🐍 Complete Python Programming & Automation Course

Instructor: Sanjeet Prasad
Email: sanjeet8.23@gmail.com
Duration: 28 Days (1.5–2 hours/day)
Format: Instructor-led, hands-on, project-based learning
Audience: Students, developers, embedded engineers, automation professionals, and aspiring Python learners

📘 Python Core Modules (Days 1–26)

Day 1: Embedded Systems & Python Setup

- Overview of embedded systems: hardware/software components
- Role of Python in automation and analytics
- Setup: Python 3.x, pip, VS Code + essential extensions
- Writing and running first Python program
- Python execution lifecycle

Day 2: Python Syntax, Variables & Comments

- Identifiers, keywords, indentation rules
- Commenting styles and docstrings
- Dynamic variable assignment
- VS Code debugger: breakpoints and watches

Day 3: Data Types & Type Conversion

- Built-in types: int, float, str, bool, list, tuple, dict, set
- Type casting and validation
- Mutable vs immutable objects
- Using type(), isinstance()

Day 4: Operators & Expressions

- Arithmetic, logical, relational, and bitwise operators
- Compound assignments and chaining
- Operator precedence and associativity

Day 5: Decision-Making in Python

- Control flow using if, elif, else
- Pattern matching using match-case (Python 3.10+)
- Nested logic and evaluation scenarios

Day 6: Looping Constructs

- Using for and while loops effectively
- Loop modifiers: break, continue, else
- Interactive loop-based exercises

Day 7: Functions & Modularization

- Defining reusable functions
- Parameters: positional, keyword, default
- Return values, scope rules
- Recursive functions

Day 8: Lambda & Built-in Functions

- Anonymous functions with lambda
- Functional tools: map(), filter(), zip()
- Built-in functions: len(), type(), range()

Day 9: Modules & Packages

- Creating and importing Python modules
- Organizing packages using __init__.py
- Script execution logic with __name__ == "__main__"

Day 10: File Handling Essentials

- Reading/writing .txt, .csv, .json
- Context managers (with open)
- Line-by-line reading, writing to structured formats

Day 11: OS-Level Automation

- Using os, sys, glob for directory operations
- File filtering and folder scanning
- Automation scripts for file management

Day 12: Exception Handling & Assertions

- Structured error handling: try, except, finally, raise
- Assertion-based input validation
- Writing custom exceptions

Day 13: Unit Testing with Pytest Framework

- Creating modular test cases
- assert statements and test coverage
- Parametrized tests with decorators

Day 14: Regular Expressions

- Pattern matching using re.search(), re.match(), re.findall()
- Log parsing and data validation with regex

Day 15: Object-Oriented Programming (OOP)

- Classes, attributes, and methods
- Constructors with __init__()
- Encapsulation and access control

Day 16: Advanced OOP Concepts

- Inheritance and method overriding
- super() and multi-level hierarchy
- Dunder methods: __str__, __repr__, __eq__

Day 17: Working with Lists & Dictionaries

- List operations: slicing, indexing, comprehensions
- Dictionary methods: keys, values, iteration
- Nested data structures and manipulation

Day 18: Working with Tuples, Sets & Collections

- Tuple packing/unpacking
- Set operations and uniqueness

Day 19: Using JSON & CSV in Real Applications

- Parsing and serializing JSON with json module
- Structured data reading/writing with csv
- Validating and transforming tabular inputs

Day 20: DateTime & Math Modules

- Working with datetime, timedelta, and timestamps
- Mathematical operations with math, random
- Building custom time-based logic

Day 21: NumPy Basics

- Creating and manipulating arrays
- Reshaping, slicing, and broadcasting
- Statistical functions and performance advantages

Day 22: Pandas for Data Analysis

- DataFrames and Series creation
- Import/export with CSV and Excel
- Grouping, filtering, and aggregating data

Day 23: Git & GitHub for Version Control

- Repository setup: init, add, commit, push
- Branching, merging, and resolving conflicts
- Collaborating on GitHub and hosting projects

Day 24: Python Packaging & Deployment

- Structuring Python projects
- Writing setup.py, requirements.txt
- Packaging CLI tools and installing as modules

Day 25: AI-Assisted Python Development

- Writing prompts for ChatGPT and Copilot
- Using AI to scaffold code and refactor functions
- Best practices for validating AI-generated logic

🗁 Real-World Python Projects (Days 26–28) 3 Days

1. 5G NR KPI Calculator

2. NR Throughput Estimator

3. LTE/NR Log Analyzer

4. Wireless AP Simulation (Function-Based)

🔧 Tools & Libraries Used

- Python 3.11+
- Visual Studio Code
- Pytest for testing
- NumPy and Pandas
- ChatGPT and GitHub Copilot

📺 Sample Demo Session

Watch the instructor-led Python session for a live preview:
🔗 Python Demo – Instructor-led Session

📞 Instructor Contact

For customization, group sessions, or project guidance:
Instructor: Sanjeet Prasad
Email: sanjeet8.23@gmail.com

GitHub (Projects & Source Code)
🔗 github.com/sanjeet9019

LinkedIn (Professional Profile)
🔗 linkedin.com/in/sanjeet9019