MSLC-PYTHON-002

1. Introduction to Python

Python is a high-level, interpreted, and general-purpose programming language known for its simplicity and readability. Python emphasizes code readability with its clean syntax, which makes it ideal for beginners and professionals alike.

Key Features:

1. **Easy to Learn and Use**: Python has an intuitive syntax resembling natural language, making it beginner-friendly.
2. **Versatile**: It supports multiple programming paradigms, including object-oriented, procedural, and functional programming.
3. **Rich Libraries**: Python comes with an extensive standard library and a vibrant ecosystem of third-party libraries for tasks like data analysis, web development, machine learning, and more.
4. **Platform-Independent**: Python is cross-platform, meaning code can run on different operating systems with minimal changes.
5. **Dynamic Typing**: Variable types are determined at runtime, adding flexibility in coding.

Applications:

- **Web Development**: Frameworks like Django and Flask.
- **Data Science & Machine Learning**: Libraries such as Pandas, NumPy, and TensorFlow.
- **Automation**: Scripting for repetitive tasks.
- **Game Development**: Tools like Pygame.
- **IoT and Hardware**: Raspberry Pi and MicroPython.

Python's simplicity, flexibility, and community support have contributed to its popularity, making it a top choice for developers worldwide.

2. Data Types-I

Python has several built-in data types.These data types form the foundation of working with data in Python:

In Python, data types are classifications that specify which type of value a variable can hold. Python supports several built-in data types, each serving a specific purpose.

1. int
2. float
3. string

3. Data Types- II

Lists: A list is a data type that stores a collection of items, which can be of any data type, including strings, integers, floats, and other lists. Lists are denoted by square brackets [] and are ordered, meaning that each item has a specific index or position. Lists are versatile data structures that can store a collection of items in a specific order. Lists are mutable, meaning their contents can be changed after they are created.

T**uple**: Tuple is an immutable built-in data type in python. Tuples are created using " ()" brackets.We can not change the existing data but we can add new data at the last position of tuple.
Tuples are immutable. This means that once a tuple is created, its elements cannot be modified, added, or removed. Tuples are defined using parentheses ().

Set: Set is the built-in data type like other data types. Set is a collection of unordered and unindexed data.

Dictionary : Dictionary is a built in data type of python , this data type stores values in "key" and "value" pairs. To define a dictionary data type we use { } .

4. String Methods

String methods are built-in functions that allow you to perform operations on strings, such as modifying them, extracting information, or performing transformations. Here are some commonly used string methods in Python.

1. len( )
2. lower( )
3. upper( )
4. strip( )
5. islower( )
6. isupper( )

5. Operators

Operators are the special type of symbol that is used to perform operations.
Type of operators

1. Arithmetic operators
2. Assignment operators
3. Comparison operators
4. Logical operators
5. Identity operators
6. Membership operators
7. Bitwise operators

6. Control Flow: Conditional Statement

Control flow is a term of programming, where a program writes a code in a way of controlling execution. Control flow controls the flow of the program. Logics create and control flow checks that logic can be anything but the program executes by the defined rule of code.
Types of control flow
1. Conditional Statement
2. Loops

1. Conditional Statement : Some conditions are set into a program to execute a block of code. Conditional statement works on the basis of condition , A condition is true than it runs a block of code otherwise terminate execution. Condition statements follow operators to create or make conditions.
    - if
    - if - else
    - elif

7. Control Flow: Loops

Loops are used to execute a block of code repeatedly based on a condition. There are two main types of loops in Python:

For Loop : The for loop is used to iterate over a sequence (like a list, tuple, string, or range) and execute a block of code for each element in the sequence. It is generally used when the number of iterations is known or when iterating over a collection of items
While Loop : The while loop repeats a block of code as long as a specified condition is true. It is useful when the number of iterations is not known in advance and you want to keep looping until a certain condition is met.

8. Functions

A function in Python is a block of reusable code designed to perform a specific task. Functions allow you to organize your code into smaller, more manageable pieces, which helps make your code more modular, readable, and easier to maintain.

Parameters and Arguments
Parameters and arguments are closely related concepts in functions, but they are used in slightly different ways. Understanding the distinction between the two is important when working with functions.

9. Exception Handling

Exception Handling is a mechanism in programming that allows a program to handle runtime errors (exceptions) gracefully, without crashing. It allows developers to anticipate and manage errors that might occur during the execution of a program, ensuring the program can either recover or provide useful feedback to the user.

10. Object Oriented Programming(OOP's)

Object-Oriented Programming (OOP) is a programming paradigm that uses "objects" to model real-world entities and organizes code around these objects rather than functions and logic. It aims to make code more modular, reusable, and maintainable by structuring it in a way that mimics real-world interactions.

a. Classes: A class is a blueprint or template for creating objects (instances). It defines the attributes (variables) and methods (functions) that the objects created from the class will have
b. Objects are instances of classes. They represent real-world entities that have both properties (attributes) and behaviors (methods).

11. Inheritance - Polymorphism

Inheritance allows a new class to inherit attributes and methods from an existing class, promoting code reuse. The new class is called a subclass, and the existing class is the superclass.

Polymorphism allows methods to take on many forms. It refers to the ability of different classes to provide a method with the same name but with different behaviors.

12. Encapsulation-Abstraction

Encapsulation refers to the bundling of data (attributes) and methods (functions) that operate on that data within one unit (class). It helps in restricting access to certain components to protect the data.

Abstraction involves hiding complex implementation details and showing only the necessary features of an object. This can be achieved using abstract classes and interfaces in some programming languages

13. SQL

SQL (Structured Query Language) is a standard language used to interact with relational databases. It is divided into two main categories based on the tasks it performs: Data Definition Language (DDL) and Data Manipulation Language (DML).

Data Definition Language (DDL): DDL is used to define, manage, and structure the database schema. It focuses on creating, altering, and deleting database objects like tables, indexes, and views

1. CREATE
2. ALTER
3. DROP

14. SQL II :

Data Manipulation Language (DML):

DML is used to manipulate the data stored in the database. These commands allow you to insert, update, delete, and retrieve data from tables.

1. SELECT
2. INSERT
3. UPDATE
4. DELETE

15. Decorator & Generator

A **decorator** in Python is a function that allows you to modify or extend the behavior of another function or method without permanently modifying it. Decorators are commonly used for adding functionality like logging, access control, or performance measurement to existing functions.

A **generator** is a special type of iterator in Python that allows you to iterate over a sequence of values, but unlike regular functions, a generator can produce a sequence of values lazily (one value at a time) using the yield keyword.

16. ChatGPT API Calling

The ChatGPT API is a service provided by OpenAI that allows developers to integrate ChatGPT (a large language model) into their applications, websites, or systems. Using the API, you can send text-based input to the model and receive responses in real time, enabling natural language processing (NLP) capabilities in your software