In [4]:

```
import sys
sys.version
```

Out[4]:

```
'3.11.7 | packaged by Anaconda, Inc. | (main, Dec 15 2023, 18:05:47) [MSC
v.1916 64 bit (AMD64)]'
```

In [5]:

```
import math
math.sqrt(100)
```

Out[5]:

```
10.0
```

In [6]:

```
round(math.sqrt(100))
```

Out[6]:

```
10
```

In [6]:

```
'welcom
```

```
  Cell In[6], line 1
    'welcom
    ^
SyntaxError: unterminated string literal (detected at line 1)
```

PYTHON IDENTIFIER

In [8]:

```
A = 33
B = 44
C = 55
D = 66

B
```

Out[8]:

```
44
```

In [10]:

```
A, B, C, D
```

Out[10]:

```
(33, 44, 55, 66)
```

In [9]:

```
print(A)
print(B)
print(C)
print(D)
```

```
33
44
55
66
```

In [12]:

```
A = 34
A
```

Out[12]:

```
34
```

In [13]:

```
id(A)
```

Out[13]:

```
140735368906568
```

In [14]:

```
NIT = 35
nit
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call las
t)
Cell In[14], line 2
      1 NIT = 35
----> 2 nit

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

In [17]:

```
abc = 67
ABC
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call las
t)
Cell In[17], line 2
      1 abc = 67
----> 2 ABC

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

In [18]:

```
python = 4
python
```

Out[18]:

```
4
```

In [19]:

```
JAVA = 5
JAVA
```

Out[19]:

```
5
```

In [20]:

```
Jupyter = 67
Jupyter
```

Out[20]:

```
67
```

In [21]:

```
nit* = 54
```

```
  Cell In[21], line 1
    nit$ = 54
       ^
SyntaxError: invalid syntax
```

In [22]:

```
nit_ = 78
nit_
```

Out[22]:

```
78
```

In [24]:

```
train_ = 80
train_
```

Out[24]:

```
80
```

In [25]:

```
if = 78
```

```
  Cell In[25], line 1
    if = 78
       ^
SyntaxError: invalid syntax
```

In [29]:

```
import keyword
keyword.kwlist
```

Out[29]:

```
['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']
```

In [28]:

```
print(len(keyword.kwlist))
```

```
35
```

In [32]:

```
local = 78
local
```

Out[32]:

```
78
```

```
In [33]: nonlocal = 89
```

```
  Cell In[33], line 1
    nonlocal = 89
             ^
SyntaxError: invalid syntax
```

```
In [31]: import = 67
```

```
  Cell In[31], line 1
    import = 67
           ^
SyntaxError: invalid syntax
```

```
In [36]: a = 6
         b = 7

         print(a)
         print(b)
```

```
6
7
```

```
In [37]: 1nit = 56
```

```
  Cell In[37], line 1
    1nit = 56
    ^
SyntaxError: invalid decimal literal
```

```
In [38]: nit1 = 67
         nit1
```

```
Out[38]: 67
```

CASE SENSITIVE

In [40]:

```
JOHN = 6
john
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Cell In[40], line 2
      1 JOHN = 6
----> 2 john

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

UPPER CASE & LOWER CASE

In [41]:

```
JOHN = 56
JOHN
```

Out[41]:

```
56
```

In [42]:

```
smith = 78
smith
```

Out[42]:

```
78
```

keyword never assigned as an identifier

In [43]:

```
if = 78
```

```
  Cell In[43], line 1
    if = 78
       ^
SyntaxError: invalid syntax
```

In [44]:

```
else = 89
```

```
  Cell In[44], line 1
    else = 89
    ^
SyntaxError: invalid syntax
```

```
In [46]:  for = 78
```

```
  Cell In[46], line 1
    for = 78
        ^
SyntaxError: invalid syntax
```

```
In [47]:  FOR = 78
          FOR
```

```
Out[47]:  78
```

import keyword keyword.kwlist == use this code to display inbuild keywords or reserved words

special character is not allowed except underscore(_)

```
In [48]:  jupyter_ = 78
          jupyter_
```

```
Out[48]:  78
```

```
In [49]:  nit@ = 78
```

```
  Cell In[49], line 1
    nit@ = 78
         ^
SyntaxError: invalid syntax
```

```
In [50]:  ni#
```

```
--------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Cell In[50], line 1
----> 1 ni

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

```
In [52]:  nit! =  0
```

```
  Cell In[52], line 1
    nit! =  0
       ^
SyntaxError: invalid syntax
```

identifier never start with digit (number)

```
In [54]:  1hi = 67
```

```
  Cell In[54], line 1
    1hi = 67
    ^
SyntaxError: invalid decimal literal
```

```
In [55]:  hi1 = 67
          hi1
```

```
Out[55]:  67
```

```
In [56]:  hi hello = 78
```

```
  Cell In[56], line 1
    hi hello = 78
       ^
SyntaxError: invalid syntax
```

```
In [57]:  __name__
```

```
Out[57]:  '__main__'
```

```
In [1]:  aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa = 78
         aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Out[1]:  78
```

```
In [2]:  true
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call las
t)
Cell In[2], line 1
----> 1 true

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

```
In [3]:  True
```

```
Out[3]: True
```

```
In [4]:  false
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call las
t)
Cell In[4], line 1
----> 1 false

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

```
In [5]:  False
```

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

```
In [7]:  none
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call las
t)
Cell In[7], line 1
----> 1 none

NameError: name 'none' is not defined
```

```
In [8]:  None
```

```
In [ ]:  a = 67 # a is object & 67 is value
```

python basic or inbuild data type

```
In [10]:  i = 34
          i

Out[10]:  34
```

```
In [11]:  type(i)

Out[11]:  int
```

```
In [12]:  print(type(i))

          <class 'int'>
```

```
In [13]:  a, b = 56

          ---------------------------------------------------------------------------
          --
          TypeError                                 Traceback (most recent call las
          t)
          Cell In[13], line 1
          ----> 1 a, b = 56

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

```
In [14]:  a, b = 56, 67
```

```
In [15]:  print(a)
          print(b)

          56
          67
```

how to swap 2 variable

```
In [16]:  a, b

Out[16]:  (56, 67)
```

```
In [19]:  a, b = b, a
In [18]:  print(0)
           print(1)
           print(1)
           print(2)
           print(3)

           0
           1
           1
           2
           3
```

```
In [20]:  a

Out[20]:  67
```

```
In [21]:  b

Out[21]:  56
```

```
In [22]:  i

Out[22]:  34
```

```
In [23]:  id(i)

Out[23]:  140735368906568
```

```
In [24]:  1i = 34

             Cell In[24], line 1
               1i = 34
               ^
           SyntaxError: invalid decimal literal
```

```
In [25]:  i

Out[25]:  34
```

```
In [26]:  i1 = 34
           i1

Out[26]:  34
```

```
In [27]:  id(i)

Out[27]:  140735368906568
```

```
In [28]:  id(i1)

Out[28]:  140735368906568
```

```
In [29]:  i2= 45
          i2

Out[29]:  45
```

```
In [30]:  i

Out[30]:  34
```

```
In [31]:  i1

Out[31]:  34
```

```
In [32]:  i2

Out[32]:  45
```

```
In [33]:  print(id(i))
          print(id(i1))
          print(id(i2))

          140735368906568
          140735368906568
          140735368906920
```

```
In [36]:  i == i1

Out[36]:  True
```

```
In [37]:  i == i2

Out[37]:  False
```

```
In [38]:  i != i2

Out[38]:  True
```

```
In [39]:  i != i1

Out[39]:  False
```

```
In [40]:  True

Out[40]:  True
```

```
In [41]:  True + True

Out[41]:  2
```

```
In [42]:  True + False

Out[42]:  1
```

```
In [43]:  int(True)

Out[43]:  1
```

```
In [44]:  int(False)

Out[44]:  0
```

```
In [45]:  False + False + True

Out[45]:  1
```

```
In [46]:  None + False
```

```
---------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
Cell In[46], line 1
----> 1 None + False

TypeError: unsupported operand type(s) for +: 'NoneType' and 'bool'
```

```
In [47]:  1 + 'summer'
```

```
---------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
Cell In[47], line 1
----> 1 1 + 'summer'

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [49]:

```
fl = 1.2
fl
```

Out[49]:

```
1.2
```

In [50]:

```
print(type(fl))
```

```
<class 'float'>
```

In [51]:

```
f = 1e0
f
```

Out[51]:

```
1.0
```

In [52]:

```
f1 = 1e1
f1
```

Out[52]:

```
10.0
```

In [54]:

```
f2 = 1e2
f2
```

Out[54]:

```
100.0
```

In [56]:

```
f3 = 1e3
f3
```

Out[56]:

```
1000.0
```

In [57]:

```
f4 = 4e4
f4
```

Out[57]:

```
40000.0
```

In [58]:

```
type(f4)
```

Out[58]:

```
float
```

In [59]:

```
f5 = m5
f5
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[59], line 1
----> 1 f5 = m5
      2 f5

NameError: name 'm5' is not defined
```

BOOL

In [148]:

```
True
```

Out[148]:

```
True
```

In [149]:

```
False
```

Out[149]:

```
False
```

In [150]:

```
True + True
```

Out[150]:

```
2
```

In [151]:

```
True + False
```

Out[151]:

```
1
```

##### complex (a + bj) # a - real part b - imaginary part j -- squre root -1

In [152]:

```
c = 1 + 2j
c
```

Out[152]:

```
(1+2j)
```

In [154]:

```
print(type(c))
```

```
<class 'complex'>
```

In [155]:

```
c
```

Out[155]:

```
(1+2j)
```

```
In [156]:  c.real

Out[156]:  1.0

In [164]:  c3 = 3.7 + 6.7j
           c3

Out[164]:  (3.7+6.7j)

In [168]:  c4 = True + 0j
           c4

Out[168]:  (1+0j)

In [157]:  c.imag

Out[157]:  2.0

In [159]:  d = 4 + 5j
           d

Out[159]:  (4+5j)

In [160]:  c + d

Out[160]:  (5+7j)

In [161]:  c

Out[161]:  (1+2j)

In [162]:  d

Out[162]:  (4+5j)

In [163]:  c + d

Out[163]:  (5+7j)
```

string

In [171]:

```
s = nit
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Cell In[171], line 1
----> 1 s = nit

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

In [173]:

```
s = 'nit'
s
```

Out[173]:

```
'nit'
```

In [177]:

```
print(type(s))
```

```
<class 'str'>
```

In [178]:

```
s1 = "nit"
s1
```

Out[178]:

```
'nit'
```

In [179]:

```
s2 = ''' nit '''
s2
```

Out[179]:

```
' nit '
```

In [181]:

```
s3 = 'naresh
        it
            technology'
s3
```

```
  Cell In[181], line 1
    s3 = 'naresh
         ^
SyntaxError: unterminated string literal (detected at line 1)
```

```
In [182]:   s4 = "naresh
                    it
                        technology"
            s4

              Cell In[182], line 1
                s4 = "naresh
                     ^
            SyntaxError: unterminated string literal (detected at line 1)
```

```
In [183]:   s5 = '''naresh
                    it
                        technology'''
            s5

Out[183]:   'naresh \n        it \n            technology'
```

```
In [184]:   i_, f_, c_, b_, s_ = 29, 34.8, 1 + 2j, False, 'nit'
```

```
In [186]:   f_

Out[186]:   34.8
```

```
In [189]:   p1 = p2 = 44
            print(p1)
            print(p2)

            44
            44
```

```
In [190]:   p1 == p2

Out[190]:   True
```

```
In [191]:   p1

Out[191]:   44
```

```
In [193]:   import sys
            print(sys.getsizeof(p1))

            28
```

```
In [195]:   str = 'nareshit'
            str

Out[195]:   'nareshit'
```

```
In [196]:  str[0]

Out[196]:  'n'

In [197]:  str[-1]

Out[197]:  't'

In [198]:  str

Out[198]:  'nareshit'

In [199]:  str[10]

           ---------------------------------------------------------------------------
           --
           IndexError                                Traceback (most recent call las
           t)
           Cell In[199], line 1
           ----> 1 str[10]

           IndexError: string index out of range

In [200]:  str

Out[200]:  'nareshit'

In [201]:  str(0)

           ---------------------------------------------------------------------------
           --
           TypeError                                 Traceback (most recent call las
           t)
           Cell In[201], line 1
           ----> 1 str(0)

           TypeError: 'str' object is not callable

In [202]:  str

Out[202]:  'nareshit'

In [203]:  str[3:8]

Out[203]:  'eshit'
```

```
In [204]:  s

Out[204]: 'nit'
```

```
In [205]:  s1

Out[205]: 'nit'
```

```
In [206]:  s + s1

Out[206]: 'nitnit'
```

type casting || type conversion

chnage 1 data type to other datatype

```
In [207]:  5.5

Out[207]: 5.5
```

```
In [208]:  int(5.5) # cast float argument to int argument

Out[208]: 5
```

```
In [209]:  int(5.5, 4.3)

           ---------------------------------------------------------------------------
           TypeError                                 Traceback (most recent call las
           t)
           Cell In[209], line 1
           ----> 1 int(5.5, 4.3)

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

```
In [210]:  int(4.3)

Out[210]: 4
```

```
In [211]:  int(True) # bool to int

Out[211]: 1
```

In [215]:

```
b = int(False)
type(b)
```

Out[215]:

```
int
```

In [216]:

```
int(10+20j)
```

```
---------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
Cell In[216], line 1
----> 1 int(10+20j)

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

In [217]:

```
int(2.6 + 5j)
```

```
---------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
Cell In[217], line 1
----> 1 int(2.6 + 5j)

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

In [218]:

```
int('10')
```

Out[218]:

```
10
```

In [219]:

```
int('ten')
```

```
---------------------------------------------------------------------------
--
ValueError                                Traceback (most recent call las
t)
Cell In[219], line 1
----> 1 int('ten')

ValueError: invalid literal for int() with base 10: 'ten'
```

In [220]:

```
float(10)
```

Out[220]:

```
10.0
```

```
In [221]:    float(1 + 2j )

             ---------------------------------------------------------------------------
             --
             TypeError                                 Traceback (most recent call las
             t)
             Cell In[221], line 1
             ----> 1 float(1 + 2j )

             TypeError: float() argument must be a string or a real number, not 'compl
             ex'
```

```
In [222]:    float(True)

Out[222]:    1.0
```

```
In [223]:    float('1')

Out[223]:    1.0
```

```
In [224]:    complex(10)

Out[224]:    (10+0j)
```

```
In [225]:    complex(10, 20)

Out[225]:    (10+20j)
```

```
In [226]:    complex(10, 20, 30)

             ---------------------------------------------------------------------------
             --
             TypeError                                 Traceback (most recent call las
             t)
             Cell In[226], line 1
             ----> 1 complex(10, 20, 30)

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

```
In [227]:    complex(1.2, 3.4)

Out[227]:    (1.2+3.4j)
```

```
In [228]:    complex(True, False) #bool to complex

Out[228]:    (1+0j)
```

```
In [229]:  complex(False, False)

Out[229]:  0j
```

```
In [233]:  complex('1')

Out[233]:  (1+0j)
```

```
In [234]:  complex('1', '2')

           ---------------------------------------------------------------------------
           --
           TypeError                                 Traceback (most recent call las
           t)
           Cell In[234], line 1
           ----> 1 complex('1', '2')

           TypeError: complex() can't take second arg if first is a string
```

```
In [237]:  complex('1',2)

           ---------------------------------------------------------------------------
           --
           TypeError                                 Traceback (most recent call las
           t)
           Cell In[237], line 1
           ----> 1 complex('1',2)

           TypeError: complex() can't take second arg if first is a string
```

```
In [238]:  bool(1)

Out[238]:  True
```

```
In [239]:  bool(2.3)

Out[239]:  True
```

```
In [240]:  bool(1 + 2j )

Out[240]:  True
```

```
In [241]:  bool()

Out[241]:  False
```

```
In [242]:  bool( )

Out[242]:  False
```

```
In [243]:  bool('3')

Out[243]:  True
```

```
In [244]:  bool('nit')

Out[244]:  True
```

convert non type to integer

```
In [62]:  print(123123123123123123123123123123123123123123123123 + 1)

