

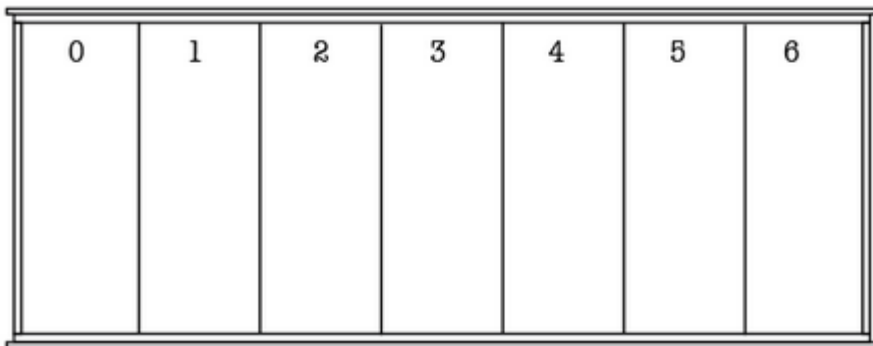
Directions for EarSketch Lists

(Taken from EarSketch website <http://ears sketch.gatech.edu/category/learning/lists>)

EarSketch 2

Description:

In programming Lists are data structures that hold multiple variables in one place. A list can be thought of as a bookshelf in which the numbered spaces represent areas in which we can store values:

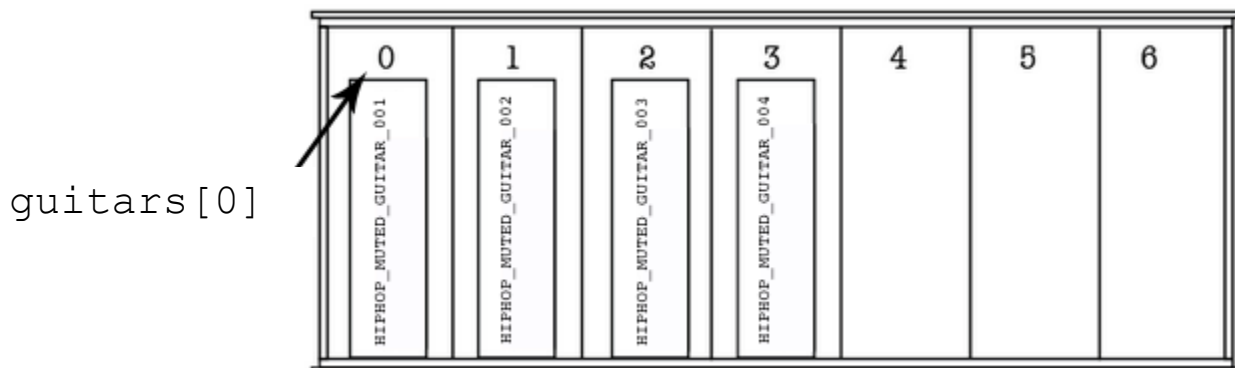


Note that the 'first' space in the bookshelf is labeled with a 0. In lists (and in most computing languages) we start counting from 0. This is called 'zero based' counting.

We declare a list similar to a variable. For example:

```
# A list of guitars
guitars = [HIPHOP_MUTED_GUITAR_001, HIPHOP_MUTED_GUITAR_002,
           HIPHOP_MUTED_GUITAR_003, HIPHOP_MUTED_GUITAR_004]
```

Note that 'guitars' is the name of the list. The values in the brackets represent the items we are placing into the list.



The numbered spaces are called 'indexes'. Therefore:

```
guitars[0] = HIPHOP_MUTED_GUITAR_001  
guitars[1] = HIPHOP_MUTED_GUITAR_002  
guitars[2] = HIPHOP_MUTED_GUITAR_003  
guitars[3] = HIPHOP_MUTED_GUITAR_004
```

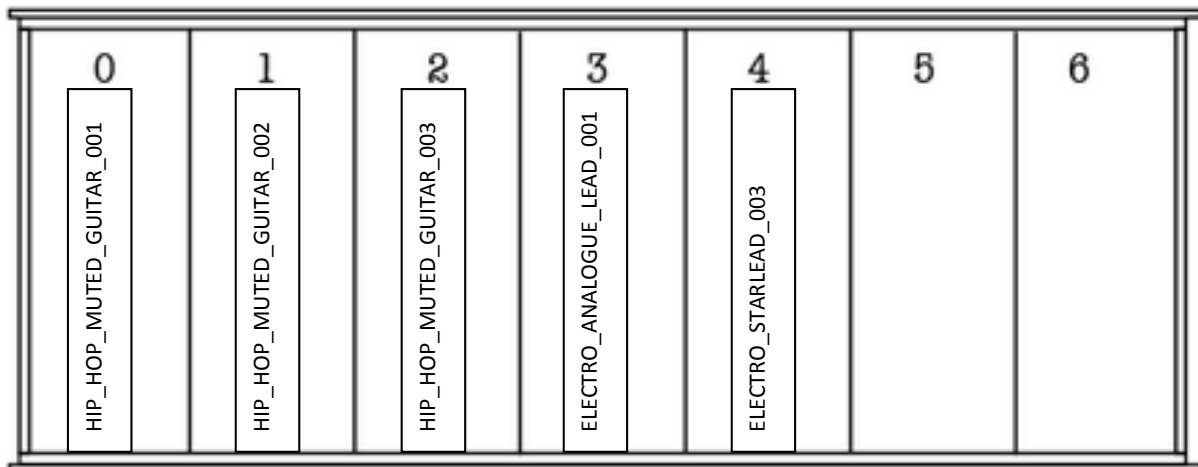
We can change values in a list by using the = assignment:

```
guitars[3] = ELECTRO_ANALOGUE_LEAD_001
```

Now the 4th space (index 3) is equal to ELECTRO_ANALOGUE_LEAD_001. We can also add items to the list using the .append function:

```
guitars.append(ELECTRO_STARLEAD_003)
```

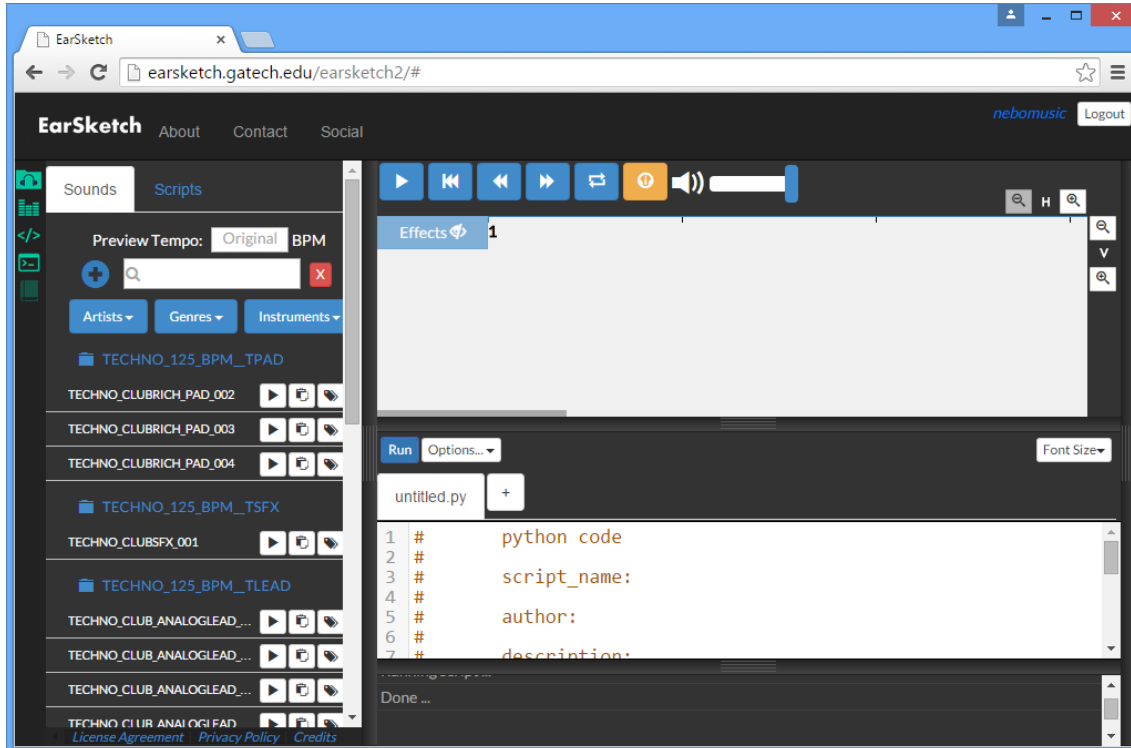
Now our list looks like:



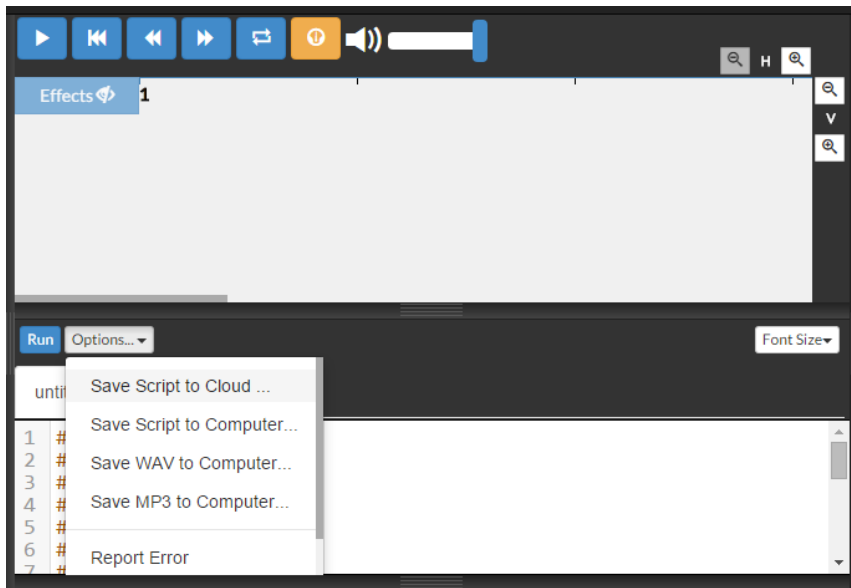
In this exercise we will create a list and then write a function to iterate (go through) the list and place each index on its own track using the fitMedia() command. We will also use Lists with makeBeat()

Process:

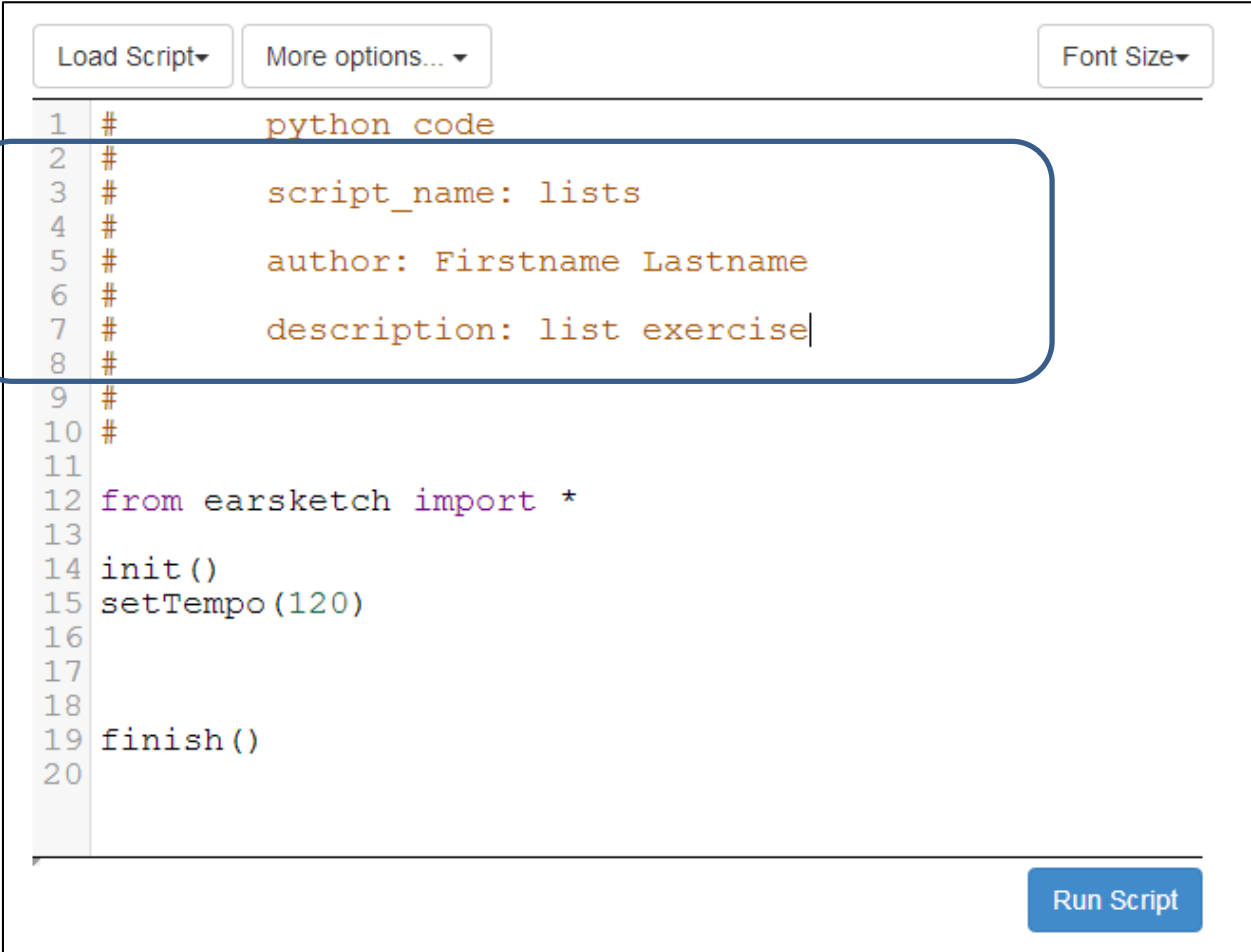
1. Go to <http://earskech.gatech.edu/earskech2/#/viewport> and login with your EarSketch Account and password.



