

TUPLE

```
In [1]: 1 t=()
        2 t
```

Out[1]: ()

```
In [2]: 1 type(t)
        2
```

Out[2]: tuple

```
In [3]: 1 t=(10,20,30)
        2 t
```

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

```
In [4]: 1 t.count(20)
```

Out[4]: 1

```
In [5]: 1 t=(10,20,20,30)
        2 t.count(20)           ## duplicates are allowed
```

Out[5]: 2

```
In [8]: 1 t=(10,20,20,30,40.5,10+5j,"sneha")
        2 t.count(20)           ## all types of data is taken
```

Out[8]: 2

```
In [9]: 1 print(t)
        (10, 20, 20, 30, 40.5, (10+5j), 'sneha')
```

```
In [10]: 1 print(len(t))
```

7

```
In [12]: 1 t[0]    ## 0-index val
```

Out[12]: 10

```
In [13]: 1 t[0]= 100 ## list is mutable but tuple is not mutable
```

```
-----
TypeError                                Traceback (most recent call last)
Input In [13], in <cell line: 1>()
----> 1 t[0]= 100

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

```
In [16]: 1 bank_acc=(123,"sneha",100000)
         2 bank_acc
```

Out[16]: (123, 'sneha', 100000)

```
In [17]: 1 bank_acc[0]=200000
```

```
-----
TypeError                                Traceback (most recent call last)
Input In [17], in <cell line: 1>()
----> 1 bank_acc[0]=200000

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

```
In [18]: 1 t
```

Out[18]: (10, 20, 20, 30, 40.5, (10+5j), 'sneha')

```
In [19]: 1 t=(10,20,30)
         2 t
```

Out[19]: (10, 20, 30)

```
In [63]: 1 t1=t*3  ## the set(10,20,30) are repeated
         2 t1
```

Out[63]: (10, 20, 30, 10, 20, 30, 10, 20, 30)

```
In [25]: 1 ##LIST ----Mutable
         2 #duplicate is allowed
         3 #append(),copy(),insert(), extend(), pop()
         4 #removet the element
         5 #multiple dat type ina list
         6 # pop()-----> defaultlu last element is removed
         7 # remove()----->occurence of 1st dulpicate value is rempoved
         8 #expandable
```

```
In [26]: 1 #Tuple:
         2 #Immutable
         3 # Not expandable
```

SET

```
In [27]: 1 s={}
         2 s
```

Out[27]: {}

```
In [28]: 1 type(s)
```

Out[28]: dict

```
In [29]: 1 s1=set()  
        2 s1
```

Out[29]: set()

```
In [31]: 1 s2={10,30,40,25,40,30}  
        2 s2                                     ##by default print the values ordered way
```

Out[31]: {10, 25, 30, 40}

```
In [32]: 1 #set duplicates are not allowed
```

```
In [33]: 1 s2
```

Out[33]: {10, 25, 30, 40}

```
In [34]: 1 s3=s2.copy()  
        2 s3
```

Out[34]: {10, 25, 30, 40}

```
In [35]: 1 s3
```

Out[35]: {10, 25, 30, 40}

```
In [62]: 1 # Add() ---> adds the values or elemnet to the set randomly  
        2 # Append()-----> FOr List not for set
```

```
In [37]: 1 s3.add(3.4)  
        2 s3
```

Out[37]: {3.4, 10, 25, 30, 40}

```
In [38]: 1 s3.add("sneha")  
        2 s3
```

Out[38]: {10, 25, 3.4, 30, 40, 'sneha'}

```
In [39]: 1 s3.add(2+3j)  
        2 s3
```

Out[39]: {(2+3j), 10, 25, 3.4, 30, 40, 'sneha'}

```
In [42]: 1 s3.add(True)  
        2 s3
```

Out[42]: {(2+3j), 10, 25, 3.4, 30, 40, True, 'sneha'}

```
In [43]: 1 #set doesn't allow 2 arguments bcz it is dictionary
```

