

Course 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.

Who Should Attend: 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.

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

- Write and run Python scripts.

Course Outline:

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 program
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
Numeric literals
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
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