# Tuple

In [6]:
```
t=()
t
```

Out[6]: ()

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

Out[2]: {}

In [3]:
```
len(t)
```

Out[3]: 0

In [7]:
```
t1=(3,5,9,10,32,45,1,6)
t1
```

Out[7]: (3, 5, 9, 10, 32, 45, 1, 6)

In [8]:
```
t2=(24,30.5,True,1+3j)
```

In [9]:
```
t2
```

Out[9]: (24, 30.5, True, (1+3j))

In [10]:
```
t1.count(3)
```

Out[10]: 1

In [11]:
```
t1.add(1)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[11], line 1
----> 1 t1.add(1)

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

In [12]:
```
t1[1]
```

Out[12]: 5

In [13]:
```
t2[0]
```

Out[13]: 24

In [14]:
```
t1[-3]
```

Out[14]: 45

In [15]:
```
t1
```

Out[15]: (3, 5, 9, 10, 32, 45, 1, 6)

In [16]:
```
t1[4]
```

Out[16]: 32

In [17]:
```
t1[:]
```

Out[17]: (3, 5, 9, 10, 32, 45, 1, 6)

In [18]:
```
t1[:2]
```

Out[18]: (3, 5)

In [19]:
```
t1[:-2]
```

Out[19]: (3, 5, 9, 10, 32, 45)

In [20]:
```
t1[2:]
```

Out[20]: (9, 10, 32, 45, 1, 6)

In [21]:
```
t1[-2:]
```

Out[21]: (1, 6)

In [22]:
```
t1[0]=10
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[22], line 1
----> 1 t1[0]=10

TypeError: 'tuple' object does not support item assignment
```

In [25]:
```
BankAccount=('swapna',100,'SBI')
```

In [26]:
```
BankAccount
```

Out[26]: ('swapna', 100, 'SBI')

In [27]:
```
BankAccount[0]='sridhar'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 BankAccount[0]='sridhar'

TypeError: 'tuple' object does not support item assignment
```

In [28]:
```
t2
```

Out[28]: (24, 30.5, True, (1+3j))

In [29]:
```
t1
```

Out[29]: (3, 5, 9, 10, 32, 45, 1, 6)

In [30]:
```
for i in t1:
    print(i)
```

```
3
5
9
10
32
45
1
6
```

In [32]:
```
for i in enumerate(t1):
    print(i)
```

```
(0, 3)
(1, 5)
(2, 9)
(3, 10)
(4, 32)
(5, 45)
(6, 1)
(7, 6)
```

# Set

In [34]:
```
s={}
s
```

Out[34]: {}

In [35]:
```
type(s)
```

Out[35]: dict

In [36]:
```
s1=set()
s1
```

Out[36]: set()

In [37]:
```
s2={4,3,7,5,3,2,9}
```

In [38]:
```
s2
```

Out[38]: {2, 3, 4, 5, 7, 9}

In [39]:
```
s1.add(3)
```

In [40]:
```
s1
```

Out[40]: {3}

In [41]:
```
s2.add(3)
```

In [42]:
```
s2
```

Out[42]:
```
{2, 3, 4, 5, 7, 9}
```

In [43]:
```
s2.add(1)
```

In [44]:
```
s2
```

Out[44]:
```
{1, 2, 3, 4, 5, 7, 9}
```

In [45]:
```
s2.add(10)
```

In [46]:
```
s2
```

Out[46]:
```
{1, 2, 3, 4, 5, 7, 9, 10}
```

In [47]:
```
s2.add(6)
```

In [48]:
```
s2
```

Out[48]:
```
{1, 2, 3, 4, 5, 6, 7, 9, 10}
```

In [49]:
```
s2.add('swapna')
```

In [50]:
```
s2
```

Out[50]:
```
{1, 10, 2, 3, 4, 5, 6, 7, 9, 'swapna'}
```

In [51]:
```
s2.add(10.5)
```

In [52]:
```
s2
```

Out[52]:
```
{1, 10, 10.5, 2, 3, 4, 5, 6, 7, 9, 'swapna'}
```

In [53]:
```
s2.add('hi')
```

In [54]:
```
s2
```

Out[54]:
```
{1, 10, 10.5, 2, 3, 4, 5, 6, 7, 9, 'hi', 'swapna'}
```

In [55]:
```
s2.add(1+2j)
```

In [56]:
```
s2
```

Out[56]:
```
{(1+2j), 1, 10, 10.5, 2, 3, 4, 5, 6, 7, 9, 'hi', 'swapna'}
```

In [57]:
```
s2.remove(10)
```

In [58]:
```
s2
```

Out[58]:
```
{(1+2j), 1, 10.5, 2, 3, 4, 5, 6, 7, 9, 'hi', 'swapna'}
```

In [59]:
```
s2.discard('hi')
```

In [60]:
```
s2
```

Out[60]:
```
{(1+2j), 1, 10.5, 2, 3, 4, 5, 6, 7, 9, 'swapna'}
```

In [61]:
```
s2.discard('hello')
```

In [62]:
```
s2
```

Out[62]:
```
{(1+2j), 1, 10.5, 2, 3, 4, 5, 6, 7, 9, 'swapna'}
```

In [63]:
```
s2.remove('hello')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[63], line 1
----> 1 s2.remove('hello')

