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

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

In [2]:
```
t1 = tuple()
t1
```

Out[2]:
```
()
```

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

Out[3]:
```
tuple
```

In [4]:
```
t1 = (10,20,30,40)
```

In [5]:
```
t1
```

Out[5]:
```
(10, 20, 30, 40)
```

In [6]:
```
t2 = (10,20.5,1+2J,'nit',True)
t2
```

Out[6]:
```
(10, 20.5, (1+2j), 'nit', True)
```

In [7]:
```
t2.append(10)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[7], line 1
----> 1 t2.append(10)

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

In [8]:
```
t1
```

Out[8]:
```
(10, 20, 30, 40)
```

In [9]:
```
t1[0]
```

Out[9]:
```
10
```

In [10]:
```
t1[0]= 100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 1
----> 1 t1[0]= 100

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

In [11]:
```
t2 = (10,20,10,20,40)
```

In [12]:
```
t2
```

Out[12]:
```
(10, 20, 10, 20, 40)
```

In [13]:
```
t1[:]
```

Out[13]:
```
(10, 20, 30, 40)
```

In [14]:
```
t2[1:]
```

Out[14]:
```
(20, 10, 20, 40)
```

In [15]:
```
t2.count(10)
```

Out[15]:
```
2
```

In [16]:
```
t2.count(40)
```

Out[16]:
```
1
```

In [17]:
```
t2.index(20)
```

Out[17]:
```
1
```

In [18]:
```
for i in t2:
    print(i)
```

```
10
20
10
20
40
```

In [19]:
```
l = [1,200,2]
l
```

Out[19]:
```
[1, 200, 2]
```

In [21]:
```
for i in enumerate(l): # Enumerate retunr the values with Indexes
    print(i)
```

```
(0, 1)
(1, 200)
(2, 2)
```

In [22]:
```
for i in enumerate(t1):
    print(i)
```

```
(0, 10)
(1, 20)
(2, 30)
(3, 40)
```

In [23]:
```
t3 = t1.copy()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[23], line 1
----> 1 t3 = t1.copy()

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

In [24]:
```
#set
s={}
s
```

Out[24]:
```
{}
```

In [25]:
```
type(s)
```

Out[25]:
```
dict
```

In [26]:
```
s1 = set()
```

In [27]:
```
type(s1)
```

Out[27]:
```
set
```

In [28]:
```
s1 = {5,3,100,59,29,20}
s1
```

Out[28]:
```
{3, 5, 20, 29, 59, 100}
```

In [29]:
```
s2 = {'z','m','b','a','x','v','l'}
s2
```

Out[29]:
```
{'a', 'b', 'l', 'm', 'v', 'x', 'z'}
```

In [31]:
```
s3 = {2,'z',4.5,1+2j,True}
s3
```

Out[31]:
```
{(1+2j), 2, 4.5, True, 'z'}
```

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

```
{3, 100, 5, 20, 59, 29}
{'l', 'x', 'v', 'm', 'a', 'z', 'b'}
{True, 2, 4.5, 'z', (1+2j)}
```

In [33]:
```
print(type(s1))
print(type(s2))
print(type(s3))
```

```
<class 'set'>
<class 'set'>
<class 'set'>
```

In [34]:
```
s1.add(200)
```

In [35]:
```
s1
```

Out[35]:
```
{3, 5, 20, 29, 59, 100, 200}
```

In [36]:
```
s1.add(200)
```

In [37]:
```
s1
```

Out[37]:
```
{3, 5, 20, 29, 59, 100, 200}
```

In [38]:
```
s4 = s1.copy()
```

In [39]:
```
s4
```

Out[39]:
```
{3, 5, 20, 29, 59, 100, 200}
```

In [40]:
```
s1[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[40], line 1
----> 1 s1[0]

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

In [41]:
```
s4.clear()
```

In [42]:
```
s4
```

Out[42]:
```
set()
```

In [45]:
```
del s4
```

In [43]:
```
s1.pop()
```

Out[43]:
```
3
```

In [46]:
```
s4 ={1,2,4}
```

In [47]:
```
id(s4)
```

Out[47]:
```
4581142848
```

In [48]:
```
s3.remove(1+2j)
```

In [49]:
```
s3
```

Out[49]:
```
{2, 4.5, True, 'z'}
```

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