIDI, chaque portée sera par défaut affectée à un canal, quel que soit le nombre de voix qu'elle contient. Ceci permet d'éviter de se retrouver à court de canaux, sachant qu'il n'y en a que seize de disponibles par piste.

Le fait de déplacer le `Staff_performer` dans le contexte `Voice` permet d'affecter à chaque voix d'une même portée un canal MIDI spécifique. Dans l'exemple suivant, la même portée donnera lieu à deux canaux MIDI différents, chacun étant affecté de son propre `midiInstrument`.

```
\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
  }
}
```



Modification de la métrique au sein d'un fragment polymétrique à l'aide de \scaleDurations

Polymétrie sans alignement des mesures

La prise en charge de contextes aux métriques indépendantes demande de supprimer le Timing_translator du contexte Score, conjointement à la création d'un contexte TimingStaffGroup disposant du Timing_translator. Ceci fera de Timing un alias de TimingStaffGroup, permettant ainsi aux commandes \time de s'appliquer au sein du TimingStaffGroup où elles apparaissent.

Contrairement à la commande native de LilyPond \enablePerStaffTiming, cette approche requiert la création explicite de contextes TimingStaffGroup mais, par contre, autorise la création de multiples contextes Staff qui suivront de concert la mesure définie dans le TimingStaffGroup qui les englobe.

Échelonnement local des métriques

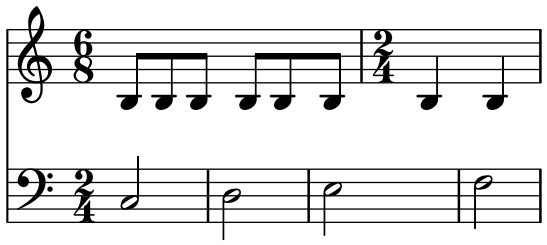
La commande \time, qui ne peut être échelonné, établit une mesure de la longueur désirée dans Timing, autrement dit dans TimingStaffGroup. Dans le code ci-dessous, toutes les portées incluses dans un TimingStaffGroup utilisent une métrique échelonnée, de sorte que, peu importe la métrique, elle tiendra dans la longueur de mesure désirée. Si l'un des contextes inclus n'avait pas de métrique échelonnée, le choix de la métrique devant être fournie à Timing aurait toute son importance.

L'utilisation de la commande \polymetric \time permet de définir les propriétés d'échelonnement de la métrique dans le contexte Timing, et l'utilisation de \scaleDurations échelonnera le mètre et les notes afin qu'elles tiennent dans la mesure.

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

```
    \alias StaffGroup
    \consists "Timing_translator"
  }
}

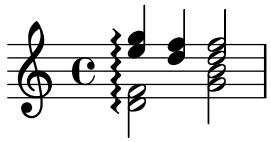
<<
  \new TimingStaffGroup <<
    \new Staff {
      \scaleDurations 8/5 {
        \time 6/5 % to set measure length in Timing
        \context Staff \polymetric \time 6/8
        b8 b b b b b
        \time 4/5 % to set measure length in Timing
        \context Staff \polymetric \time 2/4
        b4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



Arpège distribué sur plusieurs voix

Affecter le graveur `Span_arpeggio_engraver` au contexte de la portée (`Staff`) permet de distribuer un arpège sur plusieurs voix.

```
\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 }
  >>
}
```



Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le Bar_number_engraver du contexte Score, et les Time_signature_engraver, Clef_engraver et Bar_engraver du contexte Staff.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

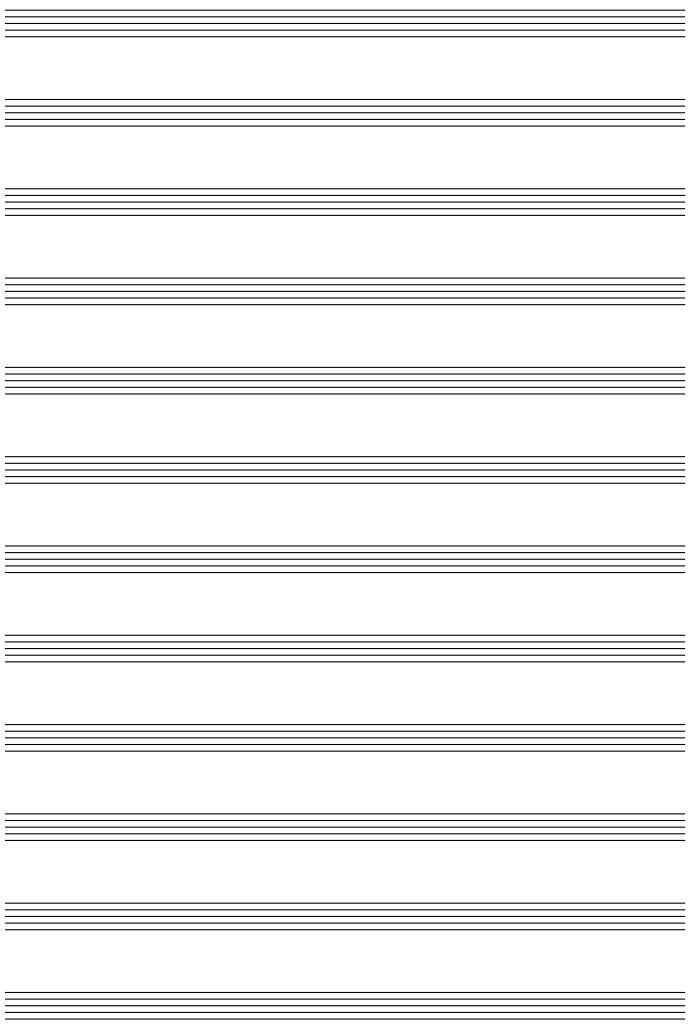
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % 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
  }
  %}
}
```



Arpège distribué pour un autre contexte que le piano

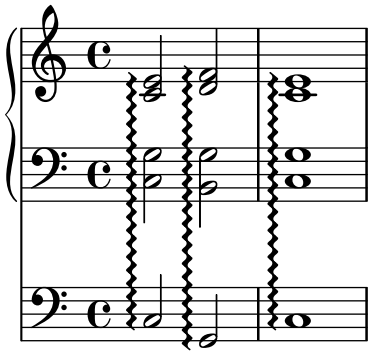
Il est possible de distribuer un arpège sur plusieurs portées d'un système autre que le GrandStaff et ses semblables (PianoStaff, +ChoirStaff et StaffGroup) dès lors que vous incluez le Span_arpeggio_engraver au contexte Score.

```
<<
  \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"
  }
}
```
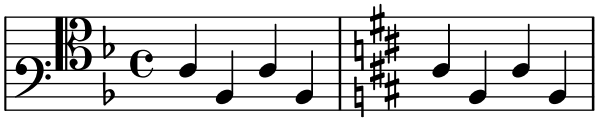


Création d'armures personnalisées

LilyPond accepte les armures personnalisées. Dans cet exemple, il s'agit d'une tonalité de ré mineur et ré majeur auxquelles on rajoute des bémols.

```
\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, |
}
```



Hampes interportées

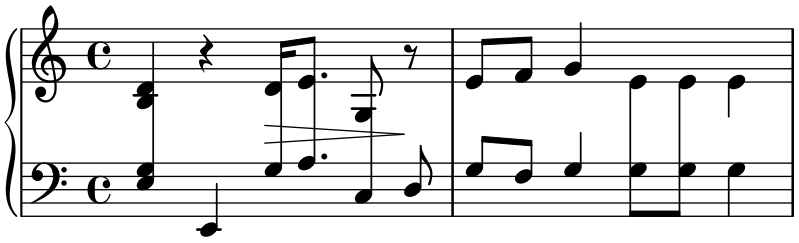
L'exemple ci-dessous illustre l'utilisation du `Span_stem_engraver` et de la commande `\crossStaff` afin de connecter des hampes entre les portées.

Nul n'est besoin de spécifier la taille des hampes ; le graveur calcule automatiquement la distance relative des têtes de note avec les portées. Il est toutefois important d'appliquer `\crossStaff` dans le contxte de voix ou de portée approprié – autrement dit en opposition à l'endroit où se trouvera une hampe) pour obtenir l'effet attendu.

```
\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 |
  }
>>
```



Définition d'un graveur en Scheme : graveur d'ambitus

Cet exemple démontre comment définir son propre graveur de tessiture à base de code Scheme. Il s'agit d'une réécriture en Scheme du code contenu dans le fichier `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))))
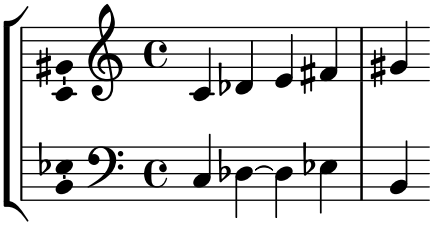
        ;; Set the ambitus clef and key signature state.
        ((stop-translation-timestep translator)
         (if ambitus
             (initialize-ambitus-state ambitus translator)))

        ;; 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 } }
}
```



Affichage d'un système complet lorsqu'une seule portée est active

Dans la plupart des conducteurs d'orchestre, il est d'usage qu'un instrument ou un groupe d'instruments ne joue pas pendant un moment ; leurs portées disparaissent alors pendant ce

temps – les anglophones appellent cela une « partition à la française » *(Frenched score)*. LilyPond dispose de cette fonctionnalité à l'aide de la commande `\RemoveEmptyStaves`.

Il est préférable, lorsque l'un d'entre eux reprend, d'imprimer **toutes les portées du groupe en question** grâce au `Keep_alive_together_engraver` placé au niveau du regroupement, qu'il s'agisse d'un `GrandStaff` ou d'un `StaffGroup`.

Dans l'exemple suivant, les violons ne jouent pas lors du deuxième système. Bien que seul le premier violon redémarre à la dernière mesure du troisième système, la portée de second violon apparaît elle aussi.

```
\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 }
    }
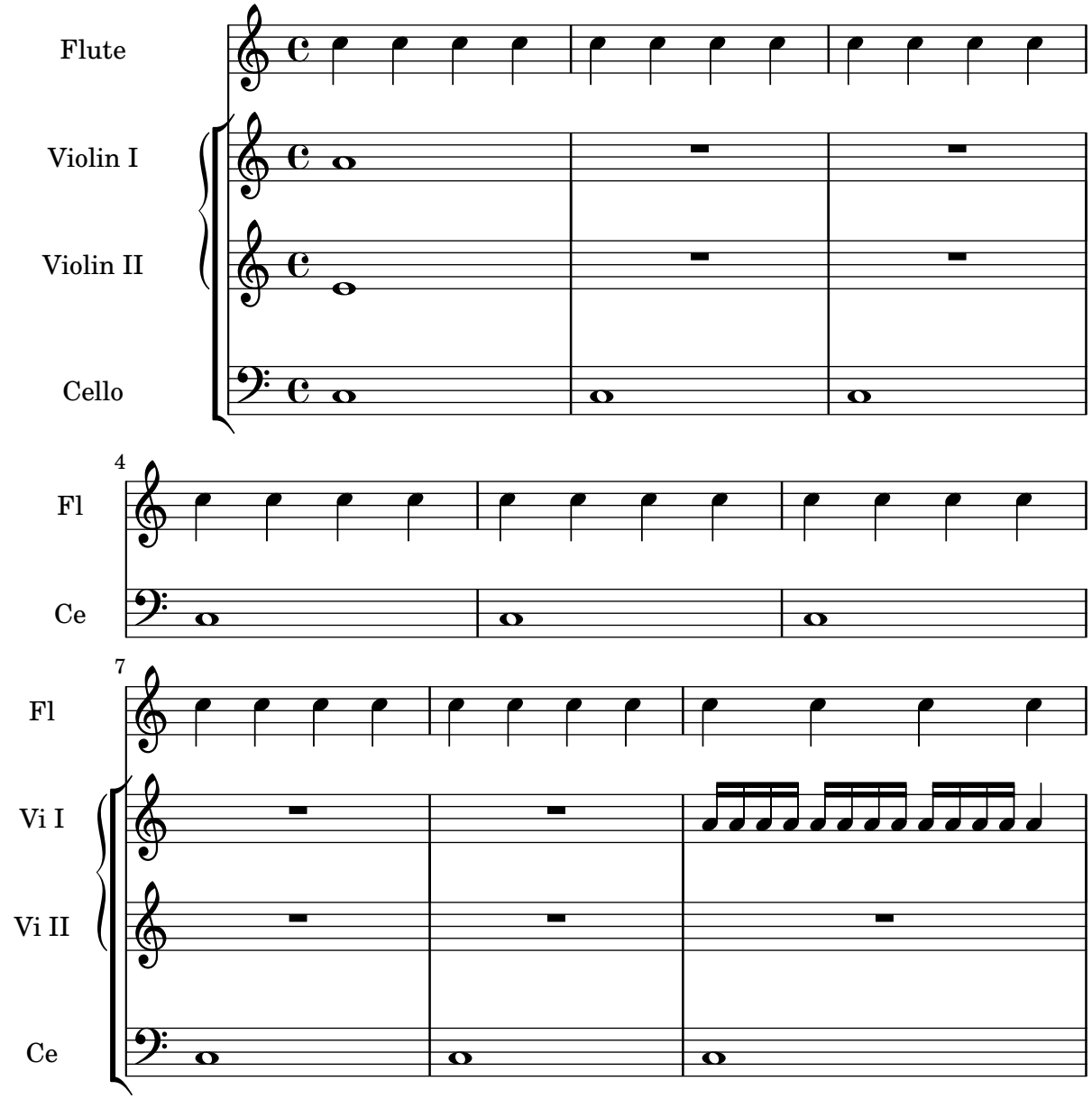
    \new StaffGroup = "StaffGroup_Strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
          shortInstrumentName = "Vi I"
        } \relative c'' {
          a1 | R1*7 | \*12 a16 a4 |
        }
        \new Staff = "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
  }
}
```



Les graveurs un par un

LilyPond gère les différents éléments nécessaires à la réalisaon d'un partision grâce à des modules. Chaque module est appelé « graveur ». Dans cet exemple, quelques graveurs sont appelés chacun à son tour, dans l'ordre suivant :

- têtes de note,
- symboles de la portée,
- clef,
- hampes,
- ligatures, liaisons, accents,

- altérations, barres de mesure, métrique et armure.

Les graveurs se regroupent. Par exemple, têtes de note, liaisons, ligature, etc. forment un contexte de voix (`Voice`). Les graveurs chargés de la métrique, des altérations, des barres de mesure, etc. forment un contexte de portée (`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
```
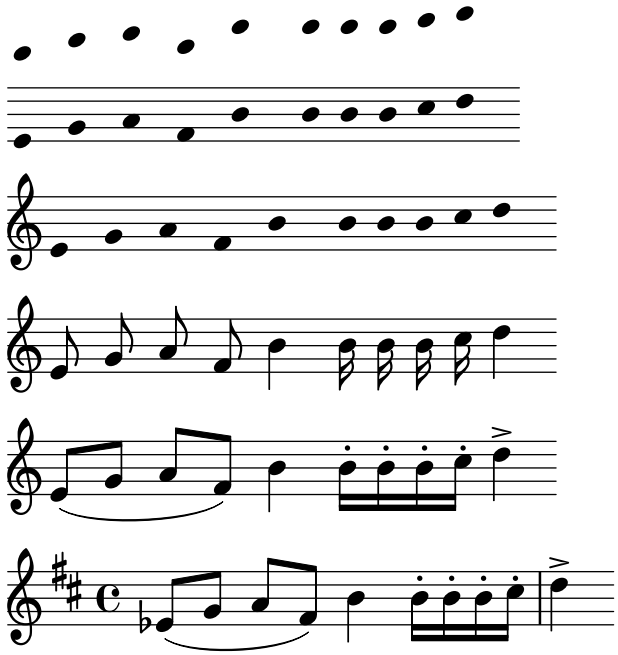
```
MyStaff = \context {
  \MyStaff
  \consists Clef_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% 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 }
  }
}
```
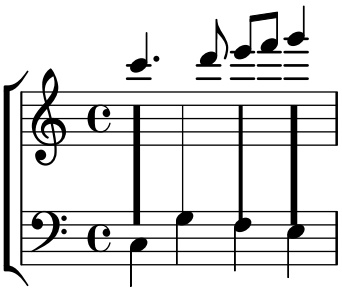


Apparence du quadrillage temporel

Modifier certaines des propriétés du quadrillage temporel aura pour effet d'en changer l'apparence.

```
\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
  }
}
```



Quadrillage temporel : mise en évidence du rythme et synchronisation des notes

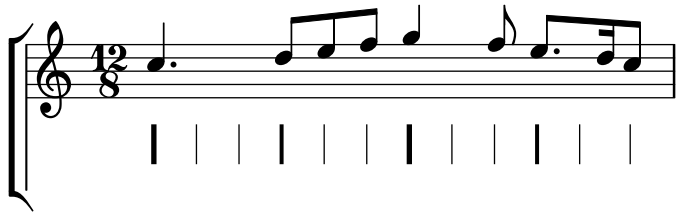
Des lignes verticales entre les portées peuvent s'ajouter dans le but d'indiquer la synchronisation entre des notes. Dans le cas de musique monophonique, on peut toutefois créer une deuxième portée, invisible, et raccourcir les lignes comme ici.

```
\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)
  }
}
```



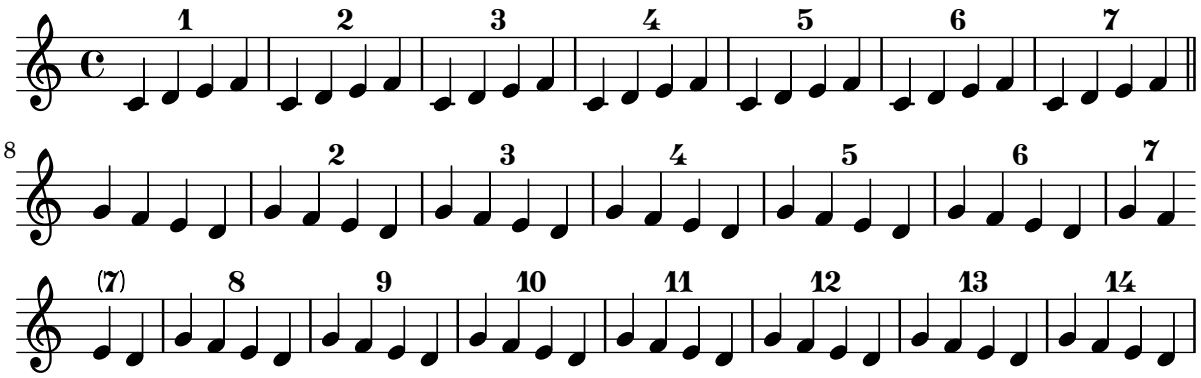
Compteur de mesures

Le code ci-dessous permet, de façon détournée, d'afficher un compteur de mesures à l'aide d'une répétition en pourcent rendue transparente.

```
\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
}
```



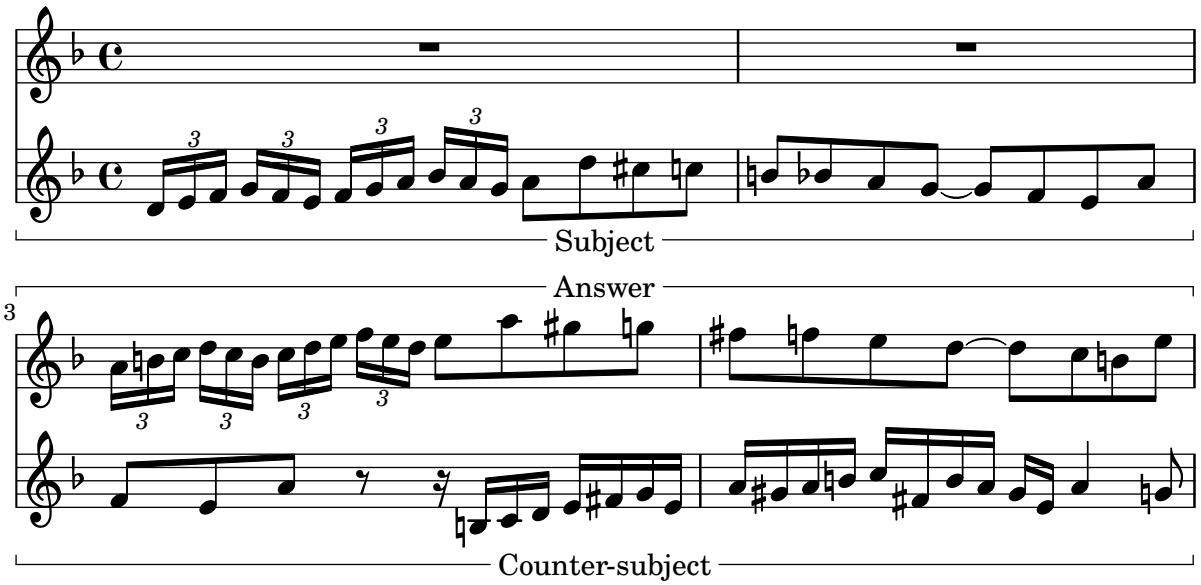
Crochet de mesure

Les bandeaux sur mesure sont un moyen alternatif d'imprimer un crochet annoté. Contrairement aux crochets horizontaux, ces bandeaux s'étendent entre deux barres de mesure plutôt qu'entre deux notes. Le texte est centré sur l'empan du crochet.

```
\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
  }
>>
```



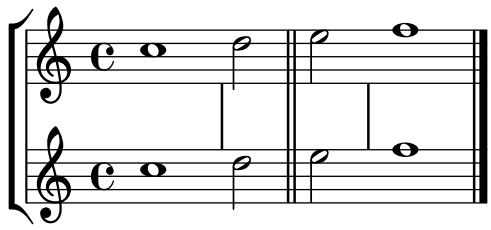
Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat il faudra définir `measureBarType` à `"-span|"` et utiliser un regroupement de portées permettant l'extension des barres entre les portées, tel un `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```

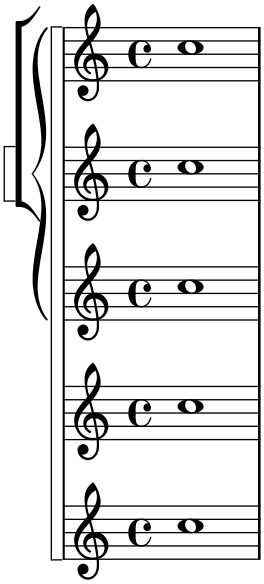


Imbrications de regroupements de portées

La propriété de contexte systemStartDelimiterHierarchy permet de créer des regroupements imbriqués complexes. Dans le cadre d'un StaffGroup, elle prend en argument la liste alphabétique des sous-groupes à hiérarchiser. Chaque sous-groupe peut être affublé d'un délimiteur particulier. Chacun des regroupements intermédiaires doit être borné par des parenthèses. Bien que des éléments de la liste puissent être omis, le premier délimiteur embrassera toujours l'intégralité des portées. Vous disposez des quatre délimiteurs SystemStartBar, SystemStartBracket, SystemStartBrace et 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 }
>>
```



Saut de ligne au milieu d'un n-olet avec ligature

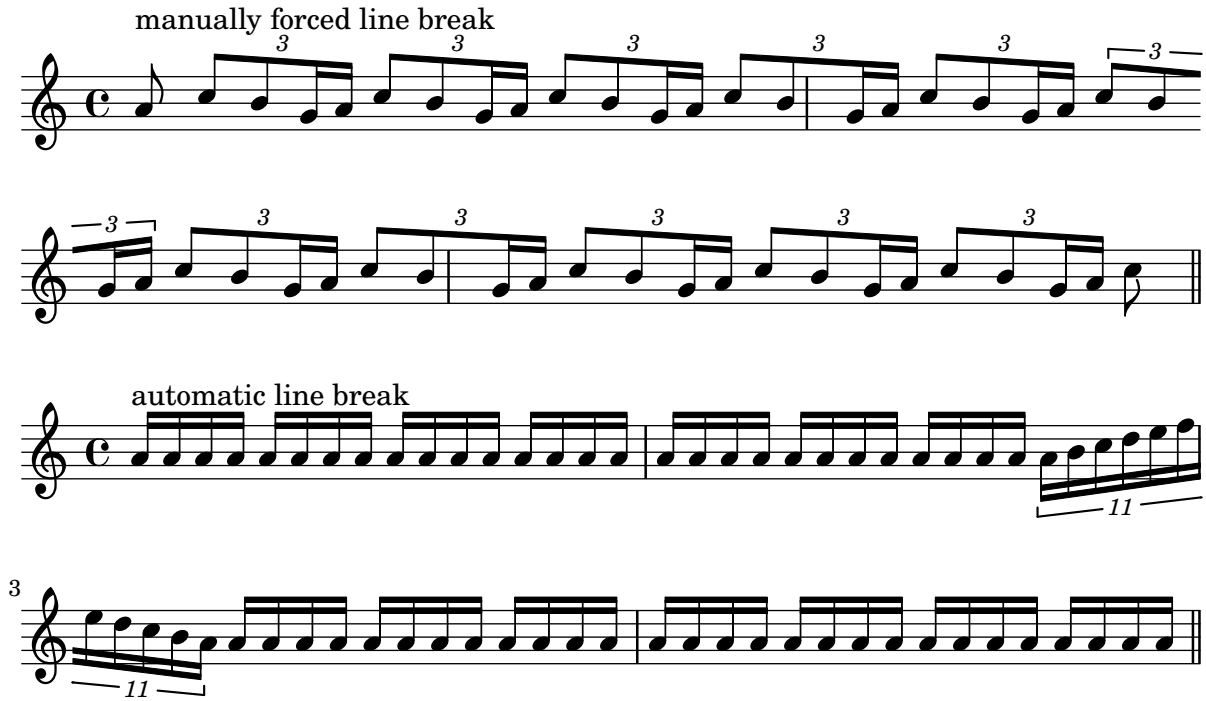
Ces exemples peu académiques démontrent comment il est possible d'insérer un saut de ligne dans un n-olet portant une ligature et qui ne sauraient se découper de manière exacte.

Cette fonctionnalité n'est opérationnelle qu'avec des ligatures manuelles.

```
\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 "||"
}
```



Impression de noms d'accords identiques à la basse différente par une oblique et nouvelle basse

Voici comment imprimer une suite d'accords nommés (objets ChordNames) dont seule la basse diffère, indiquée par une oblique et cette nouvelle basse, à l'aide du graveur Scheme ici défini. Son comportement peut se gérer finement à l'aide de la propriété de contexte chordChanges.

```
#(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

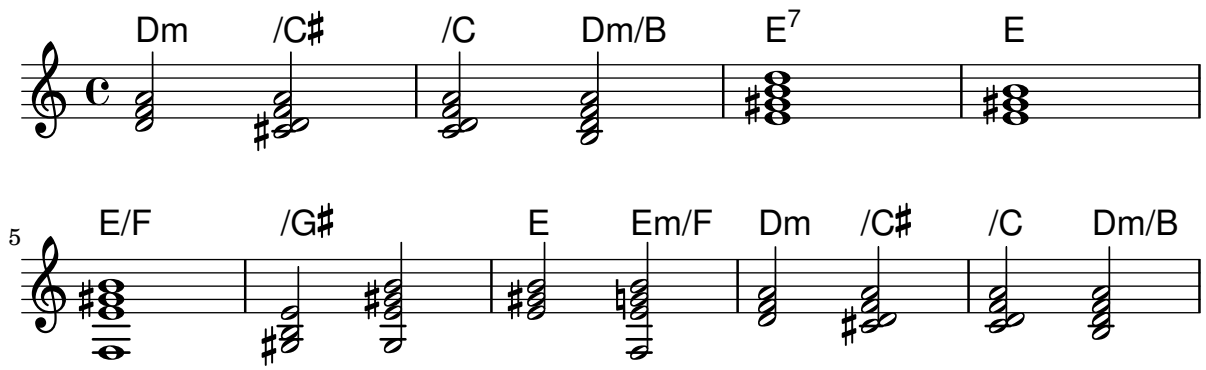
  d:m/c
  \set chordChanges = ##f
  d:m/b

  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
}

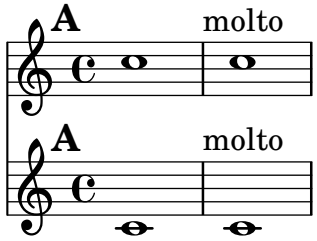
<<
  \new ChordNames
    \with { \consists #Bass_changes_equal_root_engraver }
    \myChords
  \new Staff \myChords
>>
```



Impression des indications sur toutes les portées d'un système

Bien que ces indications textuelles et repères ne soient habituellement imprimées qu'au niveau de la portée supérieure d'un système, leur affectation peut être répercutée à chacune des portées.

```
\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
    }
  }
}
```



Impression de musique aux métriques différentes

Bien qu'ayant des métriques bien différentes, les deux parties ci-dessous présentées restent synchrones.

Les barres de mesure ne peuvent plus être gérées au niveau du contexte `Score` ; le `Timing_translator` doit être déplacé du contexte `Score` au contexte `Staff` afin de permettre des barres de mesure individualisées.

Le `Bar_number_engraver` devra lui aussi être déplacé, puisqu'il dépend de propriétés attachées au `Timing_translator`, afin de numéroter les mesures. L'utilisation d'un bloc `\with` dans la portée concernée permettra un affichage des numéros de mesure.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}

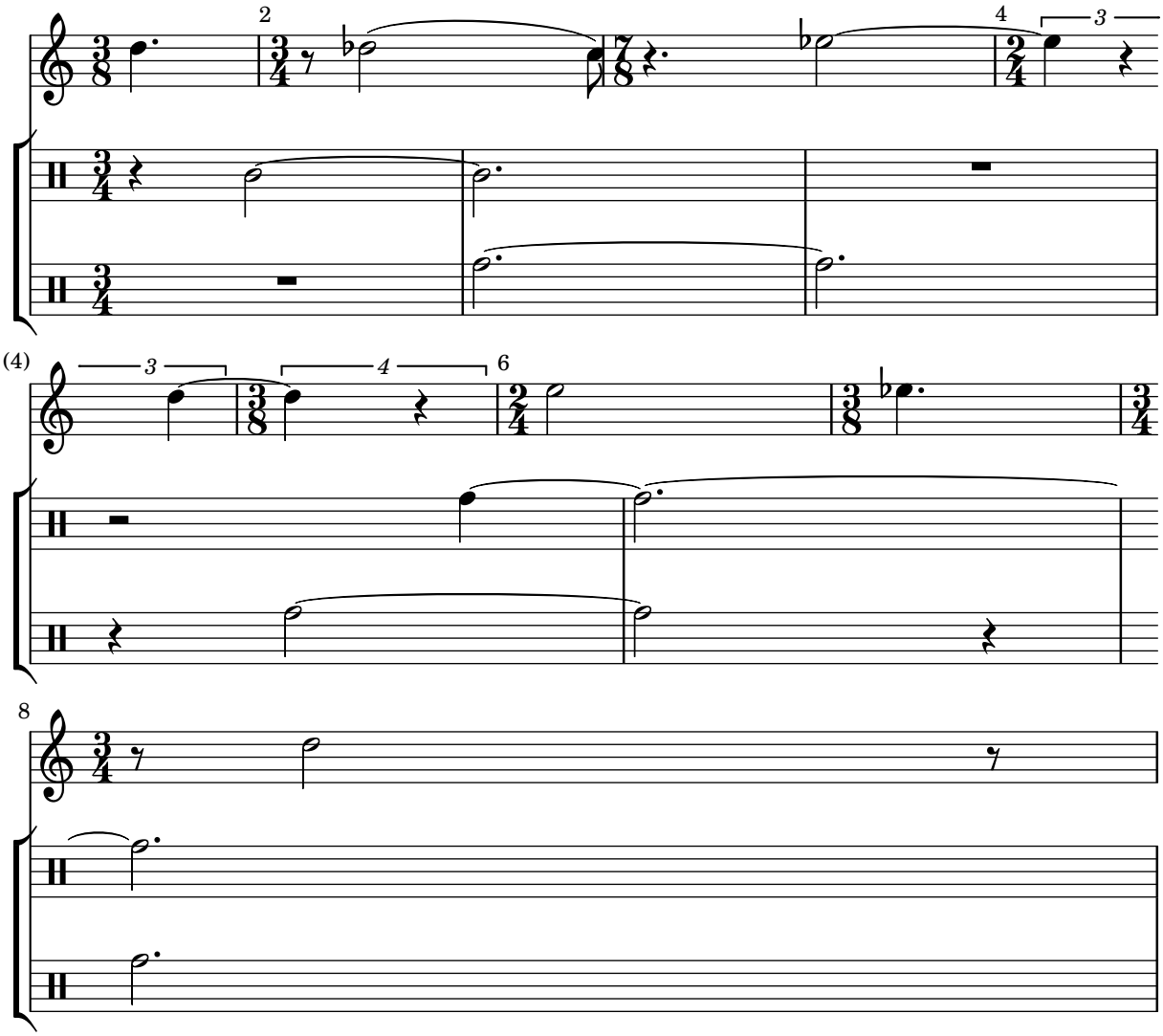
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \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
  >>
}
```
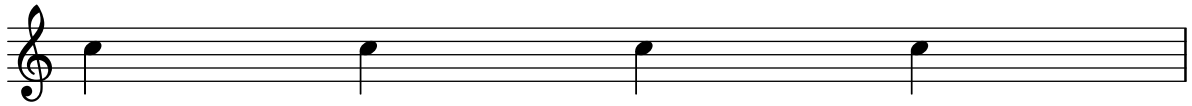


Suppression des numéros de mesure d'une partition

Désactiver le graveur concerné – `Bar_number_engraver` – donnera une partition – contexte `Score` – sans numéros de mesure.

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

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

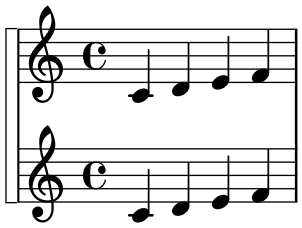




Indication de regroupement de portées par un rectangle

Un regroupement de portées sera indiqué par un simple rectangle – `SystemStartSquare` – en début de ligne dès lors que vous le mentionnerez explicitement au sein d'un contexte `StaffGroup` ou `ChoirStaff`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



Utilisation de lignes de repères dans une partition « à la française »

Le recours à des contextes `MarkLine` (tel qu'indiqué dans Inscrire les repères ailleurs que sur la première portée (`https://wiki.lilypond.community/wiki/Placing_rehearsal_marks_other_than_above_the_top_staff`)) dans une partition « à la française » peut s'avérer problématique lorsque toutes les portées entre deux contextes `Markline` sont absentes d'un système. La présence du `Keep_alive_together_engraver` au sein de chacun des `StaffGroup` permet de maintenir actif le `MarkLine` uniquement lorsque les autres portées du regroupement sont elles-mêmes actives.

```
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 = {
```

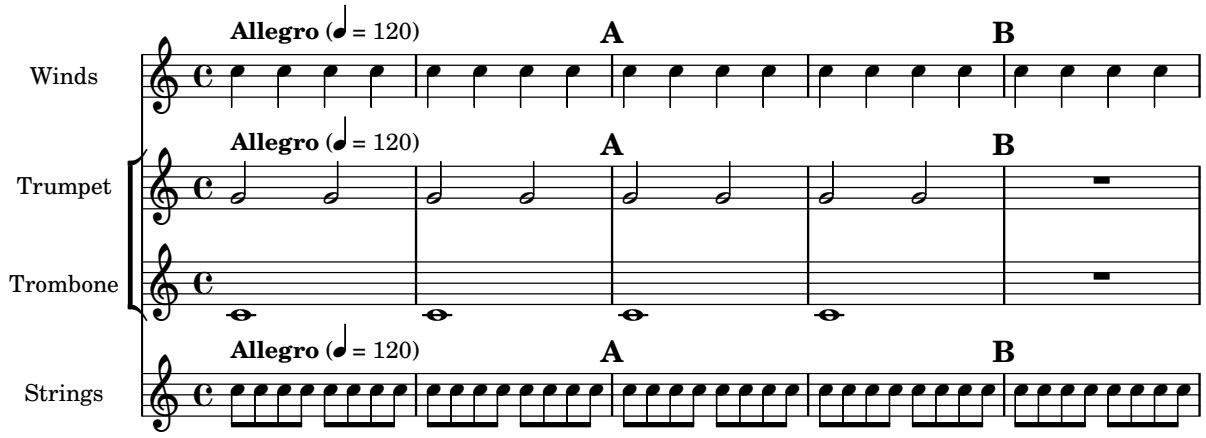
```
  \*4 c'1
  R1*8
  d'1
  R1*17
}
strings = \*240 c''8
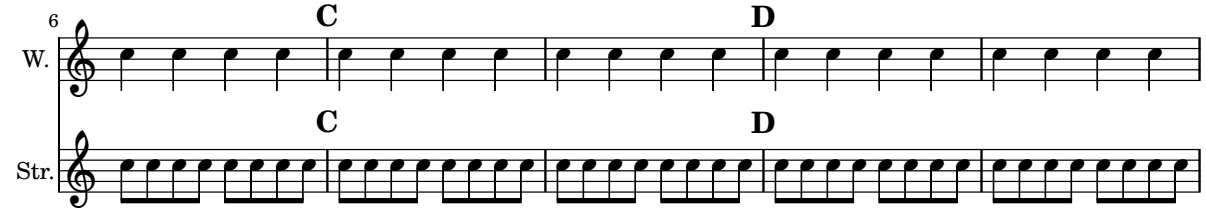
#(set-global-staff-size 16)
\paper {
  systems-per-page = 5
  ragged-last-bottom = ##f
  tagline = ##f
}
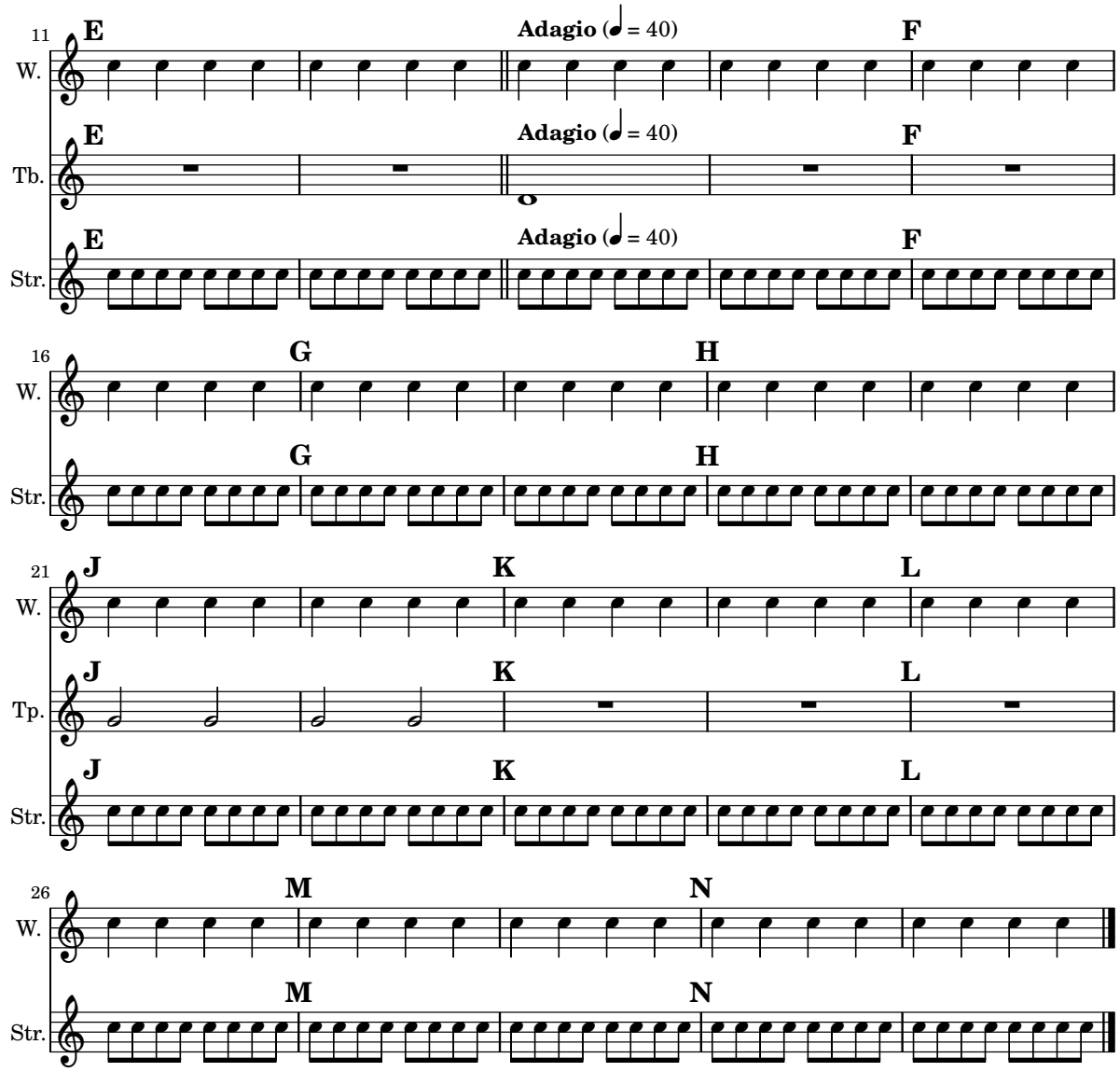
\layout {
  indent = 16\mm
  short-indent = 5\mm
  \context {
    \name MarkLine
    \type Engraver_group
    \consists Apply_output_engraver
    \consists Axis_group_engraver
    \consists Mark_engraver
    \consists Metronome_mark_engraver
    \consists Staff_collecting_engraver
    \override VerticalAxisGroup.remove-empty = ##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 }
    >>
  >>
}
```







Recours aux balises pour produire une partition ancienne et moderne à partir de la même source

Grâce aux balises (*tags*), il est possible d'utiliser une même source pour produire une partition de musique mensurale et moderne. Dans cet exemple est créée la fonction `menrest` qui permettra de positionner les silence comme dans la version originale, tout en respectant leur position sur une portée standard.

Les balises permettent aussi de gérer d'autres différenciations selon les besoins, comme des « mesures de silence » (R1, R\breve, etc.) en notation moderne mais des silences normaux (r1, r\breve, etc.) en notation ancienne. L'action de convertir de la musique mensurale en version moderne est communément appelée « transcription ».

L'appel à `c4.\Be c8 c\Am` revient au même que `c4.[ c8 c]`. Néanmoins, il évite les avertissements si le début intervient sur une note dépourvue de ligature mais pourrait s'en voir affublée de par l'utilisation du `Completion_heads_engraver`.

[La ligne légèrement plus courte dans la portée mensurale permet d'éviter que le glyphe de custode ne soit raccourci lors de la génération des images. On peut l'éviter à l'aide d'un `\with-true-dimensions` comme ici.]

```
\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
}

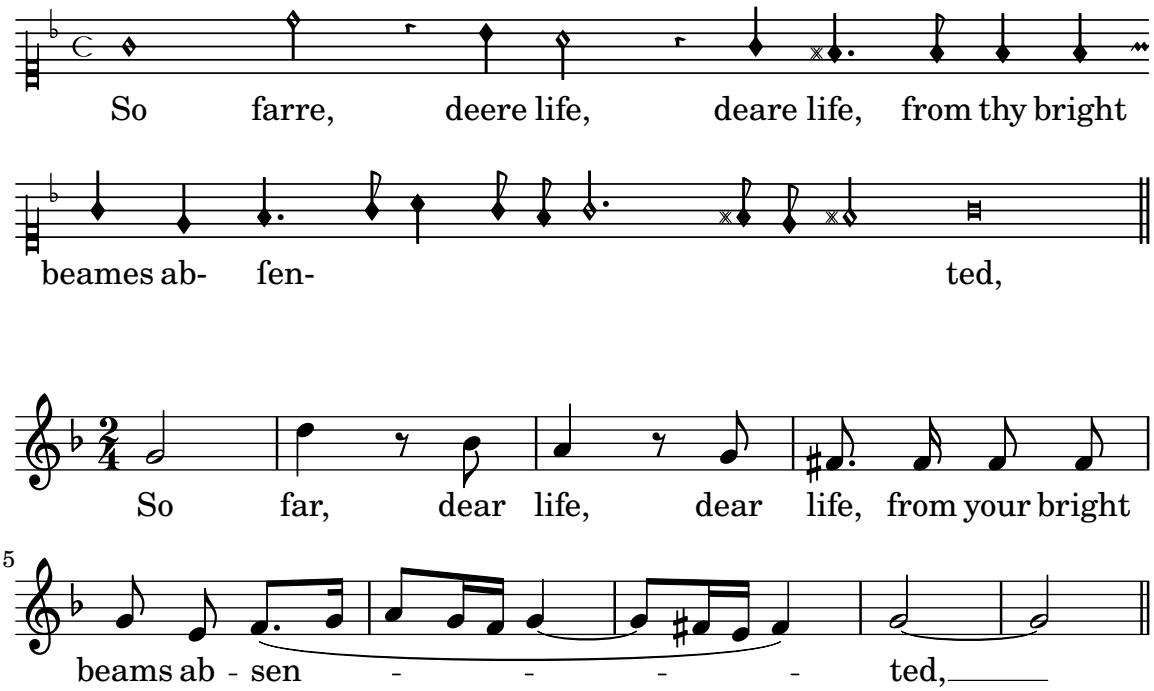
MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}

\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
    >>
  }
}
```



Ensemble vocal avec couplet et refrain

Ce canevas illustre la manière d'agencer une œuvre vocale où le couplet est chanté en solo et le refrain à deux voix. Vous noterez le recours aux silences invisibles dans la variable `\global` : ils permettent de positionner les changements de métrique et autres éléments communs à toutes les parties, ce pour l'intégralité du morceau.

```
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 |
}
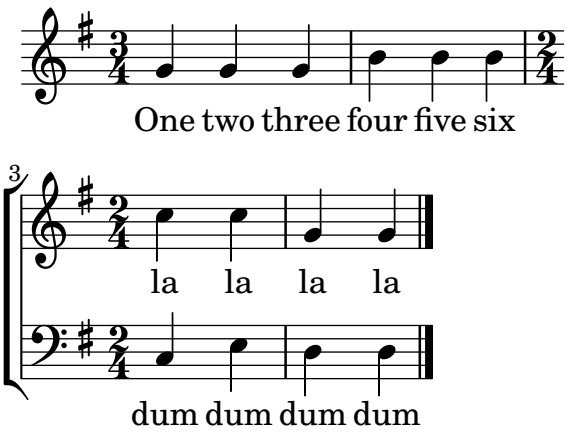
BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

\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
    >>
  >>

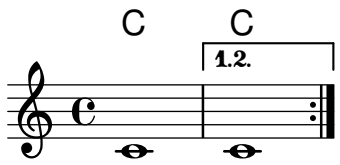
  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}
```



Crochet de reprise sous les chiffrages d'accord

L'ajout du `Volta_engraver` à la bonne portée permet d'imprimer les crochets de reprise entre les chiffrages et la portée.

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

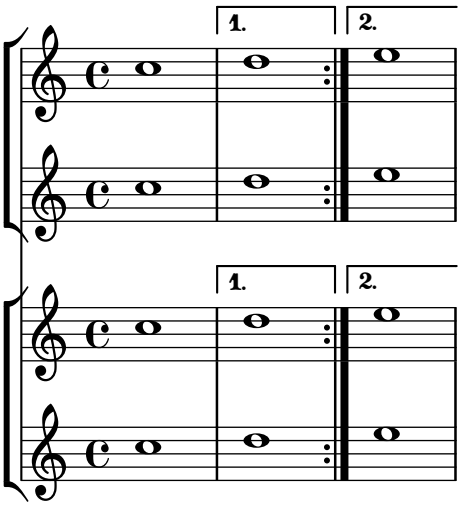


Crochets de *volta* sur plusieurs portées

L'ajout du `Volta_engraver` à une portée particulière permet que les crochets de *volta* s'affichent sur d'autres portées que la première d'une partition.

```
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.

23 Education

Quadrillage temporel : mise en évidence du rythme et synchronisation des notes

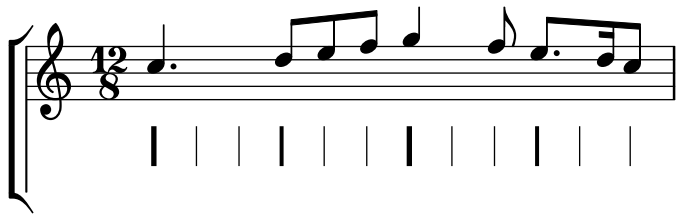
Des lignes verticales entre les portées peuvent s'ajouter dans le but d'indiquer la synchronisation entre des notes. Dans le cas de musique monophonique, on peut toutefois créer une deuxième portée, invisible, et raccourcir les lignes comme ici.

```
\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)
```
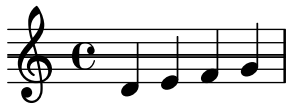
```
  }
}
```



Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane, ou bien pour mettre en exergue la ligne portant la clef de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doit être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



24 Headword

Ancient headword

Ancient headword.

```
#(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

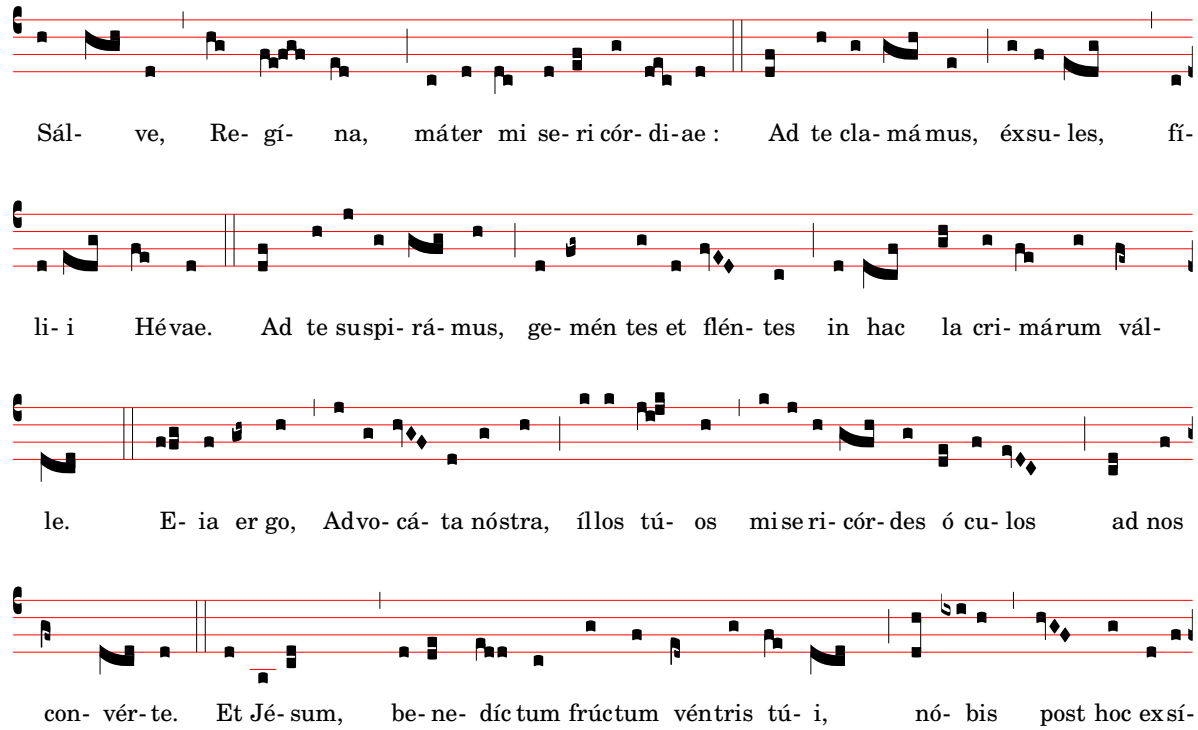
% Verse 5 --- Eia ergo, Advocáta nóstra
\[ f\melisma f \pes g\melismaEnd \] f
  \[ g\melisma \pes \deminutum a\melismaEnd \] a
  \divisioMinima
c' g \[ \virga a\melisma \inclinatum g \inclinatum f\melismaEnd \]
  d g a
  \divisioMaior
d' d' \[ c'\melisma \flexa b c' \pes d'\melismaEnd \] a
  \divisioMinima
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

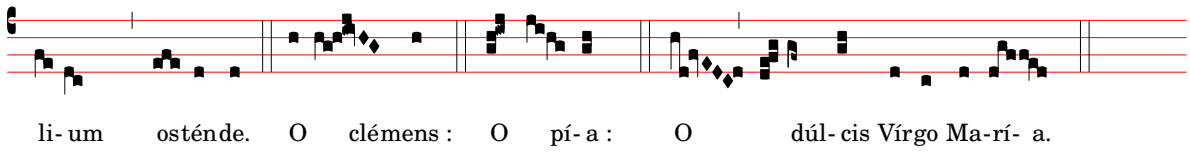
% Verse 6 --- Et Jésum
d a, \[ c\melisma \pes d\melismaEnd \]
  \divisioMinima
d \[ d\melisma \pes e\melismaEnd \]
  \[ e\melisma \flexa d d\melismaEnd \]
  c g f \[ e\melisma \flexa \deminutum d\melismaEnd \] g
  \[ f\melisma \flexa e\melismaEnd \]
  \[ d\melisma \flexa c \pes d\melismaEnd \]
  \divisioMaior
\[ d\melisma \pes a \] \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.
 }
>>
```





Chords headword

Chords headword.

```
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 |
}

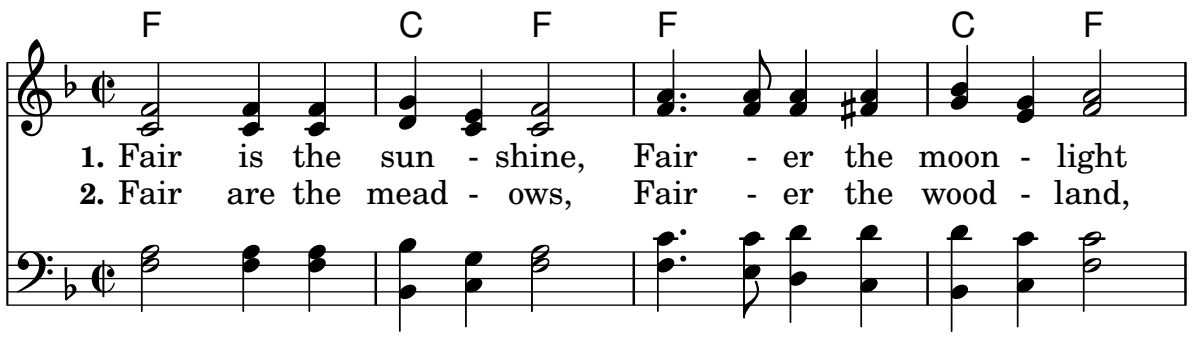
Tenor = {
  \key f \major
  \stemDown
  a2 a4 a | bes4 g a2  | c'4. c'8 d'4 d' | d'4 c' c'2 |
  a2 d'4 bes | a2 c'4 c' | c'2 c'  | c'1 |
}

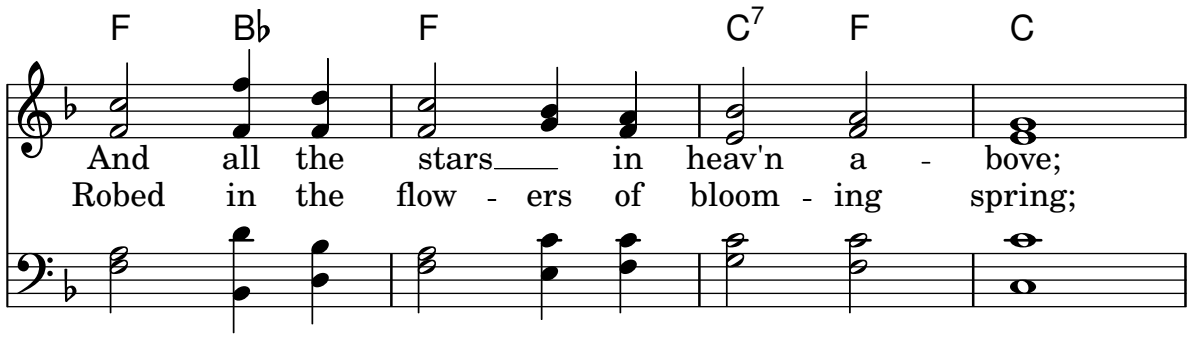
Bass = {
  \key f \major
  f2 f4 f | bes,4 c  f2 | f4. e8 d4 c | bes,4 c f2 |
```

```
  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
    }
  }
}
```





Editorial headword

Editorial headword.

```
\include "english.ly"

% Beethoven, Op. 31, No. 3
% Piano sonata 18, Movt II, Scherzo
% Measures 9 - 14

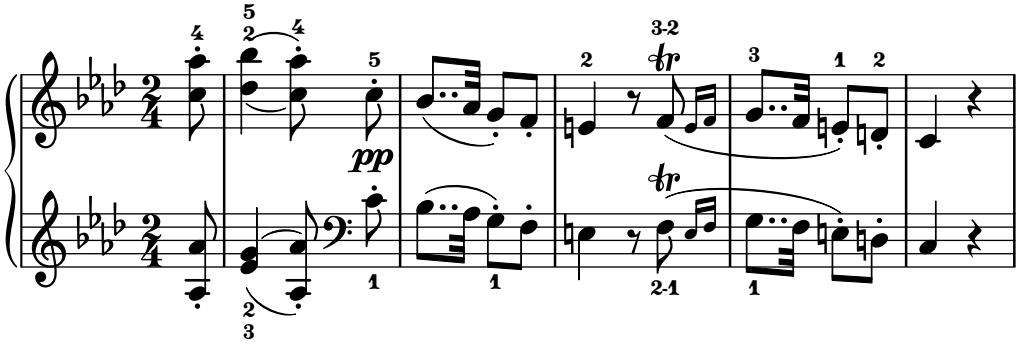
\new PianoStaff <<
  \new Staff = "right hand" {
    \clef treble
    \key af \major
    \time 2/4
    \set Staff.fingeringOrientations = #'(up)
    \set Score.currentBarNumber = #9

    \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 |
```

```
  }
>>
```



Expressive headword

Expressive headword.

```
% L. v. Beethoven, Op. 49 no. 1
% Piano sonata 19 - "Leichte Sonate"
% measures 1 - 12

\include "english.ly"

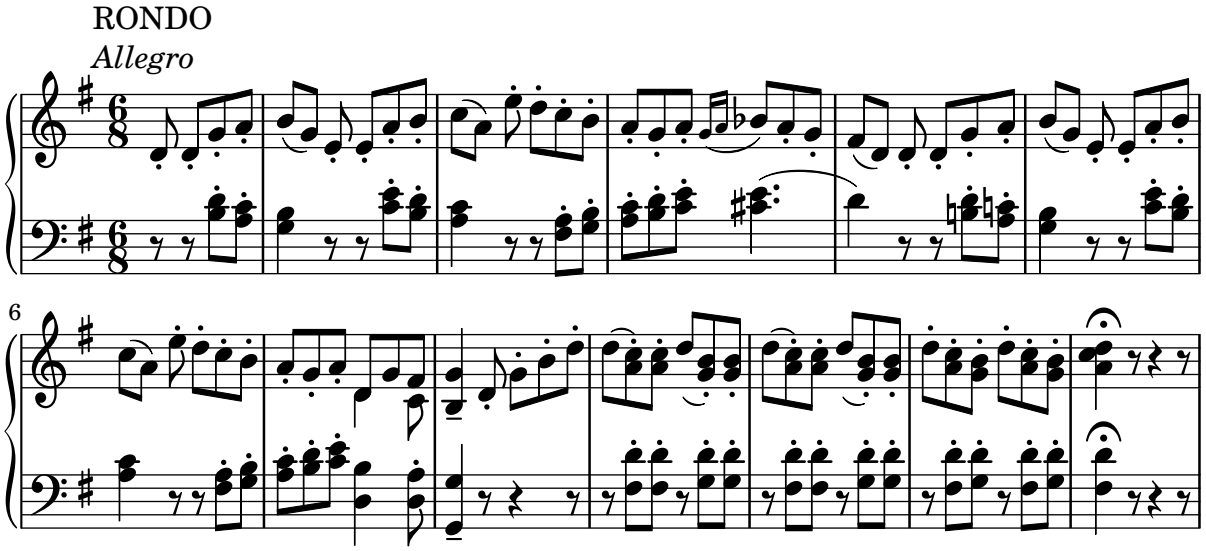
\new PianoStaff <<
  \new Staff = "right hand" {
    \clef treble
    \key g \major
    \time 6/8

    \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 |
  }
>>
```



Figured bass headword

Figured bass headword.