2. Select 'Options -> Save Script to Cloud



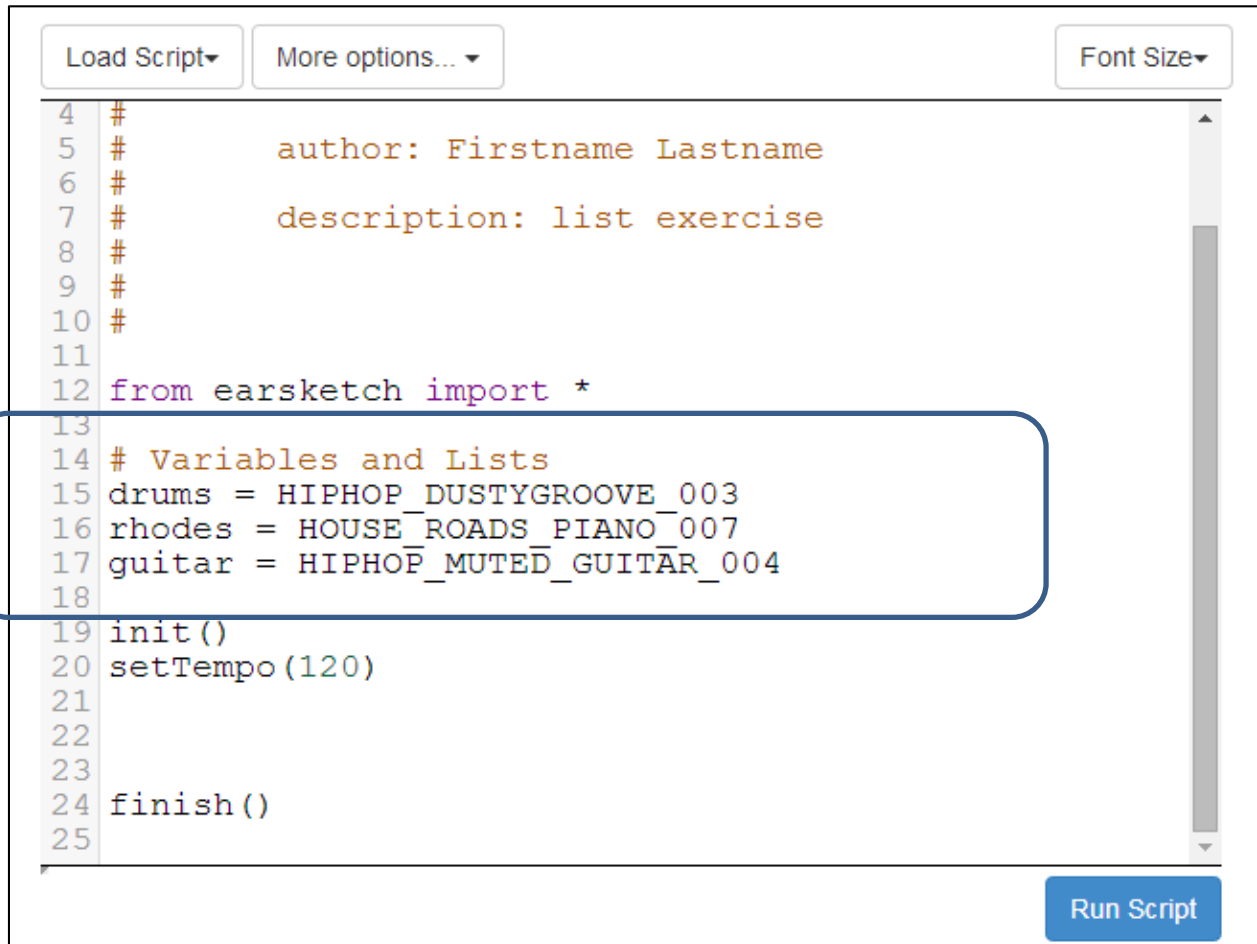
3. Name the script 'lastname_lists'
4. Fill out the information in the top of the script:



```
1 # python code
2 #
3 # script_name: lists
4 #
5 # author: Firstname Lastname
6 #
7 # description: list exercise|
8 #
9 #
10 #
11
12 from earsketch import *
13
14 init()
15 setTempo(120)
16
17
18
19 finish()
20
```

Buttons: Load Script, More options..., Font Size, Run Script

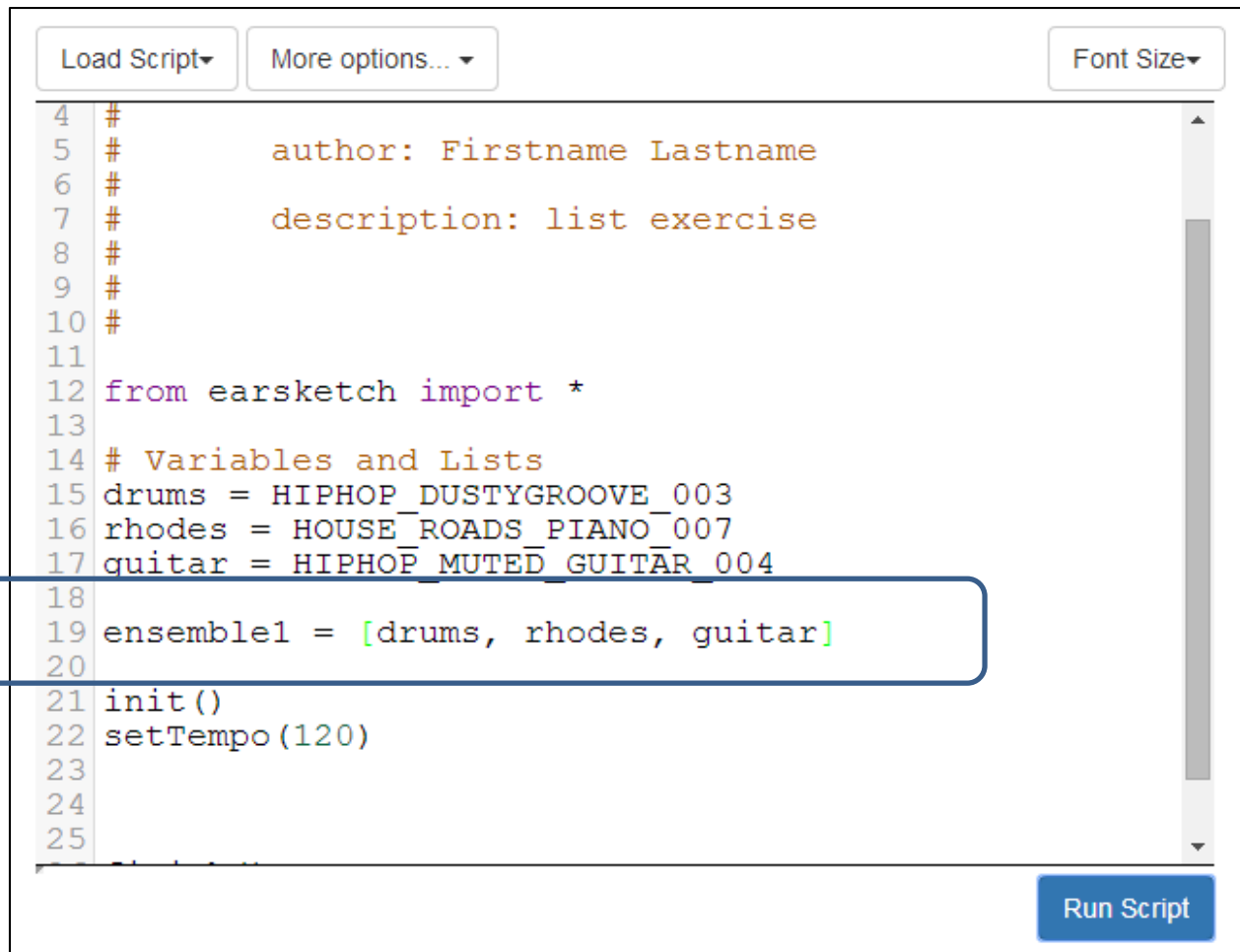
5. We will now define variables for drums, keys, and guitar. You may pick any three sounds you wish. Note that we will define them above the `init()` function. Usually we want to define variables, lists, and functions above the `init()` function and then call the functions below `setTempo()`.



```
4 #
5 #     author: Firstname Lastname
6 #
7 #     description: list exercise
8 #
9 #
10 #
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18
19 init()
20 setTempo(120)
21
22
23
24 finish()
25
```

