Number system conversion

- binary: base(0,1),and divide by number/2 & count in reverse order,octal base:(0,7)
- hexadecimal:base(0,9 and a=10,b=11,c=13...till f ,(A,F/a,f)),
- In real time we use in ip config.. when we want to know about ip address

In [4]:
```
25
```

Out[4]: 25

In [6]:
```
bin(25)
```

Out[6]: '0b11001'

In [8]:
```
0b11001
```

Out[8]: 25

In [10]:
```
bin(45)
```

Out[10]: '0b101101'

In [12]:
```
int(0b110011)
```

Out[12]: 51

In [14]:
```
oct(13)
```

Out[14]: '0o15'

In [16]:
```
oct(67)
```

Out[16]: '0o103'

In [18]:
```
int(0o103)
```

Out[18]: 67

In [20]:
```
hex(6)
```

Out[20]: '0x6'

In [22]:
```
hex(70)
```

Out[22]: '0x46'

In [24]:
```
hex(10)
```

Out[24]: '0xa'

In [26]:
```
int(0x43)
```

Out[26]: 67

In [28]:
```
0xa
```

Out[28]: 10

Swap variable between two numbers in different methods

In [31]:
```
x=5
y=4
```

In [33]:
```
x,y =y,x
```

In [35]:
```
x
```

Out[35]: 4

In [37]:
```
y
```

Out[37]: 5

In [46]:
```
x1 = 67
x2 = 43
```

In [48]:
```
temp =x1
x1 = x2
x2 =temp
```

In [50]:
```
print(x1)
print(x2)
```

```
43
67
```

In [52]:
```
# using addition , sub method
a = 67
b = 45
```

In [54]:
```
a = a+b
b = a-b
a = a-b
print(a)
print(b)
```

```
45
67
```

In [58]:
```
a1 = 10
b1 = 20
```

In [60]:
```
a1 =a1+b1
b1=a1-b1
a1=a1-b1
```

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

```
20
10
```

In [62]:
```
print(0b101)
print(0b110)
```

```
5
6
```

In [64]:
```
print(bin(11))
print(0b1011)
```

```
0b1011
11
```

In [70]:
```
a=7
b=8
```

In [72]:
```
# another way of swap variable using xor\
a=a^b
b=a^b
a=a^b
```

In [74]:
```
print(a)
print(b)
```

```
8
7
```

bitwise operator

1. complement(~)
2. And(&)
3. OR(|)
4. XOR(^)
5. Left shift(<<)
6. Right shift(>>)

complement--> you will get this key below esc character

12==>1100 ||

first thing we need to understand what is mean by complement.

completement means it will do reverse of the binary format i.e -~0 it will give 0 12 binary format is 00001100 (complement of ~00001100 reverse the number -11110011 which is (-13)

but the question is why we got -13

to understand this concept (we have concept of 2's complement 2's complement means (1's complement ## complement means it will do reverse of the binary format i.e.-~0 it will give you 1~1 it will 0t+1)

in the system we can store +ve number but how to store -ve number

lets understand binary form of 13-00001101+1

In [82]:
```
~45
```

Out[82]: -46

In [84]:
```
~786
```

Out[84]: -787

In [86]:
```
~-56 # minus value
```

Out[86]: 55

In [91]:
```
~6+10j #complex
```

Out[91]: (-7+10j)

In [93]:
```
~0.78 #error(float)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[93], line 1
----> 1 ~0.78

TypeError: bad operand type for unary ~: 'float'
```

BITWISE OPERATOR

bit wise and operator

AND - LOGICAL OPERATOR|||& -BITWISE AND OPERATOR

We know that 1 & is 1)

12 -00001100

13 -00001101

when we add both then output we will get as 12

In [104…
```
12 & 13
```
Out[104… 12

In [106…
```
12 & 13
```
Out[106… 12

In [108…
```
1&1
```
Out[108… 1

In [110…
```
1|1
```
Out[110… 1

In [112…
```
1 & 0
```
Out[112… 0

in XOR if the both number are different then we will get 1 or else we will get

In [114…
```
12^13
```
Out[114… 1

In [116…
```
25^30
```
Out[116… 7

In [118…
```
bin(25)
```
Out[118… '0b11001'

In [120…
```
bin(30)
```
Out[120… '0b11110'

In [124…
```
int(0b11110)
```
Out[124… 30

BIT WISE LEFT OPERATOR

Bit wise left operator by default you will take 2 zeroes()

10 binary operator is 1010 | also i can say 1010

In [129…
```
10<<2
```

Out[129…
```
40
```

In [131…
```
50<<3
```

Out[131…
```
400
```

BITWISE RIGHT SHIFT OPERATOR

In [134…
```
10>>2
```

Out[134…
```
2
```

In [136…
```
bin(20)
```

Out[136…
```
'0b10100'
```

In [138…
```
50>>2
```

Out[138…
```
12
```

Import math Module

In [141…
```
x = sqrt(625) # here sqrt is inbult function
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[141], line 1
----> 1 x = sqrt(625)

NameError: name 'sqrt' is not defined
```

In [143…
```
import math
```

In [147…
```
x= math.sqrt(625)
x
```

Out[147…
```
25.0
```

In [149…
```
print(math.floor(2.9)) # minimum or least value
```

```
2
```

In [153…
```
print(math.ceil(2.9))
```

```
3
```

In [157…
```
print(math.pow(6,2))
```

```
36.0
```

In [159…
```
print(math.pi) # constant value
```

```
3.141592653589793
```

In [161…
```
print(math.e)
```

```
2.718281828459045
```

In [163…
```
import math as m
```

In [165…
```
m.sqrt(1225)
```

Out[165… 35.0

In [167…
```
import math as m
m.pow(9,7)
```

Out[167… 4782969.0

In [169…
```
from math import pow
pow(2,3)
```

Out[169… 8.0

In [171…
```
from math import ceil
ceil(8.97)
```

Out[171… 9

In [173…
```
from math import *
print(pow(4,6))
print(ceil(5.5))
```

```
4096.0
6
```

In [175…
```
round(pow(9,2))
```

Out[175… 81

User input Function ||comand line input

In [178…
```
r =input()
z =input()
c=r+z
print(c)
```

```
23
```

In [180…
```
z1=input('first number')
z2=input('second number')
z3= z1+z2
z3
```

Out[180…
```
'55'
```

In [182…
```
type(z1)
type(z3)
```

Out[182…
```
str
```

In [184…
```
x1=input("enter number")
a =int(x1)
x2=input("enter a number")
b=int(x2)
c=a+b
print(c)
```

```
30
```

In [186…
```
a=int(input("1st number"))
b=int(input("2nd number"))
c=a+b
c
```

Out[186…
```
50
```

In [ ]:
```
lets take input from the user in char format, but we dont have char format in py
```

In [188…
```
ch = input("enter a char")
print(ch)
```

```
krishna
```

In [190…
```
ch
```

Out[190…
```
'krishna'
```

In [192…
```
print(ch[0])
```

```
k
```

In [194…
```
print(ch[-1])
```

```
a
```

In [196…
```
ch= input("enter a character")[0]
ch
```

Out[196…
```
'k'
```

In [198…
```
ch = input("enter ")[1:3]
ch
```

Out[198…
```
'rj'
```

In [200…
```
ch=input("enter")
ch
```

Out[200… 'arjun'

Eval function using input

In [207…
```
result =eval(input('enter a expr'))
print(result)
```

4

In [ ]: