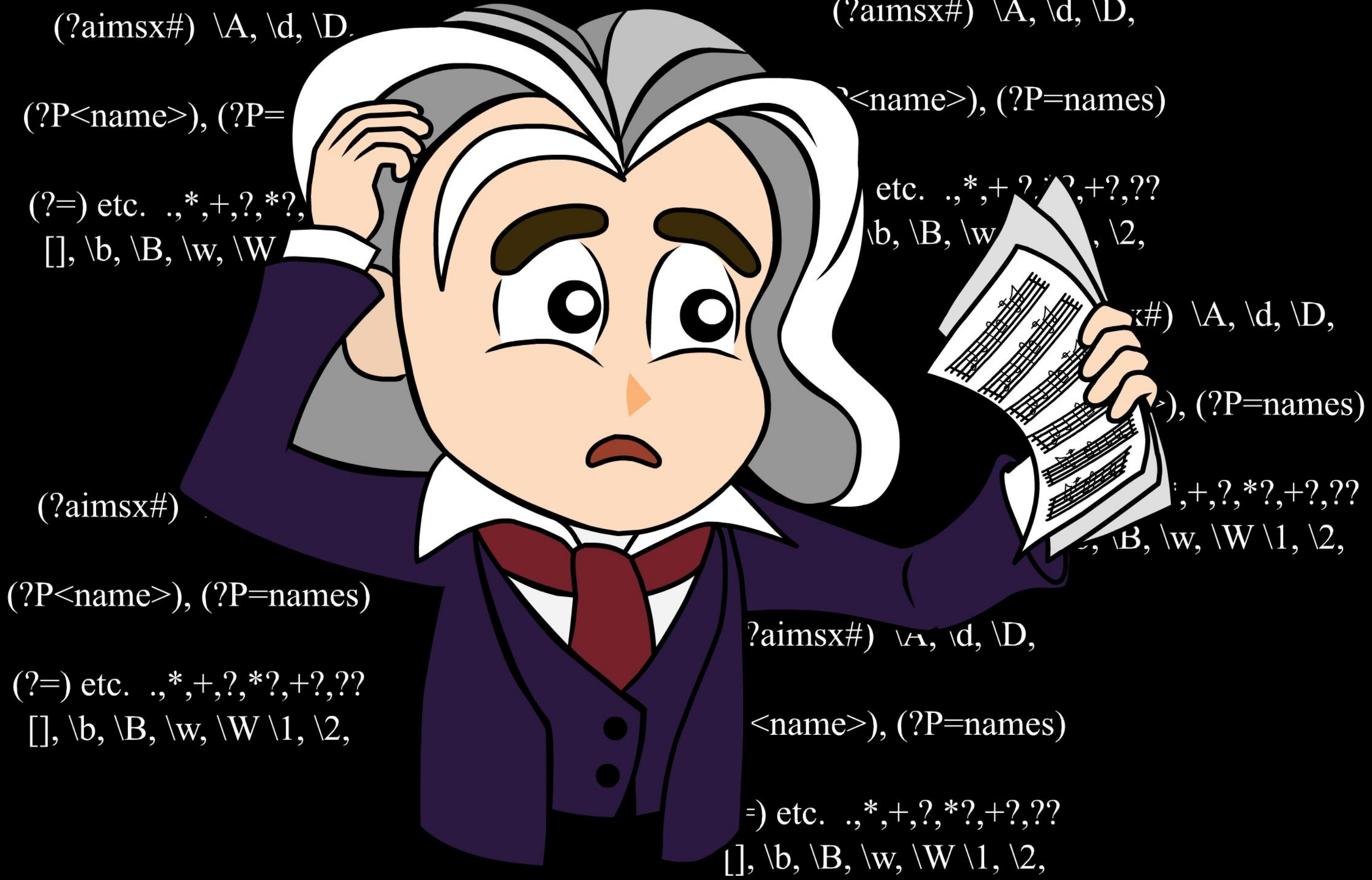
Learning Python Through Music: JythonMusic & Pyknon

Ria Baldevia | 2016 US PyCon | Portland, Oregon

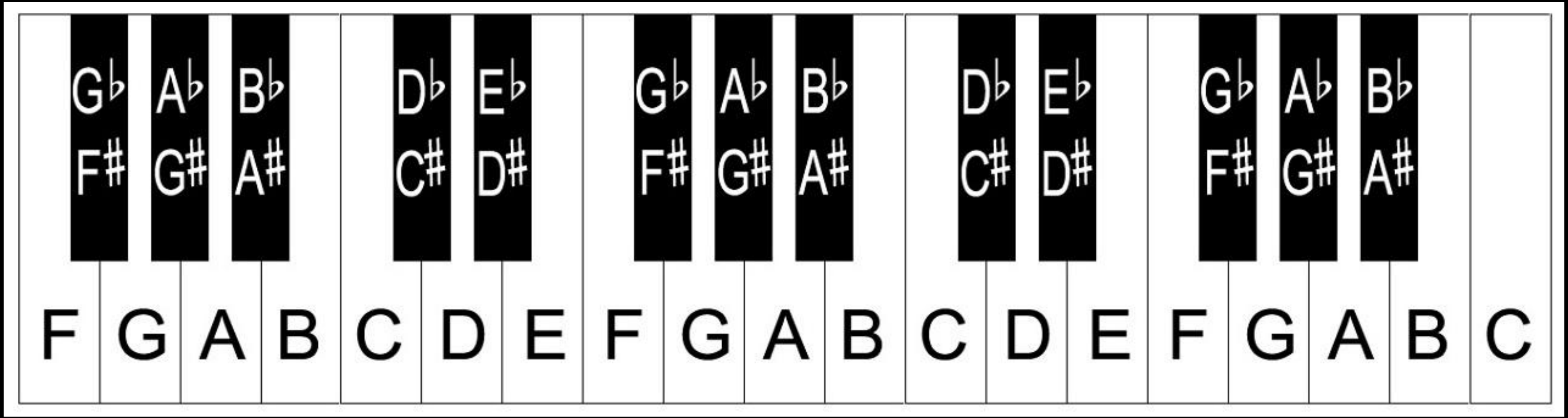

(?aimsx#) \A, \d, \D,
(?P<name>), (?P=
(?=) etc. .,*,+,?,*?,
[], \b, \B, \w, \W
(?aimsx#) \A, \d, \D,
<name>), (?P=names)
etc. .,*,+,?,*?,+?,??
\b, \B, \w
, \2,
#) \A, \d, \D,
), (?P=names)
,+,?,*?,+?,??
\B, \w, \W \1, \2,
(?aimsx#)
(?P<name>), (?P=names)
(?=) etc. .,*,+,?,*?,+?,??
[], \b, \B, \w, \W \1, \2,
?aimsx#) \A, \d, \D,
<name>), (?P=names)
=) etc. .,*,+,?,*?,+?,??
[], \b, \B, \w, \W \1, \2,

JythonMusic

JythonMusic: Jython Environment for Music

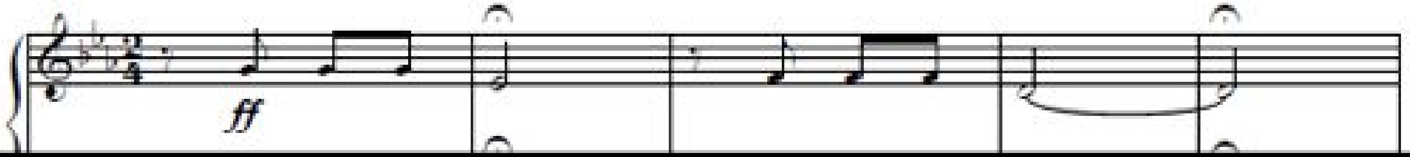
An open source environment conducive to creative programming. It is for programmers and musicians who want to explore music and coding.

