Introduction to Python

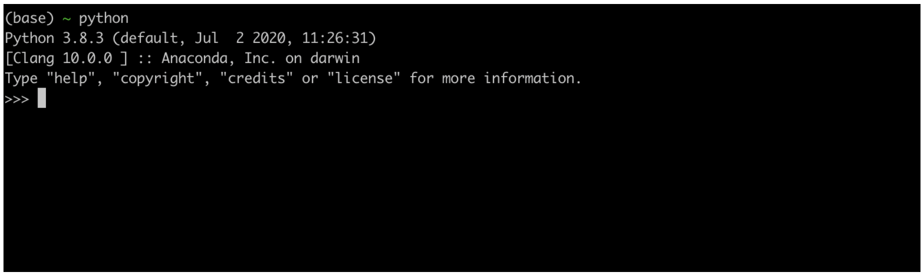
The are several ways to run a python script. That is true for other programming languages as well. One way is to use the Python interpreter.

Using the Python interpreter

In the command line type:

```
$ python
```

This will start a prompt that looks something like:



```
(base) ~ python
Python 3.8.3 (default, Jul  2 2020, 11:26:31)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

The `>>>` indicates point at which you can type python code.

Type `2 + 2` and press enter. You can see what this looks like below:

```
>>> 2 + 2
4
```

Creating numeric variables

We can assign variables to values using the `=` operator:

```
>>> the_meaning_of_life = 42
>>> the_meaning_of_life = the_meaning_of_life + 2
>>> the_meaning_of_life
44
```

Creating boolean variables

We can create boolean values using a number of comparison operators which include:

- `==` equals
- `!=` not equals
- `>` strictly greater
- `>=` greater than or equal

```
>>> is_42 = the_meaning_of_life == 42
>>> is_42
False
>>> greater_than_42 = the_meaning_of_life > 42
>>> greater_than_42
True
```

Creating list variables

Python has an indexable structure called lists:

```
>>> numbers = [1, 2, 4, 5]
>>> max(numbers)
5
>>> min(numbers)
1
>>> sum(numbers)
12
>>> numbers[0]
1
>>> numbers[-2]
4
>>> numbers.append(50)
>>> numbers
[1, 2, 4, 5, 50]
```

To close the python interpreter type:

```
$ exit()
```

Using Python scripts

Another way to run Python is using a script and the command line. Python scripts are used when writing more sophisticated code (software).

Open the `addition.py` file (we created before) using the editor you downloaded (for example VS code) for this workshop.

Edit the `addition.py` so that it looks like this:

```
print(1 + 1)
```

and save.

While making sure you are in the `rsd-workshop`, which you can check using the command:

```
$ cd
```

or

```
$ pwd
```

use the following command to run the python script `addition.py`:

```
$ python addition.py
```

If statements

Let's create another Python file called `if-statements.py`. This can be done either by using the `echo`/`touch` command or from your editor `File > New file`.

Include the following code in the file:

```
N = 572
if N % 2 == 0:
    print("N is even")
else:
    print("N is odd")
```

and then run it (type):

```
$ python if-statements.py
```

Note white space and indentation is important in python. The indented code block indicate what code to execute if the boolean variable `N % 2 == 0` is True.

While loops

It is possible to repeat code using while loops which will repeatedly check a boolean variable.

Create a file called `while-loops.py`, include the following code and run it.

```
N = 0
even_number_count = 0
while N < 10:
    if N % 2 == 0:
        even_number_count = even_number_count + 1
    N += 1
print(even_number_count)
```

Functions

It is possible to create functions in Python.

Open the file `addition.py` and we are going to implement a function which adds two input numbers.

Your `addition.py` should look like:

```
def add_two_numbers(a, b):
    return a + b

print(add_two_numbers(1, 3))
```

save and run:

```
$ python addition.py
```