```
% 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'' |
  }
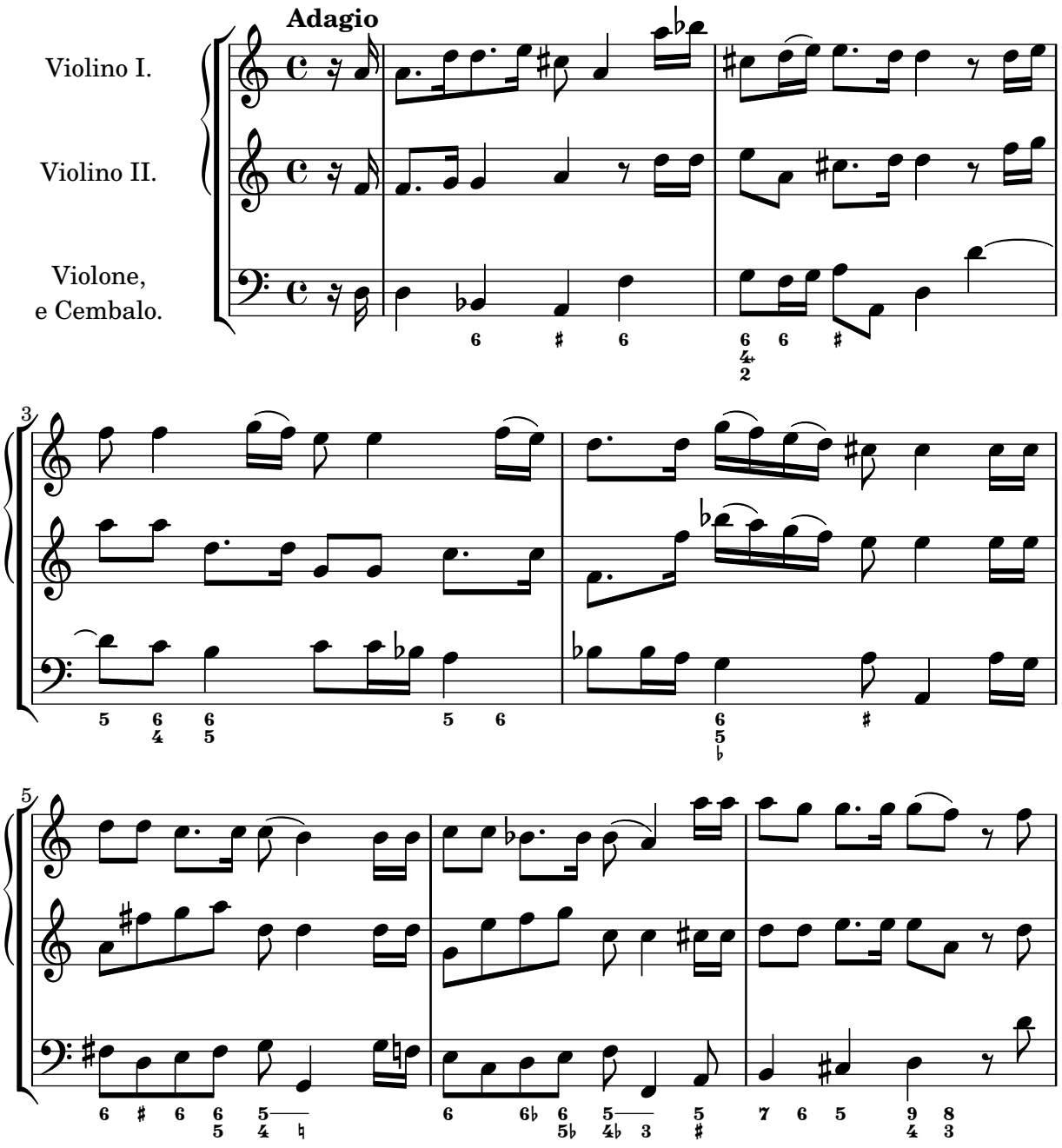
  \new Staff = "violinoII" \with { instrumentName = "Violino II." }
  {
    \time 4/4
    \partial 8 r16 f' |
    f'8. g'16 g'4 a' r8 d''16 d'' |
    e''8 a' cis''8. d''16 d''4 r8 f''16 g'' |
    a''8 a'' d''8. d''16 g'8 g' c''8. c''16 |
    f'8. f''16 bes''( a'') g''( f'') e''8 e''4 e''16 e'' |
    a'8 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
}
```



Fretted headword

Fretted headword.

```
% 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

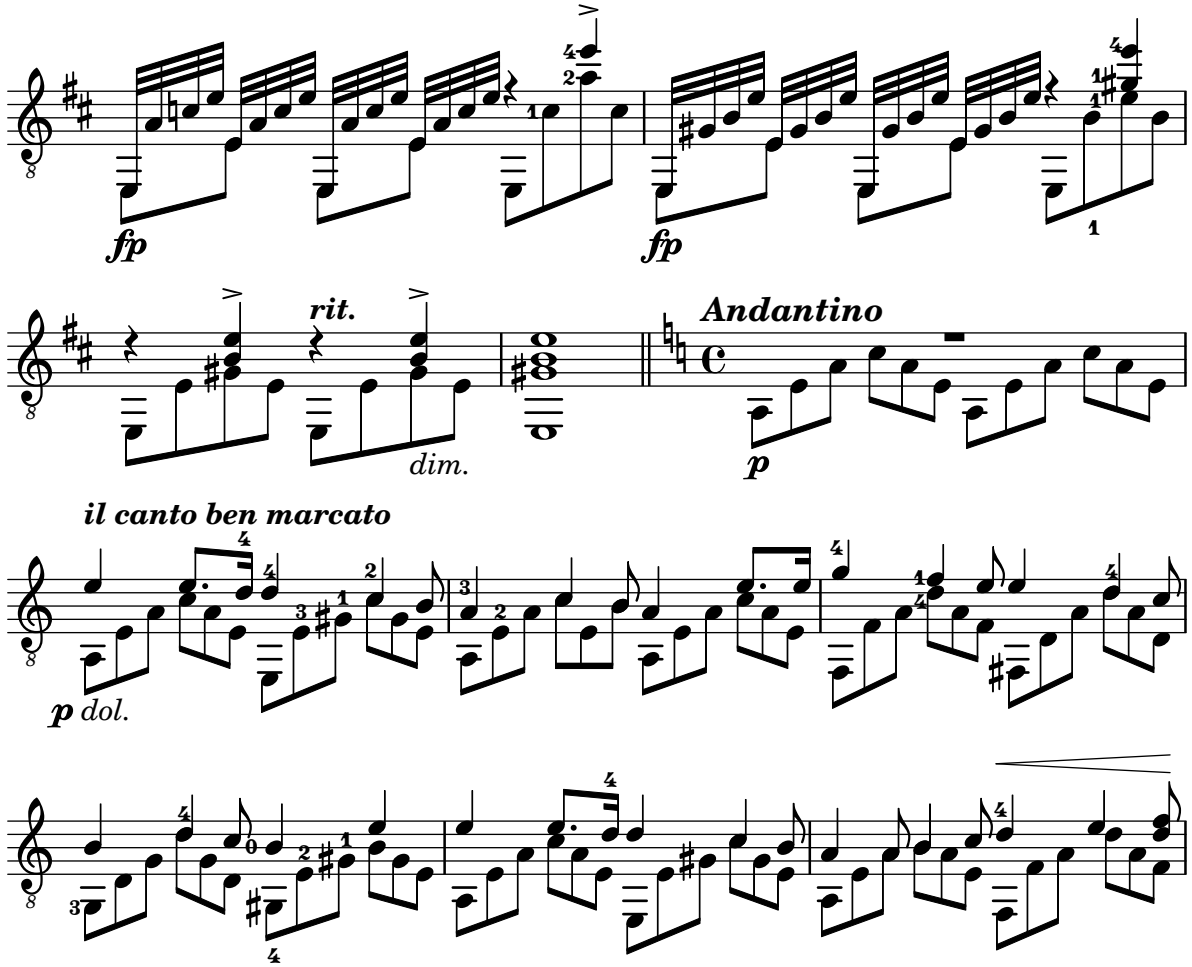
  e,8\fp[ e'] e,[ e'] e, \sfol <c''-1> <a'-2> c, | % m. 1
  e,,8\fp[ e'] e,[ e']
    e, \sfod \FO #0.2 #-0.2 <b''-1> \sfol \FO #0.3 #0.0 <e-1> b | % m. 2
  e,,8 e' gis e e, e' gis_\dimin e | % m. 3
  e,1 | % m. 4

  \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
}
```



Keyboard headword

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
    }
  }
}
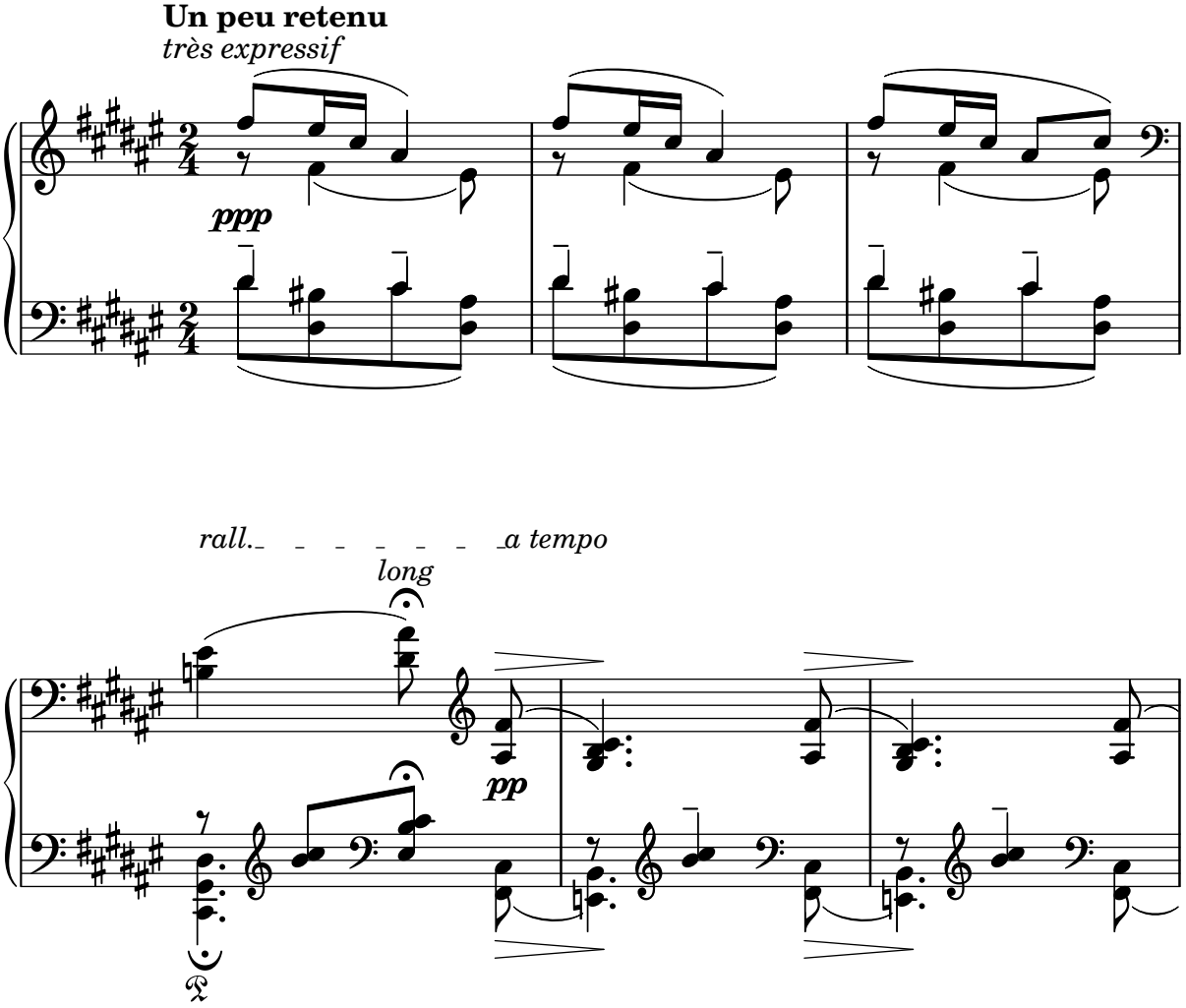
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    \time 2/4
    \key fs \major
    <<
      \new Voice {
        \voiceOne
        \textMark \markup {
          \override #'(baseline-skip . 2.4) \column {
            \line \bold { Un peu retenu }
            \line \italic { 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( |
```
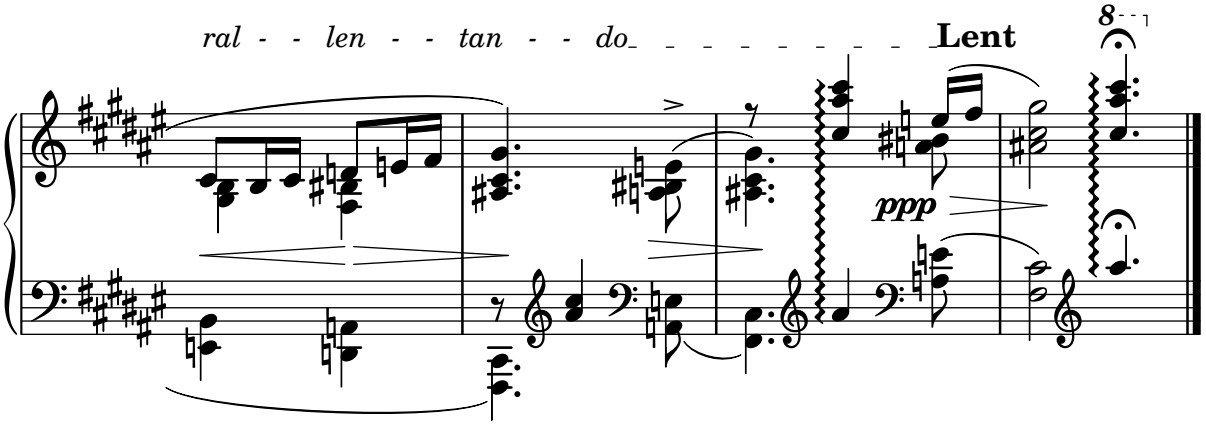
```
      \override TextSpanner.bound-details.left.text =
        "ral  -   -   len   -   -   tan   -   -   do"
        \override TextSpanner.bound-details.right.text =
          \markup \larger \upright \bold "Lent"
      cs'8\<\startTextSpan b16 cs'
        d'8\tweak to-barline ##f \> e'16 fs' |
      <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
}
```





Pitches headword

Pitches headword.

```
% 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 |
    }
  >>

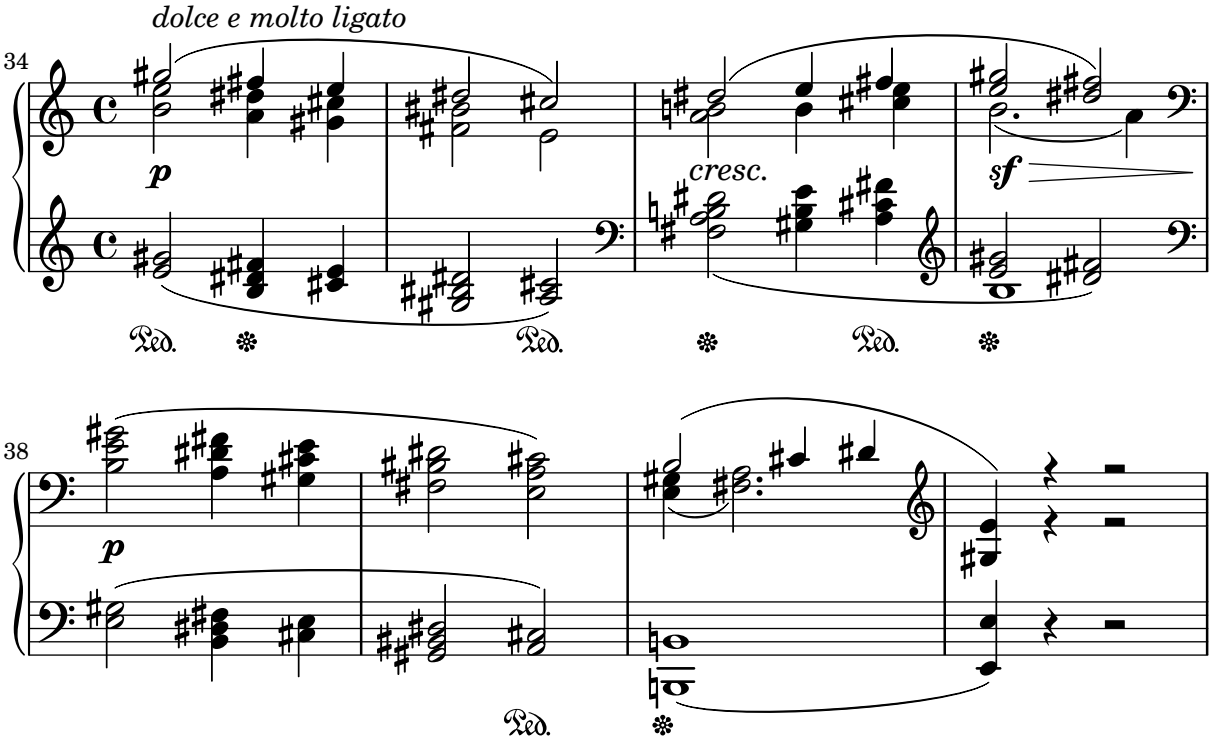
  \new Dynamics {
```

```
    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 |
  }
>>
```



Repeats headword

Repeats headword

```
% 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
    }
  }

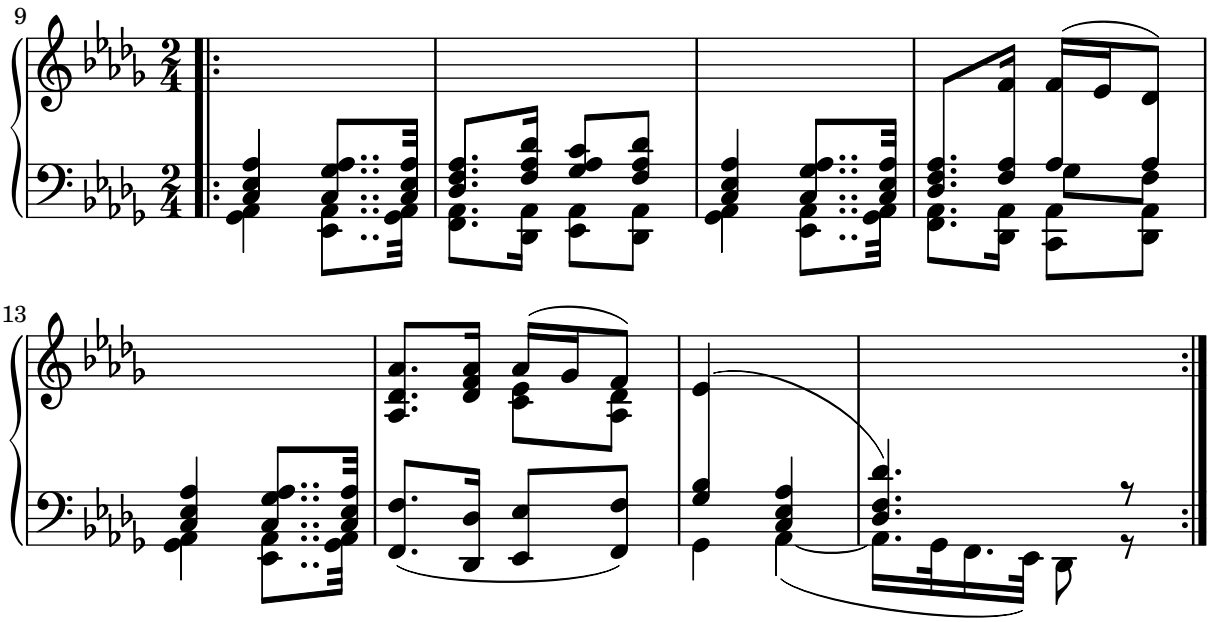
  \new Staff = LH {
    \clef bass
    \key df \major
    \time 2/4

    \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"
  }
}
```



Rhythms headword

Rhythms headword.

```
% 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''>) |
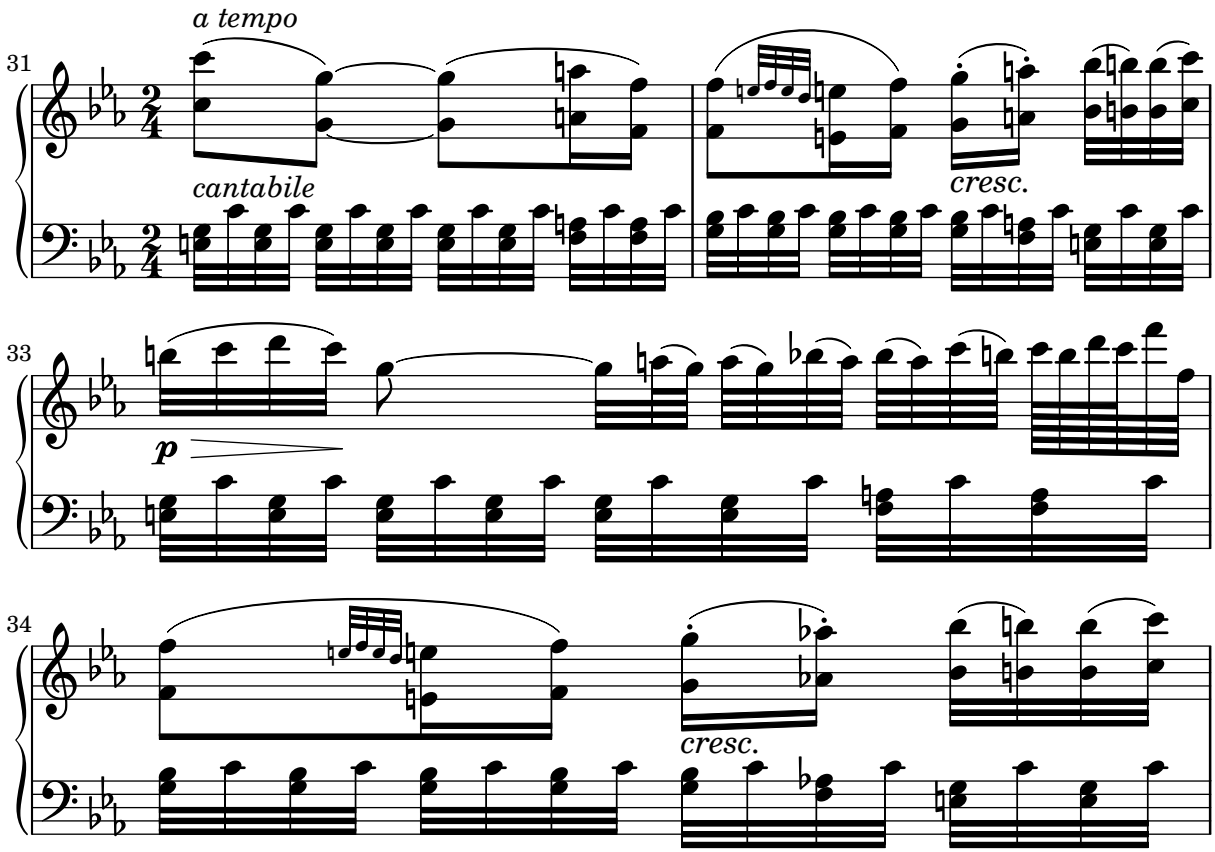
    b''32([ c''' d''' c'''32)] g''8 ~
      g''32[ a''64( g'']) a''64([ g'') bf''( a''64)]
      bf''64([ a'') c'''( b''64)] c'''128[ b'' d''' c''' f'''64 f''] |
    <f'' f'>8[( \shiftedGrace #-0.15 { e''32 f'' e'' d''] }
      <e''! e'>16 <f'' f'>]) <g'' g'>16-.([ <af''! af'!>-.)]
      <bf'' bf'>32( <b'' 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
  }
}
```



Simultaneous headword

Simultaneous headword.

```
\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 |
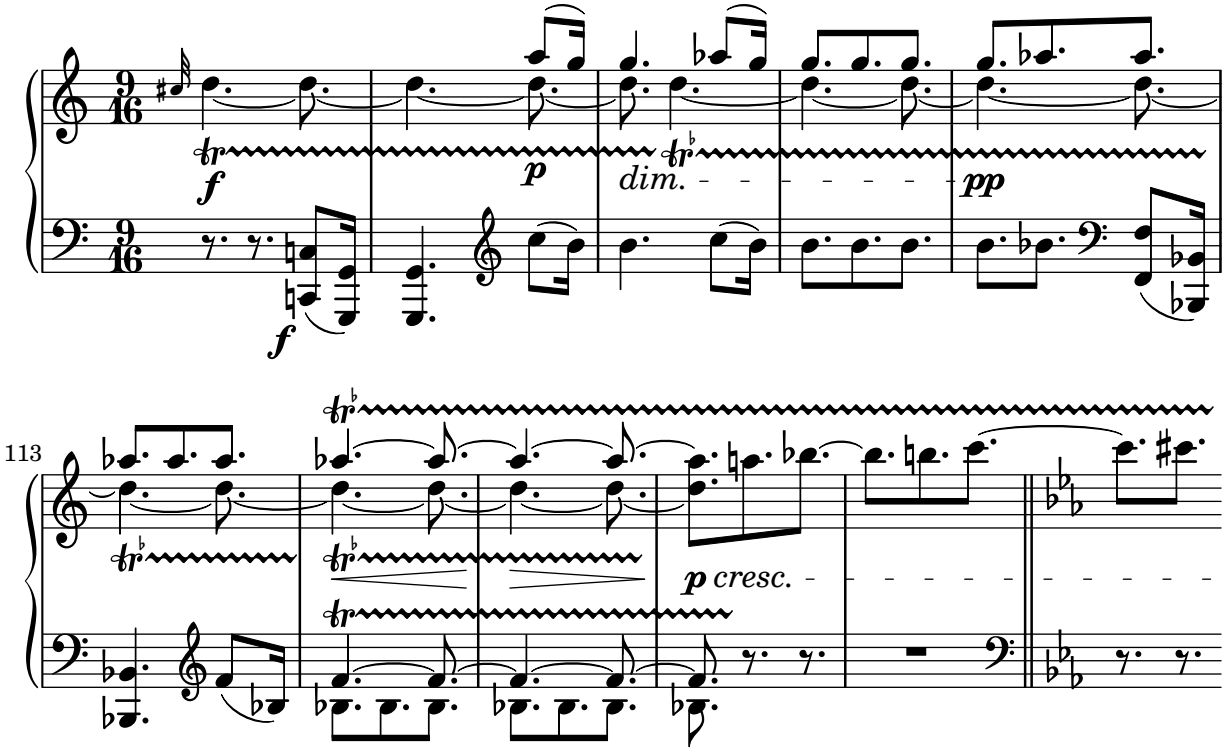
    \trillFlat d''4.\startTrillSpan ~ d''8. ~ |
    d''4. ~ d''8. ~ |
    \once \override NoteColumn.ignore-collision = ##t
      \hideNotes d''8.\stopTrillSpan s4. |
    s4. s8. |
    s4.
  }
>>

\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.
  }
>>
```



Staff headword

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-. |
}

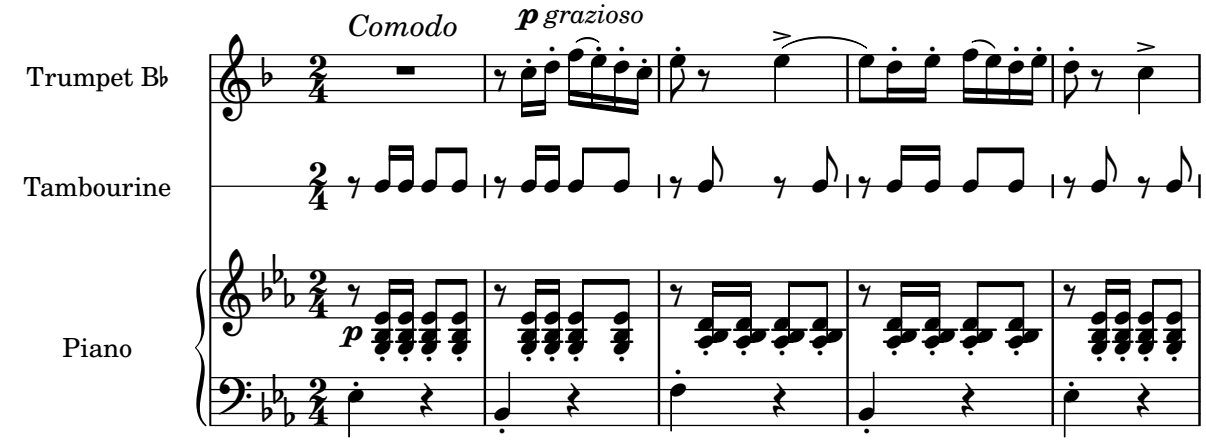
lower = \relative do {
  \clef bass
  \key mib \major
  \time 2/4

  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 }
}
```



Text headword

Text headword.

```
% L. v. Beethoven, Op. 110
% Piano sonata 31
% measures 1 - 7

\include "english.ly"

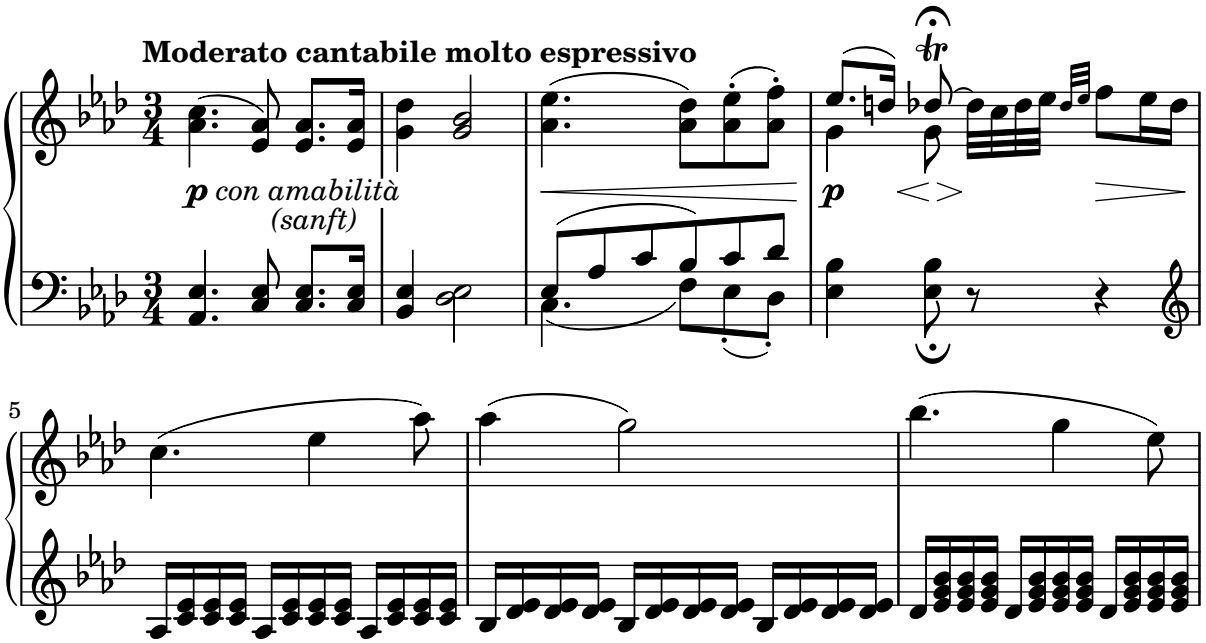
\new PianoStaff <<
  \new Staff = "right hand" {
    \clef treble
    \key af \major
    \time 3/4
    \tempo "Moderato cantabile molto espressivo"
```

```
    <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] |
  }
>>
```



Unfretted headword

Unfretted headword.

```
% 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 \<

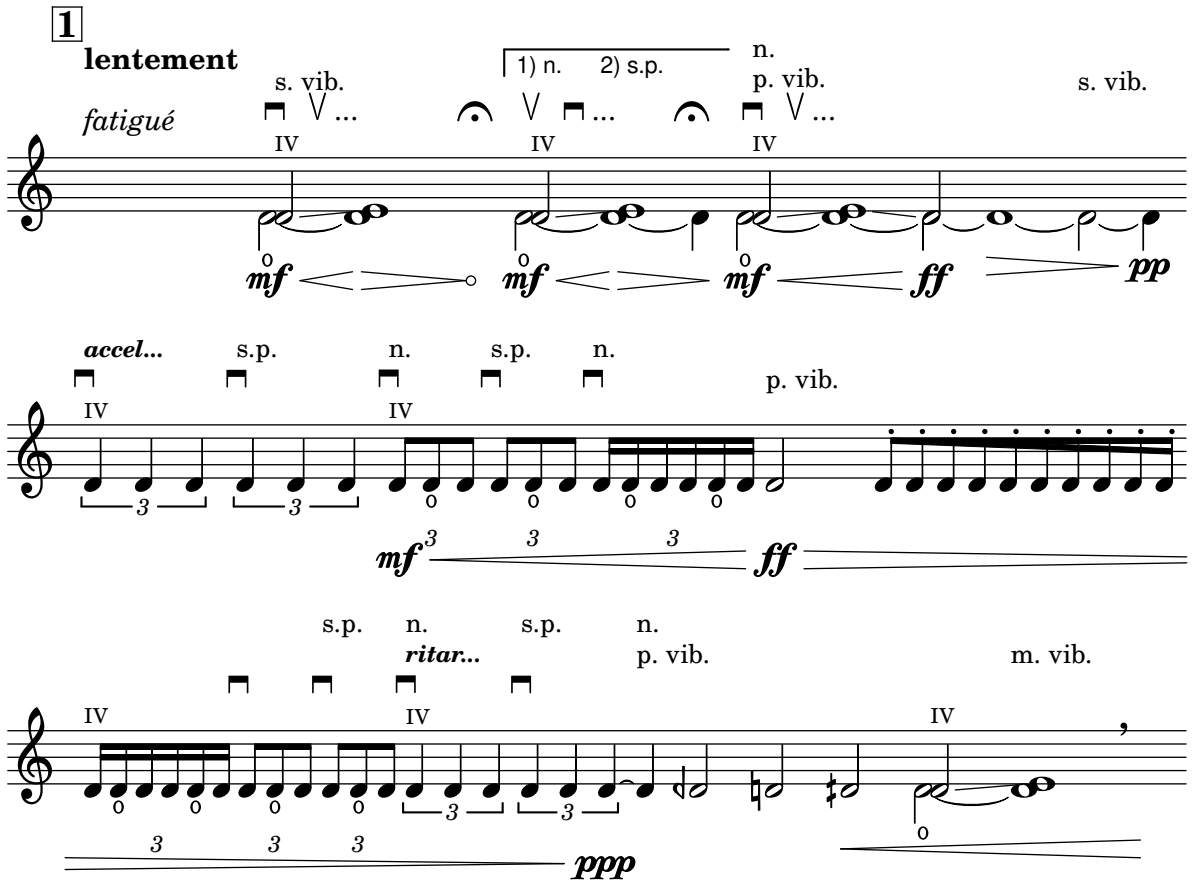
  % Measure 9
  <<
    { \shift d2\glissando^\markup \colmark { \quatre } \shifta e1 } \\
    { d2\open ~ d1^\markup \colmark { " " " " \mvib } }
  >>
  \breathe r4\!
}

\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
}
```



Vocal headword

Vocal headword.

```
% 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
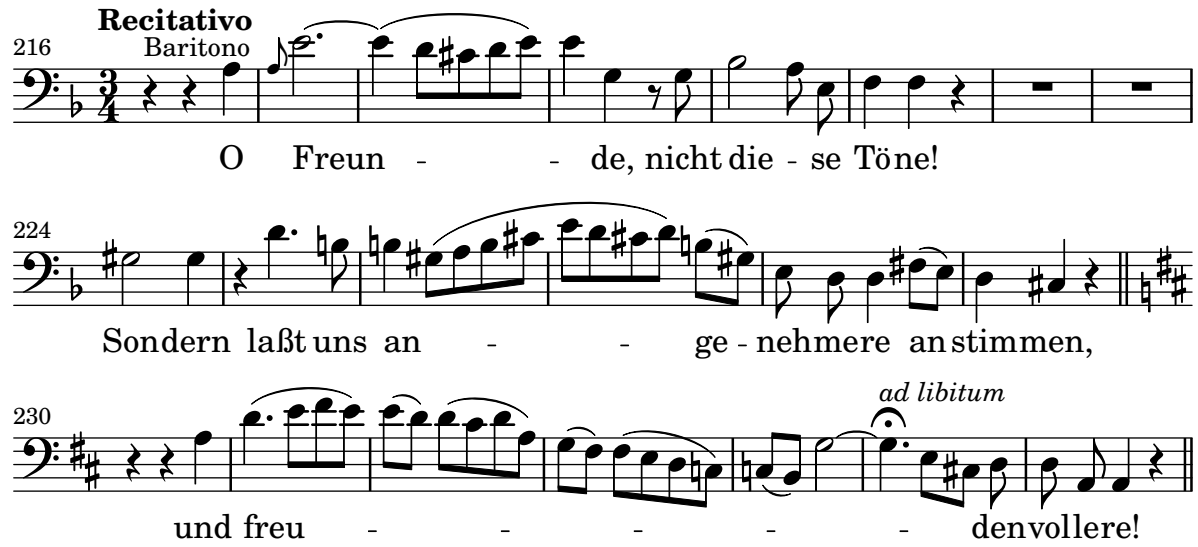
    \autoBeamOff
    \clef bass \key d \minor \time 3/4
    \tempo "Recitativo"

    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!
  }
}
```



Wind headword

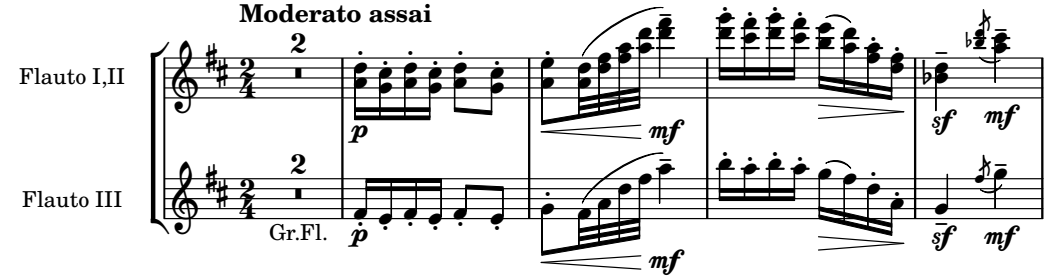
Wind headword

```
% 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
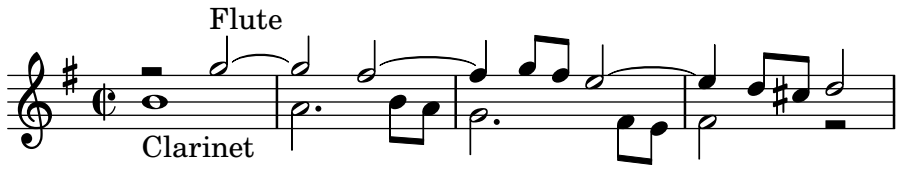
See also Section “Creating MIDI output” dans *Manuel de notation*.

Affectation d’un canal MIDI par voix

Lorsque LilyPond génère un fichier MIDI, chaque portée sera par défaut affectée à un canal, quel que soit le nombre de voix qu’elle contient. Ceci permet d’éviter de se retrouver à court de canaux, sachant qu’il n’y en a que seize de disponibles par piste.

Le fait de déplacer le `Staff_performer` dans le contexte `Voice` permet d’affecter à chaque voix d’une même portée un canal MIDI spécifique. Dans l’exemple suivant, la même portée donnera lieu à deux canaux MIDI différents, chacun étant affecté de son propre `midiInstrument`.
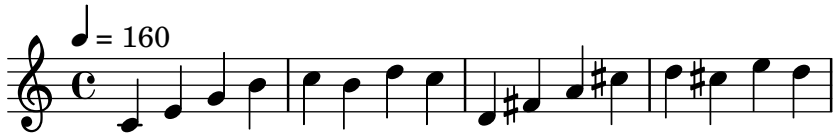
```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = "flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = "clarinet"
      \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
  }
}
```



Changement de tempo sans indication sur la partition

Vous pouvez indiquer un changement de tempo pour le fichier MIDI sans pour autant l'imprimer. Il suffit alors de le rendre invisible aux musiciens.

```
\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 { }
}
```



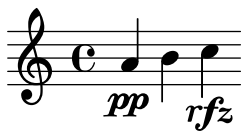
Création de nuance particulière pour la sortie MIDI

L'exemple suivant illustre la manière de créer une indication de nuance, absente de la liste par défaut, et de lui assigner une valeur spécifique utile à la sortie MIDI.

L'indication de nuance `\rfz` (*rinforzando*) se voit attribuer une valeur de `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 {}
}
```



Démonstration des instruments MIDI

Partant du problème que l'on ignore quelle valeur de `midiInstrument` serait la plus adaptée à sa composition, la compilation de ce fichier permet de faire son choix.

```
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
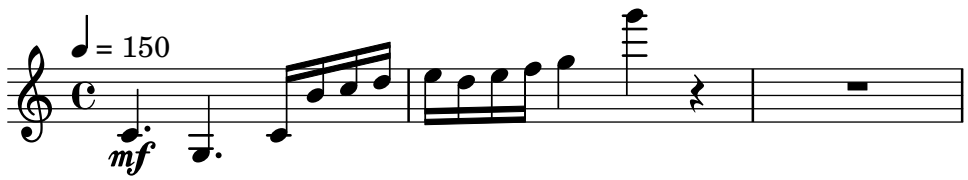
% 97-104 synth effects
\set Staff.midiInstrument = "fx 1 (rain)" \melody
\set Staff.midiInstrument = "fx 2 (soundtrack)" \melody
\set Staff.midiInstrument = "fx 3 (crystal)" \melody
\set Staff.midiInstrument = "fx 4 (atmosphere)" \melody
\set Staff.midiInstrument = "fx 5 (brightness)" \melody
\set Staff.midiInstrument = "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 { }
}
```



Réglage de l'égalisation par défaut des instruments MIDI

L'égaliseur basique peut être modifié par la définition d'une nouvelle procédure Scheme `instrumentEqualizer` au sein du contexte `Score`. Cette procédure prend en unique argument le nom d'un instrument MIDI et renverra une paire de fractions correspondant aux minimum et maximum de volume alloué à cet instrument.
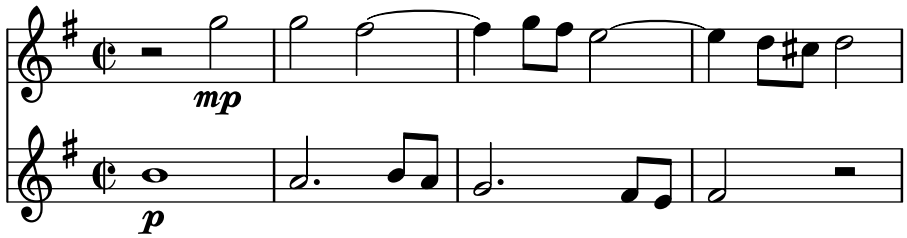
Dans l'exemple suivant sont réglés les volumes relatifs de la flûte et de la clarinette.

```
#(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

Alignement des noms d'instrument

L'alignement horizontal des noms d'instrument se gère à l'aide de la propriété `self-alignment-X` de l'objet `InstrumentName` (habitullement attaché à un contexte `Staff`). Les variables `indent` et `short-indent`, attachées au bloc `\layout`, déterminent l'espace alloué à l'alignement des noms d'instrument, respectivement dans leurs formes développée et abrégée.

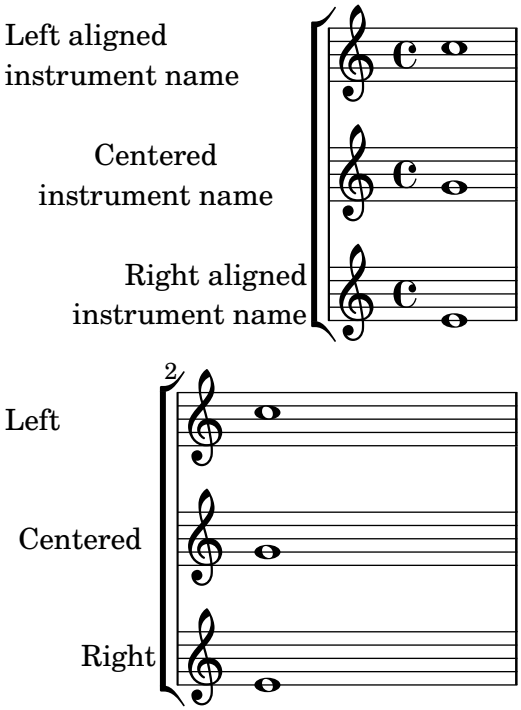
```
\paper {
  left-margin = 3\cm
}

\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
}
```



Champs d'entête

Voici la liste de tous les champs d'entête définis par défaut dans LilyPond. Grâce à l'activation de `print-all-headers`, beaucoup plus de champs s'affichent, indiquant la hiérachie des blocs `\header`.

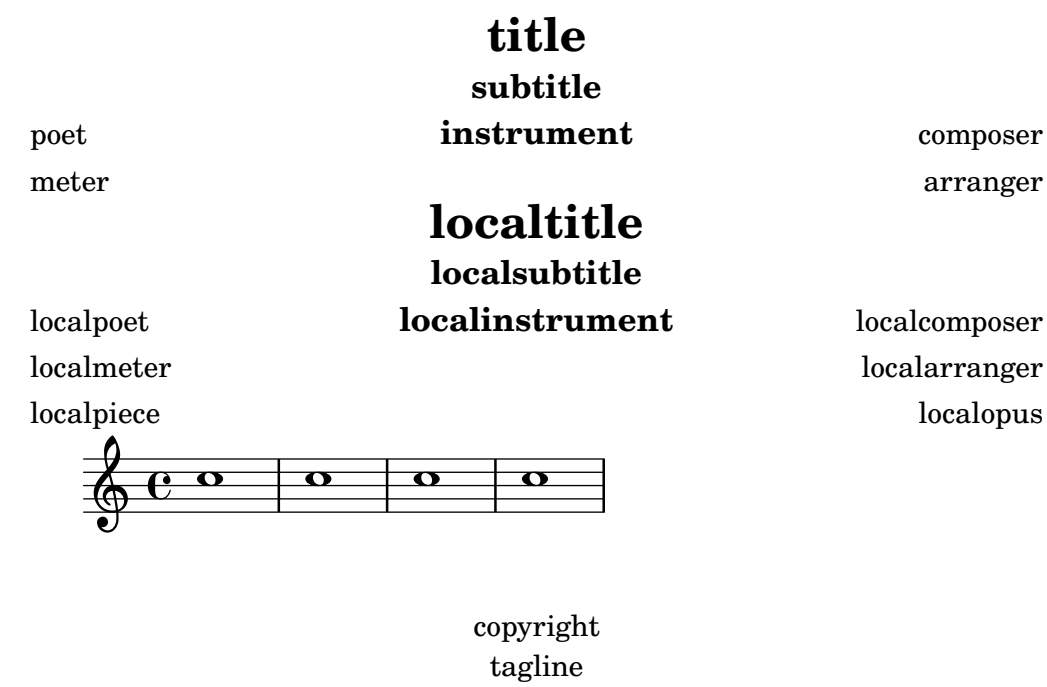
```
\paper {
  #(set-paper-size "a6" 'landscape)
  print-all-headers = ##t
}

\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"
      }
    }
  }
}
```
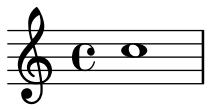


Liste des différents diagrammes de doigtés pour bois

Le code suivant permet d'obtenir une liste de toutes les possibilités en matière de doigtés pour bois, tels qu'ils sont définis dans le fichier `scm/define-woodwind-diagrams.scm`. Cette liste sera produite dans le fichier de journalisation, mais pas sous forme de musique. Pour un affichage en console, supprimez la partie `(current-error-port)` des commandes.

```
#(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 Section “Spacing issues” dans *Manuel de notation*.

Alignement des noms d’instrument

L’alignement horizontal des noms d’instrument se gère à l’aide de la propriété `self-alignment-X` de l’objet `InstrumentName` (habitullement attaché à un contexte `Staff`). Les variables `indent` et `short-indent`, attachées au bloc `\layout`, déterminent l’espace alloué à l’alignement des noms d’instrument, respectivement dans leurs formes développée et abrégée.

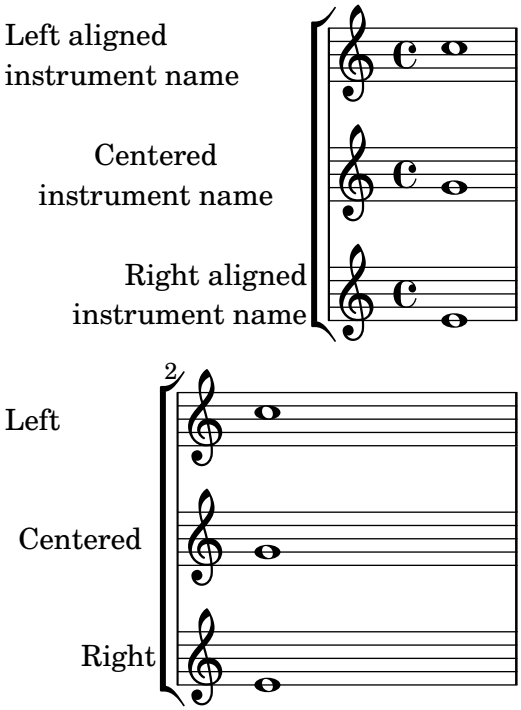
```
\paper {
  left-margin = 3\cm
}

\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
}
```



Agencement de paroles séparées sur une seule ligne

On peut vouloir positionner les paroles de différents intervenants sur une même ligne, notamment pour un dialogue par bribes. Cet extrait montre comment y parvenir en ajustant la propriété `nonstaff-nonstaff-spacing` de l'objet graphique `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
      % ...
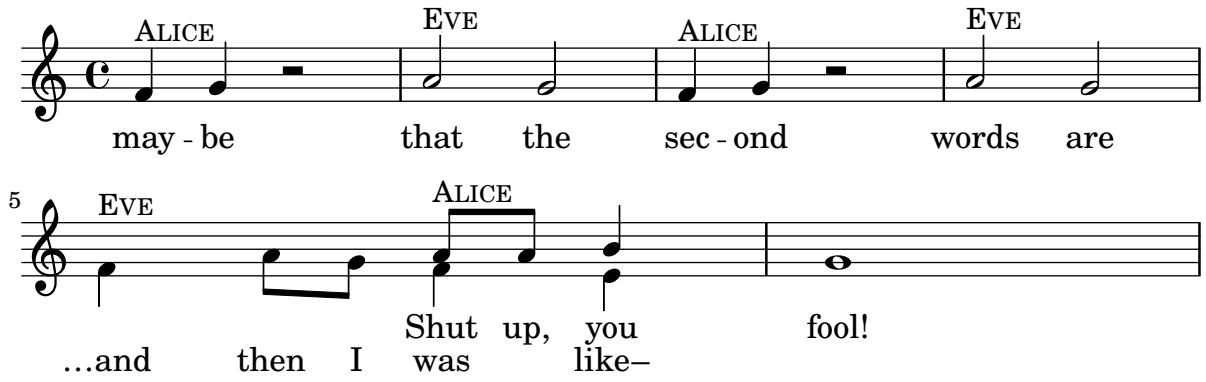
      \voiceOne
      s2 a'8^\aliceSings a' b'4 |
      \oneVoice
      g'1
    }
```

```
    \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--
  }
>>
```



Partitionnement d'un ouvrage

\bookpart permet de découper un ouvrage en plusieurs parties. La dernière page de chaque partie peut être affectée d'un ragged-last-bottom. Les routines de formatage font la distinction entre les *markups* d'entête et de pied de page selon qu'ils se rattachent à une partie ou bien à l'ouvrage entier.

```
#(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

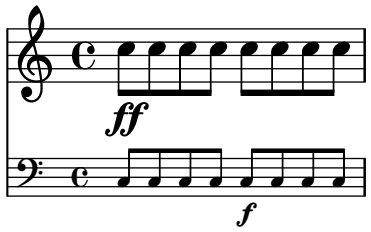
Modification de la taille d'une portée

Bien que le meilleur moyen de définir la taille des portées consiste à utiliser

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

Une portée en particulier peut se redimensionner à l'aide des propriétés `staff-space` et `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
  }
>>
```



Découpe de systèmes

Le code ci-dessous permet de réaliser des découpes à l'emporte pièce, autrement dit d'extraire des fragments, au sein d'une partition.

Dans la mesure où le rendu inclus dans ce manuel pourrait ne pas être des plus évidents, nous vous conseillons de traiter ce fichier individuellement, avec l'option `-dclip-systems`.

Les fichiers résultants sont de la forme '*base*`-from-`*début*`-to-`*fin*`[-`*compteur*`].eps`'.

Lorsque sont inclus le début ou la fin d'un système, l'extraction comportera les objets rattachés au système, comme le nom d'instrument.

Les notes d'ornement qui se trouveraient en fin de fragment ne sont pas incluses.

La découpe peut couvrir plusieurs systèmes, auquel cas seront générés plusieurs fichiers 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")
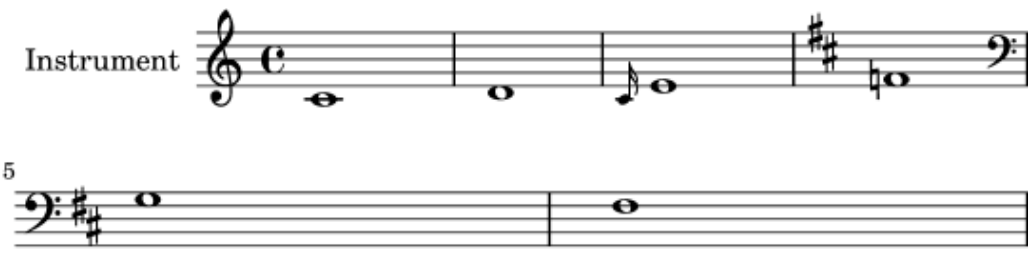
origScore = \new Staff \with { instrumentName = "Instrument" }
\relative c' {
  c1
  d1
  \grace c16 e1
  \key d \major
  f1 \break
  \clef bass
  g,1
  fis1
}
```

```
\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

Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le `Bar_number_engraver` du contexte `Score`, et les `Time_signature_engraver`, `Clef_engraver` et `Bar_engraver` du contexte `Staff`.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

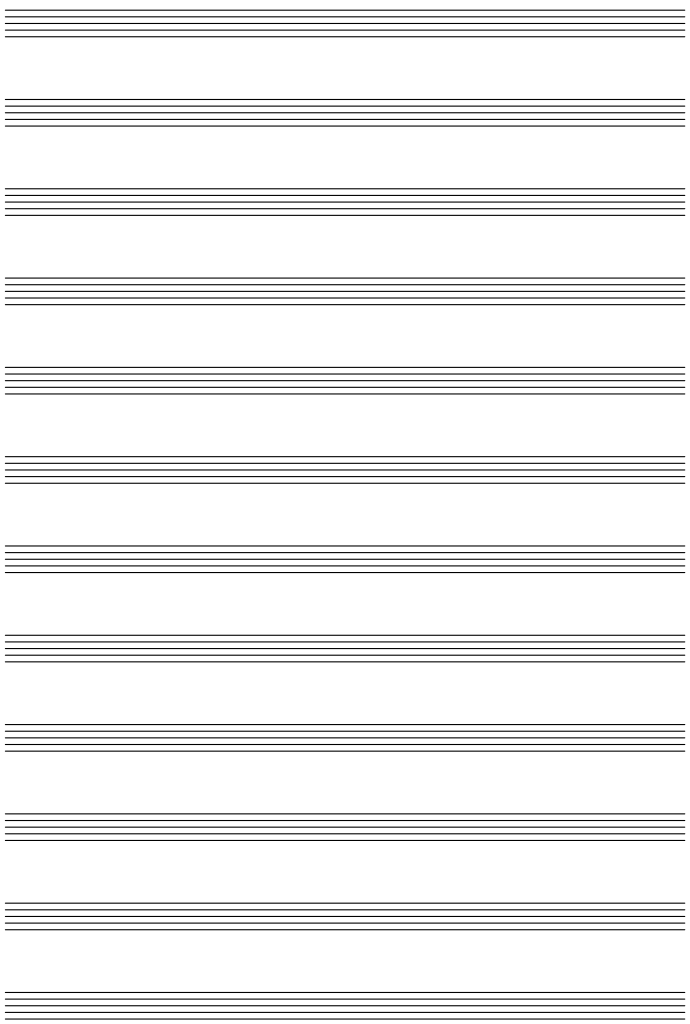
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % 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
  }
  %}
}
```



Champs d'entête

Voici la liste de tous les champs d'entête définis par défaut dans LilyPond. Grâce à l'activation de `print-all-headers`, beaucoup plus de champs s'affichent, indiquant la hiérachie des blocs `\header`.

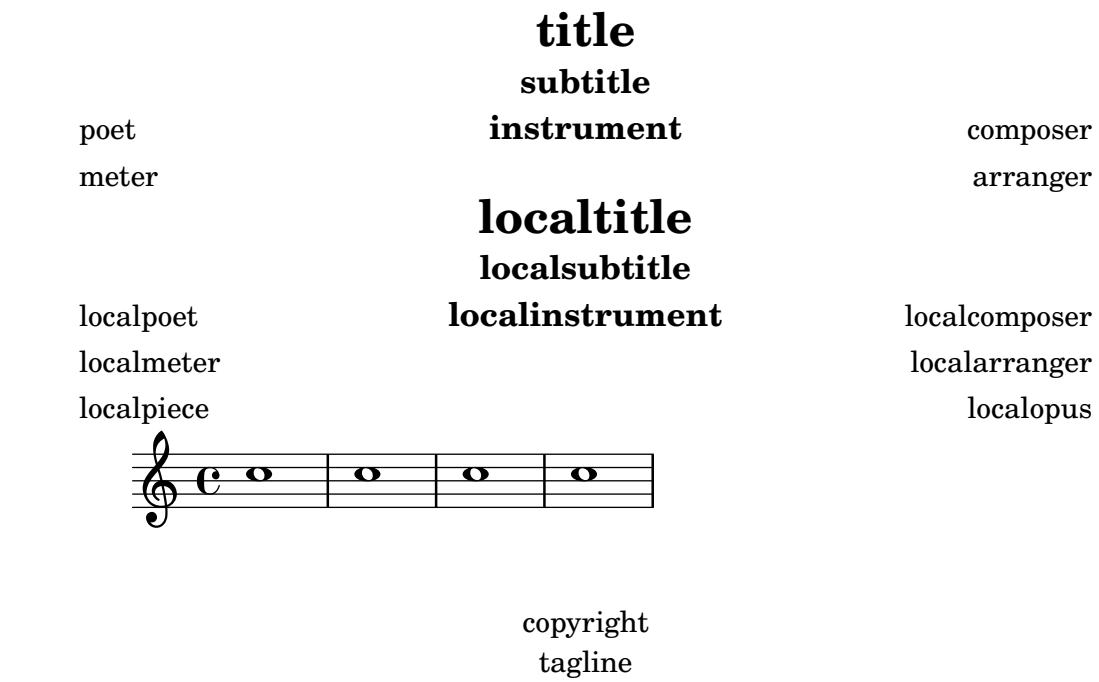
```
\paper {
  #(set-paper-size "a6" 'landscape)
  print-all-headers = ##t
}

\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"
      }
    }
  }
}
```



Affichage d'un système complet lorsqu'une seule portée est active

Dans la plupart des conducteurs d'orchestre, il est d'usage qu'un instrument ou un groupe d'instruments ne joue pas pendant un moment ; leurs portées disparaissent alors pendant ce temps – les anglophones appellent cela une « partition à la française » *(Frenched score)*. LilyPond dispose de cette fonctionnalité à l'aide de la commande `\RemoveEmptyStaves`.

Il est préférable, lorsque l'un d'entre eux reprend, d'imprimer **toutes les portées du groupe en question** grâce au `Keep_alive_together_engraver` placé au niveau du regroupement, qu'il s'agisse d'un `GrandStaff` ou d'un `StaffGroup`.

Dans l'exemple suivant, les violons ne jouent pas lors du deuxième système. Bien que seul le premier violon redémarre à la dernière mesure du troisième système, la portée de second violon apparaît elle aussi.

