

Programming In Python

Music Mixing with EarSketch

Mr. Christopher Michaud

michaudc@marist.com

www.nebomusic.net

Marist School / Georgia Institute of Technology

What is Python? python™

- General purpose Computer Programming Language
- Web development: Google and Yahoo
- Game Development, Science, Graphics
- In Business: National Weather Service, NASA, IBM, Disney, and Nokia
- Used in Development of the Google Car and other Robotic Systems

What do all Human Languages Have?

- Nouns
- Verbs
- Adverbs
- Adjectives
- Clauses
- Pronouns
- Subject and Object

All Programming Languages Have:

- Data Types
- Data Structures
- Variables
- Operators
- Control Structures
- Functions
 - Call
 - Define
- Class Structures

Data Types (Many more than on this list)

| Data Type | Definition and Examples |
|----------------|--|
| integer | Whole Numbers (-9, -4, 0, 1, 200, 500, . . .) |
| float | Decimal Numbers (-23.45, -11.2, 0.0001, 1.25, 5.67 . . .) |
| string | Characters 'strung together' into words ("Hello", "Mr. Michaud", "0+++0+0+0---0+++") |
| boolean | True or False |

Variables: Store Data

```
my_age = 15
```

```
bill = 17.25
```

```
music = "C:\MyMusic\song.mp3"
```

```
lives = 3
```

```
name = "Mr. Michaud
```

```
alive = True
```

Data Structures: Group Data together

In Python, these are called “Lists”

```
students = ["Rebecca", "Joshua", "Carter",  
            "Kelley", "Jose"]
```

```
ages = [16, 12, 13, 13, 14]
```

```
images = ["dog.png", "bird.png", "cat.png"]
```

```
c_scale = ["C", "D", "E", "F", "G", "A", "B", "C"]
```

Operators: Perform Changes or Measurements with Data

| Operator | Example |
|----------|--|
| + | Adds two values together (5 + 4) |
| - | Subtracts one value from another (5 - 4) |
| * | Multiplies two values together (5 * 4) |
| / | Divides one value by another (5 / 4) |
| % | Modulo: Returns the remainder after division (5 % 4 = 1) |
| = | Assigns a value to a variable |
| == | Compares two values |
| < | Less than |
| > | Greater Than |
| ! | Not |
| and | And |
| or | or |

Control Structures: Control flow of the Program

| Type | Example |
|--|---|
| Conditional: If a statement is true | <pre>if (age == 16): canDrive = True</pre> |
| Loop: Repeat Code for a number of times | <pre>for count in range(1, 10): print count</pre> |
| While Loop: Repeat while a condition is true | <pre>while (lives > 3): x = x + 1</pre> |
| If Else Conditional: | <pre>if (age == 16): canDrive = True else: canDrive = False</pre> |

Functions: Group Commands together

- Define Functions

```
def printName(first, last):  
    name = first + " " + last  
    print name
```

- Call Functions

```
printName("Rebecca", "Michaud")
```

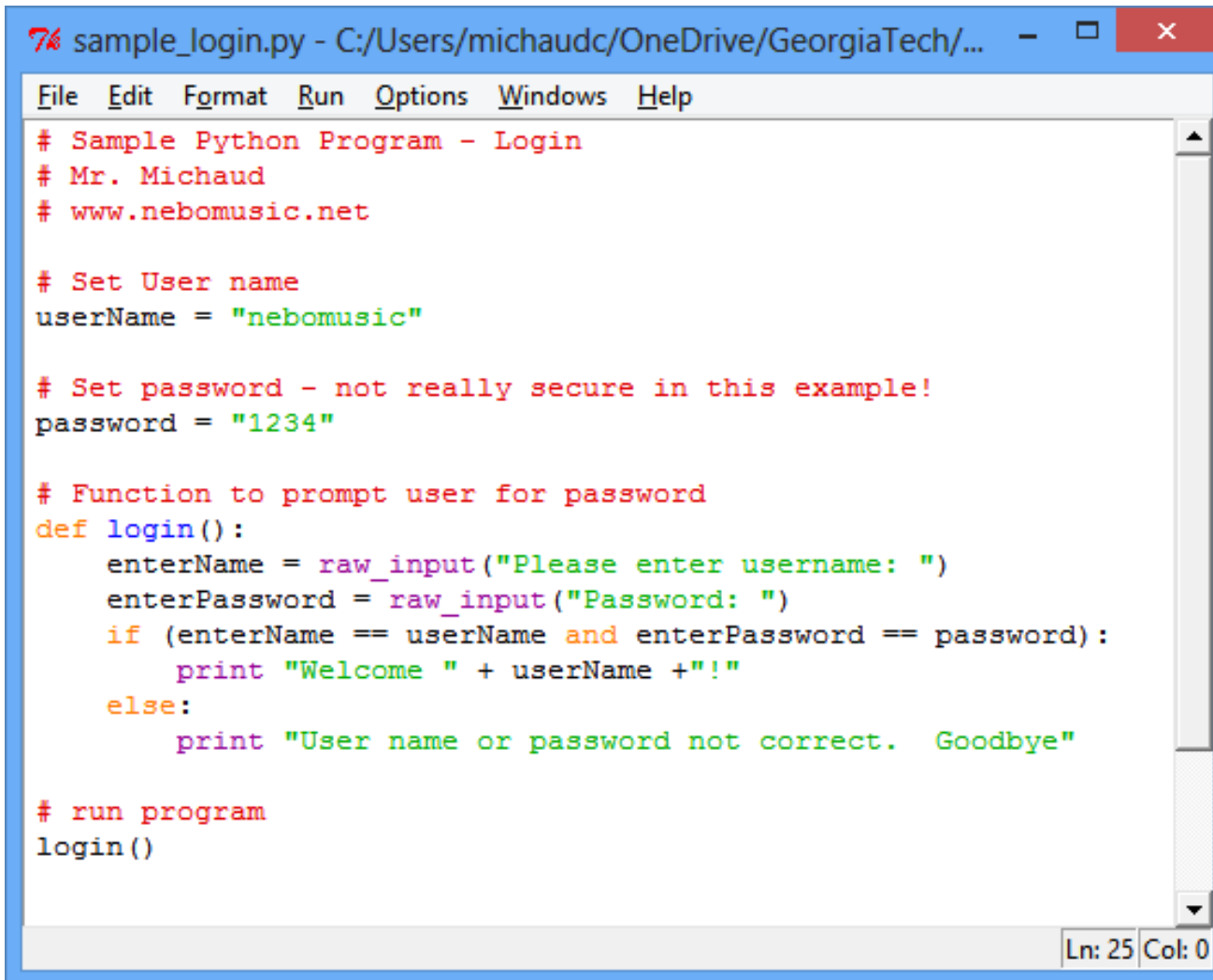
Classes: Model Objects in Programming: Store Properties and Actions of Object

```
class robot:

    def __init__(self, x = 0.0, y = 0.0, heading = 0.0):
        self.x = x
        self.y = y
        self.heading = heading
        self.turning = 0
        self.distance = distance

    def move(self, turning, distance):
        # Execute motion
        self.heading += turning
        self.x += distance * cos(self.heading)
        self.y += distance * sin(self.heading)
```

