```
In [1]: v=25 ## variable declaration
        v

Out[1]: 25

In [2]: type(v)

Out[2]: int

In [3]: id(v)

Out[3]: 140708538559672

In [4]: v1=25,29,38
        v1

Out[4]: (25, 29, 38)

In [7]: id(v1)

Out[7]: 2404010087872

In [8]: type(v1)

Out[8]: tuple

In [9]: v_1=(85)
        v_1

Out[9]: 85

In [10]: type(v_1)

Out[10]: int

In [11]: id(v_1)

Out[11]: 140708538561592

In [14]: 90=('v3')
         90

  Cell In[14], line 1
    90=('v3')
    ^
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='?

In [15]: 1s=85
         1s

  Cell In[15], line 1
    1s=85
    ^
SyntaxError: invalid decimal literal
```

In [16]:

```
@z=89
@z
```

```
  Cell In[16], line 1
    @z=89
     ^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

In [ ]:

```

```

In [17]:

```
v_1
```

Out[17]:

```
85
```

In [18]:

```
V_1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 V_1

NameError: name 'V_1' is not defined
```

In [19]:

```
p_=85
p_
```

Out[19]:

```
85
```

In [21]:

```
for =98
for
```

```
  Cell In[21], line 1
    for =98
        ^
SyntaxError: invalid syntax
```

In [22]:

```
import keyword
keyword.kwlist
```

Out[22]:
```
['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 [24]:
```
False=95
False
```

```
  Cell In[24], line 1
    False=95
    ^
SyntaxError: cannot assign to False
```

In [25]:
```
len(keyword.kwlist)
```

Out[25]:
```
35
```

In [26]:
```
len(v_1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[26], line 1
----> 1 len(v_1)

TypeError: object of type 'int' has no len()
```

In [28]:
```
For=29
For
```

Out[28]:
```
29
```

In [29]:
```
for
```

```
  Cell In[29], line 1
    for
       ^
SyntaxError: invalid syntax
```

In [38]:
```
prefix1="py"
prefix1="thon"
prefix1
```

Out[38]:
```
'thon'
```

In [33]:
```
prefix2=prefix+prefix1
prefix2
```

Out[33]:
```
'python'
```

19th June

datatypes

In [43]:
```
i=32
```

In [44]:
```
i
```

Out[44]:
```
32
```

In [45]:
```
type(i)
```

Out[45]:
```
int
```

In [46]:
```
id(i)
```

Out[46]:
```
140708538559896
```

In [47]:
```
f=14.52
f
```

Out[47]:
```
14.52
```

In [48]:
```
type(f)
```

Out[48]:
```
float
```

In [50]:
```
id(f)
```

Out[50]:
```
2404008589232
```

In [51]:
```
s="srikanth"
s
```

```
Out[51]:  'srikanth'
```

```
In [53]:  type(s)
```

```
Out[53]:  str
```

```
In [54]:  id(s)
```

```
Out[54]:  2404064432176
```

```
In [55]:  b=True
          b
```

```
Out[55]:  True
```

```
In [56]:  type(b)
```

```
Out[56]:  bool
```

```
In [57]:  id(b)
```

```
Out[57]:  140708537432960
```

```
In [58]:  c=2+10j
          c
```

```
Out[58]:  (2+10j)
```

```
In [59]:  type(c)
```

```
Out[59]:  complex
```

```
In [60]:  id(c)
```

```
Out[60]:  2404064056464
```

```
In [61]:  print(i)
          print(s)
          print(b)
          print(f)
          print(c)
```

```
32
srikanth
True
14.52
(2+10j)
```

```
In [62]:  f2=1e2
          f2
```

```
Out[62]:  100.0
```

```
In [63]:  f3=1e1
          f3
```

```
Out[63]:  10.0

In [64]:  f4=1e4
          f4

Out[64]:  10000.0

In [65]:  a = 58
          b= 39
          a+b

Out[65]:  97

In [66]:  a-b

Out[66]:  19

In [67]:  a*b

Out[67]:  2262

In [68]:  print(a)
          print(b)

        58
        39

In [69]:  a/b

Out[69]:  1.4871794871794872

In [70]:  a-b

Out[70]:  19

In [72]:  print(a+b)
          print(a-b)
          print(a*b)
          print(a/b)

        97
        19
        2262
        1.4871794871794872

In [74]:  print(a)
          print(b)

        58
        39

In [77]:  a1=85
          b1=95
          a+b

Out[77]:  97

In [88]:  print('The addition of two numbers', a1,'and',b1, 'is=',a+b)
```

```
The addition of two numbers 85 and 95 is= 97
```

In [83]:
```
num1=20
num2=30
num1+num2
```

Out[83]:
```
50
```

In [97]:
```
print('the addition of two numbers',num1,'and',num2,'is=',num1+num2)
```

```
the addition of two numbers 20 and 30 is= 50
```

In [98]:
```
z1=20
z2=25
z3=29
z1+z2+z3
```

Out[98]:
```
74
```

In [106…
```
print('The addition of {} and {} and {} is= {}'.format(z1+z2+z3))
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[106], line 1
----> 1 print('The addition of {} and {} and {} is= {}'.format(z1+z2+z3))

IndexError: Replacement index 1 out of range for positional args tuple
```

In [107…
```
c
```

Out[107…
```
(2+10j)
```

In [108…
```
c.real
```

Out[108…
```
2.0
```

In [109…
```
c.imag
```

Out[109…
```
10.0
```

In [110…
```
c=10+20j
d=15+25j
c+d
```

Out[110…
```
(25+45j)
```

In [111…
```
c-d
```

Out[111…
```
(-5-5j)
```

In [112…
```
print(c+d)
print(c-d)
print(c*d)
```

```
(25+45j)
(-5-5j)
(-350+550j)
```

In [114…

```
print(c)
print(d)
```

```
(10+20j)
(15+25j)
```

In [116…

```
c*d
```

Out[116…

```
(-350+550j)
```

In [118…

```
def team():
    print("hello")
```

In [119…

```

```

Out[119…

```
<function __main__.team()>
```

In [122…

```
b=True
b
```

Out[122…

```
True
```

In [121…

```
import keyword
keyword.kwlist
```

Out[121…

```
['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 [123…

```
b1=False
b1
```

Out[123…

```
False
```

In [124…

```
type(b1)
```

Out[124…

```
bool
```

In [125…

```
int(True)
```

Out[125…

```
1
```

In [126…

```
int(False)
```

Out[126…

```
0
```

In [127…

```
True+False
```

Out[127…

```
1
```

In [128…

```
True+True
```

```
Out[128… 2

In [129… False-True

Out[129… -1

In [130… True-False

Out[130… 1

In [131… False-False

Out[131… 0

In [132… True-True

Out[132… 0

In [133… True+True*False-True

Out[133… 0

In [136… b='True'
         b

Out[136… 'True'

In [137… b[0]

Out[137… 'T'

In [138… s="hello python"
         s

Out[138… 'hello python'

In [139… s[1] ## forward indexing

Out[139… 'e'

In [140… s[-3] #backward indexing

Out[140… 'h'

In [141… s[13]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[141], line 1
----> 1 s[13]

IndexError: string index out of range
```

```
In [142… s[12]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[142], line 1
----> 1 s[12]

IndexError: string index out of range
```

```
In [143…  s[11]

Out[143…  'n'
```

```
In [144…  s[:]

Out[144…  'hello python'
```

```
In [145…  s[:2]

Out[145…  'he'
```

```
In [1]:  s[1:5]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 1
----> 1 s[1:5]

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

```
In [2]:  s
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[2], line 1
----> 1 s

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

```
In [6]:  s='hello python'
         s

Out[6]:  'hello python'
```

```
In [7]:  s

Out[7]:  'hello python'
```

```
In [8]:  s[1:5]

Out[8]:  'ello'
```

```
In [9]:  s[:3] ## forward slicing

Out[9]:  'hel'
```

```
In [10]:  s[-4:] backward slicing

Out[10]:  'thon'
```

```
In [11]:  s[-4:-1]

Out[11]:  'tho'

In [12]:  s

Out[12]:  'hello python'

In [22]:  s[-6:]

Out[22]:  'python'

In [23]:  s[::-2]

Out[23]:  'nhy le'

In [24]:  s[::2]

Out[24]:  'hlopto'

In [25]:  s[2::]

Out[25]:  'llo python'

In [26]:  s

Out[26]:  'hello python'

In [27]:  s[1:13:3] ## three step slicing

Out[27]:  'eoyo'

In [29]:  s[-2:-6]

Out[29]:  ''

In [30]:  s

Out[30]:  'hello python'

In [31]:  s[-5:-1]

