📚 Module 22: Set Theory - Complete Notes

🎯 What You'll Learn

Master **sets** — unique collections that power deduplication, membership testing, and comparison operations.

📖 Concept Explained

The Basics

A **set** is an unordered collection of unique elements. No duplicates allowed!

```
Set A = {1, 2, 3, 4, 5}
Set B = {1, 1, 2, 2, 3} → Actually: {1, 2, 3}  (duplicates removed)

Empty set: {} or ∅
Universal set: Everything in our domain
```

Key Operations

```
a ∈ A     : a is an element of A
A ⊆ B     : A is subset of B (all of A is in B)
A ⊂ B     : A is proper subset (subset but not equal)
```

💻 Programming Connection

Code Examples

```python
# Example 1: Creating Sets

# From literal
numbers = {1, 2, 3, 4, 5}

# From list (removes duplicates!)
from_list = set([1, 2, 2, 3, 3, 3])
print(from_list)  # {1, 2, 3}

# Empty set (NOT {}, that's a dict!)
empty = set()
```

```python
# Example 2: Membership Testing (O(1) - super fast!)

valid_statuses = {"active", "pending", "completed"}

def is_valid(status):
    return status in valid_statuses
```

```python
print(is_valid("active"))   # True
print(is_valid("deleted"))  # False
```

```python
# Example 3: Deduplication

data = [1, 3, 2, 1, 4, 3, 5, 2]

# Simple dedup (loses order)
unique = list(set(data))
print(unique)  # [1, 2, 3, 4, 5] (order may vary)

# Dedup preserving order (Python 3.7+)
unique_ordered = list(dict.fromkeys(data))
print(unique_ordered)  # [1, 3, 2, 4, 5]
```

```python
# Example 4: Find Duplicates

def find_duplicates(items):
    """Find items that appear more than once"""
    seen = set()
    duplicates = set()
    for item in items:
        if item in seen:
            duplicates.add(item)
        seen.add(item)
    return list(duplicates)

print(find_duplicates([1, 2, 2, 3, 3, 3]))  # [2, 3]
```

🧪 SDET/Testing Application

```python
# SDET Scenario: Verify Unique IDs

def verify_unique_ids(records):
    """Check for duplicate IDs in records"""
    ids = [r['id'] for r in records]
    unique_ids = set(ids)

    return {
        "total": len(ids),
        "unique": len(unique_ids),
        "has_duplicates": len(ids) != len(unique_ids),
        "duplicates": find_duplicates(ids)
    }

records = [{"id": 1}, {"id": 2}, {"id": 1}, {"id": 3}]
```

```
print(verify_unique_ids(records))
# {'total': 4, 'unique': 3, 'has_duplicates': True, 'duplicates': [1]}
```

🔑 Key Takeaways

✅ **Sets = No duplicates**
✅ **Membership testing is O(1)** — Super fast
✅ **Convert list to set** — Instant dedup

💾 *Save as:* `Module_22_Set_Theory.md`