Example: Login Program (Not secure!)

A screenshot of a Python IDE window titled "sample_login.py - C:/Users/michaudc/OneDrive/GeorgiaTech/...". The window contains Python code for a simple login program. The code includes comments, variable assignments for username and password, a function definition for login(), and a call to the login() function. The code is color-coded: comments are red, function definitions are blue, and strings are green. The status bar at the bottom right shows "Ln: 25 Col: 0".

```
7% sample_login.py - C:/Users/michaudc/OneDrive/GeorgiaTech/... - □ ×
File Edit Format Run Options Windows Help
# Sample Python Program - Login
# Mr. Michaud
# www.nebomusic.net

# Set User name
userName = "nebomusic"

# Set password - not really secure in this example!
password = "1234"

# Function to prompt user for password
def login():
    enterName = raw_input("Please enter username: ")
    enterPassword = raw_input("Password: ")
    if (enterName == userName and enterPassword == password):
        print "Welcome " + userName + "!"
    else:
        print "User name or password not correct. Goodbye"

# run program
login()
Ln: 25 Col: 0
```

Example Program: Drawing with Turtle

```
7% sample_drawing_turtle.py - C:/Users/michaudc/One... - □ ×
File Edit Format Run Options Windows Help
# Sample Python Program: Drawing
# Mr. Michaud
# www.nebomusic.net

# import Turtle Library
from turtle import *

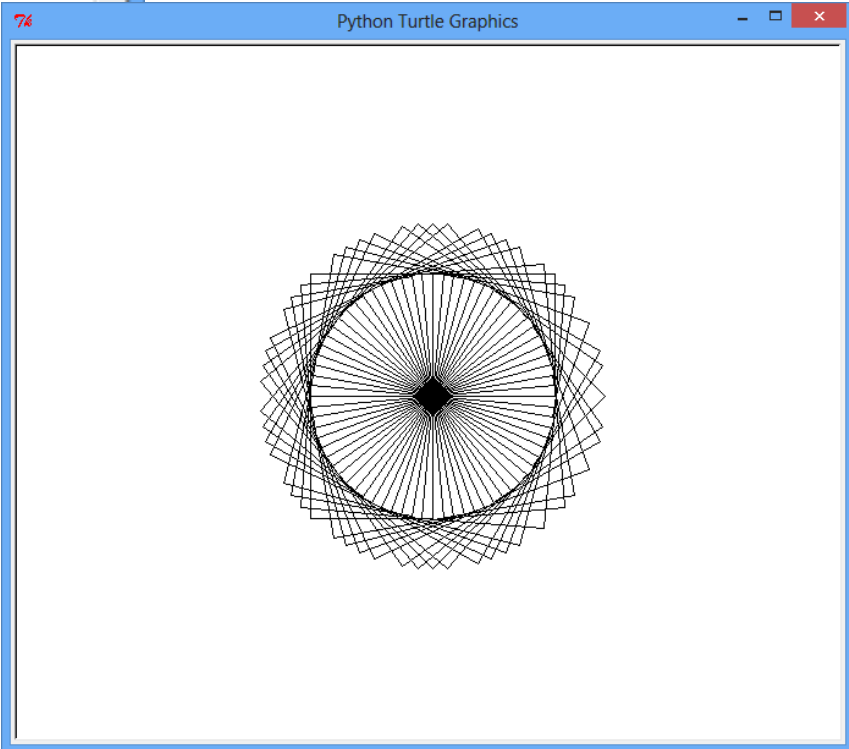
# Create a window for a Turtle
window = Screen()

# Create a turtle
rebecca = Turtle()

def drawSquare(size):
    for edge in range(4):
        rebecca.forward(size)
        rebecca.left(90)

# Draw a Design
for index in range(54):
    drawSquare(100)
    rebecca.left(25)

done ()
```

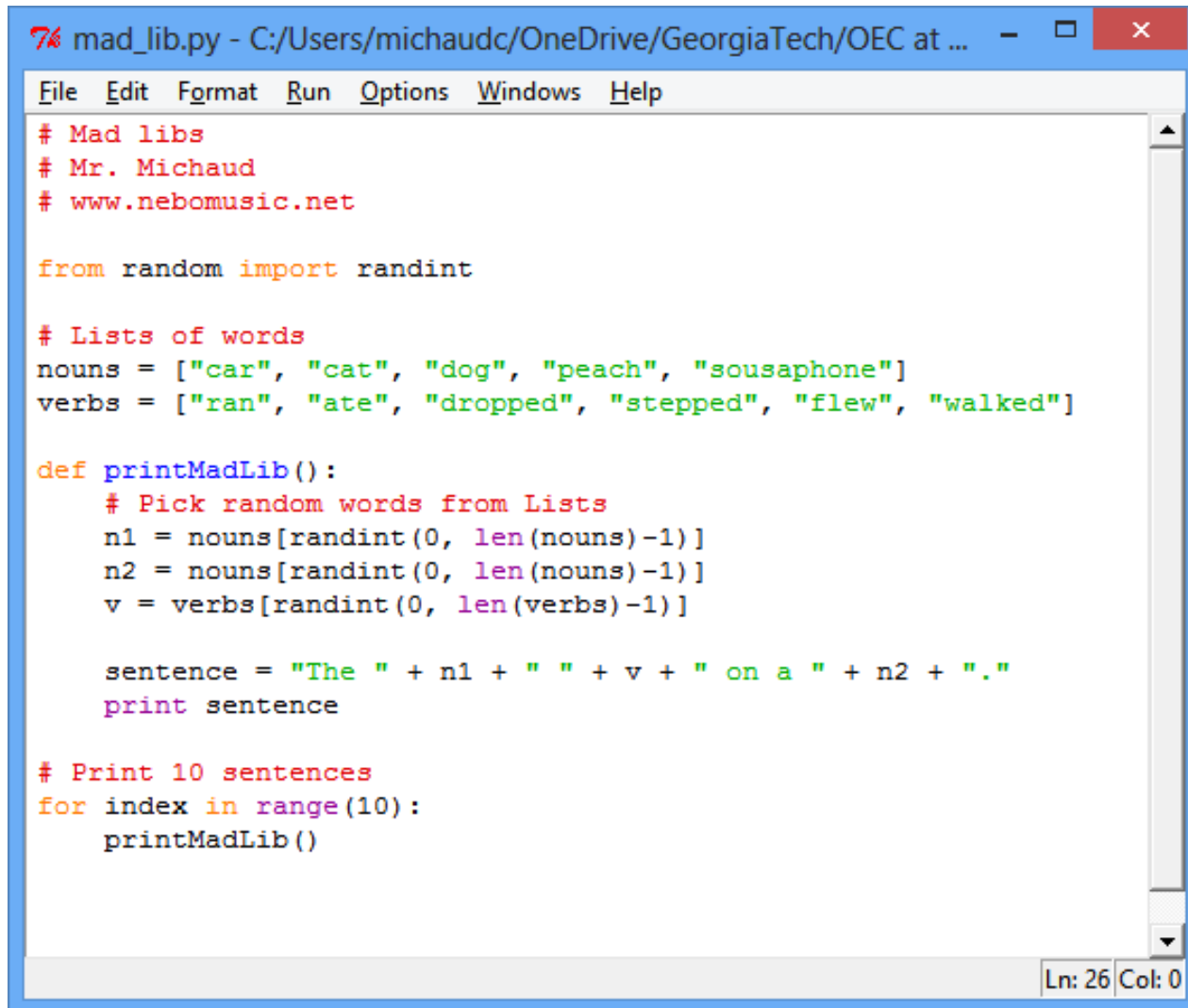


Python Turtle Graphics

Ln: 28 Col: 0

The image shows a Python Turtle Graphics window displaying a complex geometric drawing. The drawing consists of a central black square, surrounded by a series of overlapping squares. Each square is rotated 25 degrees relative to the previous one, creating a spiral-like pattern of squares that fills a circular area. The squares are drawn with black lines on a white background.

