
EasyMIDI Documentation

Release 1.0

David de Kleer

Oct 04, 2017

Contents

1	Documentation	3
1.1	EasyMIDI.EasyMIDI	3
1.2	EasyMIDI.MusicTheory	4
1.3	EasyMIDI.Track	4
1.4	EasyMIDI.Note	6
1.5	EasyMIDI.Chord	7
1.6	EasyMIDI.RomanChord	9

A simple, easy to use algorithmic composition MIDI creator for Python, based on midiutil.

Creating a MIDI file can be as simple as this:

```
from EasyMIDI import EasyMIDI, Track, Note, Chord, RomanChord
from random import choice

easyMIDI = EasyMIDI()
track1 = Track("acoustic grand pino") # oops

c = Note('C', octave = 5, duration = 1/4, volume = 100)
e = Note('E', 5)
g = Note('G', 5)
chord = Chord([c,e,g]) # a chord of notes C, E and G
track1.addNotes([c, e, g, chord])

# roman numeral chord, first inversion (defaults to key of C)
track1.addNotes(RomanChord('I*', octave = 5, duration = 1))

easyMIDI.addTrack(track1)
easyMIDI.writeMIDI("output.mid")
```


CHAPTER 1

Documentation

<i>EasyMIDI.EasyMIDI()</i>	EasyMIDI handles MIDI files with the help of midiutil.
<i>EasyMIDI.MusicTheory()</i>	MusicTheory contains some helpful music theory data, like the major and minor scales based on the circle of fifths.
<i>EasyMIDI.Track</i> (instrument[, tempo])	Simple Track class which keeps the list of Notes/Chords, the instrument and the tempo.
<i>EasyMIDI.Note</i> (name[, octave, duration, volume])	The Note class contains musical notes and their properties, like octave, duration and volume.
<i>EasyMIDI.Chord</i> ([noteList])	The Chord is a simple class that contains lists of Notes.
<i>EasyMIDI.RomanChord</i> ([numeral, octave, ...])	The RomanChord class supports Roman chord numerals for creating chord progressions in a relatively easy way.

EasyMIDI.EasyMIDI

class EasyMIDI.EasyMIDI

EasyMIDI handles MIDI files with the help of midiutil.

Initialize a midiutil MIDIFile object.

__init__()

Initialize a midiutil MIDIFile object.

Methods

__init__()	Initialize a midiutil MIDIFile object.
addTrack(track)	Add a single track/channel to the midiutil MIDIFile object.
addTracks(tracks)	Add multiple tracks to the midiutil MIDIFile object.
writeMIDI(path)	Write the MIDI file to the disk.

addTrack (*track*)

Add a single track/channel to the midiutil MIDIFile object.

Parameters **track** (*Track*) – A Track object.

addTracks (*tracks*)

Add multiple tracks to the midiutil MIDIFile object.

Parameters **tracks** (list of *Track* objects) – A list of tracks.

writeMIDI (*path*)

Write the MIDI file to the disk.

Parameters **path** (*str*) – The path to store the MIDI file at (ex. output.mid).

EasyMIDI.MusicTheory

class EasyMIDI.**MusicTheory**

MusicTheory contains some helpful music theory data, like the major and minor scales based on the circle of fifths.

`__init__()`

Methods

<code>__init__()</code>	
<code>getMajorScales()</code>	Get the scales for major keys.
<code>getMinorScales()</code>	Get the scales for minor keys.

getMajorScales ()

Get the scales for major keys.

Returns A dict of major scales (ex. 'C' : ['C', 'D', 'E', ...]).

Return type dict

getMinorScales ()

Get the scales for minor keys.

Returns A dict of minor scales (ex. 'C' : ['C', 'D', 'E', ...]).

Return type dict

majorScales

Builds the scales for major keys.

minorScales

Builds the scales for minor keys.

EasyMIDI.Track

class EasyMIDI.**Track** (*instrument*, *tempo=120*)

Simple Track class which keeps the list of Notes/Chords, the instrument and the tempo. To be used with `addTrack()` or `addTracks()` in *EasyMIDI*.

Initializes a Track object.

Parameters

- **instrument** (*str*) – A midi instrument name.
- **tempo** (*int*) – The tempo of the track.