```
\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 }
    }
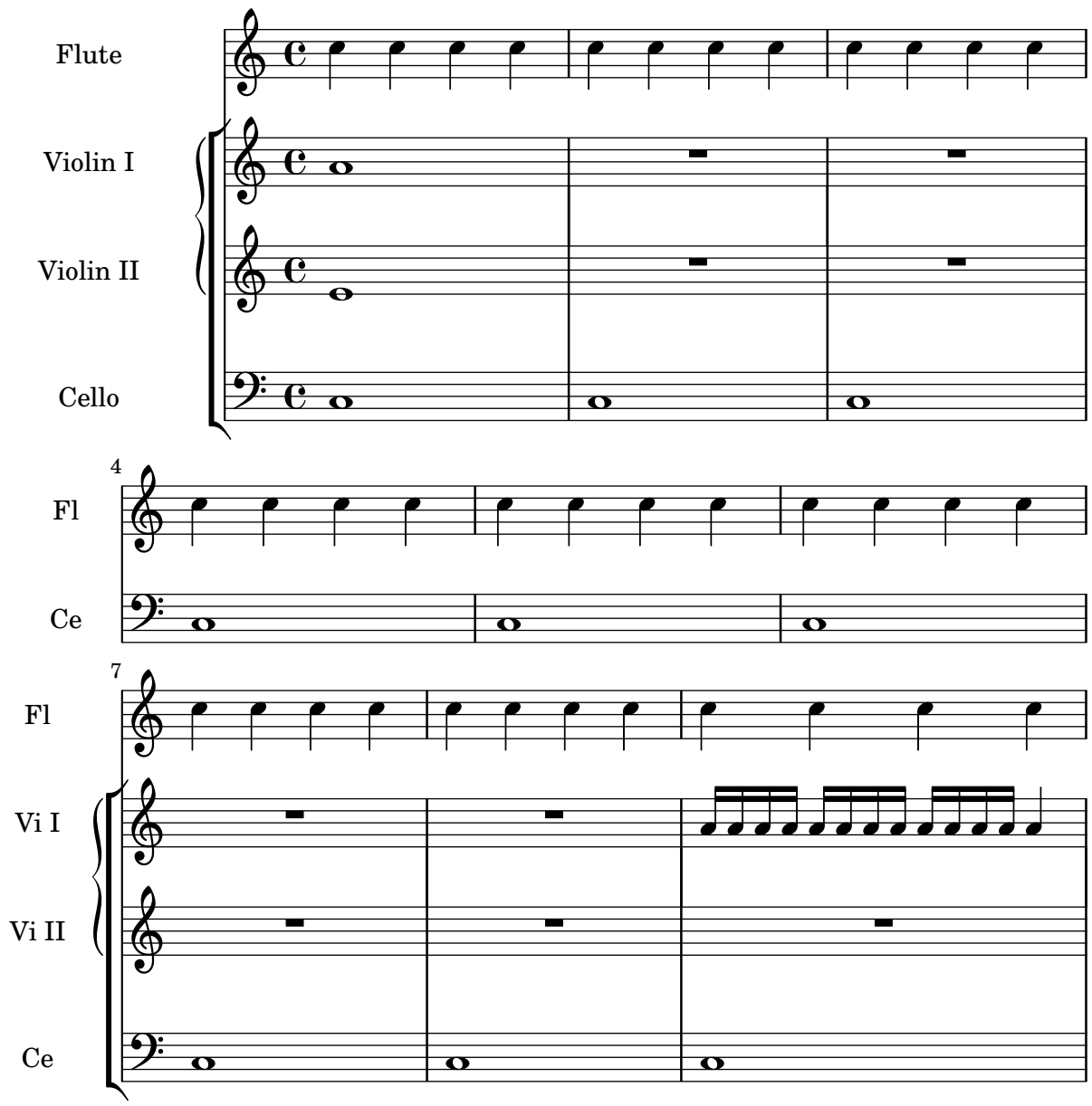
    \new StaffGroup = "StaffGroup_Strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
          shortInstrumentName = "Vi I"
        } \relative c'' {
          a1 | R1*7 | \*12 a16 a4 |
        }
        \new Staff = "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
  }
}
```



Séparation visuelle entre les systèmes

La séparation entre deux systèmes consécutifs peut être mise en évidence par n'importe quel *markup*. LilyPond dispose à cet effet d'une double oblique inversée : `\slashSeparator`.

```
#(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
    >>
  }
}
```

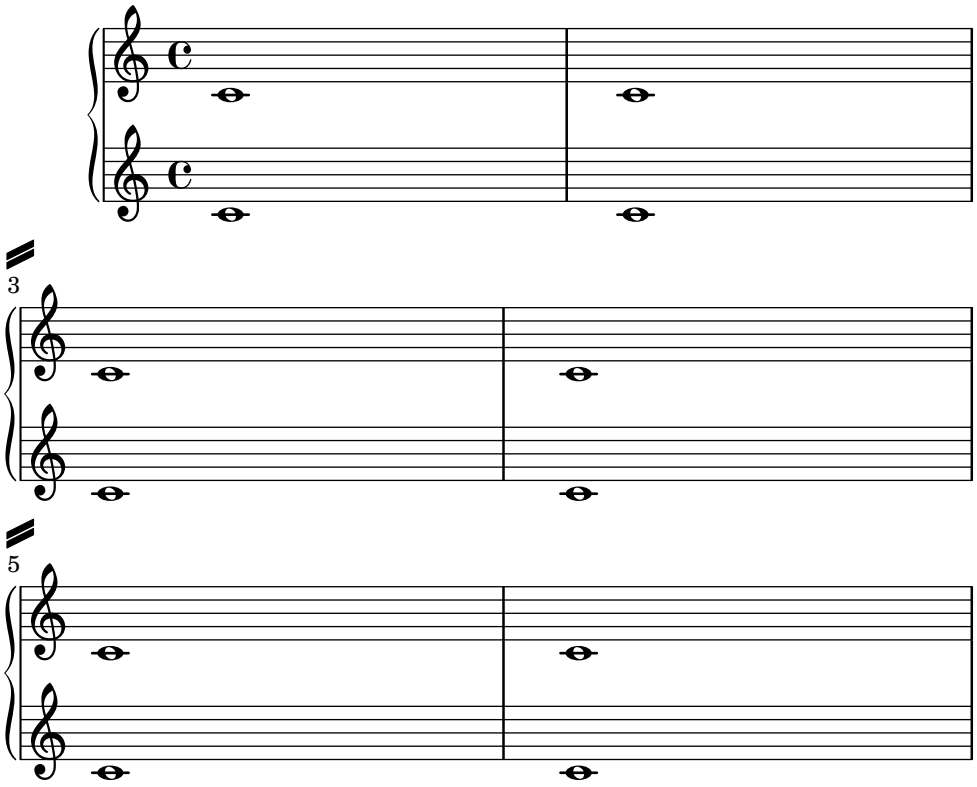


Table des matières

L'instruction \markuplist \table-of-contents permet d'inclure une table des matières. Ses éléments sont ajoutés à l'aide de la commande \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
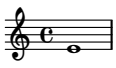
The first score **2**
Mark A **3**
The second score **4**

First score





Second score



Alignement vertical de StaffGroups sans SystemStartBar

Cet extrait indique comment obtenir des regroupements StaffGroup alignés verticalement, disposant chacun de leur indication de groupe SystemStartBar, sans qu'il ne soient connectés entre eux.

Il est à noter que ceci ne fonctionne correctement que lorsque la musique tient sur un seul système.

```
#(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

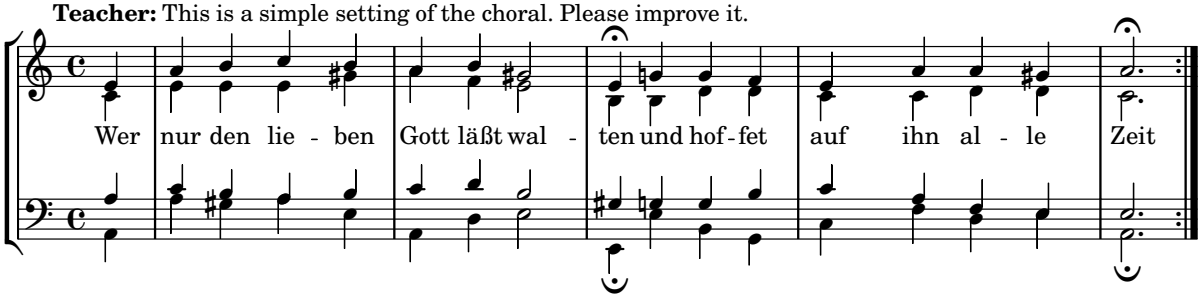
\score {
  <<
    \exercise
    \simpleBach
    \chromaticBach
  >>

  \header {
    title = \markup \column {
              \combine \null \vspace #1
              "Exercise: Improve the given choral"
              " "
            }
  }
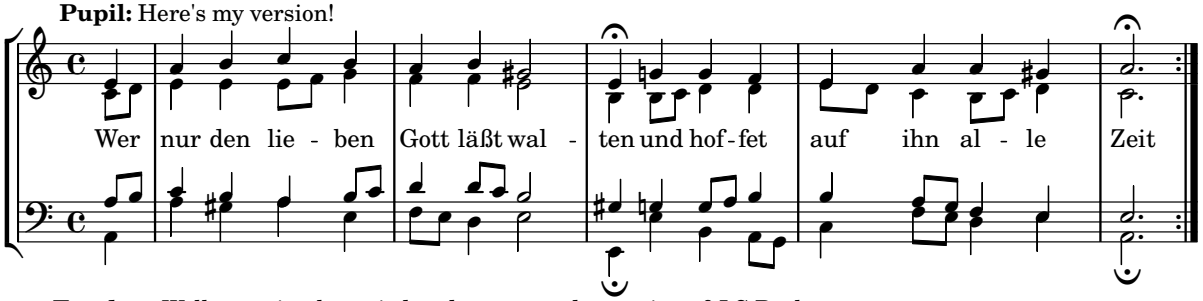
  \layout {
    \context {
      \Lyrics
      \override LyricText.X-offset = #-1
    }
  }
}
```

Exercise: Improve the given choral

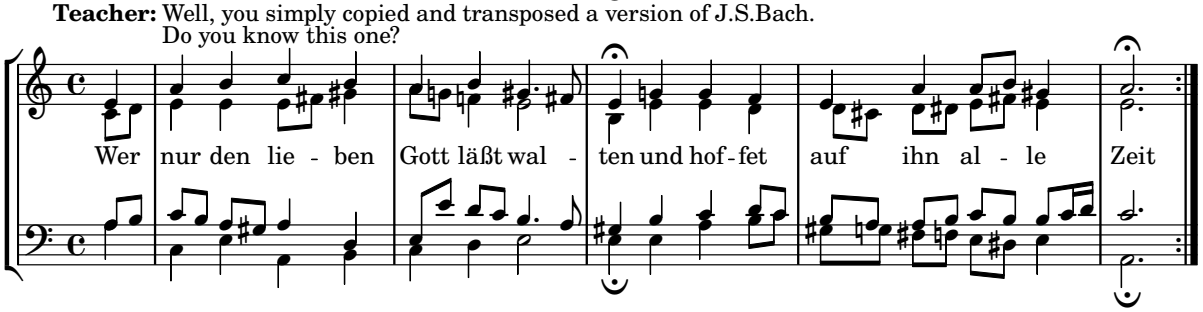
Teacher: This is a simple setting of the choral. Please improve it.



Pupil: Here's my version!



Teacher: Well, you simply copied and transposed a version of J.S.Bach.
Do you know this one?



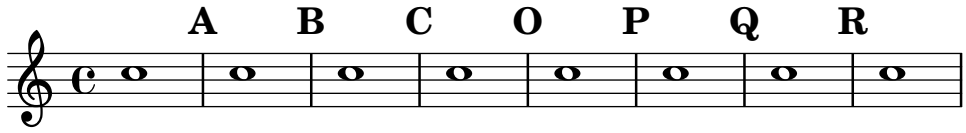
28 Preparing parts

Fixation arbitraire du numéro de repère de départ

Voici comment initialiser arbitrairement une indication automatique de repère, qu'elle soit alphabétique ou numérique :

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

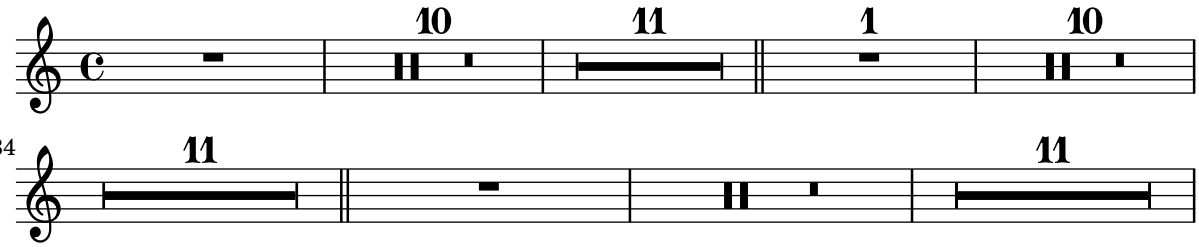
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1
}
```



Dénombrer une unique mesure de silence

Les silences multimesures affichent leur longueur sauf s'il n'y a qu'une seule mesure. Ceci peut se modifier en réglant `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
}
```



Quatuor à cordes, avec parties séparées

Grâce à ce canevas, vous pouvez obtenir une partition d'excellente facture pour quatuor à cordes mais aussi, si le besoin s'en faisait sentir, une partie séparée par instrument. Par ailleurs, cet exemple illustre l'utilisation de la fonction `\tag` dans le but d'extraire des parties séparées.

Pour des raisons techniques, un même fichier source ne peut générer ici de multiples fichiers. Il vous faudra donc découper ce canevas en plusieurs fichiers séparés ; leur nom respectif est indiqué en commentaire.

`piece.ly` comporte tout ce qui a trait à la musique. Les autres fichiers – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` et `vlc.ly` – vous permettront d'obtenir un conducteur et les parties selon le pupitre.

N'oubliez pas de supprimer les commentaires superflus des fichiers individualisés !

```
% 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.)

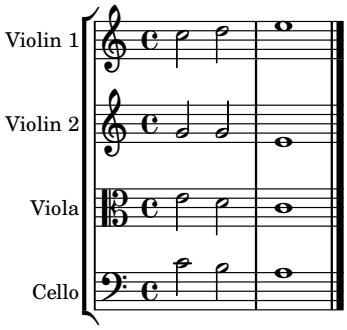
\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}

% vlc.ly
% (This is the Cello part file.)

\include "piece.ly"
```

```
\score {
  \keepWithTag #'vlc \music
  \layout { }
}
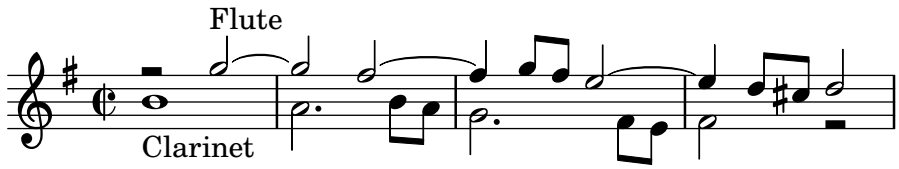
%}
```



29 Real music

Affectation d'un canal MIDI par voix

Lorsque LilyPond génère un fichier MIDI, chaque portée sera par défaut affectée à un canal, quel que soit le nombre de voix qu'elle contient. Ceci permet d'éviter de se retrouver à court de canaux, sachant qu'il n'y en a que seize de disponibles par piste.

Le fait de déplacer le `Staff_performer` dans le contexte `Voice` permet d'affecter à chaque voix d'une même portée un canal MIDI spécifique. Dans l'exemple suivant, la même portée donnera lieu à deux canaux MIDI différents, chacun étant affecté de son propre `midiInstrument`.

```
\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
  }
}
```
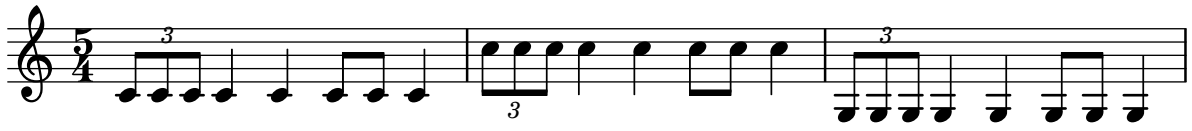


Création d'une séquence de notes de même hauteur

Lorsque la musique comporte à de nombreuses reprises une même séquence sur des hauteurs différentes, la fonction musicale ci-dessous peut s'avérer fort utile. Elle considère une note dont seule la hauteur est utilisée.

Cet exemple reproduit le rythme utilisé tout au long de « Mars », l'une des pièces de l'œuvre de Gustav Holst « Les planètes ».

```
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
}
```
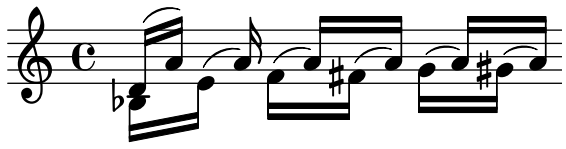


Création d'une liaison entre plusieurs voix

Dans certaines situations, il peut être nécessaire de lier des notes appartenant à des voix différentes. La solution consiste à ajouter une note invisible dans l'une des voix, à l'aide de la commande `\hideNotes`.

Voici par exemple la mesure 235 de la chaconne de la seconde partita pour violon solo, BWV 1004, de Bach.
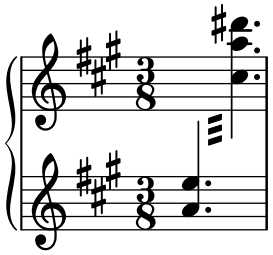
```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```



Trémolo et changement de portée

Dans la mesure où `\repeat tremolo` requiert deux arguments musicaux pour un trémolo d'accords, la note ou l'accord de la portée opposée doit être encadré par des accolades et se voir adjoindre la commande `\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
      }
    }
  }
>>
```



Démonstration des instruments MIDI

Partant du problème que l'on ignore quelle valeur de `midiInstrument` serait la plus adaptée à sa composition, la compilation de ce fichier permet de faire son choix.

```
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
```
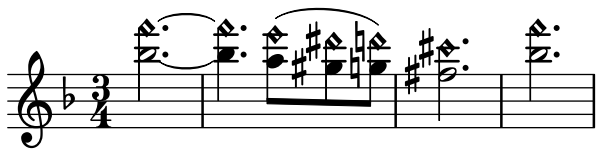
```
    % 121-128 sound effects
    \set Staff.midiInstrument = "guitar fret noise" \melody
    \set Staff.midiInstrument = "breath noise" \melody
    \set Staff.midiInstrument = "seashore" \melody
    \set Staff.midiInstrument = "bird tweet" \melody
    \set Staff.midiInstrument = "telephone ring" \melody
    \set Staff.midiInstrument = "helicopter" \melody
    \set Staff.midiInstrument = "applause" \melody
    \set Staff.midiInstrument = "gunshot" \melody
  }
  \midi { }
}
```



Harmoniques pointées

Les notes harmoniques artificielles, obtenues grâce à \harmonic, ne sont pas pointées. Ce comportement peut être modifié en activant la propriété de contexte 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.
}
```



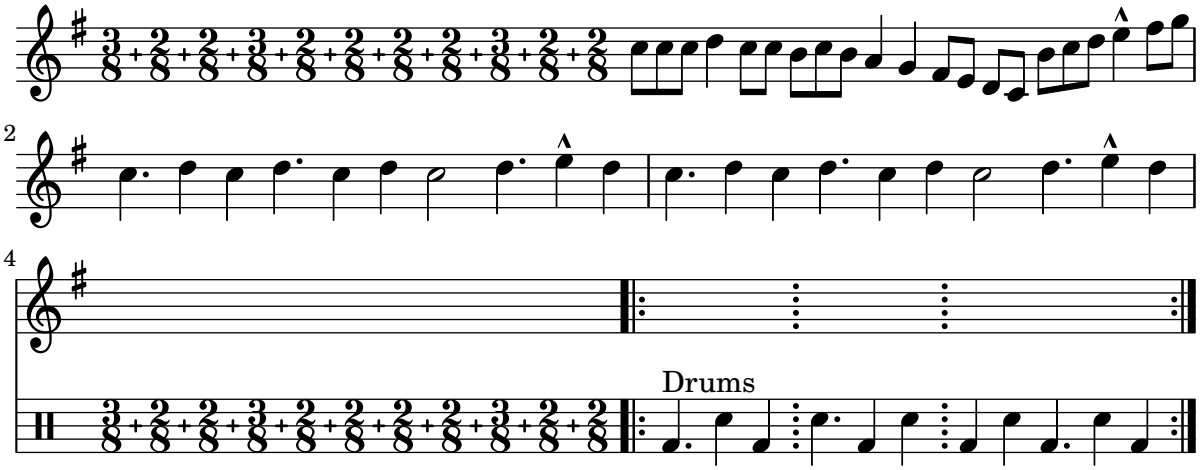
Indication personnalisée d'une polymétrie complexe

Bien que la métrique complexe ne soit pas l'élément primordial de cet exemple, elle permet d'indiquer la pulsation de cette pièce qui, par ailleurs, constitue le canevas d'une chanson traditionnelle des Balkans.

```
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
}
```



Indication d'un accord à cheval sur deux portées par un crochet

Un crochet « non-arpeggiato » peut indiquer que des notes réparties sur deux portées différentes doivent être jouées par la même main. Le contexte `PianoStaff` doit accepter ces crochets « distribués ».

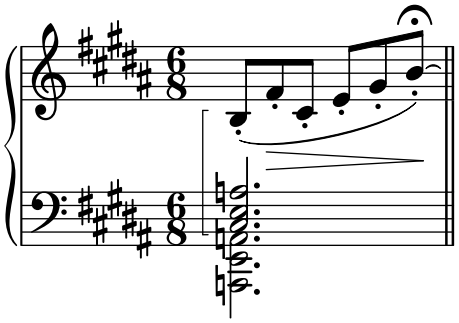
Est ici reproduite la mesure 65 du prélude de Debussy *Les collines d'Anacapri*.

```
\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
  }
```

>>



Insertion d'un fragment dans un *markup* au-dessus de la portée

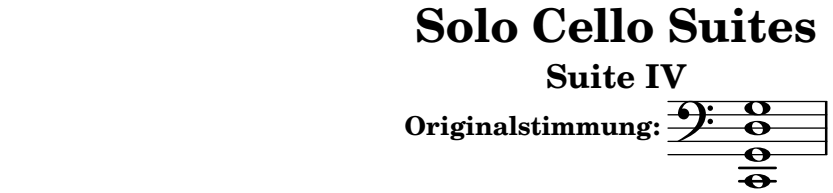
La commande \markup est polyvalente. Dans cet exemple, elle contient un bloc \score plutôt que du texte ou une marque.
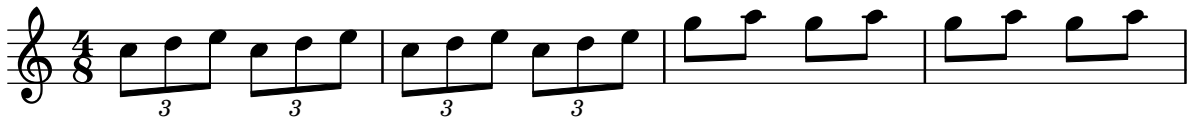
```
tuning = \markup \score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \clef bass
    <c, g, d g>1
  }
  \layout {
    indent = 0\cm
  }
}

\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
}
```





Exemple pour percussions

Ce court exemple est tiré de « L'histoire du Soldat » de Stravinsky.

```
#(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
}

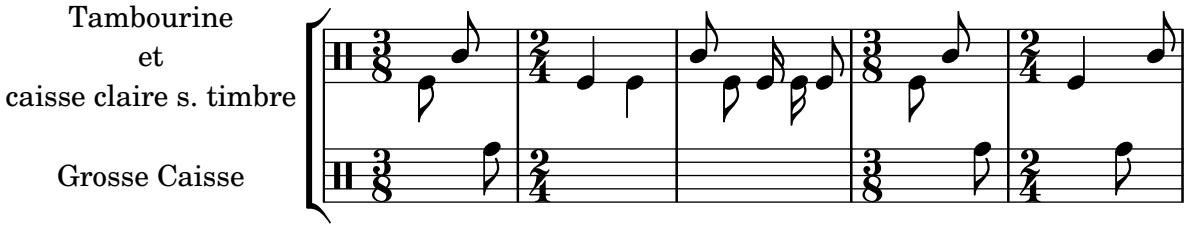
drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
      sn4 \D sn4 |
      \U tamb8 \D sn \U sn16 \D sn \U sn8 |
      \D sn8 \U tamb s |
      \U sn4 s8 \U tamb
    }
  >>
}

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
  >>
}
```



Impression de musique aux métriques différentes

Bien qu'ayant des métriques bien différentes, les deux parties ci-dessous présentées restent synchrones.

Les barres de mesure ne peuvent plus être gérées au niveau du contexte `Score` ; le `Timing_translator` doit être déplacé du contexte `Score` au contexte `Staff` afin de permettre des barres de mesure individualisées.

Le `Bar_number_engraver` devra lui aussi être déplacé, puisqu'il dépend de propriétés attachées au `Timing_translator`, afin de numéroter les mesures. L'utilisation d'un bloc `\with` dans la portée concernée permettra un affichage des numéros de mesure.
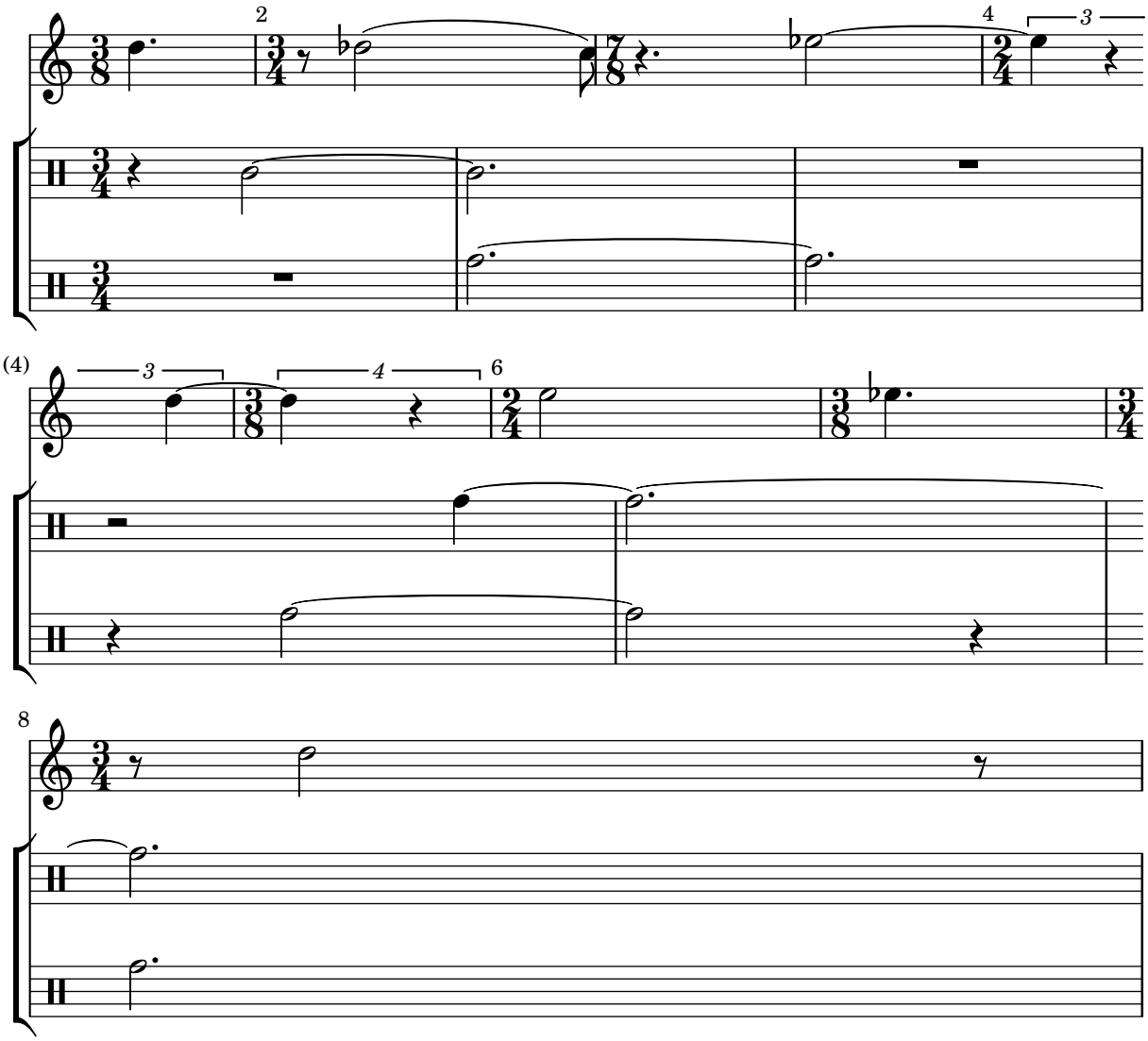
```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}

\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
  >>
}
```


2
4
3
(4)
3
4
6
8

30 Really cool

Ajout de la date du jour à une partition

Avec un peu de code Scheme, voici comment ajouter facilement la date du jour à votre partition.

```
\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
}
```

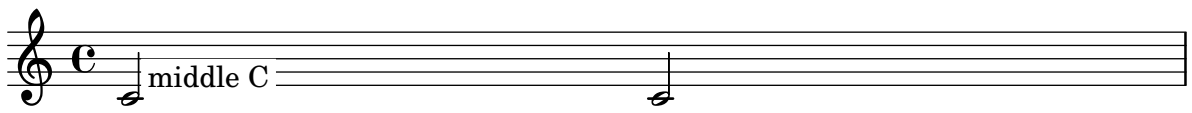


Blanchiment de lignes de portée avec la commande `\whiteout`

L'instruction `\whiteout` intercale un rectangle blanc contenant un *markup*. Dans la mesure où les lignes de la portée sont à un niveau inférieur à la plupart des autres objets graphiques, ce cache ne devrait pas masquer d'autres éléments.

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

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```



Texte centré sous un soufflet

La fonction comprise dans l'extrait suivant permet d'ajouter du texte – comme « molto » ou « poco » – à un soufflet de (de)crescendo. Le texte ajouté viendra se positionner selon le positionnement du soufflet. Le soufflet s'aligne avec un objet `DynamicText`.

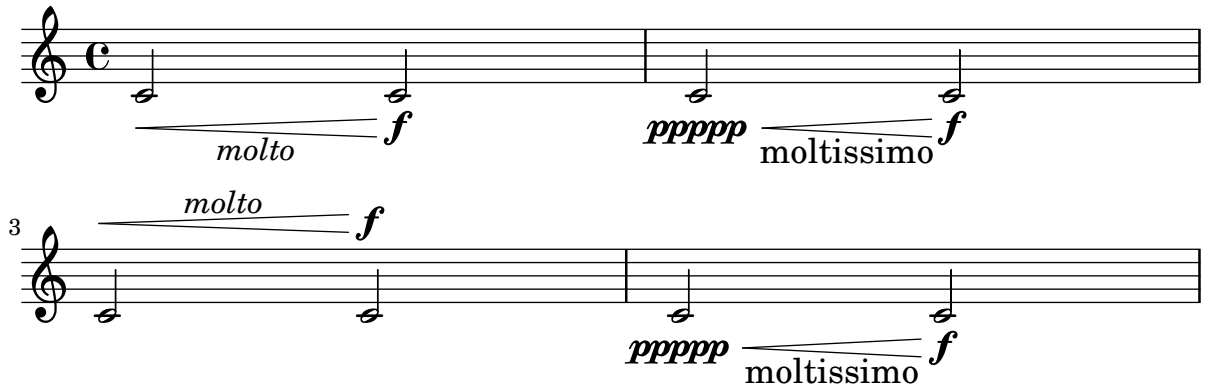
Cet exemple présente aussi comment, à l'aide de code Scheme, influencer la manière dont un objet est normalement imprimé.

```
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
}
```



Modification des propriétés d'objets particuliers

La commande `\applyOutput` permet de personnaliser n'importe quel objet de rendu. Elle requiert une fonction Scheme à trois arguments.
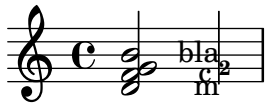
Dans l'exemple suivant, la fonction `mc-squared` est exécutée pour tous les objets `NoteHead` (au sein du contexte `Voice` en cours) à ce moment précis ; la fonction modifie le `stencil` de l'objet graphique à l'aide de la propriété `staff-position` pour remplacer certaines hauteurs par des *markups*.

De plus amples informations sont disponibles à la rubrique Section "Application d'une fonction à tous les objets de mise en forme" dans *Extension de LilyPond*.

```
#(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
```

```
}
```

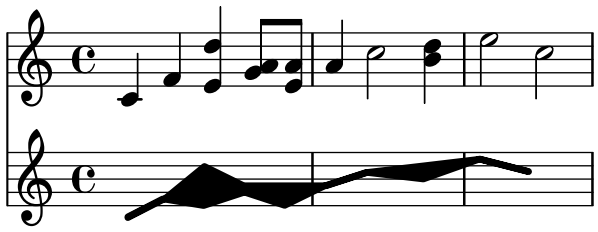


Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

```
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
>>
```



Coloration des notes selon leur hauteur

Les têtes de note peuvent adopter une couleur différente selon leur hauteur ou leur nom ; la fonction utilisée ici fait même la distinction entre enharmoniques.

```
% 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))))
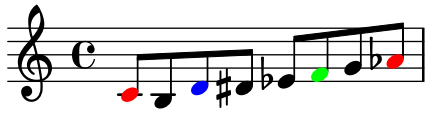
% Compare pitch and alteration (not octave).
```

```
#(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
  }
}
```
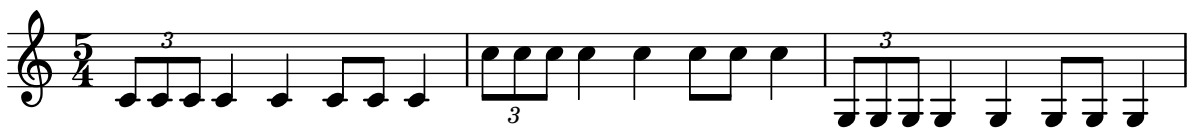


Création d'une séquence de notes de même hauteur

Lorsque la musique comporte à de nombreuses reprises une même séquence sur des hauteurs différentes, la fonction musicale ci-dessous peut s'avérer fort utile. Elle considère une note dont seule la hauteur est utilisée.

Cet exemple reproduit le rythme utilisé tout au long de « Mars », l'une des pièces de l'œuvre de Gustav Holst « Les planètes ».

```
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
}
```



Génération de notes aléatoires
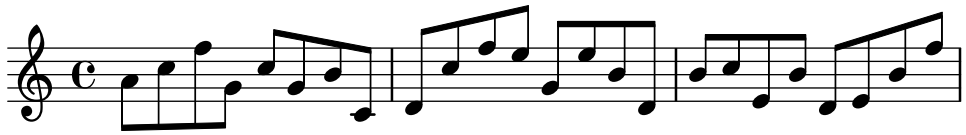
Le fragment de code Scheme ci-dessous génère aléatoirement des notes. Il s'utilise ainsi :

```
\randomNotes n de à dur
```

pour générer *n* notes dont la hauteur varie entre *de* et *à* sur une durée de *dur*.

```
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
```



Génération en Scheme de partitions complètes (y compris des parties d'ouvrage) sans utiliser l'analyseur

Une partition LilyPond, de manière interne, n'est rien d'autre qu'une expression Scheme générée par l'analyseur syntaxique de LilyPond. Il est donc possible, à l'aide de Scheme, de générer automatiquement une partition sans fichier source. Une expression musicale en Scheme sera transformée en partition par un appel à

```
(scorify-music music)
```

Ceci aura pour effet de générer un objet `score` auquel sera appliqué un bloc `layout` comportant la fonction

```
(let* ((layout (ly:output-def-clone $defaultlayout)))
   ; modification de la mise en forme, puis assignation :
   (ly:score-add-output-def! score layout))
```

Il suffit alors de transmettre ce `score` à LilyPond pour qu'il le grave. Les trois fonctions – `(add-score score)`, `(add-text text)` et `(add-music music)` – définies dans le code ci-dessous permettent de transmettre à LilyPond, aux fins de les graver, une partition complète, un *markup* ou simplement de la musique.

Cet exemple permet aussi de graver les pièces contenues dans un bloc `\book { ... }` ainsi que des partitions de niveau supérieur. Chaque partition destinée à être gravée est alors ajoutée à la liste des partitions de niveau supérieur ; le `toplevel-book-handler` – fonction Scheme appelée pour traiter un *book* dès que le bloc `\book { ... }` est clôturé – s'adapte pour prendre en charge tous les éléments `score` jusque là collectés dans l'ouvrage.

Attention : Pour des raisons techniques, seul le premier `\book` sera rendu puisque les autres commandes `\book` génèrent des fichiers additionnels.

```
#(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
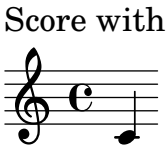
  \paper { tagline = ##f }
}


\book {
  \oneNoteScore
  \oneNoteScore

  \paper { tagline = ##f }
}

% Top-level scores are also handled correctly.
\oneNoteScore
\oneNoteScore
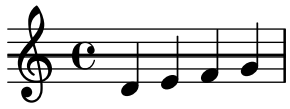
\paper { tagline = ##f }
```



Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane, ou bien pour mettre en exergue la ligne portant la clef de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doit être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Armures inhabituelles

La commande \key détermine la propriété keyAlterations d'un contexte Staff.

Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété. Il s'agit en l'occurrence de définir une liste :

```
\set Staff.keyAlterations =
  #`(((octave . pas) . altération) ((octave . pas) . altération) ...)
```
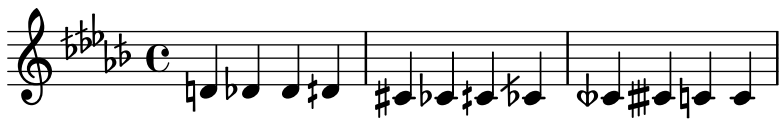
dans laquelle, et pour chaque élément, *octave* spécifie l'octave (0 pour celle allant du do médium au si supérieur), *pas* la note dans cette octave (0 pour do et 6 pour si), et *altération* sera ,SHARP ou ,FLAT ou ,DOUBLE-SHARP, etc. (attention à la virgule en préfixe).

Une formulation abrégée – (*pas* . *altération*) – signifie que l'altération de l'élément en question sera valide quelle que soit l'octave.

En ce qui concerne les gammes microtonales dans lesquelles un « dièse » n'est pas d'un centième, *altération* se réfère à un deux-centième de ton entier.

```
\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 |
}
```



Impression de musique aux métriques différentes

Bien qu'ayant des métriques bien différentes, les deux parties ci-dessous présentées restent synchrones.

Les barres de mesure ne peuvent plus être gérées au niveau du contexte Score ; le Timing_translator doit être déplacé du contexte Score au contexte Staff afin de permettre des barres de mesure individualisées.

Le Bar_number_engraver devra lui aussi être déplacé, puisqu'il dépend de propriétés attachées au Timing_translator, afin de numéroter les mesures. L'utilisation d'un bloc \with dans la portée concernée permettra un affichage des numéros de mesure.

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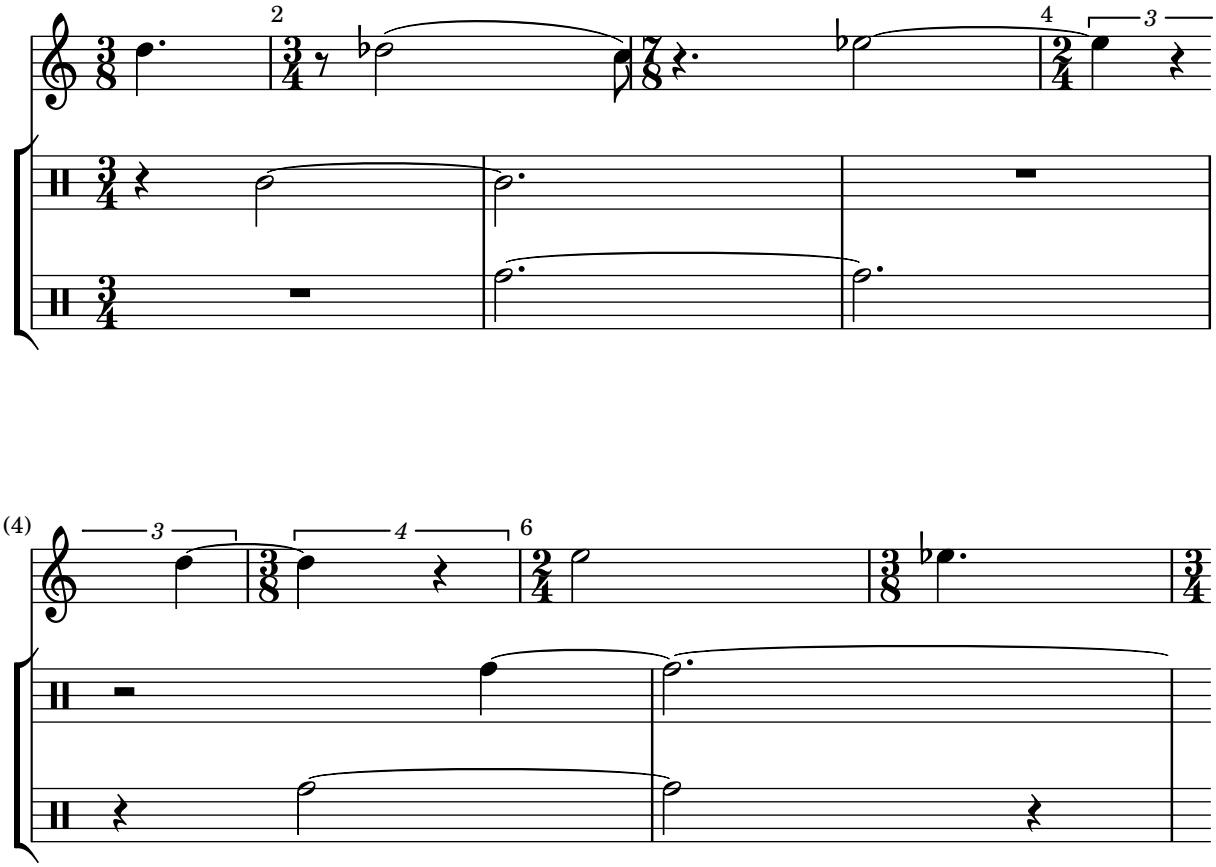
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"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

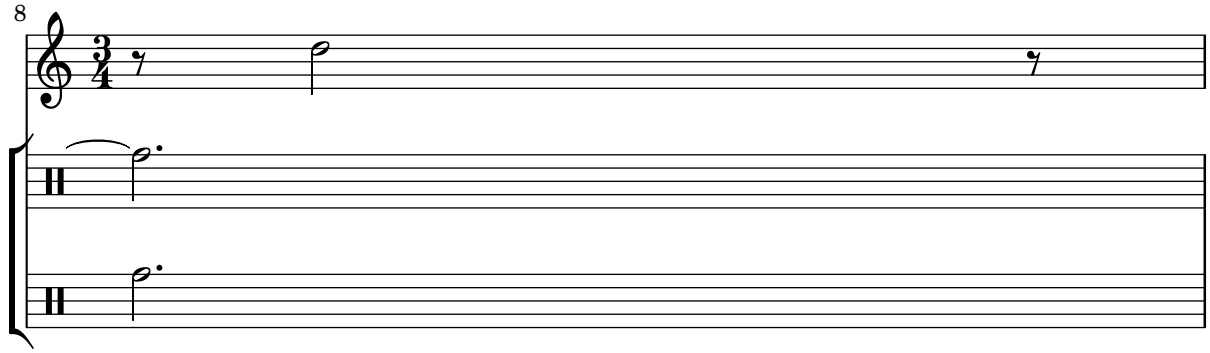
Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global
  {
    \clef treble
    \time 3/8 d''4. |
    \time 3/4 r8 des''2( c''8) |
    \time 7/8 r4. 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
  >>
}
```



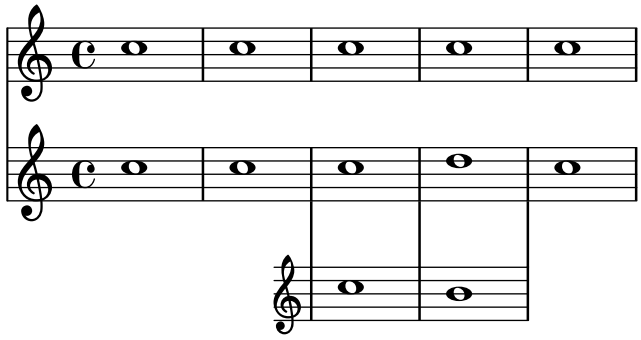

8

31 Really simple

Ajout d'une portée supplémentaire

Vous pouvez ajouter une nouvelle portée, éventuellement de manière temporaire, après le début d'un morceau.

```
\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
      }
    }
  >>
}
```



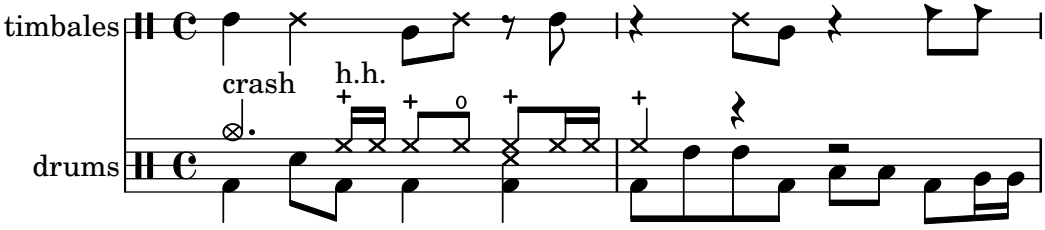
Ajout de parties de batterie

Grâce à la puissance des outils préconfigurés tels que la fonction `\drummode` et le contexte `DrumStaff`, la saisie de parties pour percussions est extrêmement simplifiée : chaque composant d'une batterie trouve sa place sur une portée dédiée (avec une clef particulière) et les têtes de note sont spécifiques à chaque élément. Il est également possible d'affecter un symbole particulier à chaque élément, tout comme de restreindre le nombre de lignes de la portée.

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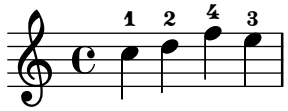
\score {
  <<
    \new DrumStaff \with {
      instrumentName = "timbales"
      drumStyleTable = #timbales-style
      \override StaffSymbol.line-count = #2
      \override BarLine.bar-extent = #'(-1 . 1)
    }
    <<
      \timb
    >>
    \new DrumStaff \with { instrumentName = "drums" }
    <<
 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



Ajout de doigtés à une partition

Les instructions de doigtés se saisissent selon une syntaxe très simple.

```
\relative c'' {
  c4-1 d-2 f-4 e-3
}
```

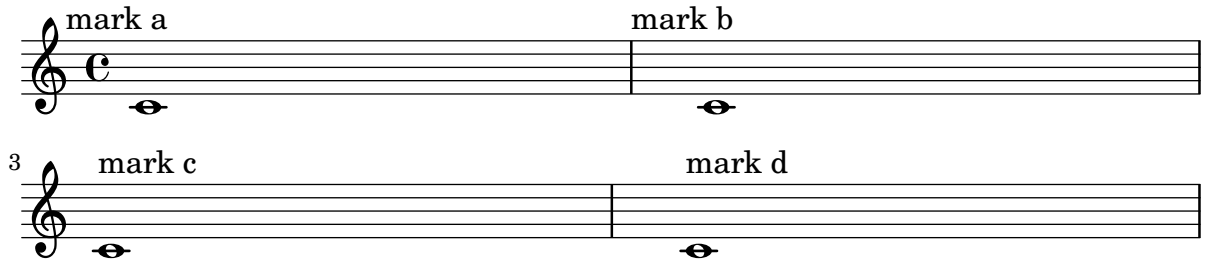


Alignement de repère sur une note

Les objets `TextMark` s'alignent par défaut sur des objets graphiques `NonMusicalPaperColumn` tels que le coin gauche de la portée ou une barre de mesure. Ils peuvent aussi venir s'aligner dès lors qu'aura été désactivée la propriété `non-musical`.

```
\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 |
}
```
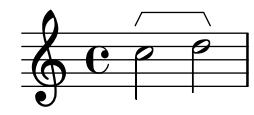


Crochets d'analyse au-dessus de la portée

Les crochets d'analyse viennent par défaut se positionner au-dessous de la portée. L'exemple suivant vous indique comment les faire apparaître en surplomb de la portée.

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

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```

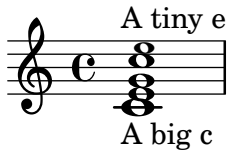


Modification de la taille d'une note particulière d'un accord

La commande `\tweak` permet de modifier la propriété `font-size` d'une note particulière d'un accord.

Il s'agit de placer, à l'intérieur même de l'accord (dans la construction `< >`) et avant la note considérée, l'instruction `\tweak` suivie de `font-size` et de définir la taille voulue – comme `#-2` pour une petite tête.

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



Changement de fontes des couplets
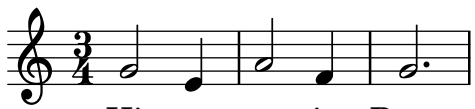
Chaque couplet peut disposer de sa propre fonte, y compris son numéro.

```
%{
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
}
```
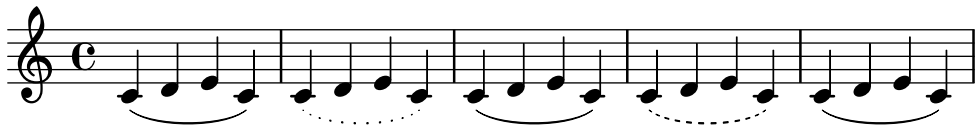


1. Hi, my name is Bert.
2. Oh, ché-ri, je t'aime

Modification de l'aspect des liaisons d'articulation

Une liaison d'articulation peut se présenter sous la forme d'un trait continu ou discontinu, voire en pointillé.

```
\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)
}
```
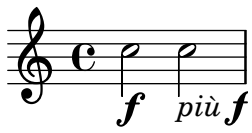


Combinaison de nuance et de texte

Certaines indications de nuance peuvent requérir un complément textuel, comme « più *f* » ou « *p* subito ». Elles se réalisent aisément à l'aide d'un bloc `\markup`. Le résultat se comportera comme un objet `TextScript`.

Voir aussi `Combinaison de nuance et de texte (2)`.

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

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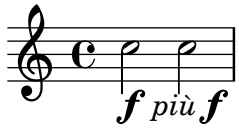


Combinaison de nuance et de texte (2)

Certaines indications de nuance peuvent requérir un complément textuel, comme « più *f* » ou « *p* subito ». Elles se réalisent à l'aide de la fonction Scheme `make-dynamic-script`. Les objets alors créés se comportent somme des objets graphiques `DynamicText`.

```
piuF = #(make-dynamic-script
          #{ \markup { \normal-text \italic più \dynamic f } #})

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

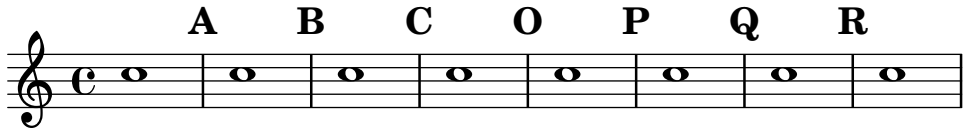


Fixation arbitraire du numéro de repère de départ

Voici comment initialiser arbitrairement une indication automatique de repère, qu'elle soit alphabétique ou numérique :

```
\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
}
```
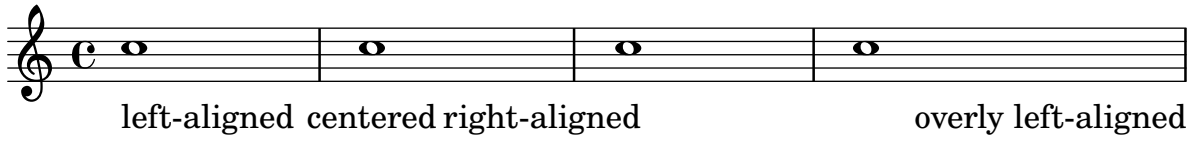


Alignement des syllabes

L'alignement horizontal des paroles peut se gérer à l'aide de la propriété `self-alignment-X` de l'objet `LyricText`. Les valeurs `-1` ou `#LEFT` produiront un alignement par la gauche, les valeurs `0` ou `#CENTER` un alignement centré, et les valeurs `1` ou `#RIGHT` un alignement par la droite. D'autres valeurs numériques sont permises – attention à ne pas oublier le préfixe Scheme '`#`' en cas de nombre négatif.

```
\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"
}
```
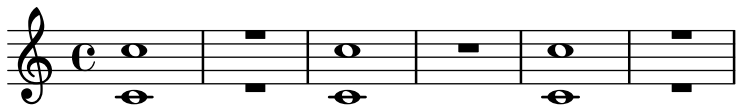


Fusion de silences multimesures dans une partie polyphonique

Dans le cadre d'un portée polyphonique, les silences multimesures sont positionnés différemment selon la voix à laquelle ils appartiennent. Le réglage suivant permet néanmoins de les imprimer sur une même ligne. En l'absence du \once, l'adaptation sera effective pour tous les autres silences à venir dans cette voix.

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

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

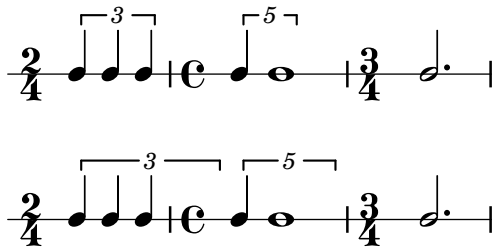


Modification de la longueur d'un crochet de n-olet

Les crochets indiquant un n-olet peuvent être prolongés jusqu'à un élément de rupture ou jusqu'à la note suivante. LilyPond termine un crochet de n-olet sur la droite de sa dernière tête de note par défaut ; un crochet de pleine longueur s'étendra plus avant, soit jusqu'à la note suivante et en traversant tous les éléments non rythmiques, soit sur tout l'espace précédant le prochain élément de notation, que ce soit une clef, une métrique, une armure ou une autre note. L'exemple suivant illustre la manière d'activer ces deux fonctionnalités.

```
\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.
}
```



Impression du numéro de version

Il est possible déjouter le numéro de la version de LilyPond utilisée dans un *markup*.

```
\markup { Processed with LilyPond version #(lilypond-version) }
```

Processed with LilyPond version 2.27.4

Piano – cannevas simple

Voici une simple partition pour piano avec quelques notes.

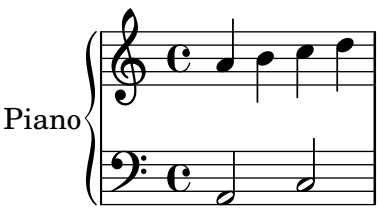
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  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 { }
}
```



Piano et paroles entre les portées

Lorsque la mélodie est doublée au piano, cela ne nécessite pas forcément une portée spécifique. Les paroles peuvent s'insérer entre les deux portées de la partition pour piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
```

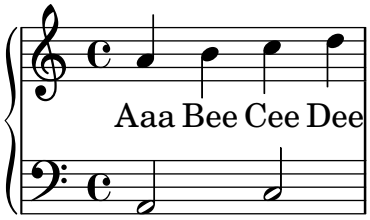
```
}

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 { }
}
```



Piano, mélodie et paroles

Il s'agit du format classique pour le chant : une portée pour la mélodie et les paroles au-dessus de l'accompagnement au piano.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

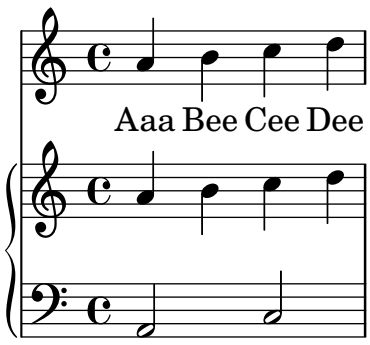
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4
```

```
  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 { }
}
```



Paroles, musique et accords

Ce canevas comporte tous les éléments d'une chanson : la mélodie, les paroles, les accords.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

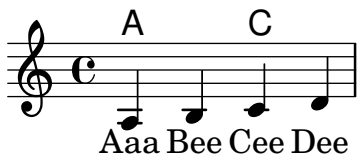
  a4 b c d
}

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 { }
}
```



Mélodie simple et accords
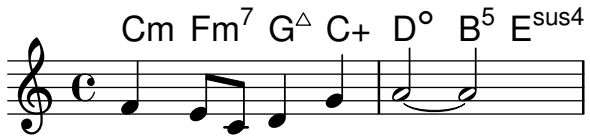
Vous avez besoin de la partition d'une mélodie avec les accords ? N'allez pas plus loin !

```
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 { }
}
```



Portée unique et paroles
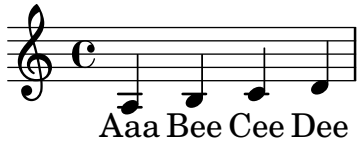
Ce canevas comporte une simple ligne mélodique agrémentée de paroles. Recopiez-le, ajoutez-y d'autres notes et paroles. Les ligatures automatiques sont ici désactivées comme il est d'usage en matière de musique vocale. Pour activer la fonction de ligature automatique, modifiez ou commentez la ligne en question.

```
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 { }
}
```



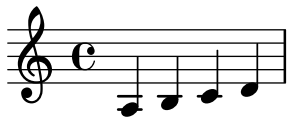
Portée unique avec quelques notes

Cet exemple simpliste se compose d'une portée agrémentée de quelques notes. Il convient tout à fait pour un instrument seul ou un fragment mélodique. Recopiez-le dans un nouveau fichier, ajoutez-y d'autres notes et c'est prêt !

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}
```

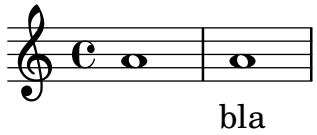
```
\score {
  \new Staff \melody
  \layout { }
  \midi { }
}
```



Sauts de notes en mode paroles

La syntaxe s qui permet de « faire un saut dans le temps » n'est disponible qu'en mode notes et en mode accords. Dans les autres situations, comme en mode paroles par exemple, la commande \skip produit les mêmes effets.
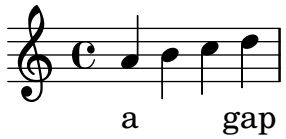
```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip1 bla1 }
>>
```



Sauts de notes en mode paroles (2)

Sachant qu'un s au sein d'un bloc \lyricmode est interprété non comme de l'espace mais comme une syllabe, l'utilisation de paires de guillemets informatiques ("") ou un caractère souligné simple (_) permet de « sauter » une note.

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



Quatuor à cordes (conducteur)

Voici un canevas pour quatuor à cordes. Notez l'utilisation de la variable \global pour traiter la métrique et la tonalité.

Voir aussi Quatuor à cordes, avec parties séparées.

