

## Sets

- {}
- Sets will remove the duplication

```
In [18]: 1 l = [1,2,4,3,5,2,1,3,4]
          2 print(set(l))
          3 list(set(l))
```

{1, 2, 3, 4, 5}

Out[18]: [1, 2, 3, 4, 5]

```
In [4]: 1 s = {1,2,4,3,5,2,1,3,4}
          2 type(s)
          3 s
```

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

```
In [5]: 1 s = l.copy()
```

Out[5]: [1, 2, 4, 3, 5, 2, 1, 3, 4]

```
In [6]: 1 l # [1,2,3,4,5]
```

Out[6]: [1, 2, 4, 3, 5, 2, 1, 3, 4]

```
In [24]: 1 l=[1, 2, 4, 3, 5,6,2, 1, 3, 4] # 1,2,4,3,5,2,1,3,4
          2 l1 = [] #[1,2,4,3,5]
          3 for i in l: #1,2,4,3,5,2,1
          4     if i not in l1: # 2 not in l1
          5         l1.append(i)
          6 print(l1)
          7 l1.sort()
          8 print(l1)
          9 print(len(l1))
```

[1, 2, 4, 3, 5, 6]

[1, 2, 3, 4, 5, 6]

6

```
In [10]: 1 print(dir(l))
```

...

```
In [20]: 1 l=[1, 2,"a", 3, 5, "a", 1, "b", 4]
          2 set(l)
```

Out[20]: {1, 2, 3, 4, 5, 'a', 'b'}

In [25]: 1 print(dir(set))

```
['_and_', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattr__', '__gt__', '__hash__', '__
iand__', '__init__', '__init_subclass__', '__ior__', '__isub__', '__iter__', '_
_ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__', '__rand
__', '__reduce__', '__reduce_ex__', '__repr__', '__ror__', '__rsub__', '__rxor_
__', '__setattr__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__x
or__', 'add', 'clear', 'copy', 'difference', 'difference_update', 'discard', 'i
ntersection', 'intersection_update', 'isdisjoint', 'issubset', 'issuperset', 'p
op', 'remove', 'symmetric_difference', 'symmetric_difference_update', 'union',
'update']
```

In [28]: 1 s = {1,2,3,4,5}  
2 s.add(10) # to add an element to the set  
3 s  
4

Out[28]: {1, 2, 3, 4, 5, 10}

In [46]: 1 s1 = {1,2,3,4,5}  
2 s2 = {10,20,20,1,2,3}  
3 s1.add("APSSDC") # you have to add single element only  
4 s1

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

In [34]: 1 s1

Out[34]: {1, 2, 3, 4, 5, 'APSSDC'}

In [37]: 1 copy\_s= s.copy()  
2 copy\_s

Out[37]: {1, 2, 3, 4, 5, 10}

In [38]: 1 copy\_s

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

In [42]: 1 print(s1)  
2 print(s2)  
3 s1.difference(s2)  
4  
  
{1, 2, 3, 4, 5, 'APSSDC'}  
{1, 2, 3, 10, 20}

Out[42]: {4, 5, 'APSSDC'}

```
In [47]: 1 print(s1)
          2 print(s2)
          3 print()
          4 print(s1.difference_update(s2))
          5 print()
          6 print(s1)
          7 print(s2)
```

```
{1, 2, 3, 4, 5, 'APSSDC'}
{1, 2, 3, 10, 20}
```

```
None
```

```
{4, 5, 'APSSDC'}
{1, 2, 3, 10, 20}
```

```
In [49]: 1 s1 = {1, 2, 3, 4, 5, 'APSSDC'}
          2 s2 = {1, 2, 3, 10, 20}
          3 s1.intersection(s2)
```

```
Out[49]: {1, 2, 3}
```

```
In [51]: 1 print(s1)
          2 print(s2)
          3 print(s1.intersection_update(s2))
          4 print(s1)
          5 print(s2)
```

```
{1, 2, 3, 4, 5, 'APSSDC'}
{1, 2, 3, 10, 20}
```

```
None
```

```
{1, 2, 3}
{1, 2, 3, 10, 20}
```

```
In [52]: 1 s1.isdisjoint(s2)
```

```
Out[52]: False
```

```
In [53]: 1 print(s1)
          2 print(s2)
```

```
{1, 2, 3}
{1, 2, 3, 10, 20}
```

```
In [54]: 1 print(s1.issubset(s2))
          2 print(s2.issuperset(s1))
```

```
True
```

```
True
```

```
In [56]: 1 s1 = {1, 2, 3, 4, 5, 'APSSDC'}
         2 s2 = {1, 2, 3, 10, 20}
         3 s1.symmetric_difference(s2)
```

Out[56]: {10, 20, 4, 5, 'APSSDC'}

```
In [59]: 1 s1.symmetric_difference_update(s2)
         2 print(s1)
         3 print(s2)
```

{4, 5, 10, 20, 'APSSDC'}  
{1, 2, 3, 10, 20}

```
In [60]: 1 s1.union(s2)
```

Out[60]: {1, 10, 2, 20, 3, 4, 5, 'APSSDC'}

```
In [61]: 1 print(s1)
         2 print(s2)
```

{4, 5, 10, 20, 'APSSDC'}  
{1, 2, 3, 10, 20}

```
In [64]: 1 s1.update(s2) # to join two sets
         2 s1
```

Out[64]: {1, 10, 2, 20, 3, 4, 5, 'APSSDC'}

```
In [65]: 1 l = [1,2,4,3,5,2,1,3,4]
         2 l.pop()
```

Out[65]: 4

```
In [66]: 1 l
```

Out[66]: [1, 2, 4, 3, 5, 2, 1, 3]

```
In [68]: 1 s.pop()
```

Out[68]: 1

```
In [71]: 1 s
```

Out[71]: {2, 3, 4, 10}

```
In [73]: 1 print(s.remove(4))
         2 s
```

None

Out[73]: {2, 3, 10}

```
In [74]: 1 print(s.discard(3))
         2
```

None

```
In [75]: 1 s
```

Out[75]: {2, 10}

```
In [76]: 1 print(s.discard(20))
         2 print(s.remove(20))
```

None

```
-----
KeyError                                Traceback (most recent call last)
<ipython-input-76-3047e5db0486> in <module>
      1 print(s.discard(20))
----> 2 print(s.remove(20))
```

**KeyError:** 20

```
In [78]: 1 s.clear()
```

```
In [80]: 1 type(s)
```

Out[80]: set

```
In [81]: 1 del s
```

```
In [82]: 1 s
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-82-ded5ba42480f> in <module>
----> 1 s
```

**NameError:** name 's' is not defined

```
In [89]: 1 s1 = {1,2,3,4,5,10,20}
        2 s2 = {10,20,"a","b"} # {a,b}
        3 s2.difference(s1)
        4 s2.difference_update(s1)
        5 s2
```

Out[89]: {'a', 'b'}

```
In [85]: 1 s1.union(s2)
```

Out[85]: {1, 10, 2, 20, 3, 4, 5, 'a', 'b'}

```
In [92]: 1 s1.update(s2)
        2 s1
```

Out[92]: {1, 10, 2, 20, 3, 4, 5, 'a', 'b'}

```
In [94]: 1 s1 = {1,2,3,4,5,10,20} #{1,2,3,4,5}
        2 s2 = {10,20,"a","b"} # {a,b}
        3 s2.difference(s1)
        4 s1
```

Out[94]: {1, 2, 3, 4, 5, 10, 20}

```
In [96]: 1 s2.symmetric_difference_update(s1)
```

```
In [97]: 1 s2
```

Out[97]: {1, 2, 3, 4, 5, 'a', 'b'}

```
In [100]: 1 s = "Python programming"
        2 s1=""
        3 for i in s:
        4     if i not in s1:
        5         s1 = s1+i
        6 c=0
        7 for j in s1:
        8     if j.isalpha():
        9         c=c+1
       10 print(c)
       11
       12
```

12

```
In [ ]: 1
```