Out[31]:  'ytho'

In [33]:  s[-6:-1]

Out[33]:  'pytho'

In [34]:  s[-1:-6]

Out[34]:  ''

In [35]:  s[::-1]

Out[35]:  'nohtyp olleh'
```

```
In [36]: s[2:]

Out[36]: 'llo python'

In [37]: s[:2]

Out[37]: 'he'

In [38]: s[:6]

Out[38]: 'hello '

In [39]: s[0:13:5]

Out[39]: 'h o'

In [40]: s1='nareshit'
         s1

Out[40]: 'nareshit'
```

Python type casting

```
In [41]: int(5.3)

Out[41]: 5

In [42]: type(int)

Out[42]: type

In [43]: int(2.3,4.5)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[43], line 1
----> 1 int(2.3,4.5)

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

In [44]: int(3.2)

Out[44]: 3

In [45]: float(3.2)

Out[45]: 3.2

In [47]: type(3.2)

Out[47]: float

In [48]: int(True)

Out[48]: 1
```

In [49]:
```
type(True)
```

Out[49]:
```
bool
```

In [50]:
```
int(1+2j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[50], line 1
----> 1 int(1+2j)

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

In [52]:
```
int('10')
```

Out[52]:
```
10
```

In [53]:
```
print(int(3.2))
print(int(True))
print(int(False))
print(int('10'))
```

```
3
1
0
10
```

In [54]:
```
print(int(1+10j)) # complext to int is not possible
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[54], line 1
----> 1 print(int(1+10j))

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

In [55]:
```
int(1)
```

Out[55]:
```
1
```

In [56]:
```
print(float(1))
```

```
1.0
```

In [57]:
```
print(float(True))
```

```
1.0
```

In [58]:
```
print(float(False))
```

```
0.0
```

In [60]:
```
print(float('10'))
```

```
10.0
```

In [61]:
```
print(float('srikanth')) string to float is not possible
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[61], line 1
----> 1 print(float('srikanth'))

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

In [62]:

```
print(bool(1))
```

```
True
```

In [63]:

```
print(bool(2.0))
```

```
True
```

In [64]:

```
print(bool('sri'))
```

```
True
```

In [65]:

```
print(bool(1+2j))
```

```
True
```

In [66]:

```
complex(2)
```

Out[66]:

```
(2+0j)
```

In [67]:

```
complex(2.0)
```

Out[67]:

```
(2+0j)
```

In [68]:

```
complex(True)
```

Out[68]:

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

In [69]:

```
complex(0.0)
```

Out[69]:

```
0j
```

In [71]:

```
complex(false,false)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[71], line 1
----> 1 complex(false,false)

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

In [73]:

```
complex(1,3)
```

Out[73]:

```
(1+3j)
```

In [75]:

```
complex(1,2,3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[75], line 1
----> 1 complex(1,2,3)

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

```
In [76]: complex(1,2.5)

Out[76]: (1+2.5j)

In [77]: complex(4,'5')

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[77], line 1
----> 1 complex(4,'5')

TypeError: complex() second arg can't be a string

In [78]: str(2)

Out[78]: '2'

In [79]: str(2.0)

Out[79]: '2.0'

In [80]: str(True)

Out[80]: 'True'

In [82]: str(False)

Out[82]: 'False'

In [83]: str(1+2j)

Out[83]: '(1+2j)'
```

List Datastructure

```
In [84]: l=[]

In [85]: l

Out[85]: []

In [86]: type(l)

Out[86]: list

In [87]: id(l)

Out[87]: 2000223581568

In [88]: print(type(1))

<class 'int'>

In [90]: l.append(10)
In [97]:  l.pop(1)
Out[97]:  10

In [98]:  l
Out[98]:  [10]

In [101…  l.append(20)
          l.append(35)
          l.append(45)
          l.append(46)
          l
Out[101…  [10, 20, 35, 45, 46, 20, 35, 45, 46, 20, 35, 45, 46]

In [103…  l.count(20)
Out[103…  3

In [104…  l.count(35)
Out[104…  3

In [105…  l.remove(35)
          l.remove(45)
          l.remove(46)

In [106…  l
Out[106…  [10, 20, 20, 35, 45, 46, 20, 35, 45, 46]

In [107…  l.index(10)
Out[107…  0

In [108…  l.index(46)
Out[108…  5

In [110…  l2=[]

In [114…  l
Out[114…  [10, 20, 20, 35, 45, 46, 20, 35, 45, 46]

In [115…  l.insert(3,30)

In [116…  l
Out[116…  [10, 20, 20, 30, 35, 45, 46, 20, 35, 45, 46]

In [117…  l.insert(37,3)

In [118…  l
```

```
Out[118…  [10, 20, 20, 30, 35, 45, 46, 20, 35, 45, 46, 3]
```

```
In [119…  l.insert(3,37)
          l
```

```
Out[119…  [10, 20, 20, 37, 30, 35, 45, 46, 20, 35, 45, 46, 3]
```

```
In [121…  l.insert(12,2)
          l
```

```
Out[121…  [10, 20, 20, 37, 30, 35, 45, 46, 20, 35, 45, 46, 2, 3, 2]
```

```
In [124…  l.copy(b)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[124], line 1
----> 1 l.copy(b)

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