```
global= {
  \time 4/4
  \key c \major
}

violinOne = \new Voice \relative c'' {
  c2 d
  e1
```

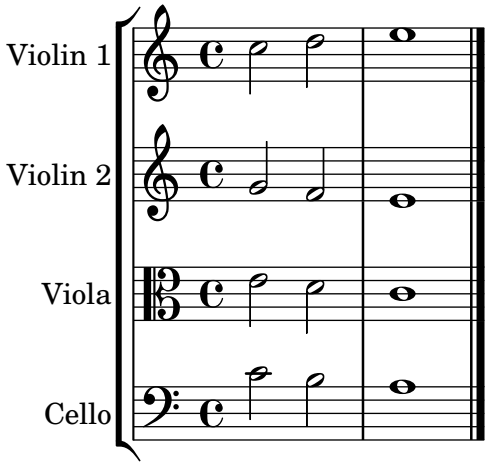
```
  \bar "|."
}

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 "|."
}

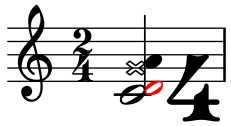
\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Violin 1" }
      << \global \violinOne >>
    \new Staff \with { instrumentName = "Violin 2" }
      << \global \violinTwo >>
    \new Staff \with { instrumentName = "Viola" }
      << \global \viola >>
    \new Staff \with { instrumentName = "Cello" }
      << \global \cello >>
  >>
  \layout { }
  \midi { }
}
```



Utilisation de \tweak pour retoucher des objets particuliers

La commande \tweak permet de retoucher directement n'importe quel objet graphique. En voici quelques exemples :

```
\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
}
```



Ensemble vocal (simple)

Ce fichier constitue un canevas standard de partition pour chœur à quatre voix mixtes. Lorsque les ensembles s'étoffent, il est judicieux de recourir à une section spécifique incluse dans chacune des parties, tout particulièrement pour gérer la métrique et la tonalité qui, la plupart du temps, sont communes à tous les pupitres. Comme il est d'usage pour les hymnes, les quatre voix sont réparties sur deux portées.

```
\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
}
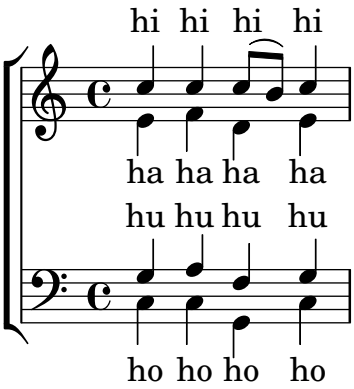
sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

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
  >>
}
```

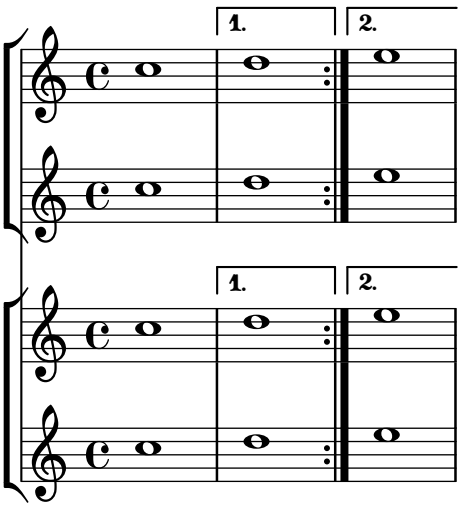


Crochets de *volta* sur plusieurs portées

L'ajout du `Volta_engraver` à une portée particulière permet que les crochets de *volta* s'affichent sur d'autres portées que la première d'une partition.

```
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
  >>
>>
```



32 Scheme

See also Section “Scheme tutorial” dans *Extension de LilyPond* and Section “Interfaces for programmers” dans *Extension de LilyPond*.

Ajout d’un doigté supplémentaire avec Scheme

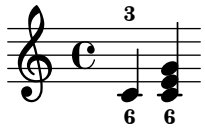
Il est possible, à l’aide de la fonction `map-some-music`, d’ajouter divers éléments à des notes. Voici comment attacher un script supplémentaire à une note ou un accord.

En règle générale, réaliser préalablement un `\displayMusic` sur de la musique similaire à ce qui est recherché permettra d’en observer la structure. Ceci pourra servir de canevas pour l’écriture de votre code 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'> }
}
```



Ajout d’indicateurs à une portée dédoublée après un saut de ligne

Dans cet extrait sont définies les commandes `\splitStaffBarLine`, `\convUpStaffBarLine` et `\convDownStaffBarLine`. Ces commandes ajoutent une double flèche après la dernière barre de mesure d’une portée, indiquant par là que ses différentes voix disposeront de leur propre portée à la ligne suivante, ou bien se regrouperont.

Il est à noter que, dans l’implémentations ici proposée, sont dessinées des flèches sans dimension dans la marge droite. Ceci ne devrait pas poser de problème pour une impression normale. Il sera cependant nécessaire d’agrandir horizontalement la boîte englobante, comme indiqué ci-après, afin d’éviter le rognage lors d’une sortie sous forme d’image.

```
#(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
}

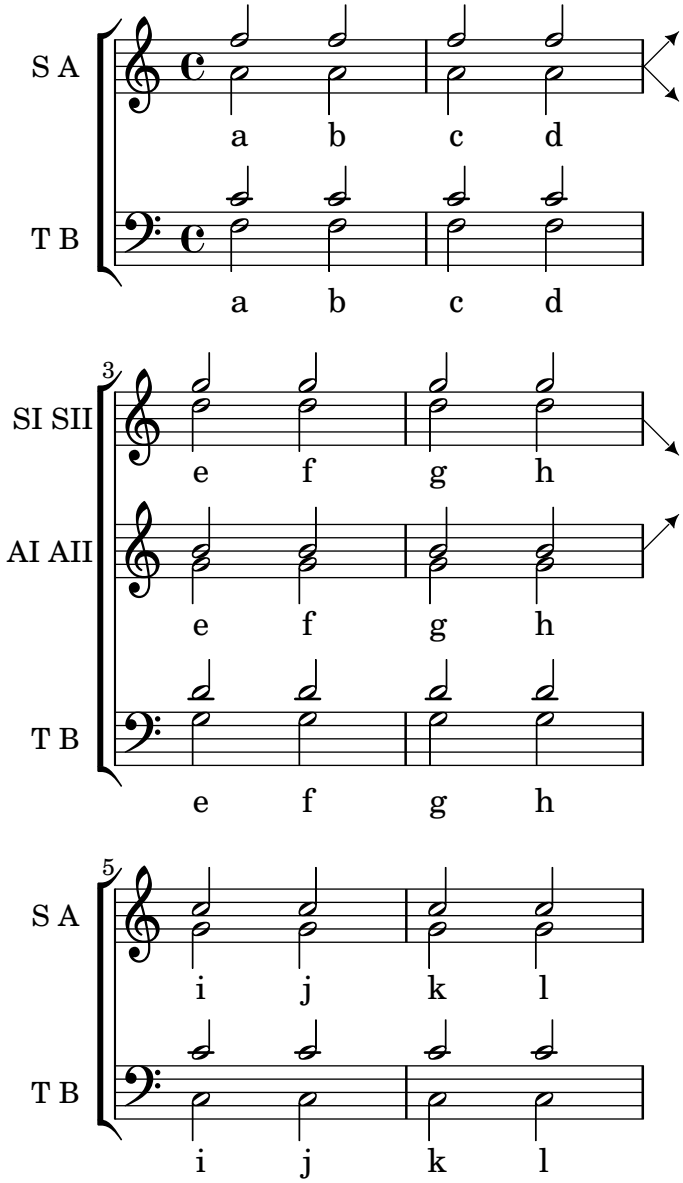
\markup \pad-x #3  % avoid cropping
  \score {
    <<
      \new ChoirStaff <<
        \new Staff = up \with {
          instrumentName = "SI SII"
          shortInstrumentName = "SI SII"
        } {
          s1*4
        }

        \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
      }
    }
}
```



Ajout de liens à des objets

La fonction `add-link`, telle que définie ci-dessous, permet d'ajouter un lien au stencil d'un objet graphique. Elle s'emploie au sein d'un `\override` ou d'un `\tweak`.

À noter que le fonctionnemment du `point-and-click` est perturbé sur les objets ainsi liés.

Cette fonction n'est opérationnelle que pour une sortie PDF.

La coloration des objets liés s'obtient par une commande séparée.

```
#(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)
```
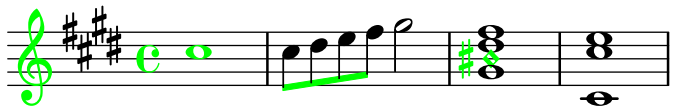
```
\relative c' {
   \key cis \minor

   \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>
}
```



Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur

Lorsque le nombre d'instruments cités dans la réduction pour piano se multiplie, vous pourriez avoir intérêt à créer votre propre fonction pour gérer ces repères. La fonction musicale `\cueWhile` prend quatre arguments : la musique d'où provient la citation, telle que définie par `\addQuote`, le nom qui sera mentionné en regard de cette citation, son positionnement – `UP` ou `DOWN` selon qu'il sera attribué à `\voiceOne` et placé au-dessus ou `\voiceTwo` et placé en dessous – et enfin la musique du piano qui interviendra en parallèle. Le nom de l'instrument en question viendra s'aligner sur la gauche de la citation. Bien que vous puissiez effectuer plusieurs citations, elle ne peuvent être simultanées.

```
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 }
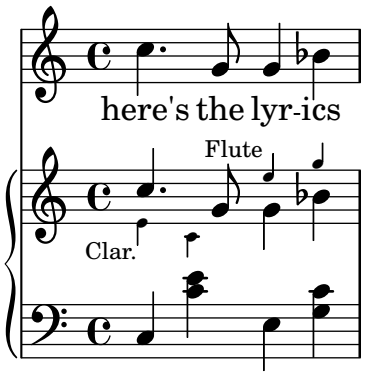
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

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
      }
    >>
  >>
}
```



Ajout de la date du jour à une partition

Avec un peu de code Scheme, voici comment ajouter facilement la date du jour à votre partition.

```
\paper { tagline = ##f }
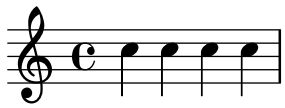
% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))

% 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

Ajustement du positionnement vertical d'une liaison

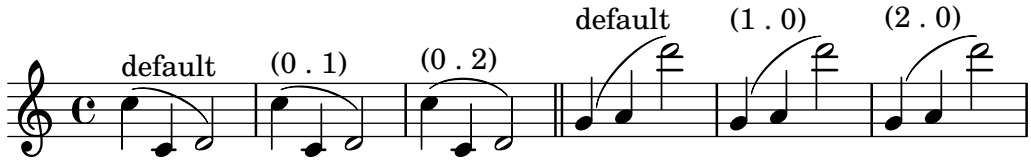
Il est possible, à l'aide d'un `\override Slur.positions`, de définir le positionnement vertical des extrémités d'une liaison à des valeurs absolues (ou, plutôt, forcer l'algoritme de LilyPond à considérer ces valeurs comme étant préférables). Dans de nombreux cas, ceci demande de procéder à tâtons pour obtenir une valeur acceptable, d'autant que la commande `\offset` est inefficace pour les liaisons et affiche un avertissement.

Le code ci-dessous parmet d'ajuster le positionnement du début et de la fin d'une liaison en spécifiant des modifications *relatives* comme le ferait un `\offset`.

Syntaxe : `\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)
}
```



Texte centré sous un soufflet

La fonction comprise dans l'extrait suivant permet d'ajouter du texte – comme « molto » ou « poco » – à un soufflet de (de)crescendo. Le texte ajouté viendra se positionner selon le positionnement du soufflet. Le soufflet s'aligne avec un objet `DynamicText`.

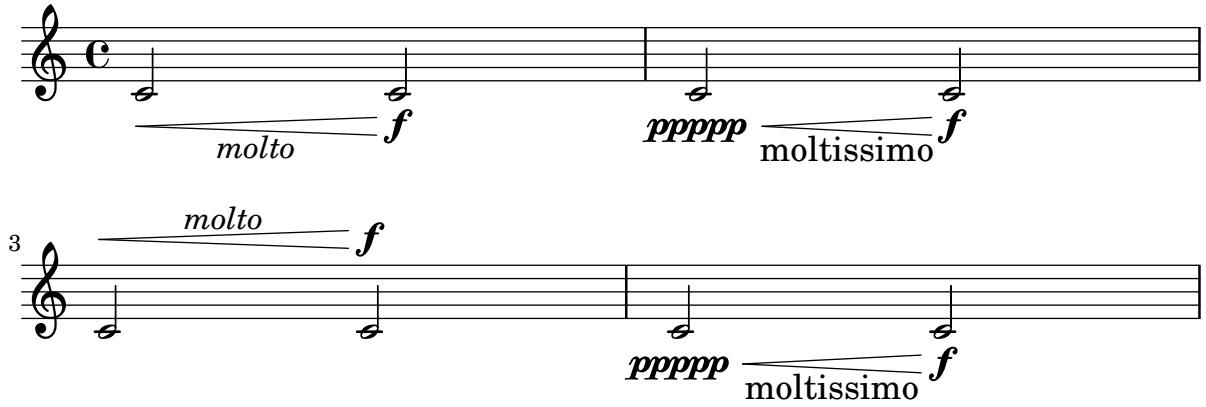
Cet exemple présente aussi comment, à l'aide de code Scheme, influencer la manière dont un objet est normalement imprimé.

```
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
}
```



Modification des propriétés d'objets particuliers
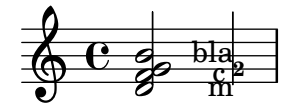
La commande `\applyOutput` permet de personnaliser n'importe quel objet de rendu. Elle requiert une fonction Scheme à trois arguments.

Dans l'exemple suivant, la fonction `mc-squared` est exécutée pour tous les objets `NoteHead` (au sein du contexte `Voice` en cours) à ce moment précis ; la fonction modifie le `stencil` de l'objet graphique à l'aide de la propriété `staff-position` pour remplacer certaines hauteurs par des *markups*.

De plus amples informations sont disponibles à la rubrique Section "Application d'une fonction à tous les objets de mise en forme" dans *Extension de LilyPond*.

```
#(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
}
```



Modèles de chiffrage d'accords

La propriété `chordNameExceptions` enregistre une liste de chiffrages personnalisés afin de traiter des cas non prévus ou incorrectement gérés.

Par défaut, LilyPond utilise le jeu de chiffrages issus de l'ouvrage de Klaus Ignatzek « Die Jazzmethode für Klavier 1 » ; l'algorithme de conversion des accords en chiffrages est inclus dans le fichier `scm/chord-ignatzek-names.scm`. Des règles additionnelles pour gérer des chiffrages exceptionnels sont enregistrées en tant que variables `ignatzekExceptions` dans le fichier `ly/chord-modifiers-init.ly`.

Le code ci-dessous modifie ces exceptions en trois étapes.

1. Saisie de musique en accords avec *markup* associé. Par convention, la base – autrement dit la note la plus basse – de chaque accord aura une hauteur de do (`c` pour les anglophones).
2. Appel de la fonction Scheme `sequential-music-to-chord-exceptions` pour créer une liste de nouvelles exceptions, puis l'ajouter à celles déjà répertoriées. Dans la mesure où `ignatzekExceptions` est défini avec le second paramètre de cette fonction fixé à `#t`, afin d'ignorer la base de chaque accord, il faut faire de même.
3. Enregistrement de la nouvelle liste d'exceptions.

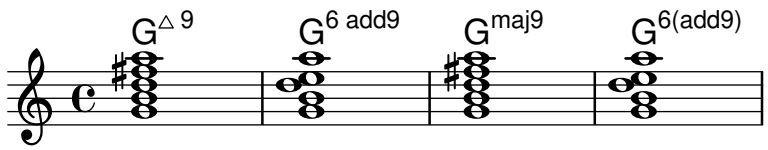
```
% Step 1: Define music with chords and markup for maj9 and 6(add9).
chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

% 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
}
```



Coloration des notes selon leur hauteur

Les têtes de note peuvent adopter une couleur différente selon leur hauteur ou leur nom ; la fonction utilisée ici fait même la distinction entre enharmoniques.
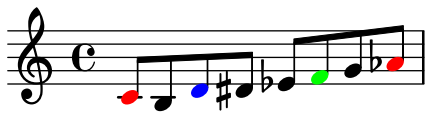
```
% Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (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
  }
}
```
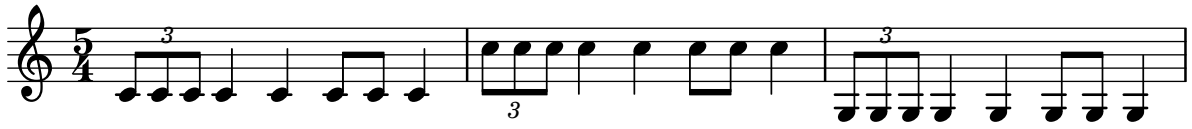


Création d'une séquence de notes de même hauteur

Lorsque la musique comporte à de nombreuses reprises une même séquence sur des hauteurs différentes, la fonction musicale ci-dessous peut s'avérer fort utile. Elle considère une note dont seule la hauteur est utilisée.

Cet exemple reproduit le rythme utilisé tout au long de « Mars », l'une des pièces de l'œuvre de Gustav Holst « Les planètes ».

```
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
}
```



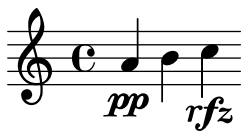
Création de nuance particulière pour la sortie MIDI

L'exemple suivant illustre la manière de créer une indication de nuance, absente de la liste par défaut, et de lui assigner une valeur spécifique utile à la sortie MIDI.

L'indication de nuance `\rfz` (*rinforzando*) se voit attribuer une valeur de `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 {}
}
```



Contrôle de la position et du nombre de points d'une barre de reprise

Il est possible de personnaliser la position et le nombre de points d'une barre indiquant une reprise, que ce soit en définissant de nouvelles barres de mesure personnalisées ou bien en redéfinissant la manière dont les indications de reprise par défaut sont dessinées. Ceci s'avère tout particulièrement utile dans le cas de portées aux lignes adaptées comme dans l'exemple ci-dessous.

```
#(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))

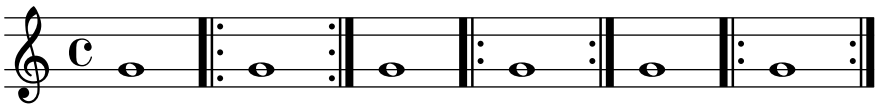
% With the procedure above we can define custom bar lines, for example,
% that resemble standard repeat sign bar lines except that there are
% three dots at staff positions -3, 0, and 3.

#(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 }
```

```
}
```



Définition d'un graveur en Scheme : graveur d'ambitus

Cet exemple démontre comment définir son propre graveur de tessiture à base de code Scheme. Il s'agit d'une réécriture en Scheme du code contenu dans le fichier `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))))
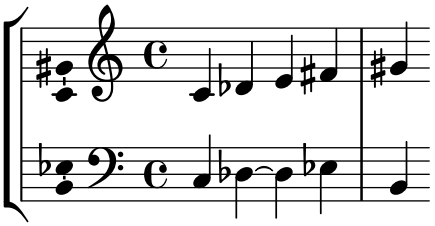
        ;; Set the ambitus clef and key signature state.
        ((stop-translation-timestep translator)
         (if ambitus
             (initialize-ambitus-state ambitus translator)))

        ;; 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 } }
}
```



Taille de police différente pour `instrumentName` et `shortInstrumentName`

Une simple dérogation à un contexte particulier permet de différencier la taille des variables *instrumentName* et *shortInstrumentName*.

```
InstrumentNameFontSize =
#(define-music-function (font-size-pair) (pair?)
   "Set the font size of `InstrumentName` grobs.

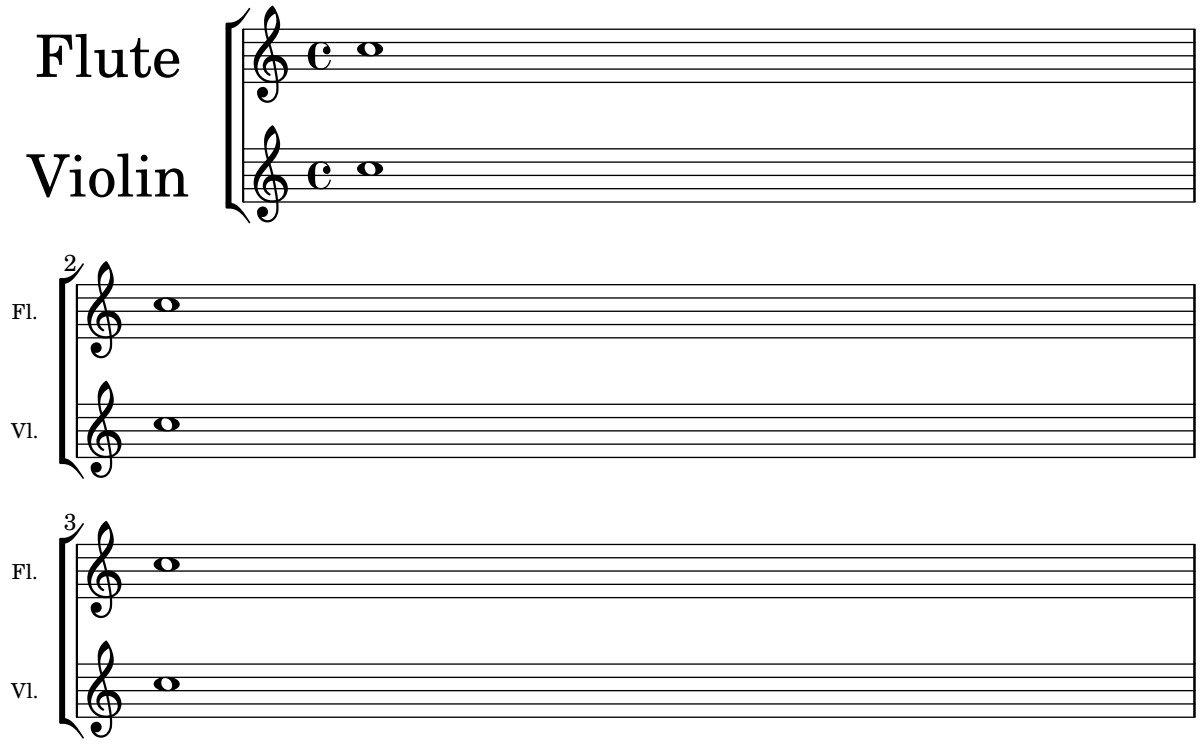
The first value of FONT-SIZE-PAIR sets the font size of the initial
`instrumentName` property, the second value sets the font size of
`shortInstrumentName`."
   ;; This code could be changed or extended to set different values
   ;; for each occurrence of `shortInstrumentName'.
   #{
     \override InstrumentName.after-line-breaking =
       #(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'' }
>>
```



Affichage de la généalogie d'un objet

Lorsque l'on manipule des rappels d'objet (*grob callbacks*), il peut être intéressant d'en maîtriser les ascendants. La plupart des objets graphiques ont des parents, lesquels auront une influence sur le positionnement de l'objet en question. Ainsi, les parents des côtés X et Y influenceront respectivement les positions horizontale et verticale de l'objet. De plus, chacun des parents peut avoir ses propres parents.

Certains aspects de la lignée d'un objet peuvent toutefois porter à confusion :

- Les types de parents d'un *grob* peuvent dépendre du contexte.
- Dans le cas de certains *grobs*, les parents X et Y peuvent être le même.
- Un « ascendant » particulier peut dépendre d'un *grob* de différentes manières.
- Le concept de « générations » est trompeur.

Par exemple, l'objet `System` peut, vis à vis d'un objet `VerticalAlignment`, être à la fois parent (par son côté Y) et grand parent (par deux fois du côté X).

La macro définie ci-dessous affiche à l'écran une représentation textuelle de l'ascendance d'un *grob*. Par exemple, appeler

```
{
  \once \override NoteHead.before-line-breaking = #display-ancestry
  c
}
```

génère la sortie suivante :

```
-----------------------------------
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
```

Par voie de conséquence, il vous faut exécuter vous-même le code de cet extrait, puisque le fichier résultant ne contient pas les données qui nous intéressent.
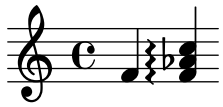
```
#(define (get-ancestry grob)
   (if (not (null? (ly:grob-parent grob X)))
       (list (grob::name grob)
             (get-ancestry (ly:grob-parent grob X))
             (get-ancestry (ly:grob-parent grob Y)))
       (grob::name grob)))

#(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
}
```
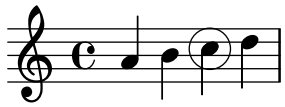


Encerclement de notes

Un cercle peut venir entourer une note à l'aide d'une fonction Scheme dédiée qui, temporairement, viendra en modifier la propriété `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'' }
```



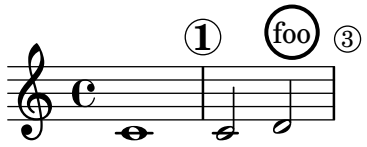
Encercler divers objets

La commande `\circle` permet de dessiner un cercle autour d'objets `\markup`. D'autres objets nécessitent de faire appel à des techniques spécifiques. Cet exemple illustre deux manières de procéder : pour les repères, et pour les numéros de mesure.

```
\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)
}
```



Rappel du glissando à l'occasion d'une alternative

Un glissando qui se prolonge sur plusieurs sections \alternative peut se rappeler à l'aide d'une note d'ornement supplémentaire et masquée, à laquelle sera attaché le départ du glissando, ce dans chaque bloc \alternative. Cette note d'ornement devrait avoir la même hauteur que la note où commençait le glissando originel. Ceci est géré par une fonction musicale qui prendra en argument la hauteur de la note d'ornement.

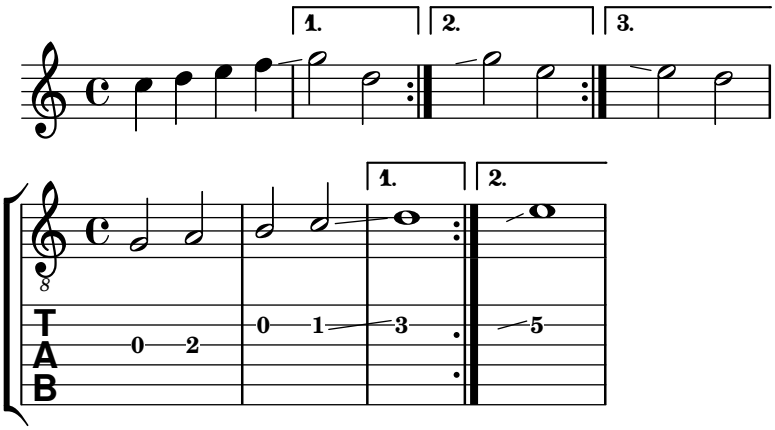
Dans le cadre d'une musique polyphonique, il ne faudra pas oublier d'ajouter une note d'ornement dans toutes les autres voix afin de préserver la synchronisation.

```
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 }
    >>
  >>
}
```



Liaison de prolongation aplatie

La fonction `flared-tie` ici présentée permet de dessiner une liaison de prolongation constituée de lignes droites. Elle est sensée remplacer la fonction par défaut de dessin des liaisons de tenue – elle remplace l'argument `stencil` de l'objet `Tie`.

L'argument de cette fonction `flared-tie` est une liste de paires de coordonnées qui spécifient des points additionnels entre l'entame et la terminaison afin de déterminer l'étendue des lignes de la liaison. L'entame et la terminaison sont identiques à ceux de la liaison originelle. La valeur des abscisses et ordonnées sont des multiples de largeur et de la hauteur de la boîte englobante de la liaison originelle (tout en tenant compte de son positionnement) ; par voie de conséquence, le premier point a pour coordonnée (0,0), et le dernier point (1,0).

La fonction `flare-tie` définit un raccourci pour l'obtention d'une liaison aplatie. Des ajustements plus poussés sont possibles, en apportant des dérogations à `Tie.details.height-limit` ou à l'aide de la fonction `\shape`. La définition personnalisée peut aussi se modifier à la volée.

```
#(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

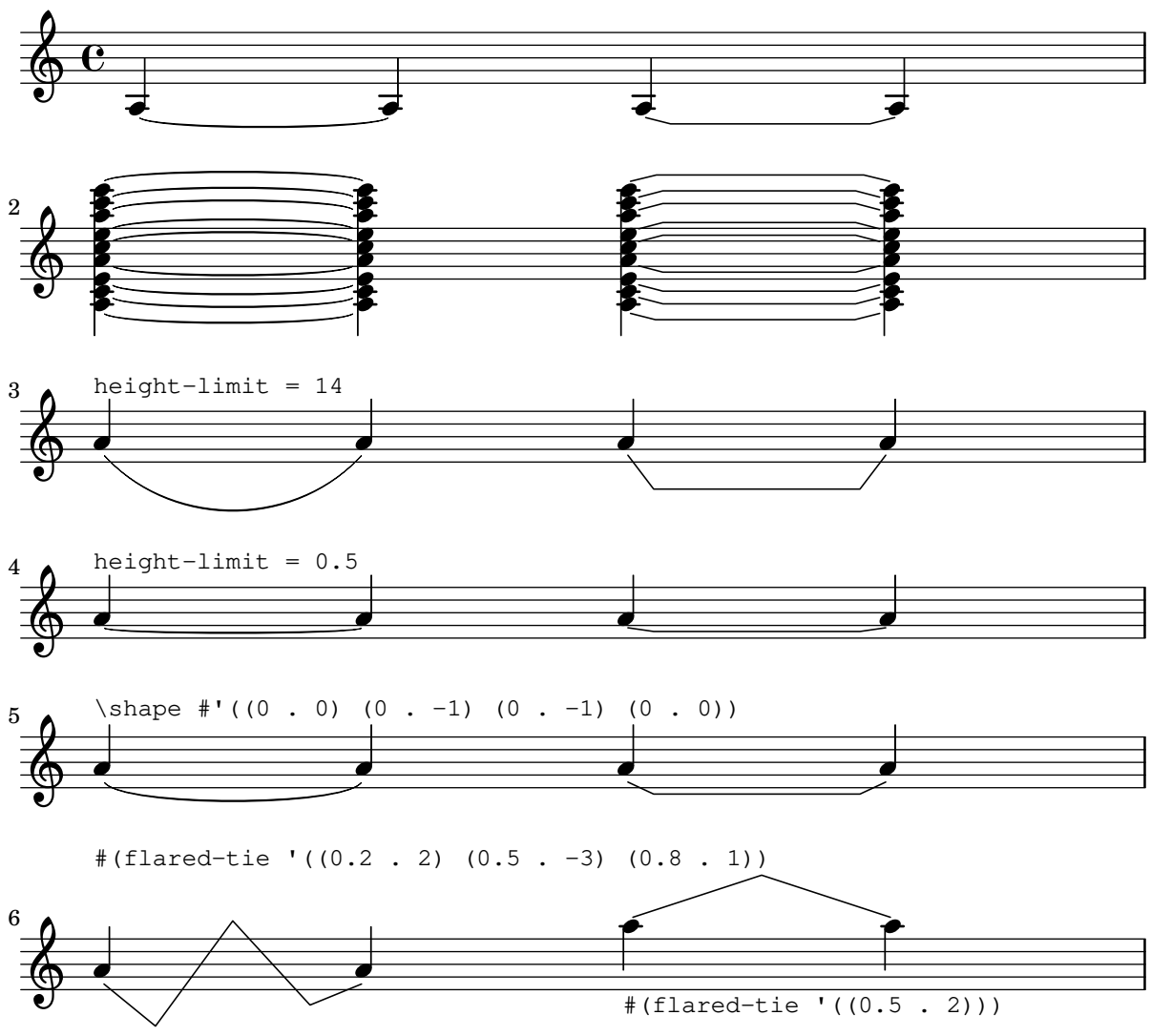
  <>^\markup \small \typewriter "height-limit = 14"
  \override Tie.details.height-limit = 14
  a'4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <>^\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
}
```
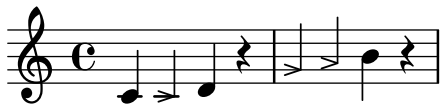


Coup de langue à la flûte

L'indication d'une technique particulière d'articulation, telle le « coup de langue » des flûtistes, s'obtient en remplaçant la tête de note par un glyphe approprié. Pour ce faire, on peut dessiner une tête de note en forme d'accent à l'aide d'un \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
}
```



Diagrammes de fret alternatifs

Vous pouvez tout à fait créer des tables de diagrammes de fret supplémentaires, notamment dans l'optique d'un fret alternatif pour un accord donné. Avant de pouvoir utiliser un diagramme alternatif, vous devrez avoir alimenté une table à cet effet. Les différents diagrammes seront ajoutés à cette table.

Il peut aussi bien s'agir d'une table vide, que de la recopie d'une table existante. La table servant de base pour les diagrammes prédéfinis est sélectionnée par la propriété \predefinedDiagramTable.

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

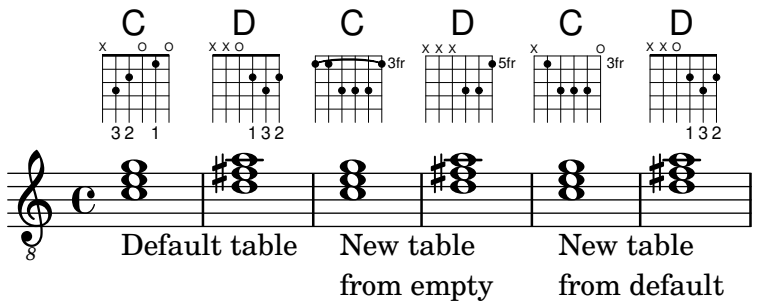
% Make a blank new fretboard table.
#(define custom-fretboard-table-one
   (make-fretboard-table))
```

```
% 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 |
      }
    >>
  }
>>
```



Génération de notes profilées spéciales

Lorsque le profil spécifique d'une tête de note ne peut être obtenu facilement à l'aide d'un *markup* graphique, il est possible d'utiliser des instructions de dessin pour `ly:make-stencil` afin de générer ce profil. Cet exemple propose une tête de note profilée en parallélogramme.

Malheureusement, les commandes disponibles pour spécifier le dessin ne sont pas documentées à ce jour – c'est l'objet du ticket #6874 (`https://gitlab.com/lilypond/lilypond/-/issues/6874`) – mais la sous-commande `path` a la signature suivante, très proche de la fonction Scheme `make-path-stencil`.
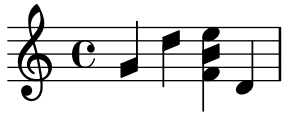
```
(path épaisseur liste-commandes style-ligne-principale style-jonction-ligne remplissage
```

Les commandes dans *liste-commandes* ressemblent aux instructions de dessin en PostScript, à ceci près que les arguments suivent ici le nom de la commande.

```
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
}
```



Génération de crochets personnalisés

Une fonction Scheme personnalisée permet de redéfinir la propriété `stencil` de l'objet `Flag`, de sorte à modifier le glyphe utilisé pour les crochets de croche.

```
#(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
}
```



Génération de notes aléatoires
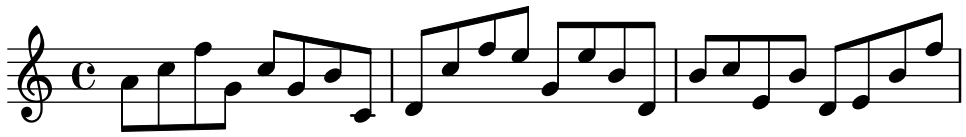
Le fragment de code Scheme ci-dessous génère aléatoirement des notes. Il s'utilise ainsi :

```
\randomNotes n de à dur
```

pour générer *n* notes dont la hauteur varie entre *de* et *à* sur une durée de *dur*.

```
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
```



Génération en Scheme de partitions complètes (y compris des parties d'ouvrage) sans utiliser l'analyseur

Une partition LilyPond, de manière interne, n'est rien d'autre qu'une expression Scheme générée par l'analyseur syntaxique de LilyPond. Il est donc possible, à l'aide de Scheme, de générer automatiquement une partition sans fichier source. Une expression musicale en Scheme sera transformée en partition par un appel à

```
(scorify-music music)
```

Ceci aura pour effet de générer un objet `score` auquel sera appliqué un bloc `layout` comportant la fonction

```
(let* ((layout (ly:output-def-clone $defaultlayout)))
   ; modification de la mise en forme, puis assignation :
   (ly:score-add-output-def! score layout))
```

Il suffit alors de transmettre ce `score` à LilyPond pour qu'il le grave. Les trois fonctions – `(add-score score)`, `(add-text text)` et `(add-music music)` – définies dans le code ci-dessous permettent de transmettre à LilyPond, aux fins de les graver, une partition complète, un *markup* ou simplement de la musique.

Cet exemple permet aussi de graver les pièces contenues dans un bloc `\book { ... }` ainsi que des partitions de niveau supérieur. Chaque partition destinée à être gravée est alors ajoutée à la liste des partitions de niveau supérieur ; le `toplevel-book-handler` – fonction Scheme appelée pour traiter un *book* dès que le bloc `\book { ... }` est clôturé – s'adapte pour prendre en charge tous les éléments `score` jusque là collectés dans l'ouvrage.

Attention : Pour des raisons techniques, seul le premier `\book` sera rendu puisque les autres commandes `\book` génèrent des fichiers additionnels.

```
#(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
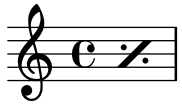


Répétition en pourcent isolée

Des symboles de pourcentage isolés peuvent aussi être obtenus au moyen d'un silence multimesure dont on modifie l'aspect :

```
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
}
```



Easy play – chiffres en lieu et place des lettres

En mode « easy play », les têtes de note utilisent la propriété `note-names` attachée à l'objet `NoteHead` pour déterminer ce qui apparaîtra dans la tête. Intervenir sur cette propriété permet d'imprimer un chiffre correspondant au degré dans la gamme.

La création d'un graveur dédié permet de traiter toutes les notes.

```
#(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)

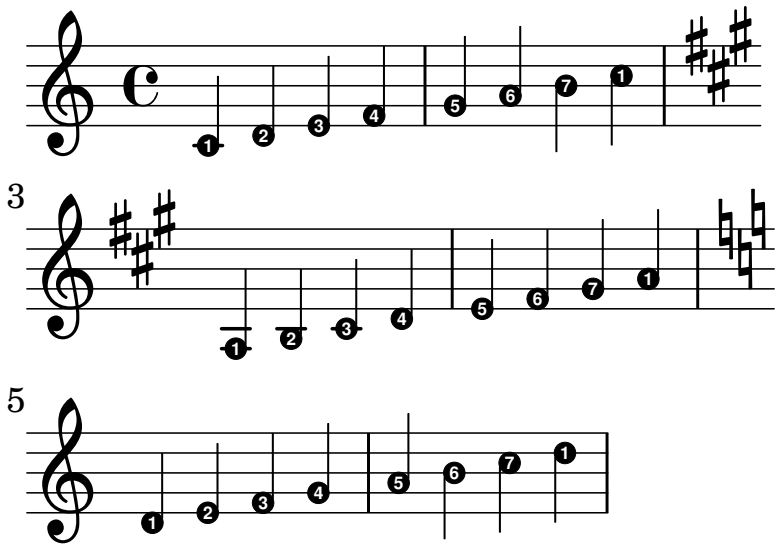
\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\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
}
```



Personnalisation de certains types d'articulation

On peut parfois vouloir modifier un seul type d'articulation. Bien que ce soit tout à fait faisable avec un \tweak, cela devient vite fastidieux d'affiner chaque occurrence d'un même signe dans toute une partition. Le code ci-dessous illustre la manière de modifier des articulations grâce à une liste de réglages personnalisés. Ceci peut servir à créer des feuilles de style.

```
#(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

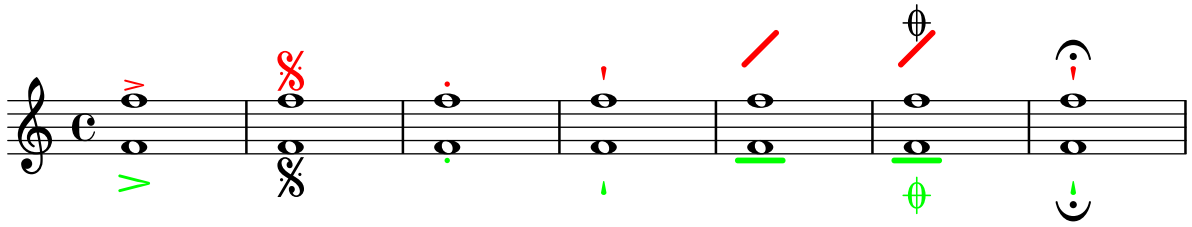
% Predefine two sets of desired tweaks.
#(define my-settings-1
   '((accent . ((font-size . 0)
                (color . (1 0 0))))
     (segno . ((font-size . 0)
               (color . (1 0 0))))
     (staccato . ((color . (1 0 0))
                  (padding . 0.5)))
     (staccatissimo . ((padding . 1)
                       (color . (1 0 0))))
     (tenuto . ((color . (1 0 0))
                (rotation . (45 0 0))
                (padding . 2)
                (font-size . 10)))
    ))

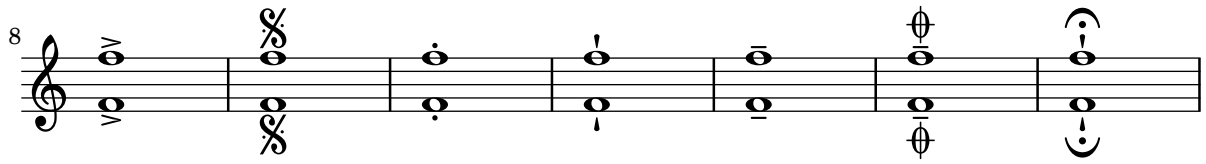
#(define my-settings-2
  '((accent . ((font-size . 4)
               (color . (0 1 0))
               (padding . 1.5)))
    (coda . ((color . (0 1 0))
             (padding . 1)))
    (staccato . ((color . (0 1 0))))
    (staccatissimo . ((padding . 2)
                      (color . (0 1 0))))
    (tenuto . ((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 }
>>
```





Positionnement des notes d'ornement avec espace flottant

Lorsque la propriété `strict-grace-spacing` est activée, l'espacement des notes d'ornement se fera de manière « élastique ». Autrement dit, elles seront décollées de leur note de rattachement : LilyPond commence par espacer les notes normales, puis les ornements sont placés à la gauche de leur note de rattachement.

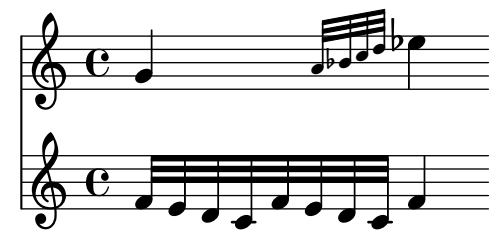
Cependant, en raison du ticket 6876 (`https://gitlab.com/lilypond/lilypond/-/issues/6876`), les altérations accidentelles sont ignorées lorsque cette propriété est activée. Le code ci-dessous propose une solution de contournement à ce problème.

Autre effet indésirable de cette propriété, LilyPond ne vérifie pas qu'il y a suffisamment d'espace pour les notes d'ornement (ceci fait l'objet du ticket 2630 (`https://gitlab.com/lilypond/lilypond/-/issues/2630`)). Il faut donc s'assurer d'avoir suffisamment d'espace disponible en recourant conjointement `\newSpacingSection` et une valeur appropriée pour le `base-shortest-duration` de l'objet graphique `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
  }
}
```



Impression de noms d'accords identiques à la basse différente par une oblique et nouvelle basse

Voici comment imprimer une suite d'accords nommés (objets ChordNames) dont seule la basse diffère, indiquée par une oblique et cette nouvelle basse, à l'aide du graveur Scheme ici défini. Son comportement peut se gérer finement à l'aide de la propriété de contexte chordChanges.

```
#(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

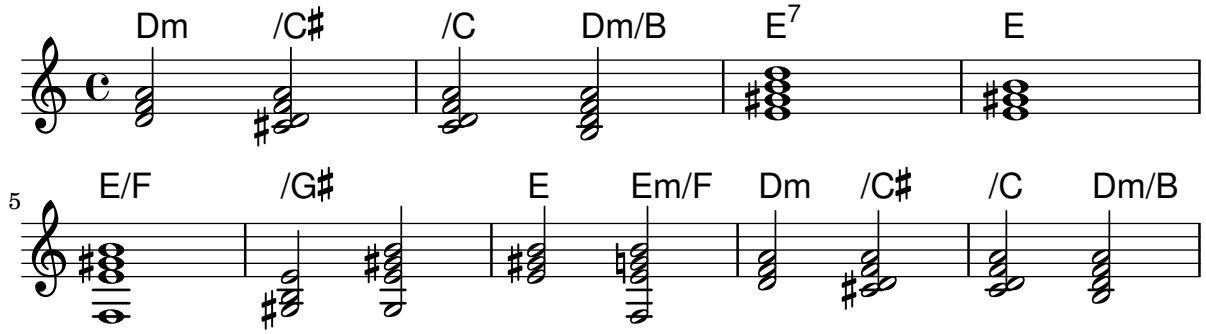
  d:m/c
  \set chordChanges = ##f
  d:m/b

  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
>>
```



Réglage de l'égalisation par défaut des instruments MIDI

L'égaliseur basique peut être modifié par la définition d'une nouvelle procédure Scheme `instrumentEqualizer` au sein du contexte `Score`. Cette procédure prend en unique argument le nom d'un instrument MIDI et renverra une paire de fractions correspondant aux minimum et maximum de volume alloué à cet instrument.

Dans l'exemple suivant sont réglés les volumes relatifs de la flûte et de la clarinette.
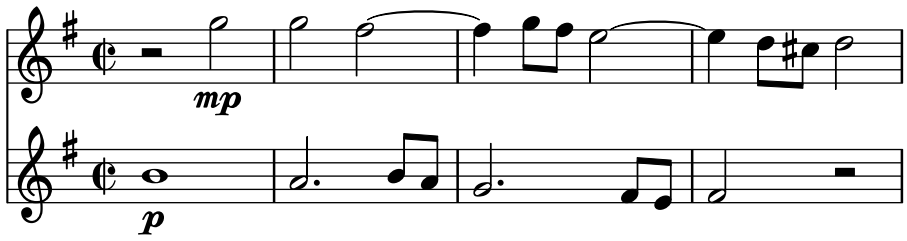
```
#(define my-instrument-equalizer-alist '())

#(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 {  }
}
```



Séparation entre altérations annulées et nouvelle armure

Les altérations supprimées lors d'un changement de tonalité sont par défaut accolées à la nouvelle armure. Ce comportement peut s'adapter grâce à la propriété `break-align-orders` de l'objet `BreakAlignment`.

Si l'on se réfère à la référence des propriétés internes à propos de l'objet Section "break-alignment-interface" dans *Référence des propriétés internes*, l'ordre du deuxième élément est :

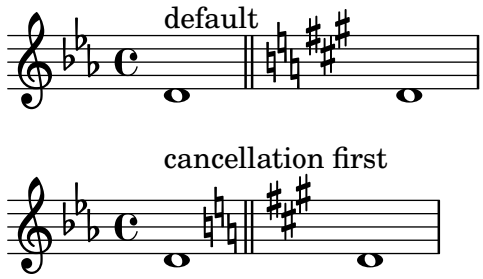
```
ambitus
breathing-sign
clef
cue-clef
cue-end-clef
custos
key-cancellation
key-signature
left-edge
signum-repetitionis
staff-bar
staff-ellipsis
time-signature
```

La consultation de cette liste nous apprend qu'il faut déplacer `key-cancellation` pour le mettre avant `staff-bar`. Ceci se réalise à l'aide de la fonction `\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 }
```
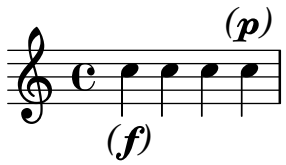


Indications de nuance entre parenthèses en italique

Le moyen le plus simple pour ajouter des parenthèses à une indication de nuance consiste à utiliser la commande \parenthesize et et d'ajuster la taille de la fonte, comme par exemple { c'4\tweak Parentheses.font-size 2 \parenthesize \f }. Toutefois, ces parenthèses se présentent droit alors que les indications de nuance apparaissent normalement en italique. Le code ci-dessous offre une solution pour présenter les parenthèses elles aussi en italique.

```
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
}
```

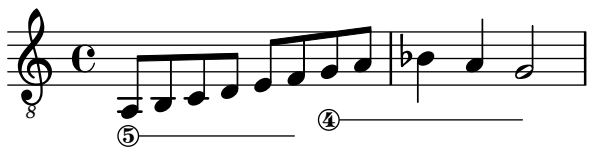


Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
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
}
```



Encadrement sur trois côtés

Voici comment construire une commande de *markup* chargée d'agrémenter du texte ou autre annotation, d'une bordure sur trois côtés.

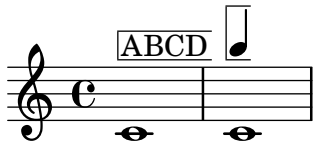
```
% New command to add a three-sided box, with sides north, west, and south.
% Based on the `box-stencil` command defined in `scm/stencil.scm`.
% Note that ";;" is used to comment a line in Scheme.
#(define-public (NWS-box-stencil stencil thickness padding)
   "Add a box around STENCIL, producing a 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 }
}
```



Transposition et réduction du nombre d'altérations accidentelles

Cet exemple, grâce à un peu de code Scheme, donne la priorité aux enharmoniques afin de limiter le nombre d'altérations supplémentaires. La règle applicable est :

- Les altérations doubles sont supprimées
- Si dièse -> Do
- Mi dièse -> Fa
- Do bémol -> Si
- Fa bémol -> Mi

Cette façon de procéder aboutit à plus d'enharmoniques naturelles.

```
#(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)
```
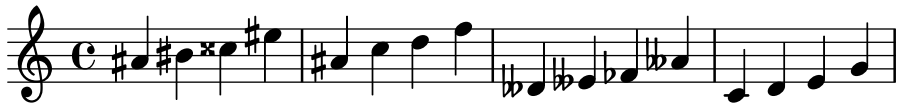
```
      (set! o (+ o 1))
      (set! n (- n 7)))
    (ly:make-pitch o n (/ a 4))))

#(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 }
}
```



Deux \partCombine sur une même portée

La fonction \partCombine prend deux expressions musicales contenant chacune une partie et les distribue dans quatre contextes Voice nommés « two », « one », « solo » et « chords » selon le moment et la manière dont ces parties fusionnent en une voix commune.

Des variantes de \partCombine – \partCombineUp et \partCombineDown – permettent de fusionner des voix aux hampes ascendantes et descendantes. Les combiner alors pour présenter quatre parties sur une seule portée requiert cependant des réglages spécifiques, comme indiqué ci-dessous.

```
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.

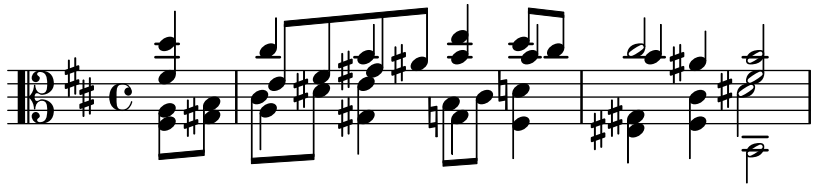
The context is called 'Down'; internally, the function calls
`\\partCombineDown`."
  #{
    \new VoiceBox = "Down" <<
      \set VoiceBox.soloText = #"Solo III"
      \set VoiceBox.soloIIText = #"Solo IV"
      \context Voice ="one" { \voiceFour }
      \context Voice ="two" { \voiceTwo }
      \context 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"
  }
}
```
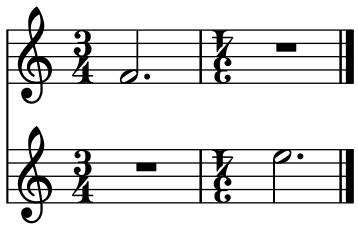


Personnalisation de la métrique

Il est possible de définir de nouveaux styles de métrique. La métrique de la deuxième mesure se présente sens dessus dessous sur les deux portées.

```
#(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''
  }
>>
```



Utilisation de `ly:grob-object` pour accéder aux *grobs* avec `\tweak`

Certains objets graphiques ne sont accessibles que par le biais d'un *callback* à partir d'un autre `grob`. Ils sont normalement listés dans les « *layout objects* » au sein de la section « Propriétés internes » d'une *grob interface*. La fonction `ly:grob-object` permet d'accéder à ces objets.

Voici plusieurs moyens d'accéder aux objets par un *callback* sur `NoteHead`. D'autres biais sont naturellement possibles ; `NoteHead` a cependant l'avantage incontestable d'être utilisé implicitement par la commande `\tweak`.

Voici ce qui sera émis dans la console suite à la fonction `display-grobs` ici construite.

```
--------------------
#<Grob Accidental >
()
#<Grob Stem >
```

Cette fonction n'est probablement pas très utile. Elle indique toutefois qu'il est tout à fait possible d'accéder aux objets.

```
#(define (notehead-get-accidental notehead)
   ;; notehead is grob
   (ly:grob-object notehead 'accidental-grob))

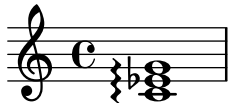
#(define (notehead-get-arpeggio notehead)
   ;; notehead is grob
   (let ((notecolumn (notehead-get-notecolumn notehead)))
     (ly:grob-object notecolumn 'arpeggio)))

#(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
}
```
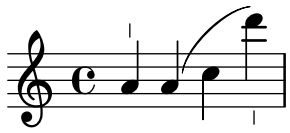


Articulation baroque en forme de coche

On trouve régulièrement, en musique baroque, cette courte ligne verticale. Sa signification peut varier, mais elle indique le plus souvent une note plus « appuyée ». Voici comment générer ce signe particulier.

```
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 Section “Spacing issues” dans *Manuel de notation*.
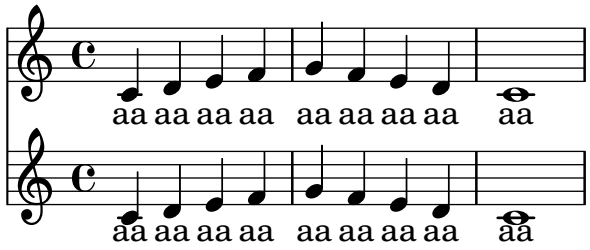
Ajustement de l’espacement vertical des paroles

Le code ci-dessous indique comment rapprocher de la portée à laquelle elle est affectée, une ligne de paroles.

```
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
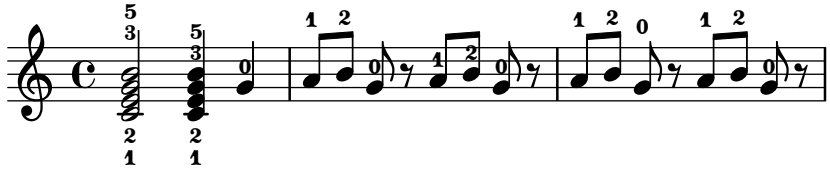
  \new Staff \new Voice = melody \music
  % Reducing the minimum space below the staff and above the lyrics.
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  } \lyricsto melody \text
>>
```



Impression des doigtés à l’intérieur de la portée

L’empilement des indications de doigté se fait par défaut à l’extérieur de la portée. Il est néanmoins possible d’annuler ce comportement. Une attention particulière doit toutefois être portée dans les cas où doigté et hampe vont dans la même direction : les indications de doigté n’évitent les hampe qu’en présence de ligature. Ce réglage peut s’adapter pour éviter toutes les hampes ou aucune d’elles. L’exemple suivant illustre ces deux options, ainsi que la manière de revenir au comportement par défaut.

```
\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
}
```



Rupture d'alignement horizontal des nuances graphiques et textuelles

Par défaut, LilyPond utilise des objets `DynamicLineSpanner` pour aligner horizontalement une succession d'objets nuance tels que soufflets et indications textuelles, et quel que soit leur positionnement par rapport à la portée. Ceci peut, parfois, ne pas être désirable et une rupture de l'étendue de l'alignement n'est pas possible comme en matière d'alignement vertical (voir l'exemple « Rupture d'alignement vertical des nuances graphiques et textuelles »).

Deux solutions permettent de pallier ce désagrément.

- Modifier la propriété `shorten-pair` de l'objet `Hairpin` pour compenser le décalage qui a été appliqué au soufflet.
- Placer les deux objets de nuance dans des voix différentes.

Le code ci-dessous illustre ces deux possibilités.

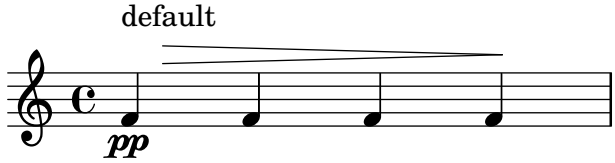
```
{
  <>^"default"
  f'_\pp ^\> f' f' f'\!
}
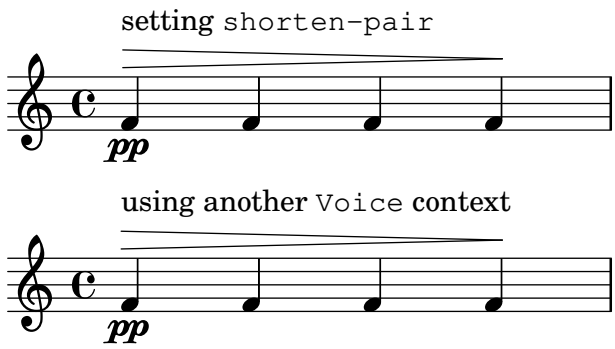
{
  <>^\markup { setting \typewriter shorten-pair }
  f'_\pp \tweak shorten-pair #'(-3 . 0) ^\> f' f' f'\!
}

{
  <>^\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
  }
}
```





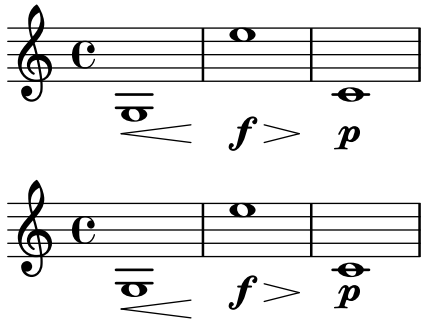
Rupture d'alignement vertical des nuances graphiques et textuelles

Par défaut, LilyPond utilise des objets DynamicLineSpanner pour aligner verticalement une succession d'objets nuance tels que soufflets et indications textuelles. Ceci peut, parfois, ne pas être désirable. L'insertion d'un \breakDynamicSpan, qui rompt prématurément l'étendue de l'alignement, permet de pallier ce désagrément.

