

Introduction to Python Programming

Length: 5 Days

Pre-requisites: Students should already be comfortable using the operating system (Linux, Unix, Windows, Solaris, Mac OS X, etc.) on which they will be running Python. While not mandatory, basic skills with at least one other programming language are desirable.

Summary: This Python training course leads the students from the basics of writing and running Python scripts to more advanced features such as file operations, regular expressions, working with binary data, and using the extensive functionality of Python modules. Extra emphasis is placed on features unique to Python, such as tuples, array slices, and output formatting.

Upon completion of this course, students will be able to:

- Master the fundamentals of writing Python scripts
- Learn core Python scripting elements such as variables and flow control structures
- Discover how to work with lists and sequence data
- Write Python functions to facilitate code reuse
- Use Python to read and write files
- Make their code robust by handling errors and exceptions properly
- Work with the Python standard library
- Explore Python's object-oriented features
- Search text using regular expressions

COURSE CONTENT

An Overview of Python

- What is Python?
- Interpreted languages
- Advantages and disadvantages
- Downloading and installing
- Which version of Python
- Where to find documentation

Running Python Scripts

- Structure of a Python script
- Using the interpreter interactively
- Running standalone scripts under Unix and Windows

Getting Started

- Using variables
- String types: normal, raw and Unicode
- String operators and expressions
- Math operators and expressions
- Writing to the screen
- Command line parameters
- Reading from the keyboard

Flow Control

- About flow control
- Indenting is significant
- The if and elif statements
- while loops
- Using lists
- Using the for statement
- The range() function

Sequence Data

- list operations
- list methods
- Strings are special kinds of lists
- Tuples
- Sets
- Dictionaries

Defining Functions

- Syntax of function definition
- Formal parameters
- Global versus local variables
- Passing parameters and returning values

Working with Files

- Text file I/O overview
- Opening a text file
- Reading text files
- Raw (binary) data
- Using the pickle module
- Writing to a text file

Dictionaries and Sets

- Dictionary overview
- Creating dictionaries
- Dictionary functions
- Fetching keys or values
- Testing for existence of elements
- Deleting elements

Errors and Exception Handling

- Dealing with syntax errors
- Exceptions
- Handling exceptions with try/except
- Cleaning up with finally

Using Modules

- What is a module?
- The import statement
- Function aliases
- Packages

Regular Expressions

- RE Objects
- Pattern matching
- Parsing data
- Subexpressions
- Complex substitutions
- RE tips and tricks

Highlights of the Standard Library

- Working with the operating system
- Grabbing web pages
- Sending email
- Using glob for filename wildcards
- math and random
- Accessing dates and times with datetime
- Working with compressed files

An Introduction to Python Classes

- About o-o programming
- Defining classes
- Constructors
- Instance methods
- Instance data
- Class methods and data
- Destructors