SETS

* Sets are unique collection of elements.
  + Heterogeneous
  + Non-iterable and No indexing
  + But Mutable
* Syntax:
  + nam\_set=set(*name*\_*list or list*)

example:-myset( [1,2,3,4] )

* It gives full set or nothing is given.
* Sets are orbitary
* Methods in sets are:
  + add( ):adds an element to the set
    - >>> a={1,2,3,4}
    - >>> b={4,5,6,7}
    - >>> a
    - {1, 2, 3, 4}
    - >>> a.add(9)
    - >>> a
    - {1, 2, 3, 4, 9}
  + remove( ):removes an element from the set
    - >>> a
    - {1, 2, 3, 4, 9}
    - >>> a.remove(9)
    - >>> a
    - {1, 2, 3, 4}
  + discard( ):
    - >>> a
    - {1, 2, 3, 4, 9}
    - >>> a.discard(9)
    - >>> a
    - {1, 2, 3, 4}
  + union( ):joins both the sets and gives the set
    - >>> a
    - {1, 2, 3, 4}
    - >>> b
    - {4, 5, 6, 7}
    - >>> a.union(b)
    - {1, 2, 3, 4, 5, 6, 7}
  + intersection( ):returns common elements from both the sets
    - >>> a
    - {1, 2, 3, 4}
    - >>> b
    - {4, 5, 6, 7}
    - >>> a.intersection(b)
    - {4}
  + difference( ):compares set‘a’ with set’b’ and returns elements that are distint in a.
    - >>> a
    - {1, 2, 3, 4}
    - >>> b
    - {4, 5, 6, 7}
    - >>> a.difference(b)
    - {1, 2, 3}
  + issubset( ):if set’a’ is found in set’c’ then it return ‘true’ otherwise ‘false’.
    - >>> c
    - {1, 2, 3, 4, 5, 6, 7}
    - >>> a
    - {1, 2, 3, 4}
    - >>> a.issubset(c)
    - True
  + issuperset( ):if set’c’ is having the set’a’ then from ‘a’ point of view ‘c’ is superset to ‘a’ in such case it return ‘true’ otherwise ‘false’
    - >>> c.issuperset(a)
    - True
    - >>> c
    - {1, 2, 3, 4, 5, 6, 7}
    - >>> a
    - {1, 2, 3, 4}
    - >>> c.issuperset(a)
    - True
* Frozenset
  + The frozenset() method returns an immutable frozenset object initialized with elements from the given iterable.
  + Syntax:
    - frozenset([iterable])