`__init__(instrument, tempo=120)`

Initializes a Track object.

Parameters

- **instrument** (*str*) – A midi instrument name.
- **tempo** (*int*) – The tempo of the track.

Methods

<code>__init__(instrument[, tempo])</code>	Initializes a Track object.
<code>addChord(chord)</code>	Is identical to <code>addNotes()</code> .
<code>addChords(chords)</code>	Is identical to <code>addNotes()</code> .
<code>addNote(note)</code>	Is identical to <code>addNotes()</code> .
<code>addNotes(notes)</code>	Add a list of Notes or Chords to the Track.
<code>getInstrument()</code>	Returns the instrument of this Track.
<code>getNotes()</code>	Returns a copy of the notes of this Track.
<code>getTempo()</code>	Returns the tempo of this Track.
<code>matchInstrument(description)</code>	(Fuzzy) matches instrument descriptions to MIDI program numbers.

addChord (*chord*)

Is identical to `addNotes()`.

addChords (*chords*)

Is identical to `addNotes()`.

addNote (*note*)

Is identical to `addNotes()`.

addNotes (*notes*)

Add a list of Notes or Chords to the Track.

Parameters **notes** (*Note* or *Chord* in list or single objects) – The list of Notes or Chords, or single Notes or Chords.

getInstrument ()

Returns the instrument of this Track.

Returns The instrument.

Return type *str*

getNotes ()

Returns a copy of the notes of this Track.

Returns A list of Chord and Note objects.

Return type list of *Chord* or *Note*.

getTempo()

Returns the tempo of this Track.

Returns The tempo.

Return type int

matchInstrument (*description*)

(Fuzzy) matches instrument descriptions to MIDI program numbers.

Parameters **description** (*str*) – The instrument description (ex. acoustic grand).

EasyMIDI.Note

class EasyMIDI.**Note** (*name*, *octave=4*, *duration=0*, *volume=100*)

The Note class contains musical notes and their properties, like octave, duration and volume.

Initializes a Note object.

Parameters

- **name** (*str*) – The name of the note (ex. C).
- **octave** (*int*) – The octave of the note (1-7).
- **duration** (*float or int*) – The duration of the note (ex. 1/4 is quarter note).
- **volume** (*int*) – The volume of the note (0 to 100)

__init__ (*name*, *octave=4*, *duration=0*, *volume=100*)

Initializes a Note object.

Parameters

- **name** (*str*) – The name of the note (ex. C).
- **octave** (*int*) – The octave of the note (1-7).
- **duration** (*float or int*) – The duration of the note (ex. 1/4 is quarter note).
- **volume** (*int*) – The volume of the note (0 to 100)

Methods

<code>__init__(name[, octave, duration, volume])</code>	Initializes a Note object.
<code>getDuration()</code>	Returns the duration of the note.
<code>getName()</code>	Returns the name of the note.
<code>getOctave()</code>	Returns the octave of the note.
<code>getVolume()</code>	Returns the volume of the note.
<code>setDuration(duration)</code>	Sets the duration of the note to duration.
<code>setName(name)</code>	Sets the name of the note to name.
<code>setOctave(octave)</code>	Sets the octave of the note to octave.
<code>setVolume(volume)</code>	Sets the volume of the note to volume.

getDuration()

Returns the duration of the note.

Returns Current duration (ex. 1/4).

Return type float or int

getName ()

Returns the name of the note.

Returns Current name (ex. C).

Return type str

getOctave ()

Returns the octave of the note.

Returns Current octave (ex. 4).

Return type int

getVolume ()

Returns the volume of the note.

Returns Current volume (ex. 80).

Return type int

setDuration (*duration*)

Sets the duration of the note to duration.

Parameters **duration** (*float or int*) – The new duration (ex. 1/4).

setName (*name*)

Sets the name of the note to name.

Parameters **name** (*str*) – The new name (ex. C).

setOctave (*octave*)

Sets the octave of the note to octave. :param octave: The new octave (1-7). :type octave: int

setVolume (*volume*)

Sets the volume of the note to volume.

Parameters **volume** (*int*) – The new volume (0-100).

