

CLASS:2

Python

1. Dynamic Typing in Python

Dynamic typing means that Python does not require explicit declaration of variable data types. The type of a variable is automatically determined at runtime based on the value assigned to it. A variable can refer to different data types at different times during program execution.

Key Points:

- No need to declare data types explicitly
- Type checking happens during execution
- Provides flexibility but requires careful handling to avoid runtime errors

2. Mutable and Immutable Data Types

Mutable data types are those whose values can be modified after creation, whereas immutable data types cannot be changed once they are created. Any modification to an immutable object results in the creation of a new object.

Examples:

- Mutable: List, Dictionary, Set
- Immutable: Integer, Float, String, Tuple

Importance:

Understanding mutability helps in efficient memory usage and prevents unexpected changes in data.

3. Local and Global Variables

Local variables are defined inside a function and are accessible only within that function. Global variables are defined outside all functions and can be accessed throughout the program.

Key Difference:

- Local variables have limited scope
- Global variables have program-wide scope

4. Purpose of the Return Statement

The return statement is used to send a result from a function back to the caller. It also terminates the execution of the function. Without a return statement, a function returns **None** by default.

Significance:

- Enables functions to produce outputs
- Helps in modular and reusable code design

5. List Comprehension

List comprehension is a concise way to create lists using a single, readable expression. It combines looping and conditional logic into one compact structure.

Advantages:

- Reduces code length
- Improves readability
- Often faster than traditional looping methods

6. Difference Between Break, Continue, and Pass

- **Break** is used to immediately terminate a loop.
- **Continue** skips the current iteration and proceeds to the next one.
- **Pass** is a null operation used as a placeholder where a statement is syntactically required.

Use Case:

These statements provide control over loop execution flow.

7. Recursion

Recursion is a programming technique where a function calls itself to solve a problem by breaking it into smaller subproblems. Every recursive function must include a base condition to prevent infinite calls.

Applications:

- Mathematical computations
- Tree and graph traversal
- Divide-and-conquer algorithms

8. Lambda Function

A lambda function is a small, anonymous function defined without a name. It is typically used for short, simple operations where defining a full function is unnecessary.

Characteristics:

- Written in a single line
- Can take multiple arguments
- Contains only one expression

9. Difference Between **is** and **==**

The **==** operator checks whether two variables have the same value, whereas the **is** operator checks whether both variables refer to the same object in memory.

Key Insight:

Two objects may be equal in value but not identical in memory.

10. Docstrings and Their Importance

Docstrings are documentation strings used to explain the purpose, parameters, and behavior of functions, classes, or modules. They are written immediately after the definition and are enclosed in triple quotes.

Importance:

- Improves code readability
- Assists in automatic documentation generation
- Facilitates collaboration and maintenance