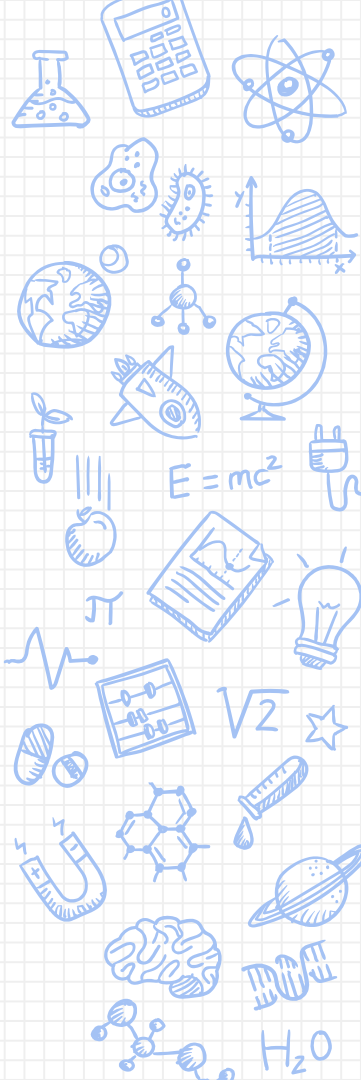
# Going Further With EarSketch!

# Goals for Today

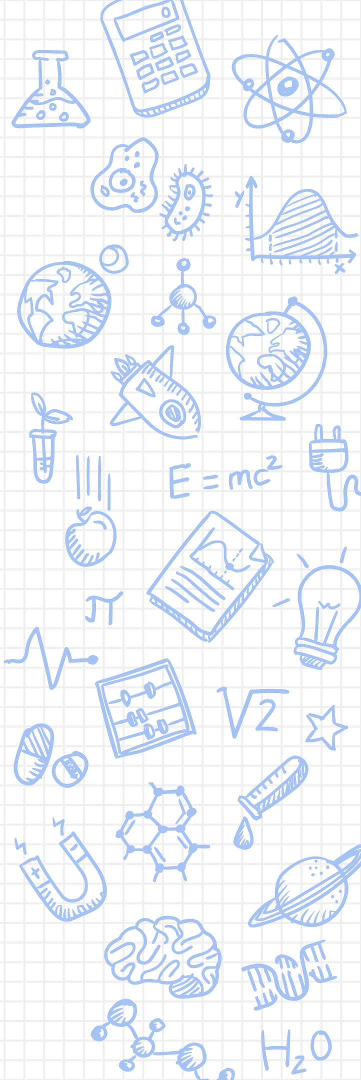
- Recap yesterday
- Functions (reusable sauces)
- Live coding demo
- Team projects



# Yesterday's Recap

Turn and tell someone **2 positive things** and **1 muddy point** you encountered working with EarSketch, and two opportunities for improvement. (30 seconds)

