

ITP 115– Programming in Python

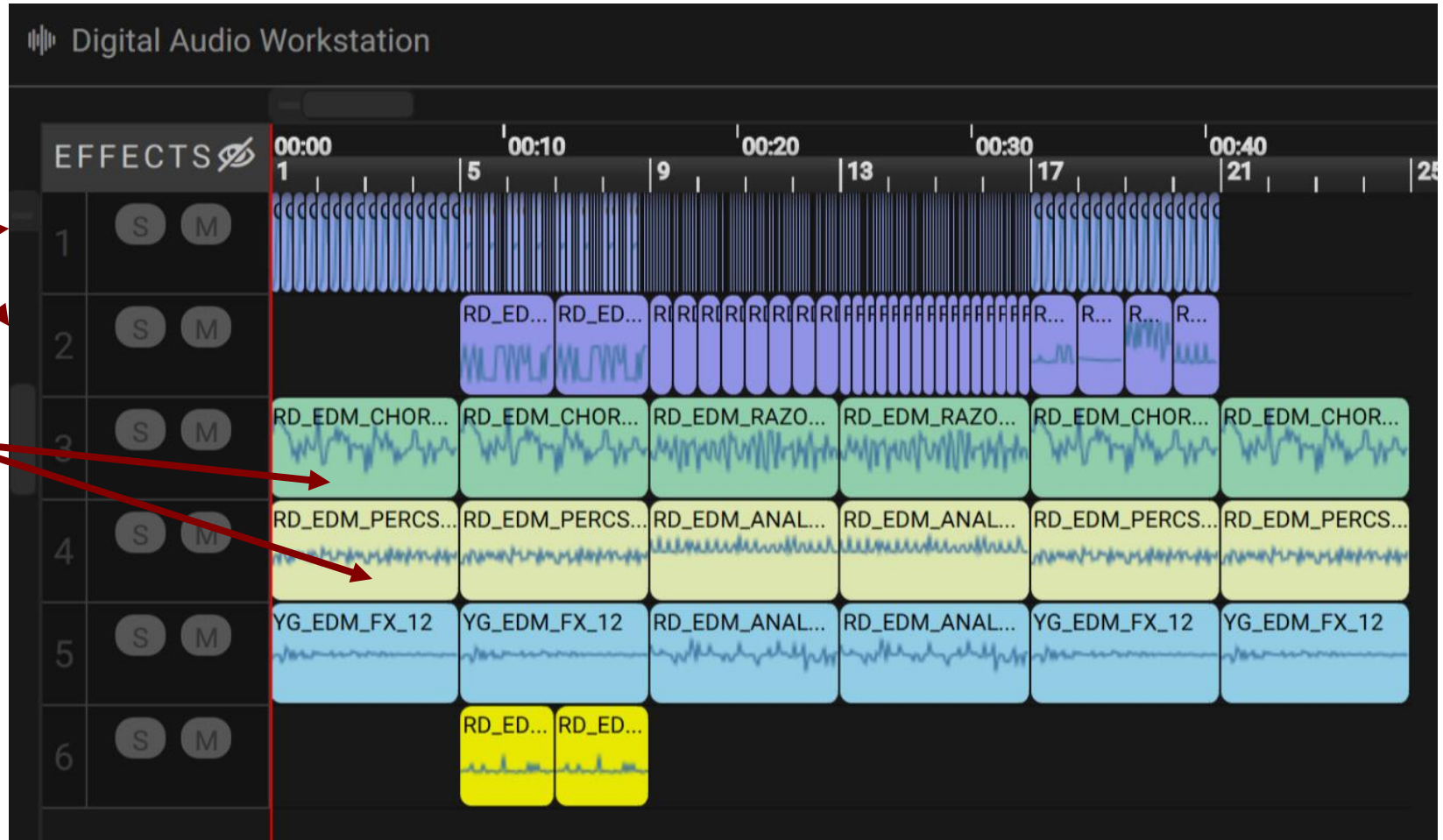
EarSketch

EarSketch

- Learn to code while making music!
- One request
 - You will have time to explore EarSketch shortly
 - When I'm explaining it, please don't play the audio since it is distracting