```
In [125…  b=[]
```

```
In [126…  l.copy(b)
          b
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[126], line 1
----> 1 l.copy(b)
      2 b

TypeError: list.copy() takes no arguments (1 given)
```

```
In [127…  b=l.copy()
          b
```

```
Out[127…  [10, 20, 20, 37, 30, 35, 45, 46, 20, 35, 45, 46, 2, 3, 2]
```

```
In [128…  l2=l.copy()
          l2
```

```
Out[128…  [10, 20, 20, 37, 30, 35, 45, 46, 20, 35, 45, 46, 2, 3, 2]
```

```
In [129…  id(b)
```

```
Out[129…  2000225922176
```

```
In [130…  id(l)==id(l2)
```

```
Out[130…  False
```

```
In [133…  l==l2==b
```

```
Out[133…  True
```

```
In [134…  id(l)!=id(b)
```

Out[134…

```
True
```

In [138…

```
len(l)
len(b)
len(l2)
```

Out[138…

```
14
```

In [137…

```
l2.pop(4)
```

Out[137…

```
30
```

In [140…

```
len(l)
```

Out[140…

```
15
```

In [141…

```
l.clear()
```

In [142…

```
l
```

Out[142…

```
[]
```

In [145…

```
del l
```

In [146…

```
l
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[146], line 1
----> 1 l

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

In [147…

```
l3=[]
```

In [148…

```
l3.append(14)
l3.append(2.0)
l3.append('srikanth')
l3.append(20+20j)
l3
```

Out[148…

```
[14, 2.0, 'srikanth', (20+20j)]
```

In [149…

```
print(b)
print(l3)
```

```
[10, 20, 20, 37, 30, 35, 45, 46, 20, 35, 45, 46, 2, 3, 2]
[14, 2.0, 'srikanth', (20+20j)]
```

In [152…

```
l3.index('srikanth')
```

Out[152…

```
2
```

In [154…

```
l3[2]
```

Out[154…

```
'srikanth'
```

In [155…

```
l3[2][3]
```

Out[155…

```
'k'
```

In [156…

```
l3
```

Out[156…

```
[14, 2.0, 'srikanth', (20+20j)]
```

In [157…

```
print(l3[3])
```

```
(20+20j)
```

In [158…

```
print(l3[-1])
```

```
(20+20j)
```

In [159…

```
print(l3[2][0])
print(l3[2][1])
print(l3[2][2])
print(l3[2][3])
print(l3[2][4])
```

```
s
r
i
k
a
```

In [161…

```
l3[:]
```

Out[161…

```
[14, 2.0, 'srikanth', (20+20j)]
```

In [163…

```
l3[:0]
```

Out[163…

```
[]
```

In [164…

```
l3[2:]
```

Out[164…

```
['srikanth', (20+20j)]
```

In [165…

```
l3[:-2]
```

Out[165…

```
[14, 2.0]
```

In [167…

```
l3[-2:]
```

Out[167…

```
['srikanth', (20+20j)]
```

In [169…

```
l3[0:4]
```

Out[169…

```
[14, 2.0, 'srikanth', (20+20j)]
```

In [172…

```
l3[0:4:3]
```

Out[172…

```
[14, (20+20j)]
```

In [173…

```
b
```

```
Out[173… [10, 20, 20, 37, 30, 35, 45, 46, 20, 35, 45, 46, 2, 3, 2]
```

```
In [177… b.insert(15,7)
```

```
In [178… b
```

```
Out[178… [10, 20, 20, 37, 30, 35, 45, 46, 20, 35, 45, 46, 2, 3, 2, 7, 2]
```

```
In [179… b.pop(15)
         b
```

```
Out[179… [10, 20, 20, 37, 30, 35, 45, 46, 20, 35, 45, 46, 2, 3, 2, 2]
```

