www.logicalimagination.com
800.657.1494

Python Intermediate

Course #: PY-102 **Duration:** 4 days

Prerequisites

Python Introduction or the ability to write simple Python scripts, using basic data types, program structures, and the standard Python library.

Details

This course picks up where Python Introduction leaves off, covering some topics in more detail, and adding many new ones, focusing 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. The length of the refresher section (Section 1) will vary based on each class's level of Python preparation. It may range from one hour to half a day. Because of this, the last two chapters (Advanced Data Handling and Type Hinting) will be covered as time permits.

Software Needed

Python 3 or higher.

Outline

Python Intermediate

- **Python Refresher**
  - Objectives
  - Variables
  - Basic Python Data Types
  - Sequence Types
  - Mapping Types
  - Program Structure
  - Files and Console I/O
  - Conditionals
  - Loops
  - Built-ins
  - Functions
  - Modules
  - Packages
- **OS Services**
  - Objectives
  - The OS Module
  - Paths, Directories, and File Names
  - Walking Directory Trees

- Environment Variables
- Launching External Programs

- **Dates and Times**
  - Objectives
  - Python Modules for Dates and Times
  - Ways to Store Dates and Times
  - Basic Dates and Times
  - Formatting Dates and Times
  - Parsing Date/Time Strings
  - Parsing Dates the Easier Way
  - Converting Dates and Times
  - Time Zones
  - Generating Calendars

- **Binary Data**
  - Objectives
  - "Binary" (raw, or non-delimited) Data
  - Binary vs. Text Data
  - Using Struct
  - Bitwise Operators

- **Pythonic Programming**
  - The Zen of Python
  - Tuples
  - Iterable Unpacking
  - Unpacking Functions Arguments
  - The sorted() Function
  - Custom Sort Keys
  - Lambda Functions
  - List Comprehensions
  - Dictionary Comprehensions
  - Set Comprehensions
  - Iterables
  - Generator Expressions
  - Generator Functions
  - String Formatting
  - f-strings

- **Functions, Modules, and Packages**
  - Functions
  - Function Parameters
  - Default Parameters
  - Python Function Parameter Behavior (from PEP 3102)
  - Name Resolution (AKA Scope)
  - The Global Statement
  - Modules
  - Using Import
  - How Import * Can Be Dangerous
  - Module Search Path
  - Executing Modules as Scripts
  - Namespaces
  - Packages
  - Configuring import with __init__.py
  - Documenting Modules and Packages
  - Python Style

- **Intermediate Classes**
  - What is a Class?
  - Defining Classes
  - Object Instances
  - Instance Attributes
  - Instance Methods
  - Constructors
  - Getters and Setters
  - Properties
  - Class Data

- Class Methods
- Inheritance
- Using super()
- Multiple Inheritance
- Abstract Base Classes
- Special Methods
- Static Methods

- **Metaprogramming**
  - Objectives
  - Metaprogramming
  - globals() and locals()
  - The Inspect Module
  - Working with Attributes
  - Adding Instance Methods
  - Decorators
  - Applying Decorators
  - Trivial Decorator
  - Decorator Functions
  - Decorator Classes
  - Decorator Parameters
  - Creating Classes at Runtime
  - Monkey Patching
  - Callable Classes
  - Do you Need a Metaclass?
  - About MetaClasses
  - Mechanics of a Metaclass
  - Singleton with a Metaclass

- **Developer Tools**
  - Objectives
  - Program Development
  - Comments
  - pylint
  - Customizing pylint
  - Using pyreverse
  - The Python Debugger
  - Starting Debug Mode
  - Stepping through a Program
  - Setting Breakpoints
  - Profiling
  - Benchmarking

- **Unit Tests with pytest**
  - What is a Unit Test?
  - The pytest Module
  - Creating Tests
  - Running Tests (Basics)
  - Special Assertions
  - Fixtures
  - User-Defined Fixtures
  - Builtin Fixtures
  - Configuring Fixtures
  - Parameterizing Tests
  - Marking Tests
  - Running Tests (Advanced)
  - Skipping and Failing
  - Mocking Data
  - pymock Objects
  - Pytest and Unittest

- **Database Access**
  - The DB API
  - Connecting to a Server
  - Creating a Cursor
  - Executing a Statement
  - Fetching Data

    - SQL Injection
    - Parameterized Statements
    - Dictionary Cursors
    - Metadata
    - Transactions
    - Object-relational mappers
    - NoSQL
- **PyQt**
    - What is PyQt?
    - Event-Driven Applications
    - External Anatomy of a PyQt Application
    - Internal Anatomy of a PyQt Application
    - Using Designer
    - Designer-based Application Workflow
    - Naming Conventions
    - Common Widgets
    - Layouts
    - Selectable Buttons
    - Actions and Events
    - Signal/Slot Editor
    - Editing Modes
    - Menu Bar
    - Status Bar
    - Forms and Validation
    - Using Predefined Dialogs
    - Tabs
    - Niceties
    - Working with Images
    - Complete Example
- **Network Programming**
    - Grabbing a Web Page
    - Consuming Web Services
    - HTTP the Easy Way
    - Sending Email
    - Email Attachments
    - Remote Access
    - Copying Files with Paramiko
- **Multiprogramming**
    - 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
    - Using Pools
    - Alternatives to Multiprogrammiong
- **Effective Scripts**
    - Using glob
    - Using shlex.split()
    - The Subprocess Module
    - Subprocess Convenience Functions
    - Capturing Stdout and Stderr
    - Permissions
    - Using shutil
    - Creating a Useful Command Line Script
    - Creating Filters
    - Parsing the Command Line
    - Simple Logging
    - Formatting Log Entries

- Logging Exception Information
- Logging to Other Destinations

- **Serializing Data**
  - About XML
  - Normal Approaches to XML
  - Which Module to Use?
  - Getting Started with ElementTree
  - How ElementTree Works
  - Elements
  - Creating a New XML Document
  - Parsing an XML Document
  - Navigating the XML Document
  - Using XPath
  - About JSON
  - Reading JSON
  - Writing JSON
  - Customizing JSON
  - Reading and Writing YAML
  - Reading CSV Data
  - Nonstandard CSV
  - Using csv.DictReader
  - Writing CSV Data
  - Pickle

- **Advanced Data Handling**
  - Deep vs. Shallow Copying
  - Default Dictionary Values
  - Counting with Counter
  - Named Tuples
  - Printing Data Structures
  - Zipped Archives
  - Tar Archives
  - Serializing Data

- **Type Hinting**
  - Type Hinting
  - Static Analysis Tools
  - Runtime Analysis Tools
  - typing Module
  - Input Types
  - Variance
  - Union and Optional
  - multimethod and functools.singledispatch
  - Stub Type Hinting