Run Script

6. We will now define a list called 'ensemble1' to hold these variables:



The image shows a code editor window with a white background and a dark border. At the top, there are three buttons: 'Load Script' with a dropdown arrow, 'More options...' with a dropdown arrow, and 'Font Size' with a dropdown arrow. The main area contains a text editor with a light gray background and a vertical scrollbar on the right. The code is as follows:

```
4 #
5 #     author: Firstname Lastname
6 #
7 #     description: list exercise
8 #
9 #
10 #
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18
19 ensemble1 = [drums, rhodes, guitar]
20
21 init()
22 setTempo(120)
23
24
25
```

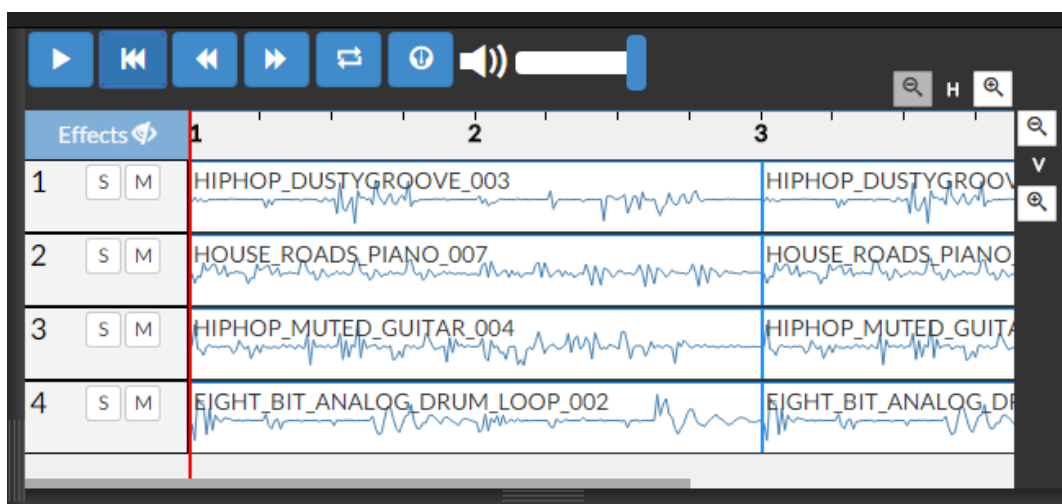
A blue rounded rectangle highlights the line `ensemble1 = [drums, rhodes, guitar]`. At the bottom right, there is a blue button labeled 'Run Script'.

7. We will now move below the `setTempo()` function and write the `fitMedia` calls to place the items from the `ensemble1` list into the tracks. Note the patterns of the indexes and tracks:

```
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18
19 ensemble1 = [drums, rhodes, guitar]
20
21 init()
22 setTempo(120)
23
24 # Set Music
25 fitMedia(ensemble1[0], 1, 1, 5)
26 fitMedia(ensemble1[1], 2, 1, 5)
27 fitMedia(ensemble1[2], 3, 1, 5)
28
29
30
31 finish()
32
```

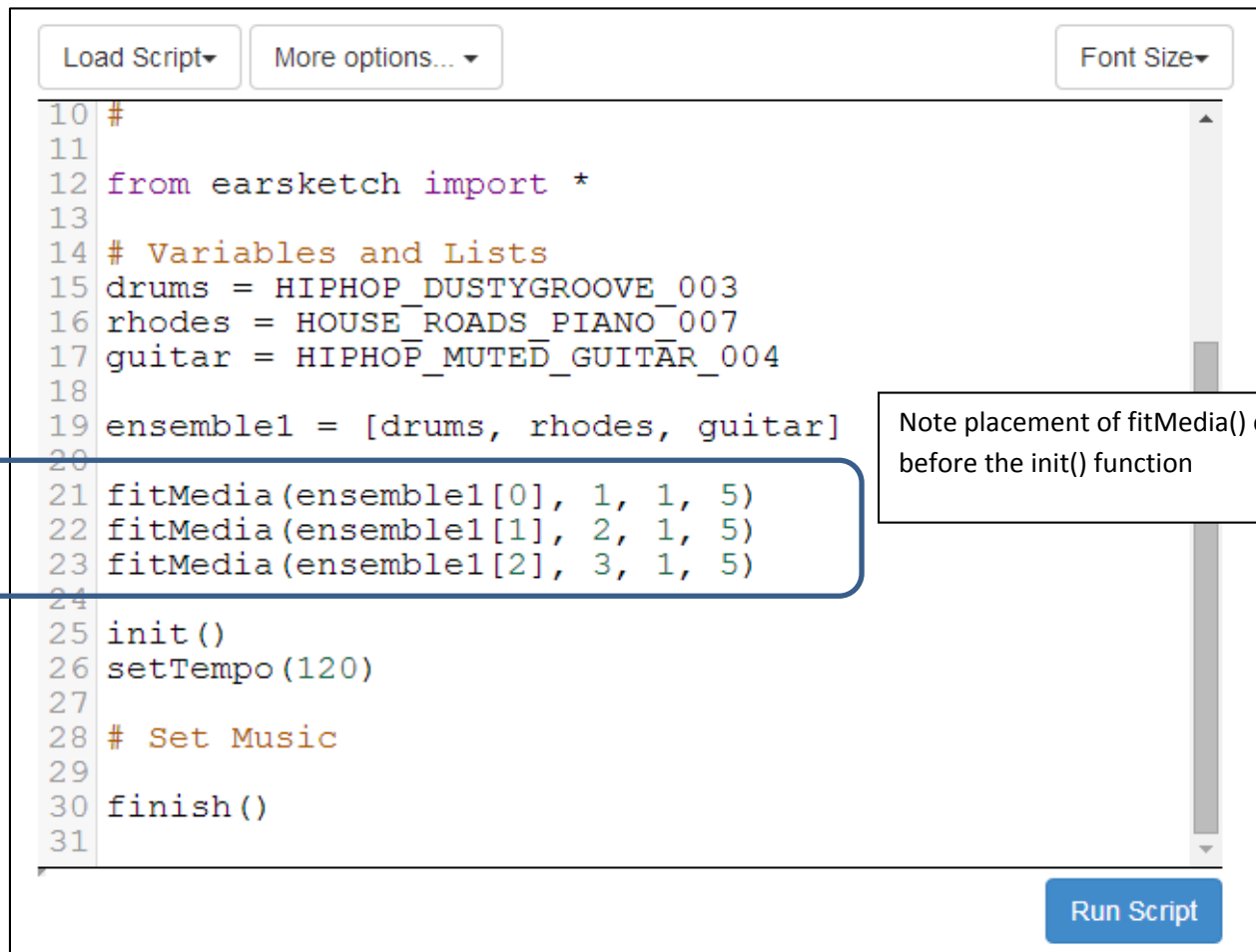
Run Script

8. Click 'Run' and the play icon to listen to your music.



9. Note that we have used `fitMedia()` in the same way as previous mixes. Now we will refactor the code to create a function that takes a list and places the music on each track depending on how many objects are in the list.