Background

- EarSketch is a Digital Audio Workstation (DAW)
 - Pro Tools, Logic, and GarageBand are other examples
 - DAWs are like PyCharm (IDE) for making music
- In a DAW, different sounds and instruments are arranged in distinct **tracks** that all play together

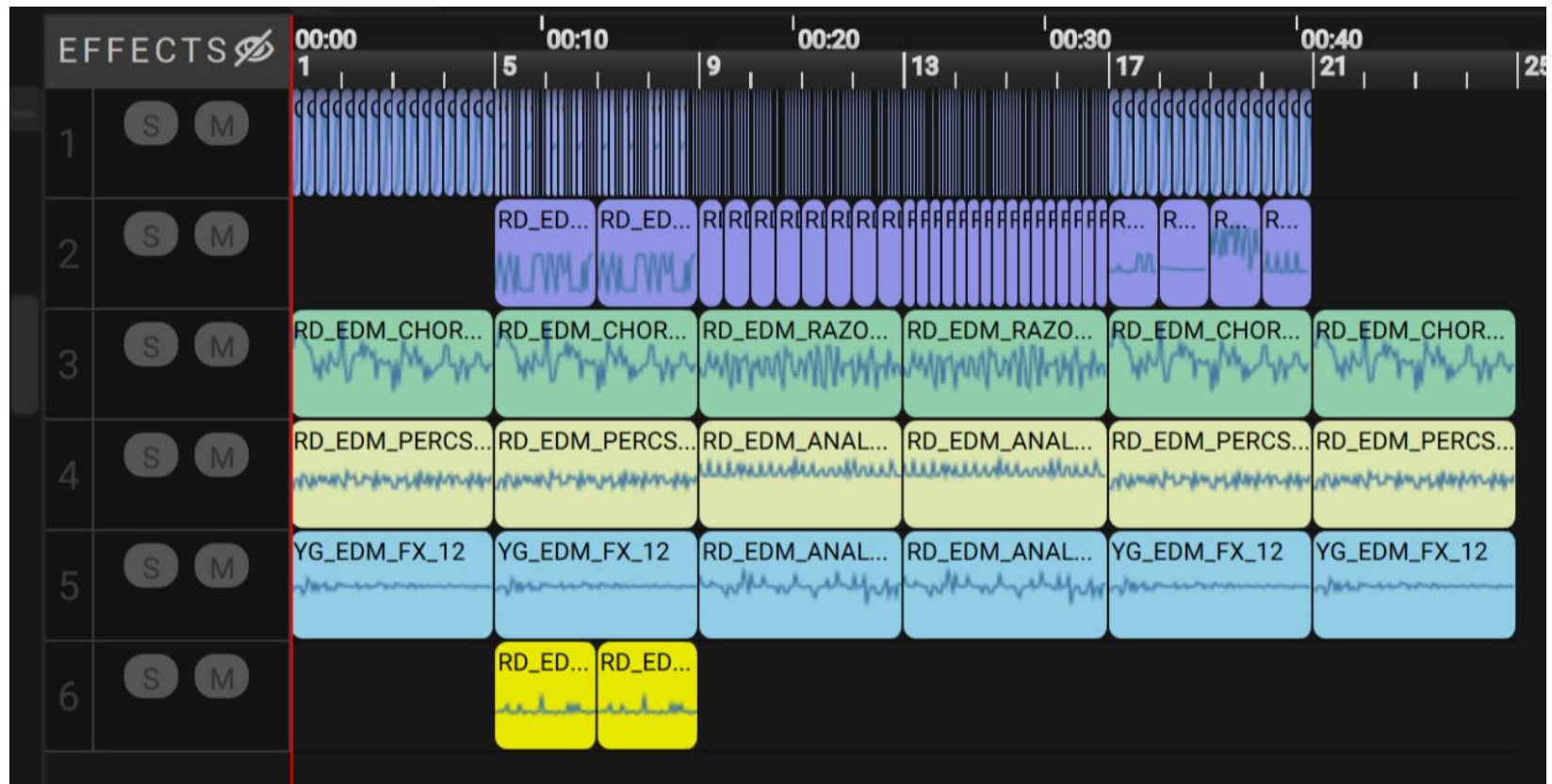


Background

- A **beat** is the basic unit of time in music
- **Tempo** is the speed of a song in beats per minute (bpm)
 - Higher tempos mean faster songs
- Beats are grouped into **measures** which all have the same number of beats.
 - In EarSketch, **one measure has four beats**