           123123123123123123123123123123123123123123123124
```

```
In [64]:  p = print(123123123123123123123123123123123123123123123123 + 1)
           print(type(p))

           123123123123123123123123123123123123123123123124
           <class 'NoneType'>
```

```
In [65]:  print(123123123123123123123 + 1)

           123123123123123123124
```

```
In [67]:  p1 = print(123123123123123123123 + 1)
           type(p1)

           123123123123123123124

Out[67]:   NoneType
```

- In this above code, print(123123123123123123123 + 1) prints the result to the console, and the print function itself returns None.
- Therefore, when you check the type of p1, it will be <class 'NoneType'>.

- If you want to store the result of the expression 123123123123123123123 + 1 in a variable, you can do it like this:

In [68]:

```python
result = 123123123123123123123 + 1
print(result)
print(type(result))
```

```
123123123123123123124
<class 'int'>
```

FLOAT

In [69]:

```python
4.
```

Out[69]:

```
4.0
```

In [70]:

```python
type(4.)
```

Out[70]:

```
float
```

In [71]:

```python
.2
```

Out[71]:

```
0.2
```

In [72]:

```python
type(.2)
```

Out[72]:

```
float
```

In [73]:

```python
.4e3
```

Out[73]:

```
400.0
```

In [74]:

```python
 4.2e-4 # introuduce to chatgpt
```

Out[74]:

```
0.00042
```

STRING

In [75]:

```python
''
```

Out[75]:

```
''
```

In [78]:

```python
es = ''
type(es)
```

Out[78]:

```
str
```

```
In [76]:  ""
```

```
Out[76]: ''
```

```
In [77]:  ''''''
```

```
Out[77]: ''
```

```
In [80]:  print("This string contains a single quote (') character.")
```

```
This string contains a single quote (') character.
```

```
In [81]:  print('This string contains a double quote (") character.')
```

```
This string contains a double quote (") character.
```

```
In [79]:  print('This string contains a single quote (') character.')
```

```
  Cell In[79], line 1
    print('This string contains a single quote (') character.')
                                                              ^
SyntaxError: unterminated string literal (detected at line 1)
```

```
In [82]:  print('This string contains a single quote (\') character.')

          '''
          Specifying a backslash in front of the quote character in a string “escape
          and causes Python to suppress its usual special meaning.
          It is then interpreted simply as a literal single quote character:
          The same works in a string delimited by double quotes as well:
          '''
```

```
This string contains a single quote (') character.
```

```
In [85]:  print('This string contains a single quote (\') character.')
```

```
This string contains a single quote (') character.
```

```
In [87]:  print("This string contains a double quote (\') character.")
```

```
This string contains a double quote (') character.
```

In [116]:

```
print('a
```

```
  Cell In[116], line 1
    print('a
          ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [147]:

```
>>> print('a\
b\
c')
```

```
abc
```

In [245]:

```
print('*\
*\
*')
```

```
***
```

In [131]:

```
print('*\
*\
*')

print('#\
#\
###')

print('*\
***\
***')
```

```
***
#####
*******
```

In [145]:

```
print('naresh\it') #To include a literal backslash in a string, escape it
```

```
naresh\it
```

In [127]:

```
print("naresh\\it")
```

```
naresh\it
```

In [134]:

```
print('naresh\nit')
```

```
naresh
it
```

- A raw string literal is preceded by r or R, which specifies that escape sequences in the associated string are not translated.
- The backslash character is left in the string:

list

In [3]:

```
l = []
l
```

Out[3]:

```
[]
```

In [4]:

```
type(l)
```

Out[4]:

```
list
```

In [5]:

```
l.append(10)
```

In [6]:

```
l
```

Out[6]:

```
[10]
```

In [7]:

```
l.append(20)
```

In [254]:

```
l
```

Out[254]:

```
[10, 20]
```

In [8]:

```
l.append(3.4)
l.append(True)
l.append('nit')
l.append(1 + 2j)
```

In [9]:

```
l
```

Out[9]:

```
[10, 20, 3.4, True, 'nit', (1+2j)]
```

In [10]:

```
len(l)
```

Out[10]:

```
6
```

In [12]:

```
l1 = l.copy()
```

In [13]:

```
l1
```

Out[13]:

```
[10, 20, 3.4, True, 'nit', (1+2j)]
```

In [14]:

```
len(l1)
```

Out[14]:

```
6
```

In [15]:

```
l == l1
```

Out[15]:

```
True
```

In [16]:

```
l
```

Out[16]:

```
[10, 20, 3.4, True, 'nit', (1+2j)]
```

In [17]:

```
l1
```

Out[17]:

```
[10, 20, 3.4, True, 'nit', (1+2j)]
```

In [18]:

```
print(id(l))
print(id(l1))
```

```
2782314677440
2782314478080
```

In [19]:

```
l.append(10)
```

In [20]:

```
l
```

Out[20]:

```
[10, 20, 3.4, True, 'nit', (1+2j), 10]
```

In [21]:

```
l1
```

Out[21]:

```
[10, 20, 3.4, True, 'nit', (1+2j)]
```

In [22]:

```
l== l1
```

Out[22]:

```
False
```

In [23]:

```
l1
```

Out[23]:

```
[10, 20, 3.4, True, 'nit', (1+2j)]
```

```
In [24]:  l1.clear()
```

```
In [25]:  l1
```

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

```
In [26]:  id(l1)
```

```
Out[26]: 2782314478080
```

```
In [273]:  len(l)
```

```
Out[273]: 7
```

```
In [274]:  len(l1)
```

```
Out[274]: 0
```

```
In [27]:  del l1
```

```
In [28]:  l1
```

```
--------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[28], line 1
----> 1 l1

NameError: name 'l1' is not defined
```

```
In [29]:  l
```

```
Out[29]: [10, 20, 3.4, True, 'nit', (1+2j), 10]
```

```
In [30]:  l.count()
```

```
--------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[30], line 1
----> 1 l.count()

TypeError: list.count() takes exactly one argument (0 given)
```

```
In [32]:   l.count(3.4)

Out[32]:   1
```

```
In [280]:  l

Out[280]:  [10, 20, 3.4, True, 'nit', (1+2j), 10]
```

```
In [281]:  l.insert(7,'datascience')
```

```
In [282]:  l

Out[282]:  [10, 20, 3.4, True, 'nit', (1+2j), 10, 'datascience']
```

```
In [283]:  l.insert(6,'ai')
```

```
In [284]:  l

Out[284]:  [10, 20, 3.4, True, 'nit', (1+2j), 'ai', 10, 'datascience']
```

```
In [285]:  l.remove('ai')
```

```
In [286]:  l

Out[286]:  [10, 20, 3.4, True, 'nit', (1+2j), 10, 'datascience']
```

```
In [287]:  l.remove(3.4, True)

           ---------------------------------------------------------------------------
           --
           TypeError                                 Traceback (most recent call las
           t)
           Cell In[287], line 1
           ----> 1 l.remove(3.4, True)

           TypeError: list.remove() takes exactly one argument (2 given)
```

```
In [288]:  l

Out[288]:  [10, 20, 3.4, True, 'nit', (1+2j), 10, 'datascience']
```

```
In [289]:  l.pop()

Out[289]:  'datascience'
```

In [290]:
```
l
```

Out[290]:
```
[10, 20, 3.4, True, 'nit', (1+2j), 10]
```

In [291]:
```
l.pop(True, (1+2j))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[291], line 1
----> 1 l.pop(True, (1+2j))

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

In [292]:
```
l.pop(True)
```

Out[292]:
```
20
```

In [293]:
```
l
```

Out[293]:
```
[10, 3.4, True, 'nit', (1+2j), 10]
```

l.pop(True) l.pop(3.4)

In [295]:
```
l.pop(True)
l.remove(3.4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[295], line 2
      1 l.pop(True)
----> 2 l.remove(3.4)

ValueError: list.remove(x): x not in list
```

In [296]:
```
l
```

Out[296]:
```
[10, 'nit', (1+2j), 10]
```

In [297]:
```
l.append(True)
l.append('hi')
l.append([1,2, 3, 4.5])
```

```
In [298]:   l

Out[298]:  [10, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]

In [299]:   len(l)

Out[299]:  7

In [300]:   l.count('nit')

Out[300]:  1

In [301]:   l

Out[301]:  [10, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]

In [302]:   len(l)

Out[302]:  7

In [303]:   l[:]

Out[303]:  [10, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]

In [304]:   l

Out[304]:  [10, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]

In [305]:   l[1:7]

Out[305]:  ['nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]

In [306]:   l

Out[306]:  [10, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]

In [307]:   l[0:5]

Out[307]:  [10, 'nit', (1+2j), 10, True]

In [308]:   l[0:0]

Out[308]:  []

In [309]:   l

Out[309]:  [10, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]
```

```
In [310]:  l[0]

Out[310]:  10
```

```
In [311]:  l[0] = 100
```

```
In [312]:  l

Out[312]:  [100, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]
```

```
In [313]:  l[1]

Out[313]:  'nit'
```

```
In [314]:  l[1][0]

Out[314]:  'n'
```

```
In [315]:  l[1]

Out[315]:  'nit'
```

```
In [316]:  print(l[1][0])
           print(l[1][1])
           print(l[1][2])

           n
           i
           t
```

```
In [318]:  l

Out[318]:  [100, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]
```

```
In [319]:  l[2:10]

Out[319]:  [(1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]
```

```
In [320]:  l

Out[320]:  [100, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]
```

```
In [321]:  l[3:-1]

Out[321]:  [10, True, 'hi']
```

```
In [322]:  l

Out[322]:  [100, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]

In [323]:  l[-1]

Out[323]:  [1, 2, 3, 4.5]

In [324]:  l[-2]

Out[324]:  'hi'

In [325]:  l

Out[325]:  [100, 'nit', (1+2j), 10, True, 'hi', [1, 2, 3, 4.5]]

In [326]:  l[-5:-1]

Out[326]:  [(1+2j), 10, True, 'hi']

  In [2]:  l

           ---------------------------------------------------------------------------
           --
           NameError                                 Traceback (most recent call las
           t)
           Cell In[2], line 1
           ----> 1 l

           NameError: name 'l' is not defined


 In [33]:  l

Out[33]:  [10, 20, 3.4, True, 'nit', (1+2j), 10]

 In [34]:  l[-1]

Out[34]:  10

 In [35]:  l[-1] = 25

 In [36]:  l

Out[36]:  [10, 20, 3.4, True, 'nit', (1+2j), 25]
```

```
In [38]:  l[:]

Out[38]: [10, 20, 3.4, True, 'nit', (1+2j), 25]
```

```
In [39]:  l[1:5]

Out[39]: [20, 3.4, True, 'nit']
```

```
In [40]:  l

Out[40]: [10, 20, 3.4, True, 'nit', (1+2j), 25]
```

```
In [41]:  l[::-1]

Out[41]: [25, (1+2j), 'nit', True, 3.4, 20, 10]
```

```
In [42]:  rev_list = l[::-1]
```

```
In [43]:  rev_list

Out[43]: [25, (1+2j), 'nit', True, 3.4, 20, 10]
```

```
In [44]:  l

Out[44]: [10, 20, 3.4, True, 'nit', (1+2j), 25]
```

```
In [45]:  l.reverse()
```

```
In [46]:  l

Out[46]: [25, (1+2j), 'nit', True, 3.4, 20, 10]
```

```
In [47]:  l.sort()

-------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
Cell In[47], line 1
----> 1 l.sort()

TypeError: '<' not supported between instances of 'complex' and 'int'
```

```
In [49]:  l3  = [20, 5, 500, 17]
```

In [50]:

```
l3.sort()
```

In [51]:

```
l3
```

Out[51]:

```
[5, 17, 20, 500]
```

In [52]:

```
l3.sort(reverse=True)
l3
```

Out[52]:

```
[500, 20, 17, 5]
```

In [53]:

```
l4 = ['a', 'zebra', 'nit', 'cricket', 'football']
```

In [54]:

```
l4.sort()
l4
```

Out[54]:

```
['a', 'cricket', 'football', 'nit', 'zebra']
```

In [55]:

```
l4.sort(reverse=True)
l4
```

Out[55]:

```
['zebra', 'nit', 'football', 'cricket', 'a']
```

In [56]:

```
l
```

Out[56]:

```
[25, (1+2j), 'nit', True, 3.4, 20, 10]
```

In [57]:

```
25 in l
```

Out[57]:

```
True
```

In [58]:

```
250 in l
```

Out[58]:

```
False
```

In [59]:

```
l
```

Out[59]:

```
[25, (1+2j), 'nit', True, 3.4, 20, 10]
```

In [60]:

```
l[:] # ENTIRE LIST
```

Out[60]:

```
[25, (1+2j), 'nit', True, 3.4, 20, 10]
```

```
In [61]:  l[:4] # start till 6th

Out[61]:  [25, (1+2j), 'nit', True]

In [62]:  l

Out[62]:  [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [63]:  l[4:]

Out[63]:  [3.4, 20, 10]

In [64]:  l[2:]

Out[64]:  ['nit', True, 3.4, 20, 10]

In [65]:  l

Out[65]:  [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [66]:  l[1:6]

Out[66]:  [(1+2j), 'nit', True, 3.4, 20]

In [67]:  l

Out[67]:  [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [68]:  l[1:6:2]

Out[68]:  [(1+2j), True, 20]

In [69]:  l

Out[69]:  [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [70]:  l[1:6:3]

Out[70]:  [(1+2j), 3.4]

In [71]:  l

Out[71]:  [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [72]:  l[::-1]

Out[72]:  [10, 20, 3.4, True, 'nit', (1+2j), 25]
```

```
In [73]:  l

Out[73]: [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [74]:  l[::-2]

Out[74]: [10, 3.4, 'nit', 25]

In [75]:  l

Out[75]: [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [76]:  l[::-3]

Out[76]: [10, True, 25]

In [77]:  l

Out[77]: [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [79]:  3.4 in l

Out[79]: True

In [80]:  True & True

Out[80]: True

In [81]:  True & False

Out[81]: False

In [ ]:  False & True

In [83]:  l

Out[83]: [25, (1+2j), 'nit', True, 3.4, 20, 10]
```

for i in l: print(i)

```
In [85]:  for i in rev_list:
               print(i)

           25
           (1+2j)
           nit
           True
           3.4
           20
           10
```

enumerate

```
In [87]:  for i in enumerate(l):
               print(i)

           (0, 25)
           (1, (1+2j))
           (2, 'nit')
           (3, True)
           (4, 3.4)
           (5, 20)
           (6, 10)
```

```
In [89]:  l3

Out[89]:  [500, 20, 17, 5]
```

```
In [90]:  for i in enumerate(l3):
               print(i)

           (0, 500)
           (1, 20)
           (2, 17)
           (3, 5)
```

ALL / ANY

-

```
In [92]:  l

Out[92]:  [25, (1+2j), 'nit', True, 3.4, 20, 10]
```

```
In [93]:  all(l)

Out[93]:  True
```

```
In [94]:  l.append(0)
          l

Out[94]:  [25, (1+2j), 'nit', True, 3.4, 20, 10, 0]

In [95]:  any(l)

Out[95]:  True

In [96]:  all(l)

Out[96]:  False

In [98]:  l

Out[98]:  [25, (1+2j), 'nit', True, 3.4, 20, 10, 0]

In [99]:  l.pop()

Out[99]:  0

In [100]: l

Out[100]: [25, (1+2j), 'nit', True, 3.4, 20, 10]

In [103]: l.pop(True)

Out[103]: (1+2j)

In [102]: l

Out[102]: [25, (1+2j), 'nit', 3.4, 20, 10]

In [104]: True + True

Out[104]: 2

In [105]: True

Out[105]: True

In [1]:   i = 1

In [2]:   i

Out[2]:   1
```

tuple

```
In [4]:    t = ()
```

```
In [5]:    t

Out[5]:    ()
```

```
In [7]:    type(t)

Out[7]:    tuple
```

```
In [9]:    t1 = (10, 20, 30)
           t1

Out[9]:    (10, 20, 30)
```

```
In [10]:   t1.append(40)

           ---------------------------------------------------------------------------
           --
           AttributeError                            Traceback (most recent call las
           t)
           Cell In[10], line 1
           ----> 1 t1.append(40)

           AttributeError: 'tuple' object has no attribute 'append'
```

```
In [11]:   t1

Out[11]:   (10, 20, 30)
```

```
In [12]:   t1[:]

Out[12]:   (10, 20, 30)
```

```
In [13]:   t1[1:]

Out[13]:   (20, 30)
```

```
In [14]:   t1[0]

Out[14]:   10
```

```
In [15]:  t1

Out[15]:  (10, 20, 30)

In [17]:  t1.count(10)

Out[17]:  1

In [21]:  t1.index(10)

Out[21]:  0

In [23]:  l_ = [10,20]

In [24]:  l_[0] = 1000

In [25]:  l_

Out[25]:  [1000, 20]

In [26]:  t1

Out[26]:  (10, 20, 30)

In [27]:  t1[0] = 1000

          ---------------------------------------------------------------------------
          --
          TypeError                                 Traceback (most recent call las
          t)
          Cell In[27], line 1
          ----> 1 t1[0] = 1000

          TypeError: 'tuple' object does not support item assignment


In [28]:  icici = ('john', '1st dec 1980', 456567, 'cizp5er')
          icici

Out[28]:  ('john', '1st dec 1980', 456567, 'cizp5er')
```

In [29]:
```
icici.remove('john')
```

```
---------------------------------------------------------------------------
--
AttributeError                            Traceback (most recent call las
t)
Cell In[29], line 1
----> 1 icici.remove('john')

AttributeError: 'tuple' object has no attribute 'remove'
```

In [30]:
```
t1
```

Out[30]:
```
(10, 20, 30)
```

In [31]:
```
t1 * 2
```

Out[31]:
```
(10, 20, 30, 10, 20, 30)
```

In [32]:
```
t1 * 3
```

Out[32]:
```
(10, 20, 30, 10, 20, 30, 10, 20, 30)
```

In [45]:
```
t2 = (10, 10, 20, 'nit', 1 +2j, True)
t2
```

Out[45]:
```
(10, 10, 20, 'nit', (1+2j), True)
```

In [35]:
```
t2.clear()
```

```
---------------------------------------------------------------------------
--
AttributeError                            Traceback (most recent call las
t)
Cell In[35], line 1
----> 1 t2.clear()

AttributeError: 'tuple' object has no attribute 'clear'
```

In [36]:
```
del t2
```

In [37]:
```
len(t1)
```

Out[37]:
```
3
```

In [38]:
```
len(t2)
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Cell In[38], line 1
----> 1 len(t2)

NameError: name 't2' is not defined
```

In [43]:
```
t1
```

Out[43]:
```
(10, 20, 30)
```

In [46]:
```
t2
```

Out[46]:
```
(10, 10, 20, 'nit', (1+2j), True)
```

In [47]:
```
t2[3]
```

Out[47]:
```
'nit'
```

In [48]:
```
t2[3][3]
```

```
---------------------------------------------------------------------------
--
IndexError                                Traceback (most recent call las
t)
Cell In[48], line 1
----> 1 t2[3][3]

IndexError: string index out of range
```

In [49]:
```
t2[3][2]
```

Out[49]:
```
't'
```

In [50]:
```
a, b = 5, 6
print(type(a))
print (a+b)
```

```
<class 'int'>
11
```

In [51]:
```
print(int.__add__(a,b)) # magic method
```

```
11
```

In [53]:

```
print(int.__sub__(a,b)) # magic method
```

```
-1
```

In [54]:

```
print(int.__mul__(a,b)) # magic method
```

```
30
```

In [55]:

```
t1
```

Out[55]:

```
(10, 20, 30)
```

t1[::-1]

In [58]:

```
t1[::-2]
```

Out[58]:

```
(30, 10)
```

In [59]:

```
for i in enumerate(t1):
    print(i)
```

```
(0, 10)
(1, 20)
(2, 30)
```

tuple we are completed

RANGE

In [62]:

```
range(5)
```

Out[62]:

```
range(0, 5)
```

In [63]:

```
list(range(5))
```

Out[63]:

```
[0, 1, 2, 3, 4]
```

In [64]:

```
r = range(10)
r
```

Out[64]:

```
range(0, 10)
```

In [65]:

```
for i in r:
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

In [66]:

```
range(1, 10)
```

Out[66]:

```
range(1, 10)
```

In [67]:

```
list(range(1,10))
```

Out[67]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [68]:

```
r
```

Out[68]:

```
range(0, 10)
```

In [69]:

```
r[0]
```

Out[69]:

```
0
```

In [70]:

```
r[5]
```

Out[70]:

```
5
```

In [71]:

```
r[0:5]
```

Out[71]:

```
range(0, 5)
```

In [72]:

```
range(1,100,10)
```

Out[72]:

```
range(1, 100, 10)
```

In [73]:

```
list(range(1,100,10))
```

Out[73]:

```
[1, 11, 21, 31, 41, 51, 61, 71, 81, 91]
```

In [74]:
```
list(range(0,10,2))
```

Out[74]:
```
[0, 2, 4, 6, 8]
```

In [75]:
```
range(0,10,2,1)
```

```
---------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
Cell In[75], line 1
----> 1 range(0,10,2,1)

TypeError: range expected at most 3 arguments, got 4
```

