🧠 1. What is a Set in Python?

👉 **Definition:**
A **set** is an **unordered**, **mutable (changeable)** collection of **unique elements**.
Duplicate values are **automatically removed**.

👉 **Syntax:**

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

👉 **Example:**

```
a = {1, 2, 3, 2, 3, 1}
print(a)  # Output: {1, 2, 3} (duplicates removed)
```

⚡ 2. Key Features of Sets

Feature	Description
Unordered	No index or order (like `{2,1}` same as `{1,2}`)
Unique items only	No duplicates allowed
Mutable	Can add or remove elements
Heterogeneous	Can contain different data types (e.g., `{1, "AI", 3.5}`)
Not hashable	You cannot store mutable types (like list or dict) inside a set

---

🧩 3. Creating Sets

✅ Using `{}`:

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

✅ Using `set()` constructor:

```
s = set([1, 2, 2, 3])
```

```
print(s)  # {1, 2, 3}
```

⚠ Empty set

```
a = {}       # ❌ This creates a dictionary
b = set()    # ✅ This creates an empty set
```

🛠 4. Common Set Methods (with Examples)

1️⃣ `add()`

Add one element.

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

2️⃣ `update()`

Add multiple elements (list, tuple, or another set).

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

3️⃣ `remove()`

Remove specific element → gives **error** if not found.

```
s = {1, 2, 3}
s.remove(2)
print(s)  # {1, 3}
```

4️⃣ `discard()`

Same as remove, but **no error** if element not found.

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

```
s.discard(4)  # no error
print(s)
```

5️⃣ `pop()`

Removes **any random element** (since sets are unordered).

```
s = {10, 20, 30}
x = s.pop()
print(x, s)
```

6️⃣ `clear()`

Removes all elements.

```
s = {1, 2, 3}
s.clear()
print(s)  # set()
```

🧮 5. Set Operations (VERY IMPORTANT for interviews)

Operation	Symbol	Example	Description
Union	`	`or`.union() `	`A
Intersection	`&` or `.intersection()`	`A & B`	Common elements only
Difference	`-` or `.difference()`	`A - B`	Elements in A not in B
Symmetric Difference	`^` or `.symmetric_difference()`	`A ^ B`	Elements not common to both

Example:

```
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(A ^ B)  # Symmetric Difference → {1, 2, 5, 6}
```

🧭 6. Comparison and Membership Methods

✅ `issubset()`

Checks if all elements of A are in B.

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

✅ `issuperset()`

Checks if A contains all elements of B.

```
print(B.issuperset(A))  # True
```

✅ `isdisjoint()`

Returns True if A and B have **no common elements**.

```
A = {1, 2}
B = {3, 4}
print(A.isdisjoint(B))  # True
```

✅ `in` and `not in`

```
s = {10, 20, 30}
print(20 in s)       # True
print(100 not in s)  # True
```

🔬 7. Set Comprehension (like list comprehension)

Quick way to create sets.

```
squares = {x**2 for x in range(5)}
print(squares)  # {0, 1, 4, 9, 16}
```

🧠 8. Real-world Uses of Sets in AI/ML

Use Case	Example
Remove duplicate data	`unique_labels = set(labels)`
Feature Engineering	Find unique categories or words in NLP dataset
Fast membership testing	Check if feature name or stopword exists
Data preprocessing	Compare unique values between datasets
Set operations for labels	e.g., find new classes not seen in training data

Example in ML:

```
train_labels = {'cat', 'dog', 'horse'}
test_labels = {'cat', 'lion'}

# Find unseen classes
unseen = test_labels - train_labels
print(unseen)  # {'lion'}
```

⚙️ 9. Advanced Topics

🔷 Frozenset (Immutable Set)

You can’t modify it once created (used as dictionary keys).

```
f = frozenset([1, 2, 3])
# f.add(4) ❌ Error: cannot add to frozenset
```

Use case → when you want a **read-only set**.

🧩 10. Interview-Level Practice Questions

1. What is the difference between `remove()` and `discard()`?

How do you remove duplicates from a list using set?

lst = [1,2,2,3,3,4]
unique = list(set(lst))

2.
3. What's the difference between `set()` and `{}`?

4. How to find common elements between two datasets?

5. Why are sets faster than lists for membership checking?
   → Because sets use **hashing** (constant-time lookup).

📜 11. Summary Notes (for quick revision)

Method	Purpose
`add()`	Add single element
`update()`	Add multiple elements
`remove()`	Remove element (error if not found)
`discard()`	Remove element (no error)
`pop()`	Remove random element
`clear()`	Remove all
`union()`	Combine sets
`intersection()`	Common elements

`difference()`	In A but not in B
`symmetric_difference()`	Uncommon elements
`issubset()`	A ⊆ B
`issuperset()`	A ⊇ B
`isdisjoint()`	No common items