Measure 1

Measure 9

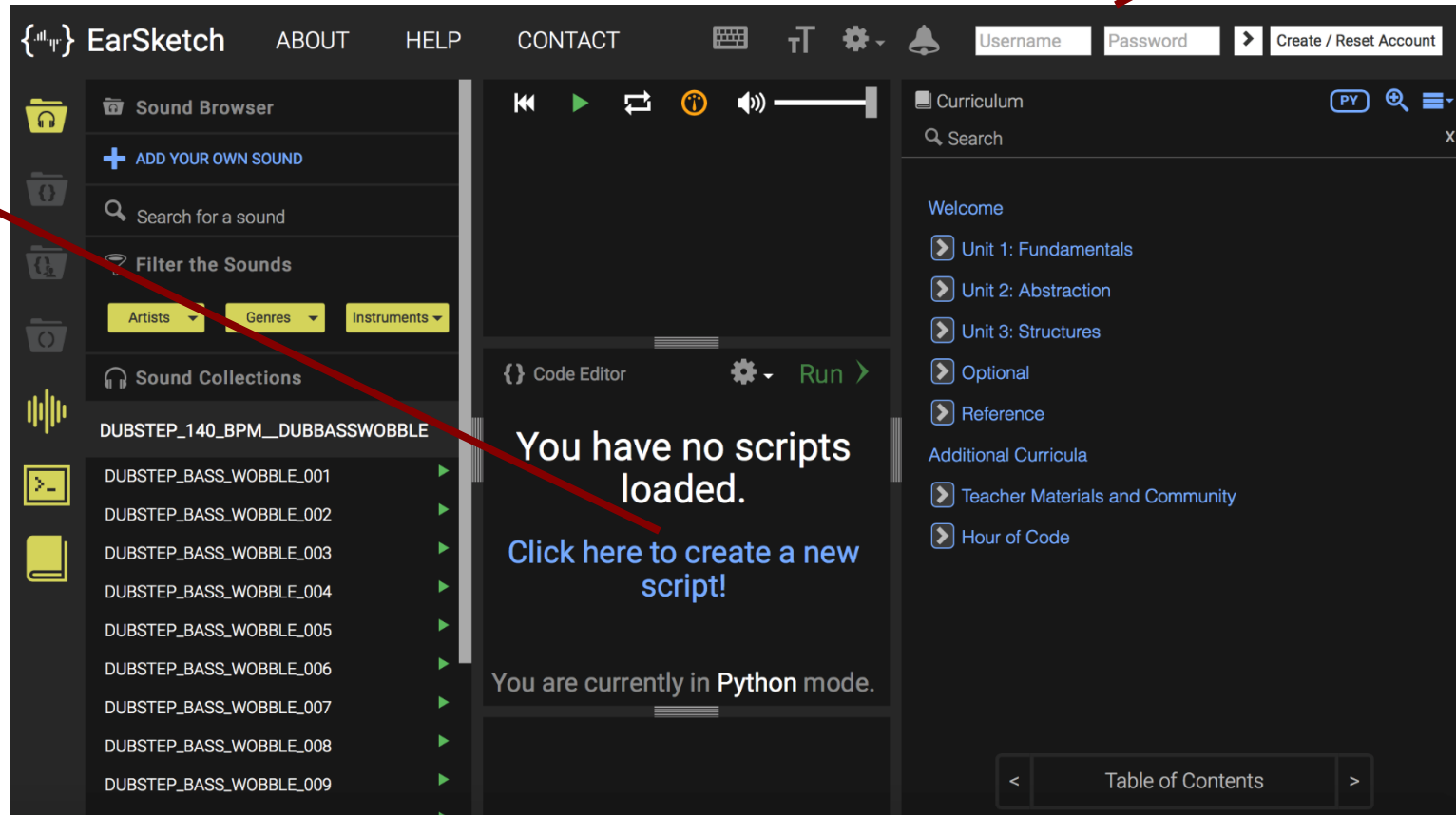


EarSketch Environment

- <https://ears sketch.gatech.edu/landing/#/>
- Click “Get Started”

EarSketch Environment

Create an account



Create a song

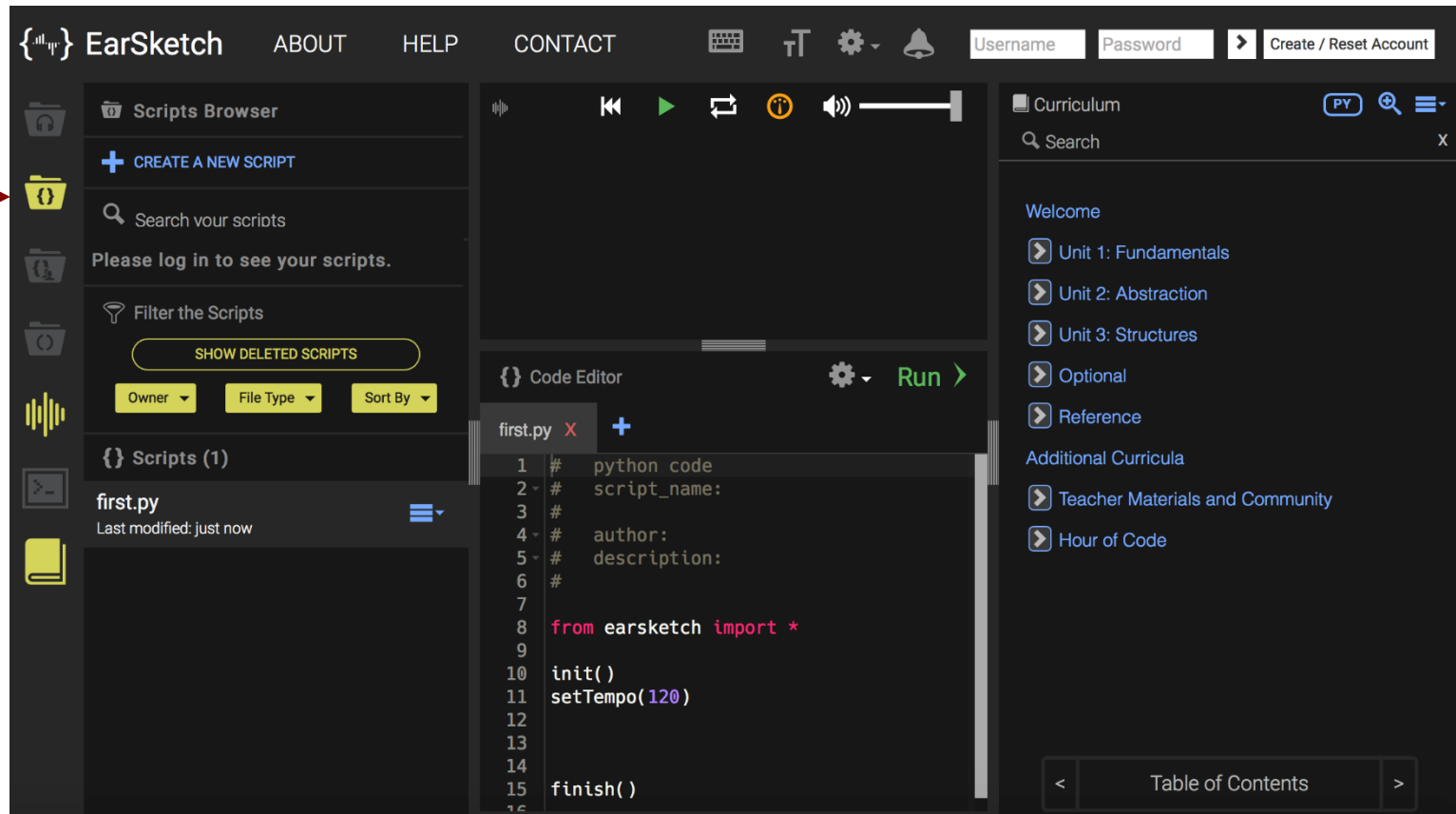
EarSketch Environment

Sounds →
Audio for
songs

The screenshot displays the EarSketch web application interface. The top navigation bar includes the EarSketch logo, links for ABOUT, HELP, and CONTACT, a keyboard icon, a text-to-speech icon, a settings gear, a bell notification icon, and a login section with fields for Username and Password, and a 'Create / Reset Account' button. The main interface is divided into three primary sections. On the left is the 'Sound Browser', which features a search bar, filter buttons for Artists, Genres, and Instruments, and a list of sound collections. The 'DUBSTEP_140_BPM_DUBBASSWOBBLE' collection is expanded, showing a list of individual sound files (e.g., DUBSTEP_BASS_WOBBLE_001) with play and download icons. A red arrow points from the 'Sounds' text to the Sound Browser icon. The center section contains a 'Code Editor' with a 'Run' button and a file explorer showing 'first.py'. The code in 'first.py' includes comments for script name, author, and description, followed by imports from the earsketch library and function calls to init(), setTempo(120), and finish(). The right section displays the 'Curriculum' with a search bar and a list of units: Unit 1: Fundamentals, Unit 2: Abstraction, Unit 3: Structures, Optional, Reference, Teacher Materials and Community, and Hour of Code. A 'Table of Contents' button is located at the bottom of the curriculum panel.

EarSketch Environment

Scripts →
Different
"songs" you
create



EarSketch Environment

Share →

The screenshot displays the EarSketch web application interface. At the top, there is a navigation bar with the EarSketch logo, links for ABOUT, HELP, and CONTACT, and a user login section with fields for Username and Password, and a 'Create / Reset Account' button. Below the navigation bar, the interface is divided into three main sections. On the left is the 'Shared-Scripts Browser' which includes a search bar, filter options for Owner, File Type, and Sort By, and a list of shared scripts. In the center is the 'Code Editor' with a file named 'first.py' open, showing Python code for initializing and running a script. On the right is the 'Curriculum' sidebar with a search bar and a list of units: Unit 1: Fundamentals, Unit 2: Abstraction, Unit 3: Structures, Optional, Reference, Teacher Materials and Community, and Hour of Code. A 'Table of Contents' button is at the bottom of the curriculum sidebar.

Shared-Scripts Browser

Search scripts shared with you

Filter the Shared Scripts

Owner File Type Sort By

Earsketch allows you to share your scripts with other users of EarSketch. When you open a share-link shared by someone, the script associated with the link appears in this space till you import it into your workspace or delete it. For more information on how to share scripts you can [click here](#). Please login to see the scripts shared with you.

