In [2]:
```
import sys
sys.version+
```

Out[2]:
```
'3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]'
```

python variable = identifier = object

In [5]:
```
a=5
a                ## a=var, 5=value
```

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

PYTHON RULES:

In [7]:
```
var=6
var
```

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

In [13]:
```
v_=23     ##Special symbols are not allowed.
            ##UnderScore is allowed
```

Keywords:

In [16]:
```
import keyword
keyword.kwlist
```

Out[16]:
```
['False',
 'None',
 'True',
 '__peg_parser__',
 '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']
```

In [19]:
```
len(keyword.kwlist)
```

Out[19]:
```
36
```

In [21]:
```
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True', '__peg_parser__', 'and', 'as', 'assert', 'async',
'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'excep
t', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambd
a', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'wit
h', 'yield']
```

PYTHON DATA TYPES

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

Out[1]: 5

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [2], in <cell line: 1>()
----> 1 type()

TypeError: type() takes 1 or 3 arguments
```

In [3]:
```
type(i)                    ###arg should be passed
```

Out[3]: int

In [6]:
```
i
```

Out[6]: 5

In [9]:
```
f=5.0
type(f)
```

Out[9]: float

In [10]:
```
f1=e0
f1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [10], in <cell line: 1>()
----> 1 f1=e0
      2 f1

NameError: name 'e0' is not defined
```

In [19]:
```
True * False           ##t=1,f=0 in backend
```

Out[19]: 0

In [18]:
```
False | True           ###f=0,
```

Out[18]: True

In [20]:
```
False + True
```

Out[20]: 1

In [26]:
```
c=10 + 15j
c.real
c.imag                  ## giving the second c.imag as o/p
print(c.real)           ## so can use print stmt
print(c.imag)
```

```
10.0
15.0
```

TYPE CASTING

In [1]:
```
## Type casting---- convert one data type to another
int(1.4)          ##1.4 is float but need interger value
```

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

In [3]:
```
int(2.4,5.5)    ### type casting take only 1 arg
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [3], in <cell line: 1>()
----> 1 int(2.4,5.5)

TypeError: 'float' object cannot be interpreted as an integer
```

In [9]:
```
int(True,False)    ## Bcz string cant convert too int
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [9], in <cell line: 1>()
----> 1 int(True,False)

TypeError: int() can't convert non-string with explicit base
```

In [8]:
```
int(1o+15j)           ###NO pOssibilities:complex - int
```

```
  Input In [8]
    int(1o+15j)           ###NO pOssibilities:complex - int
         ^
SyntaxError: invalid syntax
```

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

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

All other DT to Float

In [1]:
```
1 float(200)
```

Out[1]:
```
200.0
```

In [12]:
```
1 float(10+15j)         ### not possible
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [12], in <cell line: 1>()
----> 1 float(10+15j)

TypeError: can't convert complex to float
```

In [4]:
```
1 f2=2.2*e2
2 f2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [4], in <cell line: 1>()
----> 1 f2=2.2*e2
      2 f2

NameError: name 'e2' is not defined
```

In [13]:
```
1 float(True)
```

Out[13]:
```
1.0
```

In [14]:
```
1 float("10")      ## num string is possible
```

Out[14]:
```
10.0
```

In [15]:
```
1 float("sneha")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [15], in <cell line: 1>()
----> 1 float("sneha")

ValueError: could not convert string to float: 'sneha'
```

All other DT to Sting:

In [16]:
```
1 str(9)
```

Out[16]:
```
'9'
```

In [17]:
```
1 str(8.9)
```

Out[17]:
```
'8.9'
```

In [18]:
```
1 str(1+2J)
```

Out[18]:
```
'(1+2j)'
```

In [19]:
```
1 str(True)
```

Out[19]:
```
'True'
```

In [24]:
```
1 print(bool(10))
```

