



Course Content

Course Description:

This 4-day course picks up where Introduction to Python 3.x leaves off, covering some topics in more detail, and adding many new ones, with a focus on enterprise development. This is a hands-on programming class. All concepts are reinforced by informal practice during the lecture followed by lab exercises. Many labs build on earlier labs, which helps students retain the earlier material.

Target Student:

Advanced users, system administrators and web site administrators who want to use Python to support their server installations, as well as anyone else who wants to automate or simplify common tasks with the use of Python scripts.

Prerequisites:

All students should be able to write simple Python scripts, using basic data types, program structures, and the standard Python library.

Topics:

Python Refresher

- Variables
- Basic Python Data Types
- Sequences
- Mapping Types
- Program Structure
- Files And Console I/O
- Conditionals
- Loops
- Builtins

OS Services

- The OS Module
- Paths, Directories, and Filenames
- Environment Variables
- Launching External Programs
- Walking Directory Trees
- The Datetime Module
- The Calendar Module

Pythonic Programming

- The Zen of Python
- Common Python Idioms
- Unpacking Function Arguments
- Lambda Functions
- List Comprehensions
- Iterables
- Writing Generators
- String Tricks
- String Formatting

Modules

- Using Import
- Module Search Path
- Namespaces
- Executing Modules as Scripts
- Packages
- Configuring Import With `__init__.py`



Intermediate Python 3.x

Course ID #: 1411-901-3.x-W

Hours: 28

- Name Resolution (AKA Scope)
- Python Style

Classes

- Defning Classes
- Instance Objects
- Instance Attributes
- Instance Methods
- `__Init__`
- Properties
- Class Data
- Class Methods
- Inheritance
- Multiple Inheritance
- Using Super ()
- Special Methods
- Class-Private Variables
- Static Methods

Metaprogramming

- Globals() and Locals()
- Working with Attributes
- The Inspect Module
- Decorator Functions
- Decorator Classes
- Decorator Parameters
- Creating Classes At Runtime
- Monkey Patching

Developer Tools

- Program Development
- Comments
- Pylint
- Customizing Pylint
- Using Pyreverse
- The unittest Module
- Fixtures
- Skipping Tests
- Making a Suite of Tests
- Automated Test Discovery
- Using Nose

- The Python Debugger
- Starting Debug Mode
- Stepping Through a Program
- Setting Breakpoints
- Profiling
- Benchmarking

Database access

- The DB API
- Available Interfaces
- Connecting to a server
- `connect()` examples
- Creating a cursor
- Executing a statement
- Parameterized statements
- Dictionary cursors
- Metadata
- Transactions
- Object-relational mappers

PyQt4

- Event Driven Applications
- Gui Application Flow Chart
- External Anatomy of a PyQt4 Application
- Internal Anatomy of a PyQt4 Application
- Using Designer
- Anatomy of a Designer-Based Application
- Naming Conventions
- Common Widgets
- Layouts
- Selectable Buttons
- Actions and Events
- Menu Bar
- Status Bar
- Using Predefined Dialogs
- Creating Custom Dialogs
- Tabs
- Niceties
- Working with Images
- Complete Example



Intermediate Python 3.x

Course ID #: 1411-901-3.x-W

Hours: 28

Network Programming

- Sockets
- Socket Options
- Client Concepts
- Server Concepts
- Application Protocols
- Forking Servers
- Grabbing Html from The Web
- Consuming Web Services
- Web Data the Easier Way
- Sending Email
- Binary Data
- The Struct Module

Multiprogramming

- What Are Threads?
- The Python Thread Manager
- The Threading Module
- Threads for the Impatient
- Creating a Thread Class
- Variables Sharing
- Using Queues
- Debugging Threaded Programs
- The Multiprocessing Module
- Alternatives to Multiprogramming

System Administration and Scripting

- The Subprocess Module
- Subprocess Convenience Functions
- Using the Sh Module
- Permissions
- Saving Information
- Creating a Useful Command Line Script
- Creating Filters
- Parsing the Command Line
- Simple Logging
- Logging Levels
- Formatting Log Entries
- Logging to Other Destinations

XML and JSON

- About Xml
- Normal Approaches to Xml
- Which Module to Use?
- Getting Started With ElementTree
- How ElementTree Works
- Creating a New Xml Document
- Parsing an Xml Document
- Navigating the Xml Document
- Using XPath
- Advanced XPath
- About JSON
- Reading JSON
- Writing JSON

Extending Python with C

- Why Extend Python?
- Ways to Extend Python With C
- Hand-Coded C
- Overview
- The C Program
- Methods
- The Method Table
- The Init Function
- Handling Errors
- Custom Exception Objects
- Putting It All Together
- Using SWIG
- The Interface File
- Generating the Wrappers
- Building and Installing The Extension
- Ctypes
- For More Information