Getting Started

This lesson will walk you through some basic commands and operations in Python.

WE'LL COVER THE FOLLOWING

- Native Python for science
  - Importing functionalities
  - Lists in Python
  - Functions in Python

Native Python for science

Let's see what Python looks like! I hope you know the basics of Python, but this lesson will hopefully be a good, quick refresher. The first thing you will notice is that Python makes use of whitespace and does not use a use semicolon(;) like some other languages. Here is a very simple example:

```
x = 5
y = 10
print(x+y)
```

Importing functionalities

We will make use of many libraries, some that are pre-installed with Python and some we will have to install ourselves. To get a library use an import statement:

```
from collections import Counter
```

This command imports the class `Counter` from the `collections` library.

`Counter` is a very useful tool for data scientists; it can count the number of times items appear in collections such as lists. For example, in the code below we will create a list of marriage ages. Using `Counter` we can quickly count the number of times each unique age appears.

Lists in Python

Lists are a useful data structure to store data. They will be studied in more detail during the next lesson. For example:

```
from collections import Counter
marriage_ages = [22, 22, 25, 25, 30, 24, 26, 24, 35]  # create a list
value_counts = Counter(marriage_ages)  # apply the counter functionality
print(value_counts.most_common())
```

You can see that we created a list containing marriage ages using the **[]** at **line 2**. We then fed that list into the `Counter` function at **line 4** to print out the most common values as a list of tuples, at **line 5**.

A **tuple** is a collection inside the (). These tuples contain *two* elements: the value and then the number of times that value appeared in your list. The frequency orders the list of tuples. The value with the most occurrence appears first.

Functions in Python

Functions are also useful. Functions in Python start with the keyword `def` and the *function name* followed by the *inputs* the function expects within brackets. Here is a function that takes in 2 inputs, `x` and `y`, and returns the `sum`:

```
def add_two_numbers(x, y):  # function header
    """
    Takes in two numbers and returns the sum
    parameters
        x : str
            first number
        y : str
            second number
    returns
        x+y
    """
    z = x + y
    return z  # function return
print(add_two_numbers(100,5))  # function  call
```

Functions can also be *anonymous*, meaning that you don’t have to declare them with the above structure. Instead, you can use the `lambda` keyword. Here is the same function as above, but as an anonymous function:

```
y = lambda x, y: x + y  # an anonymous function which takes x and y and input and returns x+y
print(y(100,5))  # call the function
```

Now that you have seen the basics of Python, in the next section we will introduce the most common data structures you will need to know for data processing.