

SET & DICT

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

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

```
Out[2]: list
```

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

```
Out[3]: set
```

```
In [4]: s1.add(10)
```

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

```
Out[5]: {10}
```

```
In [6]: s1.add(20)  
s1.add('nit')  
s1.add(True)  
s1.add(1+2j)  
s1.add(3.2)
```

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

```
Out[7]: {(1+2j), 10, 20, 3.2, True, 'nit'}
```

```
In [8]: s2=set()
```

```
In [9]: s2
```

```
Out[9]: set()
```

```
In [10]: s2.add(100)  
s2.add(10)  
s2.add(200)  
s2.add(9)
```

```
In [11]: s2
```

```
Out[11]: {9, 10, 100, 200}
```

```
In [12]: s3=set()  
s3
```

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

```
In [13]: s3.add('z')  
s3.add('a')
```

```
s3.add('m')
s3.add('b')
```

In [14]: s3

Out[14]: {'a', 'b', 'm', 'z'}

```
In [15]: print(s1)
         print(s2)
         print(s3)
```

```
{True, 3.2, (1+2j), 10, 20, 'nit'}
{200, 9, 10, 100}
{'a', 'b', 'm', 'z'}
```

In []: s3[:]

In [17]: print(s1)

```
{True, 3.2, (1+2j), 10, 20, 'nit'}
```

print(s2)

In [18]: print(s2)

```
{200, 9, 10, 100}
```

print(s3)

In [19]: print(s3)

```
{'a', 'b', 'm', 'z'}
```

In [22]: s4=s3.copy()

In [23]: s4

Out[23]: {'a', 'b', 'm', 'z'}

In [24]: s3==s4

Out[24]: True

In [25]: s4

Out[25]: {'a', 'b', 'm', 'z'}

In [26]: s4.clear()

s4

In [27]: del s4

In [28]: s4

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[28], line 1  
----> 1 s4  
  
NameError: name 's4' is not defined
```

```
In [29]: s4=s2.copy()  
s4
```

```
Out[29]: {9, 10, 100, 200}
```

```
In [30]: s4.pop()
```

```
Out[30]: 200
```

```
In [31]: s4.pop()
```

```
Out[31]: 9
```

```
In [32]: s4
```

```
Out[32]: {10, 100}
```

```
In [33]: s1
```

```
Out[33]: {(1+2j), 10, 20, 3.2, True, 'nit'}
```

```
In [35]: s1.pop(0)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[35], line 1  
----> 1 s1.pop(0)  
  
TypeError: set.pop() takes no arguments (1 given)
```

```
In [36]: s1
```

```
Out[36]: {(1+2j), 10, 20, 3.2, True, 'nit'}
```

```
In [37]: s1.remove((1+2j))
```

```
In [38]: s1
```

```
Out[38]: {10, 20, 3.2, True, 'nit'}
```

```
In [39]: for i in s4:  
         print(i)
```

```
10  
100
```

```
In [40]: for i in enumerate (s4):  
         print(i)
```

```
(0, 10)  
(1, 100)
```

SET OPERATOR

```
In [43]: A={1,2,3,4,5}
        B={4,5,6.7,8}
        C={8,9,10}
```

```
In [45]: A|B # This is The Case Of Union
```

```
Out[45]: {1, 2, 3, 4, 5, 6.7, 8}
```

```
In [46]: B.union(C)
```

```
Out[46]: {4, 5, 6.7, 8, 9, 10}
```

```
In [47]: A.union(B,C)
```

```
Out[47]: {1, 2, 3, 4, 5, 6.7, 8, 9, 10}
```

```
In [48]: C|B|A
```

```
Out[48]: {1, 2, 3, 4, 5, 6.7, 8, 9, 10}
```

```
In [49]: D=C.copy()
```

```
In [50]: D
```

```
Out[50]: {8, 9, 10}
```

```
In [52]: print(C)
        print(D)
```

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

```
In [53]: C.update(B)
```

```
In [54]: C
```

```
Out[54]: {4, 5, 6.7, 8, 9, 10}
```

```
In [55]: print(A)
        print(B)
        print(C)
```

```
{1, 2, 3, 4, 5}
{8, 4, 5, 6.7}
{4, 5, 6.7, 8, 9, 10}
```

```
In [56]: A&B
```

```
Out[56]: {4, 5}
```

```
In [57]: B.intersection(C)
```

Out[57]: {4, 5, 6.7, 8}

In [58]: `A&B&C`

Out[58]: {4, 5}

In [59]: `A-B`

Out[59]: {1, 2, 3}

In [60]: `B-C`

Out[60]: `set()`

In [61]: `C-B`

Out[61]: {9, 10}

In [62]: `C.difference(A)`

Out[62]: {6.7, 8, 9, 10}

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []: