http://dev.music.free.fr/

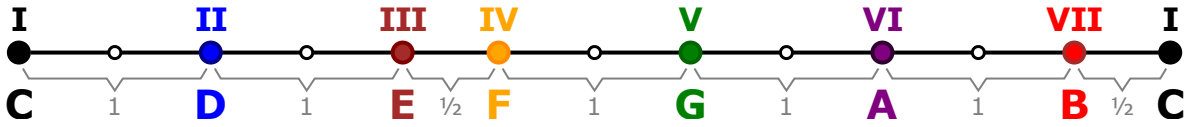
Some text and images

> When it's green, we apply the rule.

Intervals coloring:

- **P1=unison**, **P8=octave**.
- m2=minor second, **M2=major second**, A2=augmented second.
- D3=diminished third, m3=minor third, **M3=major third**, A3=augmented third.
- D4=diminished fourth, **P4=perfect fourth**, A4=augmented fourth.
- D5=diminished fifth, **P5=perfect fifth**, A5=augmented fifth.
- D6=diminished sixth, m6=minor sixth, **M6=major sixth**, A6=augmented sixth.
- D7=diminished seven, m7=minor seven, **M7=major seven**.

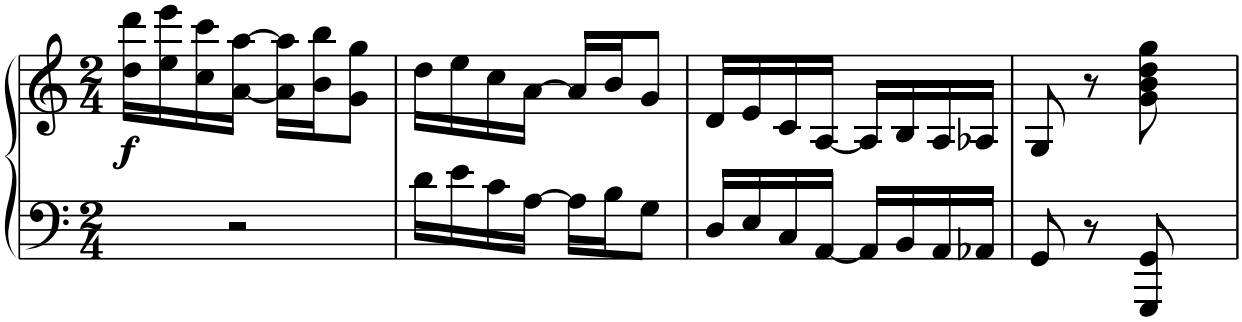
Intervals



Piano keyboards

FM7 : F, A, C, E

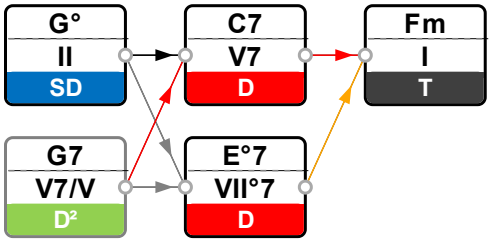
Playable scores



Harmonic sequences

Cm	Fm7	G7	Cm
I	IV	V7	I
T	SD	D	T
Ab	Db	Eb	Ab
bVI	bII	bIII	bVI
SD	SD	T	SD

Harmonic diagams



Musical content for web's pages

CSS ressources

File	Description
book-A4.min.css	Handles web page with the A4 format.
music.min.css	Handles musical components.

JavaScript modules

Module	Description
$HARMONY	Musical harmony functions.
$MUSIC_UI	Tones, tonalities and chord signatures selections.
$ABC_UI	Playable scores in abc format, for html pages.
$KEYBOARD_UI	Piano keyboard with keys selection.
$INTERVALS_UI	Musical intervals.
$SEQUENCE_UI	Harmonic sequences.
$DIAGRAM_UI	Harmonic diagrams.

Documentation download

Samples

- Chords, libraries validation tests, dual tests.
- Major gammuts, Minor natural gammuts, Minor harmonic gammuts, Minor melodic gammuts, Modes harmonisation.
- $ABC_UI demo.

Illustrations

- Cours d'harmonie de Gradus ad Parnassum.
- Cours d'harmonie de Michel Baron.
- La composition au piano de Christophe Poudras.

Musical text

To have printable page with A4 format, include **book-A4.min.css** in header of HTML page.

Files to include in HTML file header

```
<link rel="stylesheet" href="css/book-A4.min.css" />
<link rel="stylesheet" href="css/music.min.css" />
```

```
<h1>Header1</h1>
```

Header1

```
<h2>Header2</h2>
```

Header2

```
<h3>Header3</h3>
```

Header3

```
<h4>Header4</h4>
```

Header4

```
<p>paragraph</p>
```

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante.

```
<ul>non ordered list</ul>
```

- Morbi in sem quis dui placerat ornare. Pellentesque odio nisi, euismod in, pharetra a, ultricies in, diam. Sed arcu. Cras consequat.
- Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat.

```
<ol>ordered list</ol>
```

1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
2. Aliquam tincidunt mauris eu risus.

Rules

```
<div class="neutral">With gray, it's neutral.</div>
```

With gray, it's neutral.

```
<div class="definition">Use blue for definition.</div>
```

Use blue for definition.

```
<div class="warning">Use yellow for warning.</div>
```

Use yellow for warning.

```
<div class="rule use">Use green for rule to use.</div>
```

Use green for rule to use.

```
<div class="rule avoid">Use orange for rule to avoid.</div>
```

Use orange for rule to avoid.

```
<div class="rule not">Use red for
FORBIDDEN rule.</div>
```

Use red for
FORBIDDEN rule.

Intervals coloring

```
<span class="T3">diminished third</span>,
<span class="T3">minor third</span>,
<span class="T3M">major third</span>,
<span class="T3">augmented third</span>.
```

- **P1**=**unison**, **P8**=**octave**.
- m2=minor second, **M2**=**major second**, A2=augmented second.
- D3=diminished third, m3=minor third, **M3**=**major third**, A3=augmented third.
- D4=diminished fourth, **P4**=**perfect fourth**, A4=augmented fourth.
- D5=diminished fifth, **P5**=**perfect fifth**, A5=augmented fifth.
- D6=diminished sixth, m6=minor sixth, **M6**=**major sixth**, A6=augmented sixth.
- D7=diminished seven, m7=minor seven, **M7**=**major seven**.

$HARMONY

This JavaScript module handles musical harmonic functions.
Module file is: harmony-1.0.0.min.js.

$HARMONY Methods

Notes

Method	Description
getRelativeNote	Renvoie, à partir de la note donnée, la note à distance de l'intervalle montant donné.
getRelativeNotes	Renvoie, à partir de la note donnée, la liste des notes à distance des l'intervalles montants donnés.
getEnharmonic	Renvoie la note enharmonique de la note donnée.
getLatinNote	Renvoie le nom latin de la note.

Tones

Method	Description
getFarthestTone	Renvoie le ton le plus éloigné du cycle des quintes.
getSimpleTone	Renvoie le ton enharmonique simple.
getKeyNotes	Renvoie la liste des notes associées à une clé choisie.

Tonalities

Method	Description
getToneOfTonality	Renvoie le ton de la tonalité choisie.
getToneAndMinor	Renvoie le ton et mode majeur ou mineur.
getToneAndMode	Renvoie le ton et le mode de la tonalité choisie.
getEnharmonicTone	Renvoie le ton enharmonique en majeur et mineur de la tonalité choisie.

Harmonic functions

Method	Description
getHarmonicFunction	Renvoie la fonction harmonique d'un accord au degré donné, dans une gamme donnée.

Chords

Method	Description
getNotesFromSignature	Renvoie la liste des notes de l'accord ayant la signature donnée et la note donnée comme fondamentale.
getChordNotes	Renvoie la liste des notes de l'accord choisi.
getChordNumberOfNote	Renvoie le nombre de note d'un accord à partir de sa signature.
getChordSignatureName	Renvoie le nom d'usage d'une signature, en français.
getChordName	Renvoie le nom d'usage d'un accord, en français.
getChordIntervals	Renvoie la liste des intervalles en jeux pour une signature.
getChordFondamental	Renvoie la note fondamentale de l'accord choisi.
getChordBass	Renvoie la note à la basse de l'accord choisi.
getChordSignature	Renvoie la signature de l'accord choisi.

Modes

Method	Description
getModeIntervals	Renvoie la liste des intervalles contituant un mode.
getModeStructure	Renvoie une liste de deux strings, le nom du pentacorde suivi du nom du tétracorde, du mode sélectionné.
isModeHarmonisable	Renvoie un booléen indiquant si le mode choisi peut-être harmonisé.
getModeName	Renvoie une string du nom du mode choisi, en français.
getModeNaturalName	Renvoie une string du nom du mode *ancien* choisi.
getModeNotes	Renvoie la liste des notes d'un mode donné, pour un ton donné.
getModeSignature	Renvoie la signature de l'accord d'un degré dans un mode donné.
getModeSignatures	Renvoie la liste des signatures d'un mode donné.
getModeFunctions	Renvoie la liste des fonctions harmoniques d'un mode donné.
getModeChords	Renvoie la liste des accords d'un mode donné.

Validation

- Tests

Introduction aux concepts musicaux utilisés

Notes

On utilise les notes dans la notation anglaise, soit les lettres majuscules :
C (*do*), **D** (*ré*), **E** (*mi*), **F** (*fa*), **G** (*sol*), **A** (*la*), **B** (*si*).

Les altérations

- (rien) : sans altération pour un note naturelle,
- **#** : le dièse,
- **x** : le double dièse,
- **b** : le bémol,
- **bb** : le double bémol.

Certaines fonctions génèrent dans les cas limites des altérations triples : **X** pour le triple dièse et **bbb** pour le triple bémol. Il est alors d'usage d'utiliser les enharmoniques.

Degrees

Les degrés sont encodés en string en notation romaine : '**I**', '**II**', '**III**', '**IV**', '**V**', '**VI**' et '**VII**'.

Tones

Les tons sont encodés sous forme d'une string :
'**Cb**', '**Gb**', '**Db**', '**Ab**', '**Eb**', '**Bb**', '**F**', '**C**', '**G**', '**D**', '**A**', '**E**', '**B**', '**F#**', '**C#**', '**G#**', '**D#**', '**A#**'.

Intervals

Les intervalles sont encodés sous forme d'une string (case sensitive) :

String	Description	String	Description
P1	unison	**P8**	octave juste
m2	seconde mineure	**M7**	septième majeure
M2	seconde majeure	**m7**	septième mineure
A2	seconde augmentée	**D7**	septième diminuée
D3	tierce diminuée	**A6**	sixte augmentée
m3	tierce mineure	**M6**	sixte majeure
M3	tierce majeure	**m6**	sixte mineure
A3	tierce augmenté	**D6**	sixte diminuée
D4	quarte diminuée	**A5**	quinte augmentée
P4	quarte juste	**P5**	quinte juste
A4	quarte augmentée	**D5**	quinte diminuée
m9	neuvième mineure	**M9**	neuvième majeure
A9	neuvième augmentée	**TT**	triton
P11	onzième juste	**A11**	onzième augmentée
m13	treizième mineure	**M13**	treizième majeure

Tonalities

Les tonalités sont constitués d'un ton et d'un mode majeur ou mineur. Elles sont encodées sous forme d'une string. Le premier caractère represente le mode : **M** pour majeur et **m** pour mineur. Suit après le ton.

String	Description	String	Description
MCb	*do* bémol majeur	**mAb**	*la* bémol mineur
MGb	*sol* bémol majeur	**mEb**	*mi* bémol mineur
MDb	*ré* bémol majeur	**mBb**	*si* bémol mineur
MAb	*la* bémol majeur	**mF**	*fa* mineur
MEb	*mi* bémol majeur	**mC**	*do* mineur
MBb	*si* bémol majeur	**mG**	*sol* mineur
MF	*fa* majeur	**mD**	*ré* mineur
MC	*do* majeur	**mA**	*la* mineur
MG	*sol* majeur	**mE**	*mi* mineur
MD	*ré* majeur	**mB**	*si* mineur
MA	*la* majeur	**mF#**	*fa* dièse mineur
ME	*mi* majeur	**mC#**	*do* dièse mineur
MB	*si* majeur	**mG#**	*sol* dièse mineur
MF#	*fa* dièse majeur	**mD#**	*ré* dièse mineur
MC#	*do* dièse majeur	**mA#**	*la* dièse mineur

Chord's signatures

String	Description	String	Description
(vide), **Maj**	majeur	**m**	mineur
°, **dim**	diminué	**(#5)**, **+5**	augmenté
7	septième de dominante	**M7**, **Δ7**	majeure 7
m7	mineur 7	**mM7**	mineur, majeure 7
°7, **dim7**	diminué 7	**m7b5**, **Ø7**	demi-diminué
7#5	7 à quinte augmentée	**6**	majeur six
sus2	suspendu 2	**sus4**	suspendu 4
add2, **add9**	add 9	**7sus4**	7 suspendu 4

Modes

Les modes ou gammes, sont encodés par un nombre entier :

Mode	Description	Mode	Description
0	majeur diatonique, ionien	**14**	mineur harmonique
1	dorien	**15**	locrien 6
2	phrygien	**16**	ionien #5
3	lydien	**17**	dorien #4, roumain
4	mixolydien	**18**	superphrygien, andalous
5	mineur naturel, éolien	**19**	lydien #2
6	locrien	**20**	diminué
7	mineur mélodique, éolien	**21**	mineur naturel
8	javanais, dorien b2	**22**	locrien
9	lydien augmenté	**23**	majeur diatonique, ionien
10	lydien dominant, Bartók	**24**	dorien
11	mixolydien b13, hindou	**25**	phrygien
12	mineur diminué, éolien b5	**26**	lydien
13	superlocrien	**27**	mixolydien
28	éthiopien	**29**	napolitain
30	napolitain harmonique	**31**	gitan, gitan hongrois
32	tzigane, hongrois mineur	**33**	hongrois majeur
34	bohémien, byzantin	**35**	gypzy
36	enigmatic	**37**	Sebastian
38	indien	**39**	arabe
40	oriental	**41**	locrien majeur
42	persan	**43**	mixolydien augmenté
44	lydien mineur		

Harmonic functions

Les fonctions harmoniques sont encodées par une string :

String	Description
T	Tonique, degré **I** à l'état fondamental
Ts	Tonique, degré **I** à l'état fondamental avec la sensible.
t	Tonique, degré **I** à l'état de renversement ou autre degré sans la sensible.
ts	Tonique, degré **I** à l'état de renversement ou autre degré avec la sensible.
D	Dominante, avec le triton tonal (sensible et quarte).
d	Dominante, avec la sensible mais sans la quarte.
Dst	Dominante, substitution tritonique, subV7 ou bII7 avec quarte et sensible.
D2	Dominante secondaire, V7/?.
D2st	Substitution tritonique de la dominante secondaire, subV7/?.
SD	Sous-dominante.
SD2	Sous-dominante secondaire.

Les fonctions des notes

$HARMONY.getRelativeNote(note, interval)

Renvoie, à partir de la note donnée, la note à distance de l'intervalle montant donné.

- **note** : la note sélectionnée.
- **interval** : l'intervalle sélectionné.
  Par exemple '**P4**' pour la quarte', '**P5**' pour la quinte' et '**M7**' pour la sensible'.

➜ Renvoie, à partir de la note donnée, la note à distance de l'intervalle montant donné.

```
note = $HARMONY.getRelativeNote('C', 'P4'); // returns 'F' quarte
note = $HARMONY.getRelativeNote('C#', 'P5'); // returns 'G#' quinte
note = $HARMONY.getRelativeNote('Cb', 'M7'); // returns 'Bb' sensible
note = $HARMONY.getRelativeNote('Cb', 'D5'); // returns 'Gbb'
note = $HARMONY.getRelativeNote('C#', 'A5'); // returns 'Gx'
```

$HARMONY.getRelativeNotes(note, intervals)

Renvoie, à partir de la note donnée, la liste des notes à distance des l'intervalles montants donnés.

- **note** : la note sélectionnée.
- **intervals** : la liste des intervalles sélectionnés.

➜ Renvoie, à partir de la note donnée, la liste des notes à distance des l'intervalles montants donnés.

```
notes = $HARMONY.getRelativeNotes('E', ['M3', 'P5', 'M7']);
// returns ['G#', 'B', 'D#']);
notes = $HARMONY.getRelativeNotes('F#', ['M2', 'M3', 'A4']);
// returns ['G#', 'A#', 'B#']);
notes = $HARMONY.getRelativeNotes('D', ['A5','P1','M6','M3','m2','P4']);
// returns ['A#', 'D', 'B', 'F#', 'Eb', 'G']
```

$HARMONY.getEnharmonic(tone)

Renvoie la note enharmonique de la note donnée.

- **tone** : la note sélectionnée.

➜ Renvoie la note enharmonique de la note donnée.

```
note = $HARMONY.getEnharmonic('C'); // returns 'B#'
note = $HARMONY.getEnharmonic('C#'); // returns 'Db'
note = $HARMONY.getEnharmonic('E'); // returns 'Fb'
note = $HARMONY.getEnharmonic('F'); // returns 'E#'
```

$HARMONY.getLatinNote(note)

Renvoie le nom latin de la note.

- **note** : la note sélectionnée.

➜ Renvoie le nom latin de la note.

```
note = $HARMONY.getLatinNote('C'); // returns 'do'
note = $HARMONY.getLatinNote('D'); // returns 'ré'
note = $HARMONY.getLatinNote('E'); // returns 'mi'
note = $HARMONY.getLatinNote('F'); // returns'fa'
note = $HARMONY.getLatinNote('G'); // returns 'sol'
note = $HARMONY.getLatinNote('A'); // returns 'la'
note = $HARMONY.getLatinNote('B'); // returns 'si'
note = $HARMONY.getLatinNote('C#'); // returns 'do#'
note = $HARMONY.getLatinNote('Cx'); // returns 'dox'
note = $HARMONY.getLatinNote('Cb'); // returns 'dob'
note = $HARMONY.getLatinNote('Cbb'); // returns 'dobb'
```

Les fonctions des tons

$HARMONY.getFarthestTone(tone)

Renvoie le ton le plus éloigné du cycle des quintes.

- **tone** : le ton sélectionné.

➜ Renvoie le ton le plus éloigné du cycle des quintes.

```
tone = $HARMONY.getFarthestTone('C'); // returns 'Gb'
tone = $HARMONY.getFarthestTone('Bb'); // returns 'Fb'
tone = $HARMONY.getFarthestTone('C#'); // returns 'G'
tone = $HARMONY.getFarthestTone('F#'); // returns 'C'
```

$HARMONY.getSimpleTone(tone)

Renvoie le ton enharmonique simple.

- **tone** : le ton sélectionné.