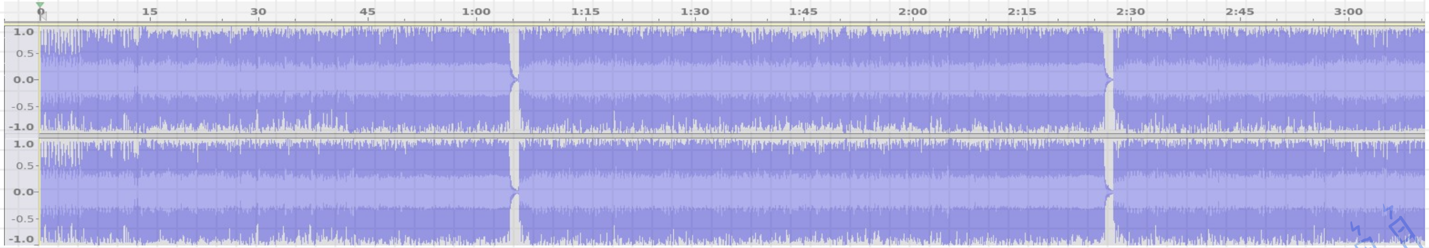
**Share out as a large group...**



# Coding Repetition

## Looping Example 1:

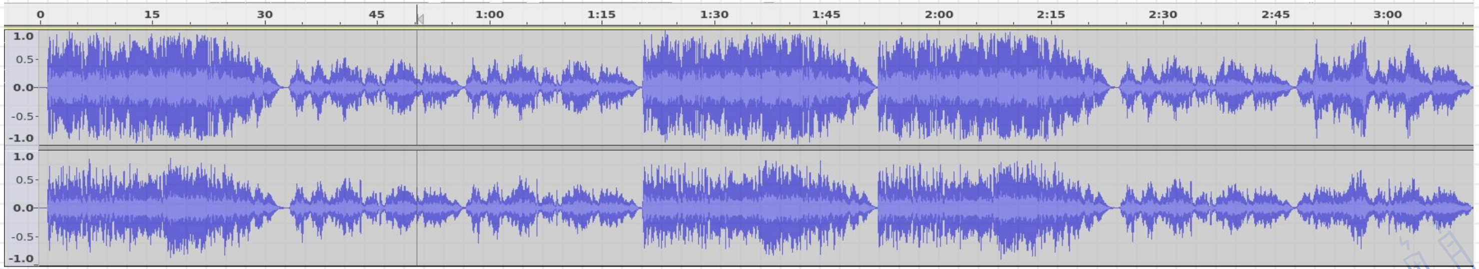
*Your favorite Song...which also happens to illustrate a few loops!*



# Coding Repetition

## Looping Example 2:

*Here is another song....does it have loops?*



# Looping Code Comparison

```
drum1 = ELECTRO_DRUM_MAIN_BEAT_008
drum2 = ELECTRO_DRUM_MAIN_BEAT_007

fitMedia(drum1, 1, 1, 1.5)
fitMedia(drum2, 1, 1.5, 2)
fitMedia(drum1, 1, 2, 2.5)
fitMedia(drum2, 1, 2.5, 3)
fitMedia(drum1, 1, 3, 3.5)
fitMedia(drum2, 1, 3.5, 4)
. . .
fitMedia(drum1,  8, 8.5)
fitMedia(drum2, 8.5, 9)
```

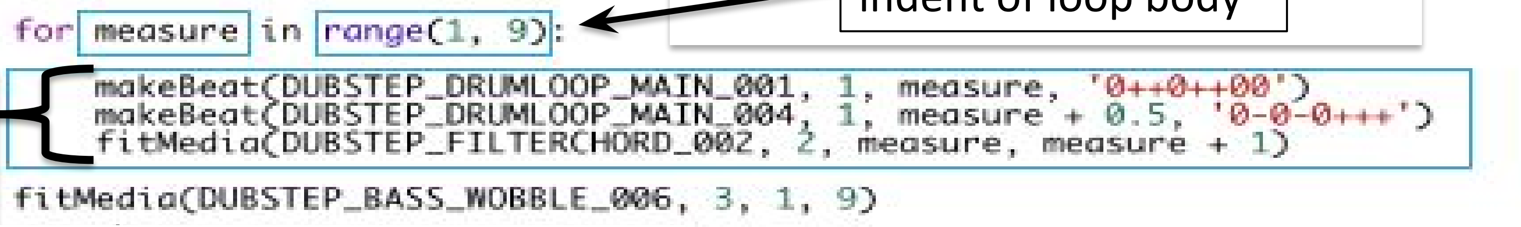
```
drum1 = ELECTRO_DRUM_MAIN_BEAT_008
drum2 = ELECTRO_DRUM_MAIN_BEAT_007

for measure in range (1, 9):
  fitMedia(drum1, 1, measure, measure + 0.5)
  fitMedia(drum2, 1, measure + 0.5 , measure + 1
```

# For Loop
# (Control Flow Structure)

Loop Counter Variable (Note: could be any variable name)

Range Function (Note: goes through loop from 1 to 8)

