Python Jumpstart Workshop/Training Course

Overview | Objectives | Suggested Audience | Duration | Prerequisites | Syllabus

Request An Onsite Class

Overview

Python Jumpstart Fundamentals is a 3-day training course in the Python language and its many applications. The course covers the language itself, explains object-oriented as well as functional programming techniques, error handling, packaging, system and network programming, many of the Python extensions (libraries), as well as best practices.

Objectives

This is a fast-paced lab course, designed to bring seasoned programmers up-to-speed in Python, as quickly as possible. At the end of Python Jumpstart training course, the participants will experience:

- Pythonic thinking.
- Python's input and output details: stdio and file io.
- Python's interesting and extra-useful flow-of-control devices.
- Python's simple and robust error handling.
- Python's flexible function protocols.
- Python's memory model.
- Python's straight-forward object-oriented features.
- Python's built-in data types: Using them and inheriting from them in classes you design.
- Python's list comprehensions, decorators, iterators, generators, context managers.
- Python's scheme for creating and using libraries and packages.
- Python's handy libraries for many developer/administrator tasks: shutil, tempfile, subprocess, glob, profile, shelve, os, sys, optparse, unittest.
- Python's architecture, which allows you to get working very quickly with any Python library (there are thousands).

Suggested Audience - This course is designed for developers, system administrators, and QA engineers, who wish to develop, automate, and test applications and systems using one of the most powerful programming languages available today.

Duration - 3 Days

Prerequisites - Attendees should have prior programming experience and be familiar with basic concepts such as variables/scopes, flow-control, and functions. Prior exposure to object-oriented programming concepts is not required, but definitely beneficial.

Syllabus

Questions? Request a call back

1. Introduction to Python

- Python – The Universal Language

2. Getting Started

- Installing Python
- Python - "Hello World"
- Using the Interpreter
- iPython - a better Python interpreter

3. Language Basics

- Types
    - Dynamic v/s Static Typing
    - Strong v/s Weak Typing
- Numbers
- Strings
- Unicode
- Complex types
- Operators
    - Operator Overloading
- Variables
- Scoping and Expressions
- Use of tabs and whitespaces as indent
- Conditionals
    - for...else

4. Functions

- The general syntax
- Default values for arguments
- Returning and receiving multiple values
- Variable number of arguments – args, kwargs
- Scope revisited

5. Collections

- Primitive v/s Composite Types
- Lists
- Tuples
- Maps (or Dictionaries)
- Sets
- Enums
- Looping Techniques

6. Modularisation of code

- Global and Local namespace
- Introduction to modules
- Using modules
- Creating your own modules
- Working with a third-party library

7. Regular Expressions

- Matching v/s Searching
- Regular Expression Objects
- Match Objects
- Examples

8. Files and Directories

- Reading Files
- Writing Files
- Handling I/O Errors
- Higher level file operations
- File and Directory comparisons

9. Exception Handling

- Exception handling basics
- try...except
- Examples

10. Socket Programming

- Introduction to networking concepts
- Creating a socket
- Using a socket
- Disconnecting
- Non-blocking sockets

11. Object Oriented Programming Basics

- Introduction to OOP
- Classes and Objects
- Instance methods and data
- Initialization of objects
- Inheritance
- Multiple and Multilevel Inheritance
- Method overriding
- Classes and Types

Questions? Request a call back

Request An Onsite Class