

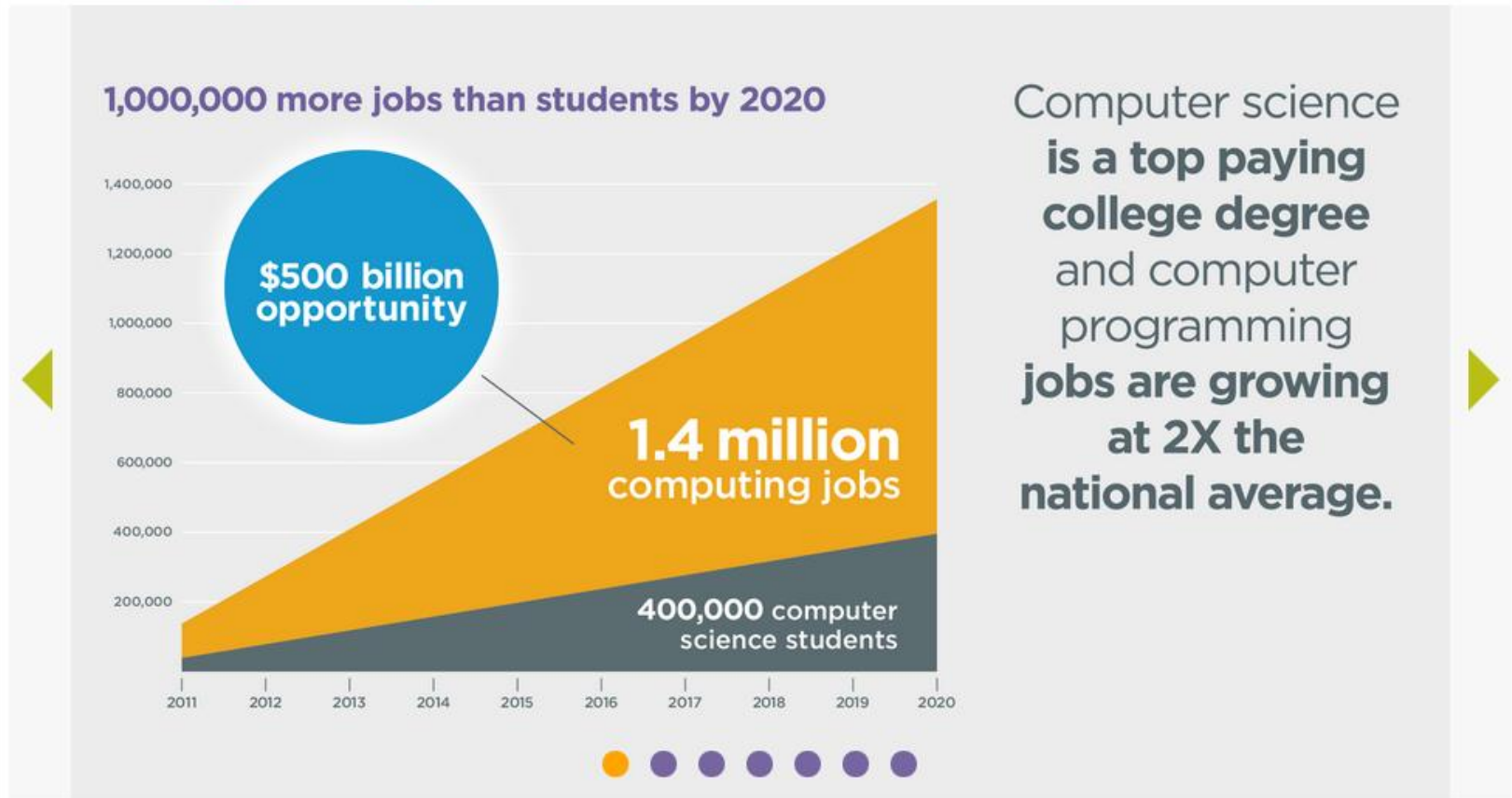
EARSKETCH MUSIC MIXING WITH PYTHON PROGRAMMING

