Set

In [1]:

```
#set is an unordered collection of unique elements
#enclosed with ()
#set objects are mutable
#duplicates are not allowed
#inseertion order is not preserved so indexing and slicing is not allowed for sets
```

Creation

In [2]:

```
#creating set with curly brackets{}
#elements are seperated by comma ,
set1={10,30,34,72,82}
set1
```

Out[2]:

```
{10, 30, 34, 72, 82}
```

In [3]:

```
#using set function
set2=set({20,49,38,47,93})
set2
```

Out[3]:

```
{20, 38, 47, 49, 93}
```

In [5]:

```
set3=set("karuna")
set3
```

Out[5]:

```
{'a', 'k', 'n', 'r', 'u'}
```

Accessing set elements

In [9]:

```
#access the element using looping statement
s1={9,4,(2,3),8}
for i in s1:
    print(i)
```

```
(2, 3)
9
4
8
```

deleting set

In [10]:

```
s1
```

Out[10]:

```
{(2, 3), 4, 8, 9}
```

In [11]:

```
# we can delete the set using del keyword
del s1
s1
```

```
-----------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
/var/folders/v8/088c1jn178q1q1c1_1b7jg000000gn/T/ipykernel_1741/479101
526.py in <module>
      1 # we can delete the set using del keyword
      2 del s1
----> 3 s1

NameError: name 's1' is not defined
```

methods on set

1.union

In [13]:

```
#it returns the combination of 2 sets
set1={1,2,3,4,5}
set2={5,6,7,8,9}
set1.union(set2)
```

Out[13]:

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

2.Intersection

In [15]:

```
#it retuens the common elements of 2 sets
set1.intersection(set2)
```

Out[15]:

```
{5}
```

difference

In [17]:

```
#It returns a set with elements which are not in all other sets
set1.difference(set2)
```

Out[17]:

```
{1, 2, 3, 4}
```

Mutation operations on set

1.add

In [19]:

```
#It adds an element to the set
s1={10,20,30,40}
s1.add(50)
s1
```

Out[19]:

```
{10, 20, 30, 40, 50}
```

2.remove

In [21]:

```
#It removes the element of set
#if element not in set it shows error
s1.remove(50)
s1
```

Out[21]:

```
{10, 20, 30, 40}
```

In [23]:

```
s1.remove(60)
s1
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/var/folders/v8/088c1jn178q1q1c1_1b7jg000000gn/T/ipykernel_1741/3224693740.py in <module>
----> 1 s1.remove(60)
      2 s1

KeyError: 60
```

3.discard

In [29]:

```
#It removes the element from set
s1=set(range(5,80,10))
s1.discard(25)
s1
```

Out[29]:

```
{5, 15, 35, 45, 55, 65, 75}
```

4.pop

In [37]:

```
#It removes and returns the removed element from set
#no indexing so it removes random last elements
s1=set(range(10,100,10))
s1.pop()
s1
```

Out[37]:

```
{10, 20, 30, 40, 50, 60, 80, 90}
```

clear

In [39]:

```
#It removes all values from set and make it empty
s1
```

Out[39]:

```
{10, 20, 30, 40, 50, 60, 80, 90}
```

In [40]:

```
s1.clear()
s1
```

Out[40]:

```
set()
```

Updation operations on sets

1.update

In [43]:

```
#It adds elements from set2,.....,setn
s1={10,20,30,40}
s2={80,60,70,50}
s1.update(s2)
s1
```

Out[43]:

```
{10, 20, 30, 40, 50, 60, 70, 80}
```

In [45]:

```
#also it represnt in the symbol |=
s1|=s2
s1
```

Out[45]:

```
{10, 20, 30, 40, 50, 60, 70, 80}
```

intersection_update

In [47]:

```
#it returns the common element from all sets
s1={10,20,30,40,50}
s2={40,50,60,70,80}
s1.intersection_update(s2)
s1
```

Out[47]:

```
{40, 50}
```

In [48]:

```
# also it represnt in the symbol &=
s1&=s2
s1
```

Out[48]:

```
{40, 50}
```

Function on Set

1.len

In [51]:

```
#this returns number of element in set
s1={70,38,57,2,83,85,793,0,39,3,6}
len(s1)
```

Out[51]:

```
11
```

2.sum

In [53]:

```
#returns the sum of element in the set
sum(s1)
```

Out[53]:

```
1176
```

3.sorted

In [56]:

```
#sorts the element of set in ascending or descending order
#Ascending order
s1={49,85,29,0,35,84,98,35,331,86,98}
sorted(s1)
```

Out[56]:

```
[0, 29, 35, 49, 84, 85, 86, 98, 331]
```

In [57]:

```
#Descending order
sorted(s1,reverse=True)
```

Out[57]:

```
[331, 98, 86, 85, 84, 49, 35, 29, 0]
```

4.max

In [59]:

```
#It returns the largest element from set
max(s1)
```

Out[59]:

```
331
```

5.min

In [61]:

```
#It returns the smallest element from set
min(s1)
```

Out[61]:

```
0
```

In [ ]:

```
 
```