Syntax:

* **Line to Line**
  + A Python program is divided into a number of logical lines and every logical line is terminated by the token NEWLINE. A logical line is created from one or more physical lines.

Example:

>>x = 1

y = 2

Python will define x first then y, by the order of lines.

* **1Comment**
  + Commenting on code is very important as it explains our code both to us and to others who are trying to understand our code. In Python, we use # in front of our comment.

Example:

>> #Define x

x = 1

* **Join lines**
  + We can separate a one line code into multiple lines code by joining them with \

Example:

>> pr\

int\

(“Hel\

lo”)

* **Multiple statement on one line**
  + We can also do the other way around, code everything in one line by using ;

**Example:**

>> print(“Hello”); print(“world”); print(“How are you”)

Variable

* **Must begin with a letter (a - z, A - B) or underscore (\_)**
* **Other characters can be letters, numbers or \_**

**Example:**

>>> Item\_name = "Computer" #A String

>>> Item\_qty = 10 #An Integer

>>> Item\_value = 1000.23 #A floating point

>>> print(Item\_name)

Computer

>>> print(Item\_qty)

10

>>> print(Item\_value)

1000.23

>>>

* **Case Sensitive**

>> X = 1

>> print(x) will give us an error since what we defined is capital X rather than a lower-case x.

* **Can be any (reasonable) length**
  + YES! >> Sdkjaniofaofnasodasodnasionfoiasnfoia =1 works!
* **There are some reserved words which you cannot use as a variable name because Python uses them for other things.**
  + and, in, continue, etc.

**Examples:**

>>> a = 12

>>> 12 = a

SyntaxError: can't assign to literal

>>>

**Multiple Assignments:**

>>> x = y = z = 1

>>> print(x)

1

>>> print(y)

1

>>> print(z)

1

>>>

>>> x,y,z = 1,2,"abcd"

>>> print(x)

1

>>> print(y)

2

>>> print(z)

abcd

Reuse variable name:

>>> x = 100

>>> print(x)

100

>>> x = "Python"

>>> print(x)

Python

>>>

**Swap variables:**

>>> x = 10

>>> y = 20

>>> print(x)

10

>>> print(y)

20

>>> x, y = y, x

>>> print(x)

20

>>> print(y)

10

>>>

Data Type

* **use type() to check data type**

**Example:**

>> a = 1; b = 1.1; c = True;

>> type(a)

>> type(b)

>> type(c)

* **string**
  + start and end with “” or ‘’

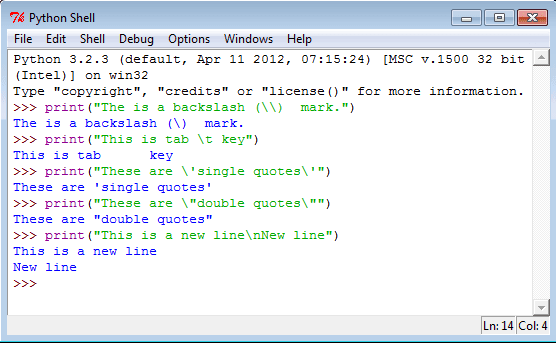
**Example:**

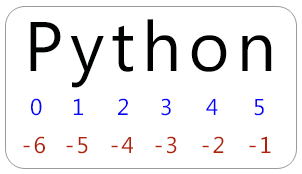
>> str = “Hello world!”

>> print(str)

>> Hello world

**Question**: how to print “” or ‘’ or tab in a sentence?

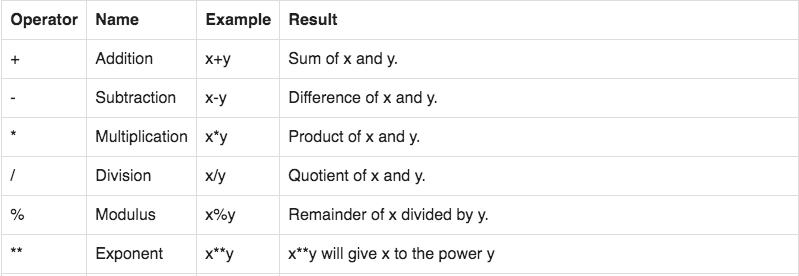


* **String indices**
  + >> str = “Python Tutorial”
  + use len() to check the length of the string
  + >> print(str[0]) # print first character
  + >>>>print(str[-15]) # print first character
  + >>print(str[14]) #print last character
  + >>print(str[-1]) #print last character
  + >>print(str[16]) #out of range
* **Immutable**
  + Str[1] = ‘A’ will result an error
* **Combine numbers and string**
  + Does not work:
    - >>num = 10
    - >> print(“we want to add”+ num+ “to our string”)
  + work
    - >>num = 10
    - >>print(“we want to add” + str(num)+ “to our string”)

Operators

**Examples:**

Basic:

>>> x = 2

>>> y = 3

>>> z = 5

>>> x \* y

6

>>> x + y

5

>>> x \* y + z

11

>>> (x + y) \* z

25

>>> 2\*\*8

256

>>> 5/2

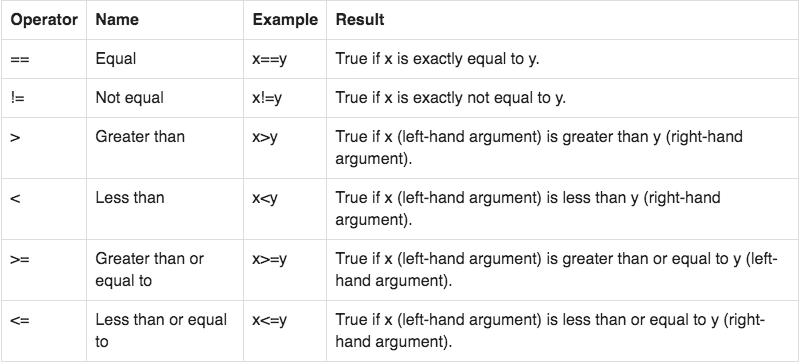
2.5

Modulo

>>> 10%7

3

>>> -10%7

4

**Comparison**

>>> 2 == 3

False

>>> 3 == 3

True

>>> 3 == '3'

False

>>> 2 < 3

True

>>> "a" < "aa"

True

EXERCISE:

1. Write a program that asks the user to type in a string, and then tells the user how long that string was.
2. /Users/Rick/Desktop/Screen Shot 2018-09-07 at 4.28.36 PM.pngAsk the user for a string, and then for a number. Print out that string, that many times. (For example, if the string is hello and the number is 3 you should print out hellohellohello.)
3. /Users/Rick/Desktop/Screen Shot 2018-09-07 at 4.28.41 PM.pngUse Python to calculate
4. Use Python to calculate
5. Use Python to calculate 11111111111111111111+22222222222222222222, but in one line of code with at most 15 characters. (Hint: each of those numbers is 20 digits long, so you have to find some other way to input those numbers)