

```
In [72]: 1 import os
          2 os.getcwd()
```

Out[72]: 'C:\\Users\\HARI KARTHIK.K'

```
In [1]: 1 import sys
         2 print(sys.version)
```

3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)]

```
In [2]: 1 !jupyter --version
```

Selected Jupyter core packages...

IPython	: 7.31.1
ipykernel	: 6.15.2
ipywidgets	: 7.6.5
jupyter_client	: 7.3.4
jupyter_core	: 4.11.1
jupyter_server	: 1.18.1
jupyterlab	: 3.4.4
nbclient	: 0.5.13
nbconvert	: 6.4.4
nbformat	: 5.5.0
notebook	: 6.4.12
qtconsole	: 5.2.2
traitlets	: 5.1.1

```
In [56]: 1 import keyword
          2 print(keyword.kwlist)
          3 len(keyword.kwlist)
```

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Out[56]: 35

Python data types

- int
- float
- bool
- string
- complex

```
In [1]: 1 i=5  
        2 i
```

```
Out[1]: 5
```

```
In [2]: 1 type(i)
```

```
Out[2]: int
```

```
In [3]: 1 f = 14.03  
        2 f
```

```
Out[3]: 14.03
```

```
In [4]: 1 type(f)
```

```
Out[4]: float
```

```
In [5]: 1 i+f
```

```
Out[5]: 19.03
```

```
In [6]: 1 i-f
```

```
Out[6]: -9.03
```

In [7]: 1 i*f

Out[7]: 70.14999999999999

In [8]: 1 i/f

Out[8]: 0.3563791874554526

In [9]: 1 i//f

Out[9]: 0.0

In [10]: 1 i%f

Out[10]: 5.0

In [11]: 1 True

Out[11]: True

In [12]: 1 False

Out[12]: False

In [13]: 1 True+False

Out[13]: 1

In [14]: 1 False+False

Out[14]: 0

In [15]: 1 True*False

Out[15]: 0

```
In [16]: 1 True-False
```

```
Out[16]: 1
```

```
In [17]: 1 s='karthik'
          2 s
```

```
Out[17]: 'karthik'
```

```
In [18]: 1 c = 10+20j
          2 d = 20+30j
```

```
In [24]: 1 print(c.real)
          2 print(c.imag)
          3 print(d.real)
          4 print(d.imag)
```

```
10.0
```

```
20.0
```

```
20.0
```

```
30.0
```

```
In [26]: 1 print(c+d)
          2 print(c-d)
          3 print(c*d)
          4 print(c/d)
```

```
(30+50j)
```

```
(-10-10j)
```

```
(-400+700j)
```

```
(0.6153846153846154+0.0769230769230769j)
```

Data Type Casting - Convert one Datatype to Other Datatype

```
In [27]: 1 int(2.4)
```

```
Out[27]: 2
```

```
In [28]: 1 int(True)
```

```
Out[28]: 1
```

```
In [29]: 1 int(False)
```

```
Out[29]: 0
```

```
In [30]: 1 int('10')
```

```
Out[30]: 10
```

```
In [32]: 1 int(1+2j)
```

```
-----  
TypeError  
Cell In[32], line 1  
----> 1 int(1+2j)
```

Traceback (most recent call last)

TypeError: int() argument must be a string, a bytes-like object or a real number, not 'complex'

```
In [31]: 1 float(20)
```

```
Out[31]: 20.0
```

```
In [33]: 1 float(10,20)
```

```
-----  
TypeError  
Cell In[33], line 1  
----> 1 float(10,20)
```

Traceback (most recent call last)

TypeError: float expected at most 1 argument, got 2

```
In [34]: 1 float(1+2j)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[34], line 1  
----> 1 float(1+2j)  
  
TypeError: float() argument must be a string or a real number, not 'complex'
```

```
In [35]: 1 float(True)
```

```
Out[35]: 1.0
```

```
In [36]: 1 float(False)
```

```
Out[36]: 0.0
```

```
In [37]: 1 float('10')
```

```
Out[37]: 10.0
```

```
In [38]: 1 float('ten')
```

