EarSketch: Loops and Conditionals
Goals for This Session

● Recap intro session
● Loops and conditionals to enhance your music
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

```python
# loop-demo_1.py

# loop demo code

# Initialize variables

# Define the drum1 and drum2 variables

drum1 = ELECTRO_DRUM_MAIN_BEAT_008

drum2 = ELECTRO_DRUM_MAIN_BEAT_007

# Function to fit media

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)

# Loop through measures

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

```python

# End of loop demo code
```
Loop Exercises (1)

- Run the previous loop demo after adding a print statement
  
  ```python
  for measure in range (1, 9):
      fitMedia(drum1, 1, measure, measure + 0.5)
      fitMedia(drum2, 1, measure + 0.5 , measure + 1)
      print measure
  ```

- Draw out what is happening on paper (or board)
Loop Exercises (2)

Draw the music for the following code:

```python
# loop_demo_2.py
setTempo(128)
drum1 = ELECTRO_DRUM_MAIN_BEAT_008
bass1 = ELECTRO_ANALOGUE_BASS_003
bass2 =

for measure in range(1, 9):
    fitMedia(drum1, 1, measure, measure + 1)
    fitMedia(bass1, 2, measure, measure + 0.5)

Add another bass instrument to fill in the missing half measures
```
Loop Exercises (3)

Add a new track and a lead instrument on it to make a richer sound

```python
#loop_demo_2.py
setTempo(128)

for measure in range(1, 9):
    fitMedia(drum1, 1, measure, measure + 1)
    fitMedia(bass1, 2, measure, measure + 0.5)
```
For Loop
(Control Flow Structure)

measure: Loop Counter Variable

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

Colon is used in Python to indicate indent of loop body

Loop Body (indented)

Not in loop (not indented)

```python
for measure in range (1, 9):
    fitMedia(drum1, 1, measure, measure + 0.5)
    fitMedia(drum2, 1, measure + 0.5, measure + 1)
    fitMedia(piano1, 2, 1, 9)
```
Play the following script. Note the difference from loop_demo_2

#loop_demo_3.py
init()
setTempo(128)
drum1 = ELECTRO_DRUM_MAIN_BEAT_008
bass1 = ELECTRO_ANALOGUE_BASS_020
for measure in range (1, 9, 2):
    fitMedia(drum1, 1, measure, measure + 2)
    fitMedia(bass1, 2, measure, measure + 1)
finish()

Again, add another bass in the empty measures (on the same track as bass1)
Conditionals

Boolean expressions evaluate to either true or false. Our programs can make decisions based on some condition.

Example: To alternate behaviour every iteration of a loop to make the music more varied
Conditionals

In Python

```python
if(x == 5):
    fitMedia(piano1, 3, start, end)
else:
    fitMedia(piano2, 3, start, end)
```
Conditionals Exercise (1)

Play the following music and observe that it is a bit repetitious...

```python
# see loop_demo_4.py
drum1 = HOUSE_MAIN_BEAT_004
drum2 = HOUSE_BREAKBEAT_022
piano1 = HOUSE_ACOUSTIC_PIANO_004
for measure in range(start,end):
    fitMedia(drum1, 1, measure, measure + 1)
    fitMedia(drum2, 2, measure, measure + 1)
    fitMedia(piano1, 3, measure + 2, measure + 3)
```

We want to vary the piano part every other measure.
Conditionals Exercise (1) contd.

Pick another piano and use an if statement to alternate between the two every other measure

// the code before same as before  
piano2 = HOUSE_ACOUSTIC_PIANO_005
for measure in range(1, 9):
    fitMedia(drum1, 1, measure, measure + 1)
    fitMedia(drum2, 2, measure, measure + 1)
if (measure % 2 == 0):
    fitMedia(piano1, 3, measure, measure + 1)
else:
    fitMedia(piano2, 3, measure, measure + 1)
1. What if we want to switch to the second piano every third measure?
2. What if we want to switch between piano1 and piano2 every other measure during measures 1-8 and then switch between piano1 and piano3 during measures 9-16?
3. Add a volume gain effect to have the drums start up slowly and fade away at the end
4. Another nice effect is to stagger the start and end of instruments (See loop_demo_5.py)
More info on loops

Tutorial section Unit2, Ch 12 - Looping.