Arithmetic Operator

```
In [1]: x1,y1=10,5
```

```
In [3]: x1+y1
Out[3]: 15
```

```
In [4]: x1-y1
Out[4]: 5
```

```
In [5]: x1*y1
Out[5]: 50
```

```
In [6]: x1/y1
Out[6]: 2.0
```

```
In [7]: x1//y1
Out[7]: 2
```

```
In [8]: x1%y1
Out[8]: 0
```

```
In [10]: x1**y1
Out[10]: 100000
```

```
In [11]: x2=3
         y2=3
         x2**y2
Out[11]: 27
```

Assignment Operator

```
In [12]: x=2
```

```
In [13]: x=x+2
```

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

```
In [15]: x+=2
```

```
In [16]: x
Out[16]: 6
```

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

```
In [18]: x
```

```
Out[18]: 8
```

```
In [19]: x*=2
```

```
In [20]: x
```

```
Out[20]: 16
```

```
In [21]: x-=2
```

```
In [22]: x
```

```
Out[22]: 14
```

```
In [23]: x/=2
```

```
In [24]: x
```

```
Out[24]: 7.0
```

```
In [25]: x//=2
```

```
In [26]: x
```

```
Out[26]: 3.0
```

```
In [27]: a,b=5,6
         print(a)
         print(b)
```

```
5
6
```

```
In [28]: a=5
         b=6
         print(a)
         print(b)
```

```
5
6
```

```
In [29]: a
```

```
Out[29]: 5
```

```
In [30]: b
```

```
Out[30]: 6
```

Unary operator

```
In [31]: n=7
In [32]: n

Out[32]: 7

In [33]: m=-(n)

In [34]: m

Out[34]: -7

In [35]: n

Out[35]: 7

In [36]: -n

Out[36]: -7
```

Relational operator

```
In [37]: a=5
         b=6

In [38]: a<b

Out[38]: True

In [39]: a>b

Out[39]: False

In [40]: a==b

Out[40]: False

In [41]: a!=b

Out[41]: True

In [43]: b=5

In [44]: a==b

Out[44]: True

In [45]: a

Out[45]: 5

In [46]: b

Out[46]: 5

In [47]: a>b
```

```
Out[47]: False
```

```
In [48]: a<b
```

```
Out[48]: False
```

```
In [49]: a>=b
```

```
Out[49]: True
```

```
In [50]: a<=b
```

```
Out[50]: True
```

```
In [51]: b=7
```

```
In [52]: a!=b
```

```
Out[52]: True
```

Logical operators

```
In [53]: a=5
         b=4
```

```
In [54]: a<8 and b<5
```

```
Out[54]: True
```

```
In [55]: a<8 and b<2
```

```
Out[55]: False
```

```
In [56]: a<8 or b<2
```

```
Out[56]: True
```

```
In [57]: a>8 or b<2
```

```
Out[57]: False
```

```
In [58]: x=False
         x
```

```
Out[58]: False
```

```
In [59]: not x
```

```
Out[59]: True
```

```
In [60]: x=not x
         x
```

```
Out[60]: True
```

```
In [61]: x

Out[61]: True

In [62]: not x

Out[62]: False
```

Number system conversions:

```
In [63]: 25

Out[63]: 25

In [64]: bin(25)

Out[64]: '0b11001'

In [65]: int(0b11001)

Out[65]: 25

In [66]: bin(30)

Out[66]: '0b11110'

In [67]: int(0b11110)

Out[67]: 30

In [68]: int(0b11001)

Out[68]: 25

In [69]: oct(25)

Out[69]: '0o31'

In [70]: int(0o31)

Out[70]: 25

In [71]: int(0b11110)

Out[71]: 30

In [72]: 0o31

Out[72]: 25

In [73]: 0b11001

Out[73]: 25

In [74]: int(0b11001)
```

```
Out[74]:  25

In [75]:  bin(7)

Out[75]:  '0b111'

In [76]:  oct(25)

Out[76]:  '0o31'

In [77]:  0o31

Out[77]:  25

In [78]:  int(0o31)

Out[78]:  25

In [79]:  hex(25)

Out[79]:  '0x19'

In [80]:  0x19

Out[80]:  25

In [81]:  hex(16)

Out[81]:  '0x10'

In [82]:  0xa

Out[82]:  10

In [83]:  0xb

Out[83]:  11

In [84]:  hex(1)

Out[84]:  '0x1'

In [85]:  hex(25)

Out[85]:  '0x19'

In [86]:  0x19

Out[86]:  25

In [87]:  0x15

Out[87]:  21
```

In [88]:

```
a=5
b=6
```

In [89]:

```
a=b
b=a
```

In [90]:

```
print(a)
print(b)
```

```
6
6
```

In [91]:

```
a1=7
b1=8
```

In [92]:

```
temp=a1
a1=b1
b1=temp
```

In [93]:

```
print(a1)
print(b1)
```

```
8
7
```

In [98]:

```
a2=5
b2=6
```

In [99]:

```
a2=a2+b2
b2=a2-b2
a2=a2-b2
```

In [100…

```
print(a2)
print(b2)
```

```
6
5
```

In [101…

```
0b110
```

Out[101…

```
6
```

In [102…

```
0b101
```

Out[102…

```
5
```

In [103…

```
print(0b110)
print(0b101)
```

```
6
5
```

In [104…

```
print(0b101)
print(0b110)
```

```
5
6
```

```
In [105…  print(bin(11))
          print(0b1011)

0b1011
11
```

```
In [106…  print(a2)
          print(b2)

6
5
```

```
In [107…  a2=a2^b2
          b2=a2^b2
          a2=a2^b2
```

```
In [108…  print(a2)
          print(b2)

5
6
```

```
In [109…  a2,b2

Out[109…  (5, 6)
```

```
In [110…  a2,b2=b2,a2
```

```
In [111…  print(a2)
          print(b2)

6
5
```

```
In [112…  print(bin(12))
          print(bin(13))

0b1100
0b1101
```

```
In [113…  0b1101

Out[113…  13
```

```
In [114…  0b1100

Out[114…  12
```

Complement

```
In [119…  ~12

Out[119…  -13
```

```
In [120…  ~46

Out[120…  -47
```

```
In [121…  ~54
```

Out[121…
```
-55
```

In [118…
```
~10
```

Out[118…
```
-11
```

Bit wise and, Or operator

In [146…
```
12 & 13
```

Out[146…
```
12
```

In [147…
```
12| 13
```

Out[147…
```
13
```

In [148…
```
1 & 0
```

Out[148…
```
0
```

In [149…
```
1 | 0
```

Out[149…
```
1
```

In [150…
```
bin(13)
```

Out[150…
```
'0b1101'
```

In [151…
```
print(bin(35))
print(bin(40))
```

```
0b100011
0b101000
```

In [152…
```
35 & 40
```

Out[152…
```
32
```

In [153…
```
35 | 40
```

Out[153…
```
43
```

In [154…
```
12 ^ 13
```

Out[154…
```
1
```

In [155…
```
print(bin(25))
print(bin(30))
```

```
0b11001
0b11110
```

In [156…
```
25^30
```

Out[156…
```
7
```

```
In [157…  bin(7)

Out[157…  '0b111'

In [158…  bin(25)

Out[158…  '0b11001'

In [135…  bin(30)

Out[135…  '0b11110'

In [136…  0b00111

Out[136…  7
```

BITWISE LEFTSHIFT OPERATOR

```
In [159…  bin(10)

Out[159…  '0b1010'

In [160…  10<<1

Out[160…  20

In [161…  10<<2

Out[161…  40

In [162…  bin(10)

Out[162…  '0b1010'

In [163…  10<<1

Out[163…  20

In [164…  10<<2

Out[164…  40

In [165…  10<<3

Out[165…  80

In [166…  bin(20)

Out[166…  '0b10100'

In [167…  20<<4

Out[167…  320
```

BITWISE RIGHTSHIFT OPERATOR

```
In [168…  bin(10)

Out[168…  '0b1010'

In [169…  10 >> 1

Out[169…  5

In [170…  10 >> 2

Out[170…  2

In [171…  10>>3

Out[171…  1

In [172…  bin(20)

Out[172…  '0b10100'

  In [ ]:
```