Code Editor

```
1 # python code
2 # script_name:
3 #
4 # author:
5 # description:
6 #
7
8 from earsketch import *
9
10 init()
11 setTempo(120)
12
13
14
15 finish()
16
```

Curriculum

Search

Welcome

- Unit 1: Fundamentals
- Unit 2: Abstraction
- Unit 3: Structures
- Optional
- Reference

Additional Curricula

- Teacher Materials and Community
- Hour of Code

Table of Contents

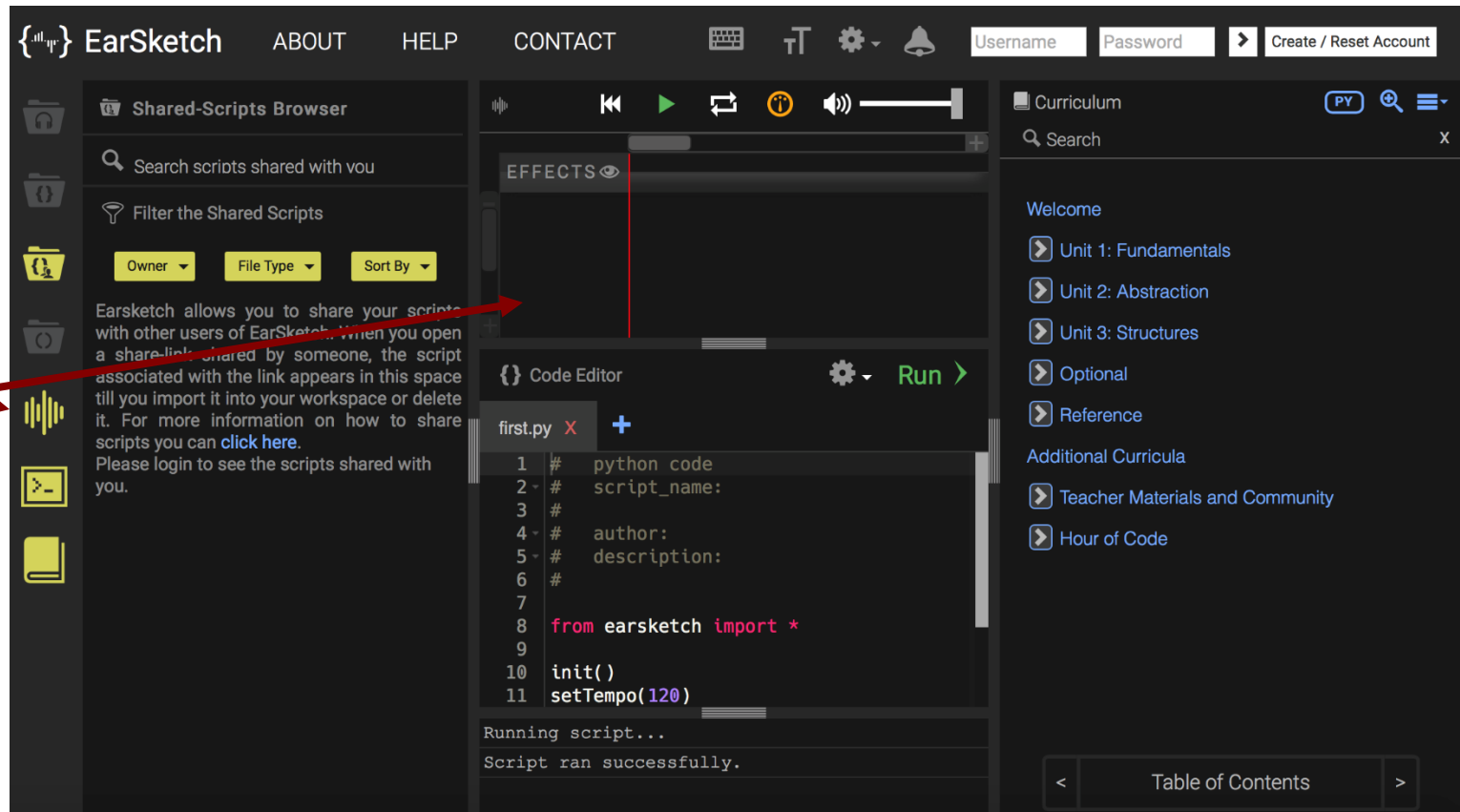
EarSketch Environment

API →
How to use
different
functions

The screenshot displays the EarSketch web application interface. The top navigation bar includes the EarSketch logo, links for ABOUT, HELP, and CONTACT, a username and password login field, and a 'Create / Reset Account' button. The left sidebar features an 'API Browser' with a search bar and a list of functions: `analyze`, `analyzeForTime`, `analyzeTrack`, `analyzeTrackForTime`, `dur`, and `finish`. The main workspace is divided into three sections: a top audio player with playback controls and a waveform, a central 'Code Editor' showing a Python script named `first.py`, and a right-hand 'Curriculum' panel. The Python script in the editor includes comments for script name, author, and description, followed by the code `from earsketch import *`, `init()`, and `setTempo(120)`. Below the code editor, a status message reads 'Running script... Script ran successfully.' The curriculum panel on the right lists units: 'Unit 1: Fundamentals', 'Unit 2: Abstraction', 'Unit 3: Structures', 'Optional', and 'Reference', along with 'Additional Curricula' like 'Teacher Materials and Community' and 'Hour of Code'. A 'Table of Contents' button is located at the bottom of the curriculum panel.

EarSketch Environment

Workstation
The music you
create



EarSketch Environment

The screenshot displays the EarSketch web application interface. At the top, there is a navigation bar with the EarSketch logo, links for ABOUT, HELP, and CONTACT, and a user login section with fields for Username and Password, and a 'Create / Reset Account' button. The main interface is divided into three primary sections:

- Shared-Scripts Browser (Left):** Contains a search bar for scripts shared with the user, filter buttons for Owner, File Type, and Sort By, and a text block explaining the sharing process. Below this is a sidebar with icons for various features, including a terminal icon.