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[38], line 1  
----> 1 float('ten')  
  
ValueError: could not convert string to float: 'ten'
```

```
In [39]: 1 str(8)
```

```
Out[39]: '8'
```

```
In [40]: 1 str(8.8)
```

```
Out[40]: '8.8'
```

```
In [41]: 1 str(1+2j)
```

```
Out[41]: '(1+2j)'
```

```
In [42]: 1 str(True)
```

```
Out[42]: 'True'
```

```
In [48]: 1 str(9,9.0)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[48], line 1  
----> 1 str(9,9.0)  
  
TypeError: str() argument 'encoding' must be str, not float
```

```
In [50]: 1 str()
```

```
Out[50]: ''
```

```
In [51]: 1 bool()
```

```
Out[51]: False
```

```
In [52]: 1 bool(10)
```

```
Out[52]: True
```

```
In [53]: 1 bool(9.0)
```

```
Out[53]: True
```

```
In [54]: 1 bool('karthik')
```

```
Out[54]: True
```

```
In [55]: 1 bool(0)
```

```
Out[55]: False
```

```
In [56]: 1 bool(1+2j)
```

```
Out[56]: True
```

```
In [57]: 1 complex(10)
```

```
Out[57]: (10+0j)
```

```
In [58]: 1 complex(10,20)
```

```
Out[58]: (10+20j)
```

```
In [59]: 1 complex(10,20,30)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[59], line 1  
----> 1 complex(10,20,30)  
  
TypeError: complex() takes at most 2 arguments (3 given)
```

```
In [60]: 1 complex(10.20,30.8)
```

```
Out[60]: (10.2+30.8j)
```

```
In [62]: 1 complex(True)
```

```
Out[62]: (1+0j)
```



```
In [63]: 1 complex(False)
```

```
Out[63]: 0j
```

```
In [64]: 1 complex('10')
```

```
Out[64]: (10+0j)
```

```
In [65]: 1 complex('10','20')
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[65], line 1  
----> 1 complex('10','20')  
  
TypeError: complex() can't take second arg if first is a string
```

```
In [66]: 1 complex(True,False)
```

```
Out[66]: (1+0j)
```

```
In [8]: 1 str1 = 'HELLO PYTHON'  
      2 str1
```

```
Out[8]: 'HELLO PYTHON'
```

```
In [9]: 1 type(str1)
```

```
Out[9]: str
```

```
In [10]: 1 str1[0]
```

```
Out[10]: 'H'
```

```
In [11]: 1 str1[1]
```

```
Out[11]: 'E'
```

Forward Index

```
In [15]: 1 print(str1[0])
          2 print(str1[1])
          3 print(str1[2])
          4 print(str1[3])
          5 print(str1[4])
          6 print(str1[5])
          7 print(str1[6])
          8 print(str1[7])
          9 print(str1[8])
         10 print(str1[9])
         11 print(str1[10])
         12 print(str1[11])
```

H
E
L
L
O

P
Y
T
H
O
N

Backward Index

```
In [17]: 1 print(str1[-1])
          2 print(str1[-2])
          3 print(str1[-3])
          4 print(str1[-4])
          5 print(str1[-5])
          6 print(str1[-6])
          7 print(str1[-7])
          8 print(str1[-8])
          9 print(str1[-9])
         10 print(str1[-10])
         11 print(str1[-11])
         12 print(str1[-12])
```

N
O
H
T
Y
P

O
L
L
E
H

```
In [18]: 1 str1
```

```
Out[18]: 'HELLO PYTHON'
```

```
In [19]: 1 len(str1)
```

```
Out[19]: 12
```

Slicing :

Forward Slicing

```
In [29]: 1 print(str1[6:12])
```

PYTHON

Step Slicing

```
In [32]: 1 str1[0:12]
```

```
Out[32]: 'HELLO PYTHON'
```

```
In [33]: 1 str1[0:12:3]
```

```
Out[33]: 'HLPH'
```

```
In [36]: 1 str2='hellopython'  
2 str2[1:8:2]
```

```
Out[36]: 'elpt'
```

```
In [37]: 1 str2
```

```
Out[37]: 'hellopython'
```

```
In [39]: 1 str2[::2]
```

```
Out[39]: 'hloyhn'
```

```
In [40]: 1 str2[::3]
```

```
Out[40]: 'hlyo'
```

```
In [43]: 1 s1 = 'HARI'
          2 s2 = ' KARTHIK'
          3 s3 = s1+s2
          4 print(s3)
```

HARI KARTHIK

Python Basic Operator

Arithmetic Operator

```
In [6]: 1 x1,y1 = 10,5
```

```
In [7]: 1 x1+y1
```

Out[7]: 15

```
In [8]: 1 x1-y1
```

Out[8]: 5

```
In [9]: 1 x1*y1
```

Out[9]: 50

```
In [10]: 1 x1/y1
```

Out[10]: 2.0

```
In [11]: 1 x1//y1
```

```
Out[11]: 2
```

```
In [12]: 1 x1%y1
```

```
Out[12]: 0
```

```
In [15]: 1 x1**y1
```

```
Out[15]: 100000
```

```
In [17]: 1 x2,y2=3,3  
2 x2**y2  
3
```

```
Out[17]: 27
```

Assignment Operator

```
In [23]: 1 y=2
```

```
In [24]: 1 y+=2
```

```
In [25]: 1 y
```

```
Out[25]: 4
```

```
In [26]: 1 y+=2  
        2 y
```

Out[26]: 6

```
In [27]: 1 y+=2  
        2 y
```

Out[27]: 8

```
In [28]: 1 y*=2
```

```
In [29]: 1 y
```

Out[29]: 16

```
In [30]: 1 y-=2  
        2 y
```

Out[30]: 14

```
In [31]: 1 y/=2
```

```
In [32]: 1 y
```

Out[32]: 7.0

```
In [33]: 1 y//=2  
        2 y
```

Out[33]: 3.0

Unary Operator

```
In [43]: 1 x = 5
          2 y = +x # y will be 5
          3 z = -3
          4 w = +z # w will be -3
          5 print(x,y,z,w)
```

5 5 -3 -3

```
In [44]: 1 x = 5
          2 y = -x # y will be -5
          3 z = -3
          4 w = -z # w will be 3
          5 print(x,y,z,w)
```

5 -5 -3 3

Relational Operator

```
In [34]: 1 a=5
          2 b=6
```

```
In [35]: 1 a<b
```

Out[35]: True


```
In [36]: 1 a>b
```

```
Out[36]: False
```

```
In [37]: 1 a==b
```

```
Out[37]: False
```

```
In [39]: 1 a!=b
```

```
Out[39]: True
```

```
In [40]: 1 b=5
```

```
In [41]: 1 a==b
```

```
Out[41]: True
```

Logical Operator

```
In [45]: 1 a,b = 5,4
```

```
In [46]: 1 a<8 and b<5
```

```
Out[46]: True
```

```
In [47]: 1 a<8 and b<2
```

```
Out[47]: False
```

```
In [48]: 1 a<8 or b<2
```

```
Out[48]: True
```

```
In [50]: 1 a>8 or b<2
```

```
Out[50]: False
```

```
In [51]: 1 x=False  
2 x
```

```
Out[51]: False
```

```
In [52]: 1 not x
```

```
Out[52]: True
```

Python Bitwise Operator

Complement Operator (~) - Reverse of Binary format

```
In [53]: 1 ~12
```

```
Out[53]: -13
```

```
In [54]: 1 ~46
```

```
Out[54]: -47
```

AND operator(&)

```
In [1]: 1 12 & 13 # 1100 & 1101 = 1100=>bin(1100)=12
```

```
Out[1]: 12
```

OR operator (|)

```
In [4]: 1 12 | 13 #1100 | 1101 = 1101 => 13
```

```
Out[4]: 13
```

XOR (^)

