🐍 Zero to Hero Python Programming Syllabus

🔰 Module 1: Introduction to Python & Programming Fundamentals

Goal: Understand what Python is, set up the environment, and learn programming basics.

Topics:

- Introduction to Python:
  - History and philosophy
  - Why Python? Use cases
  - Installing Python, IDEs (VS Code, PyCharm, Jupyter)
  - Writing and running your first Python script
- Python Interpreter & Syntax
- Variables and Data Types
  - Numbers, Strings, Booleans
- Basic I/O
- Comments and Docstrings
- Basic Operators
- Type Conversion & Casting
- String Manipulation

🧠 Module 2: Control Flow

Goal: Learn how to write logic using conditional statements and loops.

Topics:

- Boolean Logic
- if, elif, else Statements
- Comparison Operators
- Logical Operators (and, or, not)
- Loops:
  - while Loop
  - for Loop
  - Loop control (break, continue, pass)
  - Looping through Strings, Lists, Ranges, Dictionaries

- List Comprehensions

🗃️ Module 3: Data Structures in Python

Goal: Master Python's built-in data structures.

Topics:

- Lists:
    - Creation, Indexing, Slicing, Nesting
    - Methods (append(), pop(), sort(), etc.)
- Tuples:
    - Immutability
    - Tuple unpacking
- Sets:
    - Unique elements, operations (union, intersection)
- Dictionaries:
    - Key-value pairs, nesting
    - Methods (get(), keys(), values())
- Data structure conversions

🧰 Module 4: Functions and Functional Programming

Goal: Learn how to define and use functions.

Topics:

- Defining Functions
- Arguments & Parameters (positional, keyword, default)
- Return Statement
- Variable Scope (local vs global)
- *args and **kwargs
- Lambda Functions
- Built-in functions (map(), filter(), zip(), enumerate())
- Recursion

🧱 Module 5: Object-Oriented Programming (OOP)

Goal: Understand and implement OOP concepts in Python.

Topics:

- Classes and Objects
- __init__() Constructor
- Instance vs Class Variables
- Instance, Class, and Static Methods
- Encapsulation
- Inheritance
- Polymorphism
- Magic/Dunder Methods (__str__, __repr__, etc.)
- Composition vs Inheritance
- Abstract Classes (via abc module)
- Interfaces

🐞 Module 6: Error Handling and Debugging

Goal: Handle runtime errors and debug code efficiently.

Topics:

- Types of Errors: Syntax, Runtime, Logical
- Try-Except Block
- Multiple Exception Handling
- finally, else Clauses
- Custom Exceptions
- Using assert
- Debugging Tools (pdb, IDE tools)

📦 Module 7: Modules, Packages, and Virtual Environments

Goal: Organize and reuse code effectively.

Topics:

- Importing Modules (import, from ... import)
- Creating Custom Modules
- Python Standard Library Overview

- pip and PyPI
- Virtual Environments (venv, pipenv, poetry)
- Package structure and __init__.py

📁 Module 8: File Handling and I/O Operations

Goal: Learn to read/write files and manage I/O operations.

Topics:

- Reading and Writing Text Files (open, read, write)
- Context Managers (with)
- Working with CSV and JSON
- OS Module for File Paths and Directories
- Pickling and Serialization

🌐 Module 9: Python and the Internet

Goal: Learn to interact with the web and APIs.

Topics:

- HTTP Requests with requests library
- REST APIs and JSON
- Web scraping with BeautifulSoup and lxml
- Automating Browsers with selenium

🧪 Module 10: Testing and Best Practices

Goal: Write clean, testable, and maintainable code.

Topics:

- Introduction to Testing
- unittest, pytest
- Writing Test Cases
- Mocking and Fixtures
- Code Quality Tools: flake8, black, pylint
- Documentation with docstrings
- Version Control with Git (optional intro)

📊 Module 11: Intermediate to Advanced Python

Goal: Master advanced features and performance tuning.

Topics:

- Iterators and Generators
- Decorators
- Context Managers (with, __enter__, __exit__)
- *args, **kwargs Deep Dive
- Dynamic Execution (eval, exec)
- Multithreading vs Multiprocessing
- AsyncIO and Concurrency
- Memory Management and Garbage Collection
- Performance Optimization Techniques

💻 Module 12: Real-World Applications & Projects

Goal: Apply skills to real projects. Choose tracks depending on goals.

📊 Data Science Track (Optional)

- NumPy, Pandas
- Data Cleaning & Analysis
- Data Visualization (Matplotlib, Seaborn)
- Basic Statistics
- Introduction to Machine Learning (Scikit-learn)

🌐 Web Development Track (Optional)

- Flask / Django Web Frameworks
- HTML Templates with Jinja2
- REST APIs with Flask
- WebSockets (Intro)
- User Authentication & Database Integration (SQLite, PostgreSQL)

⚙️ Automation & Scripting Track (Optional)

- Task Automation with schedule and os

- Automating Excel (with openpyxl, pandas)
- Web Automation (Selenium, PyAutoGUI)

🕹️ **Game Development Track (Optional)**

- Intro to pygame
- Event Loop, Game Logic, Sound, and Graphics

🧠 **Capstone Projects (End of Course)**

- Build a Portfolio Website in Flask
- Build a Task Manager Web App
- Data Dashboard with Pandas and Plotly
- Automation Script Suite (e.g. bulk renamer, downloader)
- Game: Python Snake / Tetris
- REST API with Authentication

📝 **Certification & Assessment (Optional)**

- Quizzes per module
- Final assessment/project evaluation
- Certificate of completion