First, cut and paste the `fitMedia` calls before the `init()`:

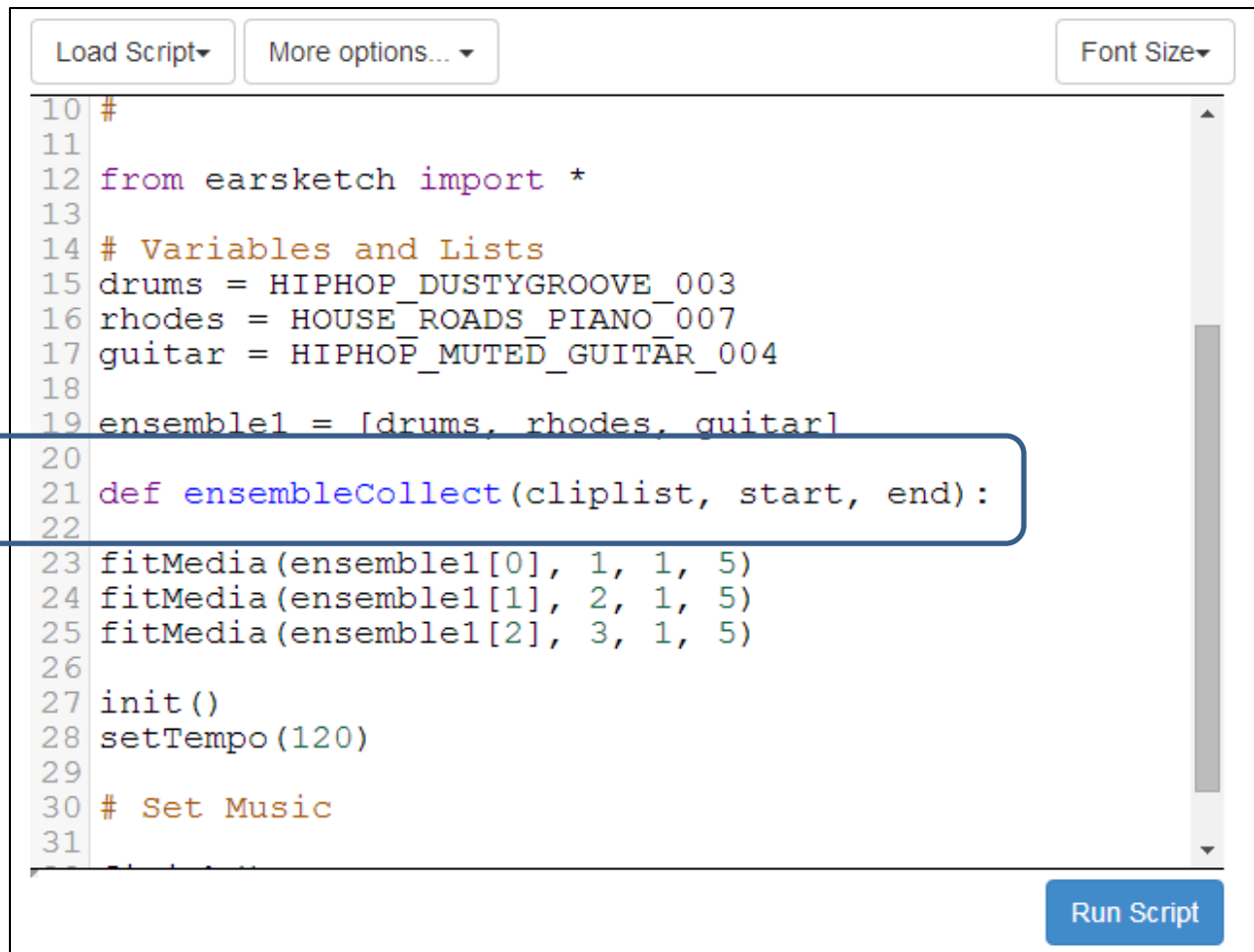


```
10 #
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18
19 ensemble1 = [drums, rhodes, guitar]
20
21 fitMedia(ensemble1[0], 1, 1, 5)
22 fitMedia(ensemble1[1], 2, 1, 5)
23 fitMedia(ensemble1[2], 3, 1, 5)
24
25 init()
26 setTempo(120)
27
28 # Set Music
29
30 finish()
31
```

Note placement of `fitMedia()` calls before the `init()` function

Run Script

10. Add a line of code above the fitMedia() calls to define a function called ensembleCollect()



```
10 #
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18
19 ensemble1 = [drums, rhodes, guitar]
20
21 def ensembleCollect(cleplis, start, end):
22
23 fitMedia(ensemble1[0], 1, 1, 5)
24 fitMedia(ensemble1[1], 2, 1, 5)
25 fitMedia(ensemble1[2], 3, 1, 5)
26
27 init()
28 setTempo(120)
29
30 # Set Music
31
```

Load Script ▾ More options... ▾ Font Size ▾

Run Script

11. Delete two fitMedia() and tab over the remaining fitMedia() to make it part of the ensembleCollect() function.



```
10 #
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18
19 ensemble1 = [drums, rhodes, guitar]
20
21 def ensembleCollect(cliplist, start, end):
22     fitMedia(ensemble1[0], 1, 1, 5)
23
24
25
26 init()
27 setTempo(120)
28
29 # Set Music
30
31 finish()
```

Run Script

12. Now watch the power of the for loop! Modify the code to use a for loop and index variables to iterate through the list:

```
Load Script ▾ More options... ▾ Font Size ▾
10 #
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18
19 ensemble1 = [drums, rhodes, guitar]
20
21 def ensembleCollect(cliplist, start, end):
22     # Watch the power of the for loop
23     for clip in range(len(cliplist)):
24         track = clip + 1
25         fitMedia(cliplist[clip], track, start, end)
26
27
28 init()
29 setTempo(120)
30
31 # Set Music
```

