

Python Programming

Course Summary

Description

This course leads the student 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. This is a hands-on programming class. All concepts are reinforced by informal practice during the lecture followed by graduated lab exercises.

Python Programming is a practical introduction to a working programming language, not an academic overview of syntax and grammar. Students will immediately be able to use Python to complete tasks in the real world.

Topics

- Running Python Scripts
- Getting Started
- Flow Control
- Sequence Data
- Defining Functions
- Working with Files
- Dictionaries and Sets
- Errors and Exception Handling
- Using Modules
- Regular Expressions
- Highlights of the Standard Library
- Introduction to Python Classes

Audience

This course is appropriate for 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

Students should already have a working, user-level knowledge of an operating system such as UNIX or Windows XP. While not mandatory, basic skills with at least one other programming language are desirable. When presenting this course to students with little or no programming experience, we suggest the class length be extended to five days, omitting chapter 19, "Putting Python to Work".

Duration

Four days

Python Programming

Course Summary

I. An Overview Of Python

- A. What is Python?
- B. Interpreted languages
- C. Advantages and disadvantages
- D. Downloading and installing
- E. Which version of Python
- F. Where to find documentation

II. Running Python Scripts

- A. Structure of a Python program
- B. Using the interpreter interactively
- C. Running standalone scripts under Unix and Windows

III. Getting Started

- A. Using variables
- B. String types: normal, raw, and Unicode
- C. String operators and expressions
- D. Numeric literals
- E. Math operators and expressions
- F. Writing to the screen
- G. Command line parameters
- H. Reading from the keyboard

IV. Flow Control

- A. About flow control
- B. Indenting is significant
- C. The if and elif
- D. while loops
- E. Using lists
- F. Using the for statement
- G. The range() function

V. Sequence Data

- A. list operations
- B. list methods
- C. Strings are special kinds of lists
- D. tuples
- E. sets
- F. dictionaries

VI. Defining Functions

- A. Syntax of function definition
- B. Formal parameters
- C. Global versus local variables
- D. Passing parameters and returning values

VII. Working With Files

- A. Text file I/O overview
- B. Opening a text file
- C. Reading text files
- D. Raw (binary) data
- E. Using the pickle module
- F. Writing to a text file

VIII. Dictionaries and Sets

- A. Dictionary overview
- B. Creating dictionaries
- C. Dictionary functions
- D. Fetching keys or values
- E. Testing for existence of elements
- F. Deleting elements

IX. Errors and Exception Handling

- A. Dealing with syntax errors
- B. Exceptions
- C. Handling exceptions with try/except
- D. Cleaning up with finally

X. Using Modules

- A. What is a module?
- B. The import statement
- C. Function aliases
- D. Packages

XI. Regular Expressions

- A. RE Objects
- B. Pattern matching
- C. Parsing data
- D. Subexpressions
- E. Complex substitutions
- F. RE tips and tricks

XII. Highlights of the Standard Library

- A. Working with the operating system
- B. Grabbing web pages
- C. Sending email
- D. Using glob for filename wildcards
- E. Math and random
- F. Accessing dates and times with datetime
- G. Working with compressed files

Python Programming
Course Summary (cont'd)

XIII. An Introduction To Python Classes

- A. About o-o programming
- B. Defining classes
- C. Constructors
- D. Instance methods
- E. Instance data
- F. Class methods and data
- G. Destructors

XIV. Special Chapter: Putting Python To Work

- A. This chapter, taught only in the 5-day version of the course, contains advanced tasks for the students to try; we encourage students to bring in and work on their real-life projects as well.