- Code Editor (Center):** Displays a Python script named 'first.py' with the following code:

```
1 # python code
2 # script_name:
3 #
4 # author:
5 # description:
6 #
7
8 from earsketch import *
9
10 init()
11 setTempo(120)
```

Below the code editor, a status bar indicates 'Running script...' and 'Script ran successfully.'.
- Curriculum (Right):** A sidebar titled 'Curriculum' with a search bar and a list of units: Unit 1: Fundamentals, Unit 2: Abstraction, Unit 3: Structures, Optional, and Reference. It also includes 'Additional Curricula' such as Teacher Materials and Community, and Hour of Code. A 'Table of Contents' button is at the bottom.

Red arrows point from the text 'Console Python code output' to the terminal icon in the left sidebar and the status bar in the code editor.

Console
Python
code output

Let's Get Started

- Click to create a new script
 - Make sure you select Python as your language
- Each script has 3 sections:
 1. Setup
 - `init()`
 - `setTempo(int BPM)`
 2. Music you define
 3. Finish
 - `finish()`

Let's Get Started

The screenshot displays the EarSketch web application interface. At the top, there is a navigation bar with links for 'ABOUT', 'HELP', and 'CONTACT', along with icons for keyboard shortcuts, text-to-speech, settings, and a notification bell. On the right side of the navigation bar are input fields for 'Username' and 'Password', and a 'Create / Reset Account' button.

The main interface is divided into three sections:

- API Browser (Left):** A sidebar containing a search bar and a list of available APIs. The APIs listed include:
 - analyze** (audioFile, featureForAnalysis)
 - analyzeForTime** (audioFile, featureForAnalysis, startTime, endTime)
 - analyzeTrack** (trackNumber, featureForAnalysis)
 - analyzeTrackForTime** (trackNumber, featureForAnalysis, startTime, endTime)
 - dur** (fileName)
 - finish** (No Parameters)
 - fitMedia** (partially visible)
- Code Editor (Center):** A workspace for writing Python code. It shows a file named 'first.py' with the following code:

```
1
2 # Don't forget to import the library!
3 from earsketch import *
4
5 # 1. Setup Section
6 init()
7 setTempo(120)
8
9 # 2. Music Section
10 # Put everything here!
11
12
13
14 # 3. Finish Section
15 finish()
16
```
- Output Console (Bottom):** A terminal window showing the execution status of the script. It displays the messages: 'Running script...' and 'Script ran successfully.'

