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

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

In [1]:
```
import sys
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]:
```
!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]:
```
import keyword
print(keyword.kwlist)
len(keyword.kwlist)
```

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

Out[56]:
```
35
```

> Python data types

- int
- float
- bool
- string
- complex

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

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

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

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

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

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

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

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

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

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

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

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

```
In [7]: i*f

Out[7]: 70.14999999999999

In [8]: i/f

Out[8]: 0.3563791874554526

In [9]: i//f

Out[9]: 0.0

In [10]: i%f

Out[10]: 5.0

In [11]: True

Out[11]: True

In [12]: False

Out[12]: False

In [13]: True+False

Out[13]: 1

In [14]: False+False

Out[14]: 0

In [15]: True*False

Out[15]: 0
```

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

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

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

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

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

In [24]:
```
print(c.real)
print(c.imag)
print(d.real)
print(d.imag)
```

```
10.0
20.0
20.0
30.0
```

In [26]:
```
print(c+d)
print(c-d)
print(c*d)
print(c/d)
```

```
(30+50j)
(-10-10j)
(-400+700j)
(0.6153846153846154+0.07692307692307691j)
```

Data Type Casting - Convert one Dataype to Other Dataype

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

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

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

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

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

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

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

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

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 1
----> 1 int(1+2j)

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

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

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

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[33], line 1
----> 1 float(10,20)

TypeError: float expected at most 1 argument, got 2
```

In [34]:
```
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]:
```
float(True)
```

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

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

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

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

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

In [38]:
```
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]:
```
str(8)
```

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

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

Out[40]: '8.8'

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

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

In [42]: str(True)

Out[42]: 'True'

In [48]: 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]: str()

Out[50]: ''

In [51]: bool()

Out[51]: False

In [52]: bool(10)

Out[52]: True

In [53]: bool(9.0)

Out[53]: True
```

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

Out[54]: True
```

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

Out[55]: False
```

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

Out[56]: True
```

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

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

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

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

```
In [59]: 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]: complex(10.20,30.8)

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

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

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

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

Out[63]: 0j
```

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

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

```
In [65]: 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]: complex(True,False)

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

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

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

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

Out[9]: str
```

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

Out[10]: 'H'
```

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

Out[11]: 'E'
```

Forward Index

In [15]:

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

```
H
E
L
L
O

P
Y
T
H
O
N
```

Backward Index

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

```
N
O
H
T
Y
P

O
L
L
E
H
```

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

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

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

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

Slicing :

Forward Slicing

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

         PYTHON
```

Step Slicing

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

Out[32]: 'HELLO PYTHON'

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

Out[33]: 'HLPH'

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

Out[36]: 'elpt'

In [37]: str2

Out[37]: 'hellopython'

In [39]: str2[::2]

Out[39]: 'hloyhn'

In [40]: str2[::3]

Out[40]: 'hlyo'
```

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

```
HARI KARTHIK
```

> Python Basic Operator

> > Arithematic Operator

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

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

Out[7]:
```
15
```

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

Out[8]:
```
5
```

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

Out[9]:
```
50
```

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

Out[10]:
```
2.0
```

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

Out[11]: 2
```

```
In [12]: x1%y1

Out[12]: 0
```

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

Out[15]: 100000
```

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

Out[17]: 27
```

Assignment Operator

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

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

```
In [25]: y

Out[25]: 4
```

```
In [26]: y+=2
         y
Out[26]: 6

In [27]: y+=2
         y
Out[27]: 8

In [28]: y*=2

In [29]: y
Out[29]: 16

In [30]: y-=2
         y
Out[30]: 14

In [31]: y/=2

In [32]: y
Out[32]: 7.0

In [33]: y//=2
         y
Out[33]: 3.0
```

Unary Operator

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

```
5 5 -3 -3
```

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

```
5 -5 -3 3
```

Relational Operator

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

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

```
Out[35]: True
```

```
In [36]: a>b
Out[36]: False
```

```
In [37]: a==b
Out[37]: False
```

```
In [39]: a!=b
Out[39]: True
```

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

```
In [41]: a==b
Out[41]: True
```

> > Logical Operator

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

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

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

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

Out[48]: True

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

Out[50]: False

In [51]: x=False
         x

Out[51]: False

In [52]: not x

Out[52]: True
```

> Python Bitwise Operator

> > Complement Operator (~) - Reverse of Binary format

```
In [53]: ~12

Out[53]: -13

In [54]: ~46

Out[54]: -47
```

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

```
In [60]: bin(25)
Out[60]: '0b11001'
```

```
In [61]: 0b11001
Out[61]: 25
```

```
In [62]: oct(25)
Out[62]: '0o31'
```

```
In [64]: hex(25)
Out[64]: '0x19'
```