Easy to download JEM via JythonMusic.org

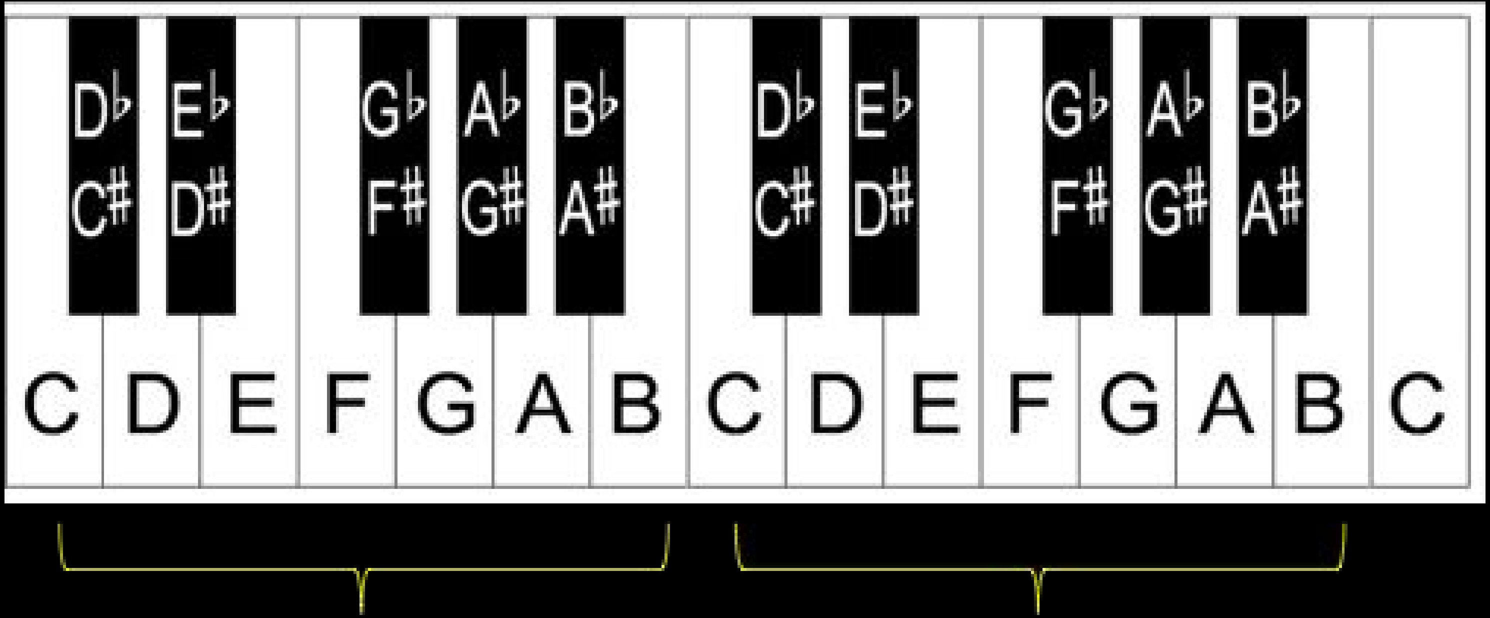

G♭ F♯
A♭ G♯
B♭ A♯
D♭ C♯
E♭ D♯
G♭ F♯
A♭ G♯
B♭ A♯
D♭ C♯
E♭ D♯
G♭ F♯
A♭ G♯
B♭ A♯
F G A B C D E F G A B C D E F G A B C

Symphony No.5 in C Minor, Op. 67

Ludwig Van Beethoven

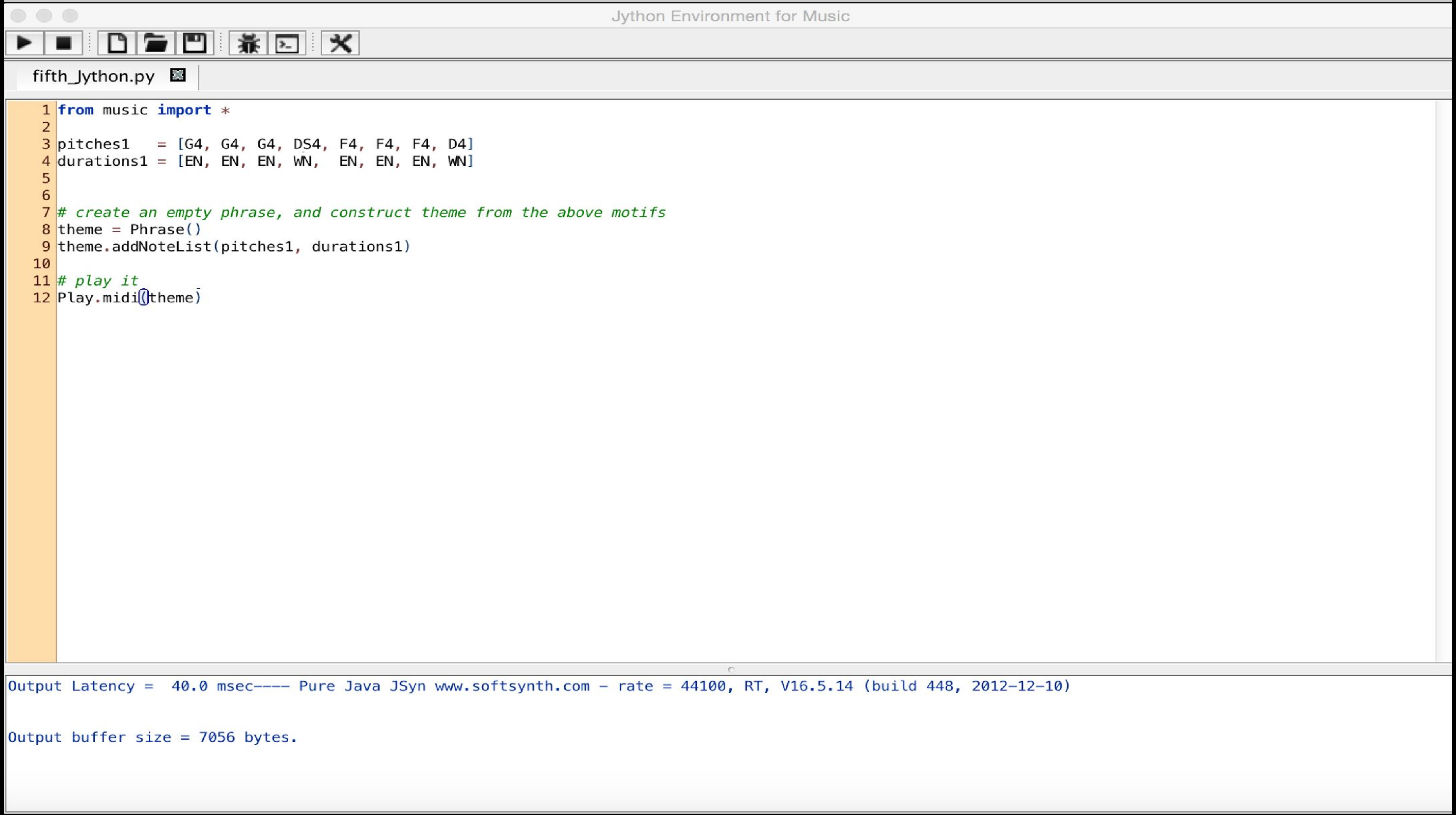


G G G Eb F F F D



Octave 4

