1. Introduction to Sets المجموعات

- A **set** is a built-in Python data type that stores **unordered**, **unindexed**, and **unique** elements.
- Sets are mutable (can be changed), but they only store **immutable elements** (like numbers, strings, tuples).

Definition:

my_set = {1, 2, 3}
print(my_set) # {1, 2, 3}

Empty set (important):

empty_set = set() # not {} because {} creates a dictionary
print(type(empty_set)) # <class 'set'>

2. Properties of Sets

1. **Unordered** – no fixed position.
2. **No duplicates** – repeated items are removed.
3. **Mutable** – can add/remove items.
4. **Heterogeneous** – can store different data types.

Example:

```
s = {1, 2, 3, 2, "hello", (4,5)}
print(s)  # {1, 2, 3, 'hello', (4, 5)}  (duplicate "2" removed)
```

3. Creating Sets

```
# Using curly braces
set1 = {10, 20, 30}

# Using set() constructor
set2 = set([1, 2, 2, 3])  # duplicates removed
print(set2)  # {1, 2, 3}
```

4. Accessing Set Elements

- Sets don't support **indexing or slicing**.
- But you can iterate over them:

```
for item in {1, 2, 3}:
```

```
    print(item)
```

5. Adding Elements

```
s = {1, 2}
s.add(3)       # Add single element
print(s)       # {1, 2, 3}

s.update([4, 5, 6])  # Add multiple elements
print(s)       # {1, 2, 3, 4, 5, 6}
```

6. Removing Elements

```
s = {1, 2, 3, 4}

s.remove(2)   # Removes 2 (error if not found)
print(s)      # {1, 3, 4}

s.discard(5)  # No error if not found
print(s)      # {1, 3, 4}

s.pop()       # Removes random element
print(s)      # e.g., {3, 4}

s.clear()     # Removes all elements
print(s)      # set()
```

7. Set Operations (Mathematics)

Sets are useful for **mathematical operations**.

```
A = {1, 2, 3, 4}
B = {3, 4, 5, 6}

print(A | B)   # Union: {1, 2, 3, 4, 5, 6}
print(A & B)   # Intersection: {3, 4}
print(A - B)   # Difference: {1, 2}
print(B - A)   # Difference: {5, 6}
print(A ^ B)   # Symmetric Difference: {1, 2, 5, 6}
```

8. Membership Test

```
s = {1, 2, 3}
print(2 in s)   # True
print(5 not in s)  # True
```

9. Set Methods (Summary)

Method	Description
add(x)	Adds element x
update(iterable)	Adds multiple elements
remove(x)	Removes x, error if not found
discard(x)	Removes x, no error if missing
pop()	Removes random element
clear()	Removes all elements
union()	Returns union
intersection()	Returns intersection
difference()	Returns difference
symmetric_difference()	Returns symmetric difference
issubset()	Checks subset
issuperset()	Checks superset
isdisjoint()	Checks no common elements

Example:

```
A = {1, 2, 3}
B = {2, 3}
print(B.issubset(A))    # True
print(A.issuperset(B))  # True
print(A.isdisjoint({5})) # True
```

10. Frozenset (Immutable Set)

- frozenset is an **immutable version of set** (cannot add/remove elements).

```
fs = frozenset([1, 2, 3, 2])
print(fs)   # frozenset({1, 2, 3})

# fs.add(4) → ERROR (immutable)
```