📚 Module 2: Comparison & Logic - Complete Notes

🎯 What You'll Learn

In this module, you'll master the art of **making decisions in code**. Every `if` statement, every filter, every validation — they all come down to comparison. By the end, you'll think in conditions naturally.

📖 Concept Explained (Like a YouTube Video)

The Basics

Alright, welcome back! In this module, we're tackling something you'll use in literally **every program you ever write** — comparisons!

Think about it: How does your code decide what to do? It **asks questions** and gets YES/NO answers.

- Is the user logged in? → YES → Show dashboard
- Is the password correct? → NO → Show error
- Is the response time under 200ms? → YES → Test passes

These are all **comparisons**!

The Six Comparison Operators

Think of these as the **six questions** you can ask:

```
┌──────────┬──────────────────────────┬──────────────┐
│ OPERATOR │ QUESTION                 │ EXAMPLE      │
├──────────┼──────────────────────────┼──────────────┤
│    >     │ Is left GREATER than right?│ 10 > 5  ✅ │
│    <     │ Is left LESS than right?   │ 3 < 7   ✅ │
│    ==    │ Are they EQUAL?            │ 5 == 5  ✅ │
│    !=    │ Are they DIFFERENT?        │ 5 != 3  ✅ │
│    >=    │ Is left >= right?          │ 5 >= 5  ✅ │
│    <=    │ Is left <= right?          │ 4 <= 7  ✅ │
└──────────┴──────────────────────────┴──────────────┘
```

Visual Understanding

Let's visualize on a number line:

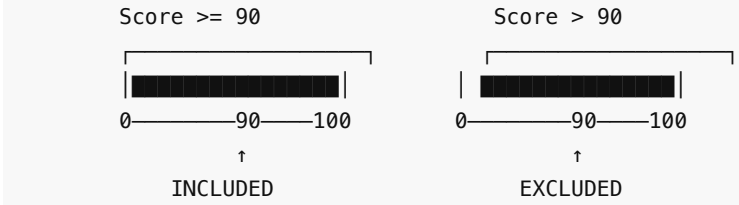
```
        ┌─────────────────────────────────────────┐
        │   0   1   2   3   4   5   6   7   8   9   │
        └─────────────────────────────────────────┘

        For x = 5:

        x > 3  → TRUE  (5 is to the RIGHT of 3)
        x < 3  → FALSE (5 is NOT to the LEFT of 3)
        x == 5 → TRUE  (same position)
        x >= 5 → TRUE  (same or right)
        x >= 6 → FALSE (5 is left of 6)
```

The Key Insight: Boundaries

The difference between `>` and `>=` is **THE BOUNDARY**:



💻 Programming Connection

Code Examples

```
# Example 1: Simple Comparison
age = 25

if age >= 18:
    print("Adult - can vote")
else:
    print("Minor - cannot vote")
```

```
# Example 2: Range Check
score = 85

if score >= 80 and score <= 89:
    grade = "B"
elif score >= 90:
    grade = "A"
elif score >= 70:
    grade = "C"
else:
    grade = "F"
```

```
# Example 3: Finding Maximum
def find_maximum(a, b, c):
    max_val = a
    if b > max_val:
        max_val = b
    if c > max_val:
        max_val = c
    return max_val
```

🧪 SDET/Testing Application

```python
# SDET Scenario: Response Validation

def validate_api_response(response):
    results = {
        "status_valid": 200 <= response["status_code"] < 300,
        "time_valid": response["time_ms"] <= 200,
        "data_valid": response["item_count"] > 0
    }
    results["overall"] = all(results.values())
    return results
```

🔑 Key Takeaways

✅ **Comparison always returns True or False**
✅ **`>=` and `<=` INCLUDE the boundary**
✅ **`>` and `<` EXCLUDE the boundary**
✅ **Range checks need two comparisons with AND**

Next up: Module 3 - Number Properties (Even & Odd)! 🚀