Number System Conversion.

```
In [2]: 15
Out[2]: 15

In [3]: bin(15)
Out[3]: '0b1111'

In [4]: bin(10)
Out[4]: '0b1010'

In [5]: bin(25)
Out[5]: '0b11001'

In [1]: int(0b11001)
Out[1]: 25

In [3]: bin(35)
Out[3]: '0b100011'

In [7]: int(0b100011)
Out[7]: 35

In [9]: oct(15)
Out[9]: '0o17'

In [12]: 0o17
Out[12]: 15

In [14]: hex(9)
Out[14]: '0x9'

In [16]: 0xf
Out[16]: 15

In [18]: hex(25)
```

```
Out[18]:  '0x19'
```

```
In [20]:  0x19
```

```
Out[20]:  25
```

```
In [22]:  0x15
```

```
Out[22]:  21
```

Swap Variable in Python

```
In [25]:  a=5         #a,b=5,6 After swap we should get ==>(a,b=6,5)
          b=6
```

```
In [27]:  a=b
          b=a
```

```
In [29]:  a,b=b,a
```

```
In [31]:  print(a)
          print(b)
```

```
6
6
```

```
In [33]:  a1=7              # In above scenario we lost the value 5
          b1=8
```

```
In [35]:  temp=a1
          a1=b1
          b1=temp
```

```
In [37]:  print(a1)
          print(b1)
```

```
8
7
```

```
In [39]:  a2=5
          b2=6
```

```
In [41]:  #swap variable formulas
          a2 = a2 + b2
          b2 = a2 - b2
          a2 = a2 - b2
```

```
In [43]:  print(a2)
          print(b2)
```

```
6
5
```

BITWISE OPERATOR

In [46]:
```
print(bin(12))
print(bin(13))
```

```
0b1100
0b1101
```

1.Complement (~) (TILDE OR TILD)

In [52]:
```
~12    # why we get -13 . first we understand what is complment means (reverse of b
```

Out[52]:
```
-13
```

In [54]:
```
~45
```

Out[54]:
```
-46
```

In [56]:
```
~88
```

Out[56]:
```
-89
```

In [58]:
```
~0
```

Out[58]:
```
-1
```

In [61]:
```
~1
```

Out[61]:
```
-2
```

2.AND (&)

In [63]:
```
12&13
```

Out[63]:
```
12
```

In [65]:
```
1&1
```

Out[65]:
```
1
```

In [67]:
```
1&0
```

Out[67]:
```
0
```

In [69]:
```
10&20
```

Out[69]:
```
0
```

In [71]:
```
30&70
```

Out[71]:
```
6
```

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

```
Out[79]:  32
```

3. OR (|)

```
In [75]:  1|0

Out[75]:  1

In [77]:  12|13

Out[77]:  13

In [81]:  35|40

Out[81]:  43

In [83]:  60|30

Out[83]:  62

In [87]:  47|23

Out[87]:  63
```

4.XOR (^) #In XOR if the both number are different then we will get 1 or else we will get 0

```
In [91]:  12^13

Out[91]:  1

In [93]:  25^30

Out[93]:  7

In [97]:  22^42

Out[97]:  60
```

5. Left Shift operator (<<) # Bit wise left shift operator bydefault you will take 2 zeros.

```
In [101…  10<<2

Out[101…  40

In [107…  20<<4

Out[107…  320

In [109…  20<<3

Out[109…  160
```

6.Right Shift operator (>>) # Bit wise right shift operator bydefault it will remove 2 zeros.

```
In [111…  10>>2
```

```
Out[111…  2

In [113…  20>>4

Out[113…  1

In [119…  60>>3

Out[119…  7
```

import math module

https://docs.python.org/3/library/math.html

```
In [122…  x=sqrt(25)

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[122], line 1
----> 1 x=sqrt(25)

NameError: name 'sqrt' is not defined

In [124…  import math

In [126…  x=math.sqrt(25)
          x

Out[126…  5.0

In [128…  x1 = math.sqrt(15)
          x1

Out[128…  3.872983346207417

In [130…  print(math.floor(2.9)) #floor - minimum or least value

2

In [134…  print(math.floor(4.5))

4

In [136…  print(math.floor(10.777))

10

In [132…  print(math.ceil(2.9)) #ceil - maximum or highest value

3

In [138…  print(math.ceil(10.77))
```

```
11
```

In [140…
```
print(math.ceil(18.33))
```
```
19
```

In [142…
```
print(math.pow(3,2))
```
```
9.0
```

In [144…
```
print(math.pow(6,3))
```
```
216.0
```

In [146…
```
print(math.pow(2,4))
```
```
16.0
```

In [148…
```
print(math.pi) #these are constant
```
```
3.141592653589793
```

In [150…
```
print(math.e) #these are constant
```
```
2.718281828459045
```

In [152…
```
import math as m
m.sqrt(10)
```
Out[152…
```
3.1622776601683795
```

In [154…
```
m.sqrt(40)
```
Out[154…
```
6.324555320336759
```

In [156…
```
m.sqrt(25)
```
Out[156…
```
5.0
```

In [158…
```
m.sqrt(625)
```
Out[158…
```
25.0
```

In [160…
```
from math import sqrt,pow # math has many function if you want to call specific fun
pow(2,3)
```
Out[160…
```
8.0
```

In [162…
```
from math import * # math has many function if you want to call specific function t

print(pow(2,3))
print(floor(2.3))
```
```
8.0
2
```

In [164…
```
round(pow(2,3))
```

Out[164…

```
8
```

User Input Function n python || Command Line Input (cli)

In [167…

```python
x = input()
y = input()
z = x + y
print(z)
```

```
46
```

In [169…

```python
x1 = input('Enter the 1st number') #whenevery you works in input function it always
y1 = input('Enter the 2nd number') # it wont understand as arithmetic operator
z1 = x1 + y1
print(z1)
```

```
105
```

In [173…

```python
type(x1)
type(y1)
```

Out[173…

```
str
```

In [175…

```python
x1 = input('Enter the 1st number') #whenevery you works in input function it always
a1 = int(x1)
y1 = input('Enter the 2nd number') # it wont understand as arithmetic operator
b1 = int(y1)
z1 = a1 + b1
print(z1)
```

```
50
```

In [177…

```python
x2 = int(input('Enter the 1st number'))
y2 = int(input('Enter the 2nd number'))
z2 = x2 + y2
z2                                              #From the above code we notice that we ar
```

Out[177…

```
100
```

Lets take input from the user in char format But we dont't have char format in python

In [179…

```python
ch = input('enter a char')
print(ch)
```

```
yashwanth
```

In [181…

```python
print(ch[0])
```

```
y
```

In [183…

```python
print(ch[1])
```

```
a
```

```
In [185…  print(ch[-1])
```
```
h
```
```
In [187…  ch = input('enter a char')[0]
          print(ch)
```
```
y
```
```
In [189…  ch = input('enter a char')[1:3]
          print(ch)
```
```
as
```
```
In [191…  ch = input('enter a char')
          print(ch) # if you enter as 2 + 6 -1 we get output as 2 + 6-1 only
```
```
2+6-1
```
```
In [193…  ch = input('enter a char')
          print(ch)
```
```
3+20-8
```

EVAL function using input

```
In [195…  result = eval(input('enter an expr'))
          print(result)
```
```
3
```
```
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
```