

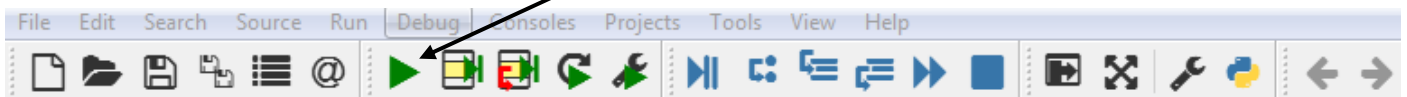
Before we begin we want to add a quick comment to the top of your script just saying what it is. When a script runs Python will look at the lines from top to bottom, if we want Python to ignore a line (i.e. it is in human speak) we can add `#` to the start and it will appear in grey. So lets all add a quick comment to the top of our code now:

```
1 #Lets Learn Python
```

We want to begin by simply adding two numbers together, Python already knows what numbers and simple mathematical functions are so lets simply try to add 3 and 4 together. To do this we will write a very simple line of code:

```
1 #Lets Learn Python
2
3 3+4
```

Now lets hit the run button to execute the code:



And..... nothing happens, that's because although Python understood what we told it, we didn't tell it what do with the result. To actually see the result of our code we need to tell it to 'print' the result. So lets try that now.

```
5 #if we want to see the result we need to print it
6
7 print(3+4)
```

Hit run again and you should see 7 pop up in the lower right hand window.

Now we can also print text, however we need to tell Python we are writing in human speak, we do this with 'quote marks'. Lets add some more to our code:

```
9 #Printing text needs 'quote marks'
10
11 print('abc')
```

You should now see 'abc' or whatever you wrote in the quote marks show up in that lower right window.

One of the great things about Python (or other languages) is that we can assign variables, we might want to do this if we have a long number we are going to use often and don't want to write it out every time. So lets assign a variable:

```
13 #assign a variable
14
15 a=1.602**2
```

Note that $(1.602)^2$ is written `1.602**2` where as $(1.602) \times 2$ is written `1.602*2`

We can now do some simple addition with our variable as before:

```
16
17 print(a+3)
18
```

Now run the code and see the result.

We don't have to have sensible names for our variables, so lets make another new variable with a more creative name, add it to our original variable and print the result.

```
19 #can have creative names
20
21 newvariablewithmorecreativename=4
22
23 print(a+newvariablewithmorecreativename)
```

Another type of variable is an array, often it is sensible to think of these as vectors. Lets make an array now and print it:

```
25 #Arrays
26
27 A=[1,2,3,4,5]
28
29 print(A)
```

So our array is a collection of numbers, it may have been useful to group these together, but sometimes we may want to use only one or some of the numbers. We can access individual numbers using Python's indexing system. We do this by calling our array A, followed by the index of the entry we want in [square brackets]. So lets call entry [1] and print it.

```
31 #indexing arrays, printing entry in position [1]
32
33 print(A[1])
```

You will notice that this printed '2' which is not the first entry in A, this is as in Python the indexing starts at [0].

We can call multiple values in A, to do this we write the A[2:5], this tells Python we are interested in the values from entries [2] to entry [5]. Lets print this:

```
35 #printing multiple values in A
36
37 print(A[2:5])
```

You should have seen three values show up in that bottom right hand window.

Now these are some other things we want to do with arrays. What if we didn't know how long A was but wanted to find out. Easy, we can use the len() function. We will also want to use this value later, so lets make it a variable, then print it:

```
49 #what if we don't know how long A is?
50
51 n=len(A)
52
53 print(n)
```

Great so now we know how long A is.... So lets do something with that, but first we will quickly learn about for loops. When Python is reading through your script and sees a for loop is called it will not just continue reading. It will execute the loop a given amount of time. In this case we want to execute the loop for each value in our array, so n times.

For loops will run through an index, we will call this index i, we also need to tell it where to start and stop, for this we use the range() function. So for our first loop lets simply print the value of the index each time we run the loop:

```
55 #simple for loop
56
57 for i in range(n):
58     print(i)
```

You should see some numbers pop up in that window.

Now lets do something slightly more useful. Lets add our first variable a to each value in A and print the result:

```
61
62 for i in range(n):
63     print(A[i]+a)
64
```

For loops are generally for longer sections of code than we have been using here, a more compact way to use them is something called a list comprehension. This is simply a for loop formatted into one line. Lets use a list comprehension to create a new array B, then print it:

```
65 #list comprehension (for loop in one line of code)
66
67 B=[A[i]+3 for i in range(n)]
68
69 print(B)
```

Now we have two arrays A and B, so lets plot them against each other. To do this we need to load something we need to import something called a module. Modules are sort of like plugins or addons, they are required as Python is huge, if everything was loaded at the beginning it would take all day, so we need to specify which modules we want to load. We are interested in one called MATPLOTLIB.

Now even MATPLOTLIB is huge, and all we want to do is plot some things, so we are going to only import a small part of MATPLOTLIB called PYPLOT.

To do this we write the following:

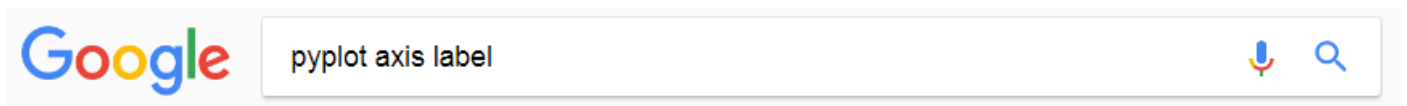
```
71 #Import Pyplot from Matplotlib
72
73 from matplotlib import pyplot as plt
74
```

We write the final bit 'as plt' so we can call this module by simply writing plt instead of matplotlib.pyplot. So lets use it to plot A vs B. To do this we simply write:

```
74
75 plt.plot(A,B)
76
```

If you run the code you should see a plot pop up.

Now we want to add some labels to the axes. One of the skills we are trying to teach in this tutorial is now how to follow instruction but how to formulate and answer questions. So Now you need to try to answer this yourselves. How can I add a label to the y-axis? A good place to begin is Google:



The first result will have the answer.

Now try to work out how to change the colour of the line in the plot.