🔢 Code:

```python
items = ["apple", "banana", "orange", "apple", "mango"] unique_item = set() for item in items: if item in unique_item: print("Duplicate: ", item) break unique_item.add(item)
```

✅ Step-by-Step Explanation:

🧺 Step 1: Your List

```python
items = ["apple", "banana", "orange", "apple", "mango"]
```

- You have a list of items (fruits), and some of them repeat.
- `"apple"` appears **twice**.

📦 Step 2: Create an Empty Set

```python
unique_item = set()
```

- You make a `set` named `unique_item`.
- A **set** in Python stores **unique** values — it does **not allow duplicates**.
- We'll use this to **track which items we've already seen**.

🔁 Step 3: Loop Through Each Item

python

```python
for item in items:
```

- You go through each item in the list **one by one**:
  1. "apple"
  2. "banana"
  3. "orange"
  4. "apple" ← Duplicate
  5. "mango"

⚠️ Step 4: Check If It's Already Seen

```python
if item in unique_item: print("Duplicate: ", item) break
```

- For each `item`, you check:
  - **Has this item been seen before?** (i.e., is it in the set?)
  - If **yes** → print that it's a duplicate and stop the loop using `break`.

➕ Step 5: Add to Set If Not Seen

```python
unique_item.add(item)
```

- If the item **was not already in the set**, you add it to `unique_item` so that it will be tracked for the next time.

🧪 Full Execution Flow:

Step	`item`	`unique_item` Set	Is it a duplicate?
1	"apple"	`{}`	No → add "apple"
2	"banana"	`{"apple"}`	No → add "banana"
3	"orange"	`{"apple", "banana"}`	No → add "orange"
4	"apple"	`{"apple", "banana", "orange"}`	Yes → Print and `break`
5	*loop stops here*	-	-

✅ Output:

```makefile
Duplicate:  apple
```

⚡ Summary:

- You're checking for the **first duplicate item**.
- You use a `set` to **track seen items**.
- The loop **stops immediately** when a duplicate is found.