```
In [1]:  #Tuple
         t=()
         t

Out[1]:  ()

In [2]:  type(t)

Out[2]:  tuple

In [3]:  t=(10,20,30)
         t

Out[3]:  (10, 20, 30)

In [4]:  t.count(10)    # occurrence of 10 in the tuple

Out[4]:  1

In [5]:  t.count(20)

Out[5]:  1

In [7]:  t1=(10,20,2.2,'ten',True,1+2j,20)
         t1

Out[7]:  (10, 20, 2.2, 'ten', True, (1+2j), 20)

In [ ]:  #Means duplicate data types can be defined in  the tuple. Also, duplicates are also

In [8]:  t1.count(20)

Out[8]:  2

In [9]:  t1.index(20)      #Returns 1st occurrence index

Out[9]:  1

In [10]: print(t)
         print(t1)

        (10, 20, 30)
        (10, 20, 2.2, 'ten', True, (1+2j), 20)

In [11]: print(len(t))
         print(len(t1))

        3
        7

In [12]: t[0]
```

```
Out[12]:  10
```

```
In [13]:  t[0] = 100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[13], line 1
----> 1 t[0] = 100

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

```
In [ ]:  #Means tuples are immutable, not changeable
```

```
In [14]:  t
```

```
Out[14]:  (10, 20, 30)
```

```
In [15]:  t2 = t *3
          t2
```

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

```
In [16]:  t3 = t + 3
          t3
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[16], line 1
----> 1 t3 = t + 3
      2 t3

TypeError: can only concatenate tuple (not "int") to tuple
```

```
In [17]:  t4 = (10,20,30(60,70))
          t4
```

```
<>:1: SyntaxWarning: 'int' object is not callable; perhaps you missed a comma?
C:\Users\SID Balurkar\AppData\Local\Temp\ipykernel_20432\2160422535.py:1: SyntaxWarn
ing: 'int' object is not callable; perhaps you missed a comma?
  t4 = (10,20,30(60,70))

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 1
----> 1 t4 = (10,20,30(60,70))
      2 t4

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

```
In [ ]:  #No nested tuple allowed
```

```
In [ ]:  #Sets
```

```
In [18]:  s = {}
          s
```

```
Out[18]: {}

In [19]: type(s)

Out[19]: dict

In [20]: s1 = set()     #Empty set
         s1

Out[20]: set()

In [21]: type(s1)

Out[21]: set

In [22]: s2 = {90, 10, 50,40,25,10,50}
         s2

Out[22]: {10, 25, 40, 50, 90}

In [ ]: #Hence, set is ordered and duplicates not allowed

In [23]: type(s2)

Out[23]: set

In [24]: s2

Out[24]: {10, 25, 40, 50, 90}

In [25]: s3 = s2.copy()
         s3

Out[25]: {10, 25, 40, 50, 90}

In [26]: id(s2)

Out[26]: 1662687465920

In [27]: id(s3)

Out[27]: 1662687466816

In [28]: s3

Out[28]: {10, 25, 40, 50, 90}

In [29]: s3.add(3.4)
         s3

Out[29]: {3.4, 10, 25, 40, 50, 90}
```

```
In [30]: s3.add('nit')
         s3

Out[30]: {10, 25, 3.4, 40, 50, 90, 'nit'}

In [31]: s3.add(1+2j)
         s3.add(True)
         s3

Out[31]: {(1+2j), 10, 25, 3.4, 40, 50, 90, True, 'nit'}

In [ ]: # Randomly added

In [32]: print(s)
         print(s1)
         print(s2)
         print(s3)

{}
set()
{50, 90, 40, 25, 10}
{True, 3.4, (1+2j), 10, 25, 90, 40, 50, 'nit'}

In [33]: s.add(1,2)

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[33], line 1
----> 1 s.add(1,2)

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

In [ ]: #add() takes one argument

In [34]: s2.clear()
         s2

Out[34]: set()

In [35]: del s2
         s2

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[35], line 2
      1 del s2
----> 2 s2

NameError: name 's2' is not defined

In [36]: s3

Out[36]: {(1+2j), 10, 25, 3.4, 40, 50, 90, True, 'nit'}
```

```
In [37]: s3.remove(2000)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[37], line 1
----> 1 s3.remove(2000)

KeyError: 2000
```

```
In [ ]: #2000 is not a member
```

```
In [38]: s3.remove(1+2j)
         s3
```

```
Out[38]: {10, 25, 3.4, 40, 50, 90, True, 'nit'}
```

```
In [39]: s3.discard(2000)
         s3
```

```
Out[39]: {10, 25, 3.4, 40, 50, 90, True, 'nit'}
```

```
In [ ]: # discard() - never gives error. removes the element if present
```

```
In [40]: s3.discard(10)
         s3
```

```
Out[40]: {25, 3.4, 40, 50, 90, True, 'nit'}
```

```
In [41]: s3.pop()
```

```
Out[41]: True
```

```
In [42]: s3
```

```
Out[42]: {25, 3.4, 40, 50, 90, 'nit'}
```

```
In [ ]: #pop() randomly removes any element
```

```
In [43]: s3.pop()
```

```
Out[43]: 3.4
```

```
In [44]: s3
```

```
Out[44]: {25, 40, 50, 90, 'nit'}
```

```
In [45]: s3.pop(0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[45], line 1
----> 1 s3.pop(0)

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

