

SET

```
In [2]: s = {}  
s
```

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

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

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

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

```
Out[4]: set()
```

```
In [5]: type(s1)
```

```
Out[5]: set
```

```
In [6]: s1.add(20)
```

```
In [7]: s1
```

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

```
In [8]: s1.add(10)  
s1.add(100)  
s1.add(25)
```

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

```
Out[9]: {10, 20, 25, 100}
```

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

```
In [11]: s1
```

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

```
In [12]: s2 = set()  
s2
```

```
Out[12]: set()
```

```
In [13]: s2.add(10)
s2.add(1.2)
s2.add(1+2j)
s2.add(True)
s2.add('nit')
```

```
In [14]: s2
```

```
Out[14]: {(1+2j), 1.2, 10, True, 'nit'}
```

```
In [15]: print(s1)
print(s2)
```

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

```
In [16]: id(s1) == id(s2)
```

```
Out[16]: False
```

```
In [17]: s3 = s2.copy()
s3
```

```
Out[17]: {(1+2j), 1.2, 10, True, 'nit'}
```

```
In [18]: s3
```

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

```
In [19]: s2 == s3
```

```
Out[19]: True
```

```
In [20]: print(s1)
print(s2)
print(s3)
```

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

```
In [21]: s2
```

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

```
In [22]: s2.pop()
```

```
Out[22]: 1.2
```

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

```
Out[23]: {(1+2j), 10, True, 'nit'}
```

In [24]: s1

Out[24]: {10, 20, 25, 100}

In [25]: s2

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

In [26]: s

Out[26]: {}

In [27]: s3

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

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

In [29]: s3

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

In [30]: s3.discard(1000)

In [31]: s3.discard(True)

In [32]: s3

Out[32]: {1.2, 10, 'nit'}

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

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

In [34]: for i in s1:
print(i)

100
25
10
20

In [35]: for i in enumerate(s1):
print(i)

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

```
In [36]: a = {1,2,3,4,5}
         b = {4,5,6,7,8}
         c = {8,9,10}
```

```
In [37]: a.union(b)
```

```
Out[37]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [38]: a | c
```

```
Out[38]: {1, 2, 3, 4, 5, 8, 9, 10}
```

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

```
In [40]: b | c
```

```
Out[40]: {4, 5, 6, 7, 8, 9, 10}
```

```
In [41]: a | b | c
```

```
Out[41]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [42]: print(a)
         print(b)
         print(c)
```

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

```
In [43]: a.difference(b)
```

```
Out[43]: {1, 2, 3}
```

```
In [44]: a.difference(c)
```

```
Out[44]: {1, 2, 3, 4, 5}
```

```
In [45]: c.difference(a)
```

```
Out[45]: {8, 9, 10}
```

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

```
Out[46]: {9, 10}
```

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

```
Out[47]: set()
```

```
In [48]: myset = {1,2,3,4,5}
myset
```

```
Out[48]: {1, 2, 3, 4, 5}
```

```
In [49]: len(myset)
```

```
Out[49]: 5
```

```
In [50]: my_set = {1,1,2,2,3,4,5,5}
my_set
```

```
Out[50]: {1, 2, 3, 4, 5}
```

```
In [51]: myset1 = {1.79,2.08,3.99,4.56,5.45} # Set of float numbers
myset1
```

```
Out[51]: {1.79, 2.08, 3.99, 4.56, 5.45}
```

```
In [52]: myset2 = {'Janhavi' , 'Ketki' , 'Krutika'} # Set of Strings
myset2
```

```
Out[52]: {'Janhavi', 'Ketki', 'Krutika'}
```

```
In [53]: myset3 = {10,20, "Hola", (11, 22, 32)} # Mixed datatypes
myset3
```

```
Out[53]: {(11, 22, 32), 10, 20, 'Hola'}
```

```
In [55]: myset4 = set()
print(type(myset4))
```

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

```
In [56]: my_set1 = set(('one' , 'two' , 'three' , 'four'))
my_set1
```

```
Out[56]: {'four', 'one', 'three', 'two'}
```

Loop through a set

```
In [59]: myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}

for i in myset:
    print(i)
```

```
two
eight
seven
four
six
one
three
five
```

```
In [60]: for i in enumerate(myset):  
         print(i)
```

```
(0, 'two')  
(1, 'eight')  
(2, 'seven')  
(3, 'four')  
(4, 'six')  
(5, 'one')  
(6, 'three')  
(7, 'five')
```

Set Membership

```
In [61]: myset
```

```
Out[61]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [63]: 'one' in myset
```

```
Out[63]: True
```

```
In [64]: 'ten' in myset
```

```
Out[64]: False
```

```
In [65]: if 'three' in myset: # Check if 'three' exist in the set  
         print('Three is present in the set')  
     else:  
         print('Three is not present in the set')
```

