

Course Outline

Python Boot Camp | Mastering Python Programming Essentials Course TTPS4820: 5 days Instructor Led

About this course

A core component of our Python Journey series, the Complete Python / Practical Python Boot Camp, is a highly rated training course that has helped thousands of students get up and running with solid, practical Python skills, enabling them to leverage Python on the job. You'll be able to use Python to automate or simplify day to day common tasks with the use of Python scripts, or learn how to use Python for web development. You'll have a solid foundation to build additional skills around using Python for analytics, data science or machine learning, or advanced web development.

This introductory-level course leads you from the basics of writing and running Python scripts, to more advanced core features such as file operations, regular expressions, working with binary data, as well as 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 you with an in-depth exploration of working with the programming language, not just an academic overview of syntax and grammar. You'll exit this course immediately able to use Python scripting to complete tasks in the real world.

This course combines expert lecture, real-world demonstrations and group discussions with machine-based practical labs and exercises. Our engaging instructors and mentors are highly experienced practitioners who bring years of current "on-the-job" experience into every classroom. Throughout the hands-on course, you'll learn to write essential Python scripts using the most current and efficient skills, best practices and techniques.

Audience profile

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

At course completion

After completing 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
- 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

Course Outline

Quick Intro to Python

- What is Python?

Course Outline

- Advantages/disadvantages
- Python timeline
- Installing Python
- Getting help

Setting up the Python Environment

- Starting Python
- Using the interpreter
- Running a Python script
- Editors and IDEs

Getting Started

- Using variables
- Builtin functions
- String data
- Numeric data
- Converting types
- Console input/output
- Command line arguments

Flow Control

- About flow control
- The if statement
- Relational and Boolean values
- while loops
- Exiting from loops

Array Types

- Sequence types in general
- Lists and list methods
- Tuples
- Indexing and slicing
- Iterating through a sequence
- Sequence functions, keywords, and operators
- List comprehensions and generators

Working with files

- File I/O overview
- Opening a text file
- Reading a text file
- Writing to a text file

Dictionaries and Sets

- About dictionaries
- Creating dictionaries
- Getting values
- Iterating through a dictionary

Course Outline

- About sets
- Creating sets
- Working with sets

Functions

- Defining functions
- Returning values
- Parameters and arguments
- Variable scope

Sorting

- The sorted() function
- Custom sort keys
- Lambda functions
- Sorting in reverse
- Using min() and max()

Errors and Exception Handling

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

Modules and Packages

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

Object Oriented Programming & Classes

- About object-oriented programming
- Defining classes
- Constructors
- Understanding self
- Properties
- Instance Methods and data
- Class methods and data
- Inheritance

Regular Expressions

- RE syntax overview
- RE objects
- Searching and matching
- Compilation flags
- Groups and special groups
- Search-and-replace
- Splitting strings

Course Outline

The standard library

- The sys module
- Math functions
- Random numbers
- Reading CSV data
- Zip files
- Reading pages and files from URLs
- Sending email

Dates and times

- Date and time representations
- Parsing dates from text
- Formatting as text
- Converting representations
- Calendar data
- Time zones

Working with the file system

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

Advanced data handling

- Defaultdict and Counter
- Pretty-printing 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