Run Script

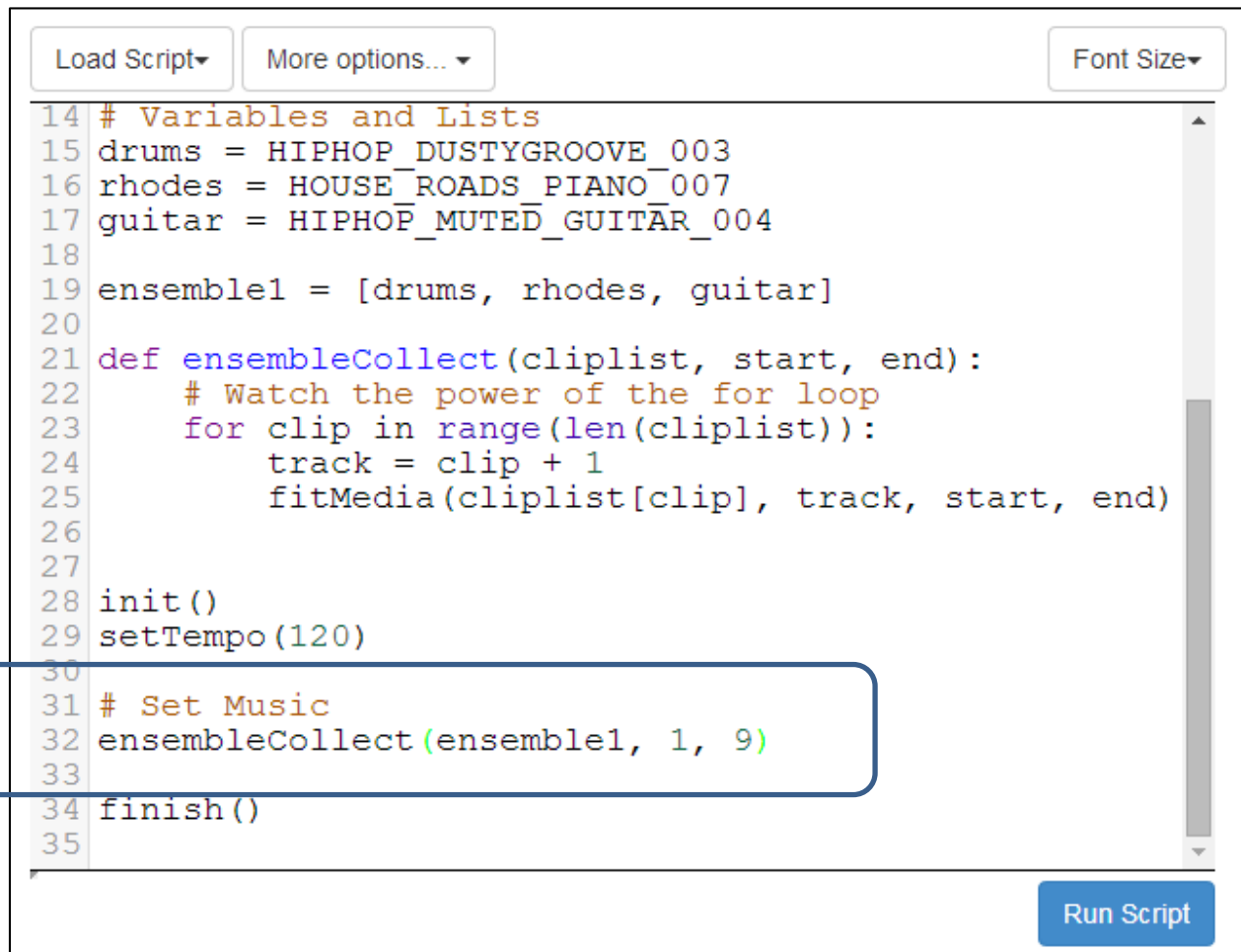
13. Now the ensembleCollect() function will take a list and:

Line 23: Count with the index 'clip' for the length of the cliplist (how many items)

Line 24: Make a variable 'track' and add 1 to clip (so track starts counting from 1)

Line 25: Place the music from the index 'clip' onto the track with the start and end

Go after the setTempo() and call the function to listen to your music.



```
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18
19 ensemble1 = [drums, rhodes, guitar]
20
21 def ensembleCollect(cliplist, start, end):
22     # Watch the power of the for loop
23     for clip in range(len(cliplist)):
24         track = clip + 1
25         fitMedia(cliplist[clip], track, start, end)
26
27
28 init()
29 setTempo(120)
30
31 # Set Music
32 ensembleCollect(ensemble1, 1, 9)
33
34 finish()
35
```

Run Script

14. Click 'Run Script' to listen to the mix.

The screenshot shows a DAW interface with a transport control bar at the top. The transport bar includes buttons for play, stop, previous, next, and a volume slider. Below the transport bar is a timeline with markers from 1 to 9. The main area displays three tracks:

Track	Label	Waveform
1	HIPHOP_DUSTYGROOV...	Waveform for track 1
2	HOUSE_ROADS_PIANO...	Waveform for track 2
3	HIPHOP_MUTED_GUITA...	Waveform for track 3

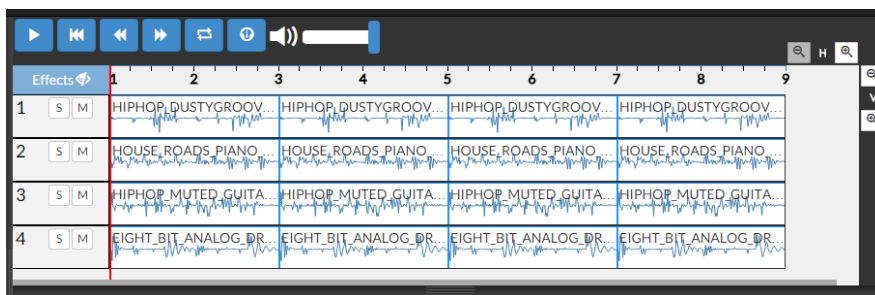
Each track has a volume knob and a solo button (S) and a mute button (M). The track 3 label includes the word 'MUTED'. The interface also features search icons and a 'H' button on the right side.

15. We will now demonstrate the power of the list and for loop. Let us say we want to add an instrument to our ensemble. We can add a new variable and then place this variable in the list. The function `ensembleCollect()` will automatically account for the new index. Modify your code to add an instrument variable and then place this variable in the list:

```
Load Script ▾ More options... ▾ Font Size ▾
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18 drums2 = EIGHT_BIT_ANALOG_DRUM_LOOP_002
19
20 ensemble1 = [drums, rhodes, guitar, drums2]
21
22 def ensembleCollect(cliplist, start, end):
23     # Watch the power of the for loop
24     for clip in range(len(cliplist)):
25         track = clip + 1
26         fitMedia(cliplist[clip], track, start, end)
27
28
29 init()
30 setTempo(120)
31
32 # Set Music
```

Run Script

16. Run the script and note that we now have a 4th track in the mix. The for loop in the `ensembleCollect` function used the new length of the `ensemble1` list (now 4) to create the new track.



17. We will now add another ensemble list to the mix. Select three more sounds and create new variables and a new list. My example is below:

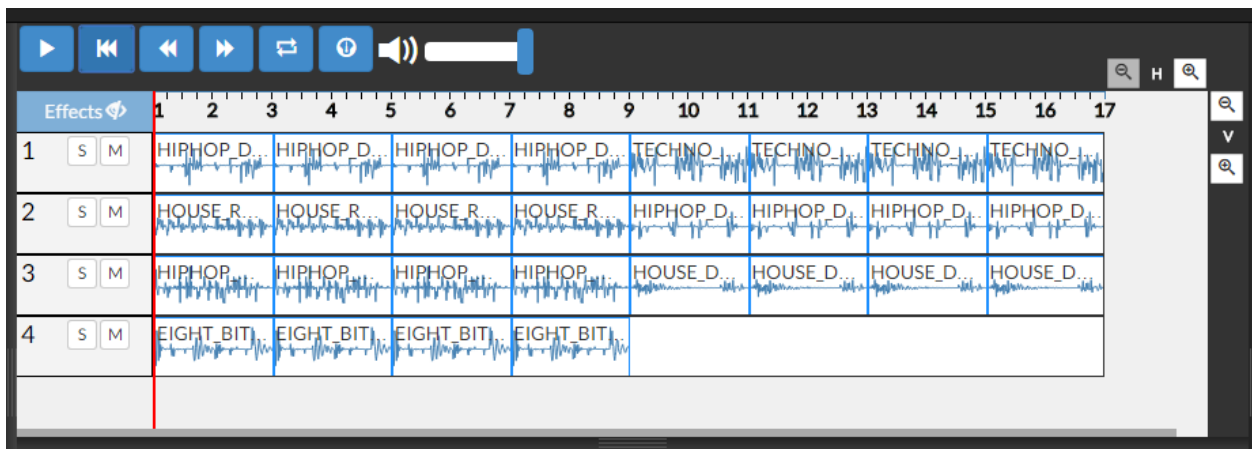
```
Load Script▼ More options...▼ Font Size▼
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18 drums2 = EIGHT_BIT_ANALOG_DRUM_LOOP_002
19
20 ensemble1 = [drums, rhodes, guitar, drums2]
21
22 # Another ensemble
23 synth = TECHNO_CLUB_ANALOGLEAD_005
24 drums3 = HIPHOP_DUSTYGROOVE_005
25 chords = HOUSE_DEEP_DREAMPAD_001
26
27 ensemble2 = [synth, drums3, chords]
28
29 def ensembleCollect(cliplist, start, end):
30     # Watch the power of the for loop
31     for clip in range(len(cliplist)):
32         track = clip + 1
Run Script
```