Octave 5



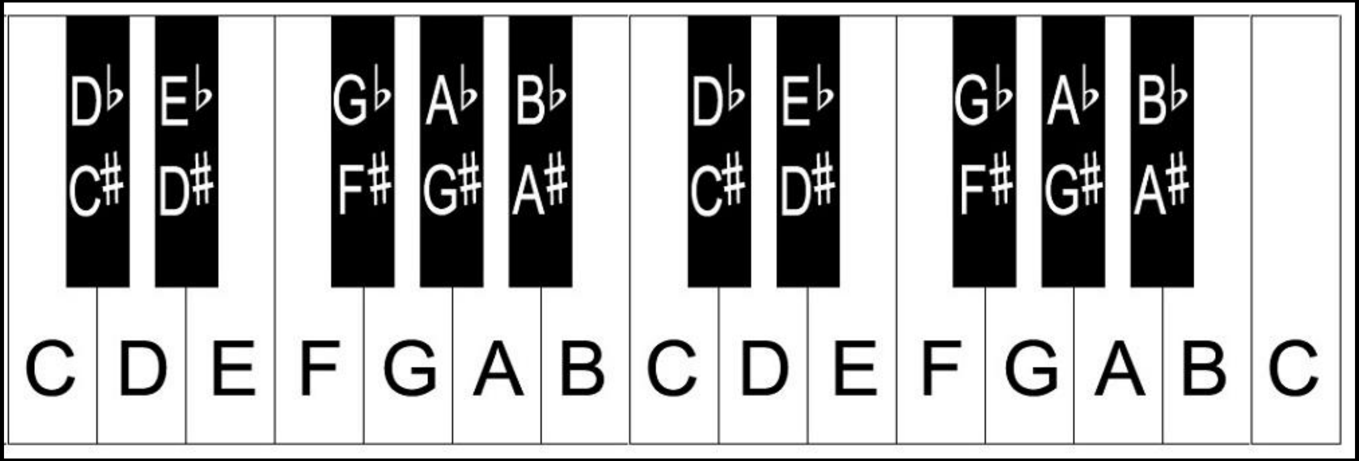
```
from music import *

pitches1   = [G4, G4, G4, DS4, F4, F4, F4, D4]
durations1 = [EN, EN, EN, WN,  EN, EN, EN, WN]


# create an empty phrase, and construct theme from the above motifs
theme = Phrase()
theme.addNoteList(pitches1, durations1)

# play it
Play.midi(theme)
```