In [44]: 1 *#clear-----> clear the all the values from the set*

In [45]: 1 s3

Out[45]: {(2+3j), 10, 25, 3.4, 30, 40, True, 'sneha'}

In [46]: 1 s3.remove(2000) *## the val 2000 is not available*

KeyError Traceback (most recent call last)
 Input In [46], in <cell line: 1>()
 -----> 1 s3.remove(2000)
KeyError: 2000

In [50]: 1 s3.remove(30)
 2 s3

Out[50]: {(2+3j), 10, 3.4, 40, True, 'sneha'}

In [51]: 1 s3

Out[51]: {(2+3j), 10, 3.4, 40, True, 'sneha'}

In [53]: 1 *# DISCARD -----> this doesn't throw the error*

In [54]: 1 s3.discard(2000)
 2 s3

Out[54]: {(2+3j), 10, 3.4, 40, True, 'sneha'}

Remove() V/s Discard() done above

In [55]: 1 s3.pop()
 2 s3

Out[55]: {(2+3j), 10, 3.4, 40, 'sneha'}

In [56]: 1 s3.pop()
 2 s3

Out[56]: {(2+3j), 10, 40, 'sneha'}

In [57]: 1 *# pop()-----> removes the random val in set*
 2 *# when index val is given to pop then ERROR*

```
In [60]: 1 s3.pop(:)      #Indexing Slicing is Allowed
          2 s3
```

```
Input In [60]
s3.pop(:)      #Indexing Slicing is Allowed
    ^
```

SyntaxError: invalid syntax

```
In [61]: 1 s3.pop(0)
          2 s3
```

```
-----
TypeError                                Traceback (most recent call last)
Input In [61], in <cell line: 1>()
----> 1 s3.pop(0)
      2 s3
```

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

SET MEMBERSHIP

SET OPERATIONS

```
In [89]: 1 #SET OPERATIONS
          2
          3 #UNION
          4 #INTERSECTION
          5 #DIFFERENCE
```

UNION====="|"

```
In [65]: 1
          2 a={1,2,3,4,5,6}
          3 b={4,5,6,7,8}
          4 c={8,9,10}
```

```
In [66]: 1 type(c)
```

Out[66]: set

```
In [67]: 1 a.union(b)
```

Out[67]: {1, 2, 3, 4, 5, 6, 7, 8}

```
In [68]: 1 b.union(c)
```

Out[68]: {4, 5, 6, 7, 8, 9, 10}

In [69]: 1 b.union(a,c)

Out[69]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [70]: 1 print(a)
2 print(b)
3 print(c)

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

In [71]: 1 a|b|c

Out[71]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [72]: 1 a|c|b

Out[72]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [74]: 1 a={1,2,3,4,5,6}
2 b={4,5,6,7,8}
3 c={4,3,1,8,9,10}

In [76]: 1 a|c|b *## structured way*

Out[76]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

Inetersection =====&"

In [79]: 1 a={1,2,3,4,5,6}
2 b={4,5,6,7,8}
3 c={4,3,1,8,9,10}

In [80]: 1 a.intersection(b)

Out[80]: {4, 5, 6}

In [82]: 1 c.intersection(a)

Out[82]: {1, 3, 4}

In [83]: 1 c&a

Out[83]: {1, 3, 4}

DIFFERNCE----->"-

```
In [84]: 1 a={1,2,3,4,5,6}
          2 b={4,5,6,7,8}
          3 c={4,3,1,8,9,10}
```

```
In [85]: 1 a.difference(b)    ## common elemnets are disabled
```

```
Out[85]: {1, 2, 3}
```

```
In [87]: 1 b.difference(c)
```

```
Out[87]: {5, 6, 7}
```

```
In [88]: 1 b-c
```

```
Out[88]: {5, 6, 7}
```

```
In [ ]: 1
```