Sets

MUKESH KUMAR

What are Sets?

- An unordered collection of unique elements.
- Enclosed within curly braces {}.
- Does not allow duplicate elements.

Creating sets

```
# Using curly braces
my_set = {1, 2, 3, 4, 5}

# Using set() function
another_set = set([3, 4, 5, 6])
```

- **Note: {} creates a dictionary, use set() for an empty set.**

Key Characteristics

- **Unordered:** The order of elements is not guaranteed.
- **Unique:** Only one instance of each element is allowed.
- **Mutable:** Elements can be added or removed after creation.

Set Methods

Method	Description
add(element)	Adds an element to the set.
remove(element)	Removes an element from the set. Raises KeyError if the element is not found.
discard(element)	Removes an element from the set if it exists. Does **not** raise an error if the element is not found.
pop()	Removes and returns an **arbitrary** element from the set.
clear()	Removes **all** elements from the set.
issubset(set2)	Checks if a set is a subset.
issuperset(set2)	Checks if a set is a superset.

Set Operations

- **union():** The union() method combines all elements from two sets, eliminating duplicates
- **intersection():** The intersection() method returns a set containing all elements that are common to both sets.
- **difference():** The difference() method returns a set containing all elements from the first set that are not in the second set.
- **symmetric_difference():** The symmetric_difference() method returns a set containing all elements from both sets except the common elements

Set Operations (Using Operators)

- | for union: set1 | set2
- & for intersection: set1 & set2
- - for difference: set1 - set2
- ^ for symmetric difference: set1 ^ set2

When to Use enumerate() with a Set

- If you don't care about order, using enumerate() is fine.
- If you want a fixed order, consider converting the set into a sorted list or tuple first.
- Since sets in Python do not maintain insertion order (except for Python 3.7+), still should not be relied upon for ordered indexing
- The indices assigned by enumerate() may vary between runs.

When to Use Sets

- Removing duplicates from a list.
- Checking for membership efficiently.
- Performing set operations like union, intersection, and difference.
- Representing collections of unique items (e.g., unique words in a text).

Summary

- Sets are unordered collections of unique elements.
- Support operations like union, intersection, and difference.
- Useful for removing duplicates and performing mathematical operations.
- Provide fast membership testing using in.