18. To use this list we will call the ensembleCollect() function another time with a new range of measures (9 to 17)

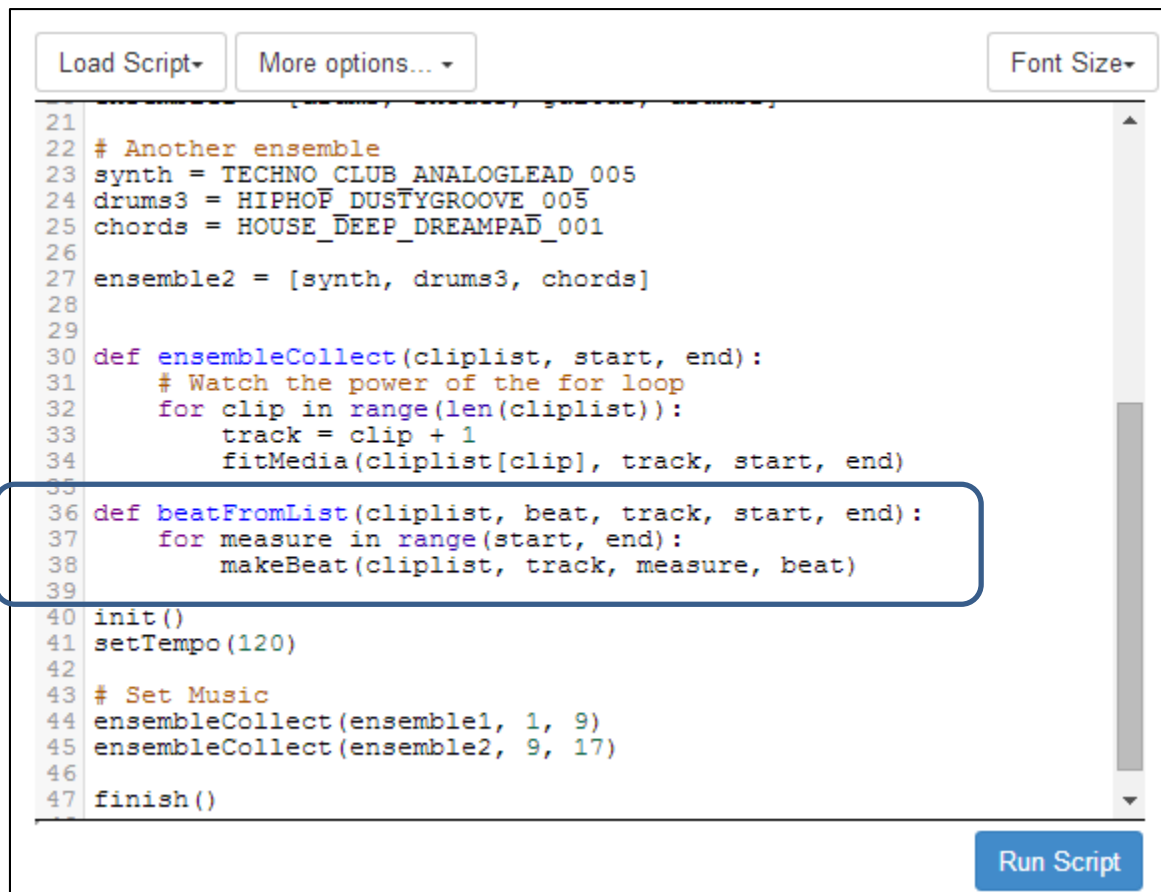
```
Load Script ▾ More options... ▾ Font Size ▾
22 # Another ensemble
23 synth = TECHNO_CLUB_ANALOGLEAD_005
24 drums3 = HIPHOP_DUSTYGROOVE_005
25 chords = HOUSE_DEEP_DREAMPAD_001
26
27 ensemble2 = [synth, drums3, chords]
28
29 def ensembleCollect(cliplist, start, end):
30     # Watch the power of the for loop
31     for clip in range(len(cliplist)):
32         track = clip + 1
33         fitMedia(cliplist[clip], track, start, end)
34
35
36 init()
37 setTempo(120)
38
39 # Set Music
40 ensembleCollect(ensemble1, 1, 9)
41 ensembleCollect(ensemble2, 9, 17) ←
42
43 finish()
```

Run Script

19. Run the script and note the new section added from 9 to 17 with the clips from ensemble2



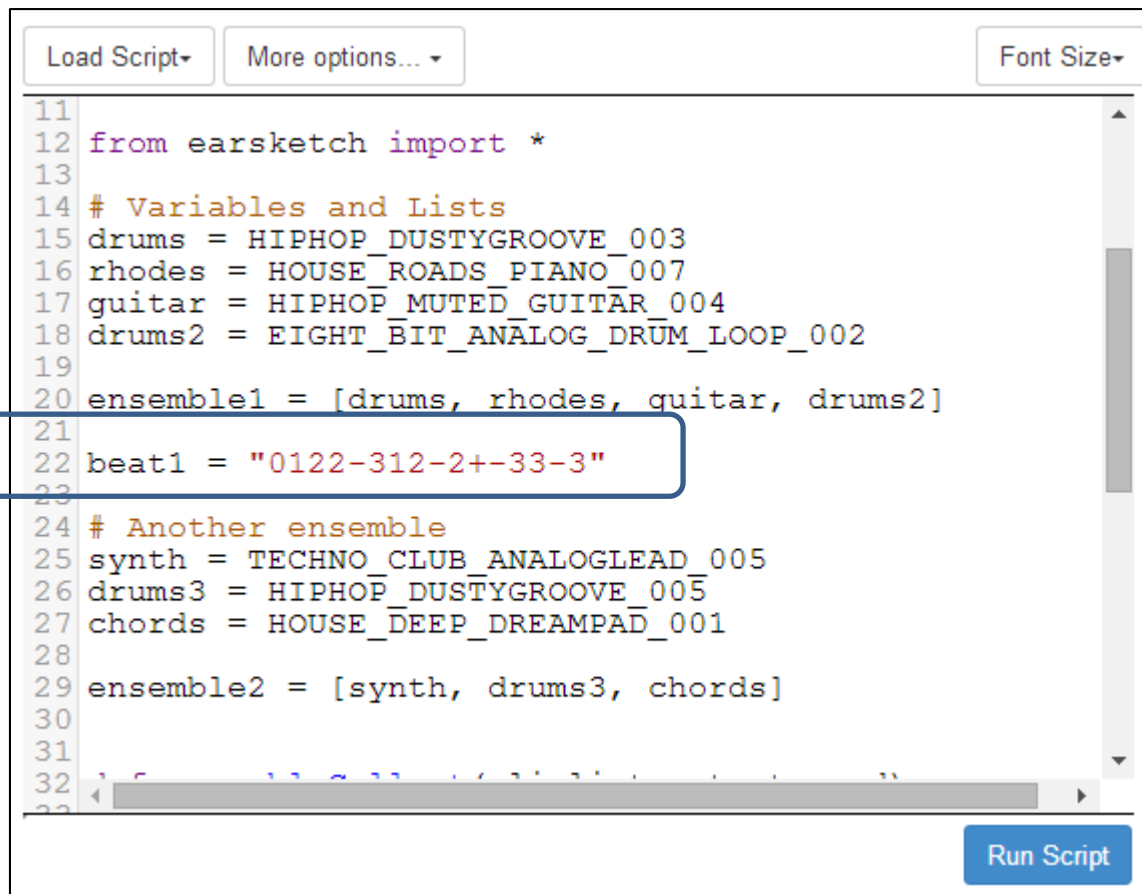
20. We can also use lists with the makeBeat() function. First, define a function that takes a list, beat, start, and end as parameters:



```
21
22 # Another ensemble
23 synth = TECHNO CLUB ANALOGLEAD 005
24 drums3 = HIPHOP DUSTYGROOVE 005
25 chords = HOUSE DEEP DREAMPAD 001
26
27 ensemble2 = [synth, drums3, chords]
28
29
30 def ensembleCollect(cliplist, start, end):
31     # Watch the power of the for loop
32     for clip in range(len(cliplist)):
33         track = clip + 1
34         fitMedia(cliplist[clip], track, start, end)
35
36 def beatFromList(cliplist, beat, track, start, end):
37     for measure in range(start, end):
38         makeBeat(cliplist, track, measure, beat)
39
40 init()
41 setTempo(120)
42
43 # Set Music
44 ensembleCollect(ensemble1, 1, 9)
45 ensembleCollect(ensemble2, 9, 17)
46
47 finish()
```

