Collection Methods

1. List Methods

Lists are **ordered** and **mutable**, so they support many methods:

Method	Description	Example
`append()`	Add item to the end	`fruits.append("mango")`
`insert(i, x)`	Insert item at index `i`	`fruits.insert(1, "grape")`
`remove(x)`	Remove first matching item	`fruits.remove("apple")`
`pop([i])`	Remove item at index (last if empty)	`fruits.pop()`
`sort()`	Sort the list	`fruits.sort()`
`reverse()`	Reverse the list	`fruits.reverse()`
`clear()`	Remove all items	`fruits.clear()`

Example:

```
fruits = ["banana", "apple", "cherry"]
fruits.append("mango")
fruits.sort()
print(fruits)  # ['apple', 'banana', 'cherry', 'mango']
```

---

2. Tuple Methods

Tuples are **immutable**, so they have **only 2 methods**:

Method	Description
`count(x)`	Count occurrences of item `x`
`index(x)`	Return first index of item `x`

Example:

```
t = (1, 2, 3, 2, 4)
print(t.count(2))   # 2
print(t.index(3))   # 2
```

---

3. Set Methods

Sets are **unordered** and contain **unique** values.

Method	Description
`add(x)`	Add an element
`update()`	Add multiple elements
`remove(x)`	Remove element (error if not found)
`discard(x)`	Remove element (no error if not found)
`pop()`	Remove a random element
`clear()`	Remove all elements
`union()`	Return union of two sets
`intersection()`	Return common elements
`difference()`	Elements not in another set

Example:

```
s = {1, 2, 3}
s.add(4)
s.update([5, 6])
s.remove(2)
print(s)  # {1, 3, 4, 5, 6}
```

---

4. Dictionary (`dict`) Methods

Dictionaries store **key-value pairs**.

Method	Description
`get(key)`	Get value for key (returns `None` if not found)
`keys()`	Return all keys
`values()`	Return all values
`items()`	Return all key-value pairs as tuples
`update(dict2)`	Add/overwrite from another dictionary
`pop(key)`	Remove item by key
`clear()`	Remove all items

Example:

```
student = {"name": "Raj", "age": 24}
student["city"] = "Kathmandu"
student.update({"age": 25})
print(student.get("name"))  # Raj
print(student.keys())       # dict_keys(['name', 'age', 'city'])
```