SETS

1. What is a Set in Python?

- A **set** is an **unordered**, **unindexed** collection of **unique** elements.
- Sets are **mutable**, but they **cannot contain mutable elements** like lists or dictionaries.
- Defined using {} or the set() constructor.

```
my_set = {1, 2, 3}
```

Key Characteristics:

- No duplicate items.
- Items are **unordered**: no indexing or slicing.
- Used for membership testing and eliminating duplicates.

2. Access Set Items

You **can't access elements using index** because sets are unordered.

But you can loop through it:

```
fruits = {"apple", "banana", "cherry"}
for fruit in fruits:
  print(fruit)
```

You **cannot** do fruits[0] — will raise a TypeError.

3. Add Items to a Set

- Use .add() to add a single item.
- Use .update() to add multiple items.

```
colors = {"red", "blue"}
colors.add("green")          # Add one item
colors.update(["yellow", "pink"])  # Add multiple
print(colors)
```

4. Remove Items from a Set

- .remove(item): removes item — raises error if not found.
- .discard(item): removes item — no error if not found.

- .pop(): removes a random item.
- .clear(): empties the set.

```
colors = {"red", "green", "blue"}
colors.remove("green")
colors.discard("purple")  # No error
colors.pop()
print(colors)
```

5. Loop Through a Set

Use a for loop:

```
numbers = {10, 20, 30, 40}
for num in numbers:
   print(num)
```

6. Join Sets

You can **combine sets** using:

➤ **.union(): Returns a new set**

➤ **.update(): Adds from another set**

```
a = {"apple", "banana"}
b = {"cherry", "banana"}

# Union (new set)
c = a.union(b)
print(c)

# Update (modify a)
a.update(b)
print(a)
```

7. Set Operations

Operation	Description	Example
union()	Combines elements	a.union(b)
intersection()	Common elements	a.intersection(b)
difference()	Elements in a not in b	a.difference(b)
symmetric_difference()	In either a or b, not both	a.symmetric_difference(b)

8. Sort a Set

Sets are unordered. To sort, **convert it to a list**:

```
my_set = {3, 1, 4, 2}
sorted_list = sorted(my_set)
print(sorted_list)  # Output: [1, 2, 3, 4]
```

9. Set Methods Summary

Method	Purpose
.add(x)	Adds x to the set
.update([x, y])	Adds multiple elements
.remove(x)	Removes x (error if not found)
.discard(x)	Removes x (no error if not found)
.pop()	Removes a random item
.clear()	Empties the set
.union(set)	Returns new set with all items
.intersection(set)	Returns items in both sets
.difference(set)	Returns items only in first set
.symmetric_difference(set)	Items in either set but not both

```
# Python Set Methods - All in One Program

# Initial Set
```

```
fruits = {"apple", "banana"}
print("Initial Set:", fruits)

# 1. add(x)
fruits.add("orange")
print("\nAfter add('orange'):", fruits)

# 2. update([x, y])
fruits.update(["mango", "grape"])
print("After update(['mango', 'grape']):", fruits)

# 3. remove(x)
fruits.remove("banana")
print("After remove('banana'):", fruits)
# fruits.remove("kiwi")  #  Will raise KeyError if uncommented

# 4. discard(x)
fruits.discard("mango")
fruits.discard("kiwi")  # No error if element not found
print("After discard('mango') and discard('kiwi'):", fruits)

# 5. pop()
removed_item = fruits.pop()
print("After pop(), removed:", removed_item)
print("Set after pop():", fruits)

# 6. clear()
temp_set = fruits.copy()
temp_set.clear()
print("After clear():", temp_set)
```

```
# 7. union(set)
a = {1, 2, 3}
b = {3, 4, 5}
print("\nUnion of a and b:", a.union(b))

# 8. intersection(set)
print("Intersection of a and b:", a.intersection(b))

# 9. difference(set)
print("Difference of a and b:", a.difference(b))

# 10. symmetric_difference(set)
print("Symmetric Difference of a and b:", a.symmetric_difference(b))
```