

Mastering Python Programming with Next-Level Topics



Mastering Python Programming is an introductory and beyond-level practical, hands-on Python training course that 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 comprehensive, practical course provides an in-depth exploration of working with the 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.

Key Learning Areas

This course is about 50% hands-on lab to 50% lecture ratio, combining engaging, informed instructor presentations, demonstrations and discussions with extensive machine-based student labs and practical project work. Throughout the course students will learn to write essential Python scripts using the most current and efficient skills and techniques.

Working within in an engaging, hands-on learning environment, guided by our expert instructor, students will learn 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
- Use lesser known but powerful Python data types
- Create "real-world", professional Python applications
- Work with dates, times, and calendars
- 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

Who Should Attend?

This in an introductory-level Python course geared for experienced users who want to use Python in web development projects, or 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.

Duration

5 days

Formats

On-site and virtual

Schedule

Contact us

Pricing

Contact us

Prerequisites

Basic familiarity with any programming or scripting language would be helpful, along with a working, user-level knowledge of Unix/Linux, Mac, or Windows.

Course Outline

An Overview of Python

- What is python?
- Python Timeline
- Advantages/Disadvantages of Python
- Getting help with pydoc

The Python Environment

- Starting Python
- Using the interpreter
- Running a Python script
- Python scripts on Unix/Windows
- Editors and IDEs

Getting Started

- Using variables
- Built-in functions
- Strings
- Numbers
- Converting among types
- Writing to the screen
- Command line parameters

Flow Control

- About flow control
- White space
- Conditional expressions
- Relational and Boolean operators
- While loops
- Alternate loop exits

Array Types

- About array types (AKA sequences)
- Lists and list methods
- Tuples
- Indexing and slicing
- Iterating through a sequence
- Nested sequences
- Sequence functions, keywords and

operators

- List comprehensions
- Generator Expressions

Working with Files

- File overview
- Opening a text file
- Reading a text file
- Writing to a text file
- Reading and writing raw (binary) data
- Converting binary data with struct

Dictionaries and Sets

- About dictionaries
- Creating dictionaries
- Iterating through a dictionary
- About sets
- Creating sets
- Working with sets

Functions

- Defining functions
- Parameters
- Global and local scope
- Nested functions
- Returning values

Sorting

- The sorted() function
- Alternate keys
- Lambda functions
- Sorting collections
- Using operator.itemgetter()
- Reverse sorting

Errors and Exception Handling

- Syntax errors
- Exceptions

- Using try/catch/else/finally
- Handling multiple exceptions
- Ignoring exceptions

Modules and Packages

- The import statement
- Module search path
- Creating Modules
- Using packages
- Function and Module aliases

Classes

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

Regular Expressions

- RE syntax overview
- RE objects
- Searching and matching
- Compilation flags
- Groups and special groups
- Replacing text
- Splitting strings

The Standard Library

- The sys module
- Launching external programs
- Math functions
- Random numbers
- Reading CSV data

Dates and Times

- Working with dates and times
- Translating timestamps
- Parsing dates from text

- Formatting dates
- Calendar data

Working with the File System

- Paths, directories, and filenames
- Checking for existence
- Permissions and other file attributes
- Walking directory trees
- Creating filters with fileinput
- Using shutil for file operations

Advanced Data Handling

- Defaultdict and Counter
- Prettyprinting data structures
- Compressed archives (zip, gzip, tar, etc.)
- Persistent data

Network Programming

- Using requests
- Grabbing web content
- Sending email
- Using SSH for remote access
- Using FTP

Writing Real-Life Applications

- Reading input files, a la Unix
- Parsing command-line options
- Detecting the current platform
- Implementing logging