Adding Sounds

- Add sounds by calling **fitMedia()** function, which takes 4 arguments:
 - **fileName**- the name of the sound to be played
 - **trackNumber**- where the sound goes
 - **startLocation**- measure to start sound
 - **endLocation**- measure to end sound

Adding Sounds

- Try it!
 - Note: you will have to click “Run” then the play button in the Workstation

- Demo

Adding Sounds

- We added a sound, but it was short
- How can we extend and repeat sounds in the background?

```
12 #Setup
13 from earsketch import *
14 init()
15 setTempo(120)
16
17 #Music
18 for i in range(3):
19     fitMedia(TECHNO_LOOP_PART_002, 1, i+1, i+2)
20
```

- Demo

Effects

- You can add effects to alter a sound
- Some common ones are **VOLUME**, **DELAY**, and **FILTER**
 - A full list of effects can be found in Chapter 30.1 of the EarSketch guide guide
- The syntax for adding an effect is very similar to adding a sound

Adding Effects

- To add an effect, call **setEffect()** function, which takes 4 arguments
 - **trackNumber**- the number to apply the effect to
 - **effectName**- the effect you want to add
 - **effectParameter**- how you want to change effect
 - **effectValue**- the value for the parameter
- Note: **trackNumber** must match an existing sound

Effect Example

- Let's say we wanted to lower the volume of our background beat

```
16
17 #Music
18 for i in range(3):
19     fitMedia(TECHNO_LOOP_PART_002, 1, i+1, i+2)
20
21     setEffect(1, VOLUME, GAIN, -5.0)
22
```


- Demo

Background

- **Four beats per measure** (aka *quarter notes*)
- Often we divide a measure further into sub-beats e.g. *sixteenth notes*



Making a Beat

- EarSketch allows us to compose our own beats note by note
- Great for drum beats
- To make a beat, though, we need a "**beat string**"

Beat Strings

- The beat string is a string of characters *"to refer to sixteenth note sub-beats of a measure"*
- The characters of use are:
 - "0" starts playing a clip
 - "-" is a rest (silence)
 - "+" extends the sounds into the next sixteenth note sub-beat

Adding a Beat

- The function to add a beat is **makeBeat()** and it takes 4 arguments:
 - **fileName**- the name of the sound to be played
 - **trackNumber**- the number to insert the file onto
 - **measureNumber**- the measure of your song you want to start the beat on
 - **beatString**

Beat Example

- Let's add a beat to our song

```
16
17 #Music
18 for i in range(3):
19     fitMedia(TECHNO_LOOP_PART_002, 1, i+1, i+2)
20
21     setEffect(1, VOLUME, GAIN, -5.0)
22
23 #Making a beat
24 beat = "0-00-00-0++0+0"
25 makeBeat(OS_CLAP01, 2, 1, beat)
26
```

- Demo

Beat Example

- Let's have it repeat over the same measures as our background track!

```
17 #Music
18 for i in range(3):
19     fitMedia(TECHNO_LOOP_PART_002, 1, i+1, i+2)
20
21     setEffect(1, VOLUME, GAIN, -5.0)
22
23 #Making a beat
24 beat = "0-00-00-0++0+0"
25 for measure in range(1, 4):
26     makeBeat(OS_CLAP01, 2, measure, beat)
27
```


So Much More!

- There are so many more features in EarSketch
 - Functions, record custom sounds, draw animations
- For more information
 - <https://earsketch.gatech.edu/earsketch2/#?curriculum=4-6-2&language=python>

Now That's What I Call Music!