Mr. Michaud

Georgia Institute of Technology

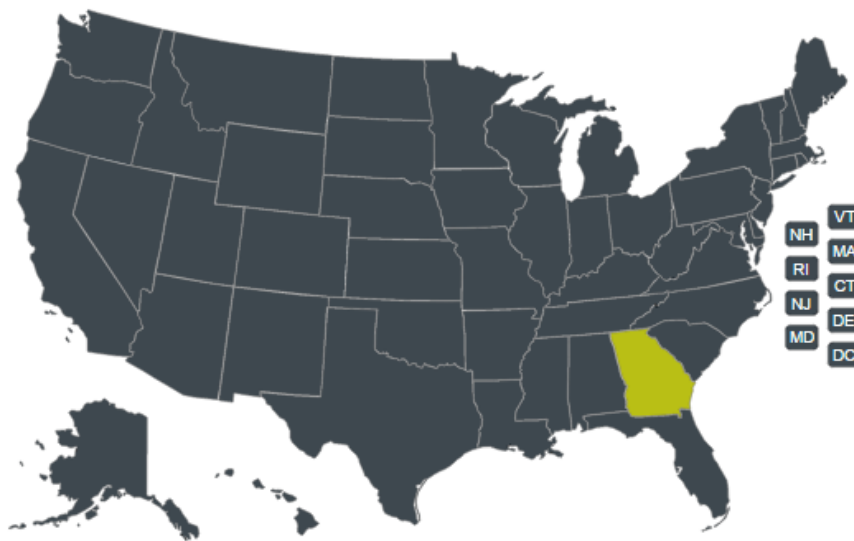
Why Learn to Program?

What's wrong with this picture? Share these stats.



Why Learn to Program?

Choose a State



Georgia

20,371 open computing jobs
(growing at 4.8x the state average)

1,836 computer science graduates

133 schools teach computer science

- ✓ CS counts as math or science credit
- ✓ Clear certification pathways for CS teachers
- ✓ CS curriculum standards

[Georgia fact-sheet](#) ↓ | [More advocacy tools](#)

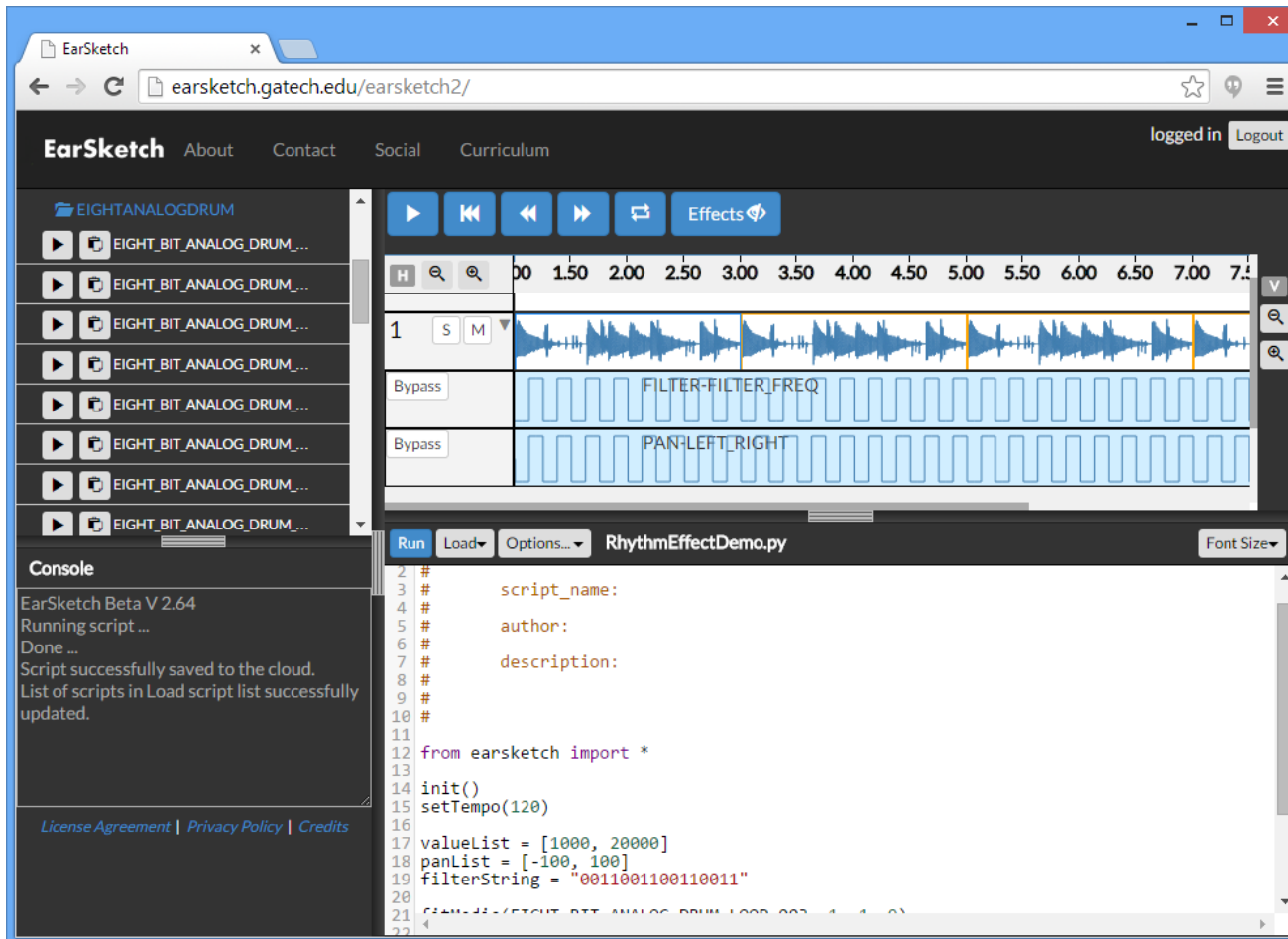
Georgia is one of 25 states where students **can** count computer science for credit towards high school graduation!

[Take Action](#)

Sources: [The Conference Board](#), [National Science Foundation](#), [Code.org database](#). [More info.](#)

What is EarSketch?

- Online Programming and Music Mixing Workstation



The screenshot displays the EarSketch web interface in a browser window. The URL is `ears sketch.gatech.edu/ears sketch2/`. The interface includes a navigation menu with links for "About", "Contact", "Social", and "Curriculum". A "logged in" status and "Logout" button are visible in the top right. The main workspace features a timeline from 0.00 to 7.00 seconds. A track labeled "1" contains an audio waveform. Below the waveform are two effect modules: "FILTER-FILTER_FREQ" and "PAN-LEFT_RIGHT", each with a "Bypass" button. A console window on the left shows messages: "EarSketch Beta V 2.64", "Running script ...", "Done ...", "Script successfully saved to the cloud.", and "List of scripts in Load script list successfully updated." At the bottom, there are links for "License Agreement", "Privacy Policy", and "Credits". The code editor at the bottom right shows the following Python code:

```
Run Load Options... RhythmEffectDemo.py Font Size
2 #
3 #   script_name:
4 #
5 #   author:
6 #
7 #   description:
8 #
9 #
10 #
11
12 from earsketch import *
13
14 init()
15 setTempo(120)
16
17 valueList = [1000, 20000]
18 panList = [-100, 100]
19 filterString = "0011001100110011"
20
21 filterList = [EIGHT_BIT_ANALOG_DRUM_LOAD_001_1_1_0]
22
```

-Used at Georgia Tech to teach Digital Music Mixing.

-Tool for Programming

-Free!

-Web based

EarSketch Workstation: 3 Windows

The screenshot displays the EarSketch Workstation interface, which is divided into three main functional windows:

- Audio Clips:** Located on the left side, this window contains a list of audio clips, each represented by a play button and a file icon. The clips are labeled "EIGHT_BIT_ANALOG_DRUM_...".
- Audio Workstation:** The central window, featuring a timeline from 0.00 to 24.00 seconds. It includes playback controls (play, stop, previous, next, refresh) and an "Effects" menu. The main area shows three tracks (1, 2, 3) with audio waveforms and a "VOLUME-GAIN" envelope. A red vertical line indicates the current playback head position.
- Coding Area:** The bottom window, displaying a Python script for a project named "Final EarSketch Project". The code includes comments for script name, author, and description, and defines a function for sectionA that uses the "TECHNO_MAINLOOP_001" audio clip.