Voir aussi l'exemple « Rupture d'alignement horizontal des nuances graphiques et textuelles ».

```
{ 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.

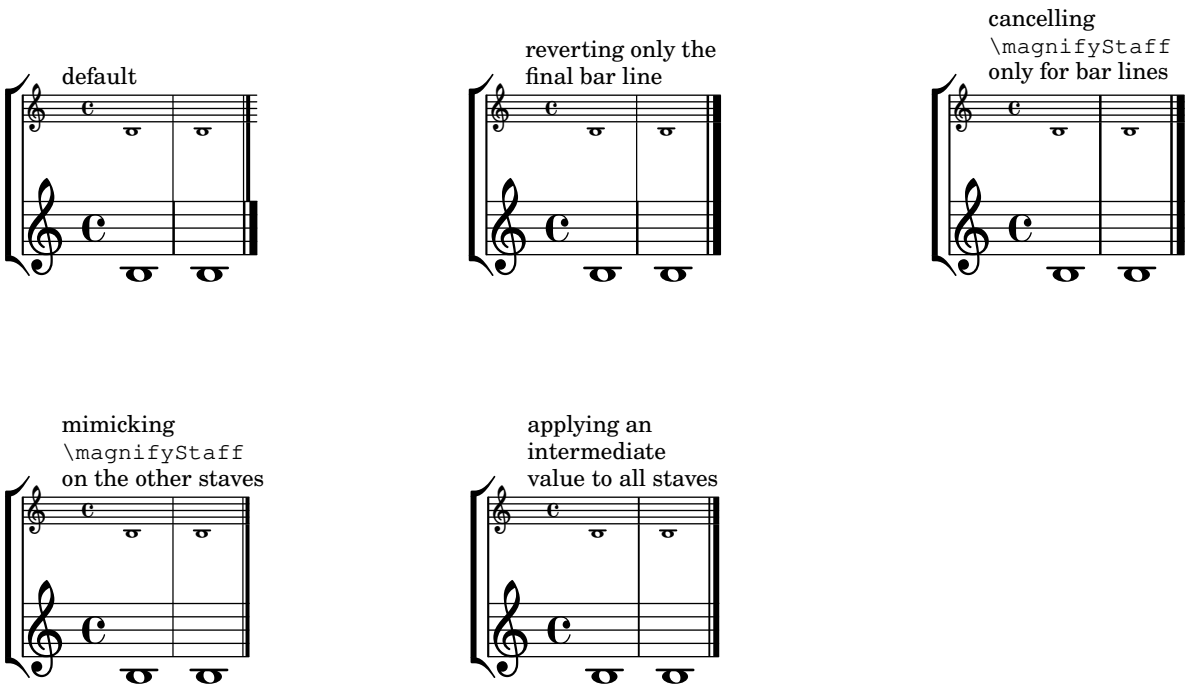
```
\markuplist {
  % First row.
  \fill-line {
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny "default"
          b1 b \bar "|."
        }
        \new Staff { b b }
```

```
    >>
  }
  \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 }
      >>
    }
    ""
  }
}
```



Référencement de page

Des références de page peuvent prendre place aussi bien dans la musique qu'en tête de partition, puis être reprises dans un *markup*.

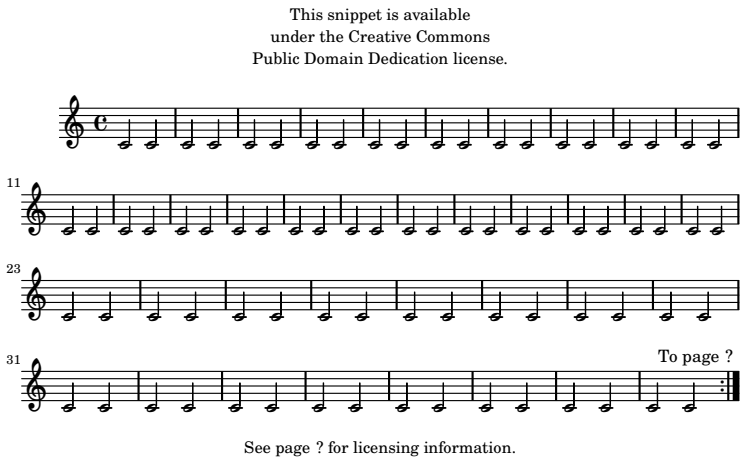
```
#(set-default-paper-size "a7" 'landscape)
#(set-global-staff-size 11)

\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. } }
```



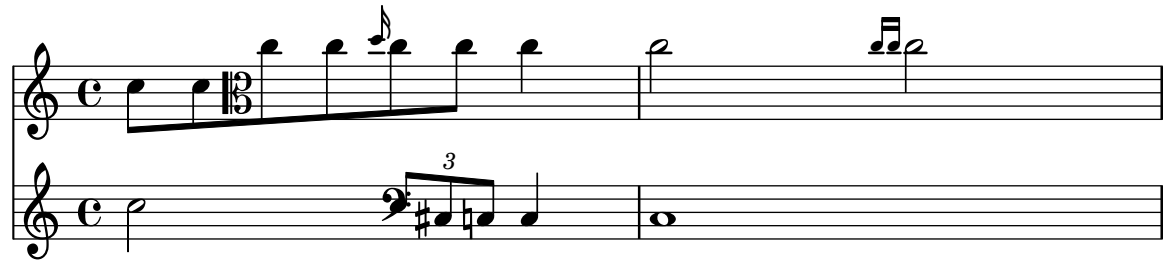
Espacement strictement proportionnel des notes

Lorsque la propriété `strict-note-spacing` de l'objet SpacingSpanner est activée, l'espacement des notes dans un système ne tient compte ni des barres de mesure ni des clefs, qui se retrouvent placées juste avant la note qui tombe au même moment. Ceci peut entraîner certaines collisions.

```
\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
```

```
  }
>>
```



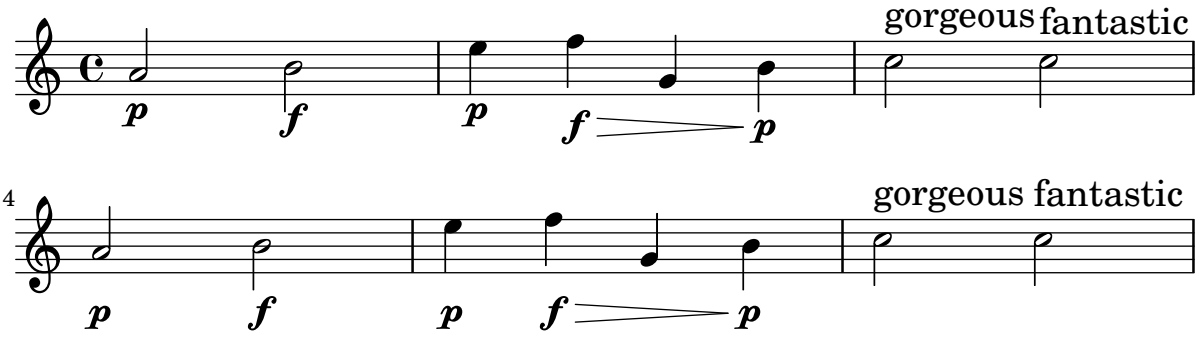
Alignement vertical des nuances et indications textuelles

Tous les objets `DynamicLineSpanner` (soufflets ou nuances textuelles) viennent s'aligner sur une ligne de référence placée, par rapport à la portée, à au moins la valeur de la propriété `staff-padding` sauf lorsque d'autres éléments de notation les en éloignent plus. Les nuances seront centrés sur une même ligne dès lors que `staff-padding` aura été défini à une valeur suffisante.

C'est le même principe – en combinaison avec `\textLengthOn` – qui sert à aligner les indications textuelles sur une ligne de référence.

```
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
}
```

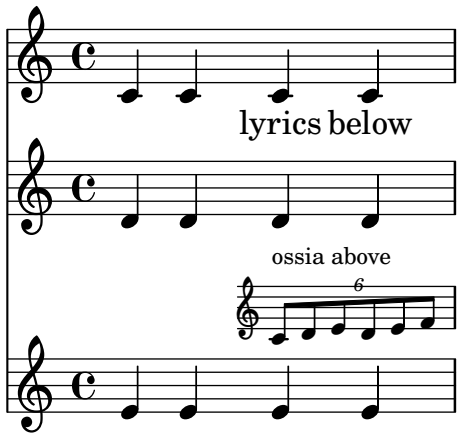


Positionnement d'une ossia et des paroles

Cet exemple illustre la manière de positionner une portée d'ossia et des paroles à l'aide des propriétés de contexte `alignBelowContext` et `alignAboveContext`.

```
\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

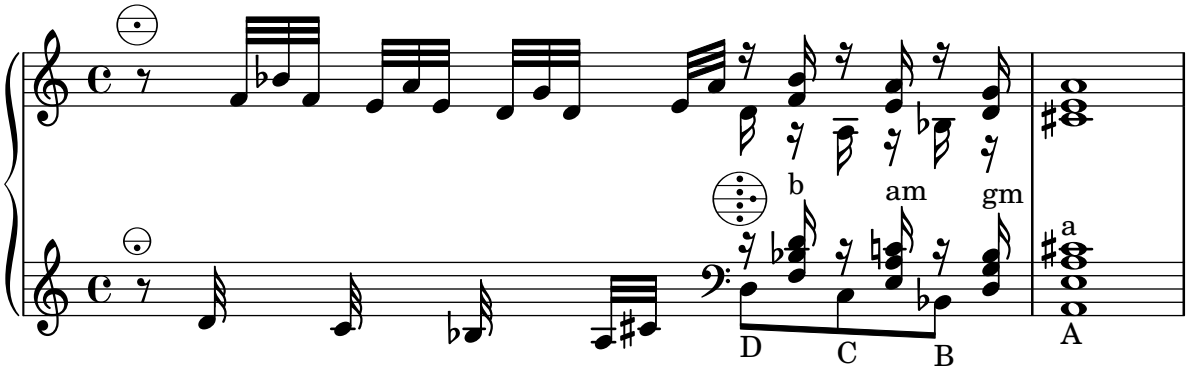
Symboles de registre pour accordéon

Les symboles spécifiques aux registres d'accordéon sont disponible tant sous forme de \markup que d'événements musicaux indépendants – un changement de registre a tendance à intervenir entre des événements musicaux réels. Les registres de basse ne sont pas très standardisés. Les différentes commandes disponibles sont regroupées à l'annexe Section "Registres d'accordéon" dans *Manuel de notation*.

```
#(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" }
    >>
  }
>>
```

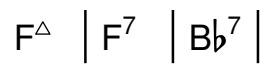


Chiffrages et barres de mesure

L'ajout du graveur `Bar_engraver` à un contexte `ChordNames` permet d'imprimer les barres de mesure entre les chiffrages.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```



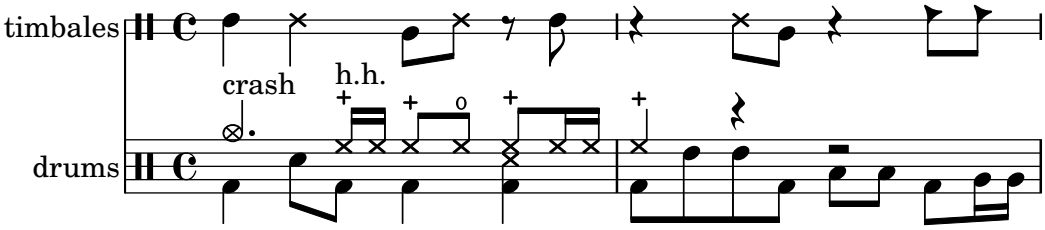
Ajout de parties de batterie

Grâce à la puissance des outils préconfigurés tels que la fonction `\drummode` et le contexte `DrumStaff`, la saisie de parties pour percussions est extrêmement simplifiée : chaque composant d'une batterie trouve sa place sur une portée dédiée (avec une clef particulière) et les têtes de note sont spécifiques à chaque élément. Il est également possible d'affecter un symbole particulier à chaque élément, tout comme de restreindre le nombre de lignes de la portée.

```
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 }
}
```

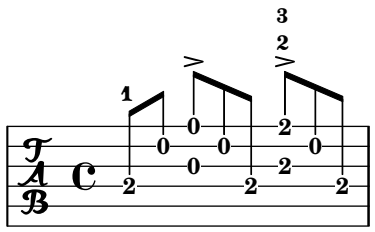


Ajout de doigtés à des tablatures

L'ajout de doigtés à des tablatures s'obtient en conjuguant des \markup et des \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]
  }
}
```
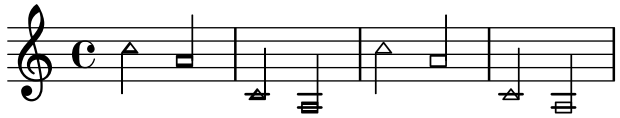


Variante fine des notes profilées Aiken

Les notes profilées Aiken, lorsqu'elles sont blanches, deviennent difficile à distinguer avec des tailles de portée réduites, notamment en présence de ligne supplémentaire. Perdant du blanc sur leur intérieur les fait alors ressembler à des noires.

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

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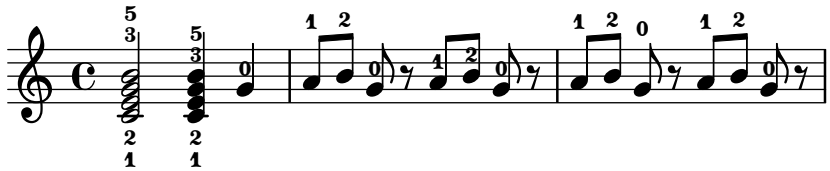
```
                                faThin sol laThin tiThin)
    c'' a' c' a
  }
}
```



Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Il est néanmoins possible d'annuler ce comportement. Une attention particulière doit toutefois être portée dans les cas où doigté et hampe vont dans la même direction : les indications de doigté n'évitent les hampe qu'en présence de ligature. Ce réglage peut s'adapter pour éviter toutes les hampes ou aucune d'elles. L'exemple suivant illustre ces deux options, ainsi que la manière de revenir au comportement par défaut.

```
\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
}
```



Modification du nombre de lignes de la portée

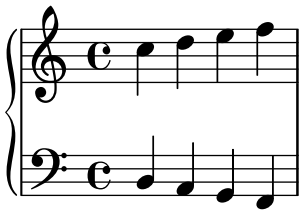
Le nombre de lignes d'une portée se modifie par adaptation de la propriété `line-count` du `StaffSymbol`.

```
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
    }
  >>
}
```

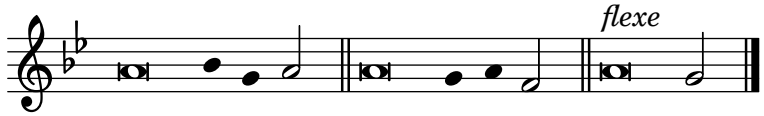


Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```
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
  }
}
```



Modèles de chiffrage d'accords

La propriété `chordNameExceptions` enregistre une liste de chiffrages personnalisés afin de traiter des cas non prévus ou incorrectement gérés.

Par défaut, LilyPond utilise le jeu de chiffrages issus de l'ouvrage de Klaus Ignatzek « Die Jazzmethode für Klavier 1 » ; l'algorithme de conversion des accords en chiffrages est inclus dans le fichier `scm/chord-ignatzek-names.scm`. Des règles additionnelles pour gérer des chiffrages exceptionnels sont enregistrées en tant que variables `ignatzekExceptions` dans le fichier `ly/chord-modifiers-init.ly`.

Le code ci-dessous modifie ces exceptions en trois étapes.

1. Saisie de musique en accords avec *markup* associé. Par convention, la base – autrement dit la note la plus basse – de chaque accord aura une hauteur de do (`c` pour les anglophones).

2. Appel de la fonction Scheme `sequential-music-to-chord-exceptions` pour créer une liste de nouvelles exceptions, puis l'ajouter à celles déjà répertoriées. Dans la mesure où `ignatzekExceptions` est défini avec le second paramètre de cette fonction fixé à `#t`, afin d'ignorer la base de chaque accord, il faut faire de même.
3. Enregistrement de la nouvelle liste d'exceptions.

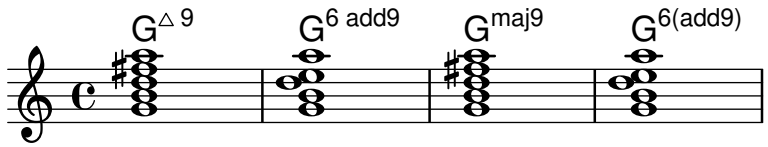
```
% Step 1: Define music with chords and markup for maj9 and 6(add9).
chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

% 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
}
```



Chiffrage d'un maj7

La représentation d'un accord de septième majeure se gère par la propriété de contexte `majorSevenSymbol`.

```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```

C△ Cj7

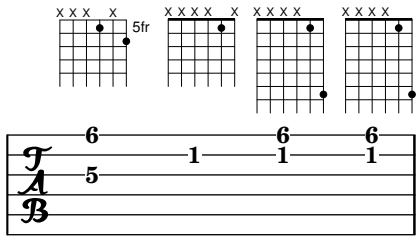
Accords avec doigté espacé pour FretBoards et TabVoice

Il peut arriver que le doigté d'un accord soit assez étendu. Sauf mention contraire, la valeur de la propriété de contexte maximumFretStretch est cependant fixée à 4, ce qui peut générer un avertissement « Pas de corde pour la hauteur... » et la note est omise. On peut régler maximumFretStretch sur une valeur appropriée ou assigner explicitement leur numéro de corde à toutes les notes d'un accord afin de palier ce problème.

```
% 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
>>
```

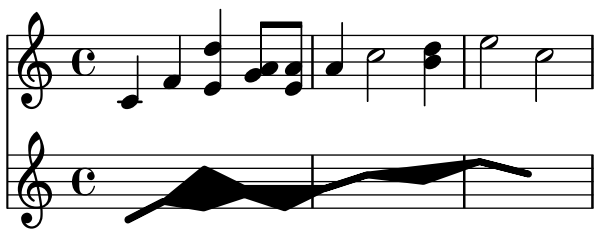


Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

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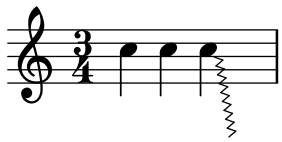
<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Glissando contemporain

De nos jours, il peut arriver que la note d'arrivée d'un glissando soit absente de la partition. Pour ce faire, il vous faudra utiliser une cadence et « masquer » la note d'arrivée.

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



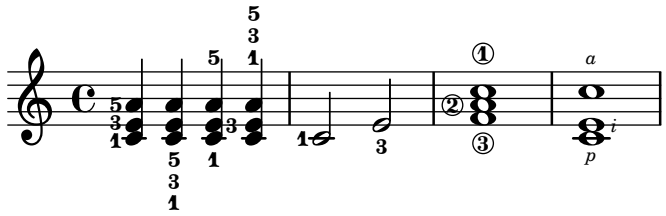
Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise à l'aide de la propriété `fingeringOrientation`. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `<...>`, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue à l'aide des propriétés `stringNumberOrientation` et `strokeFingerOrientation`.

Ces propriétés peuvent prendre une liste de trois valeurs. Elles contrôlent si les informations doivent être placées au-dessus (en présence d'un `up`) ou au-dessous (en présence d'un `down`), à gauche (en présence d'un `left` ou à droite (en présence d'un `right`). Par contre, si un positionnement n'est pas mentionné, aucun doigté n'y figurera. LilyPond tient compte de ces contraintes et trouvera le meilleur emplacement pour le doigté des notes de l'accord qui suit. Bien entendu, `left` et `right` sont exclusifs l'un de l'autre – les doigtés ne peuvent se placer que d'un seul côté.

```
\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 >
```

```
}
```



Exemple de sonnaille et clochette

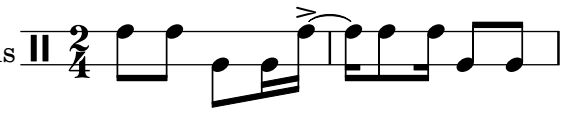
Voici deux différentes cloches, saisies « cb » (*cow bell* pour sonnaille) et « rb » (*ride bell* pour clochette).

```
#(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 |
}
```



Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le `Bar_number_engraver` du contexte `Score`, et les `Time_signature_engraver`, `Clef_engraver` et `Bar_engraver` du contexte `Staff`.

```
#(set-global-staff-size 10)   % for the documentation
% #(set-global-staff-size 20) % for letter and A4

\book {
  \score {
    { \*12 { s1 \break } }

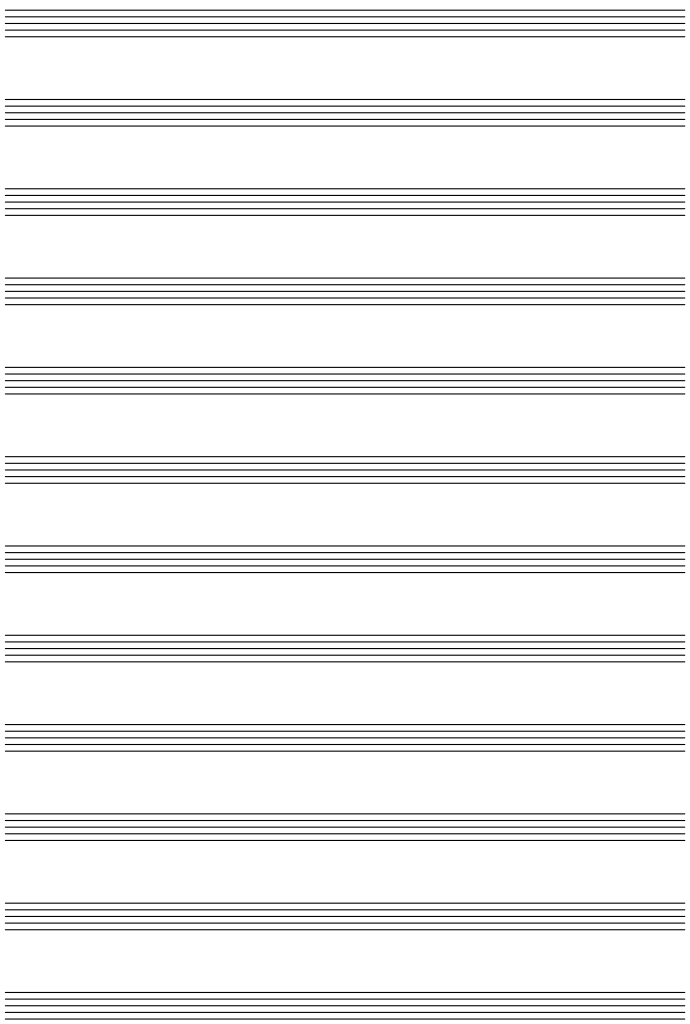
    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
```

```
      \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
  }
  %}
}
```



Guidons

Les guidons peuvent adopter différents styles.

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

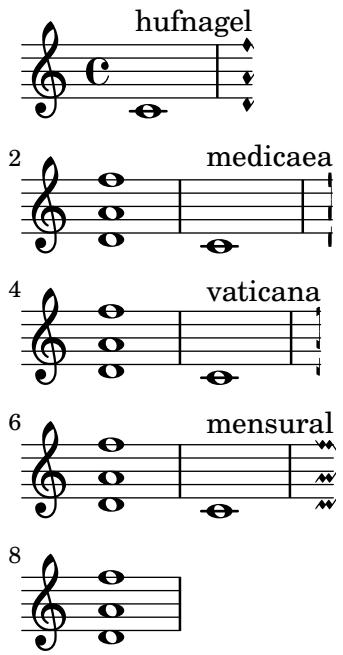
\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

    \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
  }
}
```



Démonstration des instruments MIDI

Partant du problème que l'on ignore quelle valeur de `midiInstrument` serait la plus adaptée à sa composition, la compilation de ce fichier permet de faire son choix.

```
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 { }
}
```
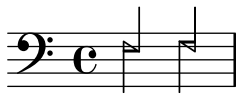


Orientation de la tête d'un « fa » profilé en cas de fusion

Grâce à la propriété `NoteCollision.fa-merge-direction`, l'orientation de la tête des notes profilées « fa » (`fa`, `faThin`, etc.) peut se gérer indépendamment de la direction des hampes lorsque deux voix ayant la même hauteur et des directions de hampes différentes sont fusionnées. Lorsque cette propriété n'est pas activée sera utilisée la variante « descendante ».

```
{
  \clef bass

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



Intégration de PostScript dans un *markup*

Du code PostScript peut directement être intégré dans un bloc `\markup`.

En règle générale, il est préférable d'utiliser la commande de *markup* `\polygon` qui est fonctionnelle avec tous les moteurs de rendu 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
}
```



Les graveurs un par un

LilyPond gère les différents éléments nécessaires à la réalisaon d'un partision grâce à des modules. Chaque module est appelé « graveur ». Dans cet exemple, quelques graveurs sont appelés chacun à son tour, dans l'ordre suivant :

- têtes de note,
- symboles de la portée,
- clef,
- hampes,
- ligatures, liaisons, accents,
- altérations, barres de mesure, métrique et armure.

Les graveurs se regroupent. Par exemple, têtes de note, liaisons, ligature, etc. forment un contexte de voix (`Voice`). Les graveurs chargés de la métrique, des altérations, des barres de mesure, etc. forment un contexte de portée (`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
```
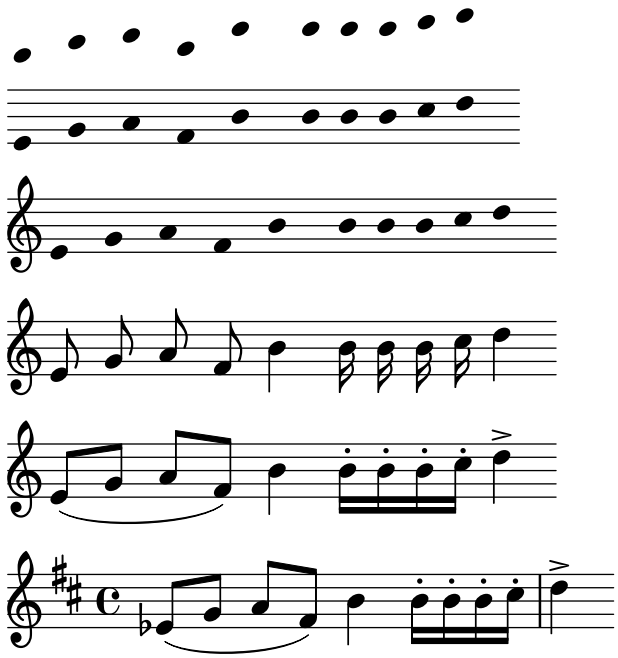
```
  \consists Staff_symbol_engraver
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% 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 }
  }
}
```



Notation de flamenco

La guitare flamenco fait appel à des éléments de notation particuliers :

- le *golpe* : coup frappé sur la caisse de résonance avec l'ongle de l'annulaire,
- une flèche pour indiquer le sens des butés (*strokes*),
- les doigtés : « p » pouce, « i » index, « m » majeur, « a » annulaire et « x » auriculaire,
- les *rasgueados* sur trois ou quatre doigts : grattage des cordes en étendant les doigts rapidement les uns après les autres dans un mouvement continu, se terminant sur l'index,
- les *abanicos* : séries de butés du pouce, index et auriculaire. Il existe aussi un *abanico 2* pour lequel l'index et l'annulaire remplacent l'auriculaire.
- *alza púa* : jeu rapide du pouce.

La plupart de ces symboles utilisent une flèche en plus des doigtés. Dans le cas d'un *abanico*, les têtes de note ne sont imprimées que pour le premier accord.

Le début du code ci-dessous répertorie le paramétrage de ces différents symboles, que vous pouvez copier dans un fichier `flamenco.ly` pour inclusion dans vos propres compositions.

```
%%%%%%  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
}

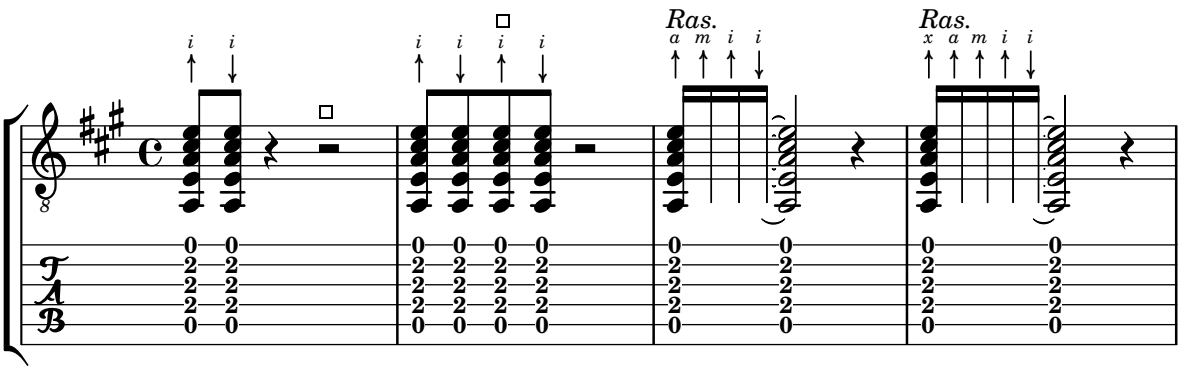
%%%%%%  Cut here ----- End of `flamenco.ly`.

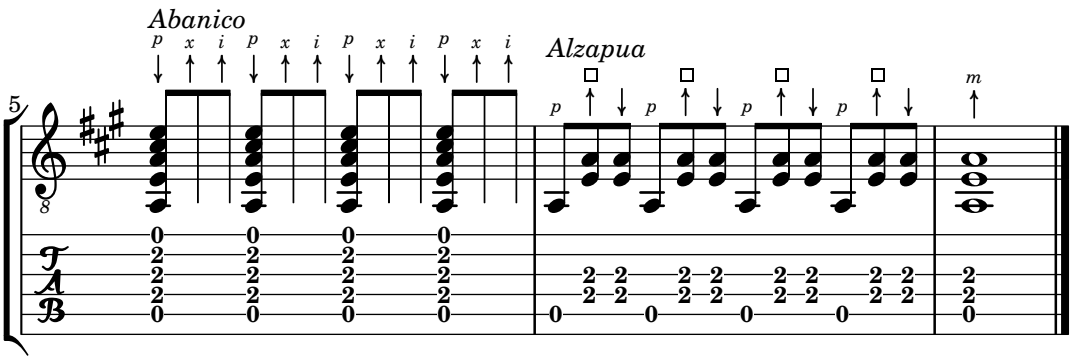

part = \relative c' {
  \set strokeFingerOrientations = #'(up)
  \key a\major

  <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
  }
}
```





Exemple de woodblocks haut et bas

Deux woodblocks se saisissent « wbh » (woodblock haut) et « wbl » (woodblock bas). La hauteur des barres de mesure a été adaptée par une commande \override afin qu'elles ne soient pas trop courtes. La position des deux lignes de la portée doit aussi être explicitement définie.

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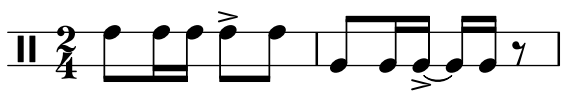
woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

  % 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 |
  }
}
```



Repositionnement d'un diagramme de fret

Différents moyens permettent de repositionner un diagramme de fret pour éviter des collisions ou le placer entre deux notes :

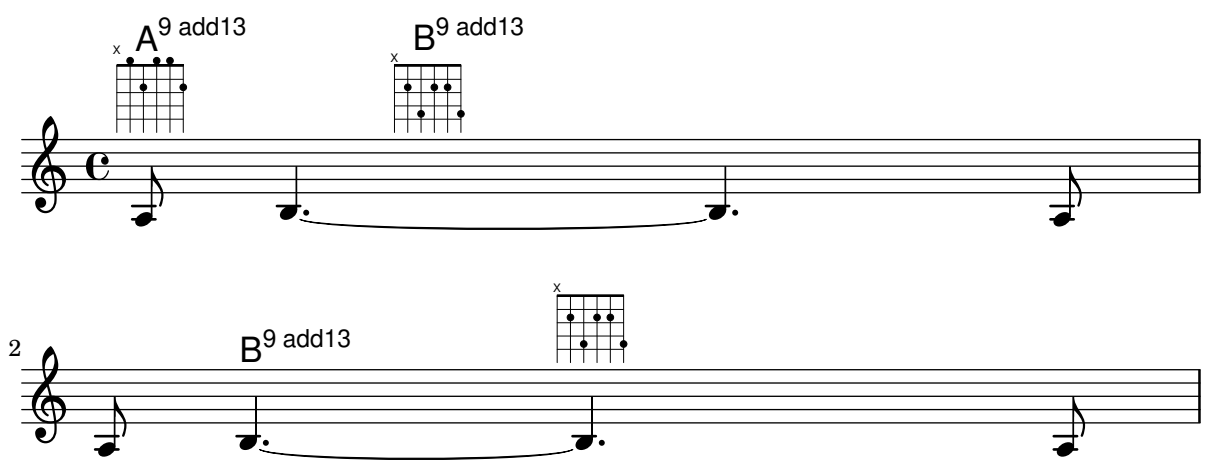
1. La modification des valeurs de `padding` ou de `extra-offset` comme pour la première ligne ;
2. L'adjonction d'une voix invisible dans laquelle les diagrammes sont attachés à des notes invisibles comme pour la deuxième ligne.

Lorsque le diagramme doit correspondre à une position rythmique dans la mesure, comme au troisième temps de la deuxième mesure, la seconde méthode est plus appropriée puisque le diagramme sera aligné sur le temps.

```
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;" }
      >> |
    }
  >>
}
```



Séparation de syllables par une liaison

Des mots monosyllabiques peuvent s'attacher par une liaison à l'aide d'un tilde.

```
\lyrics {
  wa~o~a
}
```
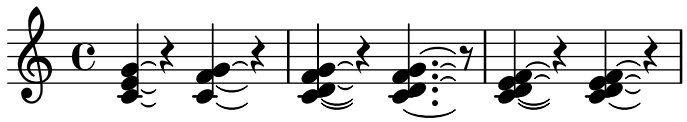
wa‿o‿a

Liaison « Laissez vibrer »

Les indications de « laisser vibrer » ont une taille fixe. Leur formatage est accessible au travers de la propriété `tie-configuration`.

```
\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
}
```



Exemple pour percussions

Ce court exemple est tiré de « L'histoire du Soldat » de Stravinsky.

```
#(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
}

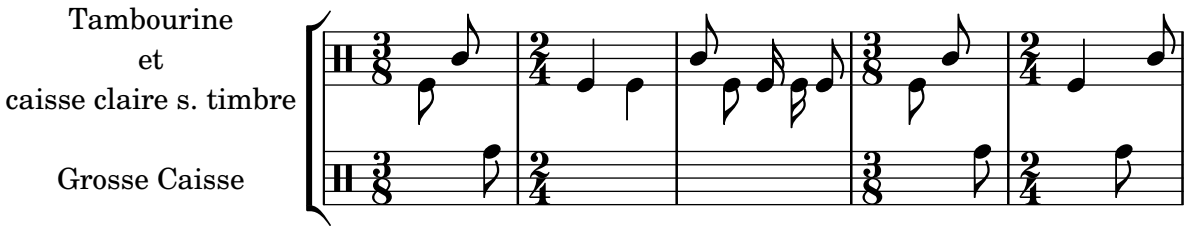
drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
```

```
      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
  >>
}
```



Exemple pour tam-tam

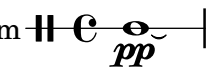
Un simple exemple pour tam-tam, saisi « tt ».

```
#(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 

Exemple pour tambourin

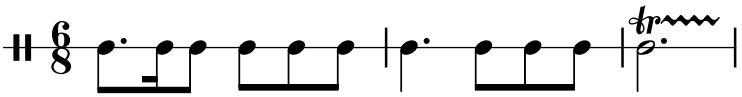
Voici une partie de tambourin, saisi par « tamb ».

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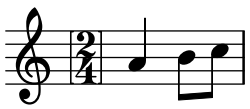
  \time 6/8
  tamb8. 16 8 8 8 8 |
  tamb4. 8 8 8 |
  % The trick with the scaled duration and the shorter rest
  % is neccessary for the correct ending of the trill-span!
  tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
}
```

Tambourine 

Métrique entre crochets

La métrique peut être mise entre crochets.
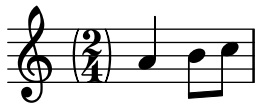
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Métrique entre parenthèses

Une métrique peut être mise entre parenthèses.

```
\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
}
```



Recours à une voix supplémentaire pour gérer les sauts

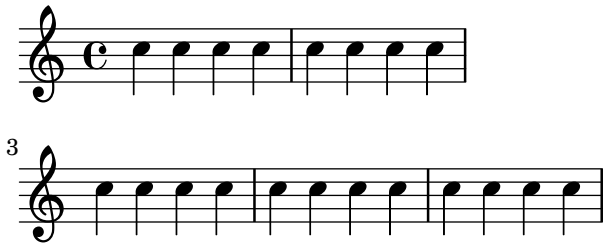
Il est souvent plus pratique de séparer ce qui est purement musical et les informations concernant les sauts de ligne ou de page, en créant une voix supplémentaire dédiée. Cette voix spécifique ne contiendra que des blancs – des silences invisibles `\skip` –, des `\break`, des `\pageBreak` et autres informations concernant les ruptures.

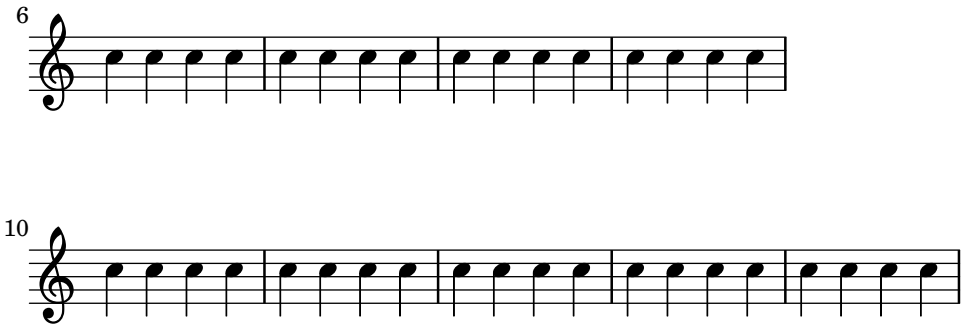
Cette manière de procéder est tout à fait indiquée lorsque vous ajustez les `line-break-system-details` et autres propriétés fort intéressantes de l'objet `NonMusicalPaperColumn`.

```
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
}
```





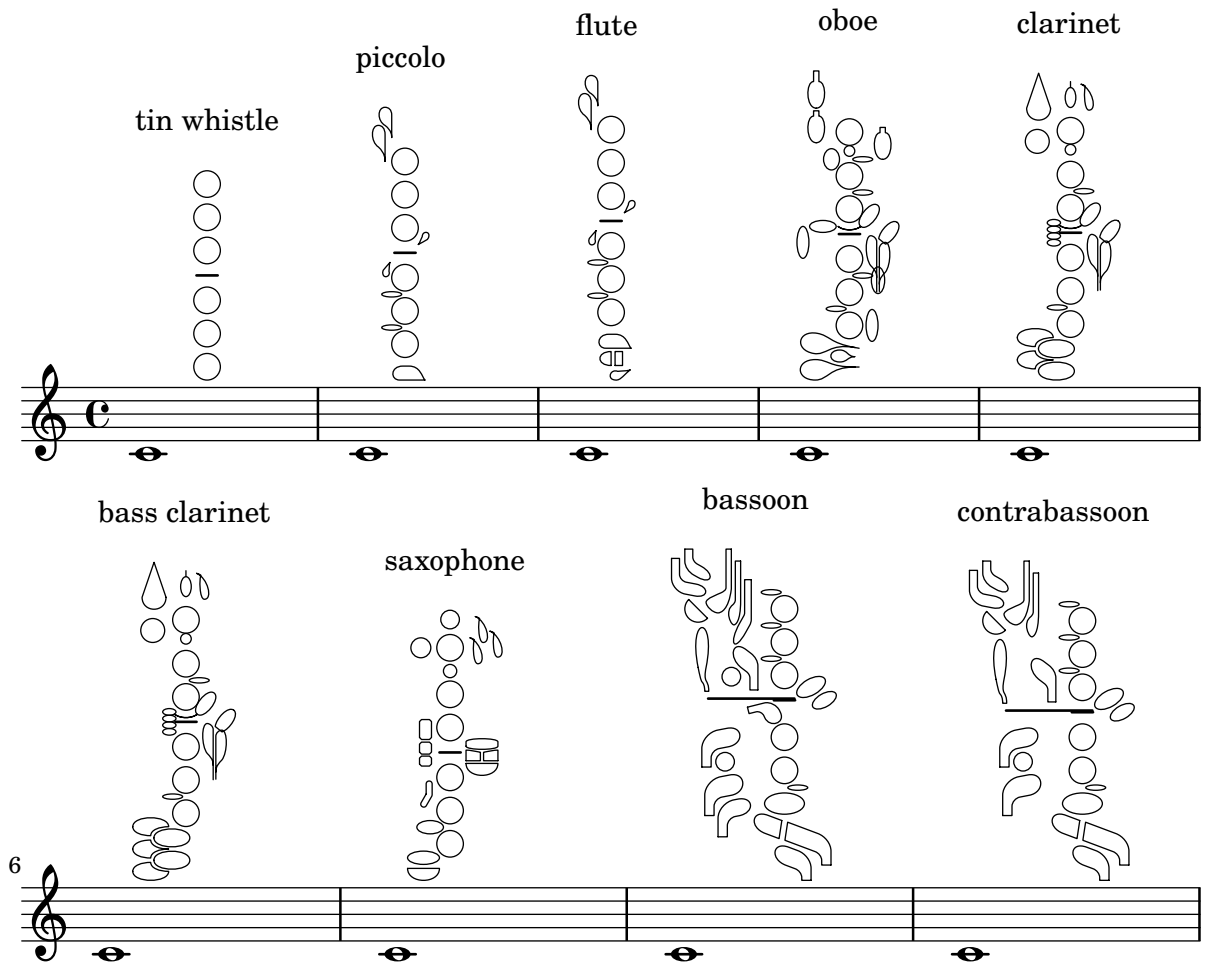
Liste des diagrammes de doigtés pour bois

Voici les différents instruments à vent de la section des bois pour lesquels LilyPond peut, à ce jour, afficher des doigtés.

```
\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
}
```


tin whistle
piccolo
flute
oboe
clarinet
bass clarinet
saxophone
bassoon
contrabassoon
6

35 Symbols and glyphs

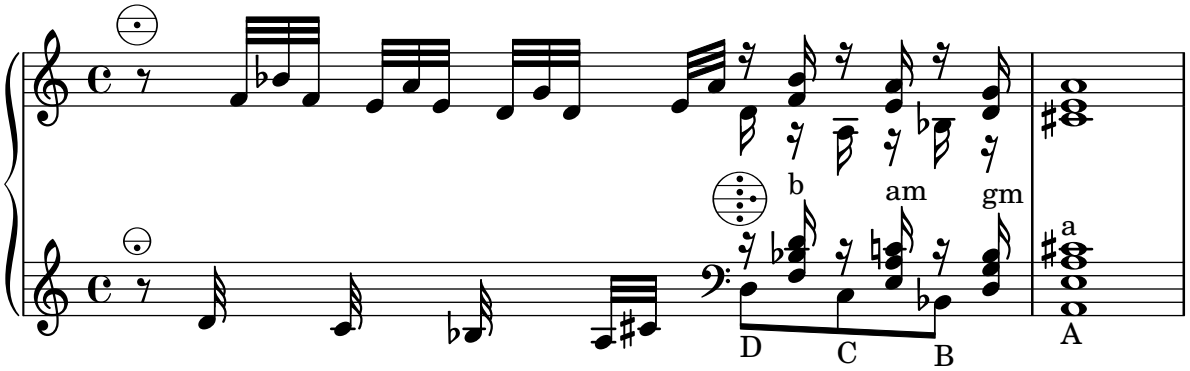
Symboles de registre pour accordéon

Les symboles spécifiques aux registres d'accordéon sont disponible tant sous forme de \markup que d'événements musicaux indépendants – un changement de registre a tendance à intervenir entre des événements musicaux réels. Les registres de basse ne sont pas très standardisés. Les différentes commandes disponibles sont regroupées à l'annexe Section "Registres d'accordéon" dans *Manuel de notation*.

```
#(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" }
    >>
  }
>>
```



Ajout d'indicateurs à une portée dédoublée après un saut de ligne

Dans cet extrait sont définies les commandes \splitStaffBarLine, \convUpStaffBarLine et \convDownStaffBarLine. Ces commandes ajoutent une double flèche après la dernière barre de mesure d'une portée, indiquant par là que ses différentes voix disposeront de leur propre portée à la ligne suivante, ou bien se regrouperont.

Il est à noter que, dans l'implémentations ici proposée, sont dessinées des flêches sans dimension dans la marge droite. Ceci ne devrait pas poser de problème pour une impression normale. Il sera cependant nécessaire d'agrandir horizontalement la boîte englobante, comme indiqué ci-après, afin d'éviter le rognage lors d'une sortie sous forme d'image.

```
#(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 = {
```

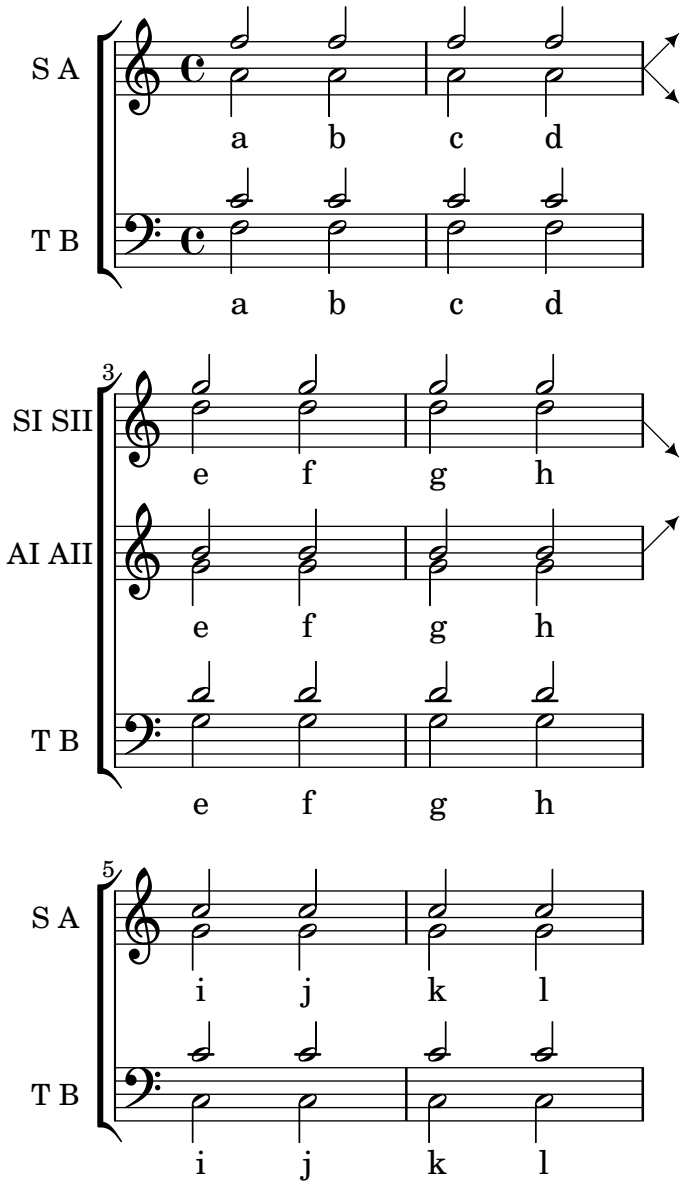
```
  \voiceTwo
  \*4 a'2
  \voiceOne
  \*4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \*4 g'2
}
aII = {
  s1*2
  \voiceTwo
  \*4 g'2
}
ten = {
  \voiceOne
  \*4 c'2
  \*4 d'2
  \*4 c'2
}
bas = {
  \voiceTwo
  \*4 f2
  \*4 g2
  \*4 c2
}

\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
      }
    }
}
```



Gravure de musique ancienne

Voici de nombreux symboles contenus dans la fonte Emmentaler et que LilyPond utilise en matière de musique ancienne.

```
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"

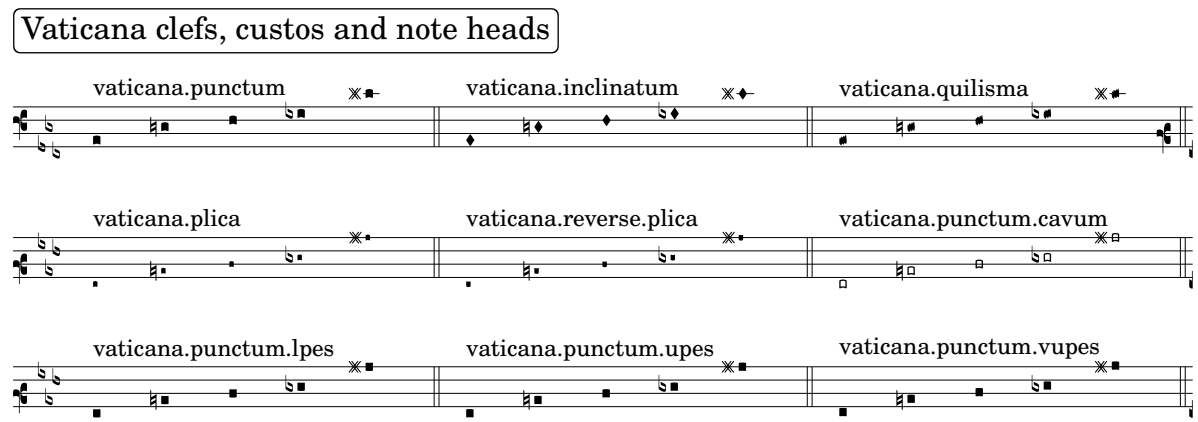
    \break

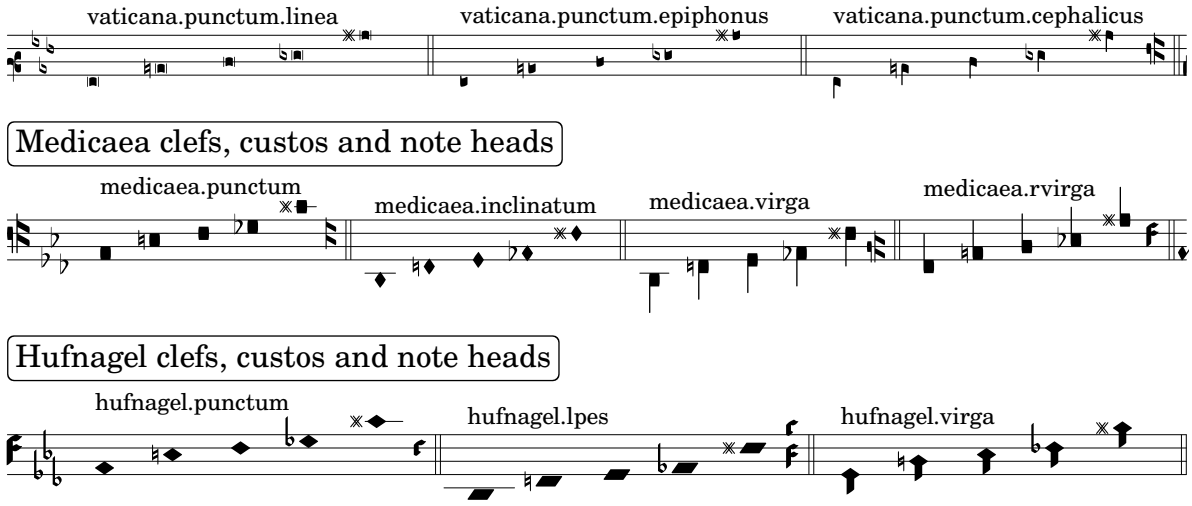
    \override NoteHead.style = #'hufnagel.punctum
    <>^"hufnagel.punctum" \m

    \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
    }
  }
}
```





Signes de respiration

Les indications de respiration sont disponibles sous différentes formes : virgule (par défaut), trait, en V ou en « voie de chemin de fer » (césure).

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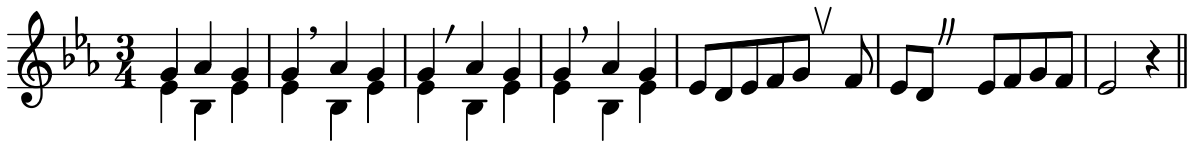
  % rvarcomma and lvarcomma are variations of the default rcomma
  % and lcomma
  % N.B.: must use Staff context here, since we start a Voice below
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % 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 "||"
}
```



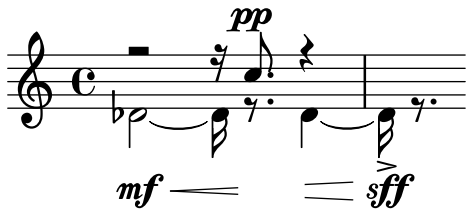
Soufflet de crescendo partiellement interrompu

Une portion d'un soufflet de crescendo peut être rendue invisible. Il suffit pour cela de dessiner un rectangle par dessus ce tronçon, ce qui aura pour effet de le rendre invisible. Ce rectangle est défini en tant que *markup* textuel.

La commande de *markup* `with-dimensions` indique à LilyPond de ne prendre en considération que l'extrémité inférieure du rectangle lors de son positionnement par rapport au soufflet. L'ajustement de la propriété `staff-padding` permet d'éviter au rectangle de venir s'intercaler entre le soufflet et la portée.

Le soufflet doit se trouver à un niveau inférieur à celui du *markup* afin que le dessin du rectangle puisse effectivement le recouvrir.

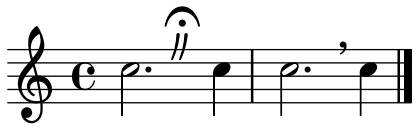
```
\relative c' {
  <<
    {
      \dynamicUp
      r2 r16 c'8.\pp r4
    }
    \\
    {
      \override DynamicLineSpanner.layer = #0
      des,2\mf\< ~
      \override TextScript.layer = #2
      \once\override TextScript.staff-padding = #6
      \once\override TextScript.vertical-skylines = #'()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



Césure en forme de « voie de chemin de fer » avec point d'orgue

Une césure peut parfois s'indiquer par une double respiration surmontée d'un point d'orgue. Le code ci-dessous permet d'obtenir la combinaison répondant à cette esthétique.

```
\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 "|."
}
```



Guidons

Les guidons peuvent adopter différents styles.

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

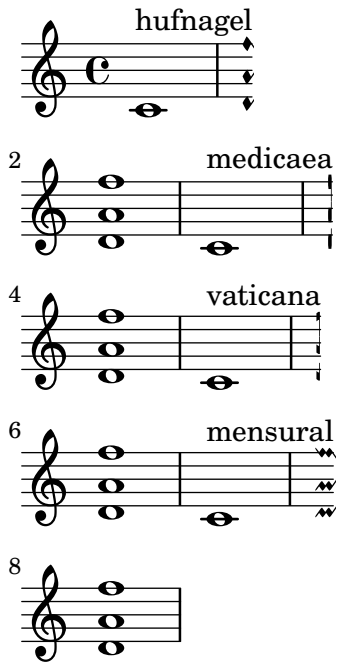
\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

    \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
  }
}
```



Contrôle de la position et du nombre de points d'une barre de reprise

Il est possible de personnaliser la position et le nombre de points d'une barre indiquant une reprise, que ce soit en définissant de nouvelles barres de mesure personnalisées ou bien en redéfinissant la manière dont les indications de reprise par défaut sont dessinées. Ceci s'avère tout particulièrement utile dans le cas de portées aux lignes adaptées comme dans l'exemple ci-dessous.

```
#(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.

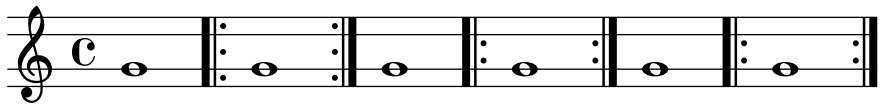
IS-SPAN is not used in this custom function."
   (let* ((staff-space (ly:staff-symbol-staff-space grob))
          (dot (ly:font-get-glyph (ly:grob-default-font grob)
               "dots.dot"))
          (stencil empty-stencil))
     (for-each
      (lambda (dp)
        (set! stencil (ly:stencil-add 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 }
}
```

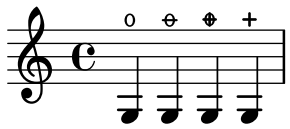


Symboles de doigtés pour instruments à vent

Des symboles spécifiques peuvent être obtenus en combinant les glyphes disponibles, ce qui est tout à fait indiqué en matière d'instrument à vent.

```
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
}
```



Séparation de syllables par une liaison

Des mots monosyllabiques peuvent s'attacher par une liaison à l'aide d'un tilde.

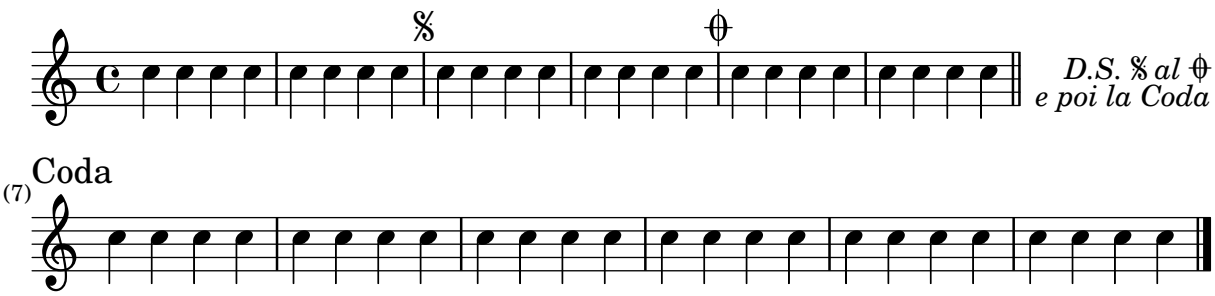
```
\lyrics {
  wa~o~a
}
```

wa‿o‿a

Positionnement des segno et coda (avec saut de ligne)

Le code ci-dessous permet d'adjoindre à un signe *segno* un texte *D.S. al Coda*, là où se trouverait normalement un bout de portée. La *coda* entamera une nouvelle ligne. Une variante, indiquée ici même, permet de laisser la *coda* sur la même ligne.

```
\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
}
```



Styles de silences

Les silences peuvent être gravés selon différents styles.

```
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

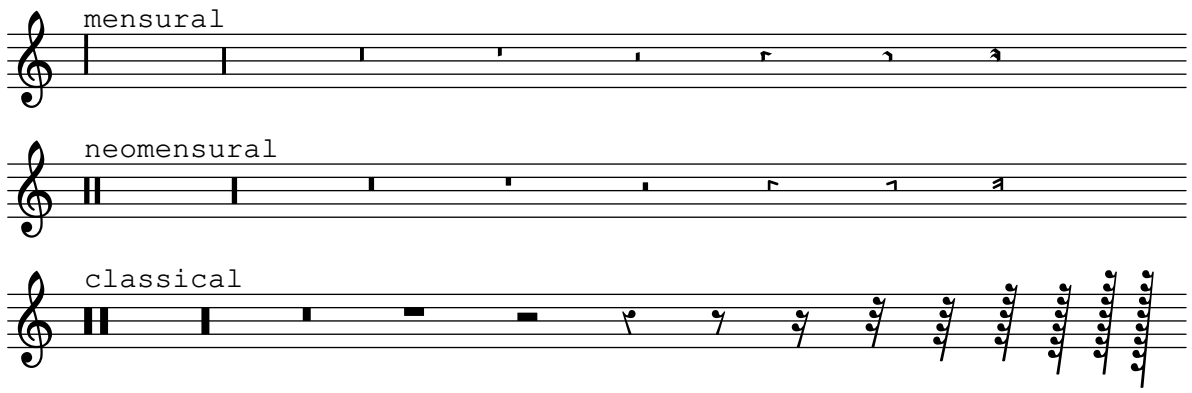
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

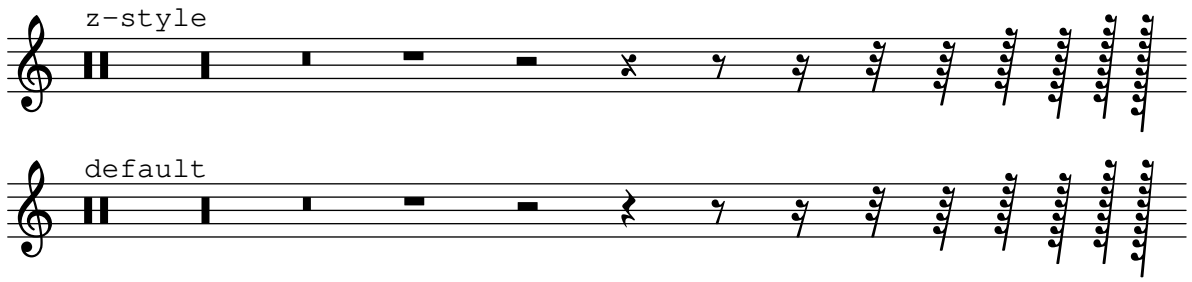
  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \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
}
```
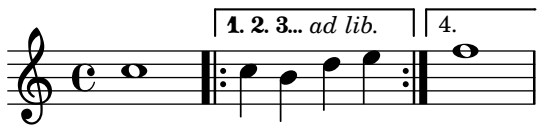




Indication de reprise avec texte grâce à `repeatCommands`

La commande `\repeat volta` permet d'indiquer facilement des reprises. Il est certains cas où l'adjonction d'un texte sous forme de `\markup` nécessite cependant de recourir à la propriété de contexte `repeatCommands`.

Dans la mesure où l'argument de `repeatCommands` est constitué d'une liste, le plus simple est de définir le texte dans une variable qui sera ensuite incorporée dans la liste en respectant la syntaxe Scheme `#`(volta` *`texteIdentificateur`*`) ...)`. Les commandes de début et de fin de reprise peuvent s'ajouter séparément à la liste des éléments.

```
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

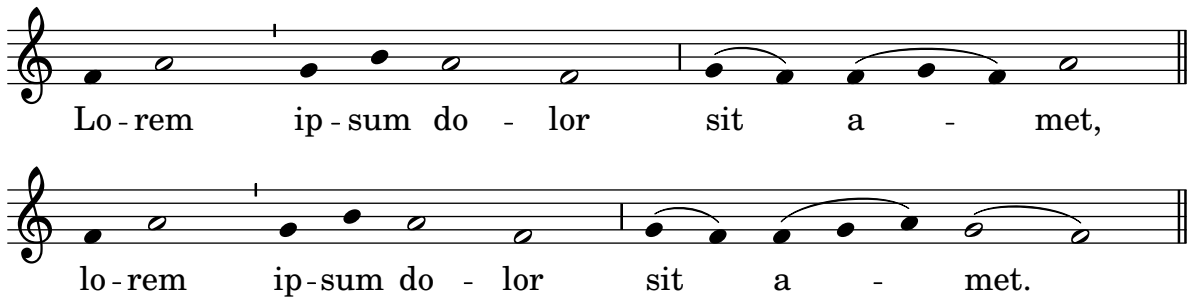
Exemples de notation ancienne – transcription moderne de musique grégorienne

Voici comment vous pourriez transcrire du grégorien. Pour mémoire, il n'y a en grégorien ni découpage en mesure, ni hampe ; seules sont utilisées des têtes de note blanches ou noires, ainsi que des signes spécifiques permettant d'indiquer des silences de différentes durées.

```
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
  >>
}
```



Psalmodie anglicane

Cet exemple illustre la manière de présenter un cantique tel qu'on le trouve dans l'église anglicane. Vous noterez comment sont ajoutés les couplets indépendamment de la musique. Dans le but de vous montrer plusieurs styles, comparez le code des deux couplets.

```
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
```

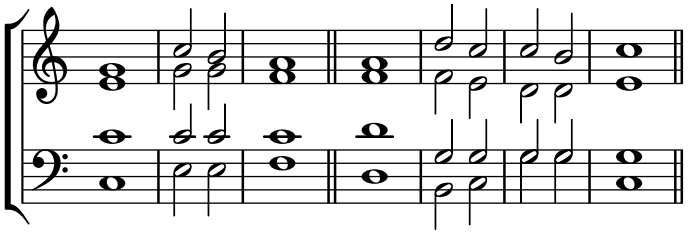
```
        >>
      >>
    >>

    \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 }
            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 in the pro- }
    \line { vocation * and as in the \bold day of tempt- \tick }
    \line { -ation \tick 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.

Modèle pour cantique

Le code ci-dessous illustre la manière d'agencer un cantique liturgique dans lequel chaque ligne débute et se termine par une mesure incomplète. Vous noterez par ailleurs l'affichage des paroles indépendamment de la musique.

```
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

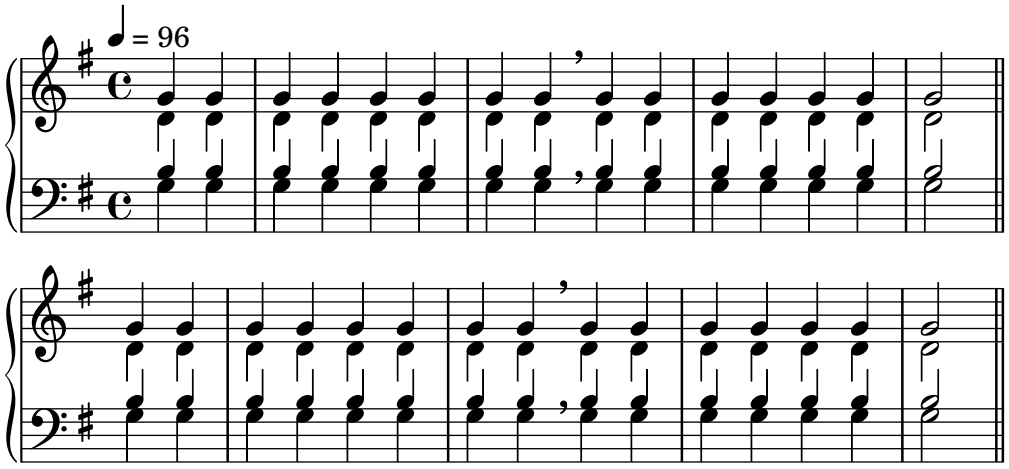
    \new Staff <<  % Start Staff = LH
      \global
      \clef "bass"
      \new Voice = "Tenor" <<  % Start Voice = "Tenor"
        \Timeline
        \voiceOne
        \TenorMusic
      >>  % End Voice = "Tenor"
      \new Voice = "Bass" <<  % Start Voice = "Bass"
        \Timeline
        \voiceTwo
        \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

Symboles de jazz

Bien que compliqué de prime abord, voici un canevas tout à fait indiqué pour les ensembles de jazz. Vous noterez que tous les instruments sont notés en ut (`\key c \major`), la tonalité de concert. Les notes seront automatiquement transposées dès lors qu'elles seront inscrites dans une section `\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
  >>
}

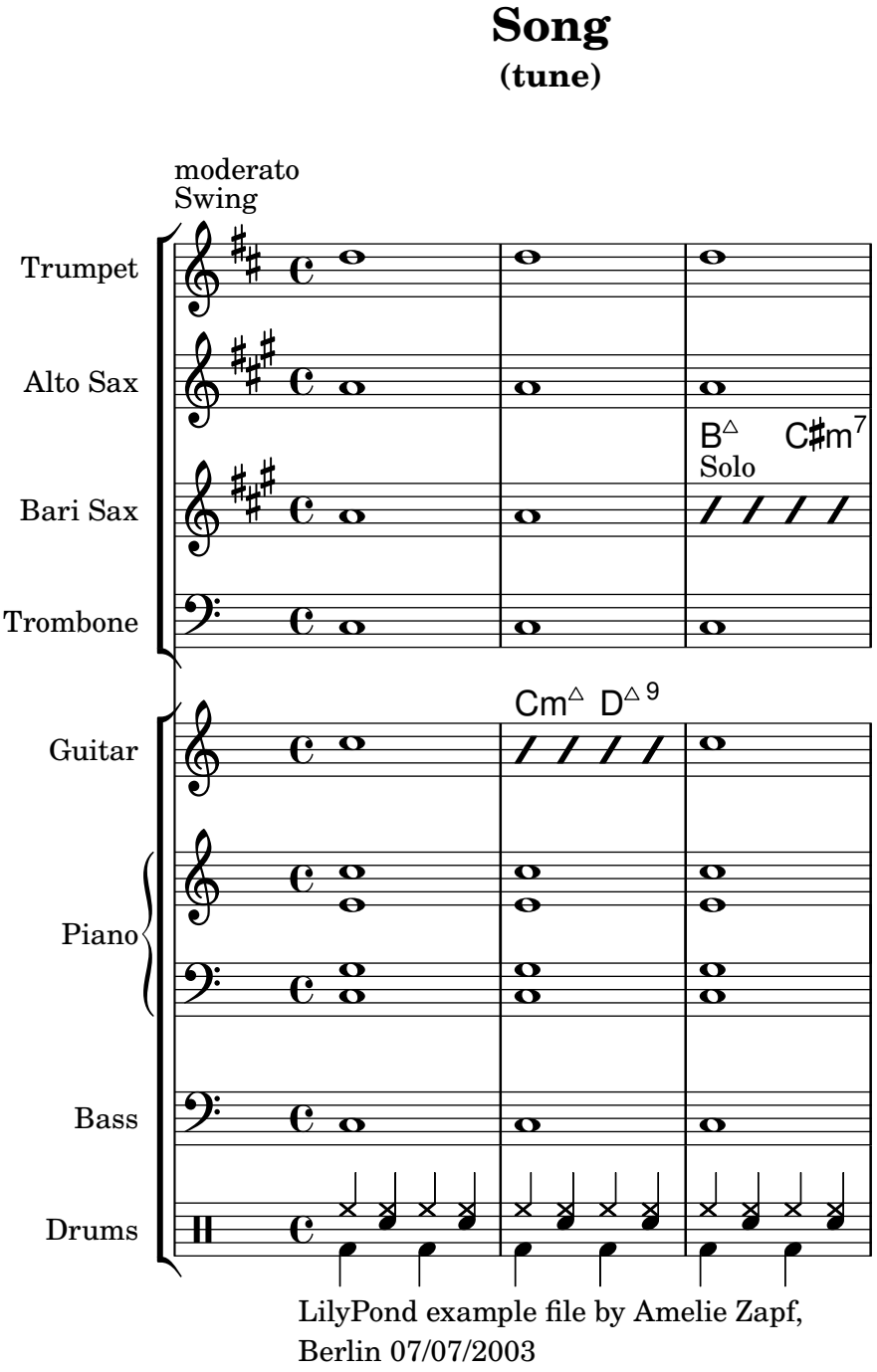
%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%%

\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 { }
  }
}
```



Orchestre, chœur et piano

Ce canevas illustre l'utilisation de contextes `StaffGroup` pour regrouper les instruments selon leur famille, imbriqués dans un `GrandStaff`, ainsi que le recours à la fonction `\transpose` pour les instruments transpositeurs. Dans tous les identificateurs, la musique est stockée en ut. Les notes peuvent tout aussi bien être saisies en ut ou dans la tonalité particulière de l'instrument avant d'être transposées puis affectées à une variable.

```
#(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 }
        >>

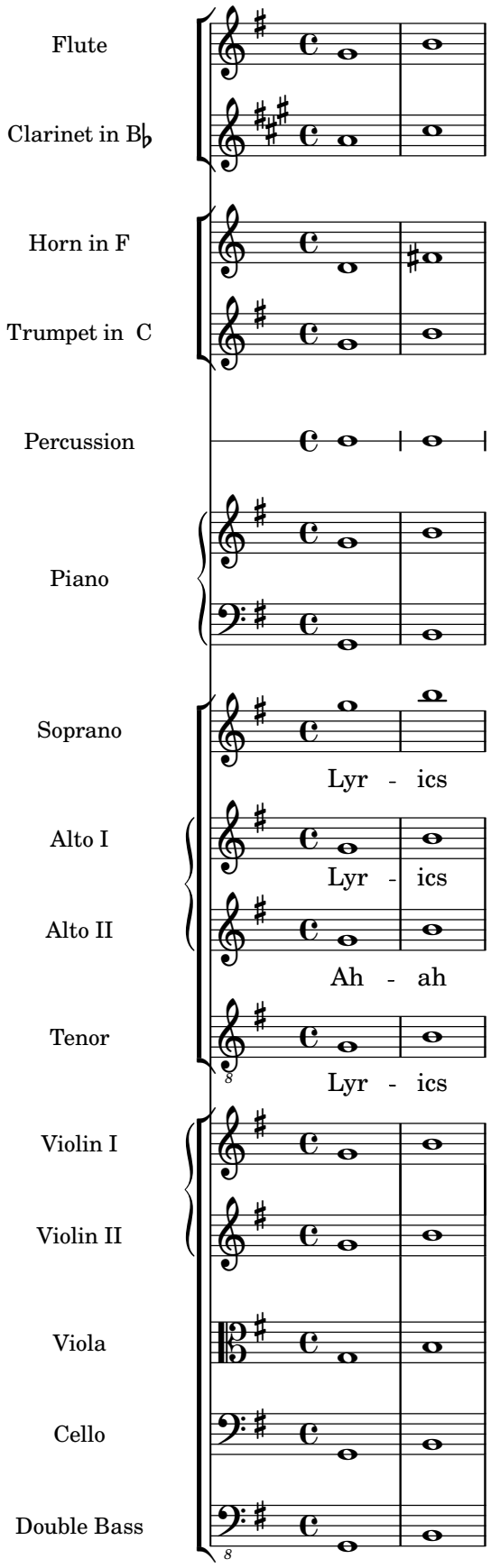
        \new Staff = "Staff_tenor" \with {
          instrumentName = "Tenor"
        }
          \new Voice = "tenor" \tenorMusic
          \new Lyrics \lyricsto "tenor" { \tenorLyrics }
      >>

      \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

Piano – cannevas simple

Voici une simple partition pour piano avec quelques notes.

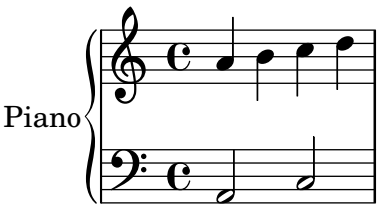
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  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 { }
}
```



Piano et paroles entre les portées

Lorsque la mélodie est doublée au piano, cela ne nécessite pas forcément une portée spécifique. Les paroles peuvent s'insérer entre les deux portées de la partition pour piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

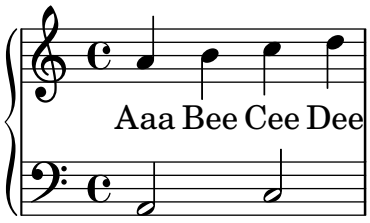
  a4 b c d
}

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 { }
}
```



Piano, mélodie et paroles

Il s'agit du format classique pour le chant : une portée pour la mélodie et les paroles au-dessus de l'accompagnement au piano.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

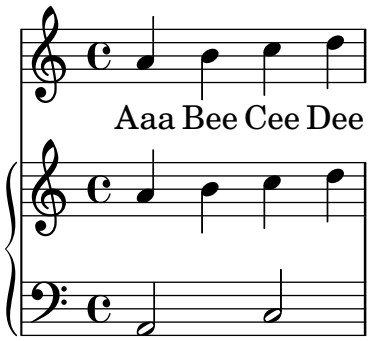
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  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 { }
}
```
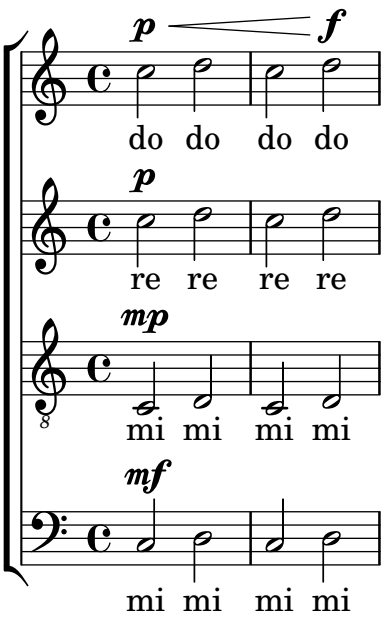


Modèle pour chœur SATB, sur quatre portées

Modèle pour chœur à quatre voix mixtes, chaque pupitre ayant sa propre portée.

```
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
    >>
  >>
}
```



Paroles, musique et accords

Ce canevas comporte tous les éléments d'une chanson : la mélodie, les paroles, les accords.

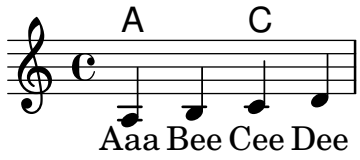
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  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 { }
}
```



Paroles, musique, accords et diagrammes de fret

Ce canevas comporte, en plus de la mélodie, des paroles et des accords, les diagrammes de fret.

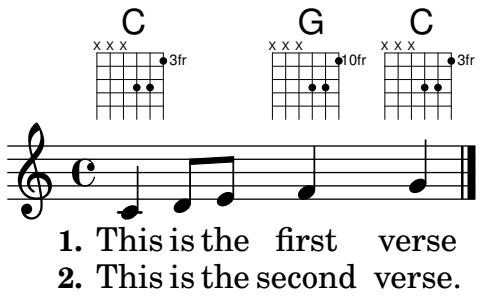
```
verseI = \lyricmode {
  \stanza "1."
  This is the first verse
}

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 { }
}
```



Mélodie simple et accords
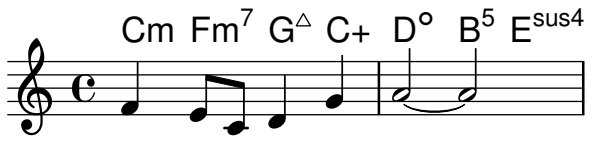
Vous avez besoin de la partition d'une mélodie avec les accords ? N'allez pas plus loin !

```
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 { }
}
```



Portée unique et paroles
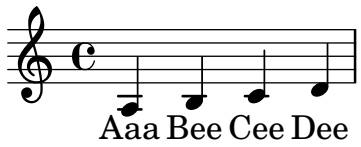
Ce canevas comporte une simple ligne mélodique agrémentée de paroles. Recopiez-le, ajoutez-y d'autres notes et paroles. Les ligatures automatiques sont ici désactivées comme il est d'usage en matière de musique vocale. Pour activer la fonction de ligature automatique, modifiez ou commentez la ligne en question.

```
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 { }
}
```
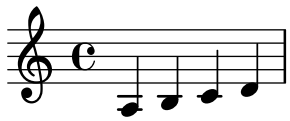


Portée unique avec quelques notes

Cet exemple simpliste se compose d'une portée agrémentée de quelques notes. Il convient tout à fait pour un instrument seul ou un fragment mélodique. Recopiez-le dans un nouveau fichier, ajoutez-y d'autres notes et c'est prêt !

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

\score {
  \new Staff \melody
  \layout { }
  \midi { }
}
```



Quatuor à cordes (conducteur)

Voici un canevas pour quatuor à cordes. Notez l'utilisation de la variable `\global` pour traiter la métrique et la tonalité.

Voir aussi `Quatuor à cordes, avec parties séparées`.

```
global= {
  \time 4/4
  \key c \major
}

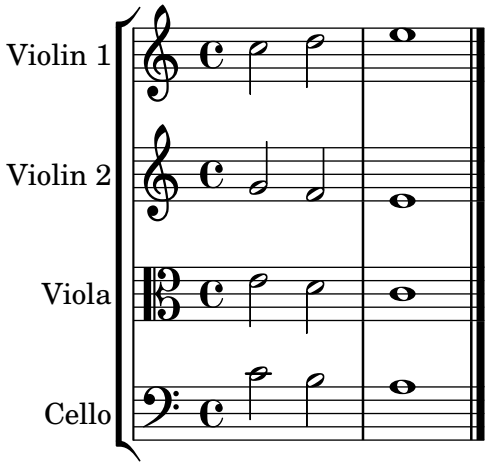
violinOne = \new Voice \relative c'' {
  c2 d
  e1
  \bar "|."
}

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 { }
}
```



Quatuor à cordes, avec parties séparées

Grâce à ce canevas, vous pouvez obtenir une partition d'excellente facture pour quatuor à cordes mais aussi, si le besoin s'en faisait sentir, une partie séparée par instrument. Par ailleurs, cet exemple illustre l'utilisation de la fonction `\tag` dans le but d'extraire des parties séparées.

Pour des raisons techniques, un même fichier source ne peut générer ici de multiples fichiers. Il vous faudra donc découper ce canevas en plusieurs fichiers séparés ; leur nom respectif est indiqué en commentaire.

`piece.ly` comporte tout ce qui a trait à la musique. Les autres fichiers – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` et `vlc.ly` – vous permettront d'obtenir un conducteur et les parties selon le pupitre.

N'oubliez pas de supprimer les commentaires superflus des fichiers individualisés !

```
% 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 { }
}

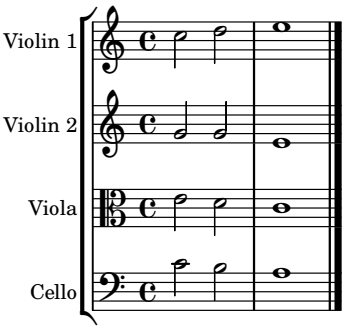
% vla.ly
% (This is the Viola part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}

% vlc.ly
% (This is the Cello part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vlc \music
  \layout { }
}

%}
```



Ensemble vocal (simple)

Ce fichier constitue un canevas standard de partition pour chœur à quatre voix mixtes. Lorsque les ensembles s'étoffent, il est judicieux de recourir à une section spécifique incluse dans chacune des parties, tout particulièrement pour gérer la métrique et la tonalité qui, la plupart du temps, sont communes à tous les pupitres. Comme il est d'usage pour les hymnes, les quatre voix sont réparties sur deux portées.

```
\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
}
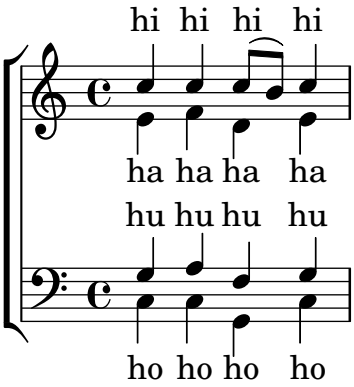
altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

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
  >>
}
```



Ensemble vocal avec réduction pour piano

Ce canevas ajoute une réduction pour piano à une partition standard pour chœur à quatre voix mixtes. Ceci illustre l'un des avantages de LilyPond : une expression musicale peut être réutilisée sans effort. Toute modification apportée à l'une des voix, mettons `tenorMusique`, est automatiquement reportée dans la réduction pour 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
}

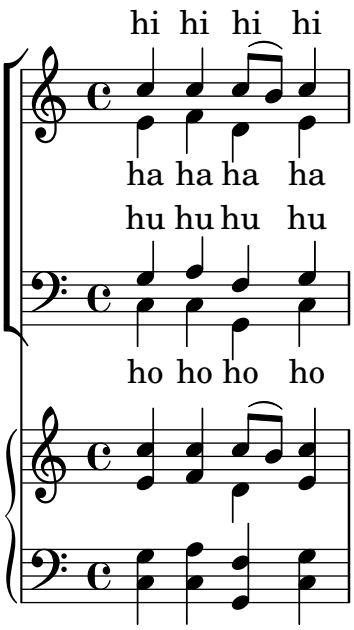
\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \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 >>
      >>
    >>
  >>
}
```



Ensemble vocal avec alignement des paroles selon le contexte

Ce canevas ressemble beaucoup à celui pour chœur à quatre voix mixtes. La différence réside dans le fait que les paroles sont positionnées en ayant recours à `alignAboveContext` et `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
}
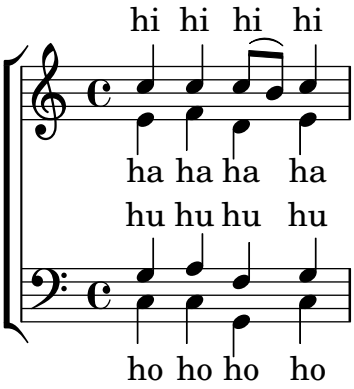
tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

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
  >>
```

```
}
```



Ensemble vocal avec couplet et refrain

Ce canevas illustre la manière d'agencer une œuvre vocale où le couplet est chanté en solo et le refrain à deux voix. Vous noterez le recours aux silences invisibles dans la variable `\global` : ils permettent de positionner les changements de métrique et autres éléments communs à toutes les parties, ce pour l'intégralité du morceau.

```
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 |
}

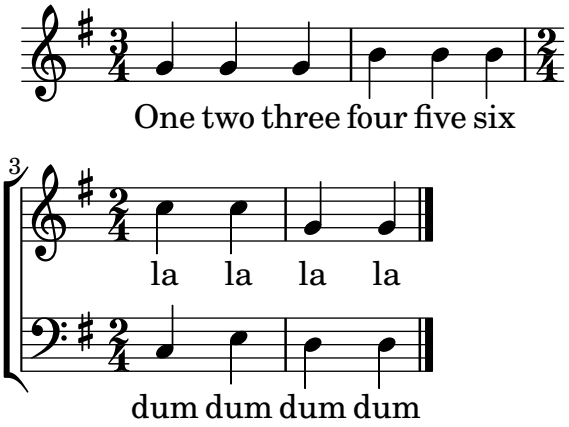
BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

\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 Section “Titles and headers” dans *Manuel de notation*.

Ajout de la date du jour à une partition

Avec un peu de code Scheme, voici comment ajouter facilement la date du jour à votre partition.

```
\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
}
```



Alignement des noms d’instrument

L’alignement horizontal des noms d’instrument se gère à l’aide de la propriété `self-alignment-X` de l’objet `InstrumentName` (habitullement attaché à un contexte `Staff`). Les variables `indent` et `short-indent`, attachées au bloc `\layout`, déterminent l’espace alloué à l’alignement des noms d’instrument, respectivement dans leurs formes développée et abrégée.

```
\paper {
  left-margin = 3\cm
}

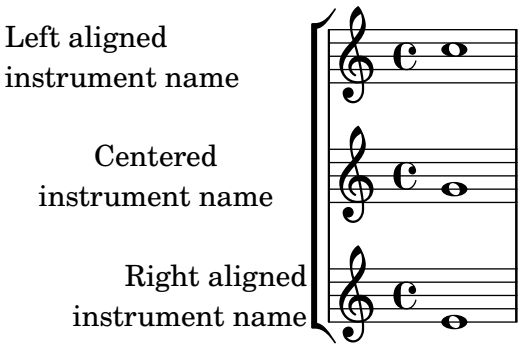
\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                            "instrument name" }
    shortInstrumentName = "Left"
```

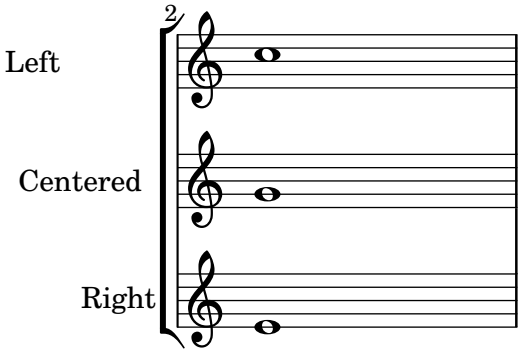
```
  } {
    c''1 \break c''1
  }

  \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
}
```





Champs d'entête

Voici la liste de tous les champs d'entête définis par défaut dans LilyPond. Grâce à l'activation de `print-all-headers`, beaucoup plus de champs s'affichent, indiquant la hiérachie des blocs `\header`.

```
\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
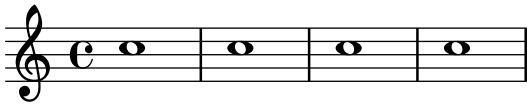
localtitle

localsubtitle

localpoet **localinstrument** localcomposer

localmeter localarranger

localpiece localopus



copyright

tagline

Impression du numéro de version

Il est possible déjouter le numéro de la version de LilyPond utilisée dans un *markup*.

```
\markup { Processed with LilyPond version #(lilypond-version) }
```

Processed with LilyPond version 2.27.4

38 Tweaks and overrides

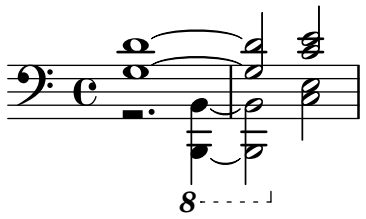
See also Section “Changing defaults” dans *Manuel de notation* and Section “Tweaking output” dans *Manuel d'initiation*.

Ajout d'une indication d'octave pour une seule voix

Lorsque plusieurs voix cohabitent sur une même portée, déterminer l'octaviation d'une voix affecte la position des notes de toutes les voix, jusqu'à la fin du crochet d'octaviation. Si l'octaviation ne doit s'appliquer qu'à une seule voix, le `Ottava_spanner_engraver` devrait être déplacé dans le contexte `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
    }
  >>
}
```



Ajout de liens à des objets

La fonction `add-link`, telle que définie ci-dessous, permet d'ajouter un lien au stencil d'un objet graphique. Elle s'emploie au sein d'un `\override` ou d'un `\tweak`.

À noter que le fonctionnemment du `point-and-click` est perturbé sur les objets ainsi liés.

Cette fonction n'est opérationnelle que pour une sortie PDF.

La coloration des objets liés s'obtient par une commande séparée.

```
#(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)
```
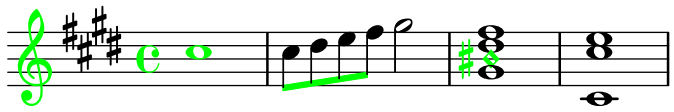
```
\relative c' {
   \key cis \minor

   \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>
}
```



Ajout de *markups* à une tablature

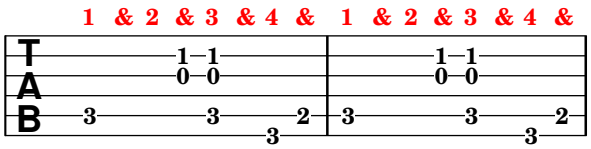
Par défaut, les *markups* n'apparaissent pas dans une tablature.

Il suffit, pour les voir apparaître, de rétablir la propriété `stencil` de l'objet `TextScript` dans le contexte `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
    }
  }
}
```



Ajout de marques temporelles à un long glissando
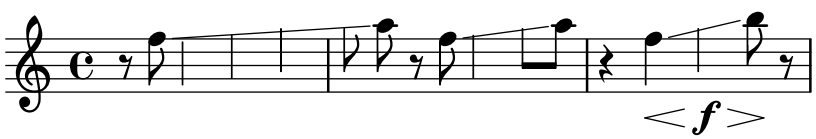
Lorsqu'un glissando s'étend dans la durée, on trouve parfois des indications temporelles, matérialisées par des hampes sans tête de note. De telles hampes permettent aussi d'indiquer des éléments intermédiaires.

L'alignement des hampes avec la ligne de glissando peut requérir quelques aménagements.

```
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 |
}
```
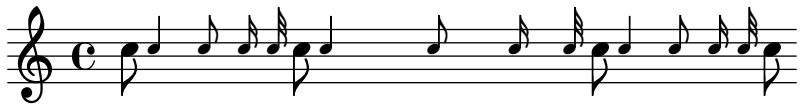


Ajustement de l'espacement des notes d'ornement

La propriété `spacing-increment` de `Score.GraceSpacing` permet d'ajuster l'espacement d'un groupe de notes d'ornement.

```
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
}
```



Ajustement du positionnement vertical d'une liaison

Il est possible, à l'aide d'un `\override Slur.positions`, de définir le positionnement vertical des extrémités d'une liaison à des valeurs absolues (ou, plutôt, forcer l'algoritme de LilyPond à considérer ces valeurs comme étant préférables). Dans de nombreux cas, ceci demande de procéder à tâtons pour obtenir une valeur acceptable, d'autant que la commande `\offset` est inefficace pour les liaisons et affiche un avertissement.
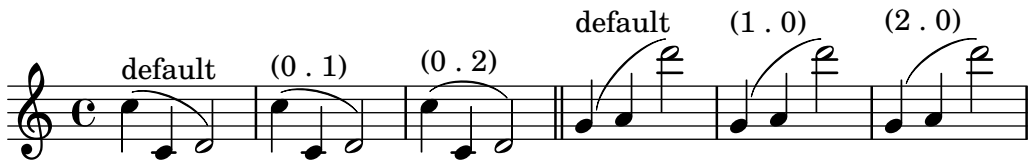
Le code ci-dessous parmet d'ajuster le positionnement du début et de la fin d'une liaison en spécifiant des modifications *relatives* comme le ferait un `\offset`.

Syntaxe : `\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)
}
```
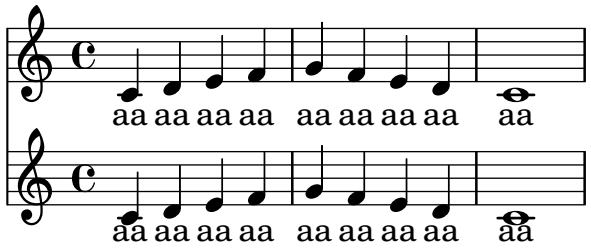


Ajustement de l'espacement vertical des paroles

Le code ci-dessous indique comment rapprocher de la portée à laquelle elle est affectée, une ligne de paroles.

```
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
>>
```



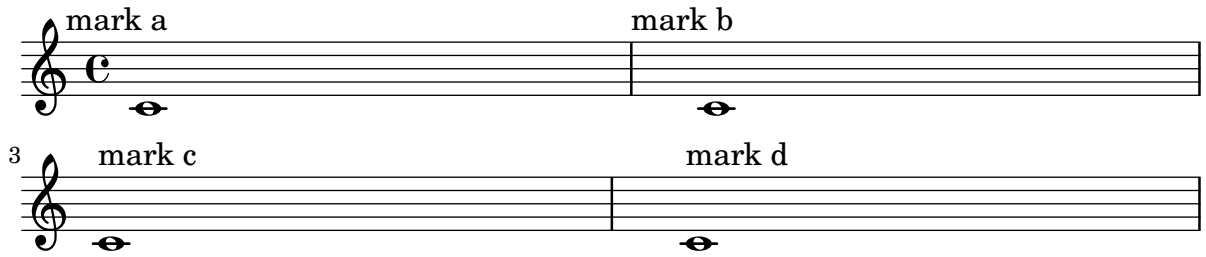
Alignement de repère sur une note

Les objets `TextMark` s'alignent par défaut sur des objets graphiques `NonMusicalPaperColumn` tels que le coin gauche de la portée ou une barre de mesure. Ils peuvent aussi venir s'aligner dès lors qu'aura été désactivée la propriété `non-musical`.

```
\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 |
```

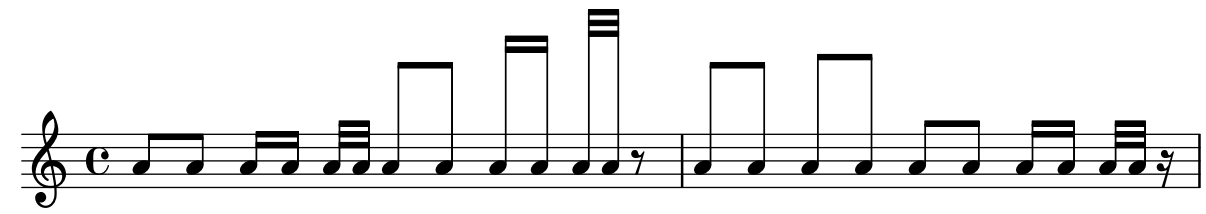
```
}
```



Modification de la hauteur de hampes ligaturées

La hauteur de hampe des notes ligaturées est gérée par la sous-propriété `beamed-lengths` des `details` de l'objet `Stem`. Lorsqu'elle ne comporte qu'une seule valeur, cette hauteur s'appliquera à toutes les hampes. En présence de plusieurs arguments, le premier affectera les croches, le second les doubles croches, et ainsi de suite. Le dernier argument s'appliquera aussi aux notes de plus courte durée. Les arguments peuvent être des valeurs décimales.

```
\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 |
}
```



Numérotation des mesures et alternatives

Un réglage de la propriété de contexte `alternativeNumberingStyle` permet d'accéder à deux méthodes alternatives de gestion de la numérotation des mesures en cas de reprises.

```
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 "|."
}

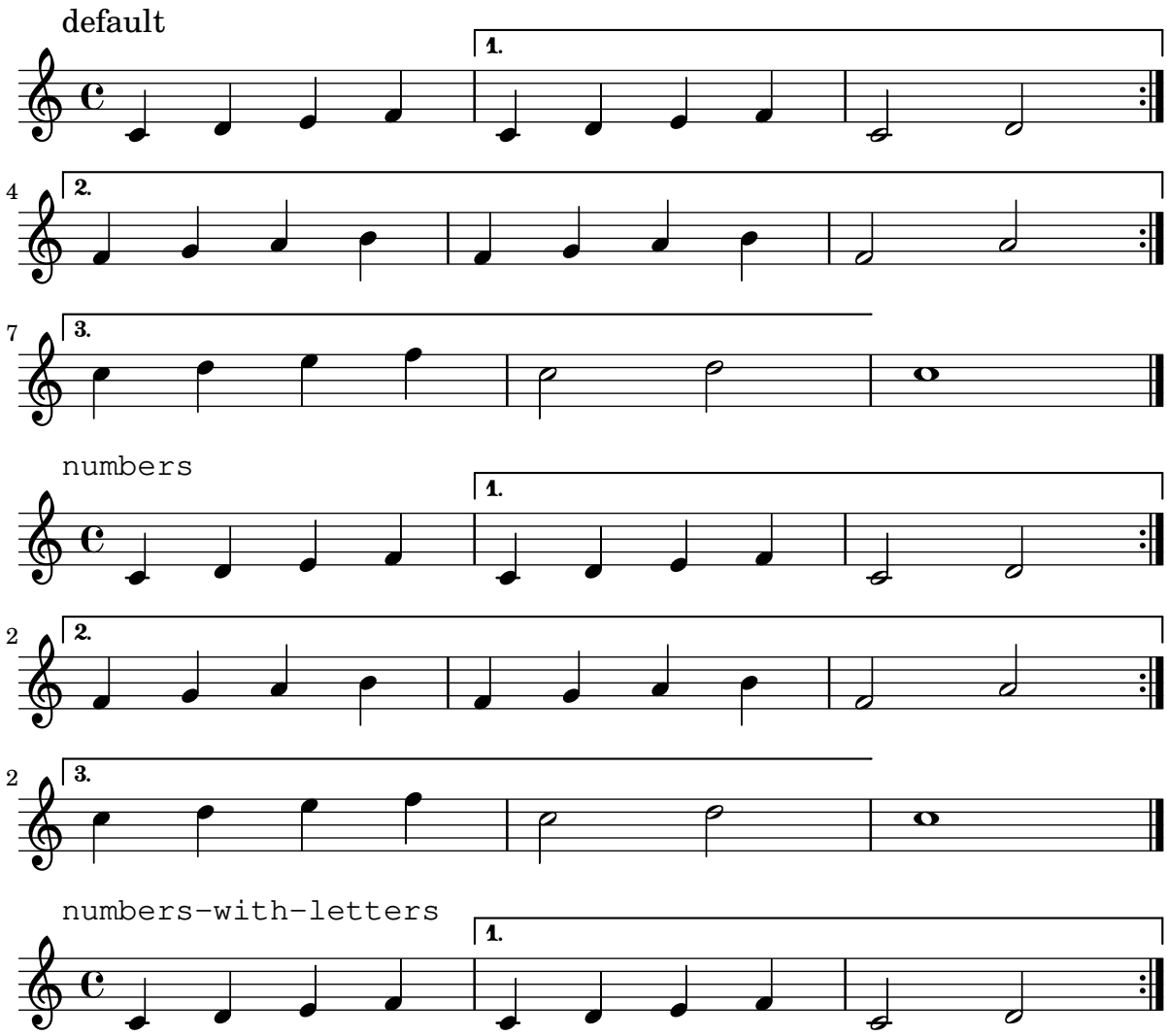
{
  \textMark \markup \large "default"
```
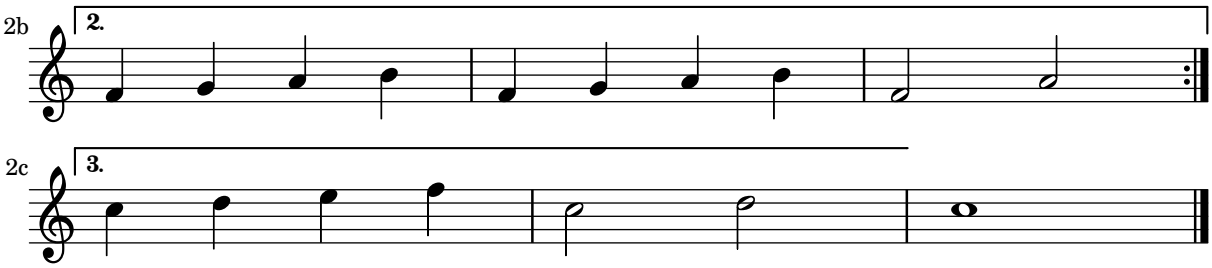
```
  \music
}

{
  \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
  }
}
```





Crochets d'analyse au-dessus de la portée

Les crochets d'analyse viennent par défaut se positionner au-dessous de la portée. L'exemple suivant vous indique comment les faire apparaître en surplomb de la portée.

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

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```



Crochet d'analyse avec texte

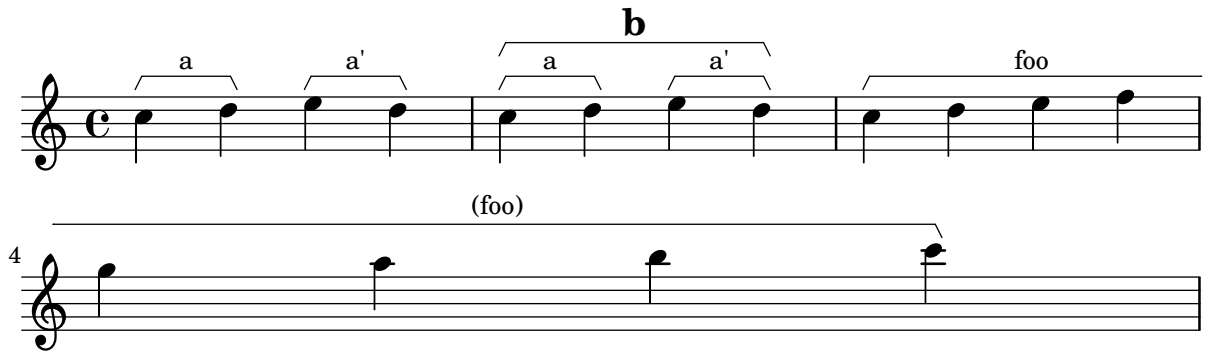
Un *markup* textuel peut venir s'ajouter aux crochets d'analyse grâce à la propriété `text` de l'objet graphique `HorizontalBracketText`. Plusieurs crochets présents en un même moment requièrent d'utiliser la commande `\tweak`.

Le texte ajouté sera répété, entre parenthèse, après un saut de ligne. L'ordre vertical des crochets imbriqués se gère à l'aide de la propriété `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
}
```

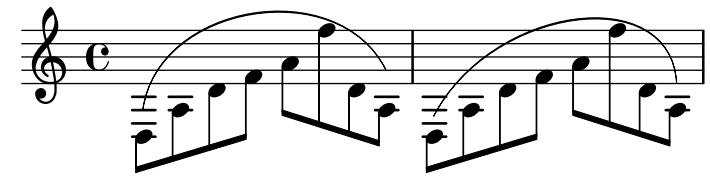


Liaison asymétrique

Une liaison peut adopter une courbe asymétrique afin de s'adapter au mieux à la ligne mélodique.

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

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



Rupture d'alignement horizontal des nuances graphiques et textuelles

Par défaut, LilyPond utilise des objets `DynamicLineSpanner` pour aligner horizontalement une succession d'objets nuance tels que soufflets et indications textuelles, et quel que soit leur positionnement par rapport à la portée. Ceci peut, parfois, ne pas être désirable et une rupture de l'étendue de l'alignement n'est pas possible comme en matière d'alignement vertical (voir l'exemple « Rupture d'alignement vertical des nuances graphiques et textuelles »).

Deux solutions permettent de pallier ce désagrément.

- Modifier la propriété `shorten-pair` de l'objet `Hairpin` pour compenser le décalage qui a été appliqué au soufflet.
- Placer les deux objets de nuance dans des voix différentes.

Le code ci-dessous illustre ces deux possibilités.

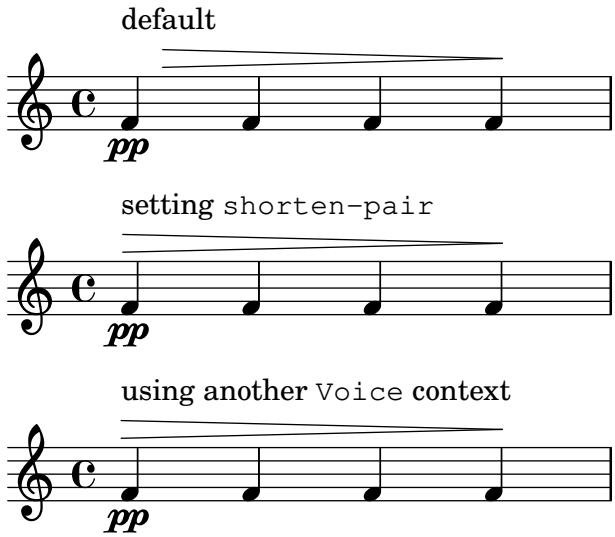
```
{
  <>^"default"
  f'_\pp ^\> f' f' f'\!
}

{
  <>^\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
  }
}
```



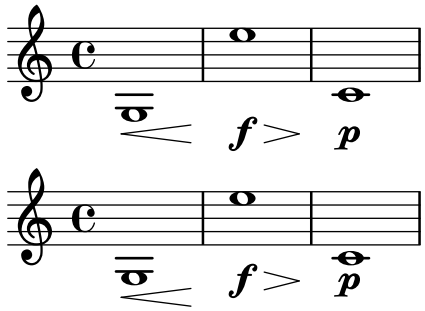
Rupture d'alignement vertical des nuances graphiques et textuelles

Par défaut, LilyPond utilise des objets `DynamicLineSpanner` pour aligner verticalement une succession d'objets nuance tels que soufflets et indications textuelles. Ceci peut, parfois, ne pas être désirable. L'insertion d'un `\breakDynamicSpan`, qui rompt prématurément l'étendue de l'alignement, permet de pallier ce désagrément.

Voir aussi l'exemple « Rupture d'alignement horizontal des nuances graphiques et textuelles ».

```
{ g1\< |
  e''\f\> |
  c'\p }
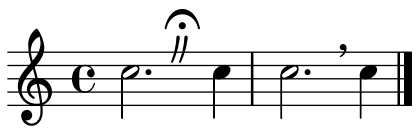
{ g1\< |
  e''\breakDynamicSpan\f\> |
  c'\p }
```



Césure en forme de « voie de chemin de fer » avec point d'orgue

Une césure peut parfois s'indiquer par une double respiration surmontée d'un point d'orgue. Le code ci-dessous permet d'obtenir la combinaison répondant à cette esthétique.

```
\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 "|."
}
```

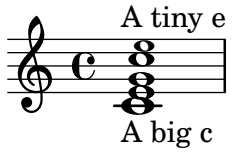


Modification de la taille d'une note particulière d'un accord

La commande \tweak permet de modifier la propriété font-size d'une note particulière d'un accord.

Il s'agit de placer, à l'intérieur même de l'accord (dans la construction < >) et avant la note considérée, l'instruction \tweak suivie de font-size et de définir la taille voulue – comme #-2 pour une petite tête.
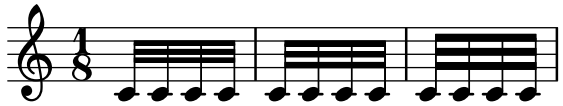
```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Modification de l'épaisseur et de l'écartement des ligatures

L'épaisseur des ligatures se gère au niveau de la propriété `beam-thickness` de l'objet graphique `Beam`. L'ajustement de l'espace entre les ligatures est géré par la propriété `length-fraction`.

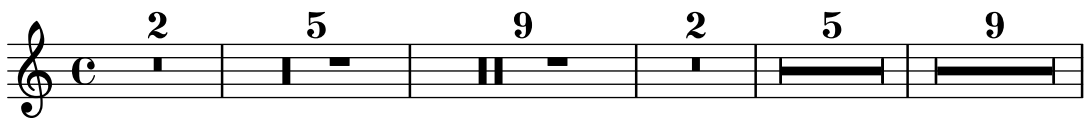
```
\relative f' {
  \time 1/8
  \override Beam.beam-thickness = #0.4
  \override Beam.length-fraction = #0.8
  c32 c c c
  \revert Beam.beam-thickness  % 0.48 is default thickness
  \revert Beam.length-fraction  % 1.0 is default spacing
  c32 c c c
  \override Beam.beam-thickness = #0.6
  \override Beam.length-fraction = #1.3
  c32 c c c
}
```



Modifier l'apparence d'un silence multimesure

Dans le cas où ce silence dure moins de dix mesures, LilyPond imprime sur la portée des « silences d'église » – *Kirchenpause* en allemand – et qui sont une simple suite de rectangles. Par défaut fixée à 10, la propriété `expand-limit` peut se régler afin d'obtenir un silence unique.

```
\relative c'' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = 3
    R1*2 | R1*5 | R1*9
  }
}
```



Modification des propriétés d'objets particuliers
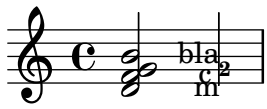
La commande `\applyOutput` permet de personnaliser n'importe quel objet de rendu. Elle requiert une fonction Scheme à trois arguments.

Dans l'exemple suivant, la fonction mc-squared est exécutée pour tous les objets NoteHead (au sein du contexte Voice en cours) à ce moment précis ; la fonction modifie le stencil de l'objet graphique à l'aide de la propriété staff-position pour remplacer certaines hauteurs par des *markups*.

De plus amples informations sont disponibles à la rubrique Section "Application d'une fonction à tous les objets de mise en forme" dans *Extension de LilyPond*.

```
#(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
}
```
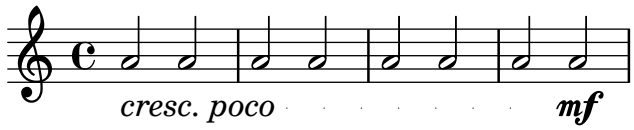


Modification du texte et de l'extension de nuances textuelles

Le texte par défaut des crescendos et decrescendos se change en modifiant les propriétés de contexte crescendoText et decrescendoText.

L'aspect de la ligne d'extension est fonction de la propriété style du DynamicTextSpanner. Sa valeur par défaut est dashed-line, mais d'autres valeurs sont disponibles, comme line, dotted-line et none.

```
\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
}
```



Changement des fontes textuelles par défaut

Les familles de fontes par défaut pour le texte peuvent être changées.

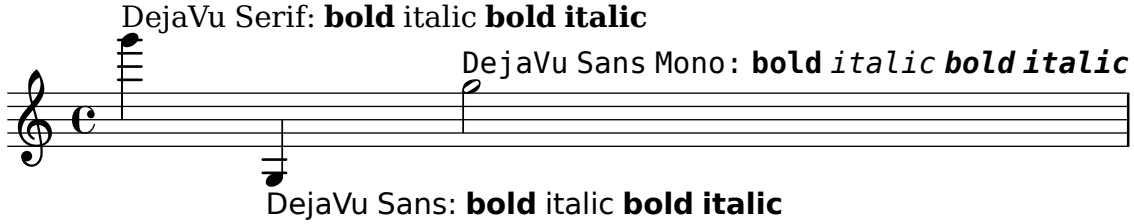
```
%{
You may have to install additional fonts.

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 }
    }
  }
}
```



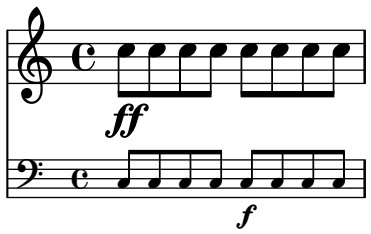
Modification de la taille d'une portée

Bien que le meilleur moyen de définir la taille des portées consiste à utiliser

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

Une portée en particulier peut se redimensionner à l'aide des propriétés `staff-space` et `fontSize`.
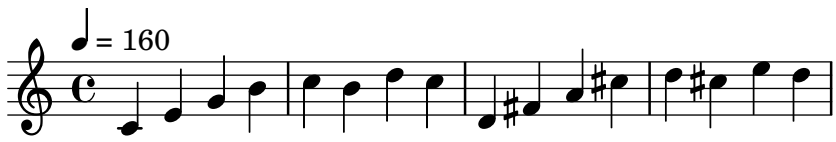
```
<<
  \new Staff \relative c'' {
    \dynamicDown c8\ff c c c c c c c
  }
  \new Staff \with {
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
  } \relative c {
    \clef bass c8 c c c c\f c c c
  }
>>
```



Changement de tempo sans indication sur la partition

Vous pouvez indiquer un changement de tempo pour le fichier MIDI sans pour autant l'imprimer. Il suffit alors de le rendre invisible aux musiciens.

```
\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 { }
}
```
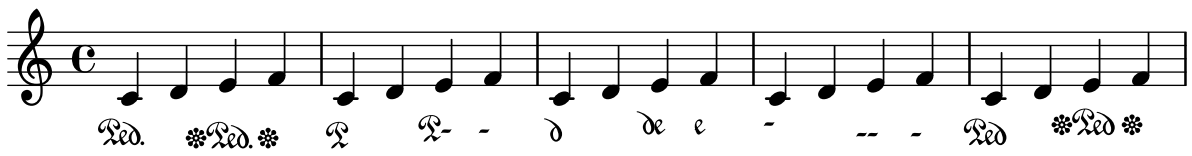


Modification du texte des indications de pédale

La propriété de contexte `Staff.pedalSustainStrings` permet de définir le texte affiché pour les instructions de pédale Les seuls caractères autorisés sont les glyphes particuliers de pédale ici présentés.

```
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
}
```



Contrôle de la visibilité des extensions d'objet après saut de ligne

La visibilité des extensions qui se terminent sur la première note après un saut de ligne est contrôlée par un appel de `after-line-breaking` à la fonction `ly:spanner::kill-zero-spanned-time`.

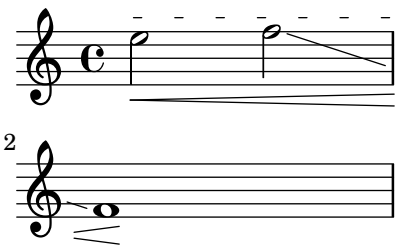
Pour des objets tels qu'un glissando ou un soufflet, le comportement par défaut est de ne pas être reportés après un saut de ligne. L'extension sera donc reprise dès lors que l'appel aura été désactivé.

Il en va inversement pour les extensions qui, par défaut pour les textes notamment, sont reportées après un saut de ligne ; il faudra alors activer l'appel pour empêcher leur report.

```
\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
}
```



Contrôle de l'apparence des traits de trémolo

Différentes propriétés de l'objet graphique `StemTremolo` permettent de gérer l'apparence des traits de trémolo.

- La propriété `slope` définit la pente des traits.
- La propriété `shape` détermine si les traits doivent ressembler à des rectangles (valeur `rectangle`) ou bien à des petites ligatures (valeur `beam-like`).
- La propriété `style` gère à la fois la pente et la forme selon que la note est affublée d'un crochet, est ligaturée ou n'a qu'une simple hampe. Ceci diffère des deux autres propriétés qui, elles, modifient l'apparence de façon inconditionnelle. Deux styles sont définis.
  - `default` : les traits en présence de hampe descendante et de crochet sont plus longs et plus pentus que pour des hampes avec crochet ascendantes ; les traits sur des notes ligaturées prennent l'apparence de rectangles et sont parallèles à la ligature.
  - `constant` : tous les traits ont l'apparence de petites ligatures et ont la même pente, hormis lorsque sur des hampes descendantes crochetées.

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

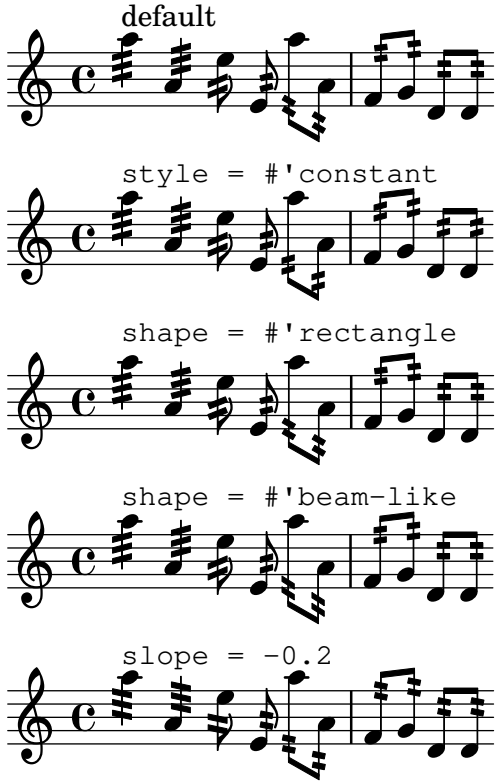
\new Staff {
  <>^\markup "default"
  \music
}

\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
}
```



Contrôle de l'ordre vertical des articulations et ornements

Les symboles s'ordonnent verticalement suivant la propriété `script-priority`. Plus sa valeur numérique est faible, plus le symbole sera proche de la note. Dans l'exemple suivant, l'objet `TextScript` – le dièse – a d'abord la propriété la plus basse et se voit donc placé au plus près de la note ; ensuite, c'est l'objet `Script` – le mordant – qui a la propriété la plus basse, et se place alors sous le dièse. Lorsque deux objets ont la même priorité, c'est l'ordre dans lequel ils sont indiqués qui détermine lequel sera placé en premier.

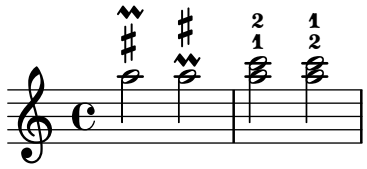
Il est à noter que pour les objets `Fingering`, `StringNumber` et `StrokeFinger`, lorsqu'ils apparaissent dans un accord, leur ordonnancement vertical est aussi déterminé par le positionnement vertical de leur tête de note d'attachement qui sera ajoutée (ou soustraite selon la direction) de la valeur de `script-priority` du *grob*. Ceci permet de s'assurer que, pour des doigtés au-dessous d'un accord, la note la plus basse soit associée au doigté le plus bas, et inversement. L'ordre dans lequel les hauteurs sont saisies dans l'accord est ici sans importance.

Par défaut, les scripts les moins techniques sont inscrits au plus près de la tête de note. L'ordre de base est articulation, flageolet, doigté, doigté main droite, numéro de corde, point d'orgue, archet et script textuel.

```
\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
}
```



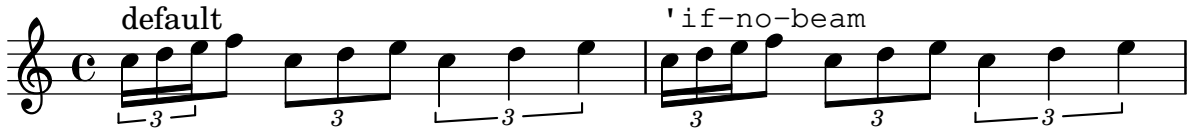
Contrôle de l'impression des crochets de n-olet

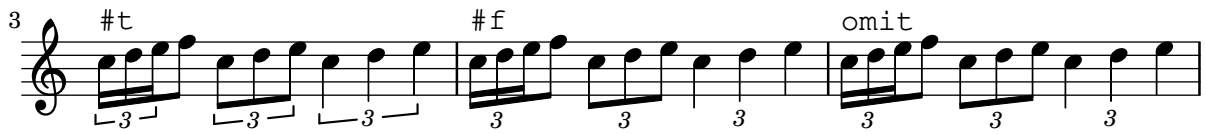
Selon la tradition, les crochets indicateurs de n-olet sont toujours imprimés, sauf dans le cas où ils seraient de la même longueur qu'une ligature.

LilyPond permet, au travers de la propriété `bracket-visibility`, de contôler précisément leur affichage : déterminée à `#t`, ils seront toujours imprimés ; `#f` permet de ne jamais les imprimer – donc omettre l'objet `TupletBracket` –, et `if-no-beam` les imprimera en l'absence de ligature (comportement par défaut).