➜ Renvoie le ton enharmonique simple.

```
tone = $HARMONY.getSimpleTone('Cb'); // returns 'B'
tone = $HARMONY.getSimpleTone('B#'); // returns 'C';
tone = $HARMONY.getSimpleTone('E#'); // returns 'F'
tone = $HARMONY.getSimpleTone('Fb'); // returns 'E'
tone = $HARMONY.getSimpleTone('Bbb'); // returns 'A'
tone = $HARMONY.getSimpleTone('Cx'); // returns 'D'
tone = $HARMONY.getSimpleTone('C'); // returns 'C'
tone = $HARMONY.getSimpleTone('Bb'); // returns 'Bb'
tone = $HARMONY.getSimpleTone('A#'); // returns 'A#'
```

$HARMONY.getKeyNotes(key)

Renvoie la liste des notes associées à une clé choisie.

- **key** : la clé sélectionnée.

➜ Renvoie la liste des notes associées à une clé choisie.

```
notes=$HARMONY.getKeyNotes('C'); // ['C','D','E','F','G','A','B']
notes=$HARMONY.getKeyNotes('Cm'); // ['C','D','Eb','F','G','Ab','Bb']
notes=$HARMONY.getKeyNotes('C#'); // ['C#','D#','E#','F#','G#','A#','B#']
notes=$HARMONY.getKeyNotes('C#m'); // ['C#','D#','E','F#','G#','A','B']
notes=$HARMONY.getKeyNotes('Cb'); //['Cb','Db','Eb','Fb','Gb','Ab','Bb']
notes=$HARMONY.getKeyNotes('A'); // ['A','B','C#','D','E','F#','G#']
notes=$HARMONY.getKeyNotes('Am'); // ['A','B','C','D','E','F','G']
```

Les fonctions des tonalités

$HARMONY.getToneOfTonality(tonality)

Renvoie le ton de la tonalité choisie.

- **tonality** : la tonalité sélectionnée.

➜ Renvoie le ton de la tonalité choisie.

```
tone = $HARMONY.getToneOfTonality('MC'); // returns 'C'
tone = $HARMONY.getToneOfTonality('mC'); // returns 'C'
tone = $HARMONY.getToneOfTonality('MCb'); // returns 'Cb'
tone = $HARMONY.getToneOfTonality('MC#'); // returns 'C#'
```

$HARMONY.getToneAndMinor(tonality)

Renvoie le ton et mode majeur ou mineur.

- **tonality** : la tonalité sélectionnée.

➜ Renvoie le ton et mode majeur ou mineur.

```
tone = $HARMONY.getToneAndMinor('MC');
// returns { tone: 'C', minor: false }
tone = $HARMONY.getToneAndMinor('mC');
// returns { tone: 'C', minor: true }
```

$HARMONY.getToneAndMode(tonality)

Renvoie le ton et le mode de la tonalité choisie.

- **tonality** : la tonalité sélectionnée.

➜ Renvoie le ton et le mode de la tonalité choisie.

```
tone = $HARMONY.getToneAndMode('MC');
// returns { tone: 'C', mode: 0, minor: false }
tone = $HARMONY.getToneAndMode('mC');
// returns { tone: 'C', mode: 14, minor: true }
```

$HARMONY.getEnharmonicTone(tonality)

Renvoie le ton enharmonique en majeur et mineur de la tonalité choisie.

- **tonality** : la tonalité sélectionnée.

➜ Renvoie le ton enharmonique en majeur et mineur de la tonalité choisie.

```
en = $HARMONY.getEnharmonicTone('MCb');
// returns { major: 'Cb', minor: 'B' }
en = $HARMONY.getEnharmonicTone('mA#');
// returns { minor: 'A#', major: 'Bb' }
```

Les fonctions des accords

$HARMONY.getNotesFromSignature(note, signature)

Renvoie la liste des notes de l'accord ayant la signature donnée et la note donnée comme fondamentale.

- **note** : la note sélectionnée, fondamentale de l'accord.
- **signature** : la signature sélectionnée.

➜ Renvoie la liste des notes de l'accord ayant la signature donnée et la note donnée comme fondamentale.

```
notes = $HARMONY.getNotesFromSignature('C', 'M7'); // ['C','E','G','B']
notes = $HARMONY.getNotesFromSignature('F#', ''); // ['F#','A#','C#']
notes = $HARMONY.getNotesFromSignature('A', '°7'); // ['A','C','Eb','Gb']
```

$HARMONY.getChordNotes(chord)

Renvoie la liste des notes de l'accord choisi.

- **chord** : l'accord sélectionné.

➜ Renvoie la liste des notes de l'accord choisi.

```
notes = $HARMONY.getChordNotes('C°7'); // ['C', 'Eb', 'Gb', 'Bbb']
notes = $HARMONY.getChordNotes('Dbm'); // ['Db', 'Fb', 'Ab']);
notes = $HARMONY.getChordNotes('Em7'); // ['E', 'G', 'B', 'D']
notes = $HARMONY.getChordNotes('F#M7'); // ['F#', 'A#', 'C#', 'E#']
notes = $HARMONY.getChordNotes('G7#9'); // ['G', 'B', 'D', 'F', 'A#']
notes = $HARMONY.getChordNotes('F(#5)'); // ['F', 'A', 'C#']
notes = $HARMONY.getChordNotes('F#(#5)'); // ['F#', 'A#', 'Cx']
notes = $HARMONY.getChordNotes('Cm/Eb'); // ['Eb', 'G', 'C']
notes = $HARMONY.getChordNotes('Cm7/A'); // ['A', 'C', 'Eb', 'G', 'Bb']
```

$HARMONY.getChordNumberOfNote(signature)

Renvoie le nombre de note d'un accord à partir de sa signature.

- **signature** : la signature sélectionnée.

➜ Renvoie le nombre de note d'un accord à partir de sa signature.

```
n = $HARMONY.getChordNumberOfNote(''); // returns 3
n = $HARMONY.getChordNumberOfNote('m'); // returns 3
n = $HARMONY.getChordNumberOfNote('7'); // returns 4
n = $HARMONY.getChordNumberOfNote('7b5'); // returns 4
n = $HARMONY.getChordNumberOfNote('7b9'); // returns 5
```

$HARMONY.getChordSignatureName(signature)

Renvoie le nom d'usage d'une signature en français.

- **signature** : la signature sélectionnée.

➜ Renvoie le nom d'usage de l'accord choisi (en français).

```
name = $HARMONY.getChordSignatureName(''); // returns 'majeur'
name = $HARMONY.getChordSignatureName('m'); // returns 'mineur'
name = $HARMONY.getChordSignatureName('°'); // returns 'diminué'
name = $HARMONY.getChordSignatureName('+5'); // returns 'augmenté'
name = $HARMONY.getChordSignatureName('7'); // returns '7'
name = $HARMONY.getChordSignatureName('m7'); // returns 'mineur 7'
name = $HARMONY.getChordSignatureName('mM7'); // returns 'mineur, majeure 7'
name = $HARMONY.getChordSignatureName('M7'); // returns 'majeure 7'
name = $HARMONY.getChordSignatureName('m7b5'); // returns 'demi-diminué'
name = $HARMONY.getChordSignatureName('°7'); // returns 'diminué 7'
name = $HARMONY.getChordSignatureName('7b9'); // returns '7 bémol 9'
name = $HARMONY.getChordSignatureName('79'); // returns '7 majeure 9'
name = $HARMONY.getChordSignatureName('7#9'); // returns '7 dièse 9'
name = $HARMONY.getChordSignatureName('7sus4'); // returns '7 suspendu 4'
```

$HARMONY.getChordName(tone, signature, html)

Renvoie le nom d'usage d'un accord en français.

- **tone** : le ton sélectionné.
- **signature** : la signature sélectionnée.
- **html** : format de la string de sortie, par défaut à true. Cela permet de mettre le nom de la note en italique.

➜ Renvoie le nom d'usage de l'accord choisi (en français).

```
name = $HARMONY.getChordName(''); // returns '<i>do</i> majeur'
name = $HARMONY.getChordName('A', 'm', false); // returns 'la mineur'
name = $HARMONY.getChordName('Eb', 'm7'); // returns'<i>mi bémol</i> mineur 7'
name = $HARMONY.getChordName('F#', 'M7', false);  // returns 'fa dièse majeure 7'
name = $HARMONY.getChordName('B', '°'); // returns '<i>si</i> diminué'
name = $HARMONY.getChordName('Bb', '7sus4', false);  // returns 'si bémol 7 suspe
```

$HARMONY.getChordIntervals(signature)

Renvoie la liste des intervalles en jeux pour une signature.

- **signature** : la signature sélectionnée.

➜ Renvoie la liste des intervalles en jeux pour une signature.

```
intervals = $HARMONY.getChordIntervals(''); // ['M3', 'P5']
intervals = $HARMONY.getChordIntervals('m'); // ['m3', 'P5']
intervals = $HARMONY.getChordIntervals('m7'); // ['m3', 'P5', 'm7']
intervals = $HARMONY.getChordIntervals('M7'); // ['M3', 'P5', 'M7']
intervals = $HARMONY.getChordIntervals('7#9'); // ['M3','P5','m7','A9']
```

$HARMONY.getChordFondamental(chord)

Renvoie la note fondamentale de l'accord choisi.

- **chord** : l'accord sélectionné.

➜ Renvoie la note fondamentale de l'accord choisi.

```
note = $HARMONY.getChordFondamental('C°7'); // returns 'C'
note = $HARMONY.getChordFondamental('Dbm'); // returns 'Db'
note = $HARMONY.getChordFondamental('A#M7'); // returns 'A#'
note = $HARMONY.getChordFondamental('Bm7b5'); // returns 'B'
note = $HARMONY.getChordFondamental('Bb(#5)'); // returns 'Bb'
note = $HARMONY.getChordFondamental('C/E'); // returns 'C'
```

$HARMONY.getChordBass(chord)

Renvoie la note à la basse de l'accord choisi.

- **chord** : l'accord sélectionné.

➜ Renvoie la note à la basse de l'accord choisi.

```
bass = $HARMONY.getChordBass('C°7'); // returns 'C'
bass = $HARMONY.getChordBass('Dbm'); // returns 'Db'
bass = $HARMONY.getChordBass('C/E');  // returns 'E'
bass = $HARMONY.getChordBass('Cm/Ab');  // returns 'Ab'
bass = $HARMONY.getChordBass('Bbm7b5/F');  // returns 'F'
```

$HARMONY.getChordSignature(chord)

Renvoie la signature de l'accord choisi.

- **chord** : l'accord sélectionné.

➜ Renvoie la signature de l'accord choisi.

```
sig = $HARMONY.getChordSignature('C°7'); // returns '°7'
sig = $HARMONY.getChordSignature('G#');  // returns ''
sig = $HARMONY.getChordSignature('Dbm'); // returns 'm'
sig = $HARMONY.getChordSignature('C/E');  // returns ''
sig = $HARMONY.getChordSignature('Bbm7b5/F');  // returns 'm7b5'
```

Les fonctions des modes

$HARMONY.getModeIntervals(mode)

Renvoie la liste des intervalles contituants un mode.

- **mode** : le mode sélectionné.

➜ Renvoie une liste de string des intervalles dans l'ordre croissant des degrés.

```
intervals = $HARMONY.getModeIntervals(0); // mode majeur diatonique
// returns ['P1', 'M2', 'M3', 'P4', 'P5', 'M6', 'M7']

intervals = $HARMONY.getModeIntervals(7); // mode mineur mélodique
// returns ['P1', 'M2', 'm3', 'P4', 'P5', 'M6', 'M7']

intervals = $HARMONY.getModeIntervals(14); // mode mineur harmonique
// returns ['P1', 'M2', 'm3', 'P4', 'P5', 'm6', 'M7']

intervals = $HARMONY.getModeIntervals(21); // mode mineur naturel
// returns ['P1', 'M2', 'm3', 'P4', 'P5', 'm6', 'm7']
```

$HARMONY.getModeStructure(mode)

Renvoie une liste de deux strings, le nom du pentacorde suivi du nom du tétracorde.

- **mode** : le mode sélectionné.

➜ Renvoie une liste de deux strings, le nom du **pentacorde** suivi du nom du **tétracorde**.

```
s = $HARMONY.getModeStructure(0); // mode majeur diatonique
// returns ['majeur', 'majeur']

s = $HARMONY.getModeStructure(7); // mode mineur mélodique
// returns ['mineur', 'majeur']

s = $HARMONY.getModeStructure(14); // mode mineur harmonique
// returns ['mineur', 'harmonique']

s = $HARMONY.getModeStructure(21); // mode mineur naturel
// returns ['mineur', 'phrygien']
```

$HARMONY.isModeHarmonisable(mode)

Renvoie un booléen indiquant si le mode choisi peut-être harmonisé ou non.

- **mode** : le mode sélectionné.

➜ Renvoie **true** si le mode est harmonisable, **false** sinon.

```
b = $HARMONY.isModeHarmonisable(0); // true, 0 = mode majeur diatonique
b = $HARMONY.isModeHarmonisable(29); // false, 29 = mode napolitain
```

$HARMONY.getModeName(mode)

Renvoie une string du nom du mode choisi.

- **mode** : le mode sélectionné.

➜ Renvoie une string du nom du mode choisi.

```
name = $HARMONY.getModeName(0); // returns "majeur diatonique"
name = $HARMONY.getModeName(7); // returns "mineur mélodique"
name = $HARMONY.getModeName(14); // returns "mineur harmonique"
name = $HARMONY.getModeName(21); // returns "mineur naturel"
```

$HARMONY.getModeNaturalName(mode)

Renvoie une string du nom du mode *ancien* choisi.

- **mode** : le mode sélectionné.

➜ Renvoie une string du nom du mode *ancien* choisi.

```
name = $HARMONY.getModeNaturalName(0); // returns "ionien"
name = $HARMONY.getModeNaturalName(1); // returns "dorien"
name = $HARMONY.getModeNaturalName(2); // returns "phrygien"
name = $HARMONY.getModeNaturalName(3); // returns "lydien"
name = $HARMONY.getModeNaturalName(4); // returns "mixolydien"
name = $HARMONY.getModeNaturalName(5); // returns "éolien"
name = $HARMONY.getModeNaturalName(6); // returns "locrien"
name = $HARMONY.getModeNaturalName(14); // returns "mode I"
name = $HARMONY.getModeNaturalName(15); // returns "mode II"
```

$HARMONY.getModeNotes(mode, tone)

Renvoie la liste des notes d'un mode donné pour un ton donné.

- **mode** : le mode sélectionné.
- **tone** : le ton sélectionné.

➜ Renvoie une liste de string des notes dans l'ordre croissant des degrés.

```
notes = $HARMONY.getModeNotes(0, 'E'); // mode majeur diatonique, ton E
// returns ['E', 'F#', 'G#', 'A', 'B', 'C#', 'D#']

notes = $HARMONY.getModeNotes(7, 'C'); // mode mineur mélodique, ton C
// returns ['C', 'D', 'Eb', 'F', 'G', 'A', 'B']

notes = $HARMONY.getModeNotes(14, 'A#'); // mode mineur harmonique, ton A#
// returns ['A#', 'B#', 'C#', 'D#', 'E#', 'F#', 'Gx']

notes = $HARMONY.getModeNotes(21, 'Ab'); // mode mineur naturel, ton Ab
// returns ['Ab', 'Bb', 'Cb', 'Db', 'Eb', 'Fb', 'Gb']
```

$HARMONY.getModeSignature(mode, degree, bSeven)

Renvoie la signature de l'accord d'un degré dans un mode donné.

- **mode** : le mode sélectionné.
- **degree** : le degré sélectionné de 0 à 6.
- **bSeven** : **true** pour une signature à quatre sons, **false** pour trois sons.

➜ Renvoie la signature de l'accord d'un degré dans un mode donné.

```
s = $HARMONY.getModeSignature(0, 1, false); // majeur diatonique, 2e degré
// returns 'm'
s = $HARMONY.getModeSignature(0, 6, true); // majeur diatonique, 7e degré
// returns 'Ø7'
```

$HARMONY.getModeSignatures(mode, bSeven)

Renvoie la liste des signatures d'un mode donné.

- **mode** : le mode sélectionné.
- **bSeven** : **true** pour une signature à quatre sons, **false** pour trois sons.

➜ Renvoie la liste de string des signatures dans l'ordre croissant des degrés.

```
s = $HARMONY.getModeSignatures(0, false); // majeur diatonique
// returns ['', 'm', 'm', '', '', 'm', '°']

s = $HARMONY.getModeSignatures(0, true); // mode majeur diatonique
// returns ['M7', 'm7', 'm7', 'M7', '7', 'm7', 'Ø7']

s = $HARMONY.getModeSignatures(7, false); // mode mineur mélodique
// returns ['m', 'm', '+5', '', '', '°', '°']

s = $HARMONY.getModeSignatures(7, true); // mode mineur mélodique
// returns ['mM7', 'm7', '+M7', '7', '7', 'Ø7', 'Ø7']

s = $HARMONY.getModeSignatures(14, false); // mode mineur harmonique
// returns ['m', '°', '+5', 'm', '', '', '°']

s = $HARMONY.getModeSignatures(14, true); // mode mineur harmonique
// returns ['mM7', 'Ø7', '+M7', 'm7', '7', 'M7', '°7']

s = $HARMONY.getModeSignatures(21, false); // mode mineur naturel
// returns ['m', '°', '', 'm', 'm', '', '']
```

$HARMONY.getModeFunctions(mode, bSeven)

Renvoie la liste des fonctions harmoniques d'un mode donné.

- **mode** : le mode sélectionné.
- **bSeven** : **true** pour une signature à quatre sons, **false** pour trois sons.

➜ Renvoie la liste de string des fonctions harmoniques dans l'ordre croissant des degrés.

```
f = $HARMONY.getModeFunctions(0, false); // majeur diatonique, 3 sons
// returns ['T',  'SD', 'd',  'SD', 'd', 't', 'D']

f = $HARMONY.getModeFunctions(0, true); // majeur diatonique, 4 sons
// returns ['Ts', 'SD', 'ts', 'SD', 'D', 't', 'D']

f = $HARMONY.getModeFunctions(14, false); // mineur harmonique, 3 sons
// returns ['T', 'SD', 'd', 'SD', 'd', 'SD', 'D']

f = $HARMONY.getModeFunctions(14, true); // mineur harmonique, 4 sons
// returns ['Ts', 'SD', 'ts', 'SD', 'D', 'SD', 'D']
```

$HARMONY.getModeChords(mode, tone, bSeven)

Renvoie la liste des accords d'un mode donné.

- **mode** : le mode sélectionné.
- **tone** : le ton sélectionné.
- **bSeven** : **true** pour une signature à quatre sons, **false** pour trois sons.

➜ Renvoie la liste de string des accords dans l'ordre croissant des degrés.

```
chords = $HARMONY.getModeChords(0, 'I', false); // Degré I
// returns ['', 'm', 'm', '', '', 'm', '°']

chords = $HARMONY.getModeChords(0, 'I', true); // Degré I
// returns ['M7', 'm7', 'm7', 'M7', '7', 'm7', 'Ø7']

chords = $HARMONY.getModeChords(0, 'C', false); // majeur diatonique
// returns ['C', 'Dm', 'Em', 'F', 'G', 'Am', 'B°']

chords = $HARMONY.getModeChords(0, 'D', true); // majeur diatonique
// returns ['DM7', 'Em7', 'F#m7', 'GM7', 'A7', 'Bm7', 'C#Ø7']

chords = $HARMONY.getModeChords(7, 'E', false); // mode mineur mélodique
// returns ['Em', 'F#m', 'G+5', 'A', 'B', 'C#°', 'D#°']

chords = $HARMONY.getModeChords(7, 'F', true); // mode mineur mélodique
// returns ['FmM7', 'Gm7', 'Ab+M7', 'Bb7', 'C7', 'DØ7', 'EØ7']

chords = $HARMONY.getModeChords(14, 'G', false); // mode mineur harmonique
// returns ['Gm', 'A°', 'Bb+5', 'Cm', 'D', 'Eb', 'F#°']

chords = $HARMONY.getModeChords(14, 'A', true); // mode mineur harmonique
// returns ['AmM7', 'BØ7', 'C+M7', 'Dm7', 'E7', 'FM7', 'G#°7']

chords = $HARMONY.getModeChords(21, 'Bb', false); // mode mineur naturel
// returns ['Bbm', 'C°', 'Db', 'Ebm', 'Fm', 'Gb', 'Ab']
```

Les fonctions harmoniques

$HARMONY.getHarmonicFunction(intervals, degree, bSeven)

Renvoie la fonction harmonique d'un accord au degré donné dans une gamme donnée.

- **intervals** : une liste de sept intervales décrivant la gamme sélectionnée.
- **degree** : le degré sélectionné de 0 à 6.
- **bSeven** : **true** pour une signature à quatre sons, **false** pour trois sons.

➜ Renvoie la fonction harmonique d'un accord au degré donné dans une gamme donnée.

```
var major = ['P1','M2','M3','P4','P5','M6','M7'];
fn = $HARMONY.getHarmonicFunction(major, 0, false); // returns 'T'
fn = $HARMONY.getHarmonicFunction(major, 0, true);  // returns 'Ts'
fn = $HARMONY.getHarmonicFunction(major, 2, true);  // returns 'ts'
fn = $HARMONY.getHarmonicFunction(major, 5, false); // returns 't'
fn = $HARMONY.getHarmonicFunction(major, 2, false); // returns 'd'
fn = $HARMONY.getHarmonicFunction(major, 4, false); // returns 'd'
fn = $HARMONY.getHarmonicFunction(major, 4, true);  // returns 'D'
fn = $HARMONY.getHarmonicFunction(major, 1, false); // returns 'SD'
```

Les fonctions harmoniques générées sont encodées par une string :

String	Description
T	Tonique, degré **I** à l'état fondamental
Ts	Tonique, degré **I** à l'état fondamental avec la sensible.
t	Tonique, degré **I** à l'état de renversement ou autre degré sans la sensible.
ts	Tonique, degré **I** à l'état de renversement ou autre degré avec la sensible.
D	Dominante, avec le triton tonal (sensible et quarte).
d	Dominante, avec la sensible mais sans la quarte.
SD	Sous-dominante.

$MUSIC_UI

This JavaScript module handles musical selectors for html pages.

- Module file is: music-ui-1.0.0.min.js.
- It has dependencies: harmony-1.0.0.min.js and music.min.css.

Files to include in HTML file header

```
<link rel="stylesheet" href="css/music.min.css" />

<script src="libs/harmony-1.0.0.min.js"></script>
<script src="libs/music-ui-1.0.0.min.js"></script>
```

$MUSIC_UI Methods

Method	Description
init()	Initializes the library to setup language.
buildTones()	Shows a tone selector.
buildTonalities()	Shows a tonality selector.
buildMajorTonalities()	Shows a major tonality selector.
buildMinorTonalities()	Shows a minor tonality selector.
buildSignatures()	Shows a chord signature selector.
buildModesTable()	Shows table with modes.
getSignatureClass()	Returns the class to setup background color.
getSignatureColor()	Returns the color associated to chord signature.
getSignatureNotation()	Returns the notation associated to chord signature.
getSignatureHtmlFormat()	Returns the notation in html format for chord signature.
getChordHtmlFormat()	Returns the chord notation in html format.
getChordHtmlFormatFromChord()	Returns the chord notation in html format.
getDegreeHtmlFormat()	Returns the degree notation in html format.
getFunctionAttributes()	Returns the attributes of a harmonic function.

Examples

- Tonalities • Scale's tables • Chords tranposition • Transposable harmonic analysis

$MUSIC_UI.init()

$MUSIC_UI.init(options);

init() Initializes the library to setup language.

- **options** { 'lang': value }, value is **'en'** for english or **'fr'** for french.

Example

JavaScript

```
$MUSIC_UI.init({'lang':'en'});
```

$MUSIC_UI.buildTones()

$MUSIC_UI.buildTones(id, tone, cb_update);

buildTones() shows a tone selector.

- **id** unique identificator of the selector.
- **tone** the tone to initialize the selector.
- **cb_update** a callback invoked when selection change.

buildTones calls the given **cb_update** callback, with the given selected **tone**.

Example

HTML

```
<div align="center">a tone selector
   <div id="tonesSelector"></div>
   <br>Selected tone is: <span id="selectedTone" class="big"></span>.
</div>
```

JavaScript

```
function updateTone(tone) {
   document.querySelector('#selectedTone').innerHTML = tone;
}
$MUSIC_UI.buildTones('tonesSelector', 'E', updateTone);
```

a tone selector

Cb C C# Db D D# Eb E E#

Fb F F# Gb G G# Ab A A# Bb B B#

Selected tone is: E.

$MUSIC_UI.buildTonalities()

$MUSIC_UI.buildTonalities(id, tonality, cb_update);

buildTonalities() shows a tonality selector. It allows to select a major or minor scale tonality.

- **id** unique identificator of the selector.
- **tonality** the tonality to initialize the selector.
- **cb_update** a callback invoked when selection change.

buildTones calls the given **cb_update** callback, with the given selected **tonality**.

> Note the tonality encoding: mode than tone.
> - mode: '**M**' for major and '**m**' for minor.
> - tone: note with English notation, **C**, **Cb**, **C#**.

Example

HTML

```
<div align="center">a tonality selector
   <div id="tonalitiesSelector"></div>
   <br>Selected tonality is: <span id="selectedTonality"></span>.
</div>
```

JavaScript

```
function updateTonality(tonality) {
   document.querySelector('#selectedTonality').innerHTML = tonality;
}
$MUSIC_UI.buildTonalities('tonalitiesSelector', 'mF#', updateTonality);
```

a tonality selector

Cb	Gb	Db	Ab	Eb	Bb	F	C	G	D	A	E	B	F#	C#
Abm	Ebm	Bbm	Fm	Cm	Gm	Dm	Am	Em	Bm	F#m	C#m	G#m	D#m	A#m

Selected tonality is: mF#.

$MUSIC_UI.buildMajorTonalities()

$MUSIC_UI.buildMajorTonalities(id, tonality, cb_update);

buildMajorTonalities() shows a major tonality selector. It allows to select a major scale tonality.

- **id** unique identificator of the selector.
- **tonality** the tonality to initialize the selector.
- **cb_update** a callback invoked when selection change.

buildMajorTonalities calls the given **cb_update** callback, with the given selected **tonality**.

> Note the tonality encoding: mode than tone.
> - mode: '**M**' for major.
> - tone: note with English notation, **C**, **Cb**, **C#**.

Example

HTML

```
<div align="center">a major tonality selector
   <div id="majorTonalitiesSelector"></div>
   <br>Selected tonality is: <span id="selectedMajorTonality"></span>.
</div>
```

JavaScript

```
function updateMajorTonality(tonality) {
   document.querySelector('#selectedMajorTonality').innerHTML = tonality;
}
$MUSIC_UI.buildMajorTonalities('majorTonalitiesSelector', 'MD', updateMajorTonality);
```

a major tonality selector

Cb	Gb	Db	Ab	Eb	Bb	F	C	G	D	A	E	B	F#	C#

Selected tonality is: MD.

$MUSIC_UI.buildMinorTonalities()

$MUSIC_UI.buildMinorTonalities(id, tonality, cb_update);

buildMinorTonalities() shows a minor tonality selector. It allows to select a minor scale tonality.

- **id** unique identificator of the selector.
- **tonality** the tonality to initialize the selector.
- **cb_update** a callback invoked when selection change.

buildMinorTonalities calls the given **cb_update** callback, with the given selected **tonality**.

> Note the tonality encoding: mode than tone.
> - mode: '**m**' for minor.
> - tone: note with English notation, **C**, **Cb**, **C#**.

Example

HTML

```
<div align="center">a minor tonality selector
   <div id="minorTonalitiesSelector"></div>
   <br>Selected tonality is: <span id="selectedMinorTonality"></span>.
</div>
```

JavaScript

```
function updateMinorTonality(tonality) {
   document.querySelector('#selectedMinorTonality').innerHTML = tonality;
}
$MUSIC_UI.buildMinorTonalities('minorTonalitiesSelector', 'mA', updateMinorTonality);
```

a minor tonality selector

Ab	Eb	Bb	F	C	G	D	A	E	B	F#	C#	G#	D#	A#

Selected tonality is: mA.

$MUSIC_UI.buildSignatures()

$MUSIC_UI.buildSignatures(id, signatures, selected, cb_update);

buildSignatures() shows a chord signature selector.

- **id** unique identificator of the selector.
- **signatures** the list of signatures for the selection. Signatures are string.
- **selected** index of the signature list for selector.
- **cb_update** a callback invoked when selection change.

buildSignatures calls the given **cb_update** callback, with the given selected **signature**.

> Note that in the signture list, fi you add char '**!**' behind signature, it's show a line return.

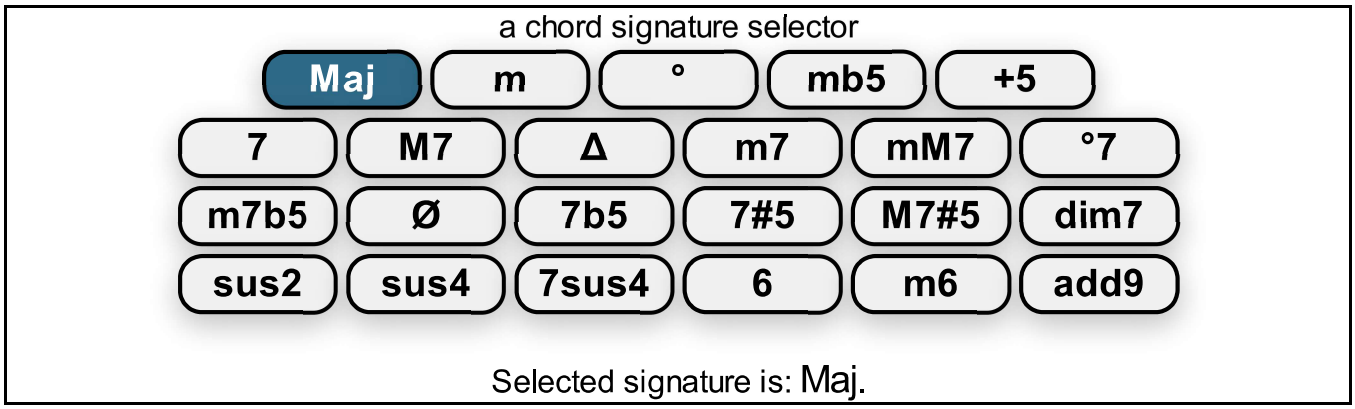
Example

HTML

```
<div align="center">a chord signature selector
    <div id="signaturesSelector"></div>
    <br>Selected signature is: <span id="selectedSignature" class="big"></span>
</div>
```

JavaScript

```
function updateSignature(signature) {
    document.querySelector('#selectedSignature').innerHTML = signature;
}
var signatures = [
    'Maj', 'm', '°', 'mb5', '+5',
    '!7', 'M7', 'Δ', 'm7', 'mM7', '°7',
    '!m7b5', 'Ø', '7b5', '7#5', 'M7#5', 'dim7',
    '!sus2', 'sus4', '7sus4', '6', 'm6', 'add9'
];
$MUSIC_UI.buildSignatures('signaturesSelector', signatures, 0, updateSignature);
```



$MUSIC_UI.buildModesTable()

$MUSIC_UI.buildModesTable(id, mode, bSeven, tone);

buildModesTable() shows table of chord for each scales of mode.

- **id** unique identificator table.
- **mode** the mode number.
- **bSeven** true for tetrads, false for triads.
- **tone** the selected tone, see with ligthgrey in background.

Example

HTML

```
<div id="modes-table"></div>
```

JavaScript

```
$MUSIC_UI.buildModesTable('modes-table', 14, true, 'E');
```

Degrees	I	II	bIII	IV	V	bVI	VII
Signatures	**m^{M7}**	**Ø**	**+M7**	**m^{7}**	**7**	**M7**	**dim^{7}**
Functions	**T**	**SD**	**T**	**SD**	**D**	**SD**	**D**
Scales	Abm^{M7}	Bb^{Ø}	Cb^{+M7}	Dbm^{7}	Eb^{7}	Fb^{M7}	G°^{7}
	Ebm^{M7}	F^{Ø}	Gb^{+M7}	Abm^{7}	Bb^{7}	Cb^{M7}	D°^{7}
	Bbm^{M7}	C^{Ø}	Db^{+M7}	Ebm^{7}	F^{7}	Gb^{M7}	A°^{7}
	Fm^{M7}	G^{Ø}	Ab^{+M7}	Bbm^{7}	C^{7}	Db^{M7}	**E°^{7}**
	Cm^{M7}	D^{Ø}	Eb^{+M7}	Fm^{7}	G^{7}	Ab^{M7}	B°^{7}
	Gm^{M7}	A^{Ø}	Bb^{+M7}	Cm^{7}	D^{7}	Eb^{M7}	F#°^{7}
	Dm^{M7}	**E^{Ø}**	F^{+M7}	Gm^{7}	A^{7}	Bb^{M7}	C#°^{7}
	Am^{M7}	B^{Ø}	C^{+M7}	Dm^{7}	**E^{7}**	F^{M7}	G#°^{7}
	Em^{M7}	**F#^{Ø}**	**G^{+M7}**	**Am^{7}**	**B^{7}**	**C^{M7}**	**D#°^{7}**
	Bm^{M7}	C#^{Ø}	D^{+M7}	**Em^{7}**	F#^{7}	G^{M7}	A#°^{7}
	F#m^{M7}	G#^{Ø}	A^{+M7}	Bm^{7}	C#^{7}	D^{M7}	E#°^{7}
	C#m^{M7}	D#^{Ø}	**E^{+M7}**	F#m^{7}	G#^{7}	A^{M7}	B#°^{7}
	G#m^{M7}	A#^{Ø}	B^{+M7}	C#m^{7}	D#^{7}	**E^{M7}**	Fx°^{7}
	D#m^{M7}	E#^{Ø}	F#$^{+M7}$	G#m^{7}	A#^{7}	B^{M7}	Cx°^{7}
	A#m^{M7}	B#^{Ø}	C#$^{+M7}$	D#m^{7}	E#^{7}	F#^{M7}	Gx°^{7}

$MUSIC_UI.getSignatureClass()

$MUSIC_UI.getSignatureClass(signature);

getSignatureClass() returns the class from music.css to setup background color from chord signature.

- **signature** chord signature in list ['', 'Maj', '+', '+5', '#5', 'm', '°', 'M7', 'M7+5', '+M7', '7', '7sus4', 'm7', 'mM7', 'dim', 'dim7', '°7', 'Ø', 'Ø7', '6', 'm6'].

$MUSIC_UI.getSignatureClass('M7') returns 'sig-majorM7'

List of possible returned value:

- sig-major, sig-major6, sig-majorM7, sig-major7
- sig-augmented, sig-augmentedM7
- sig-minor, sig-minor6, sig-minor7, sig-minorM7, sig-minor7b5
- sig-diminished, sig-diminished7
- sig-suspended4

$MUSIC_UI.getSignatureColor()

$MUSIC_UI.getSignatureColor(signature);

getSignatureColor() returns the color associated to chord signature.

- **signature** chord signature in list ['', 'Maj', '+', '+5', '#5', 'm', '°', 'M7', 'M7+5', '+M7', '7', '7sus4', 'm7', 'mM7', 'dim', 'dim7', '°7', 'Ø', 'Ø7', '6', 'm6'].

$MUSIC_UI.getSignatureClass('M7') returns '#008040'

$MUSIC_UI.getSignatureNotation()

$MUSIC_UI.getSignatureNotation(signature);

getSignatureNotation() returns the notation associated to chord signature.

- **signature** chord signature.

$MUSIC_UI.getSignatureNotation('') returns 'Maj'.

$MUSIC_UI.getSignatureHtmlFormat()

$MUSIC_UI.getSignatureHtmlFormat(signature);

getSignatureHtmlFormat() .

- **signature** chord signature.

$MUSIC_UI.getSignatureHtmlFormat('m7b5')
returns 'm7b5' seen as **m^{7b5}**

$MUSIC_UI.getChordHtmlFormat()

$MUSIC_UI.getChordHtmlFormat(fundamental, signature, bass);

getChordHtmlFormat() returns the chord notation in html format.

- **fundamental** fundamental of the chord.
- **signature** chord signature.
- **bass** bass of the chord if it's not the fundamental.

$MUSIC_UI.getChordHtmlFormat('G#', '7b9')
returns 'G#7b9' seen as **G#^{7b9}**.

$MUSIC_UI.getChordHtmlFormat('G#', '7b9', 'D#')
returns 'G#7b9/D#'
seen as **G#^{7b9}/D#**.

$MUSIC_UI.getChordHtmlFormatFromChord()

$MUSIC_UI.getChordHtmlFormatFromChord(chord);

getChordHtmlFormatFromChord() returns the chord notation in html format.

- **chord** full chord name.

$MUSIC_UI.getChordHtmlFormatFromChord('EbM7,9')
returns 'EbM7,9' seen as **Eb^{M7,9}**

$MUSIC_UI.getChordHtmlFormatFromChord('Gb7#9/Db')
returns 'Gb7#9/Db'
seen as **Gb$^{7\#9}$/Db**.

$MUSIC_UI.getDegreeHtmlFormat()

$MUSIC_UI.getDegreeHtmlFormat(degree);

getDegreeHtmlFormat() returns the degree notation in html format.

- **degree** the harmonic degree.

$MUSIC_UI.getDegreeHtmlFormat('bII=N6') returns 'bII=N6' seen as **bII=N6**.

$MUSIC_UI.getDegreeHtmlFormat('+6') returns '+6' seen as **+6**.