```
True
```

In [25]:
```
1 bool()         ## false-bcz emty by default
```

Out[25]:
```
False
```

In [26]:
```
1 complex(10)
```

Out[26]:
```
(10+0j)
```

In [28]:
```
1 complex(20, 10)
```

Out[28]:
```
(20+10j)
```

In [29]:
```
1 complex(20, 10,30)     ### 3-arg are not allowed
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [29], in <cell line: 1>()
----> 1 complex(20, 10,30)

TypeError: complex() takes at most 2 arguments (3 given)
```

In [30]:
```
1 complex(20.2, 10)
```

Out[30]:
```
(20.2+10j)
```

In [32]:
```
1 print(complex(True))
2 print(complex(10))
```

```
(1+0j)
(10+0j)
```

In [11]:
```
1 f2=1e4
2 f2                  ### e is allowed for float  &&&&&&&&& add the zeros
```

Out[11]:
```
10000.0
```

In [7]:
```
1 type(f2)
```

Out[7]:
```
float
```

In [15]:
```
p=20
q=20
r=20
print(id(r))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [15], in <cell line: 4>()
      2 q=20
      3 r=20
----> 4 print(id(r))

TypeError: 'int' object is not callable
```

DATA STRUCTURES

In [16]:
```
str="sneha"
str
```

Out[16]: 'sneha'

In [17]:
```
str[0]
```

Out[17]: 's'

In [18]:
```
str[1]
```

Out[18]: 'n'

In [19]:
```
print(str[0])
print(str[1])
print(str[2])
```

```
s
n
e
```

In [20]:
```
str[-1]   ## backward indexing
```

Out[20]: 'a'

In [22]:
```
print(str[-1])
print(str[-2])
print(str[-3])
print(str[-4])
print(str[-5])   #### reversing the string
```

```
a
h
e
n
s
```

In [24]:
```
1 len(str) ### ocunt of the indexes
```

Out[24]: 5

Slicing: defines wirh ":" ||| can find the any elements of a string of any index. forward slicing----------> back ward slicing--------> step slicing--------->

In [26]:
```
1 str
```

Out[26]: 'sneha'

In [27]:
```
1 str[1:1]
```

Out[27]: ''

In [28]:
```
1 str[1:3]
```

Out[28]: 'ne'

In [29]:
```
1 str[0:4]    ### strat from indx 0 upto 3
```

Out[29]: 'sneh'

Step Slicing: [strt index, last index, and steps]

In [35]:
```
1 s = "hellopython"
2 s[0:2]
3 s[0:10:3]
```

Out[35]: 'hlyo'

In [36]:
```
1 s = "hello_python"
2 s[0:2]
3 s[1:10:5]
```

Out[36]: 'ep'

In [38]:
```
1 s='helo_python'
2 s[::3]
```

Out[38]: 'hoyo'

In [39]:
```
1 s='helo_python'
2 s[::-2]
```

Out[39]: 'nhy_lh'

OPERATORS

"'Assignment op arthimatic op relational op logical op unary op"'

In [3]:
```
x1=10
y1=5   ### x1,y1=10,5 can be written
```

In [8]:
```
a,b=7  ### bcz only 1 value is declared. declared fro a
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [8], in <cell line: 1>()
----> 1 a,b=7

TypeError: cannot unpack non-iterable int object
```

No of var = no of values

In [16]:
```
a=7
b=7
```

In [17]:
```
print(a)
print(b)  ### get both values as 7
```

```
7
7
```

Arthematic op

BIT WISE Op

In [18]:
```
##complement op
## use tilt symboll(~)
```

In [20]:
```
~35    ## binary num 35=00100011
        ##bit wise not =11011100------> 2's comple
```

Out[20]:
```
-36
```

Number System

In [22]:
```
#binary number[0b]--base2(0,1)-----0b is a number sys
#octal num [0o]---base2(0-7)
# decimal num [0x]---base10(0-9)
# hexadecimal num []---base(0-9,A-F)
```

In [ ]:
```

```