operators

Arithmetic operators

In [1]:
```
a,b=10,20
```

In [2]:
```
print(a)
print(b)
```

```
10
20
```

In [3]:
```
a+b
```

Out[3]: 30

In [4]:
```
a-b
```

Out[4]: -10

In [5]:
```
a*b
```

Out[5]: 200

In [6]:
```
a**b
```

Out[6]: 100000000000000000000

In [7]:
```
a/b
```

Out[7]: 0.5

In [8]:
```
a//b
```

Out[8]: 0

In [9]:
```
a%b
```

Out[9]: 10

In [10]:
```
a%%b
```

```
  Cell In[10], line 1
    a%%b
      ^
SyntaxError: invalid syntax
```

Assignment operator

In [15]:
```
a,y=4,8
```

In [14]:
```
y+=2
y
```

Out[14]: 10

In [16]:
```
a-=3
a
```

Out[16]: 1

In [19]:
```
y/=2
y
```

Out[19]: 0.25

In [20]:
```
y
```

Out[20]: 0.25

In [21]:
```
a
```

Out[21]: 1

In [22]:
```
a//=2
a
```

Out[22]: 0

In [23]:
```
y1=15
```

In [24]:
```
y1%=2
y1
```

Out[24]: 1

In [25]:
```
a1=2
```

In [27]:
```
a1**=3
a1
```

Out[27]: 512

In [29]:
```
a1*=2
a1
```

Out[29]: 1024

unary operator

In [30]:
```
u=2
u
```

Out[30]: 2

In [31]:
```
n=-(u)
n
```

Out[31]: -2

In [33]:
```
n=-(n)
n
```

Out[33]: -2

In [34]:
```
-u
```

Out[34]: -2

Relational Operator

In [35]:
```
a=10
b=22
```

In [36]:
```
a<b
```

Out[36]: True

In [37]:
```
b<a
```

Out[37]: False

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

Out[38]: False

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

Out[39]: True

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

Out[40]: False

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

Out[41]: True

In [42]:
```
a>=b
```

Out[42]: False

In [43]:
```
x=2
y=2
```

In [44]:
```
x>=y
```

Out[44]: True

In [45]:
```
x==y
```

Out[45]: True

In [46]:
```
x<=y
```

Out[46]: True

Logical Operator

In [47]:
```
a2=4
b2=6
```

In [48]:
```
a2<5 and b2>5
```

Out[48]: True

In [49]:
```
a2>7 and b2>6
```

Out[49]: False

In [50]:
```
a2<5 or b2<8
```

Out[50]: True

In [51]:
```
a2>6 or b2<8
```

Out[51]: True

In [52]:
```
a3=True
a3
```

Out[52]: True

In [53]:
```
not a3
```

Out[53]: False

In [54]:
```
a3
```

Out[54]: True

In [56]:
```
a3= not a3
a3
```

Out[56]: True

Number system

Bit Binary Digit

In [57]:
```
bin(20)
```

Out[57]: '0b10100'

In [58]:
```
int(0b10100)
```

Out[58]: 20

In [59]:
```
bin(27)
```

Out[59]: '0b11011'

In [60]:
```
int(0b11011)
```

Out[60]: 27

In [61]:
```
oct(20)
```

Out[61]: '0o24'

In [63]:
```
int(0o24)
```

Out[63]: 20

In [64]:
```
oct(30)
```

Out[64]: '0o36'

In [65]:
```
oct(35)
```

Out[65]: '0o43'

In [66]:
```
oct(25)
```

Out[66]: '0o31'

In [67]:
```
hex(20)
```

Out[67]: '0x14'

In [68]:
```
hex(16)
```

Out[68]: '0x10'

In [69]:
```
hex(15)
```

Out[69]: '0xf'

In [70]:
```
hex(10)
```

Out[70]: '0xa'

In [72]:
```
print(hex(1))
print(hex(2))
print(hex(4))
print(hex(5))
print(hex(8))
print(hex(9))
print(hex(10))
print(hex(11))
print(hex(12))
print(hex(14))
print(hex(16))
```

```
0x1
0x2
0x4
0x5
0x8
0x9
0xa
0xb
0xc
0xe
0x10
```

In [74]:
```
print(bin(1))
print(bin(2))
print(bin(0))
```

```
0b1
0b10
0b0
```

In [75]:
```
print(oct(8))
print(oct(7))
print(oct(6))
```

```
print(oct(5))
print(oct(4))
print(oct(1))
print(oct(0))
```

```
0o10
0o7
0o6
0o5
0o4
0o1
0o0
```

In [76]:
```
0x1101
```

Out[76]: 4353

In [77]:
```
0x19
```

Out[77]: 25

In [78]:
```
0x17
```

Out[78]: 23

In [79]:
```
0x15
```

Out[79]: 21

In [80]:
```
0x1101
```

Out[80]: 4353

In [81]:
```
0x1001
```

Out[81]: 4097

In [83]:
```
0b1101
```

Out[83]: 13

In [85]:
```
0o1110
```

Out[85]: 584

swaping of two vaiables

In [88]:
```
a=10
b=20
a,b
```

Out[88]: (10, 20)

In [89]:
```
c=a  # by using 3 variables
a=b
b=c
a,b
```

Out[89]: (20, 10)

In [90]:
```
A=a+b # by using 2 variables
B=a-b
A=a-b
a,b
```

Out[90]: (20, 10)

In [92]:
```
a,b
```

Out[92]: (10, 20)

In [93]:
```
a,b=b,a=10,20
a,b
```

Out[93]: (20, 10)

In [94]:
```
0b110
```

Out[94]: 6

In [95]:
```
bin(11)
```

Out[95]: '0b1011'

In [96]:
```
0b1011
```

Out[96]: 11

Bitwise operator

complement (~)

In [97]:
```
~3 # works with 2's complement (1'scomplement+1)
```

Out[97]: -4

In [98]:
```
~77
```

Out[98]: -78

In [99]:
```
~99
```

Out[99]: -100

In [100…
```
~0
```

Out[100… -1

In [101…
```
~-1
```

Out[101… 0

In [102…
```
~-6
```

Out[102… 5

In [103…
```
~-99
```

Out[103… 98

Bitwise &,|,^

In [104…
```
10&11
```

Out[104… 10

In [105…
```
19&11
```

Out[105… 3

In [106…
```
20&21
```

Out[106… 20

In [107…
```
2&3
```

Out[107… 2

In [108…
```
11&9
```

Out[108… 9

In [109…
```
12|13
```

Out[109… 13

In [110…
```
16|20
```

Out[110… 20

In [111…
```
20|14
```
Out[111… 30

In [112…
```
4^6
```
Out[112… 2

In [113…
```
12^13
```
Out[113… 1

Left shift

In [114…
```
10<<11
```
Out[114… 20480

In [115…
```
10<<2
```
Out[115… 40

In [116…
```
10<<3
```
Out[116… 80

In [117…
```
11<<3
```
Out[117… 88

Right shift

In [118…
```
21>>2
```
Out[118… 5

In [119…
```
12>>3
```
Out[119… 1

In [120…
```
23>>4
```
Out[120… 1

In [121…
```
25>>1
```
Out[121… 12

In [122…
```
30>>10
```

Out[122…
```
0
```

In [128…
```
30>>4
```

Out[128…
```
1
```

In [ ]:
```

```