In [1]:
```
1+1 #addition
```
Out[1]:
```
2
```

In [2]:
```
2-1 #subtraction
```
Out[2]:
```
1
```

In [3]:
```
3*4 #multiplication
```
Out[3]:
```
12
```

In [4]:
```
8/4 #division
```
Out[4]:
```
2.0
```

In [5]:
```
8//4
```
Out[5]:
```
2
```

In [9]:
```
2 ** 5
```
Out[9]:
```
32
```

In [6]:
```
8 + 9 - 7 #BODMAS-(Brackets, orders, division, multiplication, addition, subtrac
```
Out[6]:
```
10
```

In [7]:
```
(5+5)*5
```
Out[7]:
```
50
```

In [8]:
```
2*2*2*2*2
```
Out[8]:
```
32
```

In [10]:
```
14 % 2
```
Out[10]:
```
0
```

In [13]:
```
a,b,c,d,e = 15,7.8,'Chinnu', 8+9j, True
print(a)
print(b)
print(c)
print(d)
print(e)
```
```
15
7.8
Chinnu
(8+9j)
True
```

In [14]:
```
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'complex'>
<class 'bool'>
```

```
In [16]: print("Hello World!")
```

```
Hello World!
```

```
In [17]: a = 2
         b = 3
         a+b
```

```
Out[17]: 5
```

```
In [18]: c = a+b
```

```
In [19]: c
```

```
Out[19]: 5
```

```
In [22]: print('Jujoori Greeshma')
```

```
Jujoori Greeshma
```

```
In [31]: 'Jujoori ' + 'Greeshma ' + 'Sri'
```

```
Out[31]: 'Jujoori Greeshma Sri'
```

```
In [33]: 5 * 'Chinnu'
```

```
Out[33]: 'ChinnuChinnuChinnuChinnuChinnu'
```

```
In [34]: 5 * ' Chinnu'
```

```
Out[34]: ' Chinnu Chinnu Chinnu Chinnu Chinnu'
```

```
In [37]: print('C:\nit') #\n -> next line/new line
```

```
C:
it
```

```
In [38]: print(r'C:\nit') #r -> raw string
```

```
C:\nit
```

```
In [39]: #print result with string
```

```
In [41]: num1 = 20
         num2 = 30
         add=num1+num2
         print('The addition of',num1,'and',num2,'is =',add)
```

```
The addition of 20 and 30 is = 50
```

```
In [42]: #.format( ) string print method
```

In [43]:

```
num1 = 20
num2 = 30
add=num1+num2
print('The addition of {} and {} is = {}'.format(num1,num2,add))
```

```
The addition of 20 and 30 is = 50
```

In [44]:

```
#More shorter format method (f string method)
```

In [45]:

```
num1 = 20
num2 = 30
add=num1+num2
print(f'The addition of {num1} and {num2} is = {add}')
```

```
The addition of 20 and 30 is = 50
```

In [46]:

```
#ROUND
```

In [47]:

```
num1=100
num2=25
```

In [50]:

```
num3=333
avg=round((num1+num2+num3)/3)
print(f'The avrage of {num1}, {num2} and {num3} is = {avg}')
```

```
The avrage of 100, 25 and 333 is = 153
```

In [51]:

```
# let's combine all
num1=10
num2=20
add = num1+ num2
print('The addition of',num1,'and',num2,'is=',add)
print('The addition of {} and {} is= {}'.format(num1,num2,add))
print(f'The addition of {num1} and {num2} is= {add}')
```

```
The addition of 10 and 20 is= 30
The addition of 10 and 20 is= 30
The addition of 10 and 20 is= 30
```

In [52]:

```
#end Statement
```

In [53]:

```
print('hello')
print('good moorning')
```

```
hello
good moorning
```

In [58]:

```
print('hello', end='_')
print('world have a good day')
```

```
hello_world have a good day
```

In [59]:

```
#separator
```

In [60]:

```
print('hello','hai','how are you',sep='--->')
```

```
hello--->hai--->how are you
```

In [63]:

```
print('One','Two','Three', sep="♥♥♥")
```

```
One♥♥♥Two♥♥♥Three
```

```
In [64]: #OPERATORS
```

```
In [65]: #Arithmetic Operator
```

```
In [66]: x1, y1 = 10, 5
         x1 + y1

Out[66]: 15
```

```
In [67]: x1 - y1

Out[67]: 5
```

```
In [68]: x1 * y1

Out[68]: 50
```

```
In [69]: x1 / y1

Out[69]: 2.0
```

```
In [70]: x1 // y1

Out[70]: 2
```

```
In [72]: x1 % y1

Out[72]: 0
```

```
In [73]: x1 ** y1

Out[73]: 100000
```

```
In [74]: 2 ** 3

Out[74]: 8
```

```
 In [ ]: #Assignment operator
```

```
In [75]: x = 2
```

```
In [76]: x = x + 2
         x

Out[76]: 4
```

```
In [77]: x += 2
         x

Out[77]: 6
```

```
In [78]: x += 2
         x

Out[78]: 8
```

In [79]:
```
x -= 2
x
```

Out[79]:
```
6
```

In [80]:
```
x *= 2
x
```

Out[80]:
```
12
```

In [81]:
```
x /= 2
x
```

Out[81]:
```
6.0
```

In [82]:
```
#Unary operator
```

In [83]:
```
n = 7 # negation
```

In [84]:
```
m = -(n)
```

In [85]:
```
m
```

Out[85]:
```
-7
```

In [86]:
```
n
```

Out[86]:
```
7
```

In [87]:
```
-n
```

Out[87]:
```
-7
```

In [88]:
```
#Relational operator
```

In [89]:
```
a = 5
b = 7
```

In [90]:
```
a == b
```

Out[90]:
```
False
```

In [91]:
```
a < b
```

Out[91]:
```
True
```

In [93]:
```
a > b
```

Out[93]:
```
False
```

In [94]:
```
#a = b; we cannot use = operator; because that means it is assigning
```