Additional interface elements include a top navigation bar with "EarSketch", "About", "Contact", "Social", and "Curriculum" links, a "logged in" status with a "Logout" button, and a "Console" window on the left showing system messages like "EarSketch Beta V 2.64" and "Loading File TECHNO_MAINLOOP_001 done ...".

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** – Names for information stored by program

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

- **Tabs:** Enclose code in sections

EarSketch Python Functions

- `init()`
Start New Reaper File
- `setTempo(120)`
Beats per minute in remix
- `println("Hello")` -
Prints message in console

EarSketch Python Functions

- `fitMedia(file, track, start, end)`
- `makeBeat(file, track, measure, beatString)`
- `fitMedia(music, track, start, end)`
- `setEffect(track, effect, parameter, vS, mS, vE, mE)`
- `rhythmEffects(track, effect, parameter, list, measure, string)`

“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

- VOLUME
 - GAIN
- DELAY
 - DELAY_TIME
- CHORUS
 - CHORUS_LENGTH
 - CHORUS_NUMVOICES
- DISTORTION
 - DISTO_GAIN
- FILTER
 - FILTER_FREQ
- PAN
 - LEFT_RIGHT

“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.

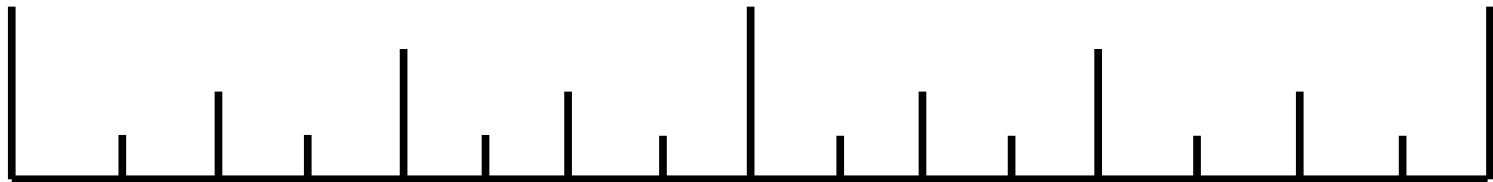
“-” Means $1/16^{\text{th}}$ measure of rest.

