



Module 32: Python Course – Welcome and Setup

1. Welcome to the Python Course

This module provides an overview of **Python programming**—a powerful, easy-to-learn language used in data analytics, web development, AI, automation, and more.

◆ 1. Python Overview

 **Python** is a **high-level**, **interpreted**, and **object-oriented** programming language known for its **simplicity** and **readability**.

 Used for:

- Data Analysis
- Web Development
- Automation
- Machine Learning & AI
- Scripting & APIs
- App/Game Development

★ 3. Advantages of Python

 Advantage	 Description
 Simple & Easy	Clean syntax, readable like English. Great for beginners.
 Open Source	Free to use and community-supported.
 Short Development Time	Fewer lines of code, faster development.
 Readable	Indentation-based structure improves clarity.
 Interpreted Language	Executes line-by-line, great for debugging.
 Portable & Versatile	Works on Windows, Linux, macOS.
 Large Library Support	Built-in and external libraries make complex tasks simple.

🧠 4. Key Characteristics of Python

 Term	 Meaning
 Interpreted	Code runs line-by-line; no need to compile first.
 Object-Oriented	Uses objects and classes to structure code.
 High-Level	Closer to human language than machine language.
 Dynamically Typed	Data type of a variable is inferred at runtime.

5. Difference: Interpreted vs. Compiled Languages

 Feature	 Interpreted	 Compiled
Execution	Line-by-line	Whole code at once
Speed	Slower	Faster
Debugging	Easier (shows line-wise error)	Harder (error shown after compiling)
Example	Python, JavaScript	C, C++

6. Dynamic Semantics in Python

- ◆ **Dynamic semantics** means the meaning of variables, types, or expressions is determined during **runtime**, not during compilation.
- ◆ This makes Python **flexible** but may also lead to **runtime errors** if not handled carefully.

7. Object-Oriented Programming in Python

Python supports:

-  **Classes & Objects**
-  **Inheritance**
-  **Encapsulation**
-  **Polymorphism**

This helps in building **modular, reusable, and structured** code.

8. Uses of Python

 Python is used in:

-  **Data Science**
-  **Artificial Intelligence & Machine Learning**
-  **Web Development** (Flask, Django)
-  **Scientific Computing**
-  **Data Visualization**
-  **Automation & Scripting**
-  **App & Game Development**

9. Python Libraries

Built-in (Standard) Libraries:

- `math` , `datetime` , `os` , `random` , `re` , `json`

External Libraries (Common in Data & Analytics):

Library	Purpose
 NumPy	Numerical operations, multi-dimensional arrays
 Pandas	Data manipulation and analysis (DataFrames)
 Matplotlib	Basic data visualization
 Seaborn	Advanced statistical data visualization
 Scikit-learn	Machine learning algorithms
 BeautifulSoup	Web scraping
 Requests	HTTP requests

 Install external libraries using `pip install <library-name>`

Key Points to Remember

-  Python is **beginner-friendly** but **powerful** enough for advanced use cases
-  It supports **multi-paradigm programming** (procedural, object-oriented, functional)
-  Use **external libraries** to speed up development and avoid writing everything from scratch
-  **Interpreted nature** makes debugging easier but may be slower than compiled languages
-  Popular among **developers, analysts, scientists, and AI engineers**