Python Operator

This notebook outlines the basic use of operators in Python. I have decided to now upload my files i used in learning.

Arithmetic operator

In [1]:
```
#Addition Operator
7 + 8
```

Out[1]: 15

In [2]:
```
#Subtraction operator
10 - 6
```

Out[2]: 4

In [5]:
```
#Integer Division Operator
4 // 3
```

Out[5]: 1

In [6]:
```
# Float Division Operator
4 / 3
```

Out[6]: 1.3333333333333333

In [7]:
```
#Multiplcation Operator
3 * 2
```

Out[7]: 6

In [8]:
```
# Wrong use of addition operators with strings
"franlklyn" + 4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-8-661f8e80a933> in <module>()
      1 # Wrong use of operators with strings
----> 2 "franlklyn" + 4

TypeError: can only concatenate str (not "int") to str
```

In [9]:
```
#You can instead use the + operators to concatenate strings
"Franklyn" + "4"
```

Out[9]: 'Franklyn4'

In [10]:
```
# The multiplication operator used with a string will multiply the occurence o
f such string
"Franklyn" * 3
```

Out[10]: 'FranklynFranklynFranklyn'

Comparison Operators. These operators only return boolean values (i.e True/False)

In [13]:
```
# Greater than operator > #
60 > 50
```

Out[13]: True

In [14]:
```
# Lesser than operator < #
60 < 50
```

Out[14]: False

In [15]:
```
#Using a comparison operator along with an Arithemtic operator
2 * 4 < 5 * 6
```

Out[15]: True

In [16]:
```
#The equal to operator. Take note to use a double equal sign ==
40 == 40
```

Out[16]: True

In [17]:
```
33 == 43
```

Out[17]: False

#The greater than or equal to operator

In [21]:
```
4 >= 3
```

Out[21]: True

In [22]:
```
4>= 4
```

Out[22]: True

In [23]:
```
4 >=5
```

Out[23]: False

The lesser than or equal to operator

In [24]:
```
4 <= 3
```

Out[24]: False

In [25]:
```
4<= 4
```

Out[25]: True

In [26]:
```
4 <=5
```

Out[26]: True

Not equal to !=

In [27]:
```
4 != 5
```

Out[27]: True

In [28]:
```
4 != 4
```

Out[28]: False

Logical Operators

the And operator. This checks if the first operand is 0 and returns it else it moves to the next operand and check the value

In [29]:
```
#This return 0 because the first operand is 0
0 and 7
```

Out[29]: 0

In [30]:
```
#This return 0 because it checks the first operand which is not 0 and move to
 consider the next operand
7 and 0
```

Out[30]: 0

In [31]:
```
#Here it returns the last operand since the first operand is not 0
8 and 9
```

Out[31]: 9

The OR operator. It considers both operands and returns the true operand

In [32]:
```
#Here the second operand 5 is returned because the first operand is false
0 or 5
```

Out[32]: 5

In [34]:
```
#Here the operator returns 5 because the first operand is true.
5 or 0
```

Out[34]: 5

In [35]:
```
#Here the operator returns 5 because the first operand is true already.
5 or 6
```

Out[35]: 5