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



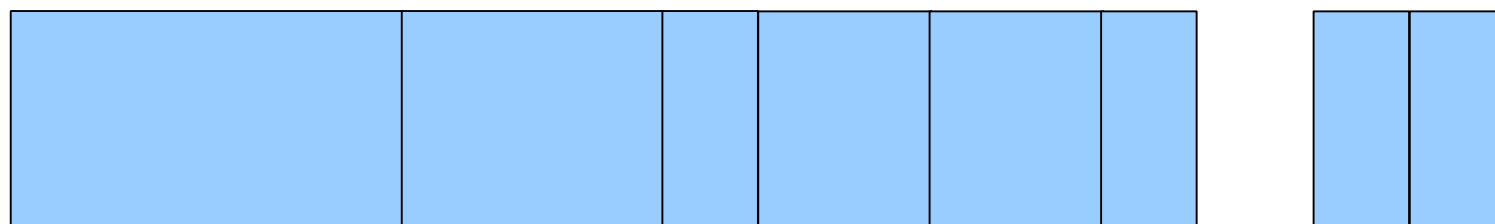
1

2



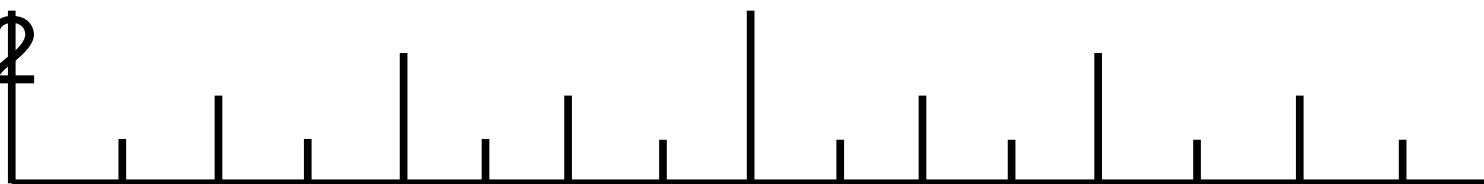
```
makeBeat(ELEKTRO_HOUSE_DRUMS3_2M, 1, 1,  
"0+++0+++0+0+0+++")
```

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



1

2



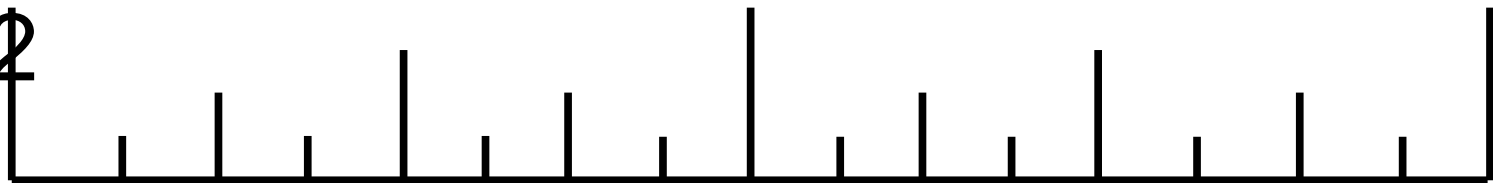
```
makeBeat(ELEKTRO_HOUSE_DRUMS3_2M, 1, 1, "0+++0++00+0+0-00")
```

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



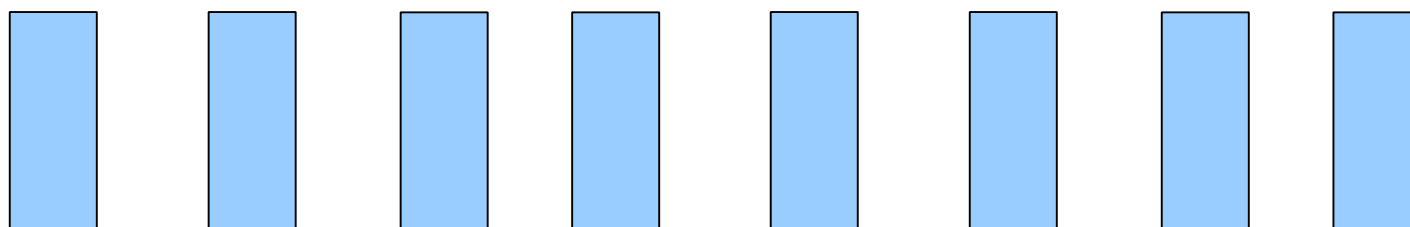
1

2



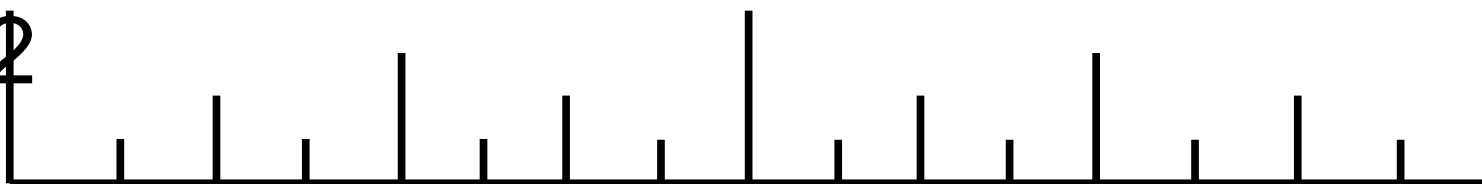
```
makeBeat(ELEKTRO_HOUSE_DRUMS3_2M, 1, 1, "0+++0+++0+0+0+++")
```


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



1

2



```
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 Workshop

- 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