

Python Programming

Course Description

Unlock the full potential of Python in this all-in-one course, designed to transform you from a beginner to an expert Python developer. Covering everything from basic syntax to advanced topics like web development, data science, and automation, this course is perfect for Python 3.x and compatible with tools used in 2025. The program bundles four levels of learning: Beginner, Intermediate, Advanced, and Expert with Projects.

What You'll Learn

- Master Python syntax and programming fundamentals
- Build applications for data analysis, web development, and automation
- Work with databases, APIs, and modern frameworks like FastAPI
- Apply Python to real-world projects, including web apps and data pipelines
- Deploy Python applications and automate workflows

Course Requirements

- No prior programming knowledge required
- A computer with Python 3.x and a code editor (e.g., VS Code) installed

Who This Course Is For

- Beginners starting their programming journey
- Intermediate developers looking to deepen Python skills
- Professionals aiming to apply Python in data science, web development, or automation

Syllabus (35 Sessions)

1. Python Fundamentals

- Installing Python and setting up VS Code
- Introduction to Python: Syntax, variables, and data types
- Running Python scripts (REPL, scripts, Jupyter notebooks)
- Common Python shortcuts and debugging

2. Basic Programming Concepts

- Working with numbers, strings, and booleans
- Operators: Arithmetic, comparison, logical
- Input/output with input() and print()
- Creating simple scripts

3. Control Flow

- Conditional statements: if, elif, else

- Loops: for and while
- Break, continue, and pass statements
- Handling user input with conditionals
- 4. **Functions in Python**
 - Defining and calling functions
 - Parameters, arguments, and return values
 - Default and keyword arguments
 - Scope and variable lifetime
- 5. **Data Structures: Lists and Tuples**
 - Creating and manipulating lists
 - List methods: append, remove, sort
 - Working with tuples
 - List comprehensions for concise code
- 6. **Data Structures: Dictionaries and Sets**
 - Creating and using dictionaries
 - Dictionary methods: get, update, keys
 - Working with sets and set operations
 - Practical applications of dictionaries
- 7. **File Handling**
 - Reading and writing text files
 - Working with CSV files
 - Handling file paths with os and pathlib
 - Error handling with try-except
- 8. **Modules and Packages**
 - Importing standard library modules
 - Creating custom modules
 - Using pip to install packages
 - Exploring popular libraries (e.g., math, random)
- 9. **Practice Session**
 - Build a simple calculator script
 - Congratulations! You've built a Python foundation.
- 10. **Object-Oriented Programming (OOP)**
 - Classes and objects
 - Attributes and methods
 - Inheritance and polymorphism
 - Encapsulation and properties
- 11. **Advanced Data Structures**
 - Nested lists and dictionaries
 - Working with collections: defaultdict, Counter
 - Stacks and queues with deque
 - Practical OOP examples
- 12. **Error Handling and Debugging**
 - Advanced try-except blocks
 - Raising custom exceptions
 - Debugging with pdb and VS Code

- Logging with the logging module
- 13. Working with APIs**
 - Introduction to REST APIs
 - Using requests to fetch data
 - Parsing JSON responses
 - Building a simple API client
- 14. Introduction to Data Analysis**
 - Setting up pandas for data manipulation
 - Working with DataFrames
 - Basic data cleaning and filtering
 - Visualizing data with matplotlib
- 15. Database Basics**
 - Introduction to SQLite
 - CRUD operations with sqlite3
 - Connecting Python to PostgreSQL
 - Handling database errors
- 16. Practice Session**
 - Build a data analysis script with pandas
 - Congratulations! You're an intermediate Python developer.
- 17. Advanced Functions**
 - Lambda functions and anonymous functions
 - Decorators for function enhancement
 - Generators and yield
 - Closures and function factories
- 18. Web Development with FastAPI**
 - Setting up a FastAPI project
 - Creating RESTful endpoints
 - Handling query parameters and request bodies
 - Testing APIs with Postman
- 19. Working with Databases in Web Apps**
 - Connecting FastAPI to PostgreSQL with SQLAlchemy
 - Defining database models
 - Performing CRUD operations
 - Handling relationships in databases
- 20. Authentication in Web Apps**
 - Implementing JWT authentication
 - User registration and login
 - Securing API endpoints
 - Password hashing with passlib
- 21. Data Science with Python**
 - Advanced pandas techniques
 - Data visualization with seaborn
 - Introduction to numpy for numerical computing
 - Basic machine learning with scikit-learn
- 22. Automation with Python**

- Automating file operations with shutil
- Scheduling tasks with schedule
- Web scraping with beautifulsoup
- Automating browser tasks with selenium

23. **Type Hints and Static Analysis**

- Introduction to Python type hints
- Using mypy for static type checking
- Type-safe code with typing module
- Refactoring code for type safety

24. **Practice Session**

- Build a web scraper with beautifulsoup
- Congratulations! You've mastered Python programming.

25. **Project 1: Task Manager API**

- Build a REST API with FastAPI
- Implement CRUD for tasks
- Add JWT authentication
- Test with Postman

26. **Project 2: Data Analysis Dashboard**

- Create a dashboard with pandas and plotly
- Analyze a sample dataset
- Visualize trends and insights
- Export results to CSV

27. **Project 3: Web Scraper**

- Build a scraper with beautifulsoup
- Extract data from a website
- Store results in SQLite
- Handle pagination and errors

28. **Project 4: Automation Script**

- Automate file organization with shutil
- Schedule tasks with schedule
- Send email notifications with smtplib
- Test automation workflow

29. **Project 5: Blog API**

- Build a blog API with FastAPI
- Manage posts and users
- Add authentication and authorization
- Deploy to Heroku

30. **Project 6: Machine Learning Model**

- Build a simple ML model with scikit-learn
- Train on a sample dataset
- Evaluate model performance
- Save and load models

31. **Testing in Python**

- Unit testing with unittest
- Mocking with unittest.mock

- Integration testing for APIs
- Testing with pytest

32. Deploying Python Applications

- Deploying FastAPI to Heroku
- Using Docker for containerization
- Setting up CI/CD with GitHub Actions
- Configuring environment variables

33. Project 7: E-Commerce API

- Build an e-commerce API with FastAPI
- Integrate Stripe for payments
- Manage products and orders
- Deploy with Docker

34. Advanced Topics

- Multithreading and multiprocessing
- Asynchronous programming with asyncio
- Caching with redis
- Performance optimization techniques

35. Final Exam and Wrap-Up

- Comprehensive exam on Python concepts
- Review of key skills and best practices
- Guidance on Python career paths
- Certificate of completion



OMEGA'S
ACADEMY