Python Programming 2

Duration : 3 Days

Overview

The Python Programming 2 course comprises sessions dealing with advanced object orientation, iterators and generators, comprehensions, decorators, multithreading, functional programming, web services, and unit testing.

The delegate will learn how to exploit advanced features of the Python language to build complex and efficient applications.

Exercises and examples are used throughout the course to give practical hands-on experience with the techniques covered.

The delegate will learn and acquire skills as follows:

- Encapsulating classes
- Exploiting polymorphism using inheritance and mixins
- Associating objects via composition and aggregation
- Working with static members
- Using iterators as an alternative to for
- Constructing custom iterators
- Constructing functions that yield generators
- Manipulating lists, sets, and dictionaries using comprehension
- Exploiting aspect oriented programming using decorators
- Writing multithreaded code
- Sharing data between threads
- Processing collections using lambdas
- Building RESTful clients
- Building RESTful APIs
- Testing units of code

Target Audience

The Python Programming 2 course is designed for existing Python developers who have a good grounding in the basics and want to exploit some of the advanced features of the language.

For the delegate for whom Python is their first programming language, we recommend taking the Python Programming 1 course first, then taking some time to practice the skills gained, before returning to take the Python Programming 2 course.

Prerequisites

Delegates should be able to build Python applications that exploit all fundamental elements of the language including variables and expressions, conditions and loops, functions, objects, and lists.

This knowledge can be gained by attendance on the pre-requisite Python Programming 1 course.

Objectives

This course aims to provide the delegate with the knowledge to be able to interpret, write, and troubleshoot complex Python applications exploiting inheritance and polymorphism, mixins, composition and aggregation, iterators, generators, decorators, comprehension, concurrency, functional programming, and RESTful web services.

Course Content

DAY 1

Session 1: ADVANCED OBJECT ORIENTATION

The self Keyword
Constructors and Destructors
Encapsulation
Inheritance
Polymorphism
Abstract Classes
Multiple Inheritance and Mixins
Composition and Aggregation
Static Members

Session 2: ITERATORS & GENERATORS

Iterables
Iterators
Custom Iterators
Generators
Yield vs. Return

Session 3: COMPREHENSIONS

List Comprehension
Set Comprehension
The zip Function
Dictionary Comprehension

DAY 2

Session 4: FUNCTIONAL PROGRAMMING

Functional Programming
Lambdas
Immutability
Mapping
Filtering
Reducing

Session 5: DECORATORS

Decorators
Decorator Functions
Decorator Annotations
Decorator Use Cases
Labs

Session 6: MULTITHREADING

Threads
Multithreading
Thread Construction
Thread Execution
Thread Sleep
Joins
Data Sharing
Synchronisation
Multithreading vs. Multiprocessing

DAY 3

Session 7: WEB SERVICES

RESTful Web Services
JSON Data
CRUD and HTTP
RESTful Clients
RESTful APIs

Session 8: UNIT TESTING

Unit Testing Terminology
Test Classes
Test Fixtures
Test Cases
Assertions
Test Runners