$MUSIC_UI.getDegreeHtmlFormat('It+6')
returns 'It+6' seen as **It^{+6}**.

$MUSIC_UI.getDegreeHtmlFormat('Fr+6')
returns 'Fr+6' seen as **Fr^{+6}**.

$MUSIC_UI.getDegreeHtmlFormat('Ger+6')
returns 'Ger+6' seen as **Ger^{+6}**.

$MUSIC_UI.getDegreeHtmlFormat('IV7')
returns 'VI7' seen as **VI^{7}**.

$MUSIC_UI.getFunctionAttributes()

$MUSIC_UI.getFunctionAttributes(fn);

getFunctionAttributes() Returns the attributes (name, color and background) of a harmonic function.

- **fn** the harmonic function in ['T', 'Ts', 't', 'ts', 'd', 'D', 'Dst', 'D2', 'D2st', 'SD', 'SD2'].

$MUSIC_UI.getFunctionAttributes('D2') returns

```
{
   name:'D²',
   background:'#92D050',
   color:'white'
}
```

$ABC_UI

abc format
score editor

This JavaScript module handles abc format scores in html pages.

- Module file is: abc-ui-1.0.0.min.js.

This module handles:

- Transposable scores (with audio and chords).
- Visual appearance with magnification factor.
- Audio rendered based on soundfont of MIDI instruments.
- Colored notes to highlight specific note like the sensible.
- Multiple voices.
- Harmonic degrees for harmonic analysis.
- Chord's name (transposable).
- Colored arrows to highlight forced movements.

Files to include in HTML file header

```
<link rel="stylesheet" href="css/music.min.css" />
<script src="js/abc-ui-1.0.0.min.js"></script>
```

To add an abc score in html page, simply use the div tag with the **abc-score** class:

```
<div class="abc-source">
    put abc text here
</div>
```

You can also have abc score text in a JavaScript string variable and use:

```
<div id="js_variable_name" class="abc-source"><div>
```

$ABC_UI Methods

Method	Description
init()	Builds all scores found in html file.
updateTonality()	Updates all transposable scores in a given tonality.
buildScore()	Render a dynamic abc score.

Links on tests using $ABC_UI

- keys
- scores
- bars
- voices
- notes & rests
- durations
- grace notes
- transpositions
- N-olets
- symbols
- texts
- chords
- zoom
- tempos
- colors
- macros
- player
- voices-piano
- voices-chorus
- scales
- MIDI
- drums

$ABC_UI.init(options)

init builds all abc scores found in html file.

Parameter Values

- options.**instruments** MIDI instruments list to load.
- options.**tonality** selected tonality to apply for transposable scores.
- options.**lang** selected language for score player, can be "fr", by default "en".
- options.**bMajorOrMinor true** to support scores with class **major** and **minor**, **false** otherwise.

Examples

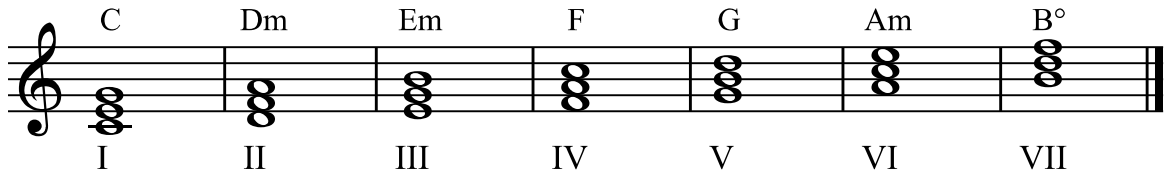
HTML

```
<div class="abc-source">
    Q:1/1=60
    "C"[CEG]|"Dm"[DFA]|"Em"[EGB]|"F"[FAc]|"G"[GBd]|"Am"[Ace]|"B°"[Bdf]|]
    w:I II III IV V VI VII
</div>
```

JavaScript

```
$ABC_UI.init();
```

Result



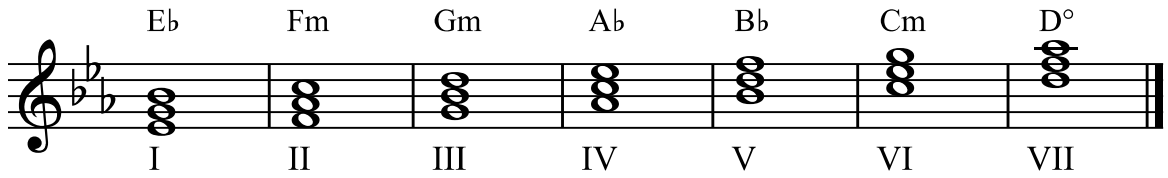
HTML

```
<div class="abc-source transposable">
    Q:1/1=60
    "C"[CEG]|"Dm"[DFA]|"Em"[EGB]|"F"[FAc]|"G"[GBd]|"Am"[Ace]|"B°"[Bdf]|]
    w:I II III IV V VI VII
</div>
```

JavaScript

```
$ABC_UI.init({"tonality": "MEb"});
```

Result



$ABC_UI.updateTonality(tonality, bMajorOrMinor);

updateTonality updates all transposable abc scores in a given tonality.

- **tonality** the selected tonality for update.
- **bMajorOrMinor true** to support scores with class **major** and **minor**, **false** otherwise.

Transposable scores are html elements with:

```
<div class="abc-source transposable">...</div>
<div class="abc-source transposable major">...</div>
<div class="abc-source transposable minor">...</div>
```

$ABC_UI.buildScore(id, abcContent, options)

buildScore render a dynamic abc score.

Parameter Values

- **id** unique identificator of abc score.
- **abcContent** abc score content in string.
- options.**zoom**: real with value by default 1.0
- options.**pagewidth**: real in cm by default 16

---

1 - Score with abc text between div in html page

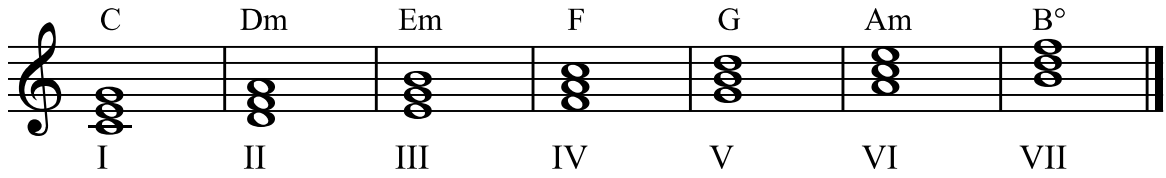
HTML

```
<div class="abc-source">
    Q:1/1=60
    "C"[CEG]|"Dm"[DFA]|"Em"[EGB]|"F"[FAc]|"G"[GBd]|"Am"[Ace]|"B°"[Bdf]|]
    w:I II III IV V VI VII
</div>
```

JavaScript

```
$ABC_UI.init();
```

Result



2 - Score with abc text between div in html page, customised visual

HTML

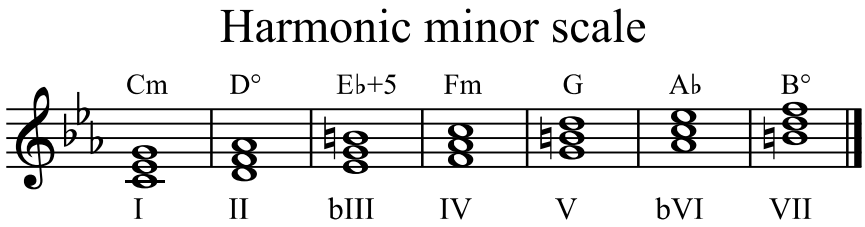
```
<div class="abc-source 12 0.9" align="center">
    T:Harmonic minor scale
    Q:1/1=70
    K:Cm
    "C"[CEG]|"Dm"[DFA]|"Em"[EGB]|"F"[FAc]|"G"[GBd]|"Am"[Ace]|"B°"[Bdf]|]
    w: I II bIII IV V bVI VII
</div>
```

- **12**, first **class** attribut after **abc-source** is the score's length in cm, default value is 16cm.
- **0.9**, second **class** attribut after **abc-source** is the zoom, default value is 1.
- in div use align="center" to center score horizontally.

JavaScript

```
$ABC_UI.init();
```

Result



3 - Score with abc text in a JavaScript string variable

score_1.abc

```
X:1
L:1/1
Q:1/1=60
K:C
"notes"x"C"C"D"D"E"E"F"F"G"G"A"A"B"B"C"c|]
```

In `<div class="abc-source">` add the **id** of the name of the string used in JavaScript.

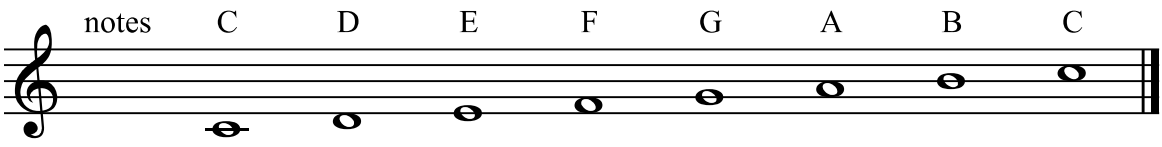
HTML

```
<div id="score_1" class="abc-source"><div>
```

JavaScript

```
var score_1='X:1\nL:1/1\nQ:1/1=60\nK:C\n\
"notes"x"C"C"D"D"E"E"F"F"G"G"A"A"B"B"C"c|]\n';
$ABC_UI.init();
```

Result



4 - Score with abc text in a JavaScript string, customised visual

HTML

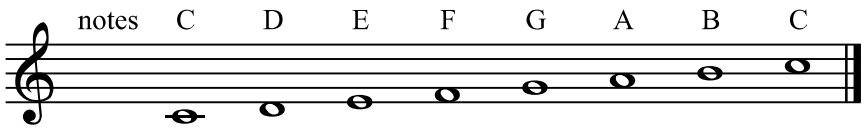
```
<div id="score_2" class="abc-source 12 0.9" align="right"></div>
```

- **12**, first **class** attribut after **abc-source** is the score's length in cm, default value is 16cm.
- **0.9**, second **class** attribut after **abc-source** is the zoom, default value is 1.

JavaScript

```
var score_2 = score_1;
$ABC_UI.init();
```

Result



5 - Score with abc text in a JavaScript, static transposition

HTML

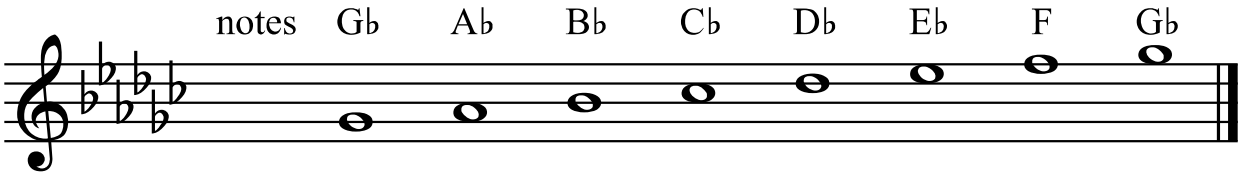
```
<div id="score_3" class="abc-source 17 1.2">Gb</div>
```

- **17**, first **class** attribut after **abc-source** is the score's length in cm.
- **1.2**, second **class** attribut after **abc-source** is the zoom.
- **Gb** between div is the requested tonality (for abc written with C tone).

JavaScript

```
var score_3 = score_1;
$ABC_UI.init();
```

Result



6 - Score with colored notes

Use a abc decoration before the note with that given list of color only: **!**color**!**note

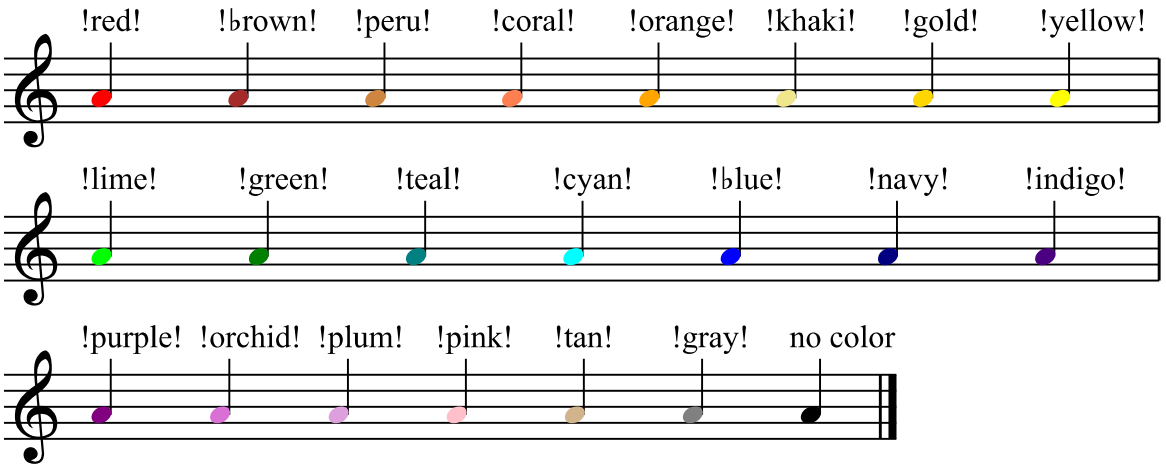
HTML

```
<div class="abc-source">
    L:1/4
    !red!A !brown!A !peru!A !coral!A !orange!A !khaki!A !gold!A !yellow!A |
    !lime!A !green!A !teal!A !cyan!A !blue!A "!navy!"!navy!A !indigo!A |
    !purple!A !orchid!A !plum!A !pink!A !tan!A !gray!A A|]
</div>
```

JavaScript

```
$ABC_UI.init();
```

Result



6 - Score with colored movements

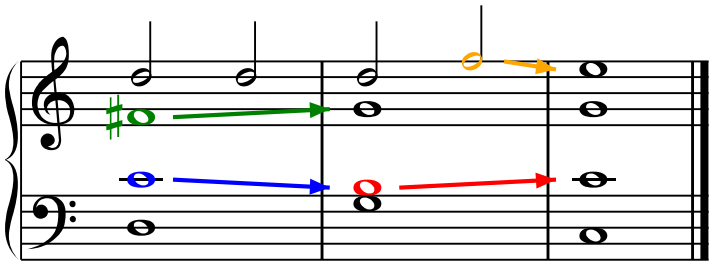
HTML

```
<div class="abc-source">
    %%no_player
    %%score {(1 2) | (3 4)}
    V:1 clef=treble
    V:2 clef=treble
    V:3 clef=bass
    V:4 clef=bass
    [V:1] d/d/           | d/!o>(!!orange!f/ | !o>)!e |]
    [V:2] !g>(!!green!^F | !g>)!G           | G      |]
    [V:3] !blue!!b>(!C   | !b>)!!r>(!!red!B, | !r>)!C |]
    [V:4] D,             | G,               | C,     |]
</div>
```

JavaScript

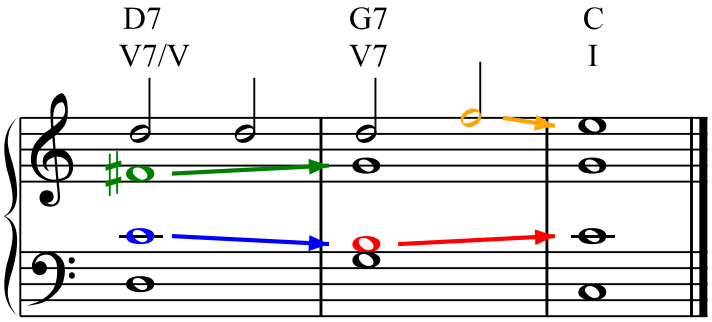
```
$ABC_UI.init();
```

Result



We can write simplest (but non standard abc), using ABC-UI macros:

```
<div class="abc-source">
    %%piano2+2CD
    [V:C] "D7"x          | "G7"x            | "C"x   |]
    [V:D] "V7/V"x        | "V7"x            | "I"x   |]
    [V:S] d/d/           | d/!o>(!!orange!f/ | !o>)!e |]
    [V:A] !g>(!!green!^F | !g>)!G           | G      |]
    [V:T] !blue!!b>(!C   | !b>)!!r>(!!red!B, | !r>)!C |]
    [V:B] D,             | G,               | C,     |]
</div>
```

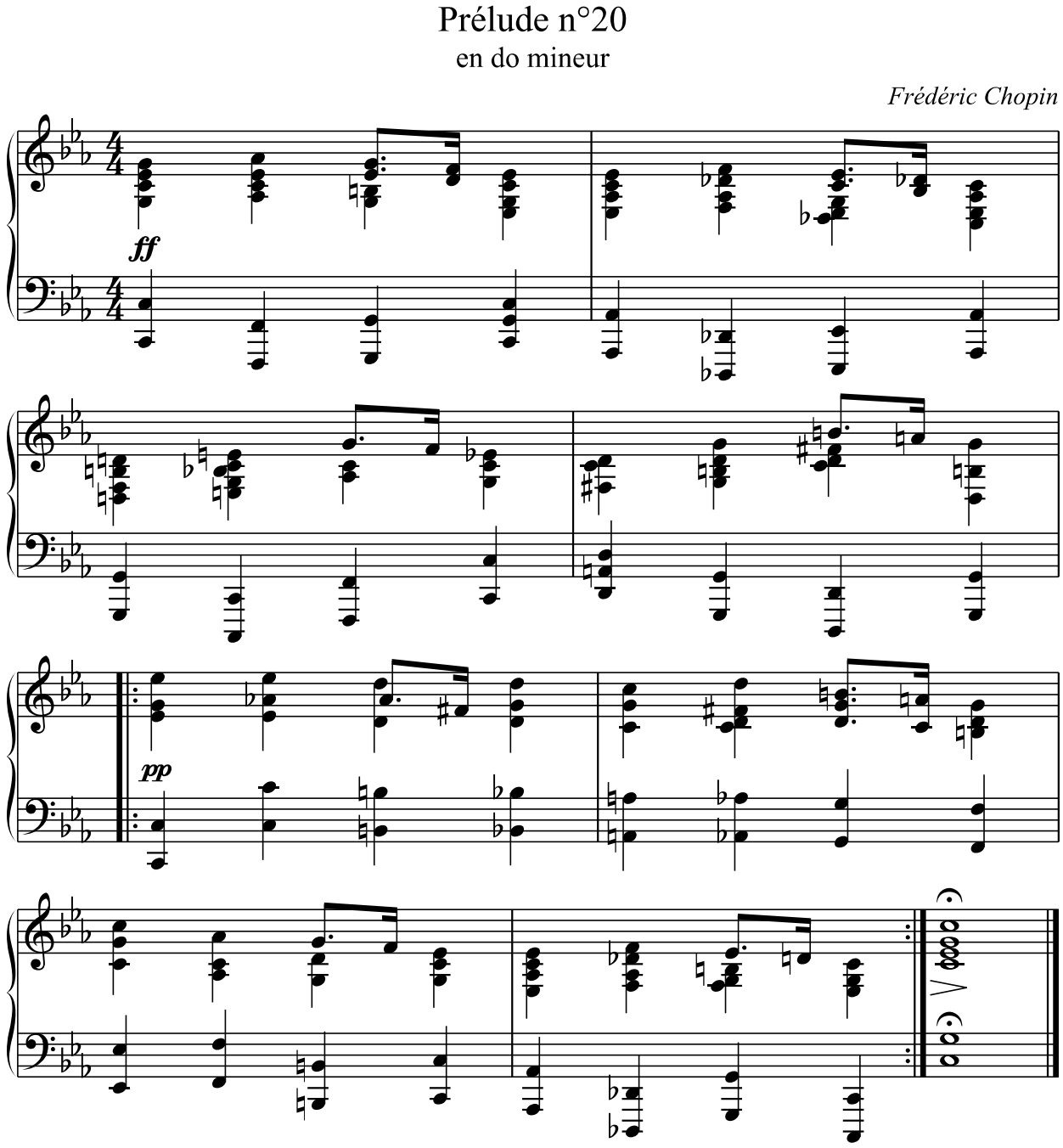


8 - Illustration with a real score

HTML

```
<div id="chopin" class="abc-source 17 0.8"></div>
```

File **chopin.abc**.



$KEYBOARD_UI

This JavaScript module handles piaono keyboards for html pages.

- Module file is: keyboard-ui-1.0.0.min.js.
- It has dependencies: harmony-1.0.0.min.js and music.min.css.

Files to include in HTML file header

```
<link rel="stylesheet" href="css/music.min.css" />

<script src="libs/harmony-1.0.0.min.js"></script>
<script src="libs/keyboard-ui-1.0.0.min.js"></script>
```

$KEYBOARD_UI Methods

Method	Description
init()	Initializes piano drawings.
clear()	Clears note selection for a given keyboard.
selectNotes()	Highlights a set of note for a given keyboard.
selectChord()	Highlights chord's notes for a given keyboard.
selectScale()	Highlights scales's notes for a given keyboard.
selectScale2()	Highlights scale's notes with colored tonal triton.
highlightTonalTriton()	Colorises tonal triton for a given keyboard.
highlightKeyFromNote()	Colorises a key for a given keyboard.

Examples

- Scale for piano
- Chords for piano
- Chords for piano in scale context
- Validation

$KEYBOARD_UI.init()

$KEYBOARD_UI.init();

init() initializes piano keyboard(s) found in html file.
This method is called by default when **keyboard-ui-1.0.0.min.js** module is include.
It can also be called after you add piano dynamically on the page.

To declare piano keyboard add in html file:

```
<div id="an_unique_id" class="keyboard z">n</div>
```

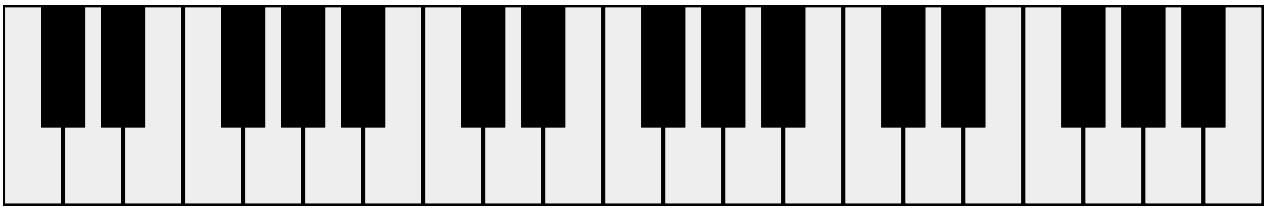
Where :

- **n** is the octave number.
- **z** is the zoom.

Examples

```
<div id="piano1" class="keyboard 1">3</div>
```

Octaves: 3, zoom: 1

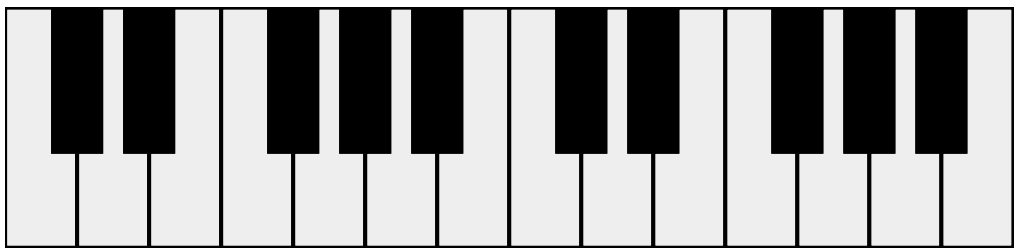


```
<div id="piano2" class="keyboard 0.5" align="center">2</div>
```

Octaves: 2, zoom: 0.5

```
<div id="piano3" class="keyboard 1.2" align="right">2</div>
```

Octaves: 2, zoom: 1.2



```
<div id="piano4" class="keyboard 0.45">7</div>
```

Octaves: 7, zoom: 0.45

$KEYBOARD_UI.clear()

$KEYBOARD_UI.clear(id);

clear() clears note selection for a given keyboard.

- **id** unique piano identificator.

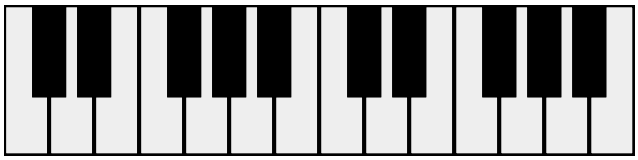
Example

HTML

```
<div id="piano5" class="keyboard .75" align="center">2</div>
```

JavaScript

```
$KEYBOARD_UI.clear('piano5');
```



$KEYBOARD_UI.selectNotes()

$KEYBOARD_UI.selectNotes(id, notes, firstOctave);

selectNotes() highlights a set of note in ascendant order for a given keyboard.

- **id** unique piano identificator.
- **notes** note list in a string, separed by comma and without space.
- **firstOctave** 0 by default, octave of the first note.
- **legend** optional, to add explicative text centered under the piano.
- **colors** optional, list of colors in order of notes.

Examples

HTML

```
<div id="piano-n1" class="keyboard .6" align="center">3</div>
<div id="piano-n2" class="keyboard .6" align="center">3</div>
<div id="piano-n3" class="keyboard .6" align="center">3</div>
<div id="piano-n4" class="keyboard .6" align="center">3</div>
```

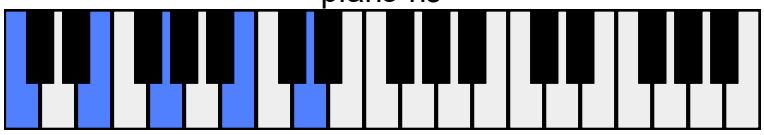
JavaScript

```
$KEYBOARD_UI.selectNotes('piano-n1', 'C,E,F,A,Bb,D,G', 1);
$KEYBOARD_UI.selectNotes('piano-n2', 'G,G,C,Eb,F#,A#');
$KEYBOARD_UI.selectNotes('piano-n3', 'C, E, G, B, D', 0, 'my legend');
$KEYBOARD_UI.selectNotes('piano-n4', 'C, Eb, G, Bb, D, F#, Ab, C, Eb', 0,
    'with colors', ['red', 'lightred', 'blue', 'lightblue', 'green',
    'lightgreen', 'orange', 'gray']);
```

piano-n1

piano-n2

piano-n3



my legend

piano-n4

with colors

$KEYBOARD_UI.selectChord()

$KEYBOARD_UI.selectChord(id, chord, firstOctave);

selectChord highlights chord's notes for a given keyboard.

- **id** unique piano identificator.
- **chord** chordname in string.
- **firstOctave** 0 by default, octave of the first note.

Examples

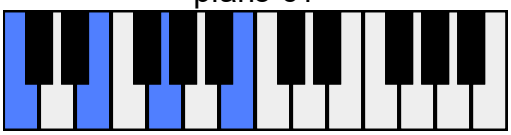
HTML

```
<div id="piano-c1" class="keyboard 0.6">2</div>
<div id="piano-c2" class="keyboard 0.6">2</div>
<div id="piano-c3" class="keyboard 0.6">2</div>
<div id="piano-c4" class="keyboard 0.6">2</div>
<div id="piano-c5" class="keyboard 0.6">2</div>
<div id="piano-c6" class="keyboard 0.6">2</div>
```

JavaScript

```
$KEYBOARD_UI.selectChord('piano-c1', 'CM7');
$KEYBOARD_UI.selectChord('piano-c2', 'G7');
$KEYBOARD_UI.selectChord('piano-c3', 'Dm7/F');
$KEYBOARD_UI.selectChord('piano-c4', 'C/A');
$KEYBOARD_UI.selectChord('piano-c5', 'E°7');
$KEYBOARD_UI.selectChord('piano-c6', 'A');
```

piano-c1

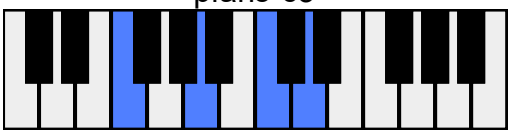


CM7 : C, E, G, B

piano-c2

G7 : G, B, D, F

piano-c3

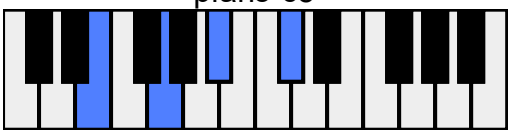


Dm7/F : F, A, C, D

piano-c4

C/A : A, C, E, G

piano-c5



E°7 : E, G, Bb, Db

piano-c6

A : A, C#, E

$KEYBOARD_UI.selectScale()

$KEYBOARD_UI.selectScale(id, mode, tone, firstOctave);

selectScale highlights scale's notes for a given keyboard.

- **id** unique piano identificator.
- **mode** a number for scale:
  - 0: major,
  - 7: melodic minor,
  - 14: harmonic minor,
  - 21: natural minor.
- **firstOctave** 0 by default, octave of the first note.

Examples

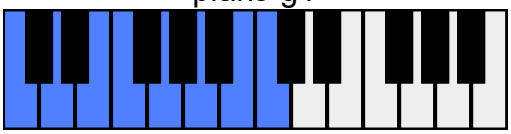
HTML

```
<div id="piano-g1" class="keyboard 0.6">3</div>
<div id="piano-g2" class="keyboard 0.6">3</div>
<div id="piano-g3" class="keyboard 0.6">3</div>
<div id="piano-g4" class="keyboard 0.6">3</div>
```

JavaScript

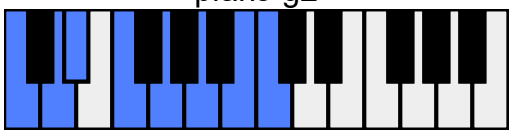
```
$KEYBOARD_UI.selectScale('piano-g1', 0, 'C', 3);
$KEYBOARD_UI.selectScale('piano-g2', 7, 'C', 3);
$KEYBOARD_UI.selectScale('piano-g3', 14, 'C', 3);
$KEYBOARD_UI.selectScale('piano-g4', 21, 'C', 3);
```

piano-g1



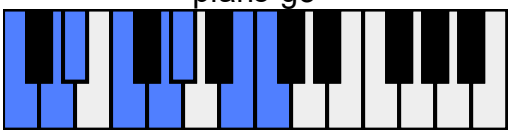
C major

piano-g2



C melodic minor

piano-g3



C harmonic minor

piano-g4

C natural minor

$KEYBOARD_UI.selectScale2()

$KEYBOARD_UI.selectScale2(id, mode, tone, firstOctave);

selectScale2 highlights scale's notes with colored tonal triton for a given keyboard.

- **id** unique piano identificator.
- **mode** a number for scale:
  - 0: major,
  - 7: melodic minor,
  - 14: harmonic minor,
  - 21: natural minor.
- **firstOctave** 0 by default, octave of the first note.

Examples

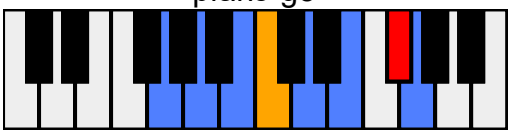
HTML

```
<div id="piano-g5" class="keyboard 0.6">3</div>
<div id="piano-g6" class="keyboard 0.6">3</div>
<div id="piano-g7" class="keyboard 0.6">3</div>
<div id="piano-g8" class="keyboard 0.6">3</div>
```

JavaScript

```
$KEYBOARD_UI.selectScale('piano-g5', 0, 'G', 3);
$KEYBOARD_UI.selectScale('piano-g6', 7, 'G', 3);
$KEYBOARD_UI.selectScale('piano-g7', 14, 'G', 3);
$KEYBOARD_UI.selectScale('piano-g8', 21, 'G', 3);
```

piano-g5

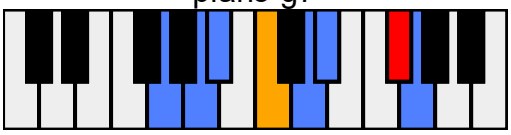


G major

piano-g6

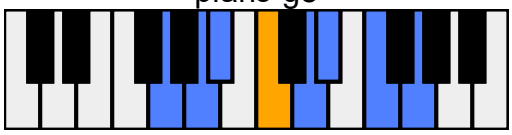
G melodic minor

piano-g7



G harmonic minor

piano-g8



G natural minor

$KEYBOARD_UI.highlightTonalTriton()

$KEYBOARD_UI.highlightTonalTriton(id, tone);

highlightTonalTriton colorises tonal triton for a given keyboard.

- **id** unique piano identificator.
- **tone** tone of selected scale.

showTonalTriton must be call after an action.
Warning, **clear** disables tonal triton highlight.

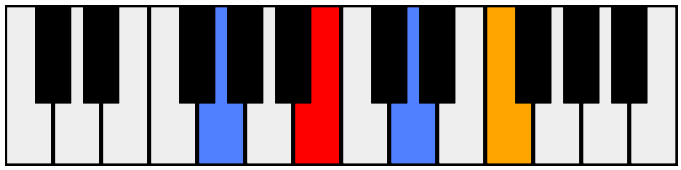
Examples

HTML

```
<div id="piano-tt1" class="keyboard 0.8" align="center">2</div>
```

JavaScript

```
$KEYBOARD_UI.selectChord('piano-tt1', 'G7');
$KEYBOARD_UI.highlightTonalTriton('piano-tt1', 'C');
```



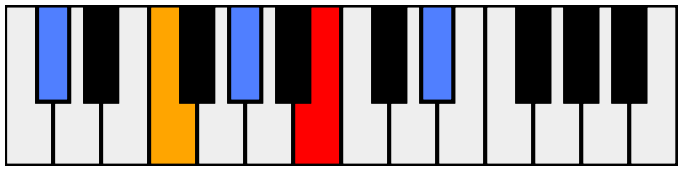
G7 : G, B, D, F

HTML

```
<div id="piano-tt2" class="keyboard 0.8" align="center">2</div>
```

JavaScript

```
$KEYBOARD_UI.selectNotes('piano-tt2', 'Db,F,Ab,B,Eb');
$KEYBOARD_UI.highlightTonalTriton('piano-tt2', 'C');
```



$KEYBOARD_UI.highlightKeyFromNote()

$KEYBOARD_UI.highlightKeyFromNote(id, note, firstOctave, pattern);

highlightKeyFromNote Colorises a key for a given keyboard.

- **id** unique piano identificator.
- **note** note to highlight.
- **firstOctave** 0 by default, octave of the first note.
- **pattern** string for color in ['red', 'lightred', 'blue', 'lightblue', 'green', 'lightgreen', 'orange', 'gray', 'optional'].

showTonalTriton must be call after an action.
Warning, **clear** disables tonal triton highlight.

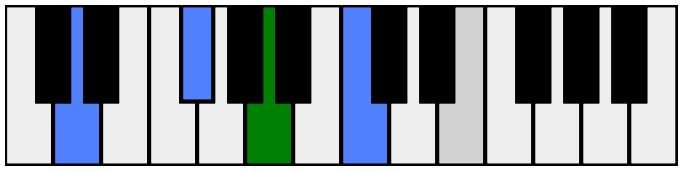
Example

HTML

```
<div id="piano-tt3" class="keyboard 0.8" align="center">2</div>
```

JavaScript

```
$KEYBOARD_UI.selectChord('piano-tt3', 'D7');
$KEYBOARD_UI.highlightKeyFromNote('piano-tt3', 'A', 0);
$KEYBOARD_UI.highlightKeyFromNote('piano-tt3', 'E', 1, "optional");
```



D7 : D, F#, A, C

Scale for piano

HTML

```
<script src="libs/music-ui-1.0.0.min.js"></script>
<div align="center">
    Select a tonality:
    <div id="tonalities_1"></div><br>
    <div id="piano-scale_1" class="keyboard 0.4">2</div>
</div>
```

JavaScript

```
function updatePianoScale1(tonality) {
    var tm = $HARMONY.getToneAndMinor(tonality);
    var g = tm.minor ? 14 : 0;
    $KEYBOARD_UI.selectScale('piano-scale_1', g, tm.tone);
}
var tonalitySelection1 = "MC";
$MUSIC_UI.buildTonalities('tonalities_1', tonalitySelection1, updatePianoScale1);
updatePianoScale1(tonalitySelection1);
```

Select a tonality:

Cb	Gb	Db	Ab	Eb	Bb	F	C	G	D	A	E	B	F#	C#
Abm	Ebm	Bbm	Fm	Cm	Gm	Dm	Am	Em	Bm	F#m	C#m	G#m	D#m	A#m

HTML

```
<script src="libs/music-ui-1.0.0.min.js"></script>
<div align="center">
    Select a tonality:
    <div id="tonalities_2"></div><br>
    <div id="piano-scale_2" class="keyboard 0.5">4</div>
</div>
```

JavaScript

```
function updatePianoScale2(tonality) {
    var tm = $HARMONY.getToneAndMinor(tonality);
    var g = tm.minor ? 14 : 0;
    $KEYBOARD_UI.selectScale2('piano-scale_2', g, tm.tone, 1);
}
var tonalitySelection2 = "mA";
$MUSIC_UI.buildTonalities('tonalities_2', tonalitySelection2, updatePianoScale2);
updatePianoScale2(tonalitySelection2);
```

Select a tonality:

Cb	Gb	Db	Ab	Eb	Bb	F	C	G	D	A	E	B	F#	C#
Abm	Ebm	Bbm	Fm	Cm	Gm	Dm	Am	Em	Bm	F#m	C#m	G#m	D#m	A#m

Chords for piano #1

HTML

```
<div align="center">
    Select a tone: <div id="tones_1"></div><br>
    Select a signature: <div id="signatures_1"></div><br>
    <div id="piano-chord_1" class="keyboard 0.8">3</div>
</div>
```

JavaScript

```
var tone1 = 'C', signature1 = 'm';
var signatures_1 = [
    'Maj', 'm', '°', 'mb5', '+5',
    '!7', 'M7', 'Δ7', 'm7', 'mM7', '°7',
    '!m7b5', 'Ø7', '7b5', '7#5', 'M7#5', 'dim7',
    '!sus2', 'sus4', '7sus4', '6', 'm6', 'add9'
];
function updatePianoChord1() {
    var chord = tone1 + signature1;
    $KEYBOARD_UI.selectChord("piano-chord_1", chord, 1);
}
function updateChordTone1(tone) {
    tone1 = tone;
    updatePianoChord1();
}
function updateChordSignature1(signature) {
    signature1 = signature;
    updatePianoChord1();
}
$MUSIC_UI.buildTones('tones_1', tone1, updateChordTone1);
$MUSIC_UI.buildSignatures('signatures_1', signatures_1, 1, updateChordSignature1);
updatePianoChord1();
```

Select a tone:

Cb C C# Db D D# Eb E E#

Fb F F# Gb G G# Ab A A# Bb B B#

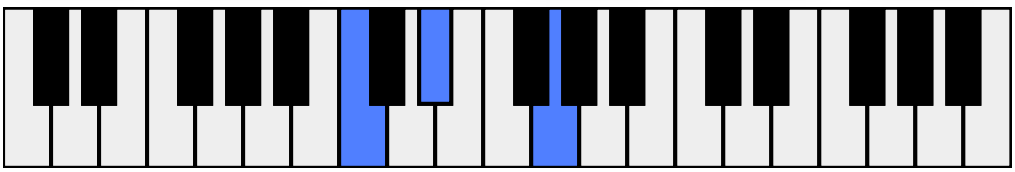
Select a signature:

Maj m ° mb5 +5

7 M7 Δ7 m7 mM7 °7

m7b5 Ø7 7b5 7#5 M7#5 dim7

sus2 sus4 7sus4 6 m6 add9



Cm : C, Eb, G

Chords for piano in scale context

HTML

```
<div align="center">
    Select scale tonality: <div id="tonalities"></div>
    Tonal triton is <span id="quarte" class="T4"></span> and <span id="sensible
    <br><br>Select chord tone: <div id="tones_2"></div>
    <br>Select chord signature: <div id="signatures_2"></div><br>
    <div id="piano-chord_2" class="keyboard .5">2</div>
</div>
```

JavaScript

```
var tonality2 = 'MC', tone2 = 'G', signature2 = '7', scaleTone = 'C';
var signatures_2 = ['Maj', 'm', '°', '+5', '7', 'M7', 'm7', '°7'];
function updatePianoChord2() {
    var chord = tone2 + signature2;
    $KEYBOARD_UI.selectChord("piano-chord_2", chord);
    $KEYBOARD_UI.highlightTonalTriton("piano-chord_2", scaleTone);
}
function updateChordTonality2(tonality) {
    scaleTone = $HARMONY.getToneOfTonality(tonality);
    document.querySelector('#quarte').innerHTML = 'quarte ' + $HARMONY.getRelativeNote
    document.querySelector('#sensible').innerHTML = 'sensible ' + $HARMONY.getRelativ
    updatePianoChord2();
}
function updateChordTone2(tone) { tone2 = tone; updatePianoChord2(); }
function updateChordSignature2(signature) { signature2 = signature; updatePianoChc
$MUSIC_UI.buildTonalities('tonalities', tonality2, updateChordTonality2);
$MUSIC_UI.buildTones('tones_2', tone2, updateChordTone2);
$MUSIC_UI.buildSignatures('signatures_2', signatures_2, 0, updateChordSignature2);
updateChordTonality2(tonality2);
```

Select scale tonality:

Cb	Gb	Db	Ab	Eb	Bb	F	C	G	D	A	E	B	F#	C#
Abm	Ebm	Bbm	Fm	Cm	Gm	Dm	Am	Em	Bm	F#m	C#m	G#m	D#m	A#m

Tonal triton is **quarte F** and **sensible B**.

Select chord tone:

Cb	C	C#	Db	D	D#	Eb	E	E#
Fb	F	F#	Gb	G	G#	Ab	A	A#	Bb	B	B#

Select chord signature:

Maj	m	°	+5	7	M7	m7	°7

GMaj : G, B, D

$INTERVALS_UI

This JavaScript module handles musical intervals for html pages.

- Module file is: intervals-ui-1.0.0.min.js.
- It has dependency: harmony-1.0.0.min.js.

Files to include in HTML file header

```
<script src="libs/harmony-1.0.0.min.js"></script>
<script src="libs/intervals-ui-1.0.0.min.js"></script>
```

$INTERVALS_UI Methods

Method	Description
init()	Initializes the library to setup language.
buildPentacord()	Shows a pentacord.
buildTetracord()	Shows a tretacord.
buildModeSteps()	Shows degrees of choosen mode.
buildModeSteps2()	Shows degrees of choosen mode.
buildChordComposition()	Shows chord's notes.
buildIntervals()	Shows intervals selector.
buildInterval()	shows interval with note or degree.
buildIntervalsFromTone()	Shows intervals from a tone.
buildSimpleFifthCircle()	Shows simple fifth cycle.
buildFifthCircle()	Shows full fifth cycle.

Examples

- Validation
- Quints cycles

$INTERVALS_UI.init()

$INTERVALS_UI.init(options);

init() function is used to setup the language.

- **options** { 'lang': value }, value is **'en'** for english or **'fr'** for french.

Example

JavaScript

```
$INTERVALS_UI.init({'lang':'en'});
```

$INTERVALS_UI.buildPentacord()

$INTERVALS_UI.buildPentacord(id, mode, tone, bStep);

buildPentacord() shows pentacord.

- **id** unique identificator for pentacord.
- **mode** is number for musical mode.
  - mode = 0 for **major** pentacord,
  - mode = 1 for **minor** pentacord,
  - mode = 18 for **harmonic** pentacord,
  - mode = 2 for **phrygian** pentacord,
  - mode = 3 for **lydian** pentacord,
  - mode = 20 for **diminished** pentacord.
- **tone** tone of the first note.
- **bStep true** to show number of semi-tone between intervals, **false** to hide them.

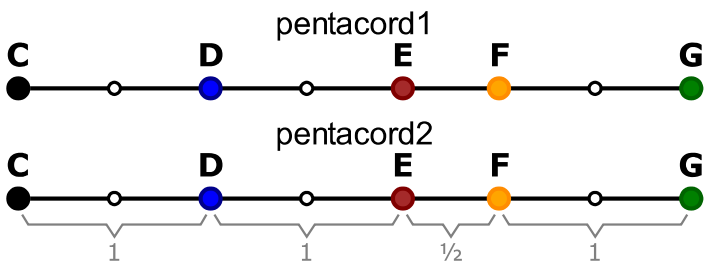
Examples

HTML

```
<div id="pentacord1"></div>
<div id="pentacord2"></div>
```

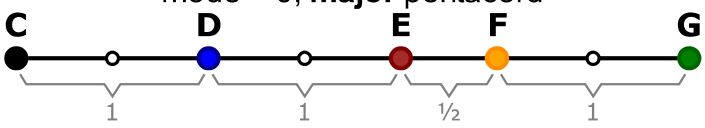
JavaScript

```
$INTERVALS_UI.buildPentacord('pentacord1', 0, 'C', false);
$INTERVALS_UI.buildPentacord('pentacord2', 0, 'C', true);
```

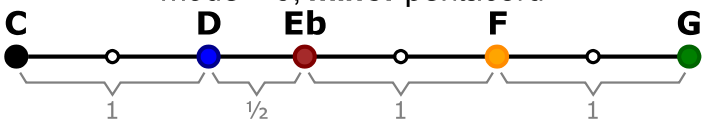


Pentacord list

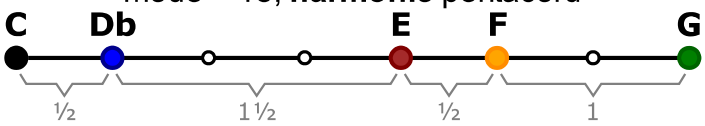
mode = 0, **major** pentacord



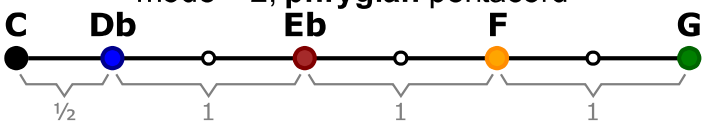
mode = 5, **minor** pentacord



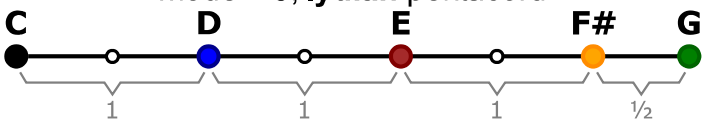
mode = 18, **harmonic** pentacord



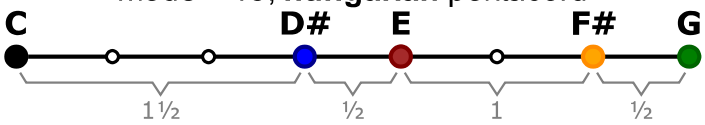
mode = 2, **phrygian** pentacord



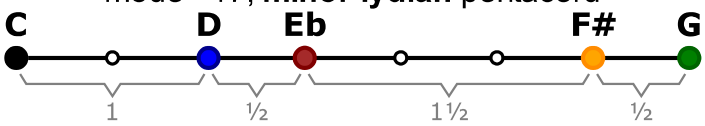
mode = 3, **lydian** pentacord



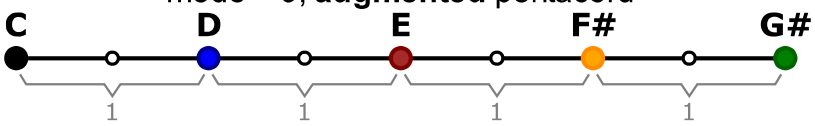
mode = 19, **hungarian** pentacord



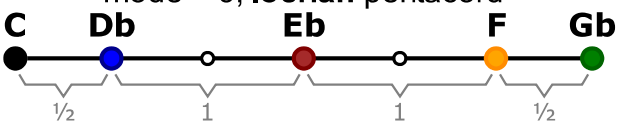
mode =17, **minor-lydian** pentacord



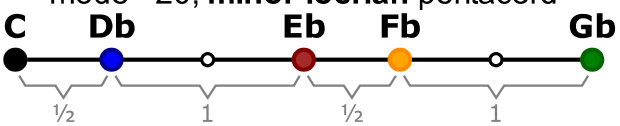
mode = 9, **augmented** pentacord



mode = 6, **locrian** pentacord



mode =20, **minor-locrian** pentacord



$INTERVALS_UI.buildTetracord()

$INTERVALS_UI.buildTetracord(id, mode, tone, bStep);

buildTetracord() shows tetrachord.

- **id** unique identificator for tetrachord.
- **mode** is number for musical mode.
  - mode = 0 for **major** tetrachord,
  - mode = 1 for **minor** tetrachord,
  - mode = 18 for **harmonic** tetrachord,
  - mode = 2 for **phrygian** tetrachord,
  - mode = 3 for **lydian** tetrachord,
  - mode = 20 for **diminished** tetrachord.
- **tone** tone of the first note.
- **bStep true** to show number of semi-tone between intervals, **false** to hide them.

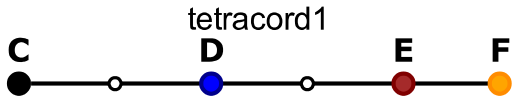
Examples

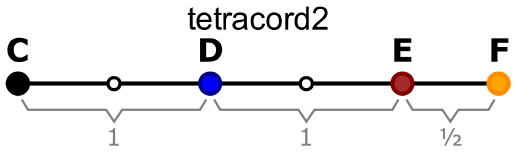
HTML

```
<div id="tetracord1"></div> <div id="tetracord2"></div>
```

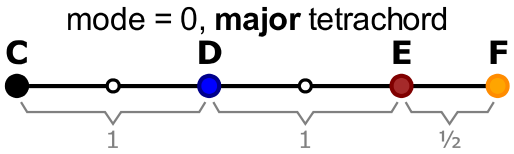
JavaScript

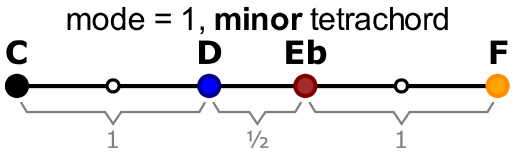
```
$INTERVALS_UI.buildTetracord('tetracord1', 0, 'C', false);
$INTERVALS_UI.buildTetracord('tetracord2', 0, 'C', true);
```

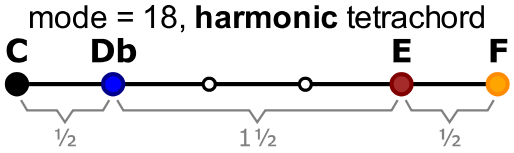


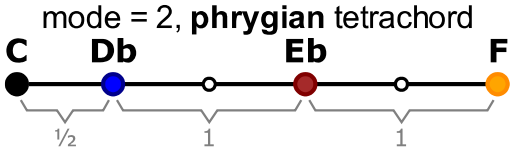


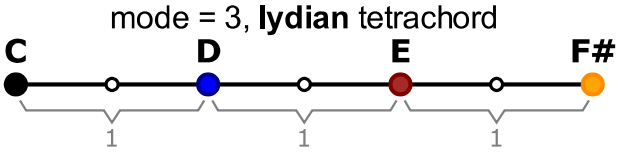
Tetrachord list

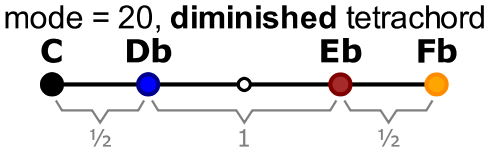












$INTERVALS_UI.buildModeSteps()

$INTERVALS_UI.buildModeSteps(id, mode, tone, pentacord, tetracord);

buildModeSteps() shows degrees of seven-tone musical scale with steps in half tone at bottom.

- **id** unique identificator of scale.
- **mode** selected musical mode.
- **tone** tone of the first note.
- **pentacord** pentacord name as string.
- **tetracord** tetracord name as string.

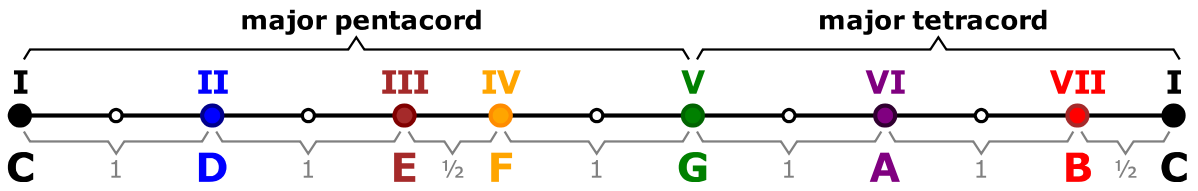
Examples

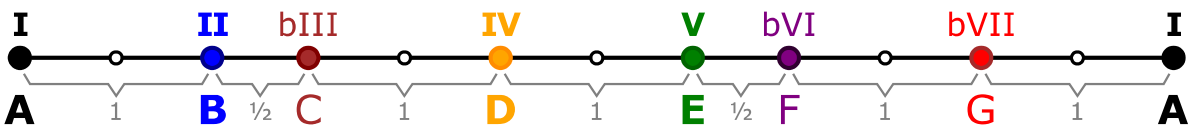
HTML

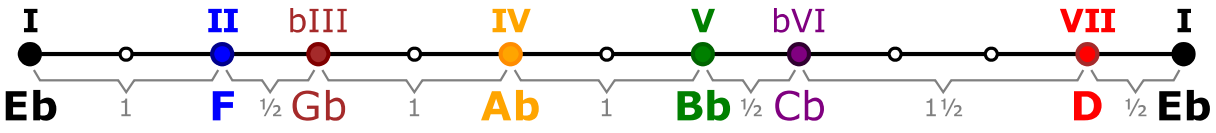
```
<div id="major"></div>
<div id="natural"></div>
<div id="harmonic"></div>
<div id="melodic"></div>
```

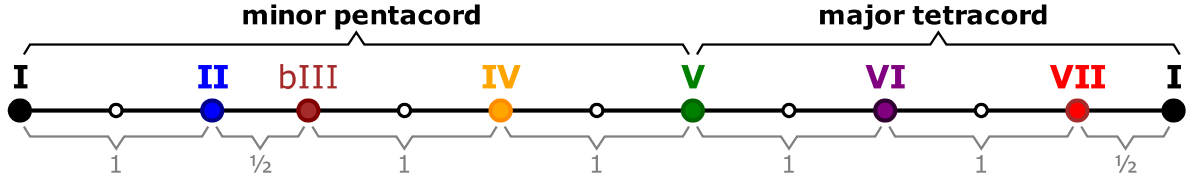
JavaScript

```
$INTERVALS_UI.buildModeSteps('major', 0, 'C', 'major', 'major');
$INTERVALS_UI.buildModeSteps('natural', 21, 'A');
$INTERVALS_UI.buildModeSteps('harmonic', 14, 'Eb');
$INTERVALS_UI.buildModeSteps('melodic', 7, 'I', 'minor', 'major');
```









$INTERVALS_UI.buildModeSteps2()

$INTERVALS_UI.buildModeSteps2(id, mode, tone, pentacord, tetracord);

buildModeSteps2() shows degrees of seven-tone musical scale with steps in half tone on top.

- **id** unique identificator of scale.
- **mode** selected musical mode.
- **tone** tone of the first note.
- **pentacord** pentacord name as string.
- **tetracord** tetracord name as string.

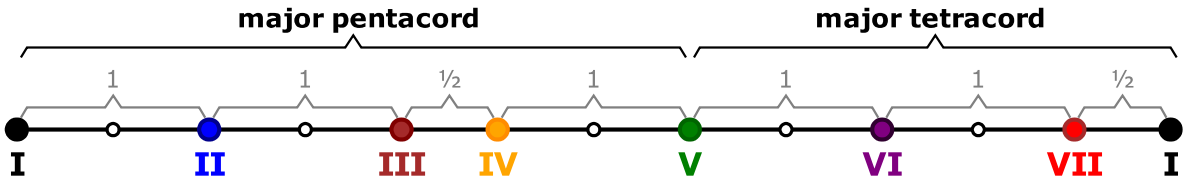
Examples

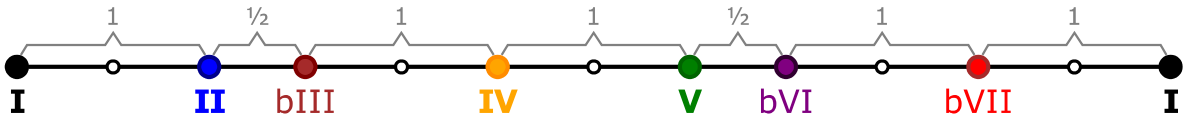
HTML

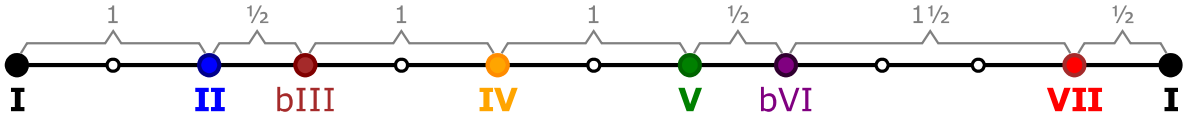
```
<div id="major2"></div>
<div id="natural2"></div>
<div id="harmonic2"></div>
<div id="melodic2"></div>
```

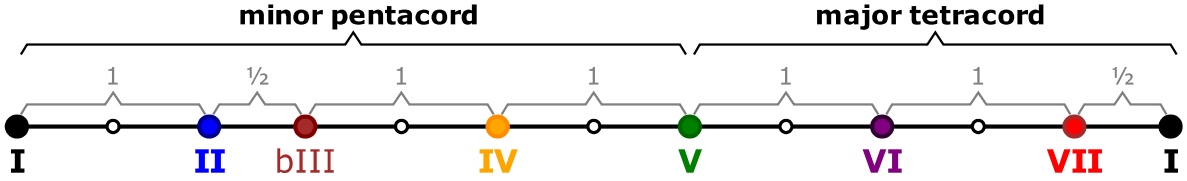
JavaScript

```
$INTERVALS_UI.buildModeSteps2('major2', 0, 'C', 'major', 'major');
$INTERVALS_UI.buildModeSteps2('natural2', 21, 'A');
$INTERVALS_UI.buildModeSteps2('harmonic2', 14, 'Eb');
$INTERVALS_UI.buildModeSteps2('melodic2', 7, 'I', 'minor', 'major');
```









$INTERVALS_UI.buildChordComposition()

$INTERVALS_UI.buildChordComposition(id, chord);

buildChordComposition() shows list of notes and intervals for a chord.

- **id** unique identificator of chord.
- **chord** the selected chord as string.

Example

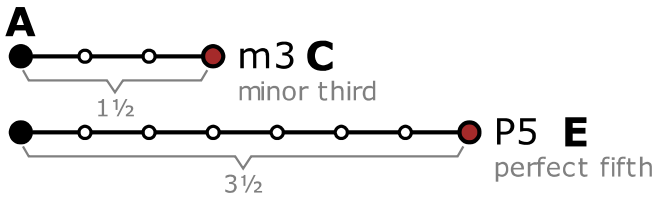
HTML

```
<div id="chord1"></div>
<div id="chord2"></div>
<div id="chord3"></div>
```

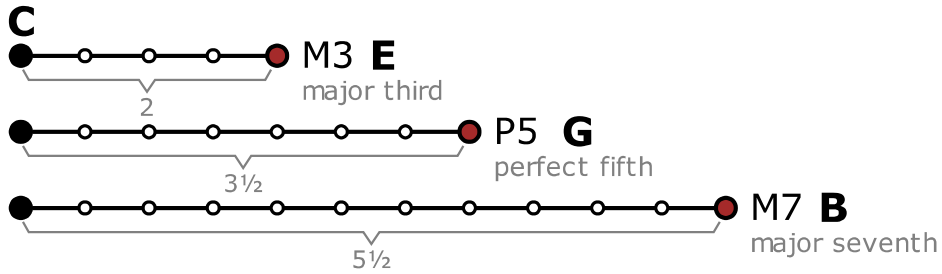
JavaScript

```
$INTERVALS_UI.buildChordComposition('chord1', 'Am');
$INTERVALS_UI.buildChordComposition('chord2', 'CM7');
$INTERVALS_UI.buildChordComposition('chord3', 'F#°7');
```

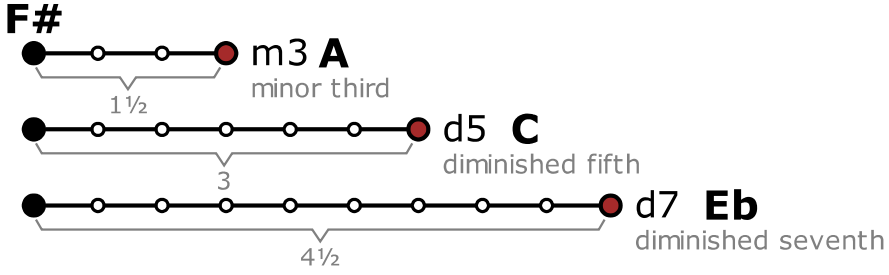
Am chord:



CM7 chord:



F#°7 chord:



$INTERVALS_UI.buildIntervals()

$INTERVALS_UI.buildIntervals(id, selection, cb_update);

buildIntervals() shows intervals selector.

- **id** unique identificator of interval doc.
- **selection** the selected interval.
- **cb_update** callback tha is called when selection changes.

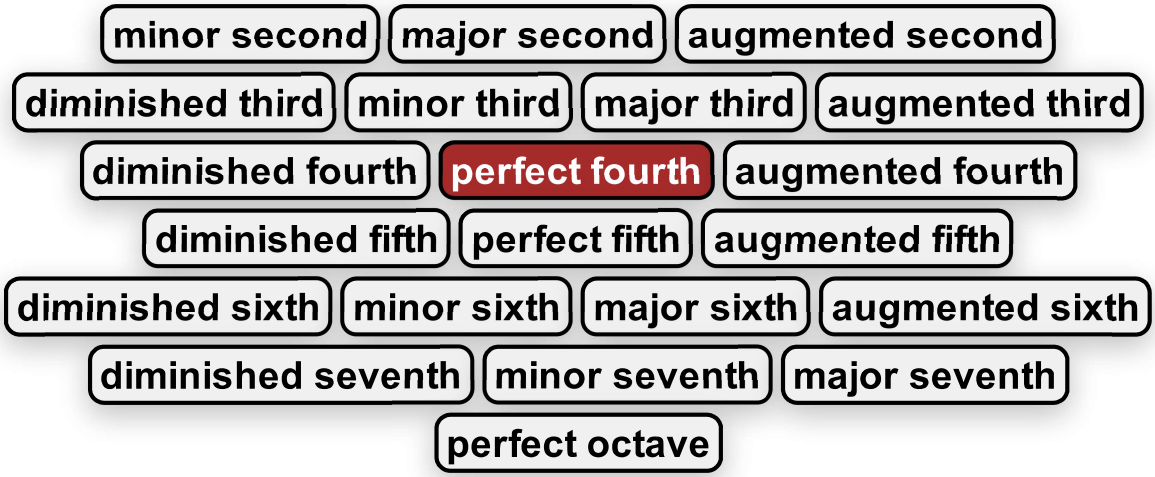
Example

HTML

```
<div align="center" id="intervalSelector"></div>
<p>Selected interval is: <b><span id="selectedInterval"></span></b>.</p>
```

JavaScript

```
function _updateInterval(interval) {
    document.querySelector("#selectedInterval").innerHTML = interval;
}
$INTERVALS_UI.buildIntervals('intervalSelector', 'P4', _updateInterval);
```



Selected interval is: **P4**.

$INTERVALS_UI.buildInterval()

$INTERVALS_UI.buildInterval(id, tone, interval);

buildInterval() shows interval with note or degree.

- **id** unique identificator of interval doc.
- **tone** the first note or **'I'** for degree.
- **interval** the interval.

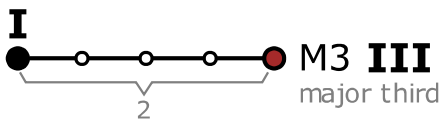
Examples

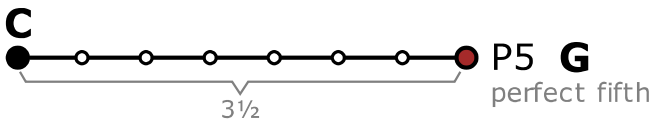
HTML

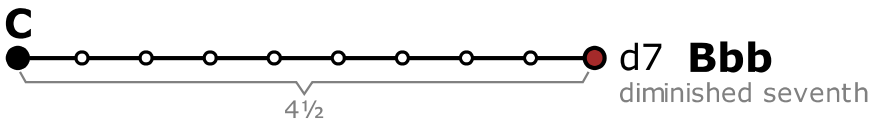
```
<div id="interval1"></div>
<div id="interval2"></div>
<div id="interval3"></div>
<div id="interval4"></div>
```

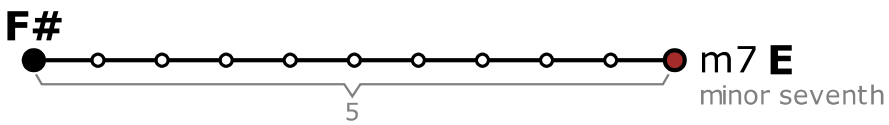
JavaScript

```
$INTERVALS_UI.buildInterval('interval1', 'I', 'M3');
$INTERVALS_UI.buildInterval('interval2', 'C', 'P5');
$INTERVALS_UI.buildInterval('interval3', 'C', 'd7');
$INTERVALS_UI.buildInterval('interval3', 'F#', 'm7');
```









$INTERVALS_UI.buildIntervalsFromTone()

$INTERVALS_UI.buildIntervalsFromTone(id, tone);

buildIntervalsFromTone() shows intervals with notes or degrees.

- **id** unique identificator of interval doc.
- **tone** the first note or **'I'** for degrees.

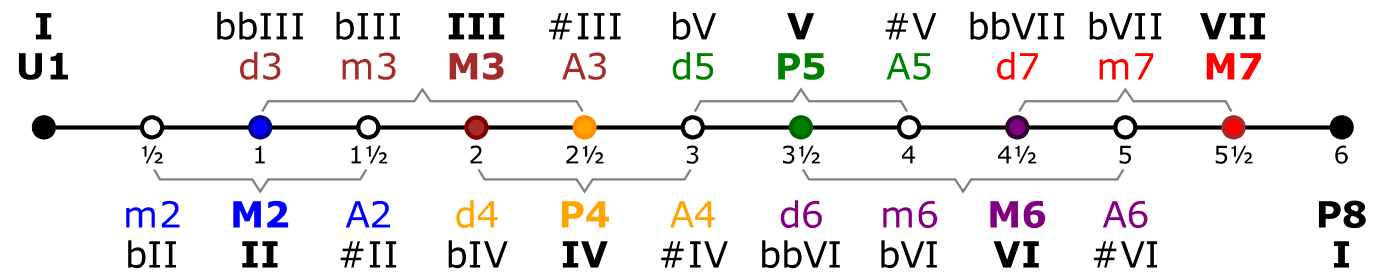
Example

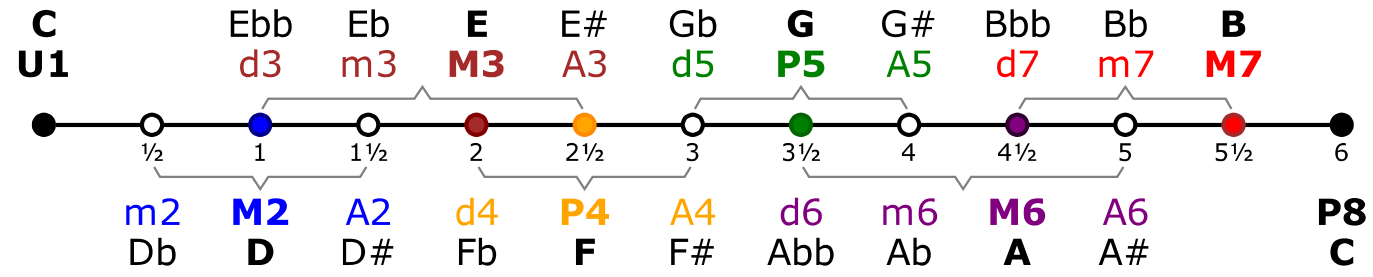
HTML

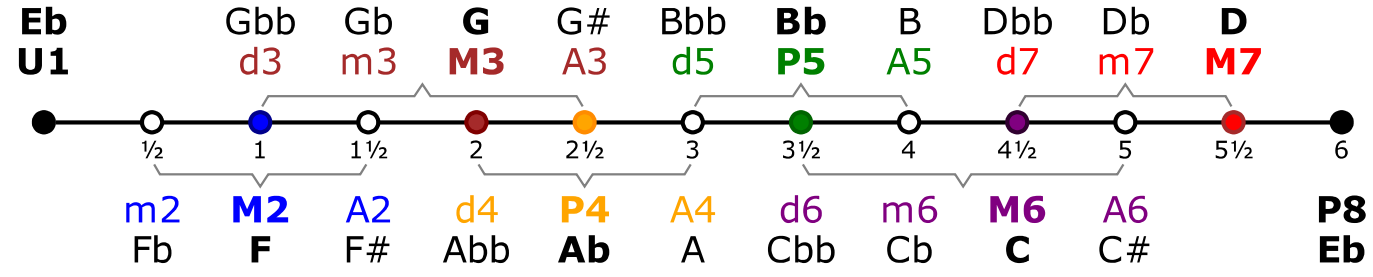
```
<div id="intervals1"></div>
<div id="intervals2"></div>
<div id="intervals3"></div>
```

JavaScript

```
$INTERVALS_UI.buildIntervalsFromTone('intervals1', 'I');
$INTERVALS_UI.buildIntervalsFromTone('intervals2', 'C');
$INTERVALS_UI.buildIntervalsFromTone('intervals3', 'Eb');
```







$INTERVALS_UI.buildSimpleFifthCircle()

$INTERVALS_UI.buildSimpleFifthCircle(id, tone);

buildSimpleFifthCircle() shows simple fifth circle.

- **id** unique identificator of interval doc.
- **tone** the first note.

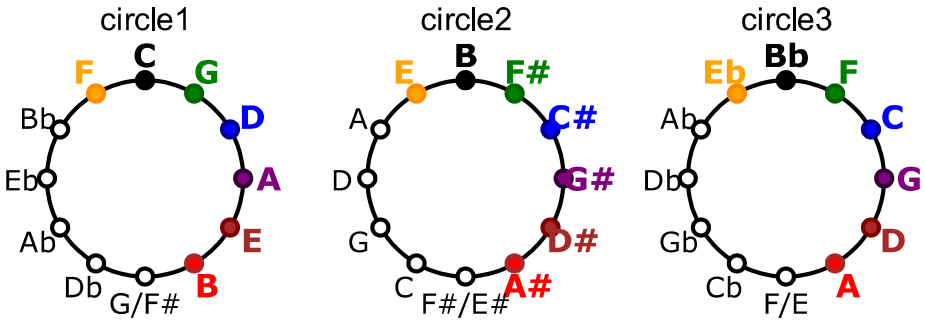
Examples

HTML

```
<div id="circle1"></div>
<div id="circle2"></div>
<div id="circle3"></div>
```

JavaScript

```
$INTERVALS_UI.buildSimpleFifthCircle("circle1", "C");
$INTERVALS_UI.buildSimpleFifthCircle("circle2", "B");
$INTERVALS_UI.buildSimpleFifthCircle("circle3", "Bb");
```



$INTERVALS_UI.buildFifthCircle()

$INTERVALS_UI.buildFifthCircle(id, tone);

buildFifthCircle() shows full fifth circle.

- **id** unique identificator of interval doc.
- **tone** the first note.

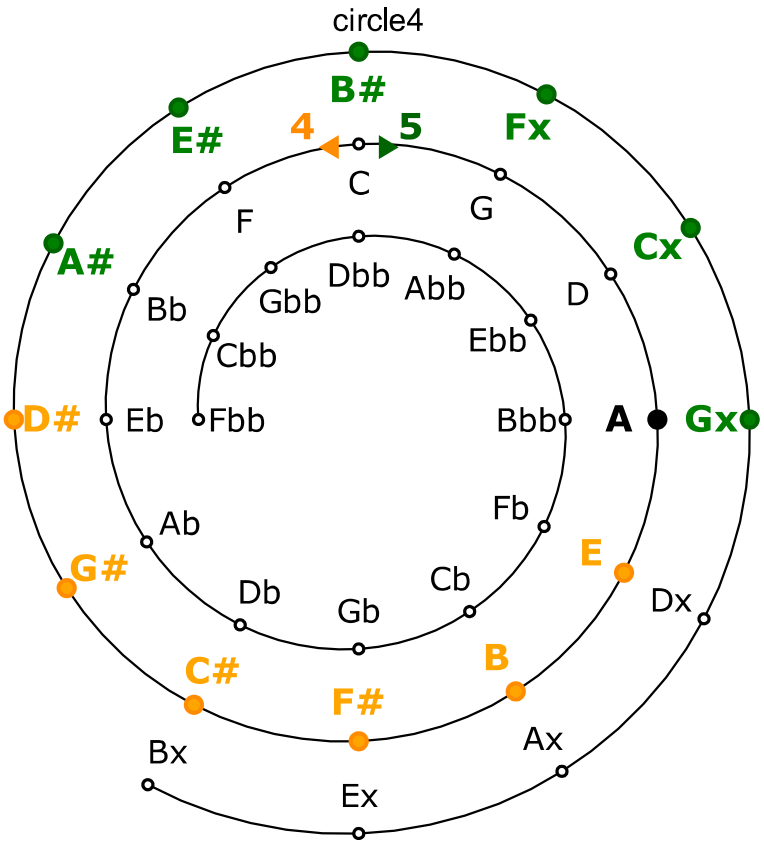
Example

HTML

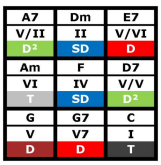
```
<div id="circle4"></div>
```

JavaScript

```
$INTERVALS_UI.buildFifthCircle("circle4", "A");
```



$SEQUENCE_UI



harmonic sequence editor with text format

This JavaScript module handles transposable harmonic sequences for html pages.

- Module file is: sequence-ui-1.0.0.min.js.
- It has dependencies: harmony-1.0.0.min.js and music.min.css.

Files to include in HTML file header

```
<link rel="stylesheet" href="css/music.min.css" />

<script src="libs/harmony-1.0.0.min.js"></script>
<script src="libs/sequence-ui-1.0.0.min.js"></script>
```

$SEQUENCE_UI Methods

Method	Description
init()	Builds all harmonic sequences found in html file.
updateTonality()	Updates transposable harmonic sequences with given tonality.
buildSequencesFromList()	Shows harmonic sequences from sequence list.
buildHarmonization()	Shows harmonization in a given mode.

Examples

- Validation.
- Tonalities harmonisations.
- Scales's harmonisations.
- Transposition.
- Sequences with major or minor.
- Where are triads of this tone?
- Where are tetrads of this tone?