Example Program: Mad Libs

A screenshot of a Python IDE window titled "mad_lib.py - C:/Users/michaudc/OneDrive/GeorgiaTech/OEC at ...". The window contains Python code for a Mad Libs program. The code includes comments, imports, list definitions, a function definition, and a loop to print 10 sentences. The status bar at the bottom right shows "Ln: 26 Col: 0".

```
7% mad_lib.py - C:/Users/michaudc/OneDrive/GeorgiaTech/OEC at ...
File Edit Format Run Options Windows Help
# Mad libs
# Mr. Michaud
# www.nebomusic.net

from random import randint

# Lists of words
nouns = ["car", "cat", "dog", "peach", "sousaphone"]
verbs = ["ran", "ate", "dropped", "stepped", "flew", "walked"]

def printMadLib():
    # Pick random words from Lists
    n1 = nouns[randint(0, len(nouns)-1)]
    n2 = nouns[randint(0, len(nouns)-1)]
    v = verbs[randint(0, len(verbs)-1)]

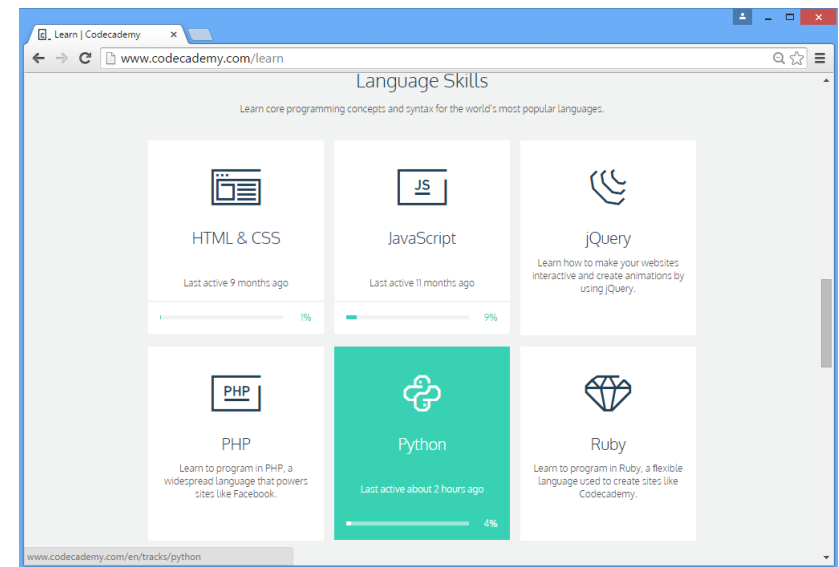
    sentence = "The " + n1 + " " + v + " on a " + n2 + "."
    print sentence

# Print 10 sentences
for index in range(10):
    printMadLib()

Ln: 26 Col: 0
```

Online Python Tutorial: Codecademy

- Go to:
<http://www.codecademy.com/>
- Create an account and Log in
- Go to the Python Tutorial
- Goal for Week: Finish Python Language Tutorial Sections
 - Python Syntax
 - Strings and Console Output
 - Conditionals and Control Flow
 - Functions
 - Lists and Dictionaries
 - Lists and Functions
 - Loops



What is EarSketch?

- Online Programming and Music Mixing Workstation

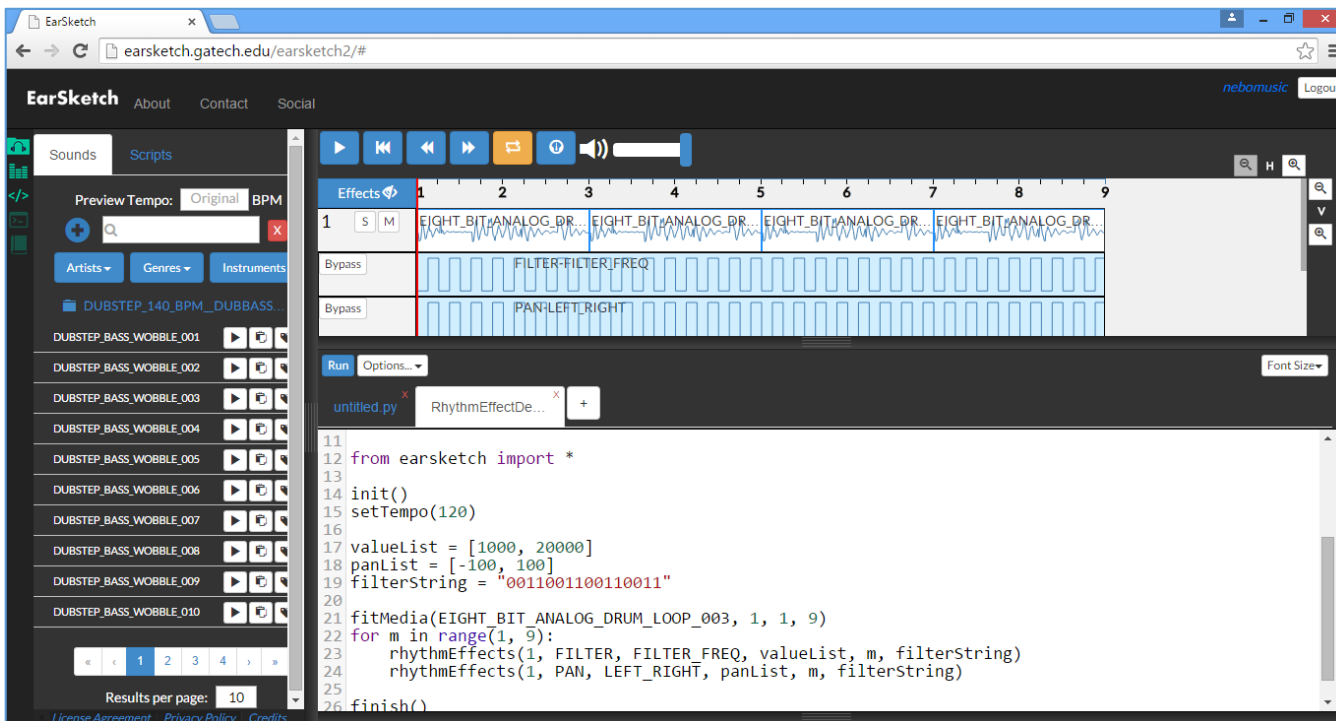
<http://earskech.gatech.edu/earskech2/>

-Used at Georgia Tech to teach Digital Music Mixing.

-Tool for Programming

-Free!

