

Introduction to Python 3.x

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.

Objectives

At the end of this course, students will be able to:

- Create working Python scripts following best practices
- Use python data types appropriately
- Read and write files with both text and binary data
- Search and replace text with regular expressions
- Get familiar with the standard library and its work-saving modules
- Know when to use collections such as lists, dictionaries, and sets
- Understand Pythonic features such as comprehensions and iterators
- Write robust code using exception handling

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 Unix/Linux, Mac, or Windows. While not required, basic skills with at least one other programming language will be helpful.

Duration

Four days

Introduction to Python 3.x

Course Outline

I. An overview of Python

- A. What is python?
- B. Python Timeline
- C. Advantages/Disadvantages of Python
- D. Getting help with pydoc

II. The Python Environment

- A. Starting Python
- B. Using the interpreter
- C. Running a Python script
- D. Python scripts on Unix/Windows
- E. Editors and IDEs

III. Getting Started

- A. Using variables
- B. Builtin functions
- C. Strings
- D. Numbers
- E. Converting among types
- F. Writing to the screen
- G. Command line parameters

IV. Flow Control

- A. About flow control
- B. White space
- C. Conditional expressions
- D. Relational and Boolean operators
- E. While loops
- F. Alternate loop exits

V. Sequences

- A. About sequences
- B. Lists and list methods
- C. Tuples
- D. Indexing and slicing
- E. Iterating through a sequence
- F. Sequence functions, keywords, and operators
- G. List comprehensions
- H. Nested sequences

VI. Working with files

- A. File overview
- B. Opening a text file
- C. Reading a text file
- D. Writing to a text file
- E. Screen output without formatting
- F. Raw (binary) data

VII. Dictionaries and Sets

- A. About dictionaries
- B. Creating dictionaries
- C. Iterating through a dictionary
- D. About sets
- E. Creating sets
- F. Working with sets

VIII. Functions

- A. About sequences
- B. Function parameters
- C. Global variables
- D. Global scope
- E. Returning values
- F. Creating a library of functions

IX. Sorting

- A. The sorted() function
- B. Alternate keys
- C. Lambda functions
- D. Sorting collections
- E. Using operator.itemgetter()
- F. Reverse sorting

X. Errors and Exception Handling

- A. Syntax errors
- B. Exceptions
- C. Using try/catch/else/finally
- D. Handling multiple exceptions
- E. Ignoring exceptions

Introduction to Python 3.x

Course Outline (cont'd)

XI. Using Modules

- A. The import statement
- B. Module search path
- C. Zipped libraries
- D. Creating Modules
- E. Function and Module aliases

XII. Regular Expressions

- A. RE syntax overview
- B. Basic patterns
- C. RE Objects
- D. Searching and matching
- E. Compilation flags
- F. Grouping
- G. Replacing text
- H. Splitting a string

XIII. Using the Standard Library

- A. The sys module
- B. Launching external programs
- C. Walking directory trees
- D. Grabbing web pages
- E. Sending e-mail
- F. Paths, directories, and filenames
- G. Math functions
- H. Random numbers
- I. Dates and times
- J. Zipped archives

XIV. 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