

# Day 2: Python Basics II

## 1. Conditional Statements

Conditional statements allow a program to make decisions based on specific conditions. Python evaluates conditions as either True or False and executes code accordingly.

### [1] if Statement

The if statement is used to execute a block of code only when a condition is true.

Concept:

- Python checks the condition.
- If the condition is True, the associated code block runs.
- If False, the code block is skipped.

Key Points:

- Conditions use comparison and logical operators.
- Indentation defines the scope of the if block.

### [2] if-else Statement

The if-else structure provides two possible execution paths.

Concept:

- If the condition is True, the if block executes.
- If the condition is False, the else block executes.

### [3] if-elif-else Statement

The elif (else if) statement is used to check multiple conditions sequentially.

Concept:

- Conditions are evaluated from top to bottom.
- The first True condition executes its block.
- If none are True, the else block executes.

#### [4] Nested Conditional Statements

A nested condition means placing an if statement inside another if or else block.

Concept:

- Used when decisions depend on multiple levels of conditions.
- Inner conditions are checked only if outer conditions are met.

## 2. Loops

Loops are used to repeat a block of code multiple times, reducing redundancy and improving efficiency.

#### [1] for Loop

The for loop is used to iterate over a sequence such as a list, string, or range.

Concept:

- Executes once for each element in the sequence.
- Automatically stops when the sequence ends.

Characteristics:

- Ideal when the number of iterations is known.
- Commonly used with collections and counters.

## [2] while Loop

The while loop executes as long as a condition remains True.

Concept:

- Condition is checked before each iteration.
- Loop continues until the condition becomes False.

Characteristics:

- Used when the number of iterations is unknown.
- Requires careful condition management to avoid infinite loops.

## [3] range() Function

The range() function generates a sequence of numbers, often used with for loops.

Concept:

- Can generate a sequence with a start, stop, and step value.
- Produces numbers one at a time, improving memory efficiency.

## 3. Control Flow Statements

Control flow statements alter the normal execution of loops.

### [1] break Statement

The break statement is used to terminate a loop immediately.

Concept:

- Stops the loop execution entirely.
- Control moves to the statement following the loop.

## [2] continue Statement

The continue statement skips the current iteration and moves to the next one.

Concept:

- Does not end the loop.
- Only skips the remaining code in the current iteration.

## [3] pass Statement

The pass statement acts as a placeholder.

Concept:

- Does nothing when executed.
- Prevents syntax errors when a statement is required but no action is needed.

## 4. Nested Loops

Nested loops occur when one loop is placed inside another loop.

### [1] Understanding Loop Nesting

Concept:

- The inner loop runs completely for each iteration of the outer loop.
- Execution follows a hierarchical structure.

Execution Pattern:

- Outer loop → Inner loop → Inner completes → Outer moves to next iteration.

## [2] Common Nested Loop Patterns

### 1. Pattern Generation

- Used for creating shapes such as triangles or grids.
- Outer loop controls rows.
- Inner loop controls columns.

### 2. Table and Matrix Processing

- Used in multi-dimensional data structures.
- Common in mathematical and data-processing applications.

### 3. Comparison Tasks

- Comparing elements from two datasets.
- Matching pairs or combinations.