Day 4 – Python Conditional Statements (Decision Making)

Simple + Detailed Notes

1️⃣ What is Decision Making in Python?

Decision making means **choosing one action from many options** based on a condition.

Python checks a condition (True or False) and then decides what code to run.

Example in real life:

- If it's raining → take an umbrella
- If marks ≥ 40 → pass
- If age ≥ 18 → eligible for voting

Python works the same way.

2️⃣ What is a Condition?

A **condition** is a statement that becomes either **True or False**.

Examples:

- `10 > 5` → True
- `age >= 18` → True/False based on age
- `marks == 100` → True/False

We write conditions using comparison operators like:

`==`, `!=`, `>`, `<`, `>=`, `<=`

3️⃣ The `if` Statement

Used when we want to run a block of code only **if** a condition is True.

Syntax

```
if condition:
    # code
```

Example

```
age = 18
if age >= 18:
    print("You are eligible to vote.")
```

4️⃣ The `if-else` Statement

Used when there are two possible outcomes:

- If condition is True → run *if block*
- Else → run *else block*

Syntax

```
if condition:
    # code 1
else:
    # code 2
```

Example

```
marks = 35
if marks >= 40:
    print("Pass")
else:
```

```
    print("Fail")
```

5️⃣ The `elif` Statement (Else If)

Used when we want to check **multiple conditions one after another**.

Python checks each condition until one becomes True.

Syntax

```
if condition1:
    # code
elif condition2:
    # code
elif condition3:
    # code
else:
    # final code
```

Example

```
marks = 85

if marks >= 90:
    print("Grade A")
elif marks >= 80:
    print("Grade B")
elif marks >= 70:
    print("Grade C")
else:
    print("Grade D")
```

6️⃣ Nested If (If inside another If)

This is used when one decision depends on another decision.

Syntax

```
if condition:
    if condition2:
        # nested code
```

Example

```
age = 20
id_proof = True

if age >= 18:
    if id_proof:
        print("Allowed inside")
    else:
        print("ID Required")
else:
    print("Not allowed")
```

7️⃣ Indentation (Very Important)

Indentation means **space before the code**.

In Python, indentation decides **which code belongs to which block**.

Example:

```
if age >= 18:
    print("Adult")  # inside if
```

```
print("Done")      # outside if
```

8️⃣ Logical Operators in Conditions

Used to combine multiple conditions.

Operator	Meaning	Example
`and`	True if both conditions are True	`age >= 18 and id == True`
`or`	True if at least one condition is True	`marks >= 90 or sports == True`
`not`	Reverses condition	`not is_raining`

9️⃣ Real-Life Examples

Example 1 – Voting Eligibility

```
age = int(input("Enter age: "))

if age >= 18:
    print("Eligible for voting")
else:
    print("Not eligible")
```

Example 2 – Even or Odd

```
num = int(input("Enter number: "))

if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

Example 3 – Simple Menu (Without match-case)

```
option = 2

if option == 1:
    print("Pizza")
elif option == 2:
    print("Burger")
elif option == 3:
    print("Momos")
else:
    print("Invalid choice")
```

🔟 Common Errors in Conditional Statements

❌ Missing indentation

❌ Using `=` instead of `==`

❌ Writing conditions without variables

❌ Not closing quotes properly

❌ Writing colon (:) incorrectly

✔️ Summary (Simple)

- Conditions help computers make decisions
- `if` is used for a single condition
- `if-else` for two outcomes
- `elif` for multiple options
- Nested if for complex decisions

- Indentation is extremely important