```
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
  }
}
```
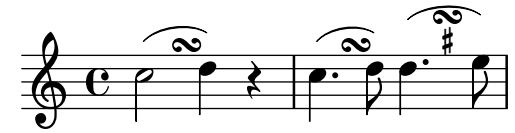




Création d'un grupetto retardé

Obtenir un *gruppetto* retardé et dans lequel la note la plus basse est altérée requiert quelques surcharges. La propriété `outside-staff-priority` doit être désactivée (`#f`) pour éviter qu'elle prenne le pas sur la propriété `avoid-slur`. L'ajustement du premier argument de `\after` (il s'agit d'une durée) permet de jouer sur le positionnement 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)
}
```



Création d'armures personnalisées
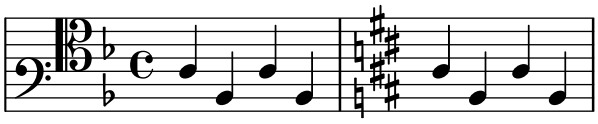
LilyPond accepte les armures personnalisées. Dans cet exemple, il s'agit d'une tonalité de ré mineur et ré majeur auxquelles on rajoute des bémols.

```
\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, |
}
```



Création d'extensions de texte

Les commandes \startTextSpan et \stopTextSpan permettent d'ajouter une ligne de prolongation aux indications textuelles, à l'instar des indications de pédale ou d'octaviation. Jouer sur les propriétés de l'objet TextSpanner permet d'en modifier le rendu.

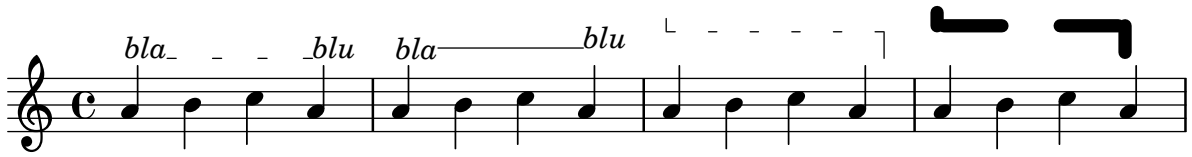
```
\paper { ragged-right = ##f }

\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
}
```



Accord distribué et problème de hampe – solution

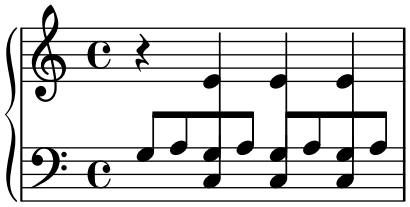
Il est parfois préférable d'utiliser les hampes de l'autre portée pour créer des accords distribués, afin de tricher avec le détecteur de collision des ligatures de LilyPond. Dans l'exemple suivant, le fait de partir des hampes de la portée inférieure aurait rendu nécessaire l'adaptation du détecteur de collision des ligatures, par une clause

```
\override Staff.Beam.collision-voice-only = ##t
```

afin que LilyPond ne déplace pas les ligatures.

```
\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
  }
>>
```



Hampes interportées

L'exemple ci-dessous illustre l'utilisation du `Span_stem_engraver` et de la commande `\crossStaff` afin de connecter des hampes entre les portées.

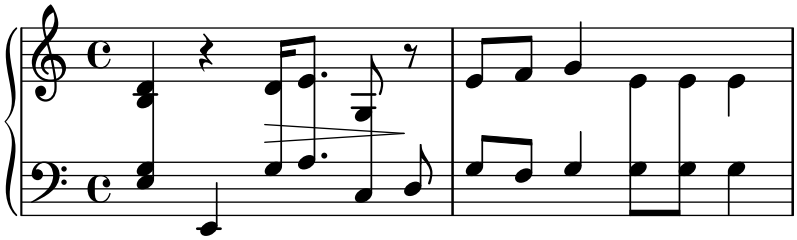
Nul n'est besoin de spécifier la taille des hampes ; le graveur calcule automatiquement la distance relative des têtes de note avec les portées. Il est toutefois important d'appliquer

\crossStaff dans le contxte de voix ou de portée approprié – autrement dit en opposition à l'endroit où se trouvera une hampe) pour obtenir l'effet attendu.

```
\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 |
  }
>>
```



Guidons

Les guidons peuvent adopter différents styles.

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

\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

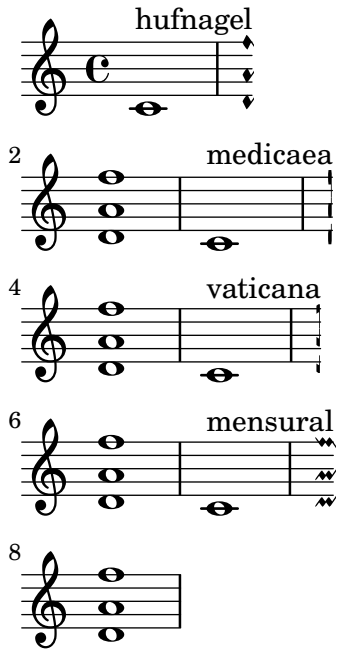
    \override Staff.Custos.style = #'hufnagel
    c1^"hufnagel" \break
    <d a' f'>1

    \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
  }
}
```



Personnalisation de diagrammes de fret

Les propriétés d'un diagramme de fret peuvent se modifier à l'aide de la propriété `'fret-diagram-details`. En matière de diagramme de fret, les adaptations s'appliquent à l'objet `FretBoards.FretBoard`. Un `FretBoards` est comparable à un `Voice` : il s'agit d'un contexte du plus bas niveau, et il n'est donc pas primordial de l'instancier de manière explicite pour adapter ses propriétés.

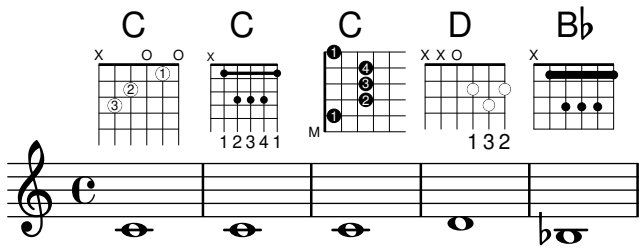
```
\include "predefined-guitar-fretboards.ly"

\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
  }
>>
```



Personnalisation des diagrammes de fret

Les propriétés d'un diagramme de fret sont modifiables grâce à la propriété `fret-diagram-details`. Lorsqu'ils sont générés sous forme de `\markup`, rien n'empêche de

modifier les diagrammes en jouant sur les réglages de l'objet `Voice.TextScript` ou bien directement sur le *markup*.

```
<<
  \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

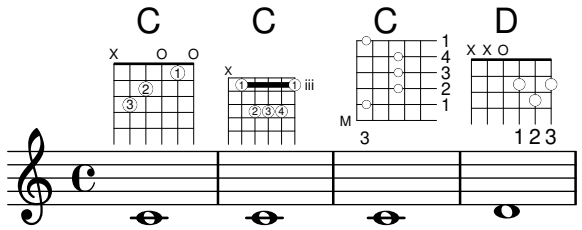
    %% C major for guitar, no barre, using defaults
       % terse style
    c'1^\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

    %% 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;"
      }
    }
  }
>>
```



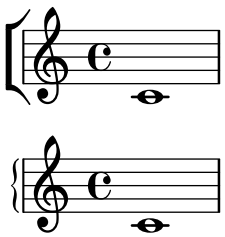
Indicateur de regroupement et portée unique

Lorsque, dans des regroupements de type `ChoirStaff` ou `StaffGroup`, une seule portée est active, aucune indication n'est donnée en début de ligne. Affecter à la propriété `collapse-height` un nombre de lignes inférieur à celui de la portée permet de modifier ce comportement par défaut.

Notez bien que dans le cas des `PianoStaff` et `GrandStaff`, pour lesquels le délimiteur de système est une accolade et non un crochet, il ne s'agit pas de la même propriété – voir le deuxième système de l'exemple.

```
\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
    }
  >>
}
```



Affichage de la généalogie d'un objet

Lorsque l'on manipule des rappels d'objet (*grob callbacks*), il peut être intéressant d'en maîtriser les ascendants. La plupart des objets graphiques ont des parents, lesquels auront une influence sur le positionnement de l'objet en question. Ainsi, les parents des côtés X et Y influenceront respectivement les positions horizontale et verticale de l'objet. De plus, chacun des parents peut avoir ses propres parents.

Certains aspects de la lignée d'un objet peuvent toutefois porter à confusion :

- Les types de parents d'un *grob* peuvent dépendre du contexte.
- Dans le cas de certains *grobs*, les parents X et Y peuvent être le même.
- Un « ascendant » particulier peut dépendre d'un *grob* de différentes manières.
- Le concept de « générations » est trompeur.

Par exemple, l'objet `System` peut, vis à vis d'un objet `VerticalAlignment`, être à la fois parent (par son côté Y) et grand parent (par deux fois du côté X).

La macro définie ci-dessous affiche à l'écran une représentation textuelle de l'ascendance d'un *grob*. Par exemple, appeler

```
{
  \once \override NoteHead.before-line-breaking = #display-ancestry
  c
}
```

génère la sortie suivante :

```
-----------------------------------

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
```
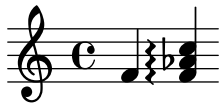
Par voie de conséquence, il vous faut exécuter vous-même le code de cet extrait, puisque le fichier résultant ne contient pas les données qui nous intéressent.

```
#(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))))
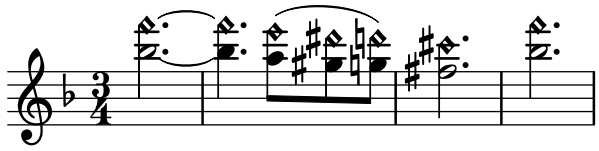
\relative c' {
  \once \override NoteHead.before-line-breaking = #display-ancestry
  f4
  \once \override Accidental.before-line-breaking = #display-ancestry
  \once \override Arpeggio.before-line-breaking = #display-ancestry
  <f as c>4\arpeggio
}
```



Harmoniques pointées

Les notes harmoniques artificielles, obtenues grâce à \harmonic, ne sont pas pointées. Ce comportement peut être modifié en activant la propriété de contexte 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.
}
```



Encadrement d'objets

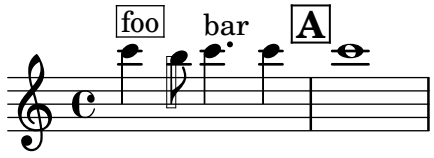
La propriété stencil peut se modifier pour obtenir l'encadrement de n'importe quel objet, que ce soit à l'aide d'un \override ou d'un \tweak.

```
\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
}
```

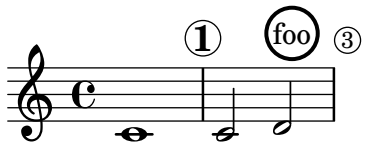


Encercler divers objets

La commande \circle permet de dessiner un cercle autour d'objets \markup. D'autres objets nécessitent de faire appel à des techniques spécifiques. Cet exemple illustre deux manières de procéder : pour les repères, et pour les numéros de mesure.

```
\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)
}
```
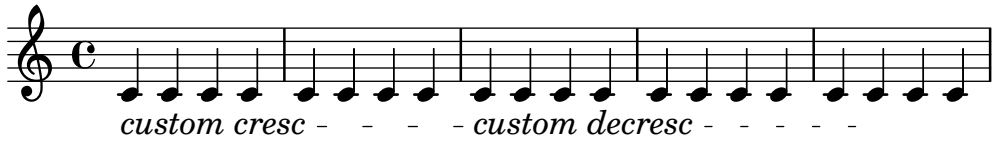


Personnalisation des extenseurs de nuance textuelle

Il s'agit de fonctions postfix pour personnaliser l'extension des crescendos textuels. L'extension devrait débuter sur la première note de la mesure. Il faut utiliser -\mycresc – comme une articulation – sous peine que le départ de l'extension n'apparaisse qu'à la note suivante.

```
% 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
}
```

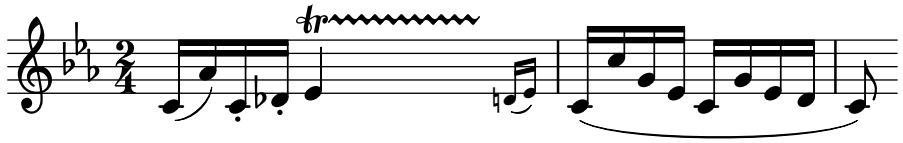


Allongement d'une marque de trille (TrillSpanner)

La propriété minimum-length d'une extension de trille – objet TrillSpanner – n'est effective qu'après un appel explicite à la procédure set-spacing-rods.

Pour ce faire, la propriété set-spacing-rods doit être définie à 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)
}
```



Rappel du glissando à l'occasion d'une alternative

Un glissando qui se prolonge sur plusieurs sections \alternative peut se rappeler à l'aide d'une note d'ornement supplémentaire et masquée, à laquelle sera attaché le départ du glissando, ce dans chaque bloc \alternative. Cette note d'ornement devrait avoir la même hauteur que la note où commençait le glissando originel. Ceci est géré par une fonction musicale qui prendra en argument la hauteur de la note d'ornement.

Dans le cadre d'une musique polyphonique, il ne faudra pas oublier d'ajouter une note d'ornement dans toutes les autres voix afin de préserver la synchronisation.

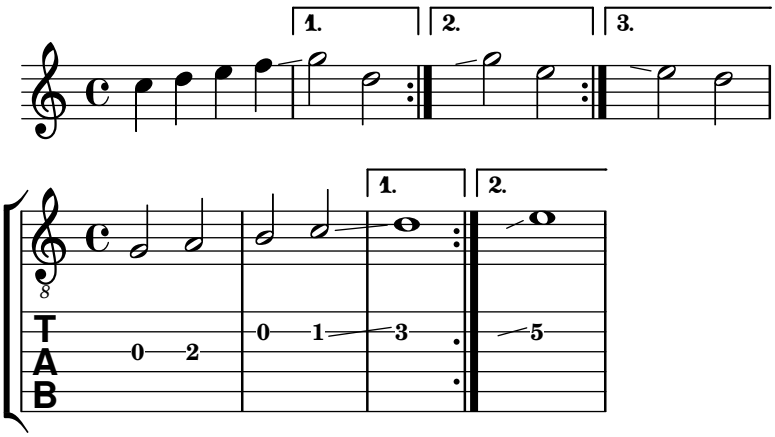
```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \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 }
    >>
  >>
}
```
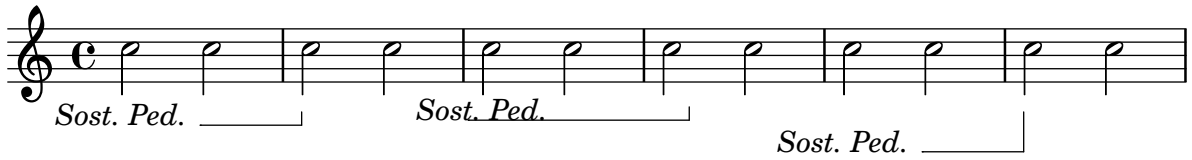


Affinage des indications de pédale

Les crochets d'indication de pédale peuvent se modifier de différentes manières.

```
\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
}
```



Liaison de prolongation aplatie

La fonction `flared-tie` ici présentée permet de dessiner une liaison de prolongation constituée de lignes droites. Elle est sensée remplacer la fonction par défaut de dessin des liaisons de tenue – elle remplace l'argument `stencil` de l'objet `Tie`.

L'argument de cette fonction `flared-tie` est une liste de paires de coordonnées qui spécifient des points additionnels entre l'entame et la terminaison afin de déterminer l'étendue des lignes de la liaison. L'entame et la terminaison sont identiques à ceux de la liaison originelle. La valeur des abscisses et ordonnées sont des multiples de largeur et de la hauteur de la boîte englobante de la liaison originelle (tout en tenant compte de son positionnement) ; par voie de conséquence, le premier point a pour coordonnée (0,0), et le dernier point (1,0).

La fonction `flare-tie` définit un raccourci pour l'obtention d'une liaison aplatie. Des ajustements plus poussés sont possibles, en apportant des dérogations à `Tie.details.height-limit` ou à l'aide de la fonction `\shape`. La définition personnalisée peut aussi se modifier à la volée.

```
#(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

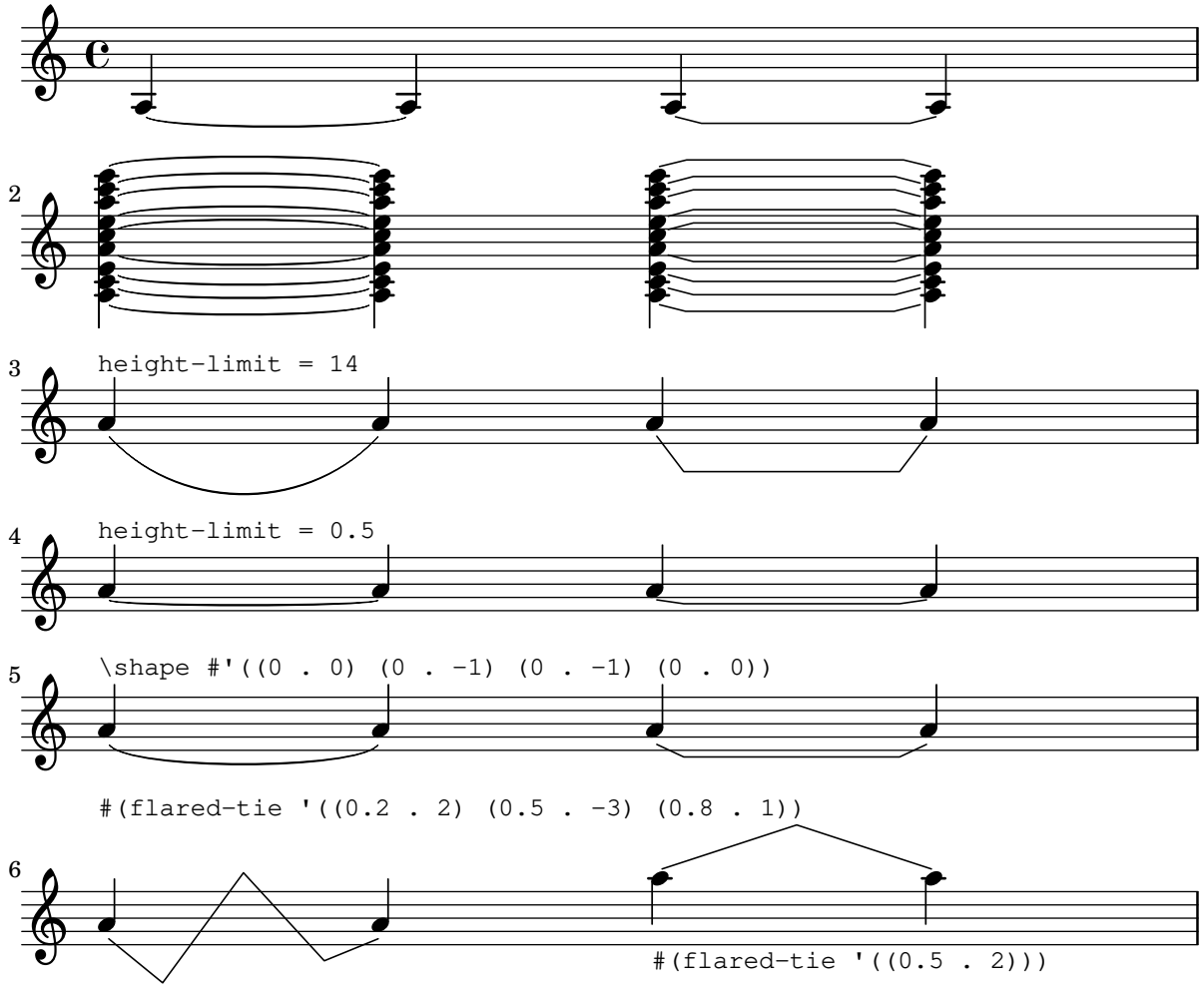
  <>^\markup \small \typewriter "height-limit = 0.5"
  \override Tie.details.height-limit = 0.5
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  \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))"
```
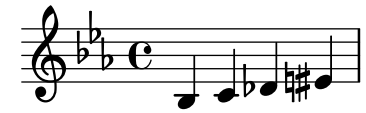
```
  \once \override Tie.stencil =
        #(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))
  a4~ a
  <>_\markup \small \typewriter
             "#(flared-tie '((0.5 . 2)))"
  \once \override Tie.stencil = #(flared-tie '((0.5 . 2)))
  a'4~ a
}
```



Impression forcée d'un bécarre avant une accidentelle

L'exemple suivant indique comment forcer l'impression d'un bécarre avant un changement d'altération.
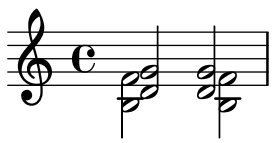
```
\relative c' {
  \key es \major
  bes c des
  \tweak Accidental.restore-first ##t
  eis
}
```



Décalage horizontal forcé

Quand LilyPond est dépassé, la propriété `force-hshift` de l'objet `NoteColumn` et des silences à hauteur déterminée peuvent s'avérer utiles pour dicter au programme les choix de placement. On travaille ici en espace de portée.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
    <b f'>2
    \once \override NoteColumn.force-hshift = 1.7
    <b f'>2
  }
>>
```



Construction et développement de diagrammes de fret

Voici différentes manières d'obtenir et de personnaliser des diagrammes de fret :

```
<<
  \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))

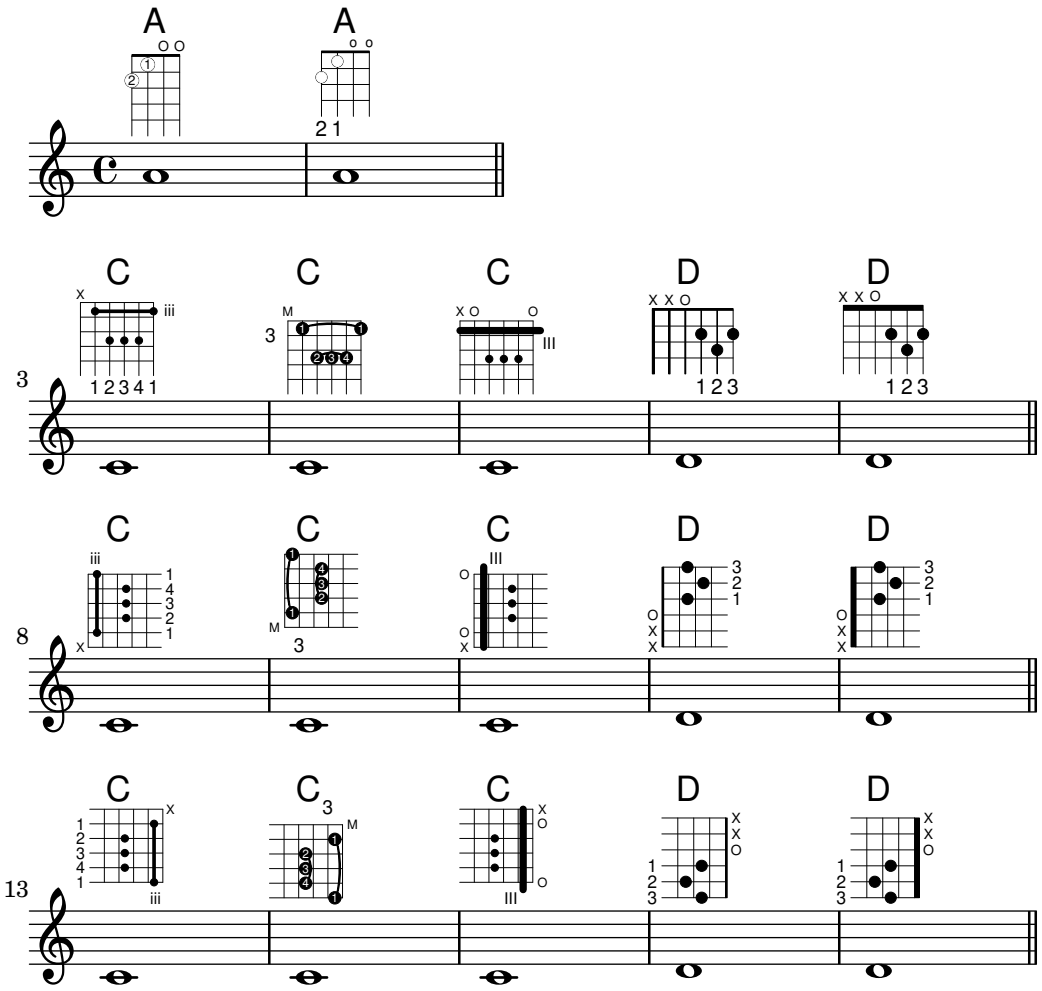
    % 16
    %
    % Simple D chord.
    d'1^\markup
      \override #'(fret-diagram-details
                   . ((finger-code . below-string)
                      (dot-radius . 0.35)
                      (dot-position . 0.5)
                      (fret-count . 3)))
      \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

    % 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
  }
}
```



Génération de notes profilées spéciales

Lorsque le profil spécifique d'une tête de note ne peut être obtenu facilement à l'aide d'un *markup* graphique, il est possible d'utiliser des instructions de dessin pour `ly:make-stencil` afin de générer ce profil. Cet exemple propose une tête de note profilée en parallélogramme.

Malheureusement, les commandes disponibles pour spécifier le dessin ne sont pas documentées à ce jour – c'est l'objet du ticket #6874 (`https://gitlab.com/lilypond/lilypond/-/issues/6874`) – mais la sous-commande `path` a la signature suivante, très proche de la fonction Scheme `make-path-stencil`.
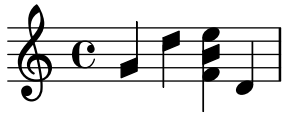
```
(path épaisseur liste-commandes style-ligne-principale style-jonction-ligne remplissage
```

Les commandes dans *liste-commandes* ressemblent aux instructions de dessin en PostScript, à ceci près que les arguments suivent ici le nom de la commande.

```
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
}
```



Génération de crochets personnalisés

Une fonction Scheme personnalisée permet de redéfinir la propriété `stencil` de l'objet `Flag`, de sorte à modifier le glyphe utilisé pour les crochets de croche.

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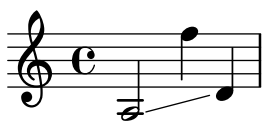
  <>_"Revert to normal"
  \revert Flag.stencil
  \snippetexamplenotes
}
```



Glissando par dessus un objet graphique

Un glissando peut sauter un objet NoteColumn.

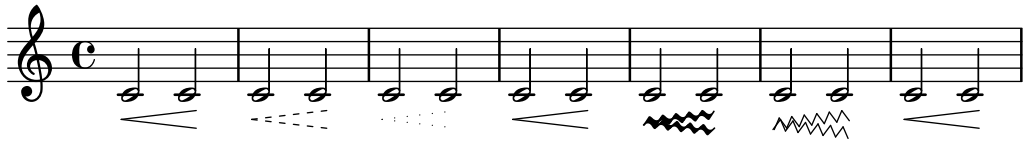
```
\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}
```



Stylisation des lignes de soufflet

Les lignes d'un soufflet peuvent adopter tous les styles permis par la line-interface : dashed-line (discontinu), dotted-line (pointillé), line (continu), trill (ondulé) ou zigzag.

```
\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\!
}
```



Alignement de nuances personnalisées comme « più f »

Certaines indications de nuance requièrent un complément textuel, comme « sempre **pp** ». Dans la mesure où les nuances sont habituellement centrées sous la note, le *pp* se trouvera repoussé loin après la note à laquelle il s'applique.

Différentes approches permettent de correctement aligner horizontalement ce « sempre **pp** » comme s'il ne s'agissait que d'un simple *pp* :

- Un simple `\once \override DynamicText.X-offset = #-9.2` avant la note considérée de telle sorte que la nuance soit décalée manuellement à la bonne place. Inconvénient : il faut le faire manuellement à chaque fois qu'intervient ce *markup*.
- L'intégration d'un rembourrage (`#:hspace 7.1`) à la définition de cette nuance personnalisée afin qu'une fois centrée par LilyPond elle soit correctement alignée. Inconvénient : le rembourrage occupera exactement cet espace et ne permettra à aucun autre *markup* ou nuance d'apparaître à cet endroit.
- L'application d'un décalage au script de nuance `\once \override ... .X-offset = ...`. Inconvénient : là aussi il faut le faire à chaque fois.
- L'attribution arbitraire d'une dimension à 0 du texte additionnel à l'aide d'un `#:with-dimensions '(0 . 0) '(0 . 0)`. Inconvénient : LilyPond considère que « sempre » n'occupe pas d'espace, et donc pourra mettre à cet endroit d'autres éléments, ce qui pourrait générer des collisions qui ne seront pas détectées par les routines *ad hoc*. D'autre part, il semble persister un espacement, ce qui laisse l'impression d'un alignement différent en l'absence de texte additionnel.
- L'ajout, explicite, du décalage directement dans la fonction Scheme du script de nuance.
- La définition d'un alignement explicite au sein du script de nuance. Ceci ne sera suivi d'effet, par défaut, qu'en jouant sur `X-offset`. Inconvénient : il faut définir `DynamicText.X-offset`, ce qui s'appliquera à toutes les nuances textuelles. Par ailleurs, l'alignement sera réalisé sur le bord droit du texte additionnel, non sur le milieu de *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"
```

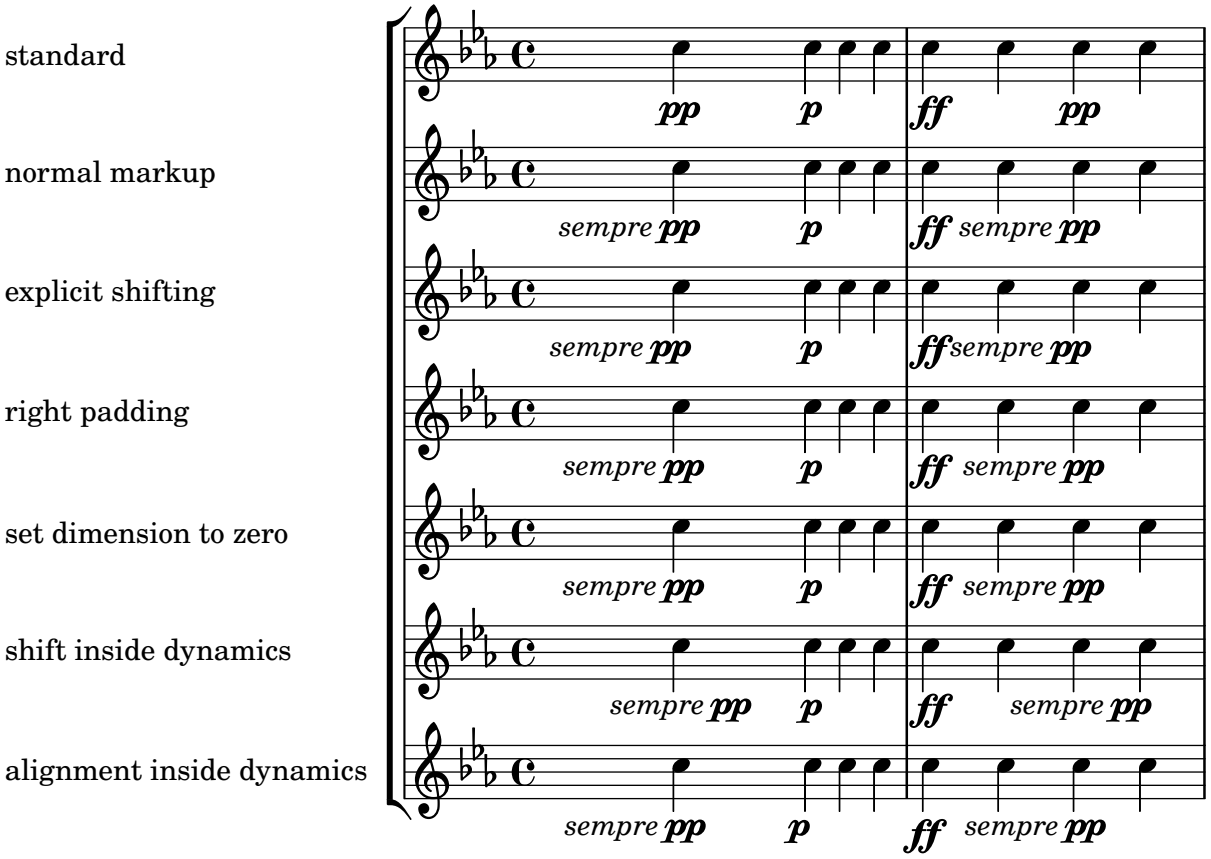
```
                #:dynamic "pp")))

% 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 }
```



Repositionnement d'un diagramme de fret

Différents moyens permettent de repositionner un diagramme de fret pour éviter des collisions ou le placer entre deux notes :

1. La modification des valeurs de `padding` ou de `extra-offset` comme pour la première ligne ;
2. L'adjonction d'une voix invisible dans laquelle les diagrammes sont attachés à des notes invisibles comme pour la deuxième ligne.

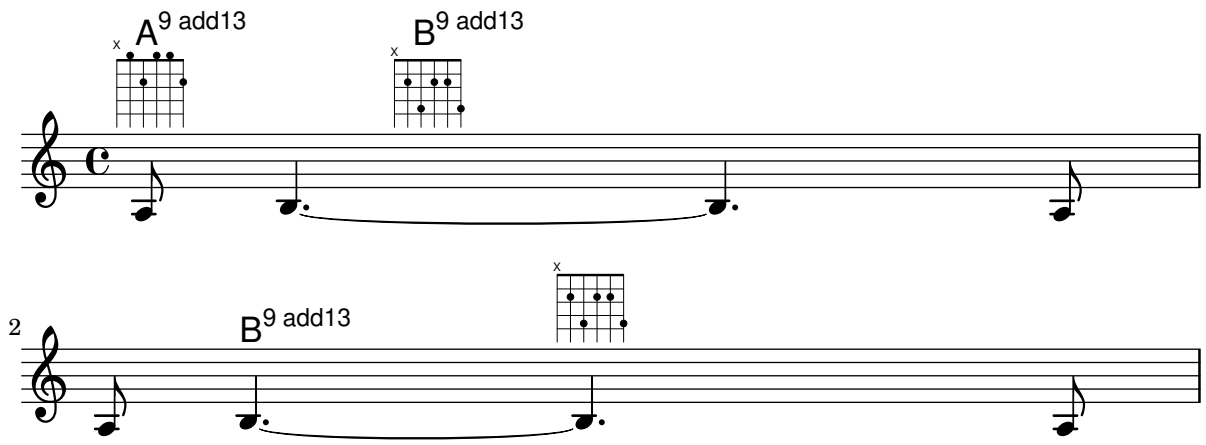
Lorsque le diagramme doit correspondre à une position rythmique dans la mesure, comme au troisième temps de la deuxième mesure, la seconde méthode est plus appropriée puisque le diagramme sera aligné sur le temps.

```
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;" }
      >> |
    }
  >>
}
```
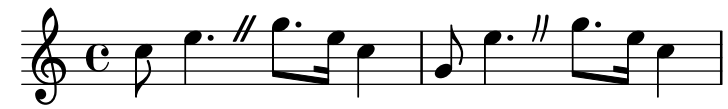


Insertion d'une césure

Une surcharge de la propriété `text` de l'objet `BreathingSign` permet de créer une marque de césure.

LilyPond dispose également d'une variante courbée.

```
\relative c'' {
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4
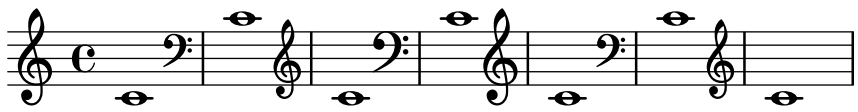
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```



Conservation de la taille lors d'un changement de clef

Lorsqu'un changement de clef intervient, le nouveau signe s'imprime dans une taille inférieure. On peut y déroger en activant la propriété de contexte `full-size-change`.

```
\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
}
```



Terminaison de ligne en flèche

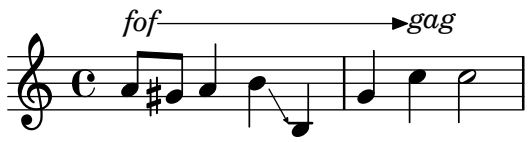
Les extensions de texte, tout comme les indications sous forme de ligne tel un glissando, peuvent se voir pourvues d'une extrémité en flèche.

```
\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
}
```



Recours à la propriété `transparent` pour rendre des objets invisibles
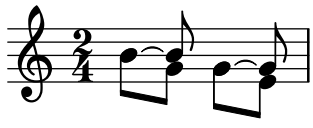
Affecter un objet graphique d'un `\hide` permet d'imprimer un objet « à l'encre sympathique » : l'objet n'est pas affiché bien que tous ses attributs soient préservés :

- l'objet en question occupe donc sa place,
- il est pris en compte lors de collisions, et
- il peut se voir attaché liaisons ou ligatures comme les autres.

Cet exemple illustre la manière de connecter deux voix par une liaison de prolongation. Les liaisons de prolongation ne peuvent normalement intervenir que dans la même voix. Dès lors que la liaison est entamée dans une autre voix et que la première hampe ascendante est masquée transparente dans cette même voix, la liaison semble passer d'une voix à l'autre.

```
\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
}
```



Saut de ligne et glissando

Normalement, LilyPond refuse de sauter automatiquement à la ligne à l'endroit où un glissando se prolonge sur la mesure qui suit. Ce comportement est modifiable par l'affectation de la valeur `#t` à la propriété `Glissando.breakable`. L'activation de la propriété `after-line-breaking` permet la continuation de l'indication de glissando après le saut de ligne.

La propriété `breakable` n'affecte pas les sauts manuels insérés par les commandes comme `\break`.
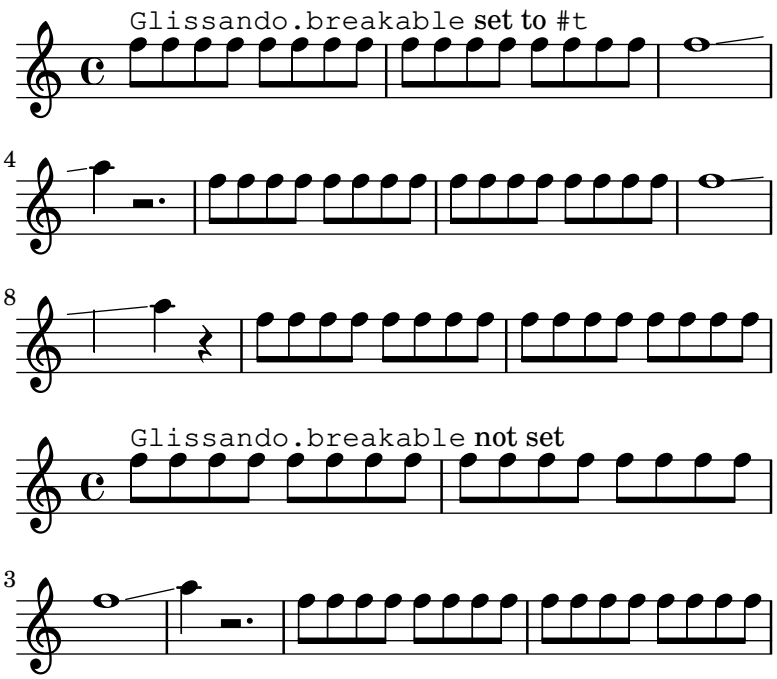
```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
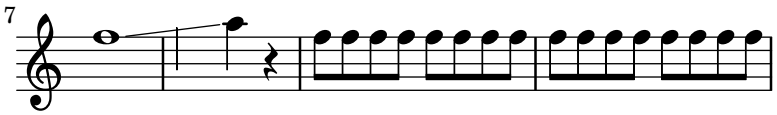
music = {
  \*16 f8 |
  f1\glissando |
  a4 r2. |
  \*16 f8 |
  f1\glissando \once\glissandoSkipOn |
  a2 a4 r4 |
  \*16 f8
}

\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
}
```
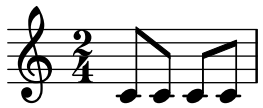




Contrôle manuel du positionnement des ligatures

Le positionnement et la pente des ligatures peuvent être contrôlés manuellement à l'aide d'une adaptation de la propriété `positions` de l'objet `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
}
```



Centrage des numéros de mesure

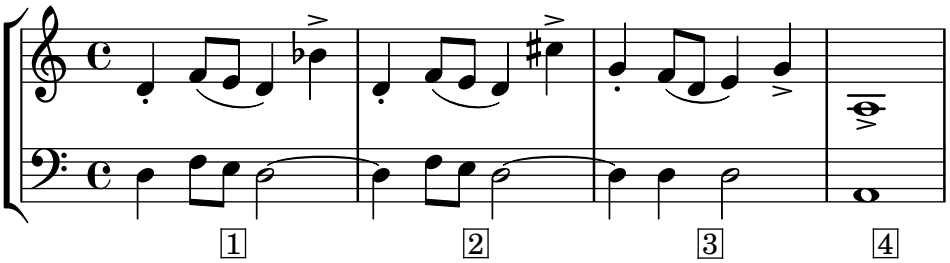
Il est d'usage, dans les partitions de musique de film, de trouver les numéros de mesure centrés sur leur mesure. Ceci s'obtient en activant la propriété de contexte `centerBarNumbers`. Lorsque cette propriété est utilisée, le type de *grob* (objet graphique) `BarNumber` est remplacé par `CenteredBarNumber` .

L'exemple ci-dessous illustre plusieurs réglages : les numéros de mesure sont à la fois centrés, encadrés, et disposés sous les portées.

```
\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 |
  }
>>
```



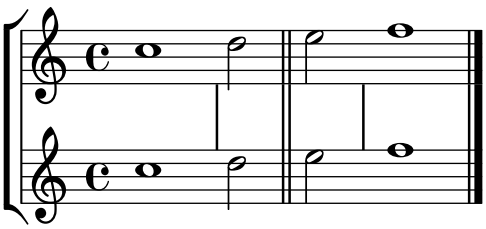
Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat il faudra définir `measureBarType` à `"-span|"` et utiliser un regroupement de portées permettant l'extension des barres entre les portées, tel un `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```

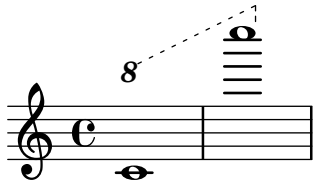


Modification de la pente de l'extension d'octaviation

Il est possible d'adapter la pente d'une indication d'octaviation.

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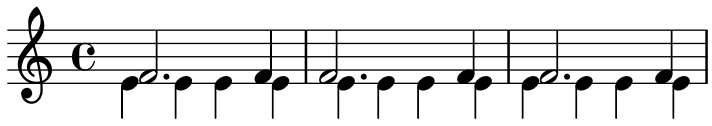
```
                (stencil-align-dir-y . ,CENTER)))
       (right . ((Y . 5.0) ; Change the number here
                 (padding . 0)
                 (attach-dir . ,RIGHT)
                 (text . ,(make-draw-dashed-line-markup
                           (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info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



Déplacement des notes pointées dans une polyphonie

Une note pointée appartenant à la voix supérieure d'une portée polyphonique sera par défaut décalée vers la droite afin d'éviter les collisions avec les autres voix. Ce comportement peut être modifié par un réglage de la propriété `prefer-dotted-right` de l'objet graphique `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 }
>>
```
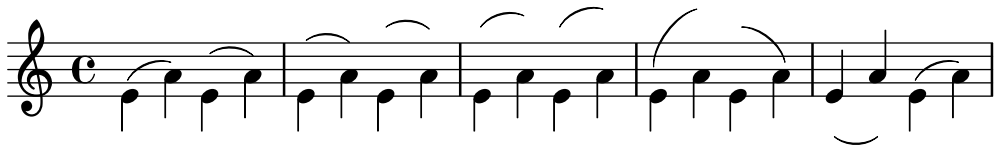


Déplacement vertical des liaisons d'articulation

Le positionnement vertical d'une liaison se contrôle par la propriété `positions` de l'objet `Slur`. Cette propriété dispose de deux paramètres : le premier affecte l'extrémité gauche de la liaison, le second son extrémité droite. La valeur des paramètres n'aura aucune influence sur le galbe de la liaison. LilyPond ne s'en servira que pour adapter le positionnement de la courbe. Des valeurs positives décalent la liaison vers le haut et s'utilisent pour des hampes descendantes. Des valeurs négatives entraînent un décalage vers le bas.

Voir aussi l'exemple « Ajustement du positionnement vertical d'une liaison ».

```
\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)
}
```



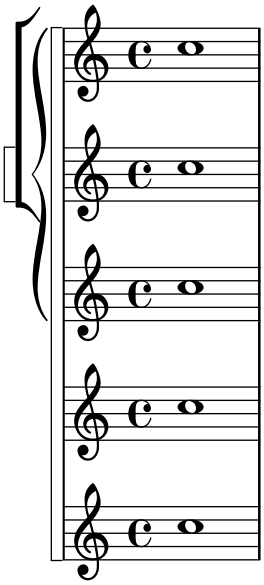
Imbrications de regroupements de portées

La propriété de contexte systemStartDelimiterHierarchy permet de créer des regroupements imbriqués complexes. Dans le cadre d'un StaffGroup, elle prend en argument la liste alphabétique des sous-groupes à hiérarchiser. Chaque sous-groupe peut être affublé d'un délimiteur particulier. Chacun des regroupements intermédiaires doit être borné par des parenthèses. Bien que des éléments de la liste puissent être omis, le premier délimiteur embrassera toujours l'intégralité des portées. Vous disposez des quatre délimiteurs SystemStartBar, SystemStartBracket, SystemStartBrace et 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 }
>>
```



Personnalisation de certains types d'articulation

On peut parfois vouloir modifier un seul type d'articulation. Bien que ce soit tout à fait faisable avec un \tweak, cela devient vite fastidieux d'affiner chaque occurrence d'un même signe dans toute une partition. Le code ci-dessous illustre la manière de modifier des articulations grâce à une liste de réglages personnalisés. Ceci peut servir à créer des feuilles de style.

```
#(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)
```

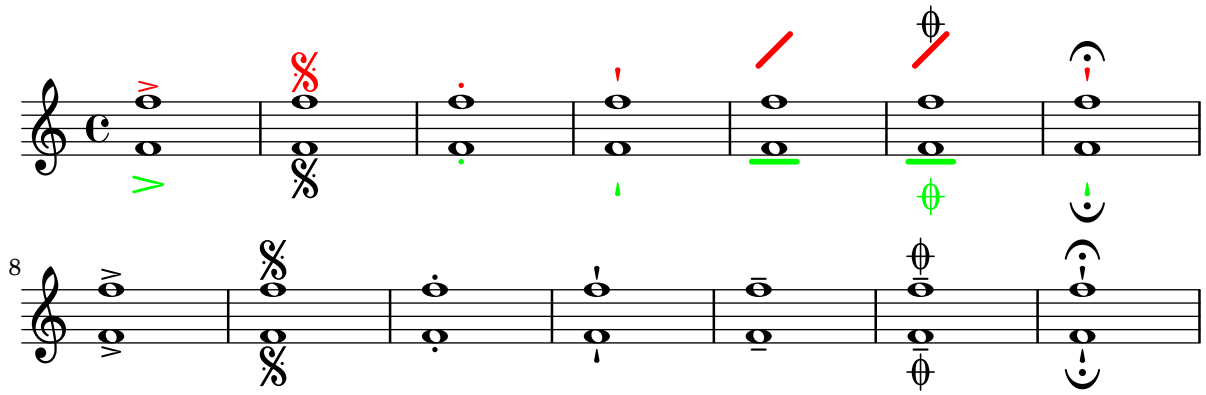
```
                (color . (1 0 0))))
     (segno . ((font-size . 0)
               (color . (1 0 0))))
     (staccato . ((color . (1 0 0))
                  (padding . 0.5)))
     (staccatissimo . ((padding . 1)
                       (color . (1 0 0))))
     (tenuto . ((color . (1 0 0))
                (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
}

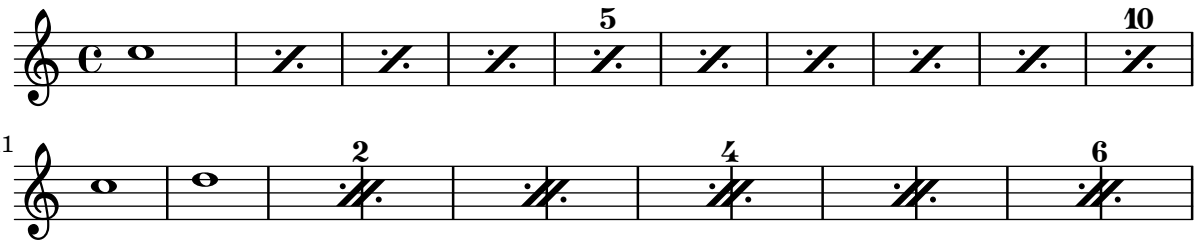
\new Staff <<
  \new Voice \with { \customScripts #my-settings-1 }
    \relative c''{ \voiceOne \block }
  \new Voice \with { \customScripts #my-settings-2 }
    \relative c' { \voiceTwo \block }
>>
```



Affichage du numéro de répétition en pourcent

Le numéro de mesure répétée sera imprimé à intervalle régulier si vous déterminez la propriété de contexte `repeatCountVisibility`.

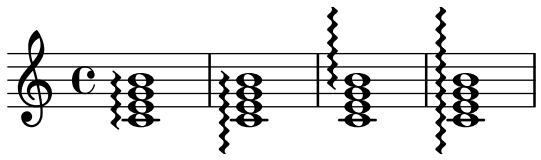
```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \%10 c1 \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \%6 { c1 d1 }
}
```



Positionnement des arpeggios

L'ajustement de la taille d'une indication d'arpeggio s'effectue au travers du positionnement de ses extrémités haute ou basse.

```
\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
}
```



Positionnement précis des indications de doigté

Les options par défaut en matière de positionnement des doigtés d'un accord donnent généralement de bons résultats. Il se peut néanmoins qu'un ajustement soit nécessaire dans certains cas particuliers, notamment en présence d'un intervalle de seconde. L'astuce ici présentée permet d'obtenir un meilleur rendu.

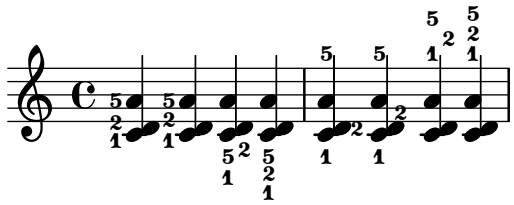
```
\score {
  \relative c' {
    \set fingeringOrientations = #'(left)
    <c-1 d-2 a'-5>4
    <c-1 d-\tweak extra-offset #'(0 . 0.2)-2 a'-5>

    \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> |
  }
}
```



Positionnement des silences multimesures

Si l'on peut positionner verticalement un silence simple en le rattachant à une note, il n'en va pas de même pour un silence multimesure. Néanmoins, et uniquement dans le cadre de musique polyphonique, les silences multimesures sont positionnés différemment selon qu'ils appartiennent à une voix au numéro pair ou impair.

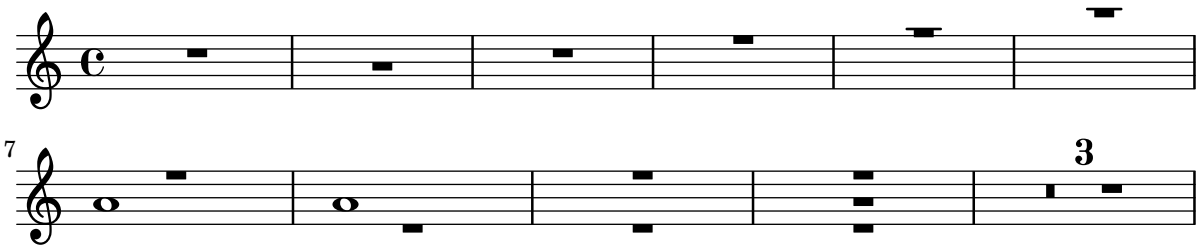
Le positionnement des silences multimesures peut se contrôler comme ici indiqué.

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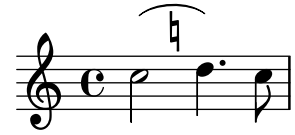
```
  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed.
  \compressMMRests
  <<
    \revert MultiMeasureRest.direction
    { R1*3 } \\
    \revert MultiMeasureRest.direction
    { R1*3 }
  >>
}
```



Positionnement d'une annotation à l'intérieur d'une liaison

Lorsqu'une annotation doit s'incrire à l'intérieur d'une liaison, la propriété `outside-staff-priority` doit être désactivée.

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



Inscrire le numéro de mesure dans un cadre ou un cercle

Les numéros de mesure peuvent être encadrés ou entourés d'un cercle.

```
\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

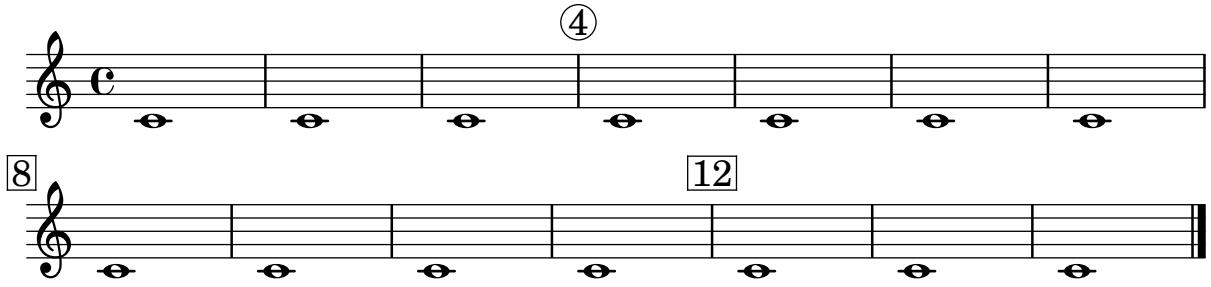
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % 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 "|."
}
```



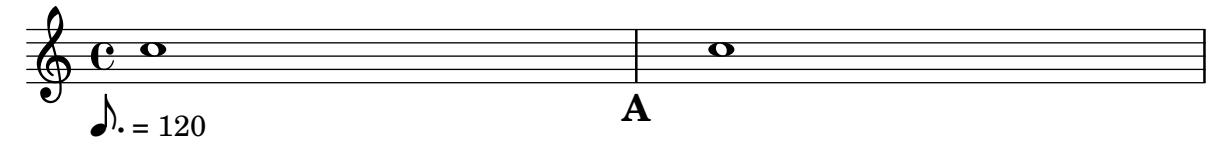
Impression du métronome et des repères sous la portée

Les indications de tempo et les marques de repère s'impriment par défaut au-dessus de la portée. Le fait de régler en conséquence la propriété `direction` des objets `MetronomeMark` et `RehearsalMark` les placera au-dessous de la portée.

```
\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
}
```



Impression du nom des notes avec ou sans indication d'octave
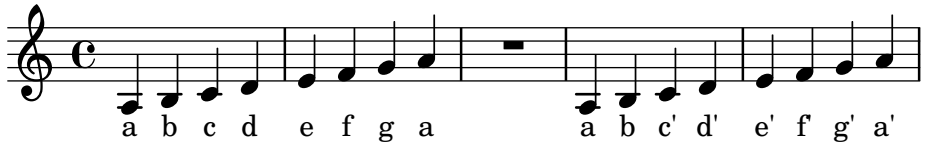
Le contexte `NoteNames` permet d'imprimer le nom des notes. La propriété `printOctaveNames`, une fois activée, leur adjoindra une indication d'octave.

```
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
  }
}
```



Affichage du crochet de n-olet du côté des têtes de note
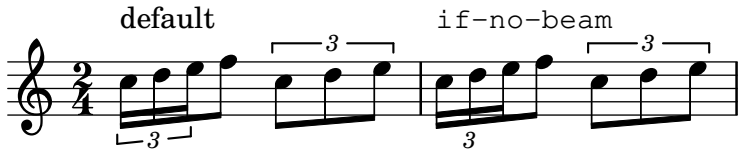
Quelle que soit l'option choisie pour contrôler la visibilité d'une indication de n-olet, elle s'affichera ou sera masquée indépendamment de son positionnement (côté hampe ou tête de note). Toutefois, lorsqu'elle se place du côté des têtes de note, certains auteurs recommandent de toujours afficher un crochet. L'option `visible-over-note-heads` permet d'y parvenir.

```
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
  }
}
```

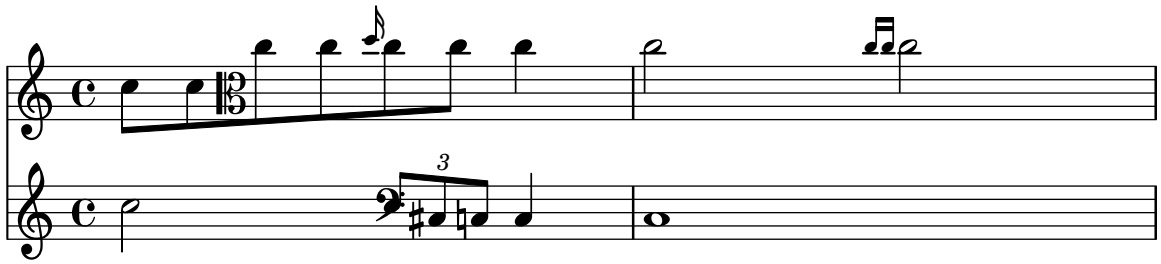


Espacement strictement proportionnel des notes

Lorsque la propriété `strict-note-spacing` de l'objet SpacingSpanner est activée, l'espacement des notes dans un système ne tient compte ni des barres de mesure ni des clefs, qui se retrouvent placées juste avant la note qui tombe au même moment. Ceci peut entraîner certaines collisions.

```
\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
  }
>>
```



Retrait de l'accolade à la première ligne d'une pièce pour piano

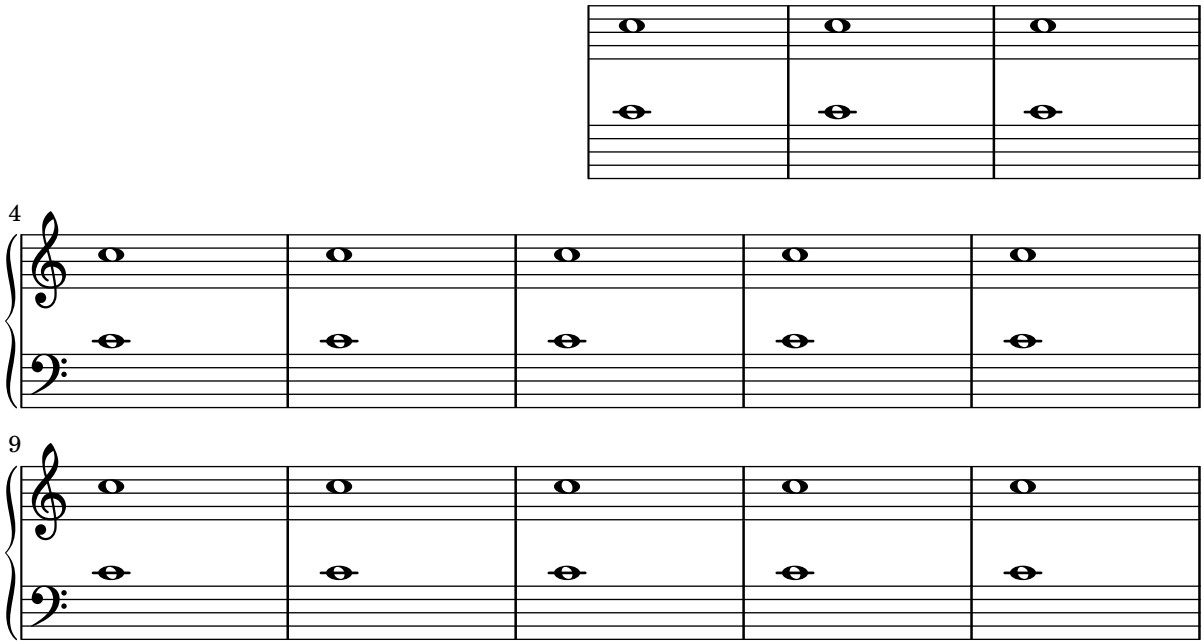
Dans cet extrait est supprimée la première accolade d'un PianoStaff ou d'un GrandStaff, ainsi que les clefs. Ceci peut s'avérer utile pour couper et coller l'image générée dans de la musique préexistante.

Est utilisée la fonction `\alterBroken` afin de masquer l'accolade du début.