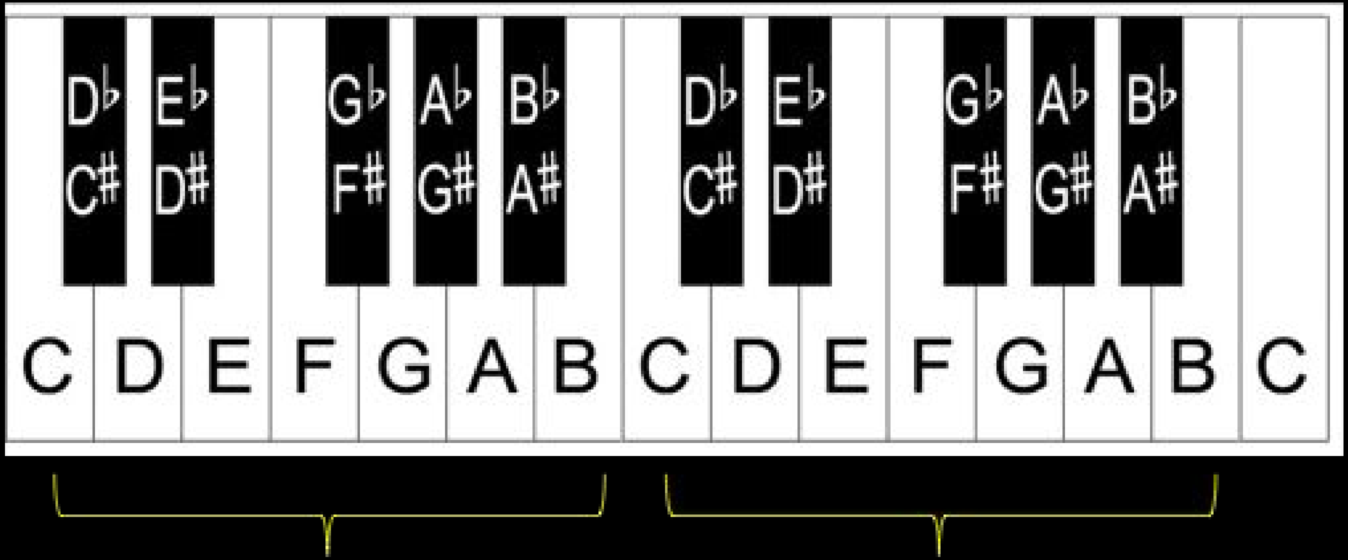
```
Output Latency =  40.0 msec---- Pure Java JSyn www.softsynth.com - rate = 44100, RT, V16.5.14 (build 448, 2012-12-10)


Output buffer size = 7056 bytes.
```