Tuple Datastructure

```
In [182… tup1=[]
```

```
In [183… tup1
```

```
Out[183… []
```

```
In [184… type(tup1)
```

```
Out[184… list
```

```
In [185… tup1=(10,20)
         tup1
```

```
Out[185… (10, 20)
```

```
In [186… type(tup1)
```

```
Out[186… tuple
```

```
In [187… tup1.count(20)
```

```
Out[187… 1
```

```
In [188… tup2=(2.5,3.0,5.5,4.6)
         tup2
```

```
Out[188… (2.5, 3.0, 5.5, 4.6)
```

```
In [189… tup3=('vigu','tippu','shalu','srik')
         tup3
```

```
Out[189… ('vigu', 'tippu', 'shalu', 'srik')
```

```
In [190… tup4=(1,8.5,'vignesh',50+50j)
         tup4
```

```
Out[190… (1, 8.5, 'vignesh', (50+50j))
```

```
In [191… tup4.count('vignesh')
```

Out[191…

```
1
```

In [193…

```
tup4.index(1)
```

Out[193…

```
0
```

In [196…

```
tup4.index(1)
```

Out[196…

```
0
```

In [199…

```
tup4[-1]
```

Out[199…

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

In [200…

```
tup4[-4]
```

Out[200…

```
1
```

In [5]:

```
tup3
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[5], line 1
----> 1 tup3

NameError: name 'tup3' is not defined
```

In [202…

```
tup3[:3]
```

Out[202…

```
('vigu', 'tippu', 'shalu')
```

In [204…

```
tup3[3:]
```

Out[204…

```
('srik',)
```

In [205…

```
tup3[0:3]
```

Out[205…

```
('vigu', 'tippu', 'shalu')
```

In [207…

```
tup3[-3:]
```

Out[207…

```
('tippu', 'shalu', 'srik')
```

In [208…

```
del tup3
```

In [209…

```
tup3
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[209], line 1
----> 1 tup3

NameError: name 'tup3' is not defined
```

In [210…

```
del tup2[2]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[210], line 1
----> 1 del tup2[2]

TypeError: 'tuple' object doesn't support item deletion
```

In [211…

```
tup1
```

Out[211…

```
(10, 20)
```

Loops

In [213…

```
for i in tup1:
    print(i)
```

```
10
20
```

In [214…

```
tup2
```

Out[214…

```
(2.5, 3.0, 5.5, 4.6)
```

In [215…

```
for i in enumerate(tup2):
    print(i)
```

```
(0, 2.5)
(1, 3.0)
(2, 5.5)
(3, 4.6)
```

Tuple Membership

In [216…

```
2.5 in tup2
```

Out[216…

```
True
```

In [218…

```
4.7 in tup2
```

Out[218…

```
False
```

In [220…

```
if 4.6 in tup2:
    print("4.6 is present in the tupe")
else:
    Print("four is not present in the tupe")
```

```
4.6 is present in the tupe
```

In [221…

```
tup2.index(4.6)
```

Out[221…

```
3
```

In [8]:

```
tup5=(1,8,19,56,25)
tup5
```

Out[8]:

```
(1, 8, 19, 56, 25)
```

```
In [223…  sorted(tup5)

Out[223…  [1, 8, 19, 25, 56]

In [224…  tup5.add(2.5)

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[224], line 1
----> 1 tup5.add(2.5)

AttributeError: 'tuple' object has no attribute 'add'

In [1]:  l2

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 1
----> 1 l2

NameError: name 'l2' is not defined

In [2]:  tup2

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[2], line 1
----> 1 tup2

NameError: name 'tup2' is not defined

In [4]:  tup6.extend(tup5)

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[4], line 1
----> 1 tup6.extend(tup5)

NameError: name 'tup6' is not defined

In [6]:  tup3=(25,98,23,64,52)
         tup3

Out[6]:  (25, 98, 23, 64, 52)

In [9]:  tup3.extend(tup5)

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[9], line 1
----> 1 tup3.extend(tup5)

AttributeError: 'tuple' object has no attribute 'extend'

In [10]:  print(tup3)
          print(tup5)

(25, 98, 23, 64, 52)
(1, 8, 19, 56, 25)
```

```
In [12]: tup3.extend(tup5)
         tup3
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[12], line 1
----> 1 tup3.extend(tup5)
      2 tup3

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

```
In [13]: tup3.sort()
         tup3
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[13], line 1
----> 1 tup3.sort()
      2 tup3

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

```
In [14]: l=[]
         l
```

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

```
In [15]: l=(25,89,65,32,78)
         l
```

```
Out[15]: (25, 89, 65, 32, 78)
```

```
In [16]: l1=[]
```

```
In [17]: l1.extend(1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 1
----> 1 l1.extend(1)

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

```
In [18]: l.sort()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[18], line 1
----> 1 l.sort()

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

```
In [19]: l3=[]
         l3
```

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

```
In [20]: l3=20
```

```
l3
```

Out[20]:

```
20
```

In [21]:

```
l3.appen(25)
l3.append(89)
l3.append(45)
l3.append(14)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[21], line 1
----> 1 l3.append(25)
      2 l3.append(89)
      3 l3.append(45)

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

In [22]:

```
l3
```

Out[22]:

```
20
```

In [23]:

```
b3=[]
b3
```

Out[23]:

```
[]
```

In [24]:

```
b3.append(14)
b3.append(85)
b3.append(47)
b3.append(48)
b3
```

Out[24]:

```
[14, 85, 47, 48]
```

In [25]:

```
b3.sort()
```

In [26]:

```
b3
```

Out[26]:

```
[14, 47, 48, 85]
```

In [27]:

```
b3.reverse()
b3
```

Out[27]:

```
[85, 48, 47, 14]
```

ALL & Any

In [28]:

```
b3
```

Out[28]:

```
[85, 48, 47, 14]
```

In [29]:

```
all(b3)
```

Out[29]:

```
True
```

In [30]:
```
any(b3)
```

Out[30]:
```
True
```

In [31]:
```
b3.append(0)
b3
```

Out[31]:
```
[85, 48, 47, 14, 0]
```

In [34]:
```
all(b3)
```

Out[34]:
```
False
```

In [33]:
```
any(b3)
```

Out[33]:
```
True
```

Set Datastructure

In [35]:
```
k={}
k
```

Out[35]:
```
{}
```

In [37]:
```
type(k)
```

Out[37]:
```
dict
```

In [38]:
```
k1=set()
k1
```

Out[38]:
```
set()
```

In [40]:
```
type(k1)
```

Out[40]:
```
set
```

In [41]:
```
k1.add(20)
k1
```

Out[41]:
```
{20}
```

In [42]:
```
k1.add(78)
k1.add(56)
k1.add(25)
k1.add(63)
```

In [43]:
```
k1
```

Out[43]:
```
{20, 25, 56, 63, 78}
```

In [44]:
```
len(k1)
```

Out[44]:
```
5
```

In [46]:
```
k2={1.5,'srik','True',1+2j}
k2
```

Out[46]:
```
{(1+2j), 1.5, 'True', 'srik'}
```

In [47]:
```
type(k2)
```

Out[47]:
```
set
```

In [48]:
```
print(k1)
print(k2)
```

```
{78, 20, 56, 25, 63}
{1.5, 'True', 'srik', (1+2j)}
```

In [51]:
```
k1.copy(k4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[51], line 1
----> 1 k1.copy(k4)

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

In [50]:
```
k4={}
k4
```

Out[50]:
```
{}
```

In [52]:
```
k4.copy(k1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[52], line 1
----> 1 k4.copy(k1)

TypeError: dict.copy() takes no arguments (1 given)
```

In [53]:
```
k4
```

Out[53]:
```
{}
```

In [54]:
```
k4=(0)
k4
```

Out[54]:
```
0
```

In [56]:
```
k2[2]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[56], line 1
----> 1 k2[2]

TypeError: 'set' object is not subscriptable
```

In [57]:
```
k2[:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[57], line 1
----> 1 k2[:]

TypeError: 'set' object is not subscriptable
```

```
In [58]: k2.add([14,56,98])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[58], line 1
----> 1 k2.add([14,56,98])

TypeError: unhashable type: 'list'
```

```
In [59]: id(k1)==id(k2)
```

```
Out[59]: False
```

```
In [60]: id(k1)!=(k2)
```

```
Out[60]: True
```

```
In [61]: s4=k2.copy()
         s4
```

```
Out[61]: {(1+2j), 1.5, 'True', 'srik'}
```

```
In [63]: s4.pop(0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[63], line 1
----> 1 s4.pop(0)

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

k2.pop()

