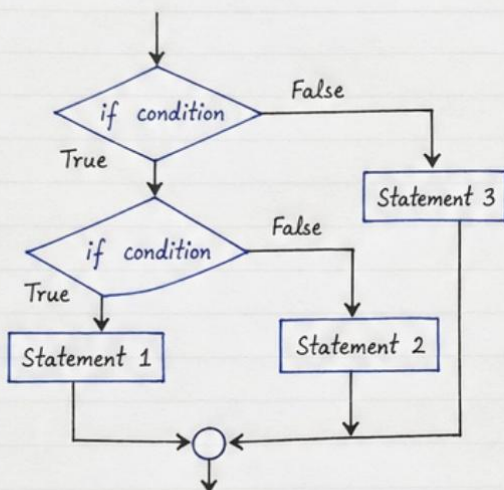


## UNIT 3 : CONDITIONAL STATEMENTS - Nested if & elif

### 1. NESTED IF

When an if or else block contains another if or else block, it is called **Nested if**.

#### FLOWCHART :



#### COMMON MISTAKES :

- Missing indentation
- Using assignment (=) instead of comparison (==)
- Forgetting the inner block
- Confusing multiple conditions

**TIP :** Always use proper indentation and test each condition carefully.

#### KEY POINTS :

- Nested if is used inside another if/else block.
- elif is used for multiple conditions.
- Python checks conditions from top to bottom.
- Use else for the default case.

### 2. ELIF LADDER

Used to check multiple conditions. The conditions are checked in order. Once one is **True**, the rest are skipped.

#### SYNTAX :

```

if condition1 :
    statement 1
elif condition2 :
    statement 2
elif condition3 :
    statement 3
.....
else :
    statement n
  
```

#### EXAMPLE :

```

marks = 85
if marks >= 90 :
    print("Grade : A+")
elif marks >= 75 :
    print("Grade : A")
elif marks >= 60 :
    print("Grade : B")
elif marks >= 40 :
    print("Grade : C")
else :
    print("Fail")
  
```

#### OUTPUTS :

If marks = 92  
Output :  
Grade : A+

If marks = 78  
Output :  
Grade : A

If marks = 55  
Output :  
Grade : B

If marks = 35  
Output :  
Fail

★ Practice more examples to become confident !



SP Group of Institute

## UNIT 3 : CONDITIONAL STATEMENTS

### — match-case & Practice

#### 1. match-case Statement (Python 3.10+)

The match-case statement is used to compare a value against multiple patterns. It is more readable and efficient than multiple if-elif-else statements.

##### Syntax:

```
match expression:
    case value1:
        # code block
    case value2:
        # code block
    ...
    case _:
        # default code block
```

##### Output (Example):

```
Enter a number (1-7): 3
Wednesday
```

##### Interview Question:

Q. What is the difference between if-elif-else ladder and match-case statement?

Ans: match-case is more readable, supports pattern matching, and can handle multiple data types.

##### Example:

```
x = int(input("Enter a number (1-7): "))
match x:
    case 1:
        print("Monday")
    case 2:
        print("Tuesday")
    case 3:
        print("Wednesday")
    case 4:
        print("Thursday")
    case 5:
        print("Friday")
    case 6:
        print("Saturday")
    case 7:
        print("Sunday")
    case _:
        print("Invalid input! Please enter 1-7.")
```

#### 2. Practice Programs

1. Write a program to find the day of the week using match-case.
2. Write a program to perform basic calculator operations using match-case.
3. Write a program to grade a student based on marks (A, B, C, D, F) using match-case.
4. Write a program to check the type of shape (circle, square, triangle, rectangle) and display its name using match-case.

\* **Note:** The case \_ (underscore) acts as a default case, similar to else in if-else.

#### 3. Flowcharts - Common Symbols

| Symbol | Name           | Use                          |
|--------|----------------|------------------------------|
|        | Terminator     | Start / End of a program     |
|        | Process        | Instruction or operation     |
|        | Input / Output | Read input or Display output |
|        | Decision       | Condition check (Yes / No)   |
|        | Flow Line      | Direction of flow            |

\* **Tip:** Use meaningful variable names and proper indentation for better readability. 😊



SP Group of Institute

## UNIT 3: CONDITIONAL STATEMENTS - if & if-else

Conditional Statements are used to make decisions in a program based on certain conditions.

### 1. if STATEMENT

#### → Concept:

The if statement is used to execute a block of code only if a given condition is **True**.

#### → Syntax:

```
if (condition):  
    # block of code
```

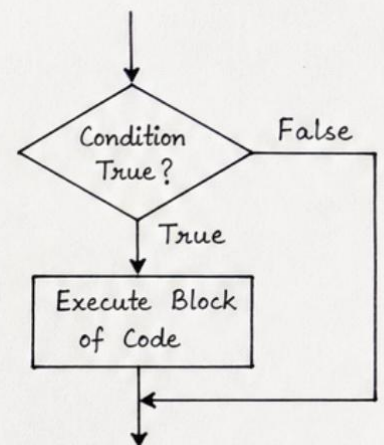
#### → Example:

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

#### → Output:

x is greater than 5

#### Flowchart:



### 2. if-else STATEMENT

#### → Concept:

The if-else statement is used to execute one block of code if the condition is **True** and another block if the condition is **False**.

#### → Syntax:

```
if (condition):  
    # block of code if condition is True  
else:  
    # block of code if condition is False
```

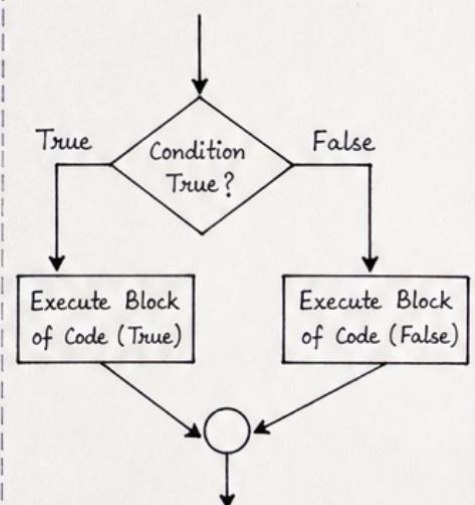
#### → Example:

```
x = 3  
if x >= 5:  
    print("x is greater than or equal to 5")  
else:  
    print("x is less than 5")
```

#### → Output:

x is less than 5

#### Flowchart:



\* Note: Use proper indentation in Python. It is very important.