MODULE 1: Python Environment & Introduction

Objective

To learn how to install Python, set up your development environment, understand how Python code executes, and write you're first working programs.

1. Installing Python

Python is a high-level, interpreted programming language known for its simplicity, readability, and community support. It runs programs directly using the interpreter, without manual compilation.

Steps to Install Python (Windows):

- Visit https://www.python.org/downloads/
- Download the latest version
- Check 'Add Python to PATH' during installation
- Verify installation using: python --version

For macOS/Linux:

- Run `python3 --version` to check if Python is installed.
- If not, use: sudo apt install python3

2. IDE Setup (Integrated Development Environment)

- An IDE helps you write, test, and debug code efficiently with features like syntax highlighting, auto-completion, and debugging.
- **Common IDEs:** VS Code, PyCharm, and Jupyter Notebook.
  In VS Code, install the Python extension and run files directly using Run ▶ or python filename.py.

3. Environment Variables

Environment variables tell your OS where to find installed software like Python and pip. Add Python's installation path to the system PATH for easy command-line use.

4. Pip & Virtual Environment

Pip (Python Installer for Packages) helps install and manage external libraries.

Examples:

- pip install requests
- pip list
- pip uninstall package_name

A virtual environment isolates project dependencies.

- **Create one using:** python -m venv env
- **Activate (Windows):** env\Scripts\activate
- **Deactivate:** deactivate

5. Writing & Executing Python Files

Python files end with .py extension.

- **Example:** print('Hello, World!')
- **Execute it using:** python hello.py

6. Python Syntax Rules

Python uses indentation (usually 4 spaces) instead of braces to define code blocks.

Variable naming rules:

- Must start with a letter or underscore
- Cannot start with a number
- Case-sensitive

7. Comments in Python

- **Single-line comment:** # This is a comment
- **Multi-line comment:** '''This is a multi-line comment'''

8. Using input () and print()

input() is used to take user input.

Example:

```
name = input('Enter your name: ')
print('Hello', name)
```

print() displays output and supports formatted strings using f-strings:

```
print(f'My name is {name}')
```

9. Basic Error Types & Fixing Them

Error Type	Cause	Example	Fix
SyntaxError	Invalid syntax	if True print('Hi')	Add ':' after condition
IndentationError	Improper indentation	print('Hi') misaligned	Use consistent indentation
NameError	Variable not defined	print(x)	Define variable before use
TypeError	Incompatible data types	'5' + 5	Convert types using int() or str()
ValueError	Invalid value	int('abc')	Ensure valid input values
ZeroDivisionError	Division by zero	10/0	Check denominator before division

10. How Python Programs Run (Execution Flow)

1. You write code in a .py file.
2. The Python interpreter reads it line by line.
3. It converts the code into bytecode (.pyc).
4. The Python Virtual Machine (PVM) executes the bytecode.
5. Output is displayed on the console.