Colon is used in Python to indicate indent of loop body



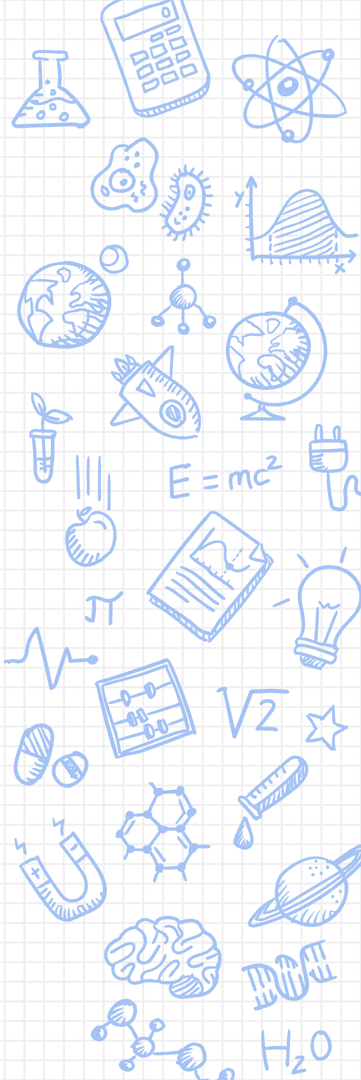
```
for measure in range(1, 9):
    makeBeat(DUBSTEP_DRUMLOOP_MAIN_001, 1, measure, '0++0++00')
    makeBeat(DUBSTEP_DRUMLOOP_MAIN_004, 1, measure + 0.5, '0-0-0+++')
    fitMedia(DUBSTEP_FILTERCHORD_002, 2, measure, measure + 1)
fitMedia(DUBSTEP_BASS_WOBBLE_006, 3, 1, 9)
```

Loop Body (indented)

Not in loop (Not indented)

# Functions with DJ Marissa

Go to Earsketch tutorial #10.



# Live Coding Concert

by

DJ Amit

# Application to the Classroom:

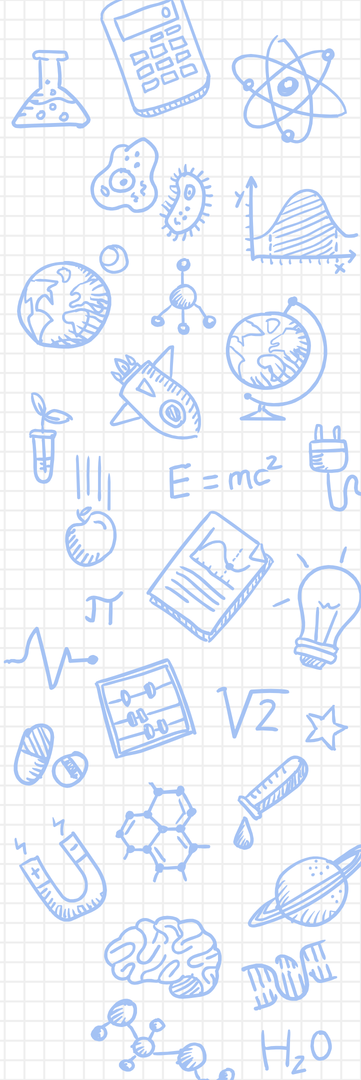
1 minute mini-discussion:

**How might this program be used in different content areas? (Math, Science, Language Arts, Art, PE, Science)**

**Share out...**

In small groups of 2 or 3:

**Decide on one of these contexts, and design a quick culminating activity, incorporating EarSketch.**



# Making Custom Beats: makeBeat

`makeBeat()` allows us to compose music note by note instead of at the measure level, perfect for drum beats. This approach is often called **step sequencing** in music production.

# Strings

We need to understand the **string** data type to use `makeBeat()`. In Python, a string is a series of characters with single or double quotation marks around it, like `"Hello World!"` or `'This is a test sentence'`. Strings are often used in programming to represent non-numerical data such as words, but can include numbers. For example, the string `"123 Ferst Dr."` could represent an address, using numbers, spaces, letters, and punctuation. Like numbers (and other types of data), strings can be assigned to variables: `address = "123 Ferst Dr."`

Note that `5` and `"5"` mean very different things to the computer: `5`is a number it can perform math on, while math operations won't work with the string `"5"`. For example, `5 + 6` is evaluated as 11, while`"5" + "6"` is evaluated as `"56"`.

# makeBeat

Like our other functions, `makeBeat()` also takes four arguments:

1. **Clip Name**
2. **Track Number**
3. **Measure Number**: `makeBeat()` only requires a starting measure; the string length determines the end measure.
4. **Beat String**

**Example:**

```
for measure in range(1,4):
    makeBeat(synth, 1, measure, beat1)
    makeBeat(cymbal, 2, measure, beat2)
```

# String Concatenation

**Concatenation** is a means to link strings together. In doing so, a new string is formed. For example, concatenating the strings "hello" and "world" yields "helloworld". Strings are concatenated with the +symbol, like in the following example. To view the new string we use the `print` statement, which prints the string to the console.

Example code:

```
#Setup
from earsketch import *
init()
#Concatenation
stringA = "Computer"
stringB = "Science!"
newString = stringA + " " + stringB # concatenating stringA and stringB with a space in between
print newString
#Finish
finish()
```