```
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
    }
  }
}
```

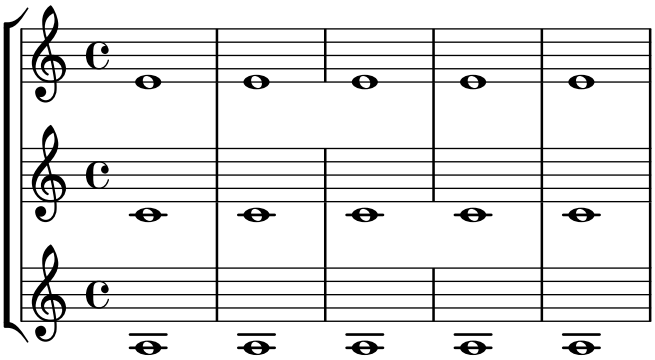


Suppression de la partie interportée des barres de mesure d'un regroupement autre que ChoirStaff

Les barres de mesure des regroupements StaffGroup, PianoStaff et GrandStaff sont par défaut d'un seul tenant. La portion entre les portées – l'objet SpanBar – peut néanmoins être supprimée, portée par portée.

```
\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
    }
  >>
}
```



Masquage de la première ligne si elle est vide

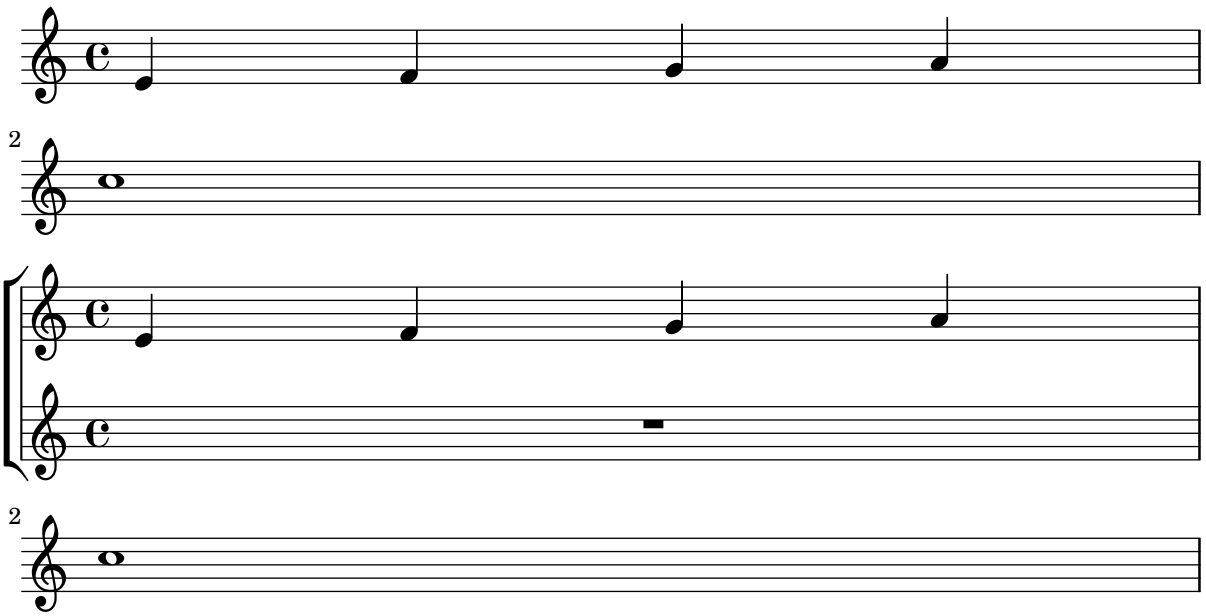
Par défaut, le premier système comportera absolument toutes les portées. Si vous préférez masquer les portées vides y compris pour le premier système, vous devrez activer la propriété `remove-first` de l'objet `VerticalAxisGroup`. Mentionnée dans un bloc `\layout`, cette commande agira de manière globale. Pour qu'elle ne soit effective que pour une portée particulière, vous devrez également spécifier le contexte (`Staff` pour qu'il ne concerne que la portée en cours) en préfixe de la propriété.

La première ligne inférieure du deuxième `StaffGroup` est bien présente, pour la simple raison que le réglage en question ne s'applique qu'à la portée dans laquelle il a été inscrit.

```
\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
  }
>>
```



Styles de silences

Les silences peuvent être gravés selon différents styles.

```
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

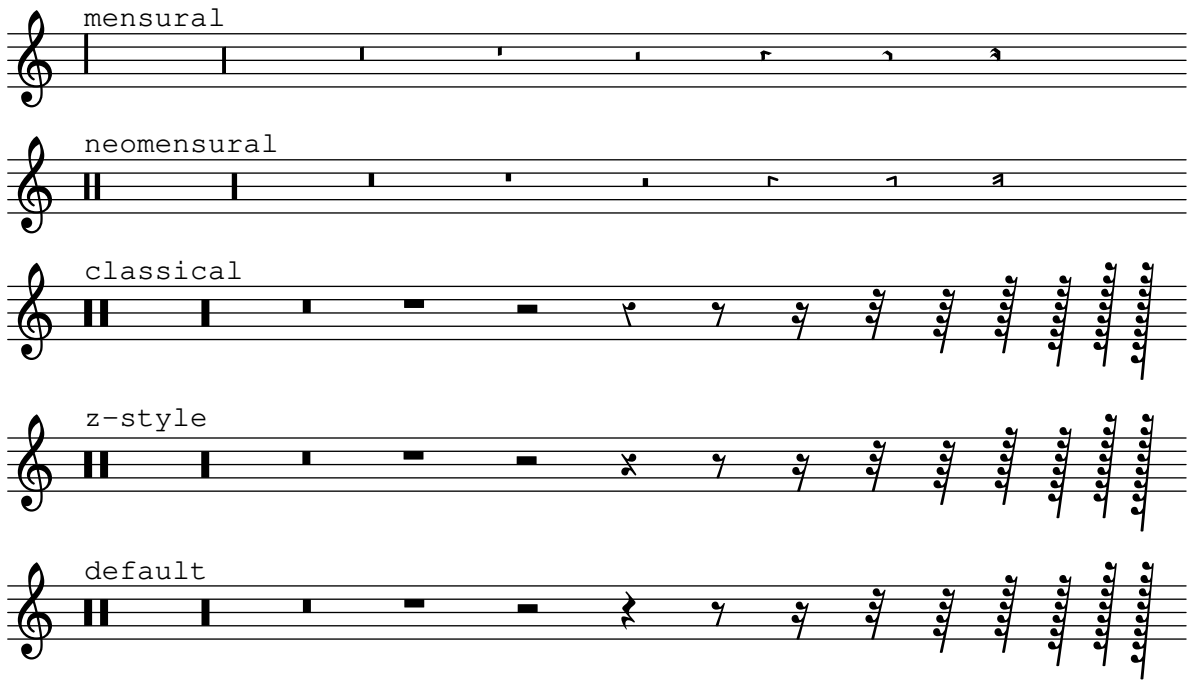
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

  \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
}
```
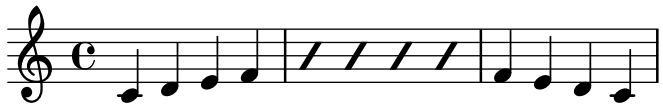


Barres rythmiques

Il arrive, dans une feuille de chant « simple », que les notes soient remplacées par une « pulsation » et que la structure de la chanson soit indiquée par les accords au-dessus des mesures. Ceci peut être utile lorsque l'on crée ou retranscrit la structure d'une chanson, ainsi que pour donner au guitariste et musiciens de jazz une pseudo partition.

```
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'
}
```



Séparation entre altérations annulées et nouvelle armure

Les altérations supprimées lors d'un changement de tonalité sont par défaut accolées à la nouvelle armure. Ce comportement peut s'adapter grâce à la propriété `break-align-orders` de l'objet `BreakAlignment`.

Si l'on se réfère à la référence des propriétés internes à propos de l'objet Section "break-alignment-interface" dans *Référence des propriétés internes*, l'ordre du deuxième élément est :

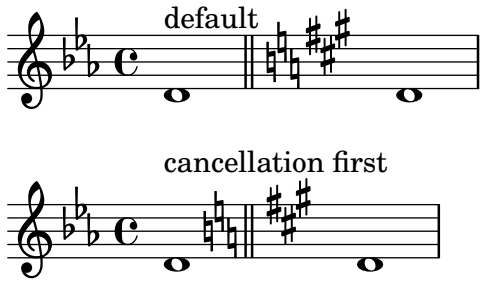
```
ambitus
breathing-sign
clef
cue-clef
cue-end-clef
custos
key-cancellation
key-signature
left-edge
signum-repetitionis
staff-bar
staff-ellipsis
time-signature
```

La consultation de cette liste nous apprend qu'il faut déplacer `key-cancellation` pour le mettre avant `staff-bar`. Ceci se réalise à l'aide de la fonction `\breakAlignInsert`.

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

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

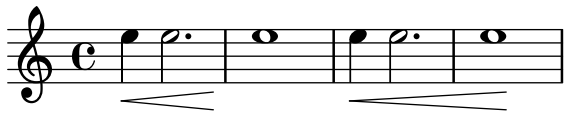
{ <>^\markup "cancellation first"
  \breakAlignInsert key-cancellation before staff-bar
  \music }
```



Soufflets et barres de mesure

En principe, un soufflet – (de)crescendo imprimé sous forme graphique – commence au bord gauche de la note de départ, et se termine au bord droit de la note d'arrivée. Cependant, si la note d'arrivée est sur un premier temps, le soufflet s'arrêtera au niveau de la barre de mesure qui la précède. Ce comportement peut être annulé en assignant *faux* (`#f`) à la propriété `to-barline`.

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



Séparation visuelle entre les systèmes

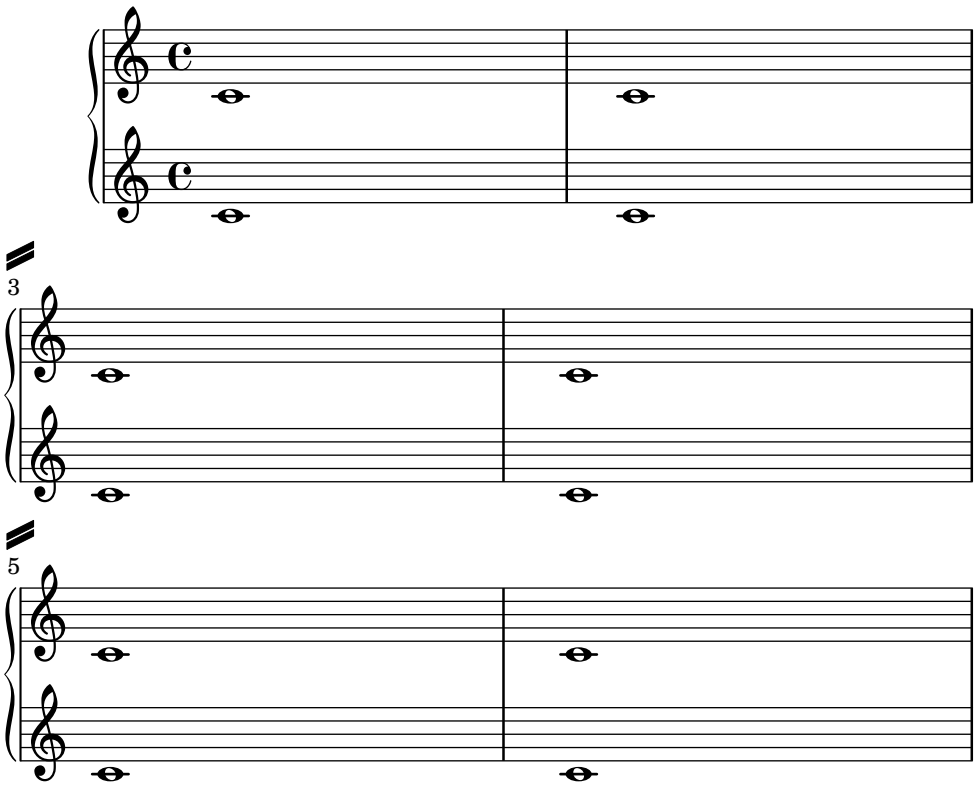
La séparation entre deux systèmes consécutifs peut être mise en évidence par n'importe quel *markup*. LilyPond dispose à cet effet d'une double oblique inversée : `\slashSeparator`.

```
#(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
    >>
  }
}
```



Galbe individuel des liaisons d'un accord

Le galbe des liaisons de prolongation d'un accord peuvent adopter des galbes différents comme illustré ci-dessous.

Dans la première ligne, on voit que toutes les notes de l'accord bénéficient d'une liaison.

Modifier ces liaisons à l'aide de `\shape` est sans effet dans la mesure où `TieColumn` les positionne de son propre chef, ignorant plus ou moins ce que contient `\shape`. On pourrait s'en sortir en affectant `#t` à `positioning-done`, mais `positioning-done` est une propriété interne et l'activer revient à dire : tout le positionnement est terminé, il n'y a plus rien à faire. La deuxième ligne illustre le fait que le positionnement n'est pas terminé : toutes les liaisons sont par dessous, et leur épaisseur n'est pas optimale.

Le fait d'indiquer explicitement le positionnement des liaisons permet d'utiliser `\shape` pour chacune des liaisons (4e ligne) et fonctionne même dans le cas d'un saut de ligne et avec la propriété `tieWaitForNote`.

```
{
  \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
}

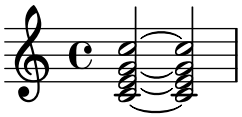
{
  \textMark \markup {
    It also works with the \typewriter tieWaitForNote property. }
  \set tieWaitForNote = ##t
  c'4-\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
  e'-\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
  g'-\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
  c''-\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
  \once \override TieColumn.positioning-done = ##t
```

```
  <c' e' g' c''>1
}

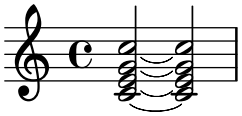
\layout {
  indent = 0
  \context {
    \Score
    \override TextMark.padding = #4
    \override TextMark.break-align-symbols = #'(left-edge)
  }
}

\paper {
  score-system-spacing.padding = 3
}
```

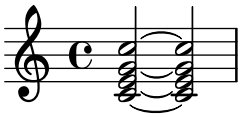
Chords can be tied note by note.



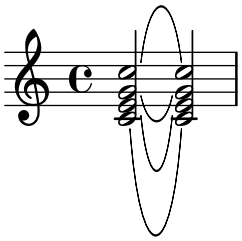
Modifying those ties with `\shape` does not succeed, because `TieColumn` positions them on its own behalf, ignoring `\shape` input more or less. You may circumvent this by setting `positioning-done` to `#t` – alas, `positioning-done` is an internal property, and setting it to `#t` means: all positioning 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.



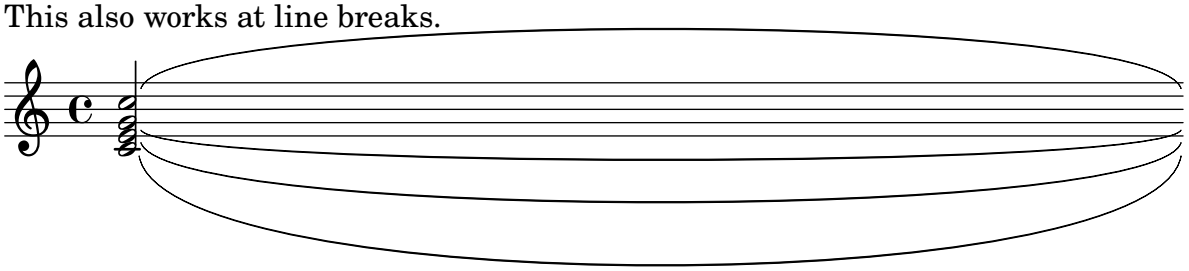
To fix that, enter ties with explicit direction modifiers.

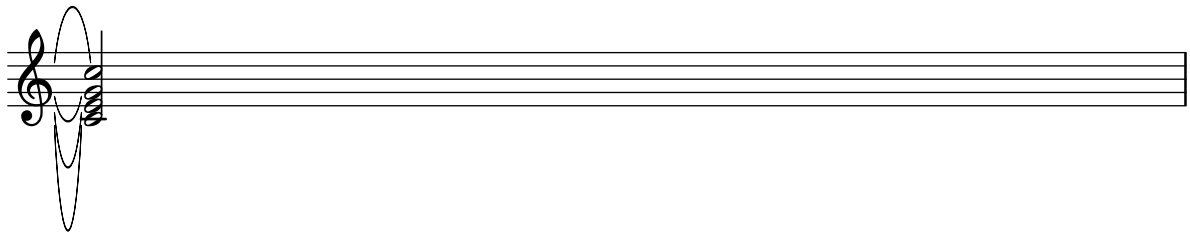


Now you can use `\shape` for each tie as usual.

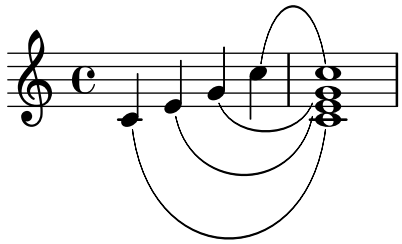


This also works at line breaks.





It also works with the `tieWaitForNote` property.



Impression d'une même articulation des deux côtés d'une note ou d'un accord

LilyPond ne permet pas, par défaut, qu'une même articulation (accent, flageolet, point d'orgue, etc.) se retrouve à la fois au-dessus et au-dessous d'une note. Par exemple, `c4_\fermata^\fermata` ne donnera qu'un seul point d'orgue en dessous du do ; celui du dessus sera tout bonnement ignoré.

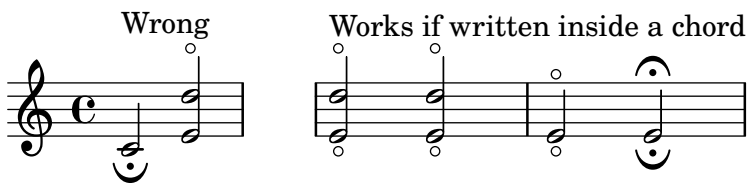
On peut néanmoins accoler des scripts, tels des doigtés, à l'intérieur d'un accord ; il peut donc y avoir autant d'articulations que de besoin, ce qui, par voie de conséquence, permet de s'affranchir de la présence des hampes et de positionner l'articulation relativement à la tête de note comme dans le cas du flageolet ci-dessous. L'imitation du traitement d'un script externe à un accord requérant un `add-stem-support` demande de libeller la note comme étant un accord et d'ajouter les articulations au sein de la construction `<...>`.

Un simple amendement permettra de rectifier le positionnement habituel en surplomb : `<c-\tweak direction #DOWN-\fermata^\fermata>`

```
\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
}
```
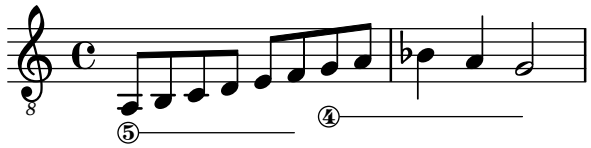


Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
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
}
```
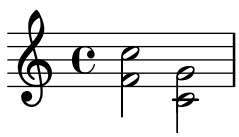


Suppression des avertissements de chevauchement

If notes from two voices with stems in the same direction are placed at the same position, and both voices have no shift or the same shift specified, the error message « warning: ignoring too many clashing note columns » appears when compiling the LilyPond file. This message can be suppressed by setting the `ignore-collision` property of the `NoteColumn` object to `#t`. Please note that this does not just suppress warnings but stops LilyPond trying to resolve collisions at all and so may have unintended results unless used with care.

```
ignore = \override NoteColumn.ignore-collision = ##t
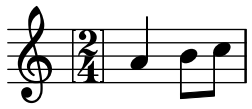
\relative c' {
  \new Staff <<
    \new Voice { \ignore \stemDown f2 g }
    \new Voice { c2 \stemDown c, }
  >>
}
```



Métrique entre crochets

La métrique peut être mise entre crochets.
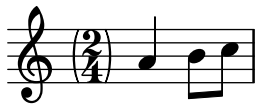
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Métrique entre parenthèses

Une métrique peut être mise entre parenthèses.
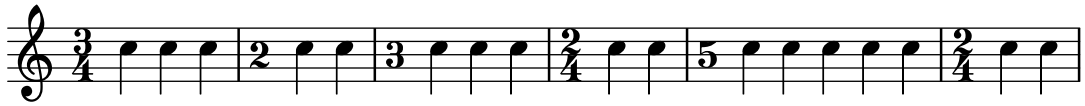
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Affichage seulement du numérateur d'une métrique (au lieu d'une fraction)

La métrique est parfois indiquée non pas par une fraction (par ex. 7/4) mais simplement par son numérateur (le chiffre 7 dans ce cas). L'instruction `\override Staff.TimeSignature.style = #'single-number` permet de déroger au style par défaut de manière permanente – un `\revert Staff.TimeSignature.style` annulera ces modifications. Lorsque cette métrique sous la forme d'un seul chiffre ne se présente qu'une seule fois, il suffit de recourir à un `\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
}
```



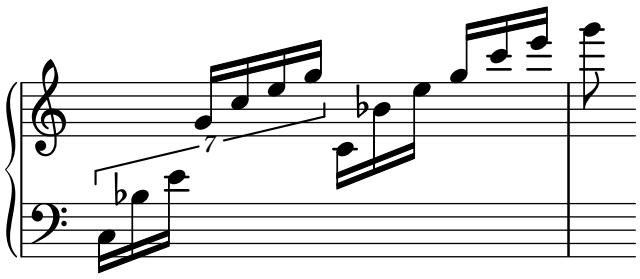
Crochet de n-olet et changement de portée

Voici comment obtenir un crochet de n-olet qui débute dans une portée et se termine dans l'autre.

```
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
>>
```



Affinage des propriétés d'une clef

Modifier le glyphe, la position de la clef ou son octaviation ne changeront pas la position des notes ; il faut pour y parvenir modifier aussi la position du do médium. La redéfinition préalable de `middleCClefPosition` permet de placer l'armure sur les bonnes lignes. Le positionnement est relatif à la ligne médiane, un nombre positif faisant monter, un nombre négatif abaissant.

Par exemple, la commande `\clef "treble_8"` équivaut à définir les propriétés de contexte `clefGlyph`, `clefPosition` – qui contrôle la position verticale de la clef –, `middleCPosition` et `clefOctavation`. Une nouvelle clef apparaîtra dès lors que l'une de ces propriétés, à l'exception de `middleCPosition`, aura été modifiée.

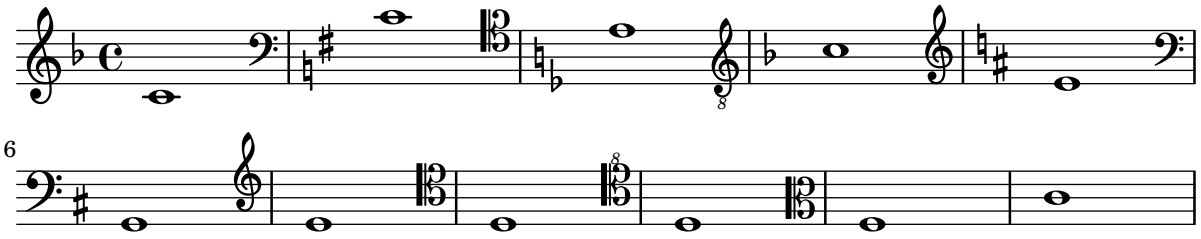
Les exemples qui suivent illustrent les différentes possibilités de définir ces propriétés manuellement. Sur la première ligne, la position relative des notes par rapport aux clefs est préservée, ce qui n'est pas le cas pour la deuxième ligne.

```
{
  % 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
}
```
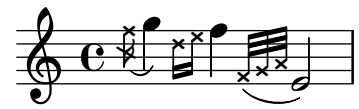


Mise en forme des notes d'ornement

Il est possible de changer globalement l'apparence des notes d'ornement dans un morceau, au moyen des fonctions `add-grace-property` et `remove-grace-property`.

Ici, par exemple, on ôte la définition de l'orientation (propriété `direction`) des objets `Stem` pour toutes les petites notes, afin que les hampes ne soient pas toujours orientées vers le haut, et on leur préfère des têtes en forme de croix.

```
\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
    }
  }
}
```



Crochets de style alternatif

Une dérogation à la propriété `stencil` de l'objet `Flag` permet aux croches et notes de durée inférieure d'adopter une autre forme de crochet. Sont disponibles les variantes `modern-straight-flag`, `old-straight-flag` et `flat-flag`. Un `\revert` permet de retrouver l'allure par défaut.

Des crochets empilés, autrement dit à l'espacement resserré, s'obtiennent à l'aide de l'instruction `\flagStyleStacked`, qui s'annule par un `\flagStyleDefault`.

Une dérogation au stencil de `Flag` ne modifie en rien le positionnement vertical individuel des crochets. Ceci s'observe avec des crochets rectilignes : LilyPond n'ajuste pas dynamiquement l'écart entre les crochets les uns par rapport aux autres de la même manière que pour les ligatures. L'une des solutions pour harmoniser l'apparence consiste à remplacer les crochets par des demi-ligatures comme indiqué dans la deuxième portée . ceci ne peut toutefois se faire automatiquement. Dans le code de cet extrait, ces demi-ligatures se mentionnent à l'aide du préfixe `@`, comme par exemple `@c8`.

Il est important de noter que des demi-ligatures *ne sont pas* des objets `Flag`, ce qui signifie que des modifications apportées aux objets `Flag` n'auront aucun effet sur elles (il faut utiliser des propriétés de `Beam`), et les propriétés de leur objet `Stem` associé adoptera le même comportement qu'avec des ligatures.

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

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

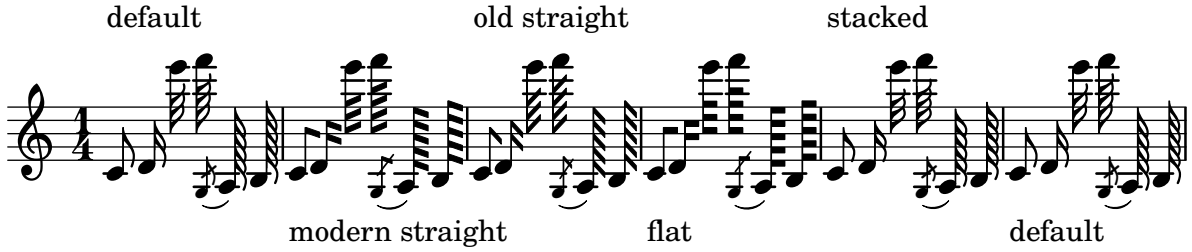
\relative c' {
  \override TextScript.staff-padding = 6
  \time 1/4
    <>^"default" \testnotes
  \override Flag.stencil = #modern-straight-flag
    <>_"modern straight" \testnotes
  \override Flag.stencil = #old-straight-flag
    <>^"old straight" \testnotes
  \override Flag.stencil = #flat-flag
  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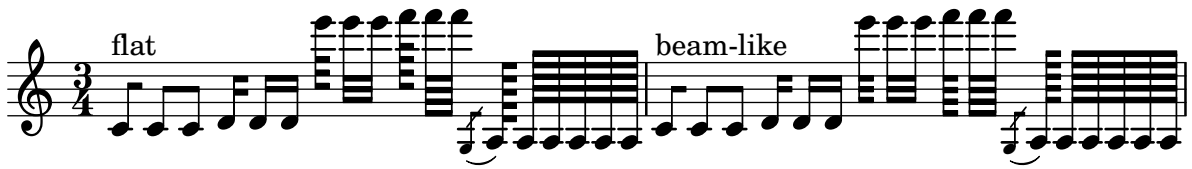
  \flagStyleStacked
    <>^"stacked" \testnotes
  \flagStyleDefault
    <>_"default" \testnotes
}

\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
  }
}
```





Utilisation de `ly:grob-object` pour accéder aux *grobs* avec `\tweak`

Certains objets graphiques ne sont accessibles que par le biais d'un *callback* à partir d'un autre `grob`. Ils sont normalement listés dans les « *layout objects* » au sein de la section « Propriétés internes » d'une *grob interface*. La fonction `ly:grob-object` permet d'accéder à ces objets.

Voici plusieurs moyens d'accéder aux objets par un *callback* sur `NoteHead`. D'autres biais sont naturellement possibles ; `NoteHead` a cependant l'avantage incontestable d'être utilisé implicitement par la commande `\tweak`.

Voici ce qui sera émis dans la console suite à la fonction `display-grobs` ici construite.

```
--------------------
#<Grob Accidental >
()
#<Grob Stem >
```

Cette fonction n'est probablement pas très utile. Elle indique toutefois qu'il est tout à fait possible d'accéder aux objets.

```
#(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)))
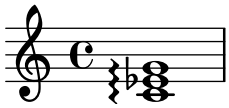
#(define (notehead-get-notecolumn notehead)
   ;; notehead is grob
   (ly:grob-parent notehead X))

#(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' {
```

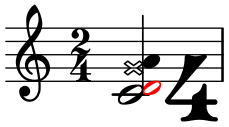
```
  %% display grobs for each note head:
  %\override NoteHead.before-line-breaking = #display-grobs
  <c
  %% or just for one:
  \tweak before-line-breaking #display-grobs
  es
  g>1\arpeggio
}
```



Utilisation de \tweak pour retoucher des objets particuliers

La commande \tweak permet de retoucher directement n'importe quel objet graphique. En voici quelques exemples :

```
\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
}
```



Alignement vertical des nuances et indications textuelles

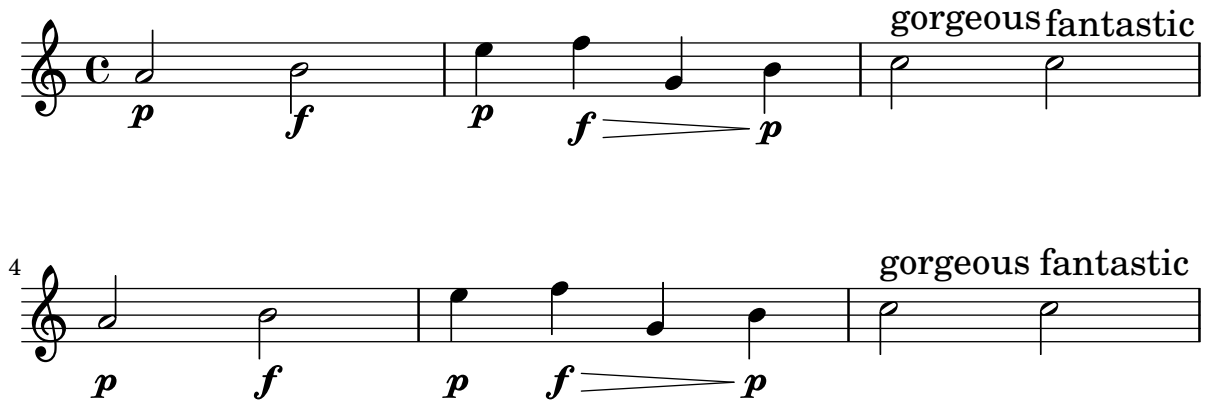
Tous les objets DynamicLineSpanner (soufflets ou nuances textuelles) viennent s'aligner sur une ligne de référence placée, par rapport à la portée, à au moins la valeur de la propriété staff-padding sauf lorsque d'autres éléments de notation les en éloignent plus. Les nuances seront centrés sur une même ligne dès lors que staff-padding aura été défini à une valeur suffisante.

C'est le même principe – en combinaison avec \textLengthOn – qui sert à aligner les indications textuelles sur une ligne de référence.

```
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
}
```

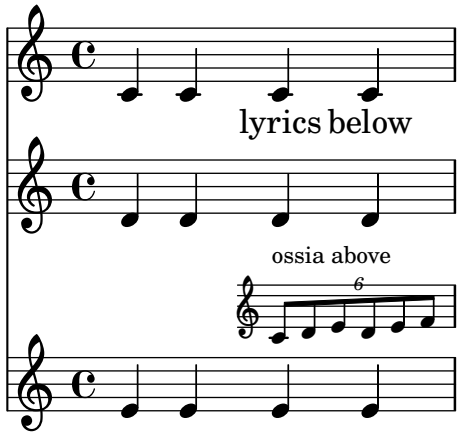


Positionnement d'une ossia et des paroles

Cet exemple illustre la manière de positionner une portée d'ossia et des paroles à l'aide des propriétés de contexte `alignBelowContext` et `alignAboveContext`.

```
\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]
        }
      }
    >>
  }
>>
```



Alignement vertical des numéros de couplet sur différentes portées

Il peut arriver que les numéros de couplet ne s'alignent pas verticalement s'ils sont attachés à différentes portées. Une adaptation de la propriété `self-alignment-X` de l'objet graphique `LyricText` permet d'y remédier.

```
\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
  }
>>

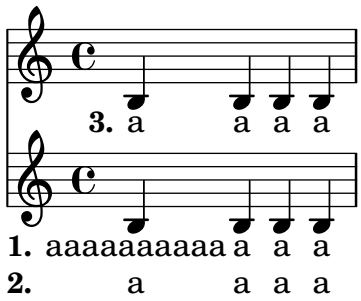
\markup \vspace #1
\markup {
  using \typewriter "self-alignment-X = #LEFT" }

<<
  \new Staff { b b b b }
  \new Lyrics \lyricmode {
    \stanza "3."
    a a a a
  }

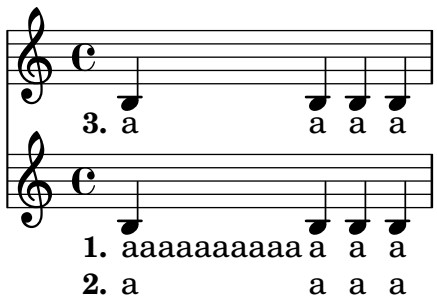
  \new Staff { b b b b }
```

```
  \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`

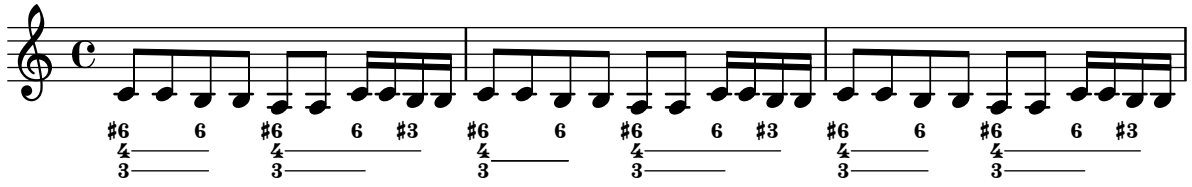


Prolongateur commun de basse figurée

L'activation de la propriété `useBassFigureExtenders` permet d'afficher des lignes de prolongation pour les chiffres qui se répètent. Deux chiffres prolongés sur la même durée se verront affublés d'un unique prolongateur, verticalement centré entre eux, dès lors que la propriété `figuredBassCenterContinuations` aura elle aussi été activée.

```
<<
  \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
  }
>>
```

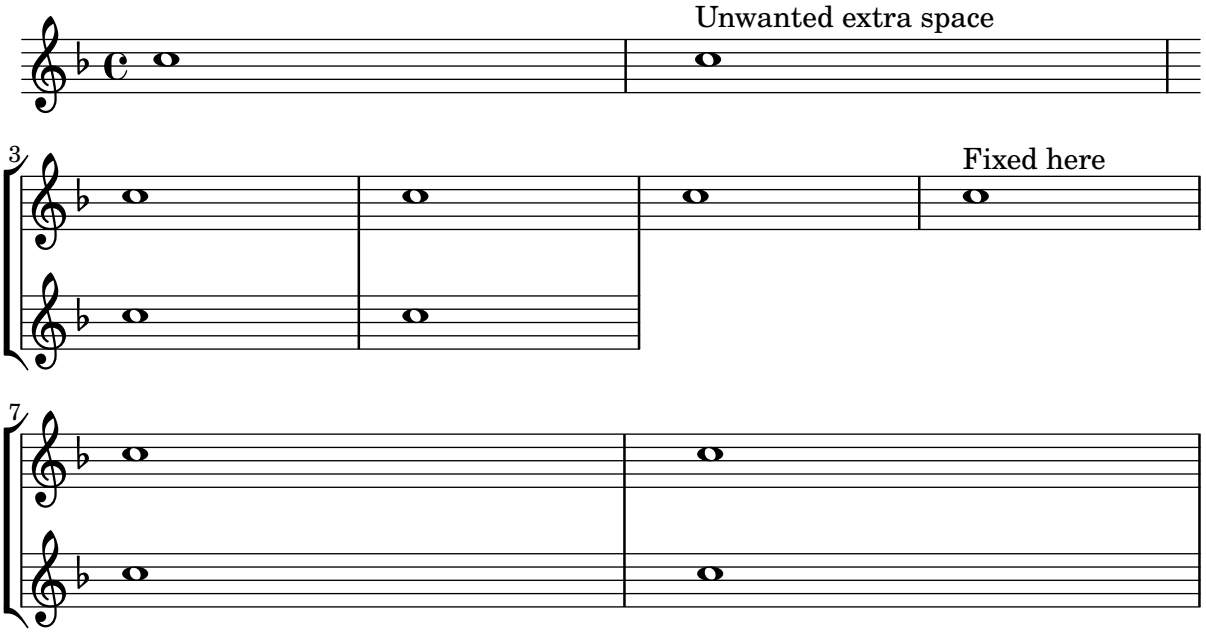


39 Workaround

Ajout d'une portée supplémentaire après un saut de ligne

Lorsqu'une nouvelle portée vient s'ajouter après un saut de ligne, LilyPond préserve un espace juste avant le saut de ligne – pour un éventuel changement d'armure qui, quoi qu'il en soit, n'est pas imprimé. L'astuce consiste alors, comme indiqué dans l'exemple suivant, à ajuster le `explicitKeySignatureVisibility` de l'objet graphique `Staff`.

```
\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
       }
    >>
  }
}
```



Appoggiature avant une barre de mesure

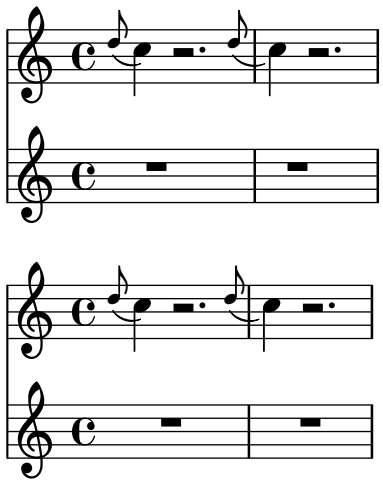
Par défaut, appogiatures et autres notes d'ornement sur le premier temps d'une mesure s'impriment après la barre. Dans le cas d'une ligne unique, elles peuvent toutefois précéder la barre grâce à l'insertion d'une barre invisible puis d'une visible.

Néanmoins, dans le cas d'un système à plusieurs portées, l'ajout d'une barre invisble gène le bonn positionnement des silences de mesure ; ils ne sont plus vraiment centrés, mais légèrement décalés sur la gauche. Une meilleure solution en pareil cas est d'utiliser la commande `\afterGrace` tout en jouant sur le réglage de sa propriété `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 }
>>
```



Rupture d'alignement horizontal des nuances graphiques et textuelles

Par défaut, LilyPond utilise des objets `DynamicLineSpanner` pour aligner horizontalement une succession d'objets nuance tels que soufflets et indications textuelles, et quel que soit leur positionnement par rapport à la portée. Ceci peut, parfois, ne pas être désirable et une rupture de l'étendue de l'alignement n'est pas possible comme en matière d'alignement vertical (voir l'exemple « Rupture d'alignement vertical des nuances graphiques et textuelles »).

Deux solutions permettent de pallier ce désagrément.

- Modifier la propriété `shorten-pair` de l'objet `Hairpin` pour compenser le décalage qui a été appliqué au soufflet.
- Placer les deux objets de nuance dans des voix différentes.

Le code ci-dessous illustre ces deux possibilités.

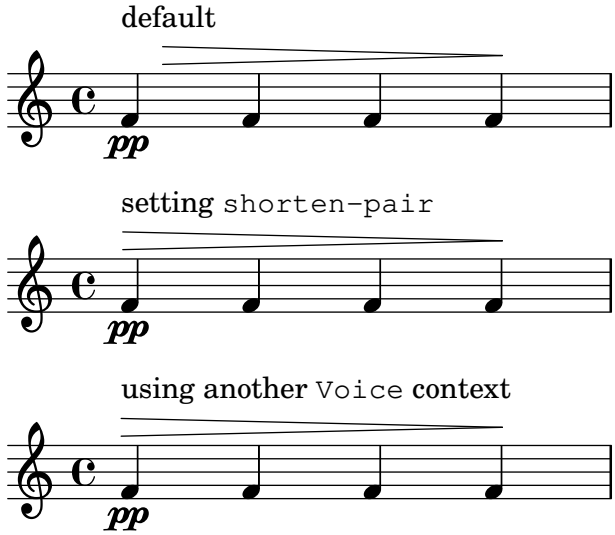
```
{
  <>^"default"
  f'_\pp ^\> f' f' f'\!
}

{
  <>^\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
  }
}
```



Rupture d'alignement vertical des nuances graphiques et textuelles

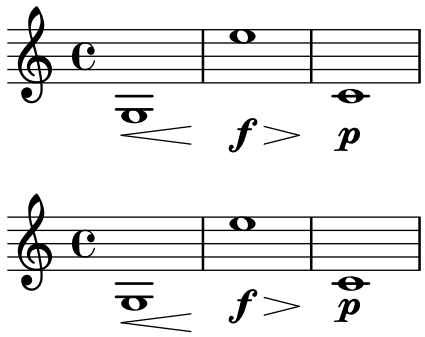
Par défaut, LilyPond utilise des objets `DynamicLineSpanner` pour aligner verticalement une succession d'objets nuance tels que soufflets et indications textuelles. Ceci peut, parfois, ne pas

être désirable. L'insertion d'un \breakDynamicSpan, qui rompt prématurément l'étendue de l'alignement, permet de pallier ce désagrément.

Voir aussi l'exemple « Rupture d'alignement horizontal des nuances graphiques et textuelles ».

```
{ g1\< |
  e''\f\> |
  c'\p }

{ g1\< |
  e''\breakDynamicSpan\f\> |
  c'\p }
```



Modification de la métrique au sein d'un fragment polymétrique à l'aide de \scaleDurations

Polymétrie sans alignement des mesures

La prise en charge de contextes aux métriques indépendantes demande de supprimer le Timing_translator du contexte Score, conjointement à la création d'un contexte TimingStaffGroup disposant du Timing_translator. Ceci fera de Timing un alias de TimingStaffGroup, permettant ainsi aux commandes \time de s'appliquer au sein du TimingStaffGroup où elles apparaissent.

Contrairement à la commande native de LilyPond \enablePerStaffTiming, cette approche requiert la création explicite de contextes TimingStaffGroup mais, par contre, autorise la création de multiples contextes Staff qui suivront de concert la mesure définie dans le TimingStaffGroup qui les englobe.

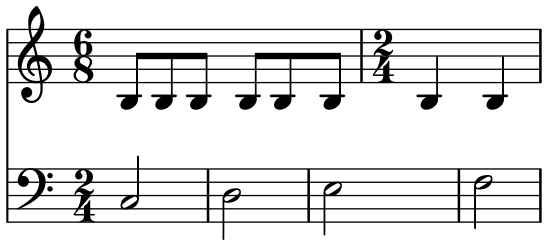
Échelonnement local des métriques

La commande \time, qui ne peut être échelonné, établit une mesure de la longueur désirée dans Timing, autrement dit dans TimingStaffGroup. Dans le code ci-dessous, toutes les portées incluses dans un TimingStaffGroup utilisent une métrique échelonnée, de sorte que, peu importe la métrique, elle tiendra dans la longueur de mesure désirée. Si l'un des contextes inclus n'avait pas de métrique échelonnée, le choix de la métrique devant être fournie à Timing aurait toute son importance.

L'utilisation de la commande \polymetric \time permet de définir les propriétés d'échelonnement de la métrique dans le contexte Timing, et l'utilisation de \scaleDurations échelonnera le mètre et les notes afin qu'elles tiennent dans la mesure.

```
\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
    }
  >>
>>
```



Accord distribué et problème de hampe – solution

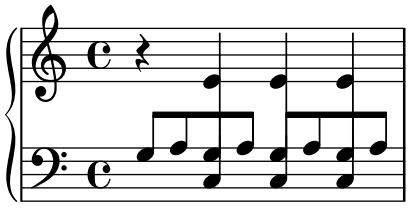
Il est parfois préférable d'utiliser les hampes de l'autre portée pour créer des accords distribués, afin de tricher avec le détecteur de collision des ligatures de LilyPond. Dans l'exemple suivant, le fait de partir des hampes de la portée inférieure aurait rendu nécessaire l'adaptation du détecteur de collision des ligatures, par une clause

```
\override Staff.Beam.collision-voice-only = ##t
```

afin que LilyPond ne déplace pas les ligatures.

```
\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
  }
>>
```
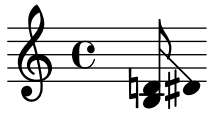


Impression d'accords complexes

Voici comment obtenir l'impression d'un accord au sein duquel une même note est jouée deux fois avec des altérations différentes.

```
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
}
```



Rappel du glissando à l'occasion d'une alternative

Un glissando qui se prolonge sur plusieurs sections `\alternative` peut se rappeler à l'aide d'une note d'ornement supplémentaire et masquée, à laquelle sera attaché le départ du glissando, ce dans chaque bloc `\alternative`. Cette note d'ornement devrait avoir la même hauteur que la note où commençait le glissando originel. Ceci est géré par une fonction musicale qui prendra en argument la hauteur de la note d'ornement.

Dans le cadre d'une musique polyphonique, il ne faudra pas oublier d'ajouter une note d'ornement dans toutes les autres voix afin de préserver la synchronisation.

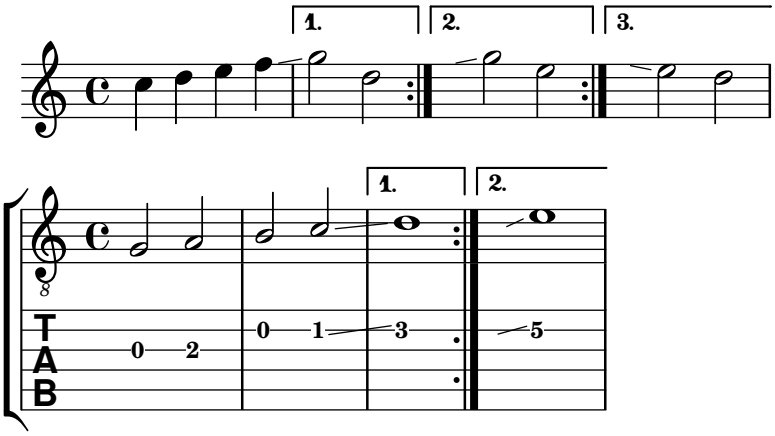
```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \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 }
    >>
  >>
}
```



Adaptation de la largeur de mesure selon le `MetronomeMark`

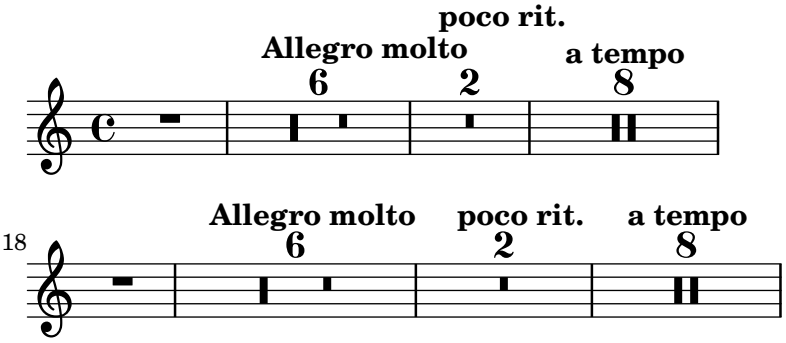
Par défaut, les indications métronomiques n'influencent en rien l'espacement horizontal.

Une simple dérogation, comme dans la deuxième partie de cet exemple, fournit la solution.

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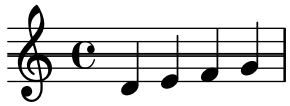
\layout {
  ragged-right = ##t
}
```



Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane, ou bien pour mettre en exergue la ligne portant la clef de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doit être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```
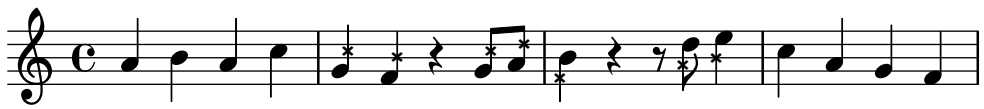


Adjonction d'une croix sur la hampe des notes d'un fragment parlé

Voici comment ajouter une croix aux hampes. Le début du fragment parlé est stipulé par une commande \speakOn, et la fin par une commande \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
  }
}
```



Positionnement des notes d'ornement avec espace flottant

Lorsque la propriété `strict-grace-spacing` est activée, l'espacement des notes d'ornement se fera de manière « élastique ». Autrement dit, elles seront décollées de leur note de rattachement : LilyPond commence par espacer les notes normales, puis les ornements sont placés à la gauche de leur note de rattachement.

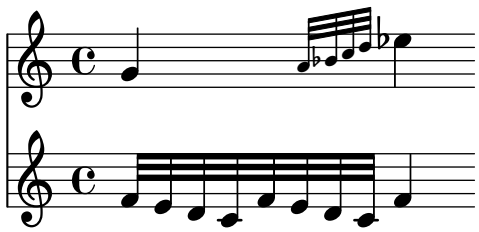
Cependant, en raison du ticket 6876 (`https://gitlab.com/lilypond/lilypond/-/issues/6876`), les altérations accidentelles sont ignorées lorsque cette propriété est activée. Le code ci-dessous propose une solution de contournement à ce problème.

Autre effet indésirable de cette propriété, LilyPond ne vérifie pas qu'il y a suffisamment d'espace pour les notes d'ornement (ceci fait l'objet du ticket 2630 (`https://gitlab.com/lilypond/lilypond/-/issues/2630`)). Il faut donc s'assurer d'avoir suffisamment d'espace disponible en recourant conjointement `\newSpacingSection` et une valeur appropriée pour le `base-shortest-duration` de l'objet graphique `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 }
>>

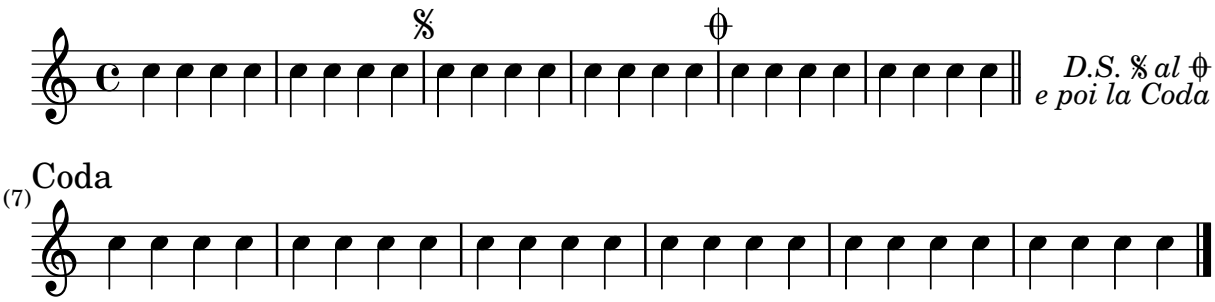
\layout {
  \context {
    \Score
    \override SpacingSpanner.strict-grace-spacing = ##t
  }
}
```



Positionnement des segno et coda (avec saut de ligne)

Le code ci-dessous permet d'adjoindre à un signe *segno* un texte *D.S. al Coda*, là où se trouverait normalement un bout de portée. La *coda* entamera une nouvelle ligne. Une variante, indiquée ici même, permet de laisser la *coda* sur la même ligne.

```
\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
}
```



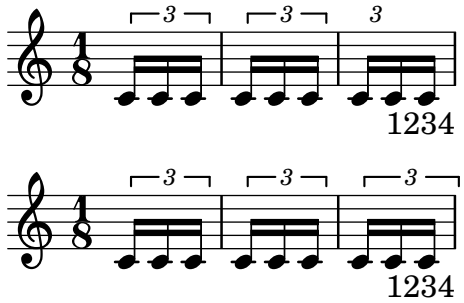
Préservation de l'indication de n-olet lors d'un repère final

En raison de ce qui est rapporté dans le ticket 2362 (`https://gitlab.com/lilypond/lilypond/-/issues/2362`), l'ajout d'une marque-repère en fin de pièce peut entraîner la perte de la dernière indication de n-olet. La désactivation de `TupletBracket.full-length-to-extent` pallie ce problème.

```
\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

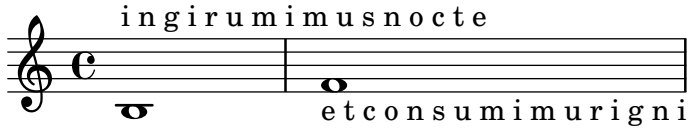
  \time 1/8
  \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' c' }
  \tweak direction #DOWN \textEndMark "1234"
}
```



Impression de texte de droite à gauche

Du texte, inclus dans un objet *markup*, peut s'imprimer de droite à gauche, comme illustré ci-dessous.

```
{
  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 }
  }
}
```
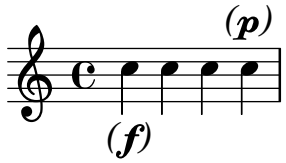


Indications de nuance entre parenthèses en italique

Le moyen le plus simple pour ajouter des parenthèses à une indication de nuance consiste à utiliser la commande `\parenthesize` et et d'ajuster la taille de la fonte, comme par exemple `{ c'4\tweak Parentheses.font-size 2 \parenthesize \f }`. Toutefois, ces parenthèses se présentent droit alors que les indications de nuance apparaissent normalement en italique. Le code ci-dessous offre une solution pour présenter les parenthèses elles aussi en italique.

```
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
}
```



Exemple pour tambourin

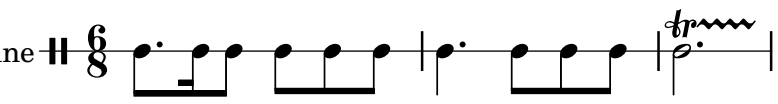
Voici une partie de tambourin, saisi par « tamb ».

```
#(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 |
}
```



Transposition et réduction du nombre d'altérations accidentelles

Cet exemple, grâce à un peu de code Scheme, donne la priorité aux enharmoniques afin de limiter le nombre d'altérations supplémentaires. La règle applicable est :

- Les altérations doubles sont supprimées
- Si dièse -> Do
- Mi dièse -> Fa
- Do bémol -> Si

- Fa bémol -> Mi

Cette façon de procéder aboutit à plus d'enharmoniques naturelles.
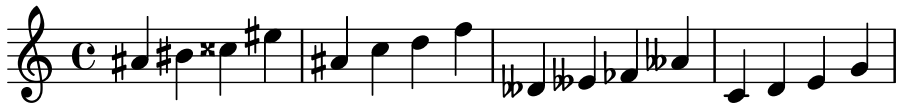
```
#(define (naturalize-pitch p)
   (let ((o (ly:pitch-octave p))
         ;; `ly:pitch-alteration` returns quarter tone steps.
         (a (* 4 (ly:pitch-alteration p)))
         (n (ly:pitch-notename p)))
     (cond
      ((and (> a 1)
            (or (eqv? n 6) (eqv? n 2)))
       (set! a (- a 2))
       (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 }
}
```



Expansion de répétition de trémolo

Actuellement, saisir *note*:*durée* est en quelque sorte un raccourci de \repeat tremolo, mais n'est pas expansé par une directive \unfoldRepeats (ceci fait l'objet du ticket 6145 (https://gitlab.com/lilypond/lilypond/-/issues/6145)). La fonction ci-dessous présentée constitue un contournement à cette faiblesse.
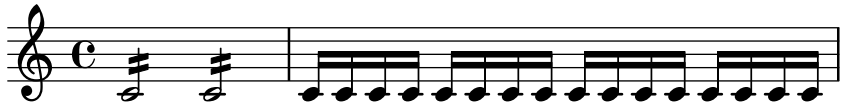
```
fixTremolos =
#(define-music-function (music) (ly:music?)
   (music-map
    (lambda (m)
      (let ((event (any (lambda (a)
                          (and (music-is-of-type? a 'tremolo-event)
                               a))
                        (ly:music-property m 'articulations))))
        (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
}
```



Recours à une voix supplémentaire pour gérer les sauts

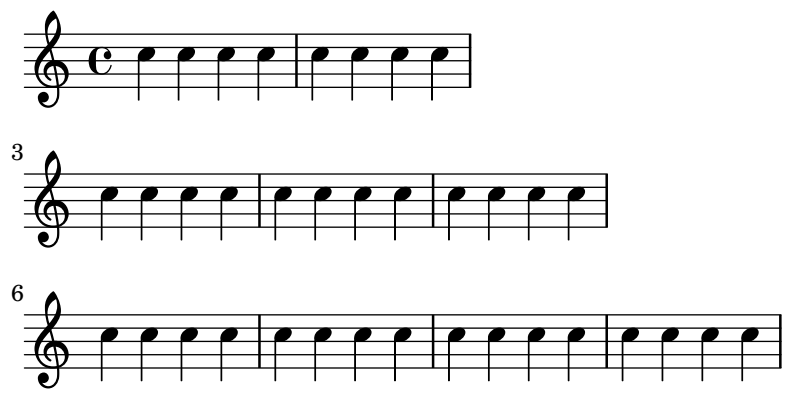
Il est souvent plus pratique de séparer ce qui est purement musical et les informations concernant les sauts de ligne ou de page, en créant une voix supplémentaire dédiée. Cette voix spécifique ne contiendra que des blancs – des silences invisibles `\skip` –, des `\break`, des `\pageBreak` et autres informations concernant les ruptures.
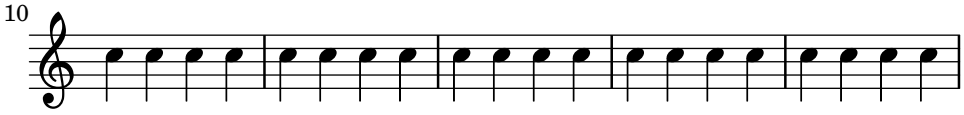
Cette manière de procéder est tout à fait indiquée lorsque vous ajustez les `line-break-system-details` et autres propriétés fort intéressantes de l'objet `NonMusicalPaperColumn`.

```
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
}
```





Alignement vertical des nuances et indications textuelles

Tous les objets `DynamicLineSpanner` (soufflets ou nuances textuelles) viennent s'aligner sur une ligne de référence placée, par rapport à la portée, à au moins la valeur de la propriété `staff-padding` sauf lorsque d'autres éléments de notation les en éloignent plus. Les nuances seront centrés sur une même ligne dès lors que `staff-padding` aura été défini à une valeur suffisante.

C'est le même principe – en combinaison avec `\textLengthOn` – qui sert à aligner les indications textuelles sur une ligne de référence.

```
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
}
```