```
In [68]: k2.pop()
```

```
Out[68]: 'True'
```

```
In [70]: k2.remove(1+2j)
```

```
In [71]: k2
```

```
Out[71]: {'srik'}
```

```
In [73]: k2.discard(0)
```

```
In [74]: k2
```

```
Out[74]: {'srik'}
```

```
In [75]: k2.discard('srik')
In [76]: k2

Out[76]: set()

In [77]: k2.discard(1.5)

In [78]: k2

Out[78]: set()

In [79]: k2=k1.copy()
         k2

Out[79]: {20, 25, 56, 63, 78}

In [80]: print(k1)
         print(k2)

        {78, 20, 56, 25, 63}
        {20, 56, 25, 78, 63}

In [81]: k2.difference(k1)

Out[81]: set()

In [82]: k2-k1

Out[82]: set()

In [83]: k3

        ---------------------------------------------------------------------------
        NameError                                 Traceback (most recent call last)
        Cell In[83], line 1
        ----> 1 k3

        NameError: name 'k3' is not defined

In [84]: s4

Out[84]: {(1+2j), 1.5, 'True', 'srik'}

In [85]: k1-s4

Out[85]: {20, 25, 56, 63, 78}

In [86]: k1.difference(s4)

Out[86]: {20, 25, 56, 63, 78}

In [87]: s4

Out[87]: {(1+2j), 1.5, 'True', 'srik'}

In [88]: s4.difference(k1)
```

```
Out[88]:  {(1+2j), 1.5, 'True', 'srik'}

In [89]:  s4.union(k1)

Out[89]:  {(1+2j), 1.5, 20, 25, 56, 63, 78, 'True', 'srik'}

In [90]:  s4

Out[90]:  {(1+2j), 1.5, 'True', 'srik'}

In [91]:  k1

Out[91]:  {20, 25, 56, 63, 78}

In [92]:  myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}

In [93]:  myset

Out[93]:  {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

In [94]:  myset.clear()

In [95]:  myset

Out[95]:  set()

In [98]:  j1={45,13,47,48,49}
          j2={12,13,14,15,16}

In [99]:  j1.difference(j2)

Out[99]:  {45, 47, 48, 49}

In [100…  j1-j2

Out[100…  {45, 47, 48, 49}

In [101…  j1.difference_update(j2)

In [103…  j2

Out[103…  {12, 13, 14, 15, 16}

In [104…  j2.add(13)
          j2.add(14)

In [105…  print(j1)
          print(j2)

         {48, 49, 45, 47}
         {16, 12, 13, 14, 15}

In [106…  j1.add(14)
          j1.add(15)
          j1.add(13)
```

In [107…

```
print(j1)
print(j2)
```

```
{48, 49, 13, 15, 45, 14, 47}
{16, 12, 13, 14, 15}
```

In [110…

```
j1.difference(j2)
```

Out[110…

```
{45, 47, 48, 49}
```

In [109…

```
j1
```

Out[109…

```
{45, 47, 48, 49}
```

In [111…

```
print(j1)
print(j2)
```

```
{48, 49, 45, 47}
{16, 12, 13, 14, 15}
```

In [112…

```
j1.add(14)
j1.add(15)
j1.add(13)
```

In [113…

```
print(j1)
print(j2)
```

```
{48, 49, 15, 13, 45, 14, 47}
{16, 12, 13, 14, 15}
```

In [114…

```
j2.symmetric_difference(j1)
```

Out[114…

```
{12, 16, 45, 47, 48, 49}
```

In [115…

```
j1.add(14)
j1.add(15)
j1.add(13)
```

In [116…

```
print(j1)
print(j2)
```

```
{48, 49, 15, 13, 45, 14, 47}
{16, 12, 13, 14, 15}
```

In [117…

```
j2.symmetric_difference_update(j1)
```

In [118…

```
j2
```

Out[118…

```
{12, 16, 45, 47, 48, 49}
```

In [119…

```
j1
```

Out[119…

```
{13, 14, 15, 45, 47, 48, 49}
```

In [120…

```
j1[:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[120], line 1
----> 1 j1[:]

TypeError: 'set' object is not subscriptable
```

In [121…

```
j1.index(0)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[121], line 1
----> 1 j1.index(0)

AttributeError: 'set' object has no attribute 'index'
```

In [122…

```
for i in j1:
    print(i)
```

```
48
49
15
13
45
14
47
```

In [124…

```
for i in enumerate(j1):
    print(i)
```

```
(0, 48)
(1, 49)
(2, 15)
(3, 13)
(4, 45)
(5, 14)
(6, 47)
```

In [130…

```
if 45 in j1:
    print('45 is present in the set')
else:
    print('45 is not present')
```

```
45 is present in the set
```

In [131…

```
14 in j1
```

Out[131…

```
True
```

In [132…

```
86 in j1
```

Out[132…

```
False
```

In [133…

```
for i in j1:
    print(i)
```

```
48
49
15
13
45
14
47
```

In [136…

```
l={10,20,30,45,55,65}
m={20,30,55}
n={70,80,90,110}
```

In [137…

```
m.issubset(l)
```

Out[137…

```
True
```

In [138…

```
l.issuperset(m)
```

Out[138…

```
True
```

In [139…

```
n.isdisjoint(m)
```

Out[139…

```
True
```

In [140…

```
n.isdisjoint(l)
```

Out[140…

```
True
```

In [141…

```
l^m
```

Out[141…

```
{10, 45, 65}
```

Dictionary Datastructure

In [142…

```
h={}
h
```

Out[142…

```
{}
```

In [143…

```
type(h)
```

Out[143…

```
dict
```

In [146…

```
h={'salaryA':10000,'salaryB':20000,'salaryc':30000}
h
```

Out[146…

```
{'salaryA': 10000, 'salaryB': 20000, 'salaryc': 30000}
```

In [149…

```
h1=h.copy()
```

In [150…

```
h1
```

Out[150…

```
{'salaryA': 10000, 'salaryB': 20000, 'salaryc': 30000}
```

```
In [151…  h2={1:'pass',2:'average',3:'fail'}
          h2

Out[151…  {1: 'pass', 2: 'average', 3: 'fail'}

In [152…  h2.pop(1)

Out[152…  'pass'

In [153…  h2

Out[153…  {2: 'average', 3: 'fail'}

In [155…  h2.items()

Out[155…  dict_items([(2, 'average'), (3, 'fail')])

In [156…  h2.values()

Out[156…  dict_values(['average', 'fail'])

In [157…  h2.keys()

Out[157…  dict_keys([2, 3])

In [161…  h1.fromkeys('salaryA')

Out[161…  {'s': None, 'a': None, 'l': None, 'r': None, 'y': None, 'A': None}

In [164…  h1.setdefault('salaryA')

Out[164…  10000

In [163…  h1

Out[163…  {'salaryA': 10000, 'salaryB': 20000, 'salaryc': 30000, 2: None}

In [165…  h1

Out[165…  {'salaryA': 10000, 'salaryB': 20000, 'salaryc': 30000, 2: None}

In [169…  h1.popitem()

Out[169…  (2, None)

In [170…  h1.popitem()

Out[170…  ('salaryc', 30000)

In [180…  keys={'a','b','c','d'}

In [172…  keys

Out[172…  {'a', 'b', 'c', 'd'}
```

```
In [178…  values = 10

In [185…  mydict=dict.fromkeys(keys,values)

In [186…  mydict

Out[186…  {'d': 10, 'a': 10, 'c': 10, 'b': 10}

In [183…  mydict=dict.fromkeys(keys,values)

In [184…  mydict

Out[184…  {'d': 10, 'a': 10, 'c': 10, 'b': 10}

In [187…  keys={'srikanth','shalini','vigu','tippu'}

In [188…  values={100,200,300,400}

In [189…  mydict2=dict.fromkeys(keys,values)

In [190…  mydict2

Out[190…  {'vigu': {100, 200, 300, 400},
           'shalini': {100, 200, 300, 400},
           'srikanth': {100, 200, 300, 400},
           'tippu': {100, 200, 300, 400}}

In [193…  values.add(40)
          mydict2

Out[193…  {'vigu': {40, 100, 200, 300, 400},
           'shalini': {40, 100, 200, 300, 400},
           'srikanth': {40, 100, 200, 300, 400},
           'tippu': {40, 100, 200, 300, 400}}

In [ ]:
```