In [95]:
```
m = 5
n = 5
```

```
In [96]:  m == n

Out[96]:  True

In [97]:  a >= b

Out[97]:  False

In [98]:  a<=b

Out[98]:  True

In [99]:  a != b

Out[99]:  True

In [100…  #LOGICAL OPERATOR

In [101…  a = 5
          b = 4

In [102…  a < 8 and b < 5

Out[102…  True

In [103…  a < 8 or b < 2

Out[103…  True

In [104…  x = False
          x

Out[104…  False

In [105…  not x

Out[105…  True

In [106…  a < 8 ^ b < 5

Out[106…  False

In [107…  #Number system conversion

In [108…  #Binary

In [109…  25

Out[109…  25

In [117…  bin(25) #in output b means -> binary

Out[117…  '0b11001'

In [118…  int(0b11001)
```

Out[118…

```
25
```

In [119…

```
#Octal
```

In [120…

```
oct(15) #in output o means -> octal
```

Out[120…

```
'0o17'
```

In [122…

```
#Hexadecimal
```

In [124…

```
hex(25) #in output x means -> hexadecimal
```

Out[124…

```
'0x19'
```

In [125…

```
hex(10)
```

Out[125…

```
'0xa'
```

In [126…

```
#Swap variable in python
```

In [145…

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

In [146…

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

```
6
6
```

In [147…

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

In [148…

```
temp = a1
a1 = b1 # swapping values using third variable
b1 = temp
```

In [149…

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

```
8
7
```

In [160…

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

In [153…

```
a2 = a2 + b2 # 5 + 6 = 11
b2 = a2 - b2 # 11 - 6 = 5
a2 = a2 - b2 # 11 - 5 = 6
```

In [154…

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

```
6
5
```

In [155…

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

```
5
6
```

In [156…

```
a2 = a2 ^ b2
b2 = a2 ^ b2
a2 = a2 ^ b2
```

In [157…

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

```
5
6
```

In [163…

```
a2,b2=b2,a2 # Swapped the values
```

In [165…

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

```
6
5
```

In [167…

```
#complement- (~)
```

In [168…

```
#1's Compliment --> reverse of binary format
```

In [169…

```
#2's complemenet --> 1's complement + 1
```

In [171…

```
~12 # why we get -13 ; To remember: ~x = -(x+1)
```

Out[171…

```
-13
```

In [172…

```
~-2
```

Out[172…

```
1
```

In [173…

```
~-13
```

Out[173…

```
12
```

In [174…

```
12 & 13
```

Out[174…

```
12
```

In [175…

```
13 | 12
```

Out[175…

```
13
```

In [176…

```
13 ^ 12
```

Out[176…

```
1
```

In [177…

```
#In XOR if the both number are different then we will get 1 or else we'll get 0
```

In [178…

```
12 ^ 13
```

```
Out[178…   1

In [179…   5^5

Out[179…   0

In [180…   #LEFT Shift (Gain bits)

In [181…   10<<3

Out[181…   80

In [185…   # 10 -> 1010
           # 10 << 2= 3 --> 1010(000) --> Here 3 bits are gained

In [186…   #Right Shift (Lose bits)

In [187…   10>>1

Out[187…   5

In [188…   # 10 -> 1010
           # 10>>1 -> 101 Here 1 bit is lost

In [189…   10>>5

Out[189…   0

In [190…   #help() --> to get help in jupyer notebok
           #type (q) to exit

In [191…   x = sqrt(25)

          ---------------------------------------------------------------------------
          NameError                                 Traceback (most recent call last)
          Cell In[191], line 1
          ----> 1 x = sqrt(25)

          NameError: name 'sqrt' is not defined

In [201…   import math as m # math is module

In [193…   x = m.sqrt(25)
           x

Out[193…   5.0

In [194…   #floor - minimum or least value

In [195…   print(m.floor(25.628292))

          25

In [196…   #ceil - maximum or highest value

In [197…   print(m.ceil(25.6259))
```

```
26
```

In [198…

```
print(m.pow(3,2)) #power, 3^2
```

```
9.0
```

In [199…

```
print(m.pi)
```

```
3.141592653589793
```

In [202…

```
print(m.e) # e --> epsilon value
```

```
2.718281828459045
```

In [203…

```
from math import pow,sqrt
```

In [204…

```
print(m.sqrt(64))
print(m.pow(10,2))
```

```
8.0
100.0
```

In [205…

```
from math import *
```

In [206…

```
#input ( ): always consider value as STRING
```

In [207…

```
x = input()
x
```

Out[207…

```
'jeon chinnu'
```

In [208…

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

```
54
```

In [209…

```
ch = input('enter a character: ')
```

In [210…

```
print(ch[:2])
```

```
Je
```

In [212…

```
print(ch[5])
```

```
C
```

In [213…

```
ch = input('enter a character: ')[1:4]
ch
```

Out[213…

```
'eon'
```

In [214…

```
#Eval function : Used to evaluate mathematical expressions
```

In [215…

```
a = eval(input('Enter expression: '))
a
```

Out[215…

```
-15
```