EasyMIDI.Chord

class EasyMIDI.Chord (*noteList=[]*)

The Chord is a simple class that contains lists of Notes.

Initializes a Chord object.

Parameters **noteList** (list of *Note*) – A list of Notes that should form a chord.

__init__ (*noteList=[]*)

Initializes a Chord object.

Parameters **noteList** (list of *Note*) – A list of Notes that should form a chord.

Methods

<code>__init__</code> ([noteList])	Initializes a Chord object.
<code>addNote</code> (note)	Adds a note to the Chord.
Continued on next page	

Table 1.6 – continued from previous page

<code>getDuration()</code>	Returns the duration of the longest Chord note.
<code>getNotes()</code>	Gets the Notes of the chord.
<code>getVolume()</code>	Returns the volume of the loudest Chord note.
<code>removeNote(note)</code>	Removes a note from the Chord.
<code>setDuration(duration)</code>	Sets the duration of the note to duration.
<code>setNotes(noteList)</code>	Sets the Chord notes to the Notes in noteList.
<code>setOctave(octave)</code>	Sets the octave of the note to octave.
<code>setVolume(volume)</code>	Sets the volume of the note to volume.

addNote (*note*)

Adds a note to the Chord.

Parameters

- **note** – The note to add.
- **type** – *Note*

getDuration ()

Returns the duration of the longest Chord note.

Returns The duration.

Return type float or int

getNotes ()

Gets the Notes of the chord.

Returns The list of Notes of the Chord.

Return type list of *Note*

getVolume ()

Returns the volume of the loudest Chord note.

Returns The volume.

Return type int

removeNote (*note*)

Removes a note from the Chord.

Parameters

- **note** – The note to remove.
- **type** – *Note*

setDuration (*duration*)

Sets the duration of the note to duration.

Parameters **duration** (*float or int*) – The new duration (ex. 1/4).

setNotes (*noteList*)

Sets the Chord notes to the Notes in noteList.

Parameters **noteList** – A list of notes that should form a chord.

setOctave (*octave*)

Sets the octave of the note to octave.

Parameters **octave** (*int*) – The new octave (1-7).

setVolume (*volume*)

Sets the volume of the note to volume.

Parameters **volume** (*int*) – The new volume (0-100).

EasyMIDI.RomanChord

class EasyMIDI.**RomanChord** (*numeral='I', octave=4, duration=0, key='C', major=True, volume=100*)

The RomanChord class supports Roman chord numerals for creating chord progressions in a relatively easy way. It's also possible to customize the numerals with intervals or signs and inversions:

- **I6, I7**: add 6th or 7th note interval to I chord
- **Isus2, Isus4**: the suspended chords relative to the I chord
- **I-, I+**: Diminished and augmented I chord
- **Imaj7, Imin7, Idom7**: major, minor and dominant 7th chord from I
- **I*, I****: first and second inversion of the I chord, can be combined with the other customizations (ex. Isus2**)

Full code example:

```
from EasyMIDI import *
mid = EasyMIDI()
track = Track('acoustic grand')
for numeral in ['I', 'IV', 'V', 'I']:
    track.addChord(RomanChord(numeral))
mid.addTrack(track)
mid.writeMIDI('output.mid')
```

Initializes a RomanChord.

Parameters

- **numeral** (*str*) – A roman numeral (I, II, III, IV, V, VI or VII)
- **octave** (*int*) – The octave of the RomanChord (1-7).
- **duration** (*float or int*) – The duration of the RomanChord (ex. 1/4).
- **key** (*str*) – The key of the RomanChord (ex. Ab or G# or D).
- **major** (*bool*) – If true, use major scale. If false, use minor scale.
- **volume** (*int*) – The volume of the RomanChord (0-100).

__init__ (*numeral='I', octave=4, duration=0, key='C', major=True, volume=100*)

Initializes a RomanChord.

Parameters

- **numeral** (*str*) – A roman numeral (I, II, III, IV, V, VI or VII)
- **octave** (*int*) – The octave of the RomanChord (1-7).
- **duration** (*float or int*) – The duration of the RomanChord (ex. 1/4).
- **key** (*str*) – The key of the RomanChord (ex. Ab or G# or D).
- **major** (*bool*) – If true, use major scale. If false, use minor scale.
- **volume** (*int*) – The volume of the RomanChord (0-100).