Range completed

SET

In [78]:
```
s_ = {}
s_
```

Out[78]:
```
{}
```

In [79]:
```
s ={10, 200, 10, 30, 5, 1000}
s
```

Out[79]:
```
{5, 10, 30, 200, 1000}
```

In [80]:
```
s1 = {6, 'nit', 1 + 2j, 10, True, 3.4}
s1
```

Out[80]:
```
{(1+2j), 10, 3.4, 6, True, 'nit'}
```

In [81]:
```
s1.add(False)
```

In [82]:
```
s1
```

Out[82]:
```
{(1+2j), 10, 3.4, 6, False, True, 'nit'}
```

In [84]:
```
s1.add(10)
```

```
In [85]:  s1

Out[85]:  {(1+2j), 10, 3.4, 6, False, True, 'nit'}
```

```
In [86]:  s2 = s1.copy()
```

```
In [87]:  s3 = s1.copy()
```

```
In [88]:  s1, s2, s3

Out[88]:  ({(1+2j), 10, 3.4, 6, False, True, 'nit'},
           {(1+2j), 10, 3.4, 6, False, True, 'nit'},
           {(1+2j), 10, 3.4, 6, False, True, 'nit'})
```

```
In [89]:  print(id(s1))
          print(id(s2))
          print(id(s3))

          3235681707936
          3235682850336
          3235683266656
```

```
In [90]:  s3.clear()
```

```
In [91]:  s3

Out[91]:  set()
```

```
In [92]:  del s3
```

```
In [93]:  s3

          ---------------------------------------------------------------------------
          --
          NameError                                 Traceback (most recent call las
          t)
          Cell In[93], line 1
          ----> 1 s3

          NameError: name 's3' is not defined
```

```
In [94]:  s1

Out[94]:  {(1+2j), 10, 3.4, 6, False, True, 'nit'}
```

```
In [95]:   s1.remove(1+2j)

In [96]:   s1

Out[96]:   {10, 3.4, 6, False, True, 'nit'}

In [97]:   s1

Out[97]:   {10, 3.4, 6, False, True, 'nit'}

In [98]:   s1.pop()

Out[98]:   False

In [99]:   s1

Out[99]:   {10, 3.4, 6, True, 'nit'}

In [100]:  s1.pop()

Out[100]:  True

In [101]:  s1

Out[101]:  {10, 3.4, 6, 'nit'}

In [103]:  s1.pop(1)

           -------------------------------------------------------------------------
           --
           TypeError                                 Traceback (most recent call las
           t)
           Cell In[103], line 1
           ----> 1 s1.pop(1)

           TypeError: set.pop() takes no arguments (1 given)


In [104]:  s1

Out[104]:  {10, 6, 'nit'}
```

```
In [105]:  s1.pop()
```

```
---------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
Cell In[105], line 1
----> 1 s1.pop('nit')

TypeError: set.pop() takes no arguments (1 given)
```

```
In [106]:  s
```

```
Out[106]: {5, 10, 30, 200, 1000}
```

```
In [107]:  s1
```

```
Out[107]: {10, 6, 'nit'}
```

```
In [108]:  s2
```

```
Out[108]: {(1+2j), 10, 3.4, 6, False, True, 'nit'}
```

```
In [113]:  s1.update([1,2])
```

```
In [111]:  s1
```

```
Out[111]: {1, 10, 2, 6, 'nit'}
```

```
In [114]:  s1
```

```
Out[114]: {1, 10, 2, 6, 'nit'}
```

```
In [115]:  len(s1)
```

```
Out[115]: 5
```

```
In [1]:  s1.discard(3)
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Cell In[1], line 1
----> 1 s1.discard(3)

NameError: name 's1' is not defined
```

In [117]:
```
s1.remove(3)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call las
t)
Cell In[117], line 1
----> 1 s1.remove(3)

KeyError: 3
```

In [118]:
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
```

In [120]:
```
A, B , C
```

Out[120]:
```
({1, 2, 3, 4, 5}, {4, 5, 6, 7, 8}, {8, 9, 10})
```

In [121]:
```
A | B
```

Out[121]:
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [122]:
```
A.union(B)
```

Out[122]:
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [123]:
```
A.union(B, C)
```

Out[123]:
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [124]:
```
A, B, C
```

Out[124]:
```
({1, 2, 3, 4, 5}, {4, 5, 6, 7, 8}, {8, 9, 10})
```

In [125]:
```
A
```

Out[125]:
```
{1, 2, 3, 4, 5}
```

In [126]:
```
A.update(B)
```

In [128]:
```
A, B, C
```

Out[128]:
```
({1, 2, 3, 4, 5, 6, 7, 8}, {4, 5, 6, 7, 8}, {8, 9, 10})
```

```
In [129]:  A & B
Out[129]:  {4, 5, 6, 7, 8}

In [130]:  B & C
Out[130]:  {8}

In [131]:  B.intersection(C)
Out[131]:  {8}

In [132]:  B.intersection(A)
Out[132]:  {4, 5, 6, 7, 8}

In [133]:  A
Out[133]:  {1, 2, 3, 4, 5, 6, 7, 8}

In [134]:  B
Out[134]:  {4, 5, 6, 7, 8}

In [135]:  A - B
Out[135]:  {1, 2, 3}

In [137]:  A
Out[137]:  {1, 2, 3, 4, 5, 6, 7, 8}

In [138]:  B
Out[138]:  {4, 5, 6, 7, 8}

In [139]:  C
Out[139]:  {8, 9, 10}

In [140]:  B ^ C
Out[140]:  {4, 5, 6, 7, 9, 10}

In [141]:  B -C
Out[141]:  {4, 5, 6, 7}
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: