

In [ ]:

```
#A set is an unordered collection of items.  
#Every set element is unique (no duplicates) and must be immutable (cannot be changed).  
#However, a set itself is mutable. We can add or remove items from it.  
#Sets can also be used to perform mathematical set operations like union, intersection,  
symmetric difference, etc.
```

In [ ]:

```
#declaring a set  
S={1,2,3}  
print(S)
```

{1, 2, 3}

In [ ]:

```
S = {1.0, "Hello", (1, 2, 3)}  
print(S)
```

{1.0, 'Hello', (1, 2, 3)}

In [ ]:

```
#set will not have duplicates  
S = {1, 2, 3, 4, 3, 2}  
print(S)
```

{1, 2, 3, 4}

In [ ]:

```
#set can be created from a list  
l=[1,2,3,4,4]  
S=set(l)  
print(S)
```

{1, 2, 3, 4}

In [ ]:

```
#set cannot have mutable item like list  
S={1,2,[3,4]}
```

```
-----  
-  
TypeError                                Traceback (most recent call las  
t)  
<ipython-input-6-9977531ccb41> in <module>()  
      1 #set cannot have mutable item  
----> 2 S={1,2,[3,4]}
```

TypeError: unhashable type: 'list'

In [ ]:

```
#Creating an empty set  
S=set()  
print(S)
```

set()

In [ ]:

```
#We can add a single element using the add() method, and multiple elements using the update() method.  
#The update() method can take tuples, lists, strings or other sets as its argument. In all cases, duplicates are avoided
```

In [ ]:

```
S={1,3}  
print(S)  
S.add(4)  
print(S)  
S.update([2,3,5])  
print(S)  
S.update([7,8],{9,10})  
print(S)
```

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

In [ ]:

```
#Removing elements from a set
#A particular item can be removed from a set using the methods discard() and remove().
#The only difference between the two is that the discard() function leaves a set unchanged
#if the element is not present in the set.
#On the other hand, the remove() function will raise an error in such a condition (if element is not present in the set).
S={1,2,3,4}
S.discard(4)
print(S)
S.remove(3)
print(S)
S.discard(3)
print(S)
S.remove(5)
```

```
{1, 2, 3}
{1, 2}
{1, 2}
```

```
-----
-
KeyError                                Traceback (most recent call last)
t)
<ipython-input-11-0110ff0be21e> in <module>()
     11 S.discard(3)
     12 print(S)
--> 13 S.remove(5)
     14
```

KeyError: 5

In [ ]:

```
#we can remove and return an item using the pop() method.
#Since set is an unordered data type, there is no way of determining which item will be popped. It is completely arbitrary
S={1,5,2,3,4}
print(S.pop())
print(S.pop())
```

```
1
2
```

In [ ]:

```
#union operation
A = {1, 2, 3, 4}
B = {4, 5, 6, 7, 8}
print(A|B)
print(A.union(B))

print(B.union(A))
```

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

In [ ]:

```
#intersection  
print(A&B)  
print(A.intersection(B))  
print(B.intersection(A))
```

```
{4}  
{4}  
{4}
```

In [ ]:

```
#Difference of the set B from set A(A - B) is a set of elements that are only in A but  
not in B.  
#Similarly, B - A is a set of elements in B but not in A  
print(A-B)  
print(B-A)  
print(A.difference(B))  
print(B.difference(A))
```

```
{1, 2, 3}  
{8, 5, 6, 7}  
{1, 2, 3}  
{8, 5, 6, 7}
```

In [ ]:

```
#Symmetric Difference of A and B is a set of elements in A and B but not in both (exclu  
ding the intersection).  
#Symmetric difference is performed using ^ operator.  
print(A^B)  
print(B^A)  
print(A.symmetric_difference(B))
```

```
{1, 2, 3, 5, 6, 7, 8}  
{1, 2, 3, 5, 6, 7, 8}  
{1, 2, 3, 5, 6, 7, 8}
```