```
In [ ]: #pop() has no arguments. No index to pass. Hence, indexing not allowed.
```

```
In [46]: s3[:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[46], line 1
----> 1 s3[:]

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

```
In [ ]: #Slicing not allowed in sets
```

```
In [47]: s3[1:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[47], line 1
----> 1 s3[1:]

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

```
In [48]: s3[2]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[48], line 1
----> 1 s3[2]

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

```
In [ ]: # Set Operations
```

```
In [ ]: # Union OR | symbol
```

```
In [49]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
         c = {8,9,10}
```

```
In [50]: print(a)
         print(b)
         print(c)

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

```
In [51]: type(a)
```

```
Out[51]: set

In [52]: type(b)

Out[52]: set

In [53]: type(c)

Out[53]: set

In [54]: a.union(b)

Out[54]: {1, 2, 3, 4, 5, 6, 7, 8}

In [55]: a.union(b,c)

Out[55]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [56]: a|b

Out[56]: {1, 2, 3, 4, 5, 6, 7, 8}

In [57]: b|c

Out[57]: {4, 5, 6, 7, 8, 9, 10}

In [58]: a|b|c

Out[58]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [59]: a|c

Out[59]: {1, 2, 3, 4, 5, 8, 9, 10}

In [60]: a|c|b

Out[60]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [ ]: #Intersection OR & symbol

In [61]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
         c = {8,9,10}
         print(a)
         print(b)
         print(c)

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

In [62]: a.intersection(b)
```

```
Out[62]: {4, 5}
```

```
In [63]: b.intersection(c)
```

```
Out[63]: {8}
```

```
In [64]: a&b
```

```
Out[64]: {4, 5}
```

```
In [65]: b&c
```

```
Out[65]: {8}
```

```
In [66]: a&a
```

```
Out[66]: {1, 2, 3, 4, 5}
```

```
In [67]: a&b&c
```

```
Out[67]: set()
```

```
In [ ]: #Difference OR - symbol
```

```
In [68]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
         c = {8,9,10}
         print(a)
         print(b)
         print(c)
```

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

```
In [69]: a.difference(b)
```

```
Out[69]: {1, 2, 3}
```

```
In [70]: b.difference(a)
```

```
Out[70]: {6, 7, 8}
```

```
In [71]: b-c
```

```
Out[71]: {4, 5, 6, 7}
```

```
In [72]: c-b
```

```
Out[72]: {9, 10}
```

```
In [73]: a-b-c
```

```
Out[73]: {1, 2, 3}
```

```
In [ ]: #Means from a remove b and c elements
```

```
In [74]: a-a
```

```
Out[74]: set()
```

```
In [ ]:
```