-Web based



The screenshot displays the EarSketch web interface. At the top, there's a navigation bar with 'EarSketch', 'About', 'Contact', 'Social', and a 'Logout' button. Below this is a control panel with 'Sounds' and 'Scripts' tabs, a 'Preview Tempo' dropdown set to 'Original' BPM, and buttons for 'Artists', 'Genres', and 'Instruments'. A list of sound effects is visible, including 'DUBSTEP_140_BPM_DUBBASS...', 'DUBSTEP_BASS_WOBBLE_001' through '010'. The main workspace features a timeline with a play button, a volume slider, and a grid of effects. The effects rack includes 'EIGHT_BIT_ANALOG_DR...', 'FILTER-FILTER|FREQ', and 'PAN-LEFT|RIGHT'. Below the effects rack is a 'Run' button and an 'Options...' dropdown. A Python code editor is open, showing the following code:

```
11
12 from earskech import *
13
14 init()
15 setTempo(120)
16
17 valueList = [1000, 20000]
18 panList = [-100, 100]
19 filterString = "0011001100110011"
20
21 fitMedia(EIGHT_BIT_ANALOG_DRUM_LOOP_003, 1, 1, 9)
22 for m in range(1, 9):
23     rhythmEffects(1, FILTER, FILTER_FREQ, valueList, m, filterString)
24     rhythmEffects(1, PAN, LEFT_RIGHT, panList, m, filterString)
25
26 finish()
```


Description of EarSketch

- Programming Environment
- Python Based
- Web Based App or Installed System of Software
- API built in Python for Music Mixing
- NSF Funded project to encourage computational interest through the mixing and sharing of music.
- Curriculum and Social Media Site

EarSketch Website: Web based IDE and DAW

Sound Library

Music View: Display and Playback

Documentation and Curriculum

Toggle Views

The screenshot shows the EarSketch web interface. On the left is a 'Sound Library' with a search bar and filters for Artists, Genres, and Instruments. Below these are sound files like 'DUBSTEP_140_BPM_DUBBASS...'. In the center is the 'Music View' showing a timeline with tracks 1 and 2. Track 1 has effects like 'EIGHT_BIT_ANALOG_DR...', 'DISTORTION-DISTO_GAIN', and 'VOLUME-GAIN'. Track 2 has 'EIGHT_BIT_ATARI_LEAD...'. Below the tracks is a 'Coding Window' with a 'Run' button and a text area containing Python code. On the right is a 'Documentation and Curriculum' section titled 'Getting Started with EarSketch' and 'Why Learn Programming for Music?'. The code in the coding window is as follows:

```
30
31 # Track 2 with loop
32
33 for measure in range(1, 9):
34     if (measure % 2 == 1):
35         setEffect(2, PAN, LEFT_RIGHT, -100, measure, measure+1)
36     else:
37         setEffect(2, PAN, LEFT_RIGHT, 100, measure, measure+1)
38
39 # Track 1: Distortion
40 setEffect(1, DISTORTION, DISTO_GAIN, 20, 1)
41 setEffect(1, VOLUME_GAIN, 10, 1)
42
```

Coding Window: Programming

Sample EarSketch Program

```
sample.py +
1 # python code
2 #
3 # script_name: sample.py
4 #
5 # author: Christopher Michaud
6 #
7 # description: Demo of EarSketch Sections
8 #
9 #
10 #
11
12 from earsketch import *
13
14 init()
15 setTempo(120)
16
17 # A Section - Three Tracks
18
19 fitMedia(TECHNO_LOOP_PART_003, 1, 1, 5)
20 fitMedia(TECHNO_CLUBLEAD_001, 2, 1, 5)
21 fitMedia(TECHNO_CLUBSFX_001, 3, 1, 5)
22
23
24
25 finish()
26
```

} Comments

} Setup Section

} Music Section

} Finish Section

Essential Elements we will use in Python:

- **Comments**

```
# This is a comment - meant for  
Humans
```

- **Includes** – loading preset methods or data

```
from earsketch import *
```

- **Functions** – telling the computer “what to do”

```
fitMedia(drums, 1, 1, 5)
```

