Set Creation

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

```
In [2]:  s
```

```
Out[2]:  {}
```

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

```
Out[3]:  dict
```

```
In [4]:  s1 = set()
```

```
In [5]:  s1
```

```
Out[5]:  set()
```

```
In [6]:  type(s1)
```

```
Out[6]:  set
```

```
In [7]:  s1.add(10)
```

```
In [9]:  s1.add(20)
```

```
In [10]:  s1
```

```
Out[10]:  {10, 20}
```

```
In [11]:  s1.add(40)
          s1.add(100)
          s1.add(59)
          s1.add(68)
          s1
```

```
Out[11]:  {10, 20, 40, 59, 68, 100}
```

Indexing

```
In [12]:  s1(0) # In set indexing is not possible
```

```
---------------------------------------------------------------------
TypeError                                Traceback (most recent call last)
Cell In[12], line 1
----> 1 s1(0)

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

Slicing

```
In [13]:  s1[:] #in set slicing is not possible
```

```
---------------------------------------------------------------------
TypeError                                Traceback (most recent call last)
Cell In[13], line 1
----> 1 s1[:]

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

```
In [14]:  s1
```

```
Out[14]:  {10, 20, 40, 59, 68, 100}
```

In [15]:
```python
s1.add([1,2,3])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 s1.add([1,2,3])

TypeError: unhashable type: 'list'
```

In [16]:
```python
s2 = set()
s2
```

Out[16]:
```
set()
```

In [17]:
```python
s2.add(10)
s2.add(1.2)
s2.add(1+2j)
s2.add(True)
s2.add('nit')
```

In [18]:
```python
s2 #Set is orderd
```

Out[18]:
```
{(1+2j), 1.2, 10, True, 'nit'}
```

In [19]:
```python
print(s1)
print(s2)
```

```
{100, 68, 40, 10, 20, 59}
{1.2, True, (1+2j), 10, 'nit'}
```

In [20]:
```python
id(s1) == id(s2)
```

Out[20]:
```
False
```

In [21]:
```python
s3 = s2.copy()
s3
```

Out[21]:
```
{(1+2j), 1.2, 10, True, 'nit'}
```

In [22]:
```python
s2 == s3
```

Out[22]:
```
True
```

In [23]:
```python
print(s1)
print(s2)
print(s3)
```

```
{100, 68, 40, 10, 20, 59}
{1.2, True, (1+2j), 10, 'nit'}
{1.2, True, 'nit', (1+2j), 10}
```

In [24]:
```python
s2.pop()#Removes random value from the set
```

Out[24]:
```
1.2
```

In [25]:
```python
s2
```

Out[25]:
```
{(1+2j), 10, True, 'nit'}
```

In [26]:
```python
s
```

Out[26]:
```
{}
```

Remove

In [27]:
```python
s3
```

Out[27]:
```
{(1+2j), 1.2, 10, True, 'nit'}
```

In [28]:
```python
s3.remove(1+2j)
```

In [29]:
```python
s3
```

Out[29]:
```
{1.2, 10, True, 'nit'}
```

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

In [31]: `s3`

Out[31]: `{True, 1.2, 10}`

In [32]: `s3.remove(1000)`

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[32], line 1
----> 1 s3.remove(1000)

KeyError: 1000
```

In [33]: `s3.discard(1000)#discard never gives you error. if number is available it will delete otherwise it will calm dow`

In [34]: `s3`

Out[34]: `{True, 1.2, 10}`

In [35]: `s3.discard(True) #removes True from the list`

In [36]: `s3`

Out[36]: `{1.2, 10}`

In [37]:
```python
print(s1)
print(s2)
print(s3)
```

```
{100, 68, 40, 10, 20, 59}
{True, (1+2j), 10, 'nit'}
{1.2, 10}
```

Looping

In [38]:
```python
for i in enumerate(s1):
    print(i)
```

```
(0, 100)
(1, 68)
(2, 40)
(3, 10)
(4, 20)
(5, 59)
```

Union

In [39]:
```python
a = {1,2,3,4,5}
b = {4,5,6,7,8}
c = {8,9,10}
```

In [40]: `a.union(b)`

Out[40]: `{1, 2, 3, 4, 5, 6, 7, 8}`

In [41]: `a|c`

Out[41]: `{1, 2, 3, 4, 5, 8, 9, 10}`

In [42]:
```python
a = {1,2,3,4,5}
b = {4,5,6,7,8}
c = {8,9,10}
```

In [43]: `b|c`

Out[43]: `{4, 5, 6, 7, 8, 9, 10}`

In [44]: `a|b|c`

Out[44]: `{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}`

In [45]:
```python
print(a)
print(b)
print(c)
```

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

In [46]:
```python
a.difference(b)
```

Out[46]: {1, 2, 3}

In [47]:
```python
a.difference(c)
```

Out[47]: {1, 2, 3, 4, 5}

In [48]:
```python
c.difference(b)
```

Out[48]: {9, 10}