Tuple

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

```
Out[1]: ()
```

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

```
Out[2]: tuple
```

```
In [3]: t = (12,15,16)
        t
```

```
Out[3]: (12, 15, 16)
```

```
In [4]: t.count(16)
```

```
Out[4]: 1
```

```
In [5]: t2 = (10,20,40,10,60)
        t2
```

```
Out[5]: (10, 20, 40, 10, 60)
```

```
In [6]: t2.count(10)
```

```
Out[6]: 2
```

```
In [7]: t2.index(40)
```

```
Out[7]: 2
```

```
In [8]: print(t)
        print(t2)
```

```
(12, 15, 16)
(10, 20, 40, 10, 60)
```

```
In [9]: print(len(t))
        print(len(t2))
```

```
3
5
```

```
In [10]: t[1]
```

```
Out[10]: 15
```

```
In [11]: t[1] = 14 # tuple is immutable hence we cannot change valeu in tuple
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[11], line 1
----> 1 t[1] = 14

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

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

Out[12]:
```
(12, 15, 16)
```

In [14]:
```
t2 = t*4
t2
```

Out[14]:
```
(12, 15, 16, 12, 15, 16, 12, 15, 16, 12, 15, 16)
```

In [15]:
```
t
```

Out[15]:
```
(12, 15, 16)
```

In [16]:
```
for i in t:
    print(i)
```

```
12
15
16
```

In [17]:
```
for i in enumerate(t):
    print(i)
```

```
(0, 12)
(1, 15)
(2, 16)
```

tuple

immutable (unchangeble) duplicate is allowed remove is not allowed only two function will work(.index,.count)

In [ ]:
```

```

set

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

Out[18]:
```
{}
```

In [19]:
```
type(s)
```

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

In [20]:
```
s1 = set()
s1
```

Out[20]:
```
set()
```

In [21]:
```
s2 = {10,20,30,40,20,50,10}
s2
```

Out[21]:
```
{10, 20, 30, 40, 50}
```

In [22]:
```
type(s2)
```

Out[22]:
```
set
```

In [23]:
```
s2
```

Out[23]:
```
{10, 20, 30, 40, 50}
```

In [24]:
```
s3 = s2.copy()
s3
```

Out[24]:
```
{10, 20, 30, 40, 50}
```

In [25]:
```
s3
```

Out[25]:
```
{10, 20, 30, 40, 50}
```

In [26]:
```
s3.add(4.6)
s3
```

Out[26]:
```
{4.6, 10, 20, 30, 40, 50}
```

In [28]:
```
s3.add('python')
s3
```

Out[28]:
```
{10, 20, 30, 4.6, 40, 50, 'python'}
```

In [29]:
```
print(s)
print(s1)
print(s2)
print(s3)
```

```
{}
set()
{50, 20, 40, 10, 30}
{50, 'python', 20, 4.6, 40, 10, 30}
```

In [30]:
```
print(len(s))
print(len(s1))
print(len(s2))
print(len(s3))
```

```
0
0
5
7
```

In [31]:
```
s3.remove(40)
s3
```

Out[31]:
```
{10, 20, 30, 4.6, 50, 'python'}
```

In [32]:
```
s3.discard(10)
s3
```

Out[32]:
```
{20, 30, 4.6, 50, 'python'}
```

In [33]:
```
s3
```

Out[33]:
```
{20, 30, 4.6, 50, 'python'}
```

In [34]:
```
s3.remove(100) #returns error when element is not present in the set
s3
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[34], line 1
----> 1 s3.remove(100)
      2 s3

KeyError: 100
```

In [36]:
```
s3.discard(100) #discard never give error
s3
```

Out[36]:
```
{20, 30, 4.6, 50, 'python'}
```

In [37]:
```
s3.pop() #delete random element from the set
```

Out[37]:
```
50
```

In [38]:
```
s3.pop()
```

Out[38]:
```
'python'
```

In [39]:
```
s3
```

Out[39]:
```
{4.6, 20, 30}
```

In [40]:
```
s3.pop(0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[40], line 1
----> 1 s3.pop(0)

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

In [43]:
```
s3.[:] # slicing and indexing is not allowed in set
```

```
  Cell In[43], line 1
    s3.[:] # slicing and indexing is not allowed in set
       ^
SyntaxError: invalid syntax
```

In [44]:
```
s3[2] # slicing and indexing is not allowed in set
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[44], line 1
----> 1 s3[2]

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

```
In [45]: s3
```

```
Out[45]: {4.6, 20, 30}
```

```
In [46]: 20 in s3
```

```
Out[46]: True
```

set operation

```
In [47]: a = {11,12,13,14,15}
         b = {14,15,16,17}
         c = {17,18,19,20}
```

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

```
Out[48]: set
```

```
In [50]: a.union (b)
```

```
Out[50]: {11, 12, 13, 14, 15, 16, 17}
```

```
In [51]: a.union(b,c)
```

```
Out[51]: {11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

```
In [52]: print(a)
         print(b)
         print(c)
```

```
{11, 12, 13, 14, 15}
{16, 17, 14, 15}
{17, 18, 19, 20}
```

```
In [54]: a | b
```

```
Out[54]: {11, 12, 13, 14, 15, 16, 17}
```

```
In [55]: b | c
```

```
Out[55]: {14, 15, 16, 17, 18, 19, 20}
```

```
In [56]: a | b | c
```

```
Out[56]: {11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

```
In [57]: a | c
```

```
Out[57]: {11, 12, 13, 14, 15, 17, 18, 19, 20}
```

intersection

```
In [58]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
         c = {7,8,9,10}
```

```
In [59]: a.intersection(b)
```

```
Out[59]: {4, 5}
```

```
In [60]: b.intersection(c)
```

```
Out[60]: {7, 8}
```

```
In [62]: a & b
```

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

```
In [63]: b & c
```

```
Out[63]: {7, 8}
```

Diffirence

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

```
In [66]: a.difference(b)
```

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

```
In [67]: b.difference(a)
```

```
Out[67]: {6, 7, 8}
```

```
In [68]: b-c
```

```
Out[68]: {4, 5, 6}
```

```
In [69]: c-a
```

```
Out[69]: {7, 8, 9, 10}
```

```
In [70]: a-b-c
```

```
Out[70]: {1, 2, 3}
```

```
In [ ]:
```

In [ ]: