Python Training Course Curriculum

Prerequisites

- Basic understanding of programming concepts
- Familiarity with any programming language
- Basic knowledge of databases
- No prior Python experience required

Lab Setup Requirements

- Python 3.9 or later
- Virtual environment (venv or conda)
- IDE (VS Code, PyCharm, or any preferred editor)
- Database (PostgreSQL or MySQL)
- Postman or similar API testing tool
- Git for version control

Course Duration: 3 Days

Day 1: Python Fundamentals and DevOps Automation

- Python Basics and Best Practices
  - Python installation and environment setup
  - PEP 8 style guide
  - Code organization and structure
  - Documentation and comments
- Essential Python Features
  - Data types and structures
  - Control flow and functions
  - Error handling
  - File I/O operations
- DevOps Automation with Python
  - Working with system modules (`os`, `sys`, `subprocess`)
  - File and directory operations
  - Process management
  - System monitoring
- Network Operations
  - `requests` library for HTTP operations
  - Working with APIs
  - Error handling in network operations
- Lab Exercise: Creating automation scripts

Day 2: Concurrency, Database Operations and Testing

- Concurrency in Python

    - Understanding threading
    - Working with `threading` module
    - Introduction to `asyncio`
    - Async/await syntax
    - Practical use cases
    - Choosing between threading and asyncio
- Python database fundamentals
    - SQL basics
    - CRUD operations
    - Joins and relationships
    - Transactions and ACID properties
- Object-Relational Mapping (ORM)
    - Introduction to SQLAlchemy
    - Model definitions
    - Query operations
    - Relationships and joins
    - Session management
- Testing in Python
    - Introduction to `unittest`
    - Writing test cases
    - Test fixtures
    - Mocking and patching
    - Test-driven development (TDD)
- Lab Exercise: Database operations and testing

Day 3: REST API Development

- REST API Development with Flask
    - Flask framework basics
    - Route handling
    - Request/response cycle
    - Error handling
    - Middleware and extensions
- API Documentation
    - Introduction to Swagger/OpenAPI
    - API documentation best practices
    - Interactive documentation
    - Versioning strategies
- Final Project
    - Building a complete REST API
    - Database integration
    - Testing implementation
    - Documentation
    - Q&A session