**1. Set Methods**

A **set** is an unordered collection of unique elements. It doesn't allow duplicates and is often used for membership testing, eliminating duplicates from a sequence, and performing mathematical set operations (e.g., union, intersection).

**Common Set Methods:**

* **add(elem)**: Adds an element to the set.

my\_set = {1, 2, 3}

my\_set.add(4) # my\_set becomes {1, 2, 3, 4}

* **remove(elem)**: Removes an element from the set. Raises a KeyError if the element is not present.

my\_set.remove(2) # my\_set becomes {1, 3, 4}

* **discard(elem)**: Removes an element if it exists, but doesn't raise an error if it's not present.

my\_set.discard(5) # No error if 5 is not in the set

* **pop()**: Removes and returns an arbitrary element from the set. Since sets are unordered, you can't predict which element will be removed.

my\_set.pop() # Removes and returns an arbitrary element

* **clear()**: Removes all elements from the set, making it empty.

my\_set.clear() # my\_set becomes set()

* **union(other\_set)**: Returns a new set with elements from both sets (i.e., set union).

set1 = {1, 2, 3}

set2 = {3, 4, 5}

set1.union(set2) # Returns {1, 2, 3, 4, 5}

* **intersection(other\_set)**: Returns a new set with elements that are in both sets (i.e., set intersection).

set1 = {1, 2, 3}

set2 = {2, 3, 4}

set1.intersection(set2) # Returns {2, 3}

* **difference(other\_set)**: Returns a new set with elements in the current set but not in the other set (i.e., set difference).

set1 = {1, 2, 3}

set2 = {2, 3, 4}

set1.difference(set2) # Returns {1}

* **issubset(other\_set)**: Returns True if all elements of the current set are in the other set.

set1 = {1, 2}

set2 = {1, 2, 3}

set1.issubset(set2) # Returns True

* **issuperset(other\_set)**: Returns True if the current set contains all elements of the other set.

set1 = {1, 2, 3}

set2 = {1, 2}

set1.issuperset(set2) # Returns True

* **copy()**: Returns a shallow copy of the set.

my\_set\_copy = my\_set.copy()

**2. Dictionary Methods**

A **dictionary** is an unordered collection of key-value pairs. Keys are unique, and each key maps to a value.

**Common Dictionary Methods:**

* **get(key, default=None)**: Returns the value for the specified key. If the key is not found, it returns the default value (or None if not specified).

my\_dict = {"a": 1, "b": 2}

my\_dict.get("a") # Returns 1

my\_dict.get("c", "not found") # Returns 'not found'

* **setdefault(key, default=None)**: Returns the value for the key if it exists. If it doesn't exist, it sets the key with the default value and returns it.

my\_dict = {"a": 1, "b": 2}

my\_dict.setdefault("c", 3) # Adds 'c' with value 3 and returns 3

* **update(other\_dict)**: Updates the dictionary with elements from another dictionary or iterable. It will overwrite existing keys with new values.

my\_dict = {"a": 1, "b": 2}

my\_dict.update({"b": 3, "c": 4}) # my\_dict becomes {"a": 1, "b": 3, "c": 4}

* **keys()**: Returns a view object displaying all keys in the dictionary.

my\_dict = {"a": 1, "b": 2}

my\_dict.keys() # Returns dict\_keys(['a', 'b'])

* **values()**: Returns a view object displaying all values in the dictionary.

my\_dict.values() # Returns dict\_values([1, 2])

* **items()**: Returns a view object displaying all key-value pairs in the dictionary as tuples.

my\_dict.items() # Returns dict\_items([('a', 1), ('b', 2)])

* **pop(key)**: Removes and returns the value for the specified key. Raises a KeyError if the key doesn't exist.

my\_dict.pop("a") # Removes and returns the value of 'a', my\_dict becomes {"b": 2}

* **popitem()**: Removes and returns an arbitrary key-value pair (as a tuple). Since dictionaries are unordered, you can't predict which item will be removed.

my\_dict.popitem() # Removes and returns an arbitrary key-value pair

* **clear()**: Removes all key-value pairs from the dictionary, making it empty.

my\_dict.clear() # my\_dict becomes {}

* **copy()**: Returns a shallow copy of the dictionary.

my\_dict\_copy = my\_dict.copy()