Three is present in the set

```
In [66]: if 'eleven' in myset: # Check if 'eleven' exist in the list  
         print('eleven is present in the set')  
     else:  
         print('eleven is not present in the set')
```

eleven is not present in the set

Add & Removes Items

```
In [67]: myset
```

```
Out[67]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [68]: myset.add('Nine')  
         myset
```

```
Out[68]: {'Nine', 'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [69]: myset.update(['TEN' , 'ELEVEN' , 'TWELVE']) # Add multiple item to a set using myset
```

```
Out[69]: {'ELEVEN',  
          'Nine',  
          'TEN',  
          'TWELVE',  
          'eight',  
          'five',  
          'four',  
          'one',  
          'seven',  
          'six',  
          'three',  
          'two'}
```

```
In [71]: myset.remove('Nine') # remove item in a set using remove() method myset
```

```
Out[71]: {'ELEVEN',  
          'TEN',  
          'TWELVE',  
          'eight',  
          'five',  
          'four',  
          'one',  
          'seven',  
          'six',  
          'three',  
          'two'}
```

```
In [72]: myset.discard('TEN') # remove item from a set using discard() method myset
```

```
Out[72]: {'ELEVEN',  
          'TWELVE',  
          'eight',  
          'five',  
          'four',  
          'one',  
          'seven',  
          'six',  
          'three',  
          'two'}
```

```
In [73]: myset.clear() # Delete all items in a set myset
```

```
Out[73]: set()
```

Copy Set

```
In [74]: myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight'}  
myset
```

```
Out[74]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [75]: myset1 = myset # Create a new reference "myset1"  
myset1
```

```
Out[75]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [76]: id(myset) , id(myset1)
```

```
Out[76]: (2966800702976, 2966800702976)
```

```
In [77]: my_set = myset.copy()  
my_set
```

```
Out[77]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [78]: id(my_set)
```

```
Out[78]: 2966800703200
```

```
In [79]: myset.add('nine')  
myset
```

```
Out[79]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [80]: myset1
```

```
Out[80]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}
```

```
In [81]: my_set
```

```
Out[81]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

Set Operation

Union

```
In [83]: A = {1,2,3,4,5}  
B = {4,5,6,7,8}  
C = {8,9,10}
```

```
In [84]: A | B
```

```
Out[84]: {1, 2, 3, 4, 5, 6, 7, 8}
```



```
In [86]: A.union(B)
```

```
Out[86]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [87]: A.union(B, C)
```

```
Out[87]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [88]: A.update(B,C)  
A
```

```
Out[88]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Intersection

```
In [90]: A = {1,2,3,4,5}  
B = {4,5,6,7,8}
```

```
In [91]: A & B
```

```
Out[91]: {4, 5}
```

```
In [92]: A.intersection_update(B)  
A
```

```
Out[92]: {4, 5}
```

Difference

```
In [93]: A = {1,2,3,4,5}  
B = {4,5,6,7,8}
```

```
In [94]: A - B
```

```
Out[94]: {1, 2, 3}
```

```
In [95]: A.difference(B)
```

```
Out[95]: {1, 2, 3}
```

```
In [96]: B - A
```

```
Out[96]: {6, 7, 8}
```

```
In [97]: B.difference(A)
```

```
Out[97]: {6, 7, 8}
```

```
In [98]: B.difference_update(A)
B
```

```
Out[98]: {6, 7, 8}
```

Symmetric Difference

```
In [99]: A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

```
In [100]: A ^ B
```

```
Out[100]: {1, 2, 3, 6, 7, 8}
```

```
In [101]: A.symmetric_difference(B)
```

```
Out[101]: {1, 2, 3, 6, 7, 8}
```

```
In [102]: A.symmetric_difference_update(B)
A
```

```
Out[102]: {1, 2, 3, 6, 7, 8}
```

Subset , Superset & Disjoint

```
In [103]: A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7,8}
C = {10,20,30,40}
```

```
In [104]: B.issubset(A)
```

```
Out[104]: True
```

```
In [105]: A.issuperset(B)
```

```
Out[105]: True
```

```
In [106]: C.isdisjoint(A)
```

```
Out[106]: True
```

```
In [107]: B.isdisjoint(A)
```

```
Out[107]: False
```

Other Builtin functions

```
In [108]: A
```

```
Out[108]: {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
In [109]: sum(A)
```

```
Out[109]: 45
```

```
In [110]: max(A)
```

```
Out[110]: 9
```

```
In [111]: min(A)
```

```
Out[111]: 1
```

```
In [112]: len(A)
```

```
Out[112]: 9
```

```
In [113]: list(enumerate(A))
```

```
Out[113]: [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]
```

```
In [114]: D= sorted(A,reverse=True)  
D
```

```
Out[114]: [9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
In [115]: sorted(D)
```

```
Out[115]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```