#Python Operator

In [1]:
```
# ---Arithmetic Operator
```

In [2]:
```
x1 , y1 = 10 , 5
```

In [3]:
```
x1
y1
```

Out[3]: 5

In [4]:
```
print(x1)
print(y1)
```

```
10
5
```

In [5]:
```
x1 + y1   # addition
```

Out[5]: 15

In [6]:
```
x1 - y1    #subtraction
```

Out[6]: 5

In [7]:
```
x1 * y1     #multiplication
```

Out[7]: 50

In [8]:
```
x1 / y1     #division
```

Out[8]: 2.0

In [9]:
```
x1 ** y1    #exponential
```

Out[9]: 100000

In [10]:
```
x1 // y1    #integer division
```

Out[10]: 2

In [11]:
```
x1 % y1  # Modulus
```

Out[11]: 0

In [12]:
```
x1 + (x1 * y1) + y1 -(x1 / y1)
```

Out[12]: 63.0

In [13]:
```
#  ------Assignment operator
```

In [14]:
```
x = 4
```

In [15]:
```
x = x + 2   #if you want to increment by 2
```

In [16]:
```
x
```

Out[16]: 6

In [17]:
```
x += 2
x
```

Out[17]: 8

In [18]:
```
x *= 3
x
```

Out[18]: 24

In [19]:
```
x -= 5
x
```

Out[19]: 19

In [20]:
```
x /= 4
x
```

Out[20]: 4.75

In [21]:
```
x //= 4
x
```

Out[21]: 1.0

In [22]:
```
a , b , c = 5 , 6 , 7   #you can assigned variable in one line
```

In [23]:
```
a
```

Out[23]: 5

In [24]:
```
b
```

Out[24]: 6

In [25]:
```
c
```

Out[25]: 7

In [26]:
```
# ------Unary opertor
```

In [27]:
```
n = 8
n
```

Out[27]: 8

In [28]:
```
m = -(n)  #negation
m
```

Out[28]: -8

In [29]:
```
n
```

Out[29]: 8

In [30]:
```
-n
```

Out[30]: -8

In [31]:
```
# -------Relational operator
```

In [32]:
```
a = 7
b = 8
```

In [33]:
```
a < b
```

Out[33]: True

In [34]:
```
a > b
```

Out[34]: False

In [35]:
```
a == b     # a=b we cannot use = operator  that means it is assigning
```

Out[35]: False

In [36]:
```
a != b
```

Out[36]: True

In [37]:
```
b = 7
```

In [38]:
```
a == b
```

Out[38]: True

In [39]:
```
a >= b
```

Out[39]: True

In [40]:
```
a <= b
```

Out[40]: True

In [41]:
```
a < b
```

Out[41]: False

In [42]:
```
# -----Logical Operator
```

In [43]:
```
a = 8
b = 3
```

In [44]:
```
a < 9 and b < 8   # refers to the truth table
```

Out[44]: True

In [45]:
```
a < 8 and b < 2
```

Out[45]: False

In [46]:
```
a < 9 or b < 2
```

Out[46]: True

In [47]:
```
a > 8 or b < 2
```

Out[47]: False

In [48]:
```
x = False
x
```

Out[48]: False

In [49]:
```
not x   # you can reverse the operation
```

Out[49]: True

#Bitwise Operator

In [50]:
```
# ---Number system conversion (bit - binary digit)
```

In [51]:
```
25
```

Out[51]: 25

In [57]:
```
bin(25)     # bin() for binary
```

Out[57]: '0b11001'

In [53]:
```
0b11001
```

Out[53]: 25

In [54]:
```
0b1100
```

Out[54]: 12

In [55]:
```
int(0b1100)
```

Out[55]: 12

In [56]:
```
bin(35)
```

Out[56]: '0b100011'

In [58]:
```
oct(25)     # oct() for octal
```

Out[58]: '0o31'

In [59]:
```
0o143
```

Out[59]: 99

In [60]:
```
0o17
```

Out[60]: 15

In [61]:
```
hex(9)     # hex() for hexadecimal
```

Out[61]: '0x9'

In [62]:
```
0xf
```

Out[62]: 15

In [63]:
```
0xad
```

Out[63]: 173

In [64]:
```
hex(25)
```

Out[64]: '0x19'

In [65]:
```
# ----6 bitwise operator
# *complement(~) ,AND(&) ,OR(|) ,XOR(^) ,left shift(<<) ,right shift(>>)
```

In [66]:
```
print(bin(12))
print(bin(35))
```

```
0b1100
0b100011
```

In [67]:
```
~12      #complement
```

Out[67]: -13

In [68]:
```
~35
```

Out[68]: -36

In [69]:
```
~-30
```

Out[69]: 29

In [70]:
```
~-1
```

Out[70]: 0

In [71]:
```
print(bin(12))
print(bin(13))
```

```
0b1100
0b1101
```

In [72]:
```
12 & 13   #AND
```

Out[72]: 12

In [73]:
```
1 & 1
```

Out[73]: 1

In [74]:
```
1 & 0
```

Out[74]: 0

In [75]:
```
35 & 40
```

Out[75]: 32

In [76]:
```
12 | 13  # OR
```

Out[76]: 13

In [77]:
```
1 | 1
```

Out[77]: 1

In [78]:
```
1 | 0
```

Out[78]: 1

In [79]:
```
35 | 40
```

Out[79]: 43

In [80]:
```
12 ^ 13  #XOR.
```

Out[80]: 1

In [81]:
```
1 ^ 1
```

Out[81]: 0

In [82]:
```
0 ^ 0
```

Out[82]: 0

In [83]:
```
1 ^ 0
```

Out[83]: 1

In [84]:
```
23 ^ 76
```

Out[84]: 91

In [85]:
```
print(bin(20))
print(bin(10))
```

```
0b10100
0b1010
```

In [86]:
```
20 << 4    #left shift   means 4bits are gaining
```

Out[86]: 320

In [87]:
```
10 << 3
```

Out[87]: 80

In [88]:
```
10 >> 2   #right shift means 2 bits losing
```

Out[88]: 2

In [89]:
```
20 >> 3
```

Out[89]: 2

#Swap variable in python

In [99]:
```
a = 4
b = 7
```

In [100…
```
a = b
b = a
```

In [101…
```
a , b = b , a
```

In [102…
```
print(a)
print(b)
```

```
7
7
```

In [103…
```
# in above scenario we lost value 4
a1 = 6
b1 = 8
```

In [109…
```
temp = a1        # swap variable using third variable
a1 = b1
b1 = temp
```

In [110…
```
print(a1)
print(b1)
```

```
8
6
```

In [111…
```
a2 = 4
b2 = 2
```

In [112…
```
a2 = a2 + b2      #without using third variable
b2 = a2 - b2
a2 = a2 - b2
```

In [114…
```
print(a2)
print(b2)
```

```
2
4
```

In [115…
```
a3 = 9
b3 = 5
```

In [117…
```
a3 = a3 ^ b3      # using bitwise operator
b3 = a3 ^ b3
a3 = a3 ^ b3
```

In [118…
```
print(a3)
print(b3)
```

```
5
9
```

In [ ]: