**If Else Conditional Statements**

-- There comes situations in real life when we need to make some decisions and based on these decisions, we decide what should we do next.

-- Similar situations arise in programming also where we need to make some decisions and based on these decisions we will execute the next block of code.

-- Decision-making statements in programming languages decide the direction(Control Flow) of the flow of program execution.

Types of Control Flow in Python

1. The if statement

2. The if-else statement

3. The nested if-statement

4. The if-elif-else ladder

IF Statement

-- The if statement is the most simple decision-making statement.

-- It is used to decide whether a certain statement or block of statements will be executed or not.

-- Syntax:

*if condition:*

*# Statements to execute if*

*# condition is true*

i = 10

*if* i < 15:

    print("Hi There")

If-Else Statement

-- The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won’t.

-- But if we want to do something else if the condition is false, we can use the *else* statement with *if* statement to execute a block of code when the if condition is false.

-- Syntax:

*if (condition):*

*# Executes this block if*

*# condition is true*

*else:*

*# Executes this block if*

*# condition is false*

i = 20

*if* i < 20:

    print("less")

*else*:

    print("Greater")

Nested IF Statement

-- A nested if is an if statement that is the target of another if statement.

-- Nested if statements mean an if statement inside another if statement.

-- Yes, Python allows us to nest if statements within if statements. i.e., we can place an if statement inside another if statement.

-- Syntax:

*if (condition1):*

*# Executes when condition1 is true*

*if (condition2):*

*# Executes when condition2 is true*

*# if Block is end here*

*# if Block is end here*

i = 10

*if* i < 15:

    print("Less than 15")

*if* 1 < 12:

        print("Less than 12 too")

*else*:

    print("Greater than 15")