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

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

```
In [2]: len(myset)
```

Out[2]: 5

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

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

```
In [4]: myset1={1.79,2.08,3.99,4.56,5.45}
        myset1
```

Out[4]: {1.79, 2.08, 3.99, 4.56, 5.45}

```
In [5]: myset2={'Priya','Anu','Mouni'}
        myset2
```

Out[5]: {'Anu', 'Mouni', 'Priya'}

```
In [6]: myset3={10,20,"Priya",(11,22,32)}
        myset3
```

Out[6]: {(11, 22, 32), 10, 20, 'Priya'}

```
In [9]: myset3={10,20,"Priya",[11,22,32]}
        myset3
```

```
        ---------------------------------------------------------------------------
        TypeError                                 Traceback (most recent call last)
        Cell In[9], line 1
        ----> 1 myset3={10,20,"Priya",[11,22,32]}
              2 myset3

        TypeError: unhashable type: 'list'
```

```
In [10]: myset4=set()
         print(type(myset4))
```

         <class 'set'>

```
In [11]: my_set1=set(('one','two','three','four'))
         my_set1
```

Out[11]: {'four', 'one', 'three', 'two'}

```
In [12]: myset={'one','two','three','four','five','six','seven','eight'}
         for i in myset:
             print(i)
```

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

In [13]:
```python
for i in enumerate(myset):
    print(i)
```

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

In [14]:
```python
myset
```

Out[14]: `{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}`

In [15]:
```python
'one' in myset
```

Out[15]: `True`

In [16]:
```python
'ten'in myset
```

Out[16]: `False`

In [17]:
```python
if 'three' in myset:
    print('Three is present in the set')
else:
    print('Three is not present in the set')
```

```
Three is present in the set
```

In [18]:
```python
if 'eleven' in myset:
    print('Eleven is present in the set')
else:
    print('Eleven is not present in the set')
```

```
Eleven is not present in the set
```

In [19]:
```python
myset
```

Out[19]: `{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}`

In [20]:
```python
myset.add('NINE')
myset
```

Out[20]: `{'NINE', 'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}`

In [21]:
```python
myset.update(['TEN','ELEVEN','TWELVE'])
myset
```

Out[21]: {'ELEVEN',
           'NINE',
           'TEN',
           'TWELVE',
           'eight',
           'five',
           'four',
           'one',
           'seven',
           'six',
           'three',
           'two'}

In [22]: ```python
          myset.remove('NINE')
          myset
          ```

Out[22]:  {'ELEVEN',
           'TEN',
           'TWELVE',
           'eight',
           'five',
           'four',
           'one',
           'seven',
           'six',
           'three',
           'two'}

In [23]:  ```python
          myset.discard('TEN')
          myset
          ```

Out[23]: {'ELEVEN',
           'TWELVE',
           'eight',
           'five',
           'four',
           'one',
           'seven',
           'six',
           'three',
           'two'}

In [24]: ```python
          myset.clear()
          myset
          ```

Out[24]:  set()

In [25]:  ```python
          del myset
          myset
          ```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[25], line 2
      1 del myset
----> 2 myset

NameError: name 'myset' is not defined
```

In [26]: ```python
          myset={'one','two','three','four','five','six','seven','eight'}
          ```

```
myset
```

Out[26]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

In [27]:
```
myset1=myset
myset1
```

Out[27]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

In [28]:
```
id(myset),id(myset1)
```

Out[28]: (2973996508192, 2973996508192)

In [29]:
```
my_set=myset.copy()
my_set
```

Out[29]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

In [30]:
```
id(my_set)
```

Out[30]: 2974003982432

In [31]:
```
myset.add('nine')
myset
```

Out[31]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}

In [32]:
```
myset1
```

Out[32]: {'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three', 'two'}

In [33]:
```
my_set
```

Out[33]: {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

In [35]:
```
A={1,2,3,4,5}
B={4,5,6,7,8}
c={8,9,10}
```

In [36]:
```
A|B
```

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

In [37]:
```
A.union(B)
```

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

In [39]:
```
A.union(B,c)
```

Out[39]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [40]:
```
A={1,2,3,4,5}
B={4,5,6,7,8}
```

In [41]:
```
A&B
```

Out[41]: {4, 5}

In [42]: `A.intersection(B) Intersection of A and B`

> **Cell In[42], line 1**
>   `A.intersection(B) Intersection of A and B`
>   ^
> **SyntaxError:** invalid syntax

In [53]:
```
A={1,2,3,4,5}
B={4,5,6,7,8}
```

In [54]: `A-B`

Out[54]: {1, 2, 3}

In [55]: `A.difference(B)`

Out[55]: {1, 2, 3}

In [56]: `B-A`

Out[56]: {6, 7, 8}

In [57]: `B.difference(A)`

Out[57]: {6, 7, 8}

In [58]:
```
A={1,2,3,4,5}
B={4,5,6,7,8}
```

In [59]: `A^B`

Out[59]: {1, 2, 3, 6, 7, 8}

In [60]: `A.symmetric_difference(B)`

Out[60]: {1, 2, 3, 6, 7, 8}

In [51]:
```
A.symmetric_difference_update(B)
A
```

Out[51]: {1, 2, 3, 6, 7, 8}

In [61]:
```
A={1,2,3,4,5,6,7,8,9}
B={3,4,5,6,7,8}
C={10,20,30,40}
```

In [62]: `B.issubset(A)`

Out[62]: True

In [63]: `A.issuperset(B)`

Out[63]: True

```
In [64]: C.isdisjoint(A)
```

Out[64]: True

```
In [65]: B.isdisjoint(A)
```

Out[65]: False

```
In [66]: A
```

Out[66]: {1, 2, 3, 4, 5, 6, 7, 8, 9}

```
In [67]: sum(A)
```

Out[67]: 45

```
In [68]: max(A)
```

Out[68]: 9

```
In [69]: min(A)
```

Out[69]: 1

```
In [70]: len(A)
```

Out[70]: 9

```
In [71]: list(enumerate(A))
```

Out[71]: [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]

```
In [72]: D=sorted(A,reverse=True)
         D
```

Out[72]: [9, 8, 7, 6, 5, 4, 3, 2, 1]

```
In [74]: sorted(D)
```

Out[74]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

```
In [ ]:
```