

## Part 0 Installing python:

To install python on windows, visit this link (press ctrl+click): [Windows x86-64 executable installer](#)

For all other options (Mac/Ubuntu), if it is not already installed, go to this link and select the appropriate installer from their list: <https://www.python.org/downloads/release/python-379/>

To ensure you have installed python correctly, open command prompt (or terminal if using Mac/Ubuntu) and type “python --version” and press enter. If it responds with the version number of the python you installed then everything has worked as expected.

One of the libraries we are going to use today is called numpy. To install this on your computer go to the terminal and type:

```
pip install numpy
```

Wait for all of the loading bars to complete and move to the next part of the lab.

\*\*\*\*\*

For this work sheet note the following: To save a python file just save the file with the extension “.py”. To run this file, go to the terminal/command prompt and type (replace MyFileName with the name of your file):

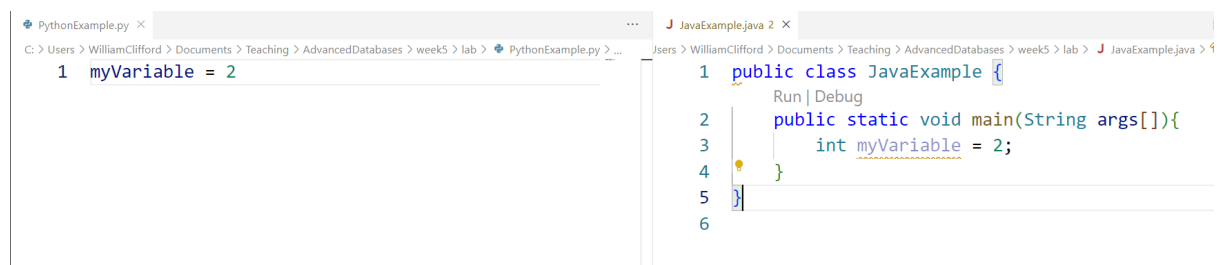
```
python MyFileName.py
```

## Part 1 Going from Java to Python:

During this module I would like to use Python for my future examples, but I am acutely aware that most of you have only ever seen Java. For this reason, I would like to begin by highlighting some of the differences between Python and Java. There are many differences, but I’ll start with the syntax and see where that takes us.

### 1.1 Declaring variables.

In Java we typically declare variables by stating the type, variable name, and the value assigning to that variable. In Python we don’t have to declare the type explicitly. Python will dynamically interpret what the type should be during run-time. We do have the option of explicitly stating types but while developing short scripts this is usually not necessary, see below:



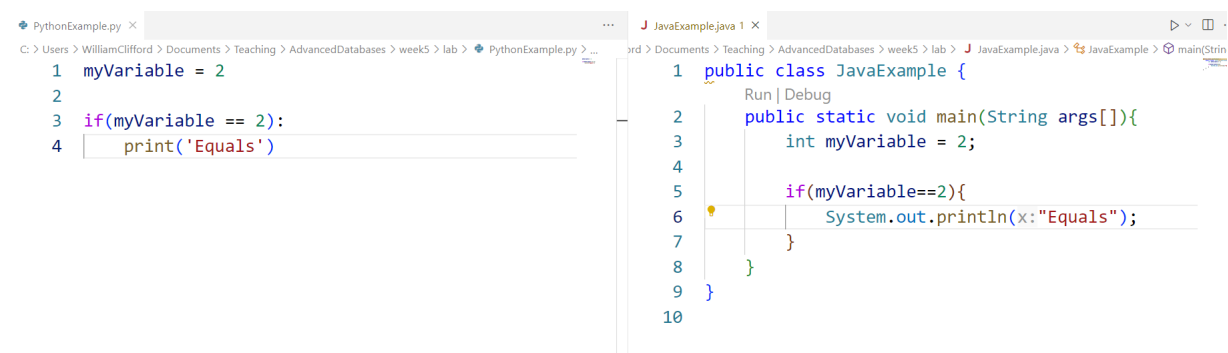
There are a few interesting things you may notice here. The python example (on the left) only requires a single statement of what the variable name is and its value. It also does not require semi-colons at the end of statements. Java however is much more verbose. It requires you to create a class and a main method. It also requires you to state variable types and use semi-colons at the end of each statement.

## 1.2 The Interpreter (just a nod):

Python uses an interpreter, it does not go through the same stages of compilation before running, as Java does. Python is run (or interpreted) one line at a time. You can have the correct syntax for every line up until where an error is present, and all that previous code will still run. This is both a blessing and a curse. In Java, all your code must be compiled before you can run it. Sometimes this means mistakes can be caught earlier (+1 for Java), but you can often go longer periods without any code running which can be quite disheartening to beginner programmers (+1 Python). Later, we will see how the interpreter can be used to enable quick piloting of code (writing short code snippets in the terminal) and making short documents with room for explanations and code side-by-side (Jupyter Notebooks). More on this in a later lab.

## 1.3 Conditions:

The syntax for ifs and else's is very similar between Java and Python:



Some of the differences: python does not use curly braces/brackets to control the beginning and end of condition scope. This is also true for loops, classes, and methods. Python tracks the numbers of tabs (remember that spacing does matter in python). Conditions, loops, methods, and classes also use the colon symbol to indicate where the block begins.

You may have also noticed that you are only required to write `print(...)` instead of `System.out.println(...)`;

Extra marks if you noticed that Strings don't require double quotes, as they do in Java. Although you can choose to use double quotes if you wish.

### 1.3 Loops:

Loops in python revolve around iterating through a list. By using the range() method you can declare a list of numbers you wish to iterate through. Using the format below i will be assigned a value from this list of numbers in their correct order after each iteration. Using the range() method you can alter what increments the values will increase (or decrease) by for each item in the list. I know how much you like W3Schools so here's a link on the use of range here:

[https://www.w3schools.com/python/ref\\_func\\_range.asp](https://www.w3schools.com/python/ref_func_range.asp)

```
9 |  
10 for i in range(0,10):  
11     print(i)
```

```
9 |  
10 for(int i=0; i<10; i++){  
11     System.out.println(i);  
    }
```

You can also use while loops like their use in Java. Why don't you try to guess how a while loop might look based on what you have learned so far.

### 1.3 Classes and methods:

I've shown you quite a bit in python so far. How about I get you to work on a task. For this part I want you to create a Coffee class in Python. Similar to something you would have done at the end of your software development module in Java. Make some attributes for the coffee, like espressoMeasurement, creamRatio, syrupOption etc. and getters and setters for each. Finally make a computeIngredients() method that prints all of the coffee parameters/attributes to the screen.

## Part 2 Data Management:

Using Python, I would like you to construct an array of size 100 with random integers.

In python it looks like this:

```
import numpy as np  
# 100 random ints between 0-1000  
myRandoms = np.random.randint(1000, size=100)
```

If you search the arrays now, you'll have to use linear search to find specific values. Write a loop to find the value and report back the number of times your loop had to iterate to find the number (use the 76<sup>th</sup> value in the array as your search query to guarantee that you find a value).

Now sort the array. This is how it looks in python:

```
import numpy as np  
# 100 random ints between 0-1000  
myRandoms = np.random.randint(1000, size=100)  
sorted_randoms = np.sort(myRandoms)
```

Finally implement binary search as described in class (see lecture notes for details). Then report how many iterations it took to find a value in the array (again use the 76<sup>th</sup> value in the array to guarantee that the query is within the array).