Methods

<code>__init__</code> ([numeral, octave, duration, key, ...])	Initializes a RomanChord.
<code>addNote</code> (note)	Adds a note to the Chord.
<code>getDuration</code> ()	Returns the duration of the longest Chord note.
<code>getNotes</code> ()	Gets the Notes of the chord.
<code>getNumeral</code> ()	Returns the numeral of the RomanChord.
<code>getVolume</code> ()	Returns the volume of the loudest Chord note.
<code>removeNote</code> (note)	Removes a note from the Chord.
<code>setDuration</code> (duration)	Sets the duration of the note to duration.
<code>setKey</code> (key[, major])	Sets the key of the RomanChord, optionally changes scales.
<code>setNotes</code> (noteList)	Sets the Chord notes to the Notes in noteList.
<code>setOctave</code> (octave)	Sets the octave of the note to octave.
<code>setVolume</code> (volume)	Sets the volume of the note to volume.

getNumeral ()

Returns the numeral of the RomanChord.

Returns The numeral.

Return type str

setKey (key, major=True)

Sets the key of the RomanChord, optionally changes scales.

Parameters

- **key** (str) – The key of the RomanChord (ex. Ab or G# or D).
 - **major** (bool) – If true, use major scale. If false, use minor scale.
- genindex
 - search

Symbols

`__init__()` (EasyMIDI.Chord method), 7
`__init__()` (EasyMIDI.EasyMIDI method), 3
`__init__()` (EasyMIDI.MusicTheory method), 4
`__init__()` (EasyMIDI.Note method), 6
`__init__()` (EasyMIDI.RomanChord method), 9
`__init__()` (EasyMIDI.Track method), 5

A

`addChord()` (EasyMIDI.Track method), 5
`addChords()` (EasyMIDI.Track method), 5
`addNote()` (EasyMIDI.Chord method), 8
`addNote()` (EasyMIDI.Track method), 5
`addNotes()` (EasyMIDI.Track method), 5
`addTrack()` (EasyMIDI.EasyMIDI method), 4
`addTracks()` (EasyMIDI.EasyMIDI method), 4

C

Chord (class in EasyMIDI), 7

E

EasyMIDI (class in EasyMIDI), 3

G

`getDuration()` (EasyMIDI.Chord method), 8
`getDuration()` (EasyMIDI.Note method), 6
`getInstrument()` (EasyMIDI.Track method), 5
`getMajorScales()` (EasyMIDI.MusicTheory method), 4
`getMinorScales()` (EasyMIDI.MusicTheory method), 4
`getName()` (EasyMIDI.Note method), 7
`getNotes()` (EasyMIDI.Chord method), 8
`getNotes()` (EasyMIDI.Track method), 5
`getNumeral()` (EasyMIDI.RomanChord method), 10
`getOctave()` (EasyMIDI.Note method), 7
`getTempo()` (EasyMIDI.Track method), 5
`getVolume()` (EasyMIDI.Chord method), 8
`getVolume()` (EasyMIDI.Note method), 7

M

majorScales (EasyMIDI.MusicTheory attribute), 4
`matchInstrument()` (EasyMIDI.Track method), 6
minorScales (EasyMIDI.MusicTheory attribute), 4
MusicTheory (class in EasyMIDI), 4

N

Note (class in EasyMIDI), 6

R

`removeNote()` (EasyMIDI.Chord method), 8
RomanChord (class in EasyMIDI), 9

S

`setDuration()` (EasyMIDI.Chord method), 8
`setDuration()` (EasyMIDI.Note method), 7
`setKey()` (EasyMIDI.RomanChord method), 10
`setName()` (EasyMIDI.Note method), 7
`setNotes()` (EasyMIDI.Chord method), 8
`setOctave()` (EasyMIDI.Chord method), 8
`setOctave()` (EasyMIDI.Note method), 7
`setVolume()` (EasyMIDI.Chord method), 8
`setVolume()` (EasyMIDI.Note method), 7

T

Track (class in EasyMIDI), 4

W

`writeMIDI()` (EasyMIDI.EasyMIDI method), 4