

# Set {}

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

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

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

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

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

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

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

```
In [5]: s1.add(20)
```

```
In [6]: s1
```

```
Out[6]: {20}
```

```
In [7]: s1.add(30,40)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[7], line 1  
----> 1 s1.add(30,40)  
  
TypeError: set.add() takes exactly one argument (2 given)
```

```
In [8]: s1
```

```
Out[8]: {20}
```

```
In [9]: s1.add(10)  
s1.add(100)
```

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

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

```
In [11]: s1.add(25)
```

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

```
Out[12]: {10, 20, 25, 100}
```

```
In [13]: s1.add(10)
```

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

```
Out[14]: {10, 20, 25, 100}
```

```
In [15]: s1[0]
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[15], line 1  
----> 1 s1[0]  
  
TypeError: 'set' object is not subscriptable
```

```
In [16]: s1[:]
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[16], line 1  
----> 1 s1[:]  
  
TypeError: 'set' object is not subscriptable
```

```
In [17]: s1.add([1,2,3])
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[17], line 1  
----> 1 s1.add([1,2,3])  
  
TypeError: unhashable type: 'list'
```

```
In [18]: s2=set()  
s2
```

```
Out[18]: set()
```

```
In [19]: s2.add(10)  
s2.add(1.2)
```

```
In [20]: s2
```

```
Out[20]: {1.2, 10}
```

```
In [21]: s2.add(1+2j)  
s2.add(True)
```

```
In [22]: s2
```

```
Out[22]: {(1+2j), 1.2, 10, True}
```

```
In [23]: s2.add('nit')
```

```
s2.add('False')
```

```
In [24]: s2
```

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

```
In [25]: s2.add(10+10j)
```

```
In [26]: s2
```

```
Out[26]: {(1+2j), (10+10j), 1.2, 10, 'False', True, 'nit'}
```

```
In [27]: s2.add(15.5)
```

```
In [28]: s2
```

```
Out[28]: {(1+2j), (10+10j), 1.2, 10, 15.5, 'False', True, 'nit'}
```

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

```
{100, 25, 10, 20}
```

```
{1.2, True, (1+2j), (10+10j), 'False', 10, 15.5, 'nit'}
```

```
In [30]: s1
```

```
Out[30]: {10, 20, 25, 100}
```

```
In [31]: print(s1)
```

```
{100, 25, 10, 20}
```

```
In [32]: id(s1)==id(s2)
```

```
Out[32]: False
```

```
In [33]: s3=s2.copy()
```

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

```
Out[34]: {(1+2j), (10+10j), 1.2, 10, 15.5, 'False', True, 'nit'}
```

```
In [35]: s2==s3
```

```
Out[35]: True
```

```
In [36]: print(s1)
         print(s2)
         print(s3)
```

```
{100, 25, 10, 20}
```

```
{1.2, True, (1+2j), (10+10j), 'False', 10, 15.5, 'nit'}
```

```
{1.2, True, (1+2j), (10+10j), 'False', 10, 15.5, 'nit'}
```

```
In [37]: s2
```

```
Out[37]: {(1+2j), (10+10j), 1.2, 10, 15.5, 'False', True, 'nit'}
```

```
In [38]: s2.pop()
```

```
Out[38]: 1.2
```

```
In [39]: s2
```

```
Out[39]: {(1+2j), (10+10j), 10, 15.5, 'False', True, 'nit'}
```

```
In [40]: s2.pop()
```

```
Out[40]: True
```

```
In [41]: s2
```

```
Out[41]: {(1+2j), (10+10j), 10, 15.5, 'False', 'nit'}
```

```
In [42]: s2.remove((1+2j))
```

```
In [43]: s2
```

```
Out[43]: {(10+10j), 10, 15.5, 'False', 'nit'}
```

```
In [44]: s2.remove(100)
```

```
-----  
KeyError                                Traceback (most recent call last)  
Cell In[44], line 1  
----> 1 s2.remove(100)  
  
KeyError: 100
```

```
In [45]: s2.discard(100)
```

```
In [46]: s2.discard(10+10j)
```

```
In [47]: s2
```

```
Out[47]: {10, 15.5, 'False', 'nit'}
```

```
In [48]: print(s1)  
         print(s2)  
         print(s3)
```

```
{100, 25, 10, 20}  
{'False', 10, 15.5, 'nit'}  
{1.2, True, (1+2j), (10+10j), 'False', 10, 15.5, 'nit'}
```

## Looping in set

```
In [49]: for i in s1:  
        print(i)
```

```
100  
25  
10  
20
```

## enumerate in set

```
In [50]: for i in enumerate(s1):  
        print(i)
```

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

## Union

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

```
In [52]: a.union(b)
```

```
Out[52]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [53]: a | b
```

```
Out[53]: {1, 2, 3, 4, 5, 6, 7, 8}
```

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

```
Out[54]: {4, 5, 6, 7, 8, 9, 10}
```

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

```
Out[55]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [56]: print(a)  
        print(b)  
        print(c)
```

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

## difference()

```
In [57]: a.difference(b)
```

```
Out[57]: {1, 2, 3}
```

```
In [58]: a.difference(c)
```

```
Out[58]: {1, 2, 3, 4, 5}
```

```
In [59]: c.difference(a)
```

```
Out[59]: {8, 9, 10}
```

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

```
Out[60]: {9, 10}
```

```
In [61]: c.difference(c)
```

```
Out[61]: set()
```

```
In [4]: print(a)
        print(b)
        print(c)
```

```
{1, 2, 3, 4, 5}
```

```
{4, 5, 6, 7, 8}
```

```
{8, 9, 10}
```

```
In [5]: c - a
```

```
Out[5]: {8, 9, 10}
```

## Symmetric\_difference

```
In [6]: a.symmetric_difference(b)
```

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

```
In [7]: a ^ b
```

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

```
In [8]: b ^ a
```

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

```
In [10]: a ^ b ^ c
```

```
Out[10]: {1, 2, 3, 6, 7, 9, 10}
```

## Symmetric\_difference\_update

```
In [11]: print(a)
         print(b)
         print(c)
```

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

```
In [12]: a.symmetric_difference_update(b)
```

```
In [13]: print(a)
         print(b)
         print(c)
```

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

## intersection()

```
In [14]: a.intersection(b)
```

```
Out[14]: {6, 7, 8}
```

```
In [15]: a & b
```

```
Out[15]: {6, 7, 8}
```

```
In [16]: b & c
```

```
Out[16]: {8}
```

```
In [18]: a & b & c
```

```
Out[18]: {8}
```

```
In [19]: print(a)
         print(b)
         print(c)
```

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

## intersection\_update()

```
In [20]: c.intersection_update(b)
```

```
In [21]: print(a)
         print(b)
         print(c)
```

```
{1, 2, 3, 6, 7, 8}
```

```
{4, 5, 6, 7, 8}
```

```
{8}
```

## issubset(),issuperset(),isdisjoint()

```
In [22]: A1={1,2,3,4,5,6,7,8,9}
         B1={3,4,5,6,7,8}
         C1={10,20,30,40}
```

```
In [23]: B1.issubset(A1)
```

```
Out[23]: True
```

```
In [24]: A1.issuperset(B1)
```

```
Out[24]: True
```

```
In [25]: A1.isdisjoint(B1)
```

```
Out[25]: False
```

```
In [26]: C1.isdisjoint(B1)
```

```
Out[26]: True
```

## other funtions

```
In [28]: A1
```

```
Out[28]: {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
In [29]: sum(A1)
```

```
Out[29]: 45
```

```
In [30]: min(A1)
```

```
Out[30]: 1
```

```
In [31]: max(A1)
```



Out[31]: 9

In [32]: `len(A1)`

Out[32]: 9

In [33]: `list(enumerate(A1))`

Out[33]: [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]

In [34]: `D=sorted(A1,reverse=True)`

In [35]: `D`

Out[35]: [9, 8, 7, 6, 5, 4, 3, 2, 1]

In [36]: `sorted(D)`

Out[36]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: