Python Sets Methods - Detailed & Formatted

3. Purpose of add() method in Python sets

- Inserts a single new element into an existing set.
- If the element already exists, the set remains unchanged.
- Syntax: my_set.add(element)

4. Purpose of update() method in Python sets

- Adds multiple elements from one or more iterables to the set.
- Syntax: my_set.update(iterable1, iterable2, ...)

5. Purpose of remove() method in Python sets

- Deletes a specific element from the set.
- Raises KeyError if the element is not found.
- Syntax: my_set.remove(element)

6. Purpose of discard() method in Python sets

- Deletes a specific element from the set.
- Does NOT raise an error if the element does not exist.
- Syntax: my_set.discard(element)

7. Differences between add() vs update()

- add(): Adds one element.
- update(): Adds multiple elements.

8. Differences between remove() vs discard()

- remove(): Raises KeyError if the item doesn't exist.
- discard(): No error if the item doesn't exist.

9. How to remove all items from a set

- Use the clear() method: my_set.clear()

10. How to use copy() to create a shallow copy

- Syntax: set_copy = original_set.copy()

11. Concept of frozen set

- A frozenset is an immutable (unchangeable) version of a set.

- Syntax: frozen = frozenset([1, 2, 3])

12. How to perform union operation on multiple sets

- Use union() method.

- Syntax: set1.union(set2, set3)

13. How to perform intersection operation on multiple sets

- Use intersection() method.

- Syntax: set1.intersection(set2, set3)

14. How to check if one set is a subset of another

- Use issubset() method.

- Syntax: set1.issubset(set2)

15. How to check if one set is a superset of another

- Use issuperset() method.

- Syntax: set1.issuperset(set2)

16. Purpose of pop() method in sets

- Removes and returns an arbitrary element from the set.

- Syntax: element = my_set.pop()

17. Difference between union() and update()

- union(): Returns a new set.

- update(): Modifies the existing set.

18. Role of symmetric_difference() method

- Returns a set with elements in either set but not in both.

- Syntax: result = set1.symmetric_difference(set2)

19. Role of symmetric_difference_update() method

- Updates the set to keep only elements in either set but not in both.

- Syntax: set1.symmetric_difference_update(set2)

20. Role of intersection_update() method

- Updates the set with elements common to itself and another set.

- Syntax: set1.intersection_update(set2)