Conditional Statements in Python

1. Introduction:

Conditional statements allow you to execute specific blocks of code depending on whether a condition is true or false. These statements are the foundation of decision-making in any program.

2. Importance:

- Helps in decision-making
- Enables branching logic
- Makes programs dynamic and interactive

3. Types of Conditional Statements:

- if statement
- if-else statement
- if-elif-else ladder
- Nested if statements

4. if Statement:

Executes a block if a condition is true.

Example:

```
x = 10
if x > 5:
    print("x is greater than 5")
```

5. if-else Statement:

Adds an alternative block if the condition is false.

Example:

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

6. if-elif-else Statement:

Used for multiple conditions.

Example:

```
score = 85
if score >= 90:
    print("Grade A")
elif score >= 75:
    print("Grade B")
else:
    print("Grade C")
```

7. Nested if Statements:

An if block within another if block.

Example:

```
x = 10
if x > 0:
    if x < 100:
        print("x is a positive number less than 100")
```

8. Logical Operators:

Used to combine conditions:

- and, or, not

Example:

```
if age > 18 and age < 60:
    print("Eligible")
```

9. Comparison Operators:

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

10. Practice Questions:

1. Write a program to check if a number is positive, negative, or zero.
2. Check whether a user-provided year is a leap year.
3. Create a simple grading system based on marks.
4. Determine if a character is a vowel or consonant.
5. Compare three numbers and print the largest.

11. Summary:

Conditional statements make your programs smart by enabling decision-making. You can control the flow based on logical conditions, making Python suitable for real-world scenarios.