$SEQUENCE_UI.init()

$SEQUENCE_UI.init(options);

init() Builds all harmonic sequences found in html file.

```
options {
    'tonality': 'MC',
    'bMajorOrMinor': true or false,
    'lang': 'en' or 'fr' for language
}
```

Example

JavaScript

```
$SEQUENCE_UI.init({'lang':'en'});
```

$SEQUENCE_UI.updateTonality()

$SEQUENCE_UI.updateTonality(tonality, bMajorOrMinor);

updateTonality() Updates transposable harmonic sequences with given tonality.

- **tonality** the selected tonality for update.
- **bMajorOrMinor true** to support sequences with class **major** and **minor**, **false** otherwise.

$SEQUENCE_UI.buildSequencesFromList()

$SEQUENCE_UI.buildSequencesFromList(tone, list);

buildSequencesFromList() Shows harmonic sequences from sequence list.

- **tone** selected tone.
- **list** list of sequence to show.

$SEQUENCE_UI.buildHarmonization()

$SEQUENCE_UI.buildHarmonization(id, mode, tone, bSeven);

buildHarmonization() Shows harmonization in a given mode.

- **id** unique identificator of sequence.
- **mode** the seleted mode as integer.
- **tone** the selected tone.
- **bSeven true** to use tetrads, **false** to use triads.

Harmonic sequence format

It's an array of chord:

```
[ chord, chord, chord, ... ]
```

Chord format

It's an array of five fields:

```
[ degree, signature, notation, function, options ]
```

degree

The degree of chord is the degree in main tonality, **"P1"** stands for the fondamental.

Degree	I	#I	bII	II	#II	bbIII	bIII	III	#III	bIV	IV	#IV
Note	C	C#	Db	D	D#	Ebb	Eb	E	E#	Fb	F	F#
Code	"P1"	"A1"	"m2"	"M2"	"A2"	"d3"	"m3"	"M3"	"A3"	"d4"	"P4"	"A4"

Degree	bV	V	#V	bbVI	bVI	VI	#VI	bbVII	bVII	VII
Note	Gb	G	G#	Abb	Ab	A	A#	Bbb	Bb	B
Code	"d5"	"P5"	"A5"	"d6"	"m6"	"M6"	"A6"	"d7"	"m7"	"M7"

signature

It's the string of chord signature.

notation

This is the degree notation in **roman number**.

function

T	**"T" Tonic** degree **I** in **fondamental state** without **leading-tone**.
T	**"Ts" Tonic** degree **I** in **fondamental state** with **leading-tone**.
T	**"t" Tonic** degré **I** à l'**état de renversement** ou autre degré sans la **leading-tone**.
T	**"ts" Tonique** degré **I** à l'**état de renversement** ou autre degré avec la **leading-tone**.
D	**"d" Dominante** avec la **sensible** mais sans la **quarte**.
D	**"D" Dominante** avec le **triton tonal** donc **sensible** et **quarte**.
D	**"Dst" substitution tritonique**, **subV7** ou **bII7** avec **quarte** et **sensible**.
D^2	**"D2" Dominante secondaire**, **V7/?**.
D^2	**"D2st" substitution tritonique** de la **dominante secondaire**, **subV7/?**.
SD	**"SD" Sous-dominante**.
SD^2	**"SD2" Sous-dominante secondaire**.

options

1 - Static sequence, data in div, non given tone

HTML

```
<div class="harmonic-sequence">
    ['P4','m','IV','SD'], ['P5','7','V7','D'], ['P1','m','I','T']
</div>
```

JavaScript

```
$SEQUENCE_UI.init({"tonality":"mG"});
```

Cm	D^7	Gm
IV	V^7	I
SD	D	T

Tone (degree **I**) is set by the given value in $SEQUENCE_UI.init({"tonality":"mG"}), here it's **G**.

2 - Static sequence, data in JavaScript variable, non given tone

HTML

```
<div id="sequence_1" class="harmonic-sequence"></div>
```

JavaScript

```
var sequence_1 = [['M2','m','II','SD'],['P5','','V','d'],['P1','','I','T']];
$SEQUENCE_UI.init({"tonality":"MG"});
```

Am	D	G
II	V	I
SD	D	T

Tone (degree **I**) is set by the given value in $SEQUENCE_UI.init({"tonality":"MG"}), here it's **G**.

3 - Static sequence, data in JavaScript variable, given tone D#

HTML

```
<div align="center">
    <div id="sequence_2" class="harmonic-sequence">D#</div>
</div>
```

JavaScript

```
var sequence_2 = sequence_1;
$SEQUENCE_UI.init({"tonality":"mG"});
```

E#m	A#	D#
II	V	I
SD	D	T

Tone (degree **I**) is set by the given value in tone , here it's **D#**.

4 - Transposable sequence, data in JavaScript variable, non given tone

HTML

```
<div id="sequence_3" class="harmonic-sequence transposable"></div>
```

JavaScript

```
var sequence_3 = [
    ['M2','7','II/V','D2'], ['P5','7','V7','D'], ['P1','M7','I','Ts']
];
$SEQUENCE_UI.init({"tonality":"MC#"});
```

D#^{7}	G#^{7}	C#^{M7}
II/V	V^{7}	I
D^{2}	D	T

Tone (degree **I**) is initially set by the given value in $SEQUENCE_UI.init(tone), here it's **C#**.
It can be updated calling to $SEQUENCE_UI.updateTonality(tonality).

```
$SEQUENCE_UI.updateTonality("MF");
```

G^{7}	C^{7}	F^{M7}
II/V	V^{7}	I
D^{2}	D	T

5 - Transposable sequence, data in JavaScript variable, given tone Ab

HTML

```
<div id="sequence_4" class="harmonic-sequence transposable">Ab</div>
```

JavaScript

```
var sequence_4 = [
    ['P5','','V','d'],['P1','','I','T'],['P4','','IV','SD'],['P1','','I','T']
];
```

Eb	Ab	Db	Ab
V	I	IV	I
D	T	SD	T

Tone (degree **I**) is initially set by the given value in $SEQUENCE_UI.init(tone), here it's **Ab**.
Il peut être mis à jour par l'appel à $SEQUENCE_UI.updateTonality(tonality).

```
$SEQUENCE_UI.updateTonality("MF");
```

C	F	Bb	F
V	I	IV	I
D	T	SD	T

6 - Available chord notations

Triads with inversions

C	Cm	C^{dim}	C°	Cm^{b5}	$C^{\#5}$	C^{+5}	$C^{\#5}$
x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T
C/E	Cm/Eb	C^{dim}/Eb	C°/Eb	Cm^{b5}/Eb	$C^{\#5}$/E	C^{+5}/E	$C^{\#5}$/E
x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T
C/G	Cm/G	C^{dim}/Gb	C°/Gb	Cm^{b5}/Gb	$C^{\#5}$/G#	C^{+5}/G#	$C^{\#5}$/G#
x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T

Seventh chords with inversions

C^{M7}	C^{7}	Cm^{7}	Cm^{M7}	$C°^{7}$	C^{dim7}	$C^{Ø}$	Cm^{7b5}
x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T
C^{M7}/E	C^{7}/E	Cm^{7}/Eb	Cm^{M7}/Eb	$C°^{7}$/Eb	C^{dim7}/Eb	$C^{Ø}$/Eb	Cm^{7b5}/Eb
x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T
C^{M7}/G	C^{7}/G	Cm^{7}/G	Cm^{M7}/G	$C°^{7}$/Gb	C^{dim7}/Gb	$C^{Ø}$/Gb	Cm^{7b5}/Gb
x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T
C^{M7}	C^{7}/Bb	Cm^{7}/Bb	Cm^{M7}/B	$C°^{7}$/Bbb	C^{dim7}/Bbb	$C^{Ø}$/Bb	Cm^{7b5}/Bb
x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T

Jazz chords with tensions

C^{6}	Cm^{6}	C^{9}	C^{b9}	$C^{\#9}$	C^{M9}	C^{11}	$C^{\#11}$	C^{13}	C^{b13}
x	x	x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T	T	T
$C^{7\ alt}$	C^{7b9}	$C^{7,b9}$	C^{79}	$C^{7,9}$	$C^{7\#9}$	$C^{7,\#9}$	$C^{7,13}$	C^{7b13}	$C^{7,b13}$
x	x	x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T	T	T
$C^{7\#9b13}$	$C^{7,\#9,b13}$	$C^{79\#11}$	$C^{7,9,\#11}$	C^{79b13}	$C^{7,9,b13}$	$C^{7,9,13}$	$C^{7b9\#11}$	$C^{7,b9,\#11}$	C^{7b9b13}
x	x	x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T	T	T
$C^{7,b9,b13}$	$C^{7,b9,13}$	$C^{7b9,13}$	C^{7+5}	$C^{7\#5}$	CØ9	$C^{Ø11}$	$C^{Øb13}$	$C^{7,b13}$	$Cm^{7,9}$
x	x	x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T	T	T
Cm^{79}	$Cm^{7,11}$	$Cm^{7,9,13}$	Cm^{7b9}	$Cm^{7,b9}$	C^{M79}	$C^{M7,9}$	$C^{M7,13}$	$C^{M7,9,\#11}$	$C^{M7,9,13}$
x	x	x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T	T	T
C^{+M7}	C^{M7+}	C^{M7+5}	$C^{M7\#5}$	$C^{9,11}$	$C^{9\#11}$	$C^{9,\#11}$	C^{9b13}	$C^{9,b13}$	$C^{9,13}$
x	x	x	x	x	x	x	x	x	x
T	T	T	T	T	T	T	T	T	T
$C^{9,\#11,13}$	C_{sus2}	C_{sus4}	C^{add9}	C^{sus4}_{add9}	C?				
x	x	x	x	x	x				
T	T	T	T	T	T				

7 - Available degree notations

?	C?	min	?	Maj	7
?	?	?	?	?	V7/?
?	?	?	SD	SD	D^2

8 - Available function notations

Tonics

C	C/E	A
I	I/3	VI
T	T	T

C^{M7}	C/E	Em^7
I	I/3	III
T	T	T

Dominants

G	G^7
V	V^7
D	D

B°	$B°^7$
VII	$VII°^7$
D	D

Db^7
bII^7
D

Sub-Dominants

Dm	F
II	IV
SD	SD

D°	Fm	F^7	Gm	Ab	Bb
II	IV	IV	V	bVI	bVII
SD	SD	SD	SD	SD	SD

Secondary dominants

C^7	D^7	E^7	A^7
V^7/IV	V^7/V	V^7/VI	V^7/II
D^2	D^2	D^2	D^2

8 - Sequence rendered modes

```
var mySequence = [
    ['P1','','I','T'],['P1','7','V7/IV','D2'],['P4','','IV','SD'],['P5','','V',
    ['P5','7','V7','D'],['M3','m','VI','t'],['M3','7','V7/II','D2'],['M2','m','
    ['M7','°7','VII°7','D'],['P1','','I','T']
];
```

Harmonic functions only: @

```
<div id="mySequence" class="harmonic-sequence">@</div>
```

T	D²	SD	D	D	T	D²	SD	D	T

Harmonic functions with degrees: !

```
<div id="mySequence" class="harmonic-sequence">!</div>
```

I	V⁷/IV	IV	V	V⁷	VI	V⁷/II	II°	VII°⁷	I
T	D²	SD	D	D	T	D²	SD	D	T

Harmonic functions with colored degrees (from chord signature): !%

```
<div id="mySequence" class="harmonic-sequence">!%</div>
```

I	V⁷/IV	IV	V	V⁷	VI	V⁷/II	II°	VII°⁷	I
T	D²	SD	D	D	T	D²	SD	D	T

Harmonic functions with degrees and signatures: $

```
<div id="mySequence" class="harmonic-sequence">$</div>
```

Maj	7	Maj	Maj	7	min	7	dim	dim⁷	Maj
I	V⁷/IV	IV	V	V⁷	VI	V⁷/II	II°	VII°⁷	I
T	D²	SD	D	D	T	D²	SD	D	T

Harmonic functions with degrees and colored signatures: $%

```
<div id="mySequence" class="harmonic-sequence">$%</div>
```

Maj	7	Maj	Maj	7	min	7	dim	dim⁷	Maj
I	V⁷/IV	IV	V	V⁷	VI	V⁷/II	II°	VII°⁷	I
T	D²	SD	D	D	T	D²	SD	D	T

Harmonic functions with degrees and chords: tone

```
<div id="mySequence" class="harmonic-sequence">A</div>
```

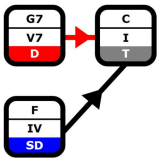
A	A⁷	D	E	E⁷	C#m	C#⁷	B°	G#°⁷	A
I	V⁷/IV	IV	V	V⁷	VI	V⁷/II	II°	VII°⁷	I
T	D²	SD	D	D	T	D²	SD	D	T

Harmonic functions with degrees and colored chords: tone%

```
<div id="mySequence" class="harmonic-sequence">A%</div>
```

A	A⁷	D	E	E⁷	C#m	C#⁷	B°	G#°⁷	A
I	V⁷/IV	IV	V	V⁷	VI	V⁷/II	II°	VII°⁷	I
T	D²	SD	D	D	T	D²	SD	D	T

$DIAGRAM_UI



harmonic diagram
editor
with json format

This JavaScript module handles insertion in html page of transposable harmonic diagram.

- Module file is: diagram-ui-1.0.0.min.js.
- It has dependencies: harmony-1.0.0.min.js, music-ui-1.0.0.min.js and music.min.css.

Files to include in HTML file header

```
<link rel="stylesheet" href="css/music.min.css" />

<script src="libs/harmony-1.0.0.min.js"></script>
<script src="libs/music-ui-1.0.0.min.js"></script>
<script src="libs/diagram-ui-1.0.0.min.js"></script>
```

$DIAGRAM_UI Methods

Method	Description
init()	Builds all harmonic diagram found in html file.
updateTonality()	Updates transposable harmonic diagrams with given tonality.

Examples

- Validation

$DIAGRAM_UI.init()

$DIAGRAM_UI.init(options);

init() Builds all harmonic diagrams found in html file.

```
options {
    'tonality': 'MC',
    'bMajorOrMinor': true or false,
    'lang': 'en' or 'fr' for language
}
```

$DIAGRAM_UI.updateTonality()

$DIAGRAM_UI.updateTonality(tonality, bMajorOrMinor);

updateTonality() Updates transposable harmonic diagrams with given tonality.

- **tonality** the selected tonality for update.
- **bMajorOrMinor true** to support diagrams with class **major** and **minor**, **false** otherwise.

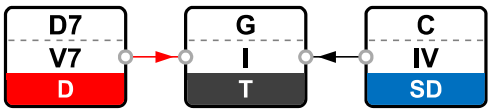
1 - Static diagram with given tone

HTML

```
<div id="diagram_1" class="harmonic-diagram"></div>
```

JavaScript

```
var diagram_1 = [[
  { interval: "P5", signature: "7", degree:"V7", fn: "D", pins: "right",
    links: [{from:"right", to:"left", x:1, y:0, color:'red'}]},
  { interval: "P1", signature: "", degree:"I", fn: "T", pins: "left+right" },
  { interval: "P4", signature: "", degree:"IV", fn: "SD", pins: "left",
    links: [{from:"left", to:"right", x:-1, y:0, color:'black'}] }
]];
$DIAGRAM_UI.init({tonality:"MG"});
```



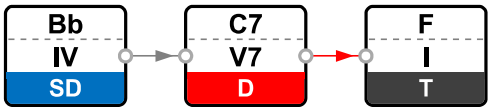
2 - Transposable diagram with given tone

HTML

```
<div id="diagram_2" class="harmonic-diagram transposable"></div>
```

JavaScript

```
var diagram_2 = [[
  { interval: "P4", signature: "", degree:"IV", fn: "SD", pins: "right",
    links: [{from:"right", to:"left", x:1, y:0, color:'gray'}] },
  { interval: "P5", signature: "7", degree:"V7", fn: "D", pins: "left+right",
    links: [{from:"right", to:"left", x:1, y:0, color:'red'}]},
  { interval: "P1", signature: "", degree:"I", fn: "T", pins: "left" }
]];
$DIAGRAM_UI.updateTonality("MF");
```



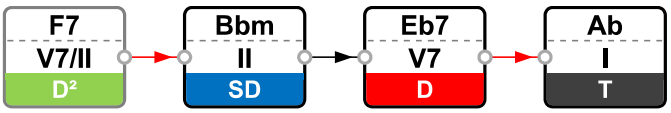
3 - Static diagram with fixed tone (Ab)

HTML

```
<div id="diagram_3" class="harmonic-diagram">Ab</div>
```

JavaScript

```
var diagram_3 = [[
  { interval: "M6", signature: "7", degree:"V7/II", fn: "D2", pins: "right",
    diatonic: false,
    links: [{from:"right", to:"left", x:1, y:0, color:'red'}] },
  { interval: "M2", signature: "m", degree:"II", fn: "SD",
    pins: "left+right",
    links: [{from:"right", to:"left", x:1, y:0, color:'black'}] },
  { interval: "P5", signature: "7", degree:"V7", fn: "D",
    pins: "left+right",
    links: [{from:"right", to:"left", x:1, y:0, color:'red'}]},
  { interval: "P1", signature: "", degree:"I", fn: "T", pins: "left" }
]];
$DIAGRAM_UI.init({tonality:"MG"});
```



4 - Transposable diagram with initial fixed tone (C#)

HTML

```
<div id="diagram_4" class="harmonic-diagram transposable">C#</div>
```

JavaScript

```
var diagram_4 = [
[ // line1
  { interval: "M2", signature: "°", degree:"II",  fn: "SD", pins: "right",
    links: [
      {from:"right", to:"left", x:1, y:0, color:'black'},
      {from:"right", to:"left", x:1, y:1, color:'gray'}] },
  { interval: "P5", signature: "7", degree:"V7", fn: "D",
    pins: "left+right",
    links: [{from:"right", to:"left", x:1, y:0, color:'red'}] },
  { interval: "P1", signature: "m", degree:"I", fn: "T", pins: "left" }
],[ // line2
  { interval: "M2", signature: "7", degree:"V7/V", fn: "D2", pins: "right",
    diatonic: false,
    links: [
      {from:"right", to:"left", x:1, y:0, color:'gray'},
      {from:"right", to:"left", x:1, y:-1, color:'red'}] },
  { interval: "M7", signature: "°7", degree:"VII°7", fn: "D",
    pins: "left+right",
    links: [{from:"right", to:"left", x:1, y:-1, color:'orange'}] }
]];
$DIAGRAM_UI.updateTonality("MF");
```

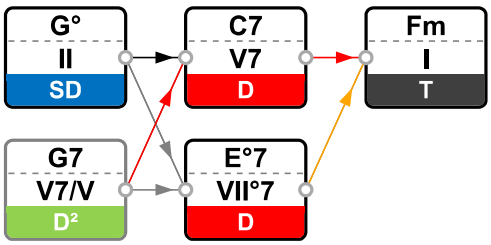


Diagram is transposable with *transposable* key in class list, call to *$DIAGRAM_UI.updateTonality* changes the initial tone value. We start with G and change it to F.