Pyknon

By Pedro Kroger

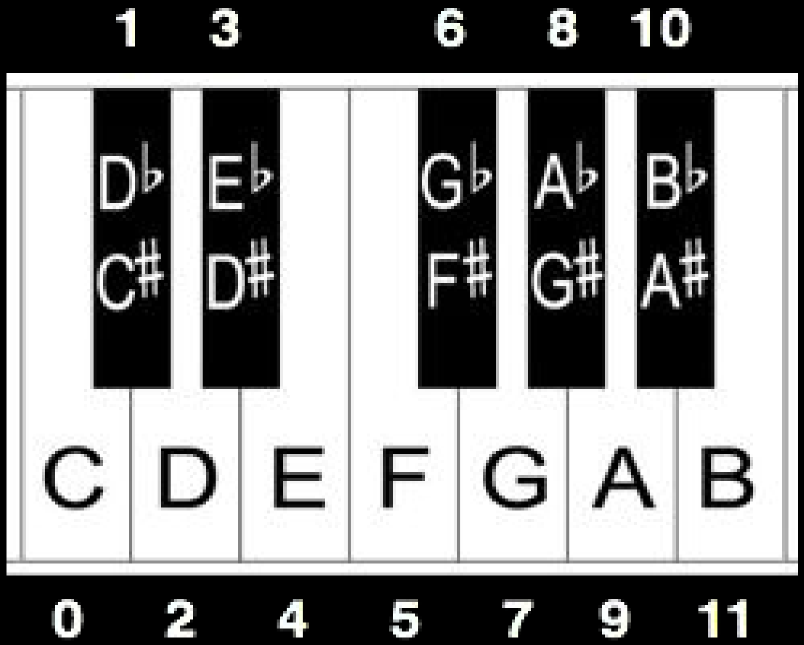
Book: Music for Geeks & Nerds


D♭ C♯
E♭ D♯
G♭ F♯
A♭ G♯
B♭ A♯
D♭ C♯
E♭ D♯
G♭ F♯
A♭ G♯
B♭ A♯
C D E F G A B C D E F G A B C



Octave 4 Octave 5

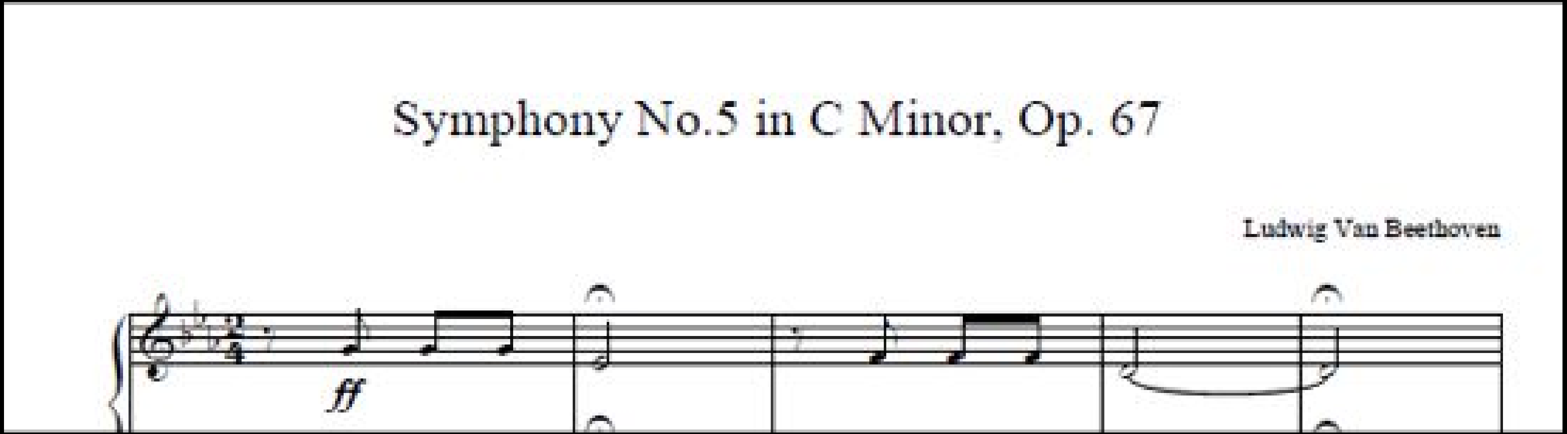
```
>>> from pyknon.music import Note
>>> def mod12(n):
...     return n % 12
...
>>> def note_name(number):
...     notes = "C C# D D# E F F# G G# A A# B".split()
...     return notes[mod12(number)]

>>> note_name(5)
'F'
>>> note_name(0)
'C'
>>> note_name(60)
'C'
```


1
3
6
8
10
D♭ C#
E♭ D#
G♭ F#
A♭ G#
B♭ A#
C
D
E
F
G
A
B
0
2
4
5
7
9
11

```
>>> from pyknon.music import Note
>>> from pyknon.music import NoteSeq
>>> a = Note()
>>> a
<C>
>>> a.octave
5
>>> a.value
0
>>> a.volume
100
>>> a.dur
0.25
>>> a.midi_number
60
>>> a = Note(3)
>>> a
<D#>
>>> a.octave
5
>>> a.volume
100
>>> a.value
3
>>> a.dur
0.25
>>> a.midi_number
63
```

