Running Python

Learning Objectives:

- Students will use the Python shell to run basic python codes.
- Students will create and execute files using Python code.

There are 2 ways to execute Python code. For most of the course, we'll be working with Python files, but it's good to know about the shell, especially for testing small snippets of code.

The Python Shell

A **shell** is simply an interface for interacting with services on our computer. Up to this point, we have been using the terminal or command prompt as a shell to interact with all the files and services on our computer.

We can turn our terminal into a shell for Python by typing `python` for **Windows** , `python3` for **Mac** in our terminal.

What is the *Python shell*? The shell is a command line interface we can use to interact with the Python interpreter. Similar to how we were able to use our web browser's console to run JavaScript, the Python shell allows us to run Python code in our terminal. Once activated, we can type some Python code and see the results immediately. Let's try it out by simply typing `python` in our terminal:

```
Adriens-MacBook-Air:~ adrienjdion$ python3
Python 3.9.2 (v3.9.2:1a79785e3e, Feb 19 2021, 09:06:10)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

When we see these `>>>` , we know we're in the shell. That means it's no longer functioning as our terminal, but will only understand Python code. When we type a statement, the shell will output the result. Try out these commands:

```
>>> 4 + 5
>>> 31/2
>>> x = 9
>>> y = "hello"
>>> x * y               # what do you expect this one to do??
```

To exit the shell, type: `exit()` .

Python Files

We just learned how to experiment with Python code in the shell, but as soon as we close the shell, we've lost all of our hard work. Now we need a way of saving our code so we can use it later.

Navigate to the fundamentals directory in the terminal. Create a new file called hello_world.py by running `touch hello_world.py` on a Mac or GitBash or `type nul > hello_world.py` in a Windows command prompt.

Open the hello_world.py file we just created and add the following line of code and save the file:

```
print("Hello World!")
```

Now we'll run it from the terminal: `python hello_world.py`. If you are on a Mac, use `python3 hello_world.py`

The print statement tells the Python interpreter to output whatever follows into the terminal. It's a lot like `console.log()` in JavaScript. The terminal output will look like the following:

```
Adriens-MacBook-Air:desktop adrienjdion$ cd python/fundamentals/fundamentals/
Adriens-MacBook-Air:fundamentals adrienjdion$ ls
for_loop_basic_I                hello_world
functions_basic_i               hello_world.py
functions_basic_ii              recognize_python
functions_intermediate_i        rock_paper_scissors
greetings                       students_grades
Adriens-MacBook-Air:fundamentals adrienjdion$ python3 hello_world.py
Hello World
Adriens-MacBook-Air:fundamentals adrienjdion$ _
```

Let's add some variables along with print statements. Try entering the following in your file and run your file again:

```
x = "Hello Python"
print(x)
y = 42
print(y)
```