```
In [5]: 1 12 ^ 13 # 1100 ^ 1101 = 0001 => 1
```

```
Out[5]: 1
```

```
In [6]: 1 25 ^ 30 #11001 ^ 11110 => 00111 = 7
```

```
Out[6]: 7
```

Left Shift (<<) -Gain bits

```
In [8]: 1 10<<1 #001010<<1 = 010100 = 20
```

```
Out[8]: 20
```

```
In [9]: 1 10<<2 #001010<<2 = 101000 = 40
```

```
Out[9]: 40
```

Right Shift(>>) - Lose bits

```
In [10]: 1 10>>1 #001010>>1 = 00101 => 5
```

```
Out[10]: 5
```

```
In [11]: 1 10>>2 #001010>>2 = 10 => 2
```

```
Out[11]: 2
```

Python Number System

- Binary Number System - base 2 (0,1)
- Octal Number System - base 8 (0,1,2,3,4,5,6,7)
- Decimal Number System - base10 (0,1,2,3,4,5,6,7,8,9)
- Hexadecimal Number System - base16 (0-9,a,b,c,d,e,f)

Binary Number System

```
In [59]: 1 25
```

```
Out[59]: 25
```

```
In [60]: 1 bin(25)
```

```
Out[60]: '0b11001'
```

```
In [61]: 1 0b11001
```

```
Out[61]: 25
```

```
In [62]: 1 oct(25)
```

```
Out[62]: '0o31'
```

```
In [64]: 1 hex(25)
```

```
Out[64]: '0x19'
```

```
print()
```

```
In [15]: 1 print(10)
          2 print(10,20)
          3 print('python')
          4 print(10,20,'python',1+2j,True,2.3,[1,2],{1,2})
```

```
10
10 20
python
10 20 python (1+2j) True 2.3 [1, 2] {1, 2}
```

```
In [16]: 1 num1=20
          2 num2=30
          3 res = num1+num2
          4 print(res)
```

```
50
```

```
print result with string
```

```
In [19]: 1 print("The addition of",num1,"&",num2,"=",res)
```

```
The addition of 20 & 30 = 50
```

```
In [20]: 1 name='Hari Karthik'
          2 age = 22
          3 city = 'Vjw'
          4 print('My name is',name,"I'm",age,'years old',"and I'm from",city)
```

```
My name is Hari Karthik I'm 22 years old and I'm from Vjw
```

```
In [24]: 1 #print format method
          2 num1 = 20
          3 num2 = 30
          4 res = 20+30
          5 print('The addition of {} and {} is = {}'.format(num1,num2,res))
```

The addition of 20 and 30 is = 50

```
In [25]: 1 # print f string method
          2
          3 num1 = 20
          4 num2 = 30
          5 res = 20+30
          6 print(f'The addition of {num1} and {num2} is = {res}')
```

The addition of 20 and 30 is = 50

```
In [26]: 1 # END statement
          2 print("hello",end="***")
          3 print("good morning", end="^^^")
```

hello***good morning^^^

```
In [27]: 1 #Seprator
          2 print('hello','hai','how are you',sep='--->')
```

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

```
In [32]: 1 print('hello',end=' ')
          2 print('good morning')
```

hello good morning

```
In [35]: 1 print('hello',sep='')
          2 print('good morning')
```

hello
good morning

```
In [1]: 1 a=5
        2 b=6
        3 print(a,b)
```

5 6

```
In [2]: 1 #Here we lost 5 value for variable a
        2 a=b
        3 b=a
        4 print(a,b)
```

6 6

Swapping two variables in python

```
In [5]: 1 a1 = 10
        2 b1 = 20
        3 print(10,20)
```

10 20

- Swap using three variables

```
In [6]: 1 temp = a1 # temp = 10
        2 a1 = b1 # a1 = 20
        3 b1 = temp # b1 = 10
        4 print(a1,b1)
```

20 10

- Swap using two variables (useful in interviews)

In [8]:

```
1 a2 = 5
2 b2 = 6
3 print(a2,b2)
```

5 6

In [9]:

```
1 a2 = a2+b2 # 5+6 = 11
2 b2 = a2 - b2 #11-6 = 5
3 a2 = a2-b2 # 11-5 = 6
4 print(a2,b2)
```

6 5