

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

```
Out[1]: {}
```

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

```
Out[2]: dict
```

```
In [3]: s1 = set()  
s1
```

```
Out[3]: set()
```

```
In [4]: type(s1)
```

```
Out[4]: set
```

```
In [7]: s1.add(20)  
s1
```

```
Out[7]: {20}
```

```
In [8]: s1.add(30,10)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[8], line 1  
----> 1 s1.add(30,10)  
  
TypeError: set.add() takes exactly one argument (2 given)
```

```
In [9]: s1
```

```
Out[9]: {20}
```

```
In [10]: s1.add(10)  
s1.add(100)  
s1.add(25)  
s1
```

```
Out[10]: {10, 20, 25, 100}
```

```
In [11]: s1.add(10)  
s1
```

```
Out[11]: {10, 20, 25, 100}
```

```
In [12]: s1[10]
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[12], line 1  
----> 1 s1[10]  
  
TypeError: 'set' object is not subscriptable
```

```
In [13]: s1
```

```
Out[13]: {10, 20, 25, 100}
```

SLICING

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

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[14], line 1  
----> 1 s1[:]  
  
TypeError: 'set' object is not subscriptable
```

```
In [15]: s1
```

```
Out[15]: {10, 20, 25, 100}
```

```
In [16]: s1.add([1,2,3])
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[16], line 1  
----> 1 s1.add([1,2,3])  
  
TypeError: unhashable type: 'list'
```

```
In [17]: s2 = set()  
s2
```

```
Out[17]: set()
```

```
In [19]: s2.add(10)  
s2.add(1.2)  
s2.add(1+2j)  
s2.add(True)  
s2.add('nit')  
s2
```

```
Out[19]: {(1+2j), 1.2, 10, True, 'nit'}
```

```
In [21]: print(s1)  
print(s2)
```

```
{100, 25, 10, 20}  
{1.2, True, (1+2j), 10, 'nit'}
```

```
In [23]: id(s1) == id(s2)
```

```
Out[23]: False
```

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

```
Out[24]: {(1+2j), 1.2, 10, True, 'nit'}
```

```
In [25]: s2 == s3
```

```
Out[25]: True
```

```
In [26]: print(s1)  
print(s2)  
print(s3)
```

```
{100, 25, 10, 20}  
{1.2, True, (1+2j), 10, 'nit'}  
{1.2, True, (1+2j), 'nit', 10}
```

```
In [27]: s2
```

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

```
In [28]: s2.pop()
```

```
Out[28]: 1.2
```

```
In [29]: s2.pop()
```

```
Out[29]: True
```

```
In [30]: s2
```

```
Out[30]: {(1+2j), 10, 'nit'}
```

```
In [31]: s1
```

```
Out[31]: {10, 20, 25, 100}
```

```
In [32]: s2
```

```
Out[32]: {(1+2j), 10, 'nit'}
```

```
In [33]: s
```

```
Out[33]: {}
```

```
In [34]: s3
```

```
Out[34]: {(1+2j), 1.2, 10, True, 'nit'}
```

In [35]: `s3.remove(1000)`

```
-----
KeyError                                Traceback (most recent call last)
Cell In[35], line 1
----> 1 s3.remove(1000)

KeyError: 1000
```

In [36]: `s3.discard(1000)`

In [37]: `s3.discard(True)`

In [38]: `s3`

Out[38]: `{(1+2j), 1.2, 10, 'nit'}`

In [39]: `print(s2)`  
`print(s3)`

```
{(1+2j), 10, 'nit'}
{1.2, (1+2j), 'nit', 10}
```

In [40]: `for i in s1:`  
`print(i)`

```
100
25
10
20
```

In [41]: `for i in enumerate(s1):`  
`print(i)`

```
(0, 100)
(1, 25)
(2, 10)
(3, 20)
```

### SET OPERATIONS

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

Out[42]: `{1, 2, 3, 4, 5, 6, 7, 8}`

In [43]: `a | c`

Out[43]: `{1, 2, 3, 4, 5, 8, 9, 10}`

In [44]: `b|c`

Out[44]: {4, 5, 6, 7, 8, 9, 10}

In [45]: `c|a`

Out[45]: {1, 2, 3, 4, 5, 8, 9, 10}

In [46]: `a.difference(c)`

Out[46]: {1, 2, 3, 4, 5}

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

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

In [48]: `print(a)`  
`print(b)`  
`print(c)`

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

In [49]: `c.difference(a)`

Out[49]: {8, 9, 10}

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