Number system conversion(bit-binary digit)

In [2]:
```
25
```

Out[2]: 25

In [3]:
```
bin(25)
```

Out[3]: '0b11001'

In [4]:
```
int(0b11001)
```

Out[4]: 25

In [5]:
```
bin(35)
```

Out[5]: '0b100011'

In [6]:
```
int(0b100011)
```

Out[6]: 35

In [7]:
```
bin(20)
```

Out[7]: '0b10100'

In [8]:
```
int(0b10100)
```

Out[8]: 20

In [9]:
```
0b1111
```

Out[9]: 15

In [10]:
```
oct(15)
```

Out[10]: '0o17'

In [11]:
```
0o17
```

Out[11]: 15

In [12]:
```
hex(9)
```

Out[12]: '0x9'

In [13]:
```
0xf
```

Out[13]: 15

In [14]:
```
hex(10)
```

Out[14]: '0xa'

In [15]:
```
hex(25)
```

Out[15]: '0x19'

In [16]:
```
0x15
```

Out[16]: 21

Swap variable in python

In [18]:
```
a=5
b=6
```

In [19]:
```
a=b
b=a
```

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

In [21]:
```
print(a)
print(b)
```

```
6
6
```

In [22]:
```
a1=7
b1=8
```

In [23]:
```
temp=a1
a1=7
b1=temp
```

In [24]:
```
print(a1)
print(b1)
```

```
7
7
```

In [25]:
```
a2=5
b2=6
```

In [26]:
```
a2=a2+b2
b2=a2-b2
a2=a2-b2
```

In [27]:
```
print(a2)
print(b2)
```

```
6
5
```

In [28]:
```
print(0b101)
print(0b110)
```

```
5
6
```

In [29]:
```
print(bin(11))
print(0b1011)
```

```
0b1011
11
```

In [30]:
```
a2=a2^b2
b2=a2^b2
a2=a2^b2
```

In [31]:
```
print(a2)
print(b2)
```

```
5
6
```

In [32]:
```
print(a2)
print(b2)
```

```
5
6
```

In [33]:
```
a2,b2=b2,a2
```

In [34]:
```
print(a2)
print(b2)
```

```
6
5
```

Bitwise Operator

In [36]:
```
print(bin(12))
print(bin(13))
```

```
0b1100
0b1101
```

Complement

In [38]:
```
~12
```

Out[38]: -13

In [39]:
```
~45
```

Out[39]: -46

In [40]:
```
~6
```

Out[40]: -7

In [41]:
```
~-6
```

Out[41]: 5

In [42]:
```
~-1
```

Out[42]: 0

Bitwise and operator

In [44]:
```
12&13
```

Out[44]: 12

In [45]:
```
1&1
```

Out[45]: 1

In [46]:
```
1|0
```

Out[46]: 1

In [47]:
```
1&0
```

Out[47]: 0

In [48]:
```
12|13
```

Out[48]: 13

In [49]:
```
35&40
```

Out[49]: 32

In [50]:
```
35|40
```

Out[50]: 43

In [51]:
```
12^13
```

Out[51]: 1

In [52]:
```
25^30
```

Out[52]: 7

In [53]:
```
bin(25)
```

Out[53]: '0b11001'

In [54]:
```
bin(30)
```

Out[54]: '0b11110'

In [55]:
```
int(0b000111)
```

Out[55]: 7

Bitwise left operator

In [56]:
```
10<<2
```

Out[56]: 40

In [57]:
```
20<<4
```

Out[57]: 320

In [58]:
```
10>>2
```

Out[58]: 2

In [59]:
```
bin(20)
```

Out[59]: '0b10100'

In [60]:
```
20>>4
```

Out[60]: 1

Import math module

In [62]:
```
import math
import math as m
```

In [63]:
```
x=math.sqrt(25)
x
```

Out[63]: 5.0

In [64]:
```
x1=math.sqrt(15)
x1
```

Out[64]: 3.872983346207417

In [65]:
```
print(math.floor(2.9))
```

```
2
```

In [66]:
```
print(math.ceil(2.9))
```

```
3
```

In [67]:
```
print(math.pow(3,2))
```

```
9.0
```

In [68]:
```
print(math.pi)
```

```
3.141592653589793
```

In [69]:
```
print(math.e)
```

```
2.718281828459045
```

In [70]:
```
m.sqrt(10)
```

Out[70]: 3.1622776601683795

In [71]:
```
from math import sqrt,pow
pow(2,3)
```

Out[71]: 8.0

In [72]:
```
round(pow(2,3))
```

Out[72]: 8

User input function in python

In [74]:
```
x=input()
y=input()
z=x+y
print(z)
```

```
56
```

In [75]:
```
x1=input('Enter the 1st number')
y1=input('Enter thr 2nd number')
z1=x1+y1
print(z1)
```

```
56
```

In [76]:
```
type(x1)
type(y1)
```

Out[76]: str

In [77]:
```
x1=input('Enter the 1st number')
a1=int(x1)
y1=input('Enter the 2nd number')
b1=int(y1)
z1=a1+b1
print(z1)
```

```
11
```

Optimum the code

In [78]:
```
x2=int(input('Enter the 1st number'))
y2=int(input('Enter the 2nd number'))
z2=x2+y2
z2
```

Out[78]: 5

lets take input from the user in char format,but we don't have char format in python

In [80]:
```
ch=input('enter a char')
print(ch)
```

```
hello
```

In [81]:
```
print(ch[0])
```

```
h
```

In [82]:
```
print(ch[1])
```

```
e
```

In [83]:
```
print(ch[-1])
```

```
o
```

In [84]:
```
ch=input('enter a char')[0]
print(ch)
```

```
h
```

In [85]:
```
ch=input('enter a char')[1:3]
print(ch)
```

```
el
```

In [86]:
```
ch=input('enter a char')
print(ch)
```

```
2+3-4+10
```

EVAL function using input

In [168…
```
result=eval(input('enter an expr'))
print(result)
```

```
11
```

In [ ]:
```

```