* Set.

Set is an unordered collection of unique item.

unordered: There's no fixed indexing. (like list/tuple)
unique: Duplicate values (objects) are ignored

· Creation of set.

1. using curly braces.

my_set = { 10, 20, "shaill"}

2. using set() function.

my_set = set([1, 2, 2, 3]) // {1, 2, 3}

3. Empty set

my_set = set()

IF you write just my_set as {} it will become a dictionary.

· Properties

1. unordered → indexing/slicing not allowed.
2. unique → Removes duplicates automatically.
3. Mutable → can add/remove elements.
4. Heterogeneous elements allowed (int, str, tuple but not mutable type like list/dict inside a set).

- Basic Operations.

1. Add Element

   ```
   s = {1, 2}

   s.add(3)
   print(s)  // {1, 2, 3}
   ```

2. Remove element.

   ```
   s.remove(2)    # throws exception if not present.
   s.discard(4)   # no error if not present.
   ```

3. Pop element (removes random element).

4. Clear Set

   ```
   s.clear()  →  returns you empty set (set()).
   ```

- Set Operations (Mathematical).

   Python sets behaves like Math sets.

   ```
   A = {1, 2, 3, 4}      print(A | B)   # union → {1, 2, 3, 4, 5, 6}
                         print(A & B)   # intersection
   B = {3, 4, 5, 6}      print(A - B)   # Difference {1, 2}
                         print(A ^ B)   # symmetric Difference
                                        {1, 2, 5, 6}
   ```

Some useful Methods.

S.update ([list]) → adds multiple
S. issubset (other) → subset check.
S. issuperset (other) → superset check.
S. isdisjoint (other) → True if no common elements.
len(S) → size of set.