

Advanced Python

Length: 5 Days

Summary: How do the world's best engineering teams use Python? What language features do they use, and how? How do you do test-driven development, leverage Python's object model, build concurrent servers, and more? This course for experienced developers helps you take your expertise in Python to a whole new level. This course is taught using Python 3, with instruction throughout on how to apply the concepts to Python.

Objectives: By the end of this course, students will be able to:

- Understand the most powerful patterns and tools modern Python has to offer,
- Know how to leverage them to create reliable, maintainable applications - either individually, or as part of a development team.

Audience This course is designed for programmers looking to take their existing Python skills to a new level

Topics:

- Test-driven Python development
- Writing scalable Python code
- Python's logging module
- Python's concurrency model
- Context managers
- All about decorators
- Object-oriented programming with Python
- REST APIs
- Mastering list comprehensions
- Functional Python programming
- Practical agile software development in Python
- Advanced data types and collections

COURSE CONTENT

I. Test-driven Python development

II. Writing scalable Python code

- iterators and Python's iterator protocol
- Generators
- Views
- Leveraging built-in types for improved performance

III. Python's logging module

- Getting the most out of Python's amazing and rich logging module

IV. Python's concurrency model

- Understanding the important distinction between OS threads and Python threads, and the implications for concurrent Python software
- Scaling CPU-bound tasks with multiprocessing
- Asynchronous programming with asyncio
- Multiple threads in Python: when to do it, when to avoid it, and best practices

V. Context managers

VI. All about decorators

- Review of basic decorator patterns
- Creating decorators that take arguments
- Powerfully extensible class-based decorators
- Creating decorators for classes (which is a completely different thing)

VII. Object-oriented programming with Python

- The Python object model
- Creating new syntax and expressive code with "magic methods"
- Patterns of abstraction and code organization
- Metaclasses: what they do, when to use them, and when to avoid them

VIII. REST APIs

- RESTful API integration
- Building REST servers in Python

IX. Mastering list comprehensions

X. Functional Python programming

XI. Practical agile software development in Python

- Virtual environments
- Package management
- Version control considerations
- Maintainability and readability
- Best Practices for reliability and robustness

XII. Advanced data types and collections

XIII. Additional topics depending on goals and desires of participants