# Conditional Statements in Python

Conditional statements let you run **specific blocks of code** only when certain **conditions are true**.

### 1. `if` Statement

```
x = 10

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

If the condition (`x > 5`) is **True**, the indented block is executed.

### 2. `if...else` Statement

```
x = 3

if x > 5:
    print("x is greater than 5")
else:
    print("x is not greater than 5")
```

Executes the `else` block when the `if` condition is **False**.

### 3. `if...elif...else` Ladder

Use `elif` (short for **else if**) to check **multiple conditions**:

```
x = 5

if x > 5:
    print("x is greater than 5")
elif x == 5:
    print("x is equal to 5")
else:
    print("x is less than 5")
```

Only **one block** executes, based on the **first true condition**.

### 4. Nested if Statements

You can place one `if` inside another.

```
x = 10
y = 20

if x > 5:
    if y > 10:
        print("x > 5 and y > 10")
```

### 5. Short-Hand `if` (One-Liner)

If you only have one statement to execute, you can write it in one line:

```
x = 7
if x > 5: print("x is greater than 5")
```

### 6. Ternary Operator (Short-Hand `if...else`)

```
x = 8
result = "Even" if x % 2 == 0 else "Odd"
print(result)
```

Great for **assigning a value** based on a condition.

# Logical Operators in Conditions

You can combine conditions with:

- `and` → both must be true
- `or` → at least one must be true
- `not` → inverts a condition

```
x = 10
y = 5

if x > 5 and y < 10:
    print("Both conditions are true")
```