



## Python Programming

### Duration

5 days

### Course Description

This course, designed for professional software developers, scientists, and engineers, is a comprehensive introduction to the Python programming language, standard library, and Python programming techniques. Although the course assumes no prior experience with Python, the course is strongly focused on practical applications including scripting, data processing, systems administration, and integrating Python with other software.

### Course Objectives

Upon completion of this training, the participants will be able to do:

- Scripting
- Systems Administration
- Integrate Python with other software
- Work effectively with the Python programming language, standard library, and Python programming techniques
- Data processing

### Audience

- Programmers who want to learn Python
- Anyone interested in learning backend web languages



## Prerequisites

- Although no prior experience with Python is required, this course assumes that students have prior experience with some other programming language such as C++, Java, or Perl. This is not an introductory class for absolute beginners on how to program a computer. Participants should already be familiar with the basic concepts of programming such as variables, statements, control-flow, functions, arrays, data structures, and common programming problems (e.g., searching, sorting, etc.).

## Recommended next course

- Website Development with HTML5 and CSS3
- Java Programming Language - Intermediate Level

## Course Topics

### Unit 1 - Modules and Libraries

- How to organize programs into modules and details on using modules as a tool for creating extensible programs. Concludes with a tour of some of the most commonly used library modules including those related to system administration, text processing, subprocesses, XML parsing, binary data handling, and databases. In addition, an optional section on using numpy and matplotlib to process numeric data can be taught depending on student interest.

### Unit 2 - Text I/O Handling

- More information on text-based I/O. Topics include text generation, template strings, and Unicode.

### Unit 3 - Python Integration Primer

- A survey of how Python is able to interact with programs written in other programming languages. Topics include network programming, accessing C code, COM extensions, Jython, and IronPython.



## Unit 4 - Introduction to Python

- Covers details of how to start and stop the interpreter and write programs. Introduces Python's basic datatypes, files, functions, and error handling.

## Unit 5 - Working with Data

- A detailed tour of how to represent and work with data in Python. Covers tuples, lists, dictionaries, and sets. Students will also learn how to effectively use Python's very powerful list processing primitives such as list comprehensions. Finally, this section covers critical aspects of Python's underlying object model including variables, reference counting, copying, and type checking.

## Unit 6 - Program Organization and Functions

- More information about how to organize larger programs into functions. A major focus of this section is on how to design functions that are reliable and can be easily reused in other settings. Also covers technical details of functions including scoping rules and documentation strings.

## Unit 7 - Classes and Objects

- An introduction to object-oriented programming in Python. Describes how to create new objects, overload operators, and utilize Python special methods. Also covers basic principles of object oriented programming including inheritance and composition.

## Unit 8 - Inside the Python Object System

- A detailed look at how objects are implemented in Python. Major topics include object representation, attribute binding, inheritance, memory management, and special properties of classes including properties, slots, and private attributes.

## Unit 9 - Testing, Debugging, and Software Development Practice

- This section discusses many issues that are considered important to Python software development. This includes effective use of documentation strings, program testing using both the doctest and unittest modules, and effective use of assertions. The Python debugger and profiler are also described.



## Unit 10 - Iterators and Generators

- Covers the iteration protocol, iterable objects, generators and generator expressions. A major focus of this section concerns the use of generators to set up data processing pipelines--a particularly effective technique for addressing a wide variety of common systems programming problems (e.g., processing large datafiles, handling infinite data streams, etc.).

## Unit 11 - Some Advanced Topics

- A variety of more advanced programming topics including variable argument functions, anonymous functions (lambda), closures, decorators, static and class methods, and packages.