

In [1]:

```
1 #Set data structure:-  
2 #1.Duplicates are not allowed  
3 #2.Order is not applicable  
4 #3.Indexing and slicing concepts are not applicable  
5 #4.{1,23,4,5}  
6 #5.Heterogenous objects  
7 #6.Mutable  
8 #7. Union, intersection difference
```

In [2]:

```
1 s={}  
2 print(type(s))
```

<class 'dict'>

In [3]:

```
1 s=set()  
2 print(type(s))
```

<class 'set'>

In [5]:

```
1 s.add(1)  
2 s.add(4)  
3 s.add("akhil")  
4 print(s)
```

{1, 4, 'akhil'}

## Creation of set object:-

In [6]:

```
1 #1.empty set:-  
2 s=set()  
3 print(type(s))
```

<class 'set'>

In [7]:

```
1 #2.If we have data already:-  
2 s={1,35,5,7,8}  
3 print(type(s))
```

<class 'set'>

In [8]:

```
1 #3.by using set():-
2 s=set()
3 print(type(s))
```

&lt;class 'set'&gt;

In [9]:

```
1 #range():-
2 s=set(range(1,11))
3 print(s)
```

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [11]:

```
1 s=set("akhiil")
2 print(s)
```

{'h', 'l', 'a', 'i', 'k'}

In [12]:

```
1 s=eval(input("enter set:"))
2 print(s)
```

enter set:{3,4,6,8}

{8, 3, 4, 6}

In [13]:

```
1 #mathematical operators for set:-
2 s1={1,2,3,4}
3 s2={4,3,2,1}
4 print(s1==s2)
5 print(s1!=s2)
```

True

False

In [20]:

```
1 #relational operators for set:-
2 s1={1,3,2,4}
3 s2={2,32,43,4,6}
4 print(s1<s2)
5 print(s1>s2)
6
```

False

False

In [21]:

```
1 #Membership :-  
2 # in, not in  
3 s={1,2,3,5,7}  
4 print(10 in s)  
5 print(7 in s)
```

False

True

## Important function and methods for set:-

In [1]:

```
1 #1.Len()=====>returns no. of elements present in set  
2 #2.add()  
3 s={10,20,30}  
4 s.add(40)  
5 print(s)
```

{40, 10, 20, 30}

In [2]:

```
1 #3.update()=====>to add mutiple items to the set  
2 s={10,20}  
3 l=[11,22]  
4 s.update(l)  
5 print(s)
```

{10, 11, 20, 22}

In [3]:

```
1 #Remove elements from set:-  
2 #1.remove()  
3 #s.remove(x)=====>x will be deleted  
4 s={1,2,3,4,5}  
5 s.remove(5)# If the element is not in the set gets KeyError  
6 print(s)
```

{1, 2, 3, 4}

In [4]:

```
1 #2.s.discard()  
2 s={12,4,5,7,9}  
3 s.discard(12)  
4 print(s)  
5 s.discard(99)# we didn't get error for discard  
6 print(s)
```

{4, 5, 7, 9}

{4, 5, 7, 9}

In [6]:

```
1 #3. pop()
2 s={1,2,3,4}
3 s.pop()
4 print(s)# if the elements not present in the set it raise Error
```

{2, 3, 4}

In [7]:

```
1 #4. s.clear()
2 s={1,2,3,4,5,6}
3 s.clear()
4 print(s)
```

set()

In [9]:

```
1 #set specific methods for mathematical operations:-
2 #1. union()=====>s1|s2
3 s1={1,2,4,5}
4 s2={7,6,9,10}
5 s3=s1.union(s2)
6 print(s3)
```

{1, 2, 4, 5, 6, 7, 9, 10}

In [10]:

```
1 #2. intersection()
2 s1={4,2,1,7}
3 s2={38,8,3,2,4}
4 s3=s1.intersection(s2)
5 print(s3)
```

{2, 4}

In [13]:

```
1 #3. difference()
2 s1={1,2,4,5,6}
3 s2={4,5,7,8,9}
4 s3=s2.difference(s1)
5 print(s3)
```

{8, 9, 7}

In [14]:

```
1 #4. symmetric_difference()
2 s1={1,2,3,4,5,6,7}
3 s2={6,7,5,8,10}
4 s3=s1.symmetric_difference(s2)
5 print(s3)
```

{1, 2, 3, 4, 8, 10}

In [15]:

```
1 #Aliasing & Cloning:-
2 s1={1,2,3,4,5}
3 s2=s1
4 print(id(s1))
5 print(id(s2))
```

2195046225728

2195046225728

In [17]:

```
1 s3=s1.copy()
2 print(id(s3))
3 print(id(s1))
```

2195044851040

2195046225728

In [18]:

```
1 #set comprehension:-
2 s={x for x in range(1,11)}
3 print(s)
```

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [21]:

```
1 #write a program to eliminate duplicates in the list:-
2 l=[10,30,29,10,39,20,30]
3 s=set(l)
4 print(s)
5 l=list(s)
6 print(l)
```

{39, 10, 20, 29, 30}

[39, 10, 20, 29, 30]

In [22]:

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

[1, 2, 3, 5, 6, 8, 9, 4]

In [23]:

```
1 #write a program to print different vowels present in the given word
2 s=input("enter any string to search for vowels:")
3 s1=set(s)
4 s2={"o","u","e","a","i"}
5 result=s1.intersection(s2)
6 print(result)
```

enter any string to search for vowels:akhilerenagjalj  
{'i', 'a', 'e'}

In [ ]:

1