Run Script

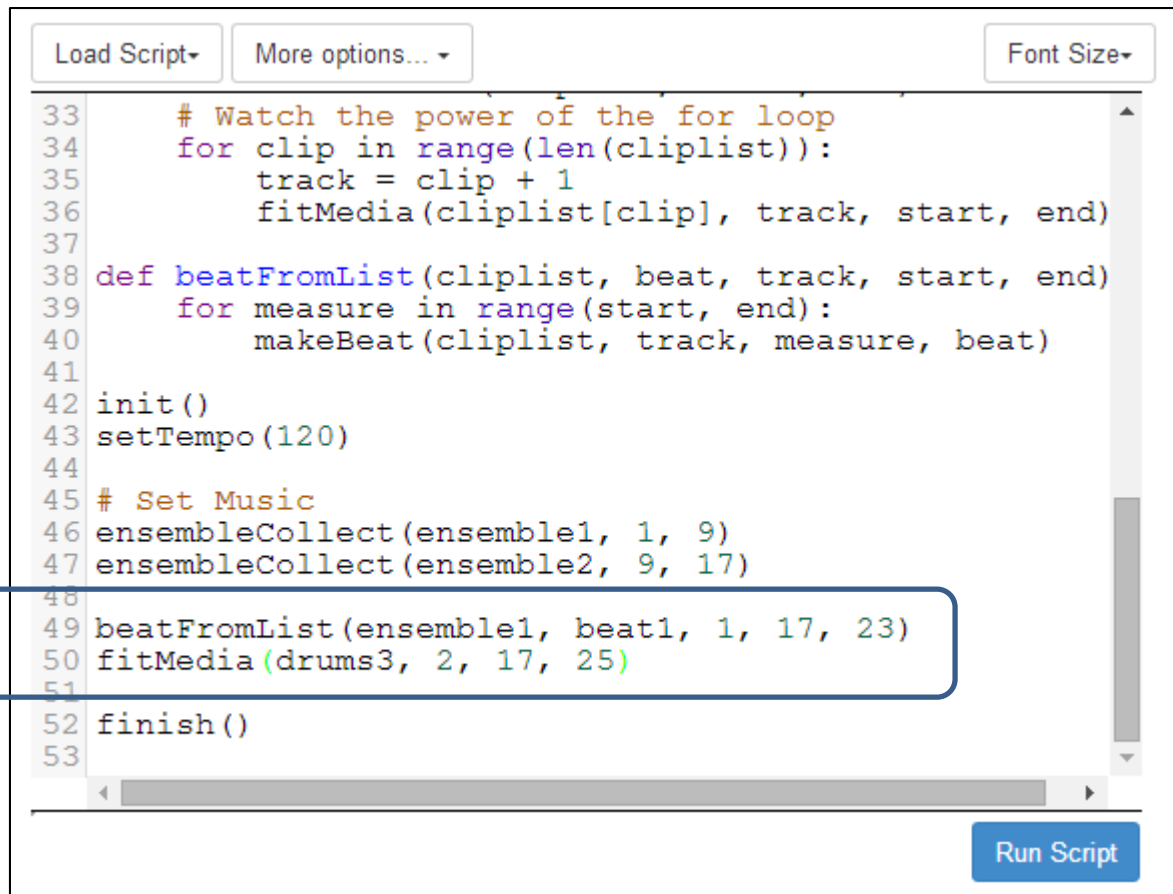
21. We will now define a beat for ensemble1. We can use a combination of numbers from 0 to the length of the list-1. (0 to 3 in this example). We can also use + and -.



```
11
12 from earsketch import *
13
14 # Variables and Lists
15 drums = HIPHOP_DUSTYGROOVE_003
16 rhodes = HOUSE_ROADS_PIANO_007
17 guitar = HIPHOP_MUTED_GUITAR_004
18 drums2 = EIGHT_BIT_ANALOG_DRUM_LOOP_002
19
20 ensemble1 = [drums, rhodes, guitar, drums2]
21
22 beat1 = "0122-312-2+-33-3"
23
24 # Another ensemble
25 synth = TECHNO_CLUB_ANALOGLEAD_005
26 drums3 = HIPHOP_DUSTYGROOVE_005
27 chords = HOUSE_DEEP_DREAMPAD_001
28
29 ensemble2 = [synth, drums3, chords]
30
31
32
```

Run Script

22. Now we will call the `beatFromList()` function and add another track on measures 17 to 25.



```
33     # Watch the power of the for loop
34     for clip in range(len(cliplist)):
35         track = clip + 1
36         fitMedia(cliplist[clip], track, start, end)
37
38 def beatFromList(cliplist, beat, track, start, end)
39     for measure in range(start, end):
40         makeBeat(cliplist, track, measure, beat)
41
42 init()
43 setTempo(120)
44
45 # Set Music
46 ensembleCollect(ensemble1, 1, 9)
47 ensembleCollect(ensemble2, 9, 17)
48
49 beatFromList(ensemble1, beat1, 1, 17, 23)
50 fitMedia(drums3, 2, 17, 25)
51
52 finish()
53
```

Buttons: Load Script, More options..., Font Size, Run Script

23. Run the Script and note measures 17 to 25.

24. To permanently save your work, make sure to select 'More Options -> Save Script to Cloud'. You will select 'Save' when prompted.

Assignment:

1. Complete the Directions as presented.
2. Add an additional list `ensemble3` with at least 4 clips. Use the `ensembleCollect()` function to place this new list from measures 25 to 33.
3. Create a new beat string for any of your lists. Use the `beatFromList()` function and additional `fitMedia()` calls to place music from 33 to 41.