KeyError: 'hello'
```

In [64]:
```
s1=s2.copy()
```

In [65]:
```
s1
```

Out[65]:
```
{(1+2j), 1, 10.5, 2, 3, 4, 5, 6, 7, 9, 'swapna'}
```

In [66]:
```
s1==s2
```

Out[66]:
```
True
```

In [67]:
```
s1.add('hi')
```

In [68]:
```
s1
```

Out[68]:
```
{(1+2j), 1, 10.5, 2, 3, 4, 5, 6, 7, 9, 'hi', 'swapna'}
```

In [69]:
```
s1==s2
```

Out[69]:
```
False
```

In [70]:
```
s1.pop()
```

Out[70]:
```
1
```

In [71]:
```
s1.pop()
```

Out[71]:
```
2
```

In [72]:
```
s1.pop()
```

Out[72]:
```
3
```

In [73]:
```
s1.pop(3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[73], line 1
----> 1 s1.pop(3)

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

In [74]:
```
s2[2]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[74], line 1
----> 1 s2[2]

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

In [75]:
```
s2[:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[75], line 1
----> 1 s2[:]

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

In [76]:
```
s1
```

Out[76]:
```
{(1+2j), 10.5, 4, 5, 6, 7, 9, 'hi', 'swapna'}
```

In [77]:
```
s2
```

Out[77]:
```
{(1+2j), 1, 10.5, 2, 3, 4, 5, 6, 7, 9, 'swapna'}
```

In [78]:
```
3 in s2
```

Out[78]:
```
True
```

In [79]:
```
3 in s1
```

Out[79]:
```
False
```

In [80]:
```
for i in s1:
    print(i)
```

```
4
5
6
7
(1+2j)
9
10.5
swapna
hi
```

In [81]:
```
for i in enumerate (s1):
    print(i)
```

```
(0, 4)
(1, 5)
(2, 6)
(3, 7)
(4, (1+2j))
(5, 9)
(6, 10.5)
(7, 'swapna')
(8, 'hi')
```

# SET OPERATIONS(Union)

In [82]:
```
a={2,3,1,6,7}
b={1,5,7,8,3}
c={1,3,4,9,10}
```

In [83]:
```
a
```

Out[83]:
```
{1, 2, 3, 6, 7}
```

In [84]:
```
type(a)
```

Out[84]:
```
set
```

In [85]:
```
a.union(b)
```

Out[85]:
```
{1, 2, 3, 5, 6, 7, 8}
```

In [86]:
```
a.union(c)
```

Out[86]:
```
{1, 2, 3, 4, 6, 7, 9, 10}
```

In [87]:
```
b.union(c)
```

Out[87]:
```
{1, 3, 4, 5, 7, 8, 9, 10}
```

In [101…
```
a|b|c
```

Out[101…
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [ ]:
```
#Intersection
```

In [89]:
```
print(a)
print(b)
print(c)
```

```
{1, 2, 3, 6, 7}
{1, 3, 5, 7, 8}
{1, 3, 4, 9, 10}
```

In [90]:
```
a|b
```

Out[90]:
```
{1, 2, 3, 5, 6, 7, 8}
```

In [91]:
```
a|c
```

Out[91]: {1, 2, 3, 4, 6, 7, 9, 10}

In [92]:
```
b|c
```

Out[92]: {1, 3, 4, 5, 7, 8, 9, 10}

In [93]:
```
a.intersection(b)
```

Out[93]: {1, 3, 7}

In [94]:
```
b.intersection(c)
```

Out[94]: {1, 3}

In [95]:
```
a.intersection(c)
```

Out[95]: {1, 3}

In [96]:
```
a&b
```

Out[96]: {1, 3, 7}

In [97]:
```
a&c
```

Out[97]: {1, 3}

In [98]:
```
b&c
```

Out[98]: {1, 3}

In [100…
```
a&b&c
```

Out[100… {1, 3}

# Difference

In [99]:
```
print(a)
print(b)
print(c)
```

```
{1, 2, 3, 6, 7}
{1, 3, 5, 7, 8}
{1, 3, 4, 9, 10}
```

In [102…
```
a.difference(b)
```

Out[102… {2, 6}

In [103…
```
b.difference(c)
```

Out[103… {5, 7, 8}

In [104…
```
a.difference(c)
```

Out[104… {2, 6, 7}

In [106…
```
b.difference(a)
```

Out[106… {5, 8}

In [107…
```
c.difference(b)
```

Out[107… {4, 9, 10}

In [108…
```
c.difference(a)
```

Out[108… {4, 9, 10}

In [109…
```
a-b
```

Out[109… {2, 6}

In [110…
```
b-c
```

Out[110… {5, 7, 8}

In [ ]:
```
a-c
```