- **Variables and data types** – Names for information stored by program

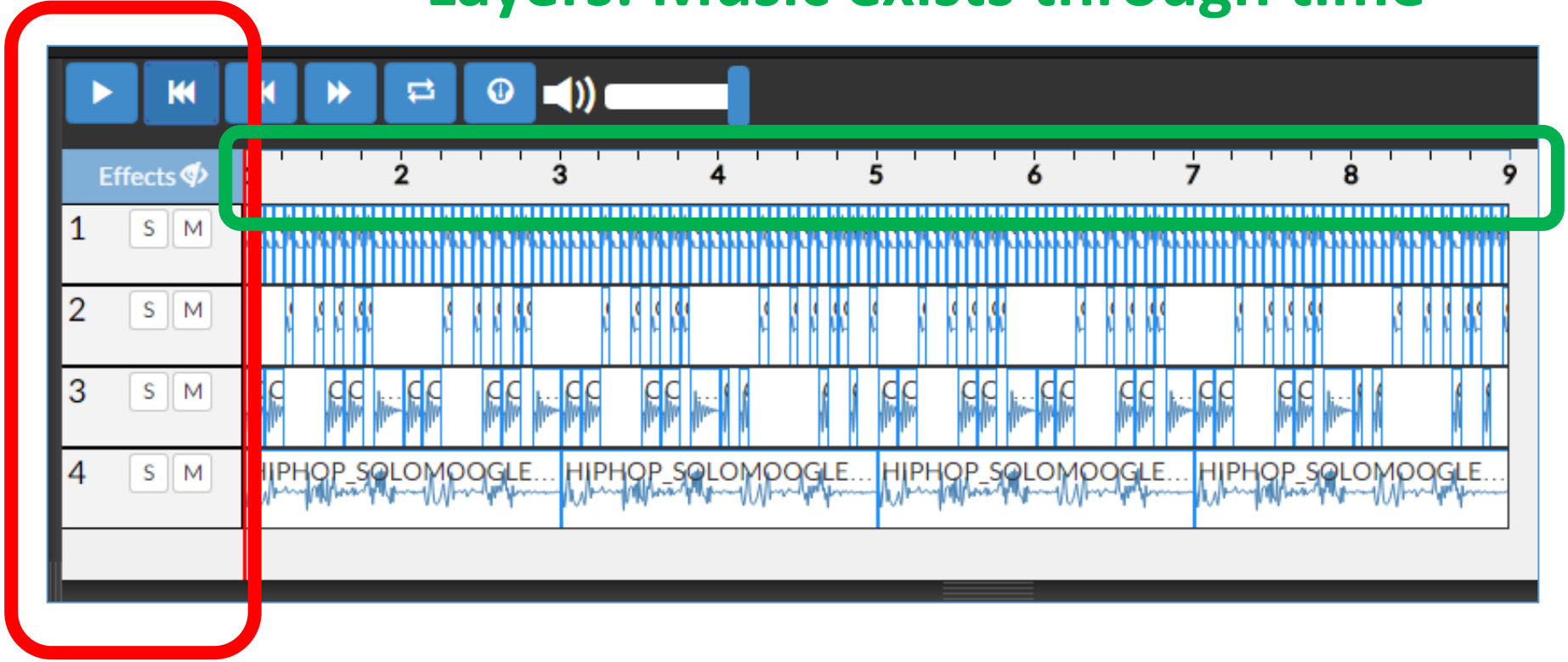
```
Beat1 = "0+++0+++0+0+0+++"
```

- **Tabs:** Enclose code in sections

- **Lists:** Groups variables into one data structure

How is Music Organized? How do we encode musical clips?

Layers: Music exists through time



Tracks: Vertical Layers of Music

EarSketch Python Functions

| EarSketch Function | Description |
|-------------------------------|--|
| <code>init()</code> | Start new EarSketch Mix and clear the DAW |
| <code>setTempo(120)</code> | Sets the beats per minute (speed) of the Mix |
| <code>println("Hello")</code> | Prints message to console |

EarSketch Python Functions

Music Mixing Functions

```
fitMedia(file, track, start, end)
```

```
makeBeat(file, track, measure, beatString)
```

```
setEffect(track, effect, parameter, sV,  
sM, eV, eM)
```

```
rhythmEffects(track, effect, parameter,  
list, measure)
```

“fitMedia” Function

```
fitMedia(file, track, start, end)
```

Location of
Media
Sound

Which Track
in Reaper

Start
measure.

End Measure

Example:

```
fitMedia(HIP_HOP_DRUMS1_2M, 1, 1, 9)
```


Setting Volume Effects

```
setEffect(track, VOLUME, GAIN, level, start,  
level2, end)
```

- Example

```
setEffect(1, VOLUME, GAIN, -40, 1, 10, 5)
```

Selected List of Effects and Parameters

| Effect | Parameter | Min to Max Values |
|------------|------------------|-----------------------------|
| VOLUME | GAIN | -60 to 12 |
| DELAY | DELAY_TIME | 0 to 300.0 |
| CHORUS | CHORUS_LENGTH | 1.0 to 15.0 |
| CHORUS | CHORUS_NUMVOICES | 1.0 to 8.0 |
| DISTORTION | DISTO_GAIN | 0.0 to 50.0 |
| FILTER | FILTER_FREQ | 20.0 to 20000.0 |
| PAN | LEFT_RIGHT | -100 to 100 (Left to Right) |

Complete Effect list at:

<http://ears sketch.gatech.edu/category/learning/reference/every-effect-explained>

“makeBeat” Method

```
makeBeat(file, track, measure, BeatString)
```

Location of
Media
Sound

Which Track
in Reaper

What
measure.

Example: “0+++0+++0+0+0+++”

Example:

```
makeBeat(drums, 1, 1, “0+0+0+++00-00+++”)
```

Beat String notation

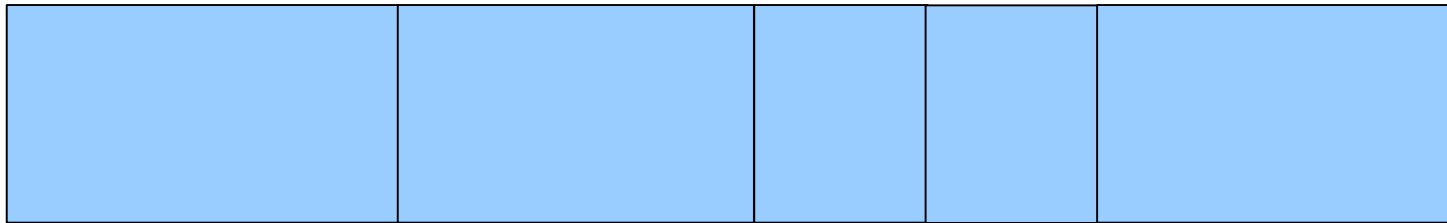
“0, 1, 2, 3 . . . ” = Which Media Sound you want for the segment of beat. Correspond to placement in a List that is one based.

Note: 0 will refer to a sound if it is the only media file in the argument.

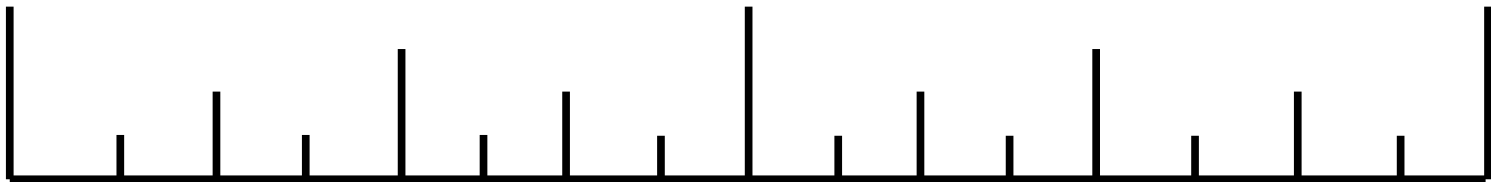
“+” Means extend or loop the Media sound $1/16^{\text{th}}$ of a measure.

“-” Means $1/16^{\text{th}}$ measure of rest.

"0+++0+++0+0+0+++"

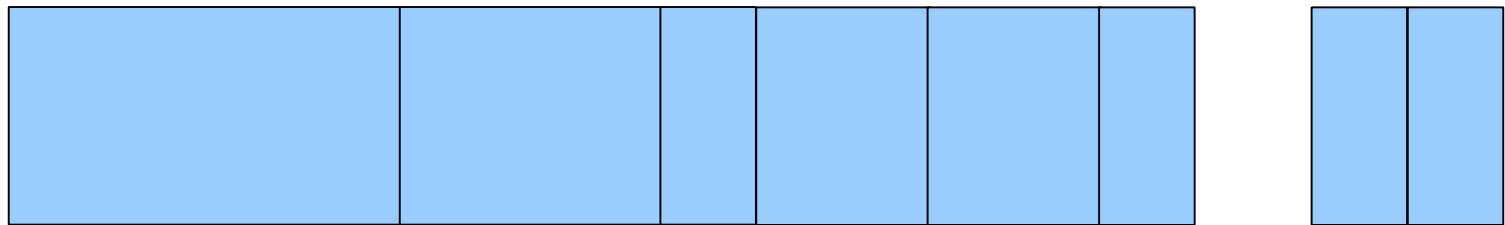


1

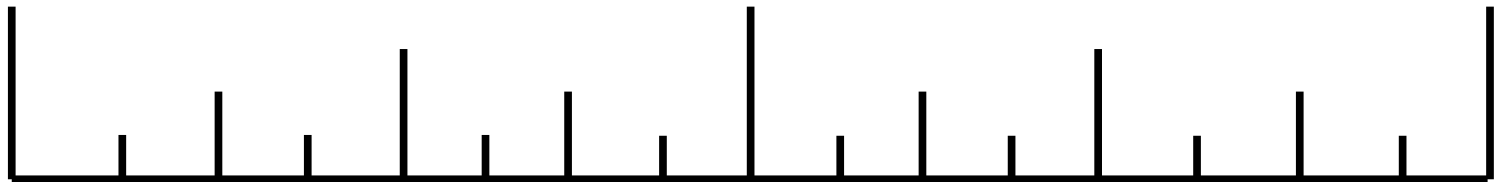


```
makeBeat(ELEKTRO_HOUSE_DRUMS3_2M, 1, 1,  
"0+++0+++0+0+0+++")
```

"0+++0++00+0+0-00"

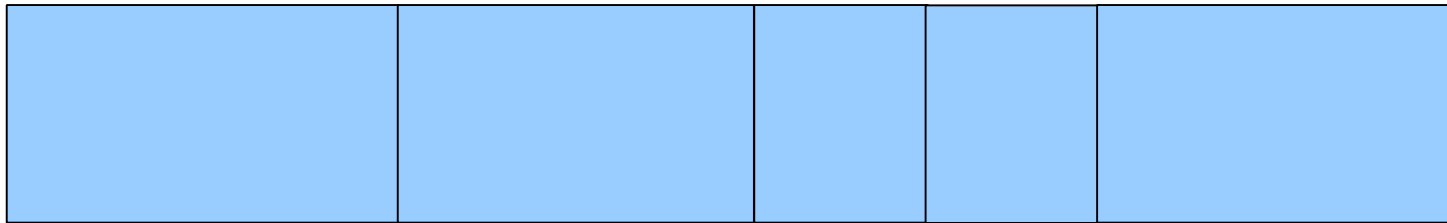


1

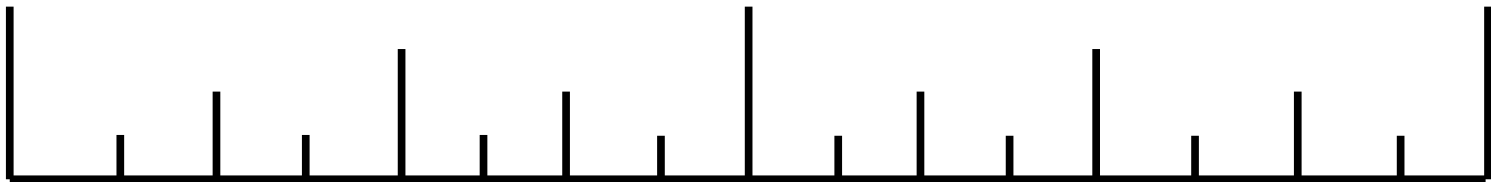


```
makeBeat(ELEKTRO_HOUSE_DRUMS3_2M, 1, 1, "0+++0++00+0+0-00")
```

"0+++0+++0+0+0+++"

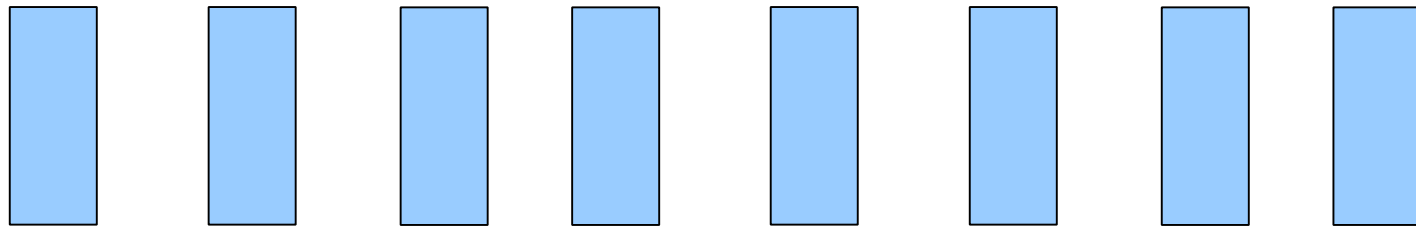


1

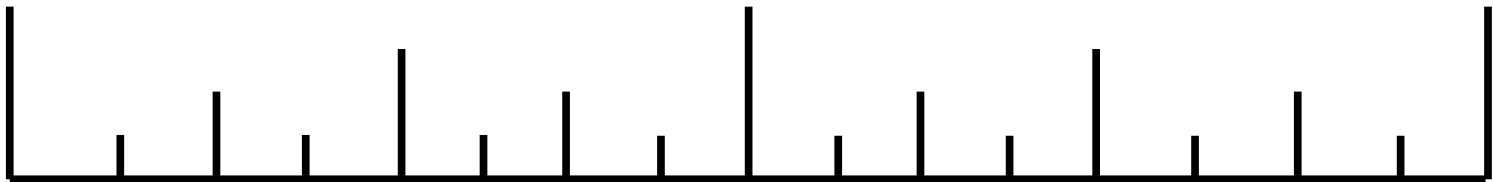


```
makeBeat(ELEKTRO_HOUSE_DRUMS3_2M, 1, 1, "0+++0+++0+0+0+++")
```

"0-0-0-0-0-0-0-0-0-"



1



```
makeBeat(ELEKTRO_HOUSE_DRUMS3_2M, 1, 1, "0-0-0-0-0-0-0-0-0-")
```


For Loops: Skip Counting

```
fillDrum = HIP_HOP_SYNTHDRUMS2_2M
beat = "0+++0+++0-000+00"

for measure in range((1, 9, 2)):
    makeBeat(fillDrum, 1, measure, beat)
```

measure is the “index variable” = assigned values from the range()

(1, 9, 2) means start counting at 1,
end before 9 [meaning 8] and skip count by 2:

(1, 3, 5, 7)

Functions: Recycle and Reuse!

```
def sectionA(start, end):
    stompDrums = HIPHOP_STOMP_BEAT_002
    bongoDrums = HIPHOP_DUSTYPERCUSSION_002
    keys = HIPHOP_SOLOMOOGLEAD_001
    scratch = ELECTRO_SFX_WHITE_NOISE_SCATTER_002

    fitMedia(stompDrums, 1, start, end)
    fitMedia(bongoDrums, 2, start, end)
    fitMedia(keys, 3, start, end)

    for measure in range(start, end):
        if measure % 2 == 0:
            fitMedia(scratch, 4, measure, measure+1)
```

Now I can use this section anywhere!

```
sectionA(1, 9)
sectionA(17, 25)
```

Creating a Function

1. Definition:

```
def sectionA(start, end):
```

2. Decide on Variables for Music

```
stompDrums = HIPHOP_STOMP_BEAT_002  
bongoDrums = HIPHOP_DUSTYPERCUSSION_002  
keys = HIPHOP_SOLOMOOGLEAD_001  
scratch = ELECTRO_SFX_WHITE_NOISE_SCATTER_002
```

3. Write fitMedia() calls

```
fitMedia(stompDrums, 1, start, end)  
fitMedia(bongoDrums, 2, start, end)  
fitMedia(keys, 3, start, end)
```

4. Write any For Loops

```
for measure in range(start, end):  
    if measure % 2 == 0:  
        fitMedia(scratch, 4, measure, measure+1)
```

5. Set Effects

Rhythm Effects

```
# Define List of Values for Effects
valueList = [1000, 20000]
panList = [-100, 100]

# Define BeatString for Effects
filterString = "0011001100110011"

# Music for Track
fitMedia(EIGHT_BIT_ANALOG_DRUM_LOOP_003, 1, 1, 9)

# For Loop to call Effects
for m in range(1, 9):
    rhythmEffects(1, FILTER, FILTER_FREQ, valueList, m, filterString)
    rhythmEffects(1, PAN, LEFT_RIGHT, panList, m, filterString)
```

Exercises for Classroom

- Create EarSketch Account
- Mix1: (AB Section Exercise)
 - Use fitMedia and setEffect
 - Music in sections
- Mix2: (makeBeat and For Loop Exercise)
 - makeBeat
 - For Loop Structure
- Mix3: (Defining Functions Exercise)
- Final Mix

Final Mix Project Goal

- Define Three Functions
 - sectionA(start, end)
 - sectionB(start, end)
 - sectionC(start, end)
- Each function will have at least 3 musical clips
- At least one function will use a for loop and makeBeat
- At least one function will use effects

- Call your functions to create a music mix
 - ABABCBB
 - At least 64 Measures