Today I'll Cover :

1. Set Data Structure
2. Creation of Set
3. Methods of Set
4. Mathematical operation on set.
5. Nested Set & Set Comprehension

Mohammad Altaf Hussain

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

In the name of ALLAH,
the Most Beneficent, the Most Merciful

Mohammad Altaf Hussain

Set {}

If we want to represent a group of unique objects as a single entity where :-

- insertion order is not preserved.
- duplicate objects are not allowed
- Indexing and slicing not allowed
- heterogeneous objects are allowed
- Modification are allowed, once object is created. Then we should go for Set. The elements are placed within curly braces and with comma separator.

Mohammad Altaf Hussain

Creation of Set

- ❑ Empty Set:-
  1. Set()

- ❑ Set with element:-

  { element1, element2, elementN }

- ▪ Set() function used to cast other data type into Set type.

Mohammad Altaf Hussain

Traversing in Set

We can traverse in set using:-

- ❑ For loop

Methods of Set

add() :- It add a single element in the Set.

Syntax: any_Set.add(element)

Update() :- It add multiple item from a iterable
object (list, range, tuple) to the set.

Syntax: any_Set.update(iterable1, iterableN)

Mohammad Altaf Hussain

❖ Delete element from given Set

➢ **pop() :-** It removes and returns random element from the set.

Syntax: any_Set.pop()

➢ **remove() :-** It removes specified element from the set.

- If the specified element not present in the Set then we will get KeyError.

Syntax: any_Set.remove(element)

➢ **discard() :-** It removes the specified element from the set.

- If the specified element not present in the set then we won't get any error.

Syntax: any_Set.discard(element)

❖ Cloning of Set

➢ **copy()** :- It returns the copy the set.

Syntax: any_Set.copy()

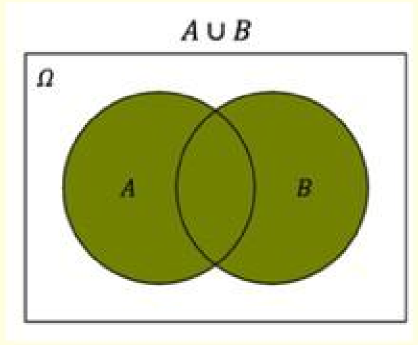
❖ Remove all the element of Set

➢ **clear()** :- It removes all the element from the set.

Syntax: any_Set.clear()

❖ Mathematical operation on the set :-

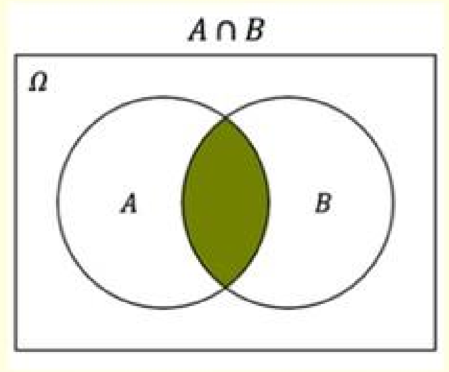
❑ **Union:** It return all elements present in both sets.



Syntax: any_Set.union(another_set)

any_set | another_set

❑ **Intersection** : It Returns common elements present in both x and y.

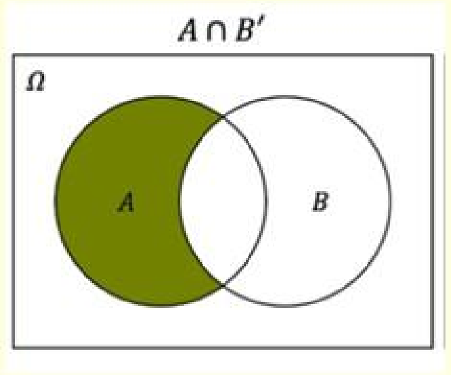


Syntax: any_Set.intersection(another_set)

any_set & another_set

Mohammad Altaf Hussain

- **Difference:** It returns the elements present in First set but not in Second Set.



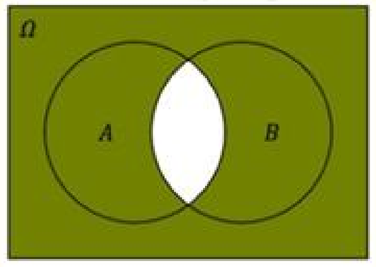
Syntax: **any_Set.difference(another_set)**

any_set - another_set

- **Symmetric_difference() :** It s elements present in either x or y but not in both.

Syntax: **any_Set.symmetric_difference(another_set)**

any_set ^ another_set



Mohammad Altaf Hussain

❑ Membership Operator:-

1. in

2. not in

Set Comprehension

- It is very easy and compact way of creating set objects from any iterable objects(like list,tuple,dictionary,range etc) based on some condition.

Syntax:-

```
sets={expression for item in iterable if condition}
```

Mohammad Altaf Hussain