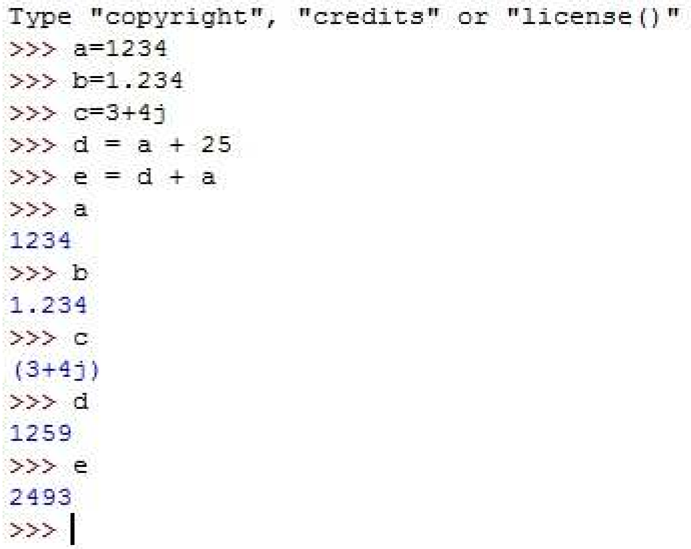
Data types

This tutorial is an introduction to python data types. We will go through the basics of integers, strings, lists, tuples and dictionaries. If you are coming from another programming language try to keep an open mind as python is a little different than the typical C/C++.

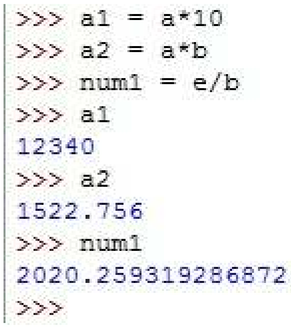
Idle is used in this tutorial because of the simplicity of the code. Once you install python (the latest version) you will see something called “Idle” in your program files in the python directory. When you run this a terminal comes up that allows you to enter code. Idle comes in very handy later when you need to view or edit code alongside of the code you are working with. You can actually right click on any python file and choose “edit with Idle” to bring up your code in an editable Idle window.

Integers:
In python data types are not defined as int, char, float, long int etc… A variable is simply initialized by setting it to a value.



```
Type "copyright", "credits" or "license()"
>>> a=1234
>>> b=1.234
>>> c=3+4j
>>> d = a + 25
>>> e = d + a
>>> a
1234
>>> b
1.234
>>> c
(3+4j)
>>> d
1259
>>> e
2493
>>> 
```

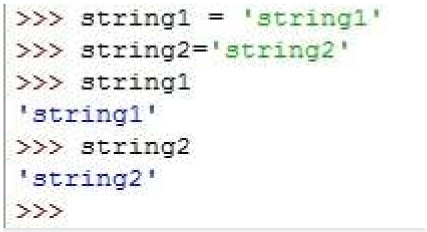
This is pretty straight forward. Variable a is set to integer 1234, b is set to the floating point number 1.234, c is an imaginary number. D and E are examples of using variables with a simple operator.



```
>>> a1 = a*10
>>> a2 = a*b
>>> num1 = e/b
>>> a1
12340
>>> a2
1522.756
>>> num1
2020.259319286872
>>>
```

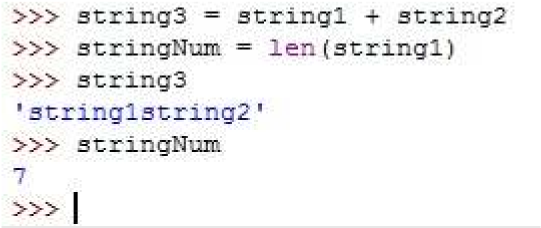
Strings:

Strings work the same way, there is no initialization. Just set the string name equal to the value. One side note here is that python allows you to use either double (“”) or single (‘’) quotes for strings. There is no difference between the two.



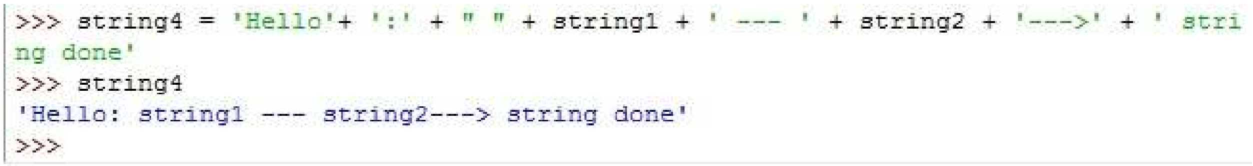
```
>>> string1 = 'string1'
>>> string2='string2'
>>> string1
'string1'
>>> string2
'string2'
>>>
```

This is all pretty simple, but when we introduce operands we begin to see python go it’s separate way from the rest of the lower level languages. Just like adding numbers together you can add strings and any other data types together.



```
>>> string3 = string1 + string2
>>> stringNum = len(string1)
>>> string3
'string1string2'
>>> stringNum
7
>>> 
```

Now, this isn’t quite as intuitive. Strings can be added together to form larger strings. The idea here is that a string is actually an array of letters indexed from 0 to the end of the string. When you add two strings the second string is just appended to the end of the first string. This is actually easier to conceptualize if you think of the ‘+’ sign as concatenation rather than mathematical addition. With string you can concatenate many different segments of text to form a single string.



```
>>> string4 = 'Hello'+ ':' + " " + string1 + ' --- ' + string2 + '--->' + ' stri
ng done'
>>> string4
'Hello: string1 --- string2---> string done'
>>>
```

There are also many different built in functions that python provides. These take time to find and become accustomed to using, but after some time of being aware of them they become very useful. The one we used above, len(), calculates the length of a list, tuple or dictionary and returns a value. The value returned is the total displacement from 0 with 0 being the first letter in our string.