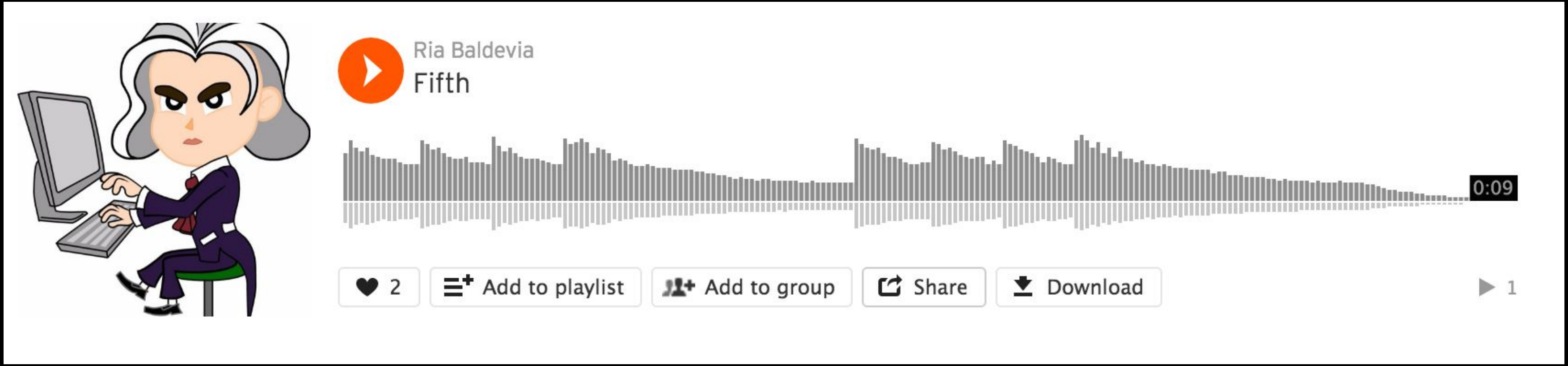
```
>>> from pyknon.music import NoteSeq
>>> from pyknon.music import Note
>>> NoteSeq([Note(0), Rest(1), Note("C#8")])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'Rest' is not defined
>>> from pyknon.music import Rest
>>> NoteSeq([Note(0), Rest(1), Note("C#8")])
<Seq: [<C>, <R: 1>, <C#>]>
>>> NoteSeq("C R C#8")
<Seq: [<C>, <R: 0.25>, <C#>]>
>>> NoteSeq("C#2 D#")
<Seq: [<C#>, <D#>]>
>>> NoteSeq("C# D#")
<Seq: [<C#>, <D#>]>
>>> NoteSeq([Note(0), Note(2)])
<Seq: [<C>, <D>]>
>>> NoteSeq("G G Eb")
<Seq: [<G>, <G>, <D#>]>
>>> NoteSeq("F F F D")
<Seq: [<F>, <F>, <F>, <D>]>
```

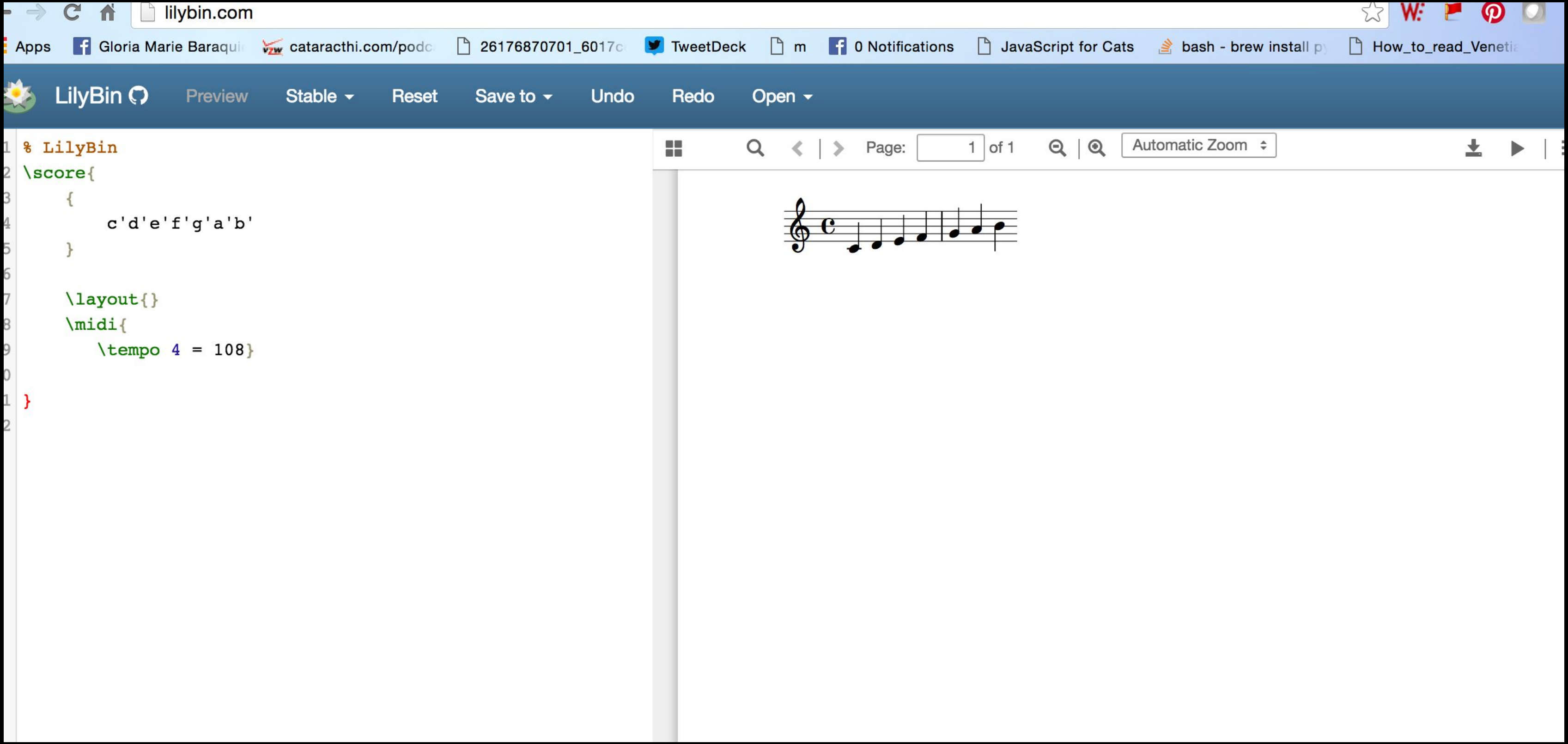
```
>>> from pyknon.music import Note
>>> from pyknon.music import NoteSeq
>>> c = Note("C")
>>> c
<C>
>>> c_major_scale = NoteSeq("C D E F G A B")
>>> c_major_scale
<Seq: [<C>, <D>, <E>, <F>, <G>, <A>, <B>]>
>>> c.harmonize(c_major_scale)
[<C>, <E>, <G>]
>>> d = Note("D")
>>> d
<D>
>>> d_major_scale = NoteSeq("D E F# G A B C#)
  File "<stdin>", line 1
    d_major_scale = NoteSeq("D E F# G A B C#)
                           ^
SyntaxError: EOL while scanning string literal
>>> d_major_scale = NoteSeq("D E F# G A B C#")
>>> d_major_scale
<Seq: [<D>, <E>, <F#>, <G>, <A>, <B>, <C#>]>
>>> d.harmonize(d_major_scale)
[<D>, <F#>, <A>]
>>> d_major_scale.harmonize(size=4)
[<Seq: [<D>, <F#>, <A>, <C#>]>, <Seq: [<E>, <G>, <B>, <D>]>, <Seq: [<F#>, <A>, <C#>, <E>]>, <Seq: [<G>, <B>, <D>, <F#>]>, <Seq:
[<A>, <C#>, <E>, <G>]>, <Seq: [<B>, <D>, <F#>, <A>]>, <Seq: [<C#>, <E>, <G>, <B>]>]
```



	G	G	G	Eb	F	F	F	D
Jython	G4	G4	G4	EF4/DS4	F4	F4	F4	D4
Pyknon	7	7	7	3	5	5	5	2
	G5	G5	G5	D#5	F5	F5	F5	D5

```
>>> from pyknon.genmidi import Midi
>>> from pyknon.music import NoteSeq
>>> from pyknon.music import Note
>>> notes1=NoteSeq([Note(7, dur=.25), Note(7, dur=.25), Note(7, dur=.25), Note(3, dur=1),
Note(5, dur=.25), Note(5, dur=.25), Note(5, dur=.25), Note(2, dur=1)])
>>> midi = Midi(number_tracks=1, tempo=108)
>>> midi.seq_notes(notes1, track=0)
14.0
>>> midi.write("fifth10.mid")
```





```
% LilyBin
\score{
    {
        c'd'e'f'g'a'b'
    }

    \layout{}
    \midi{
       \tempo 4 = 108}

}
```

...

	Jython	Pyknon	Others
Center of the Keyboard	Octave 4	Octave 5	.ly ‘
Environment	JEM	Can code from Command Line	EarSketch: application
Difficulty	Beginners	Beginners-Advanced	