Document Generated: 11/03/2026

Learning Style: Virtual Classroom

Technology:

Difficulty: Advanced

Course Duration: 5 Days

Mastering Python Programming Boot Camp (TTPS4820)



About This Course:

As a cornerstone of our Python Skill Journey series, our Mastering Python Programming Boot Camp stands as a top-tier training experience, acclaimed for transforming Python novices into capable developers. Whether you aim to streamline routine tasks through automated Python scripts or venture into the world of web development, this course serves as both a launchpad and a compass, guiding you toward exciting horizons in analytics, data science, machine learning, and beyond.

Course Objectives:

This course combines engaging instructor-led presentations and useful demonstrations with valuable hands-on labs and engaging group activities. Throughout the course you'll learn how 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, tuples, dictionaries, and sets
- Understand Pythonic features such as list comprehensions and generators
- Write robust code using exception handling
- Create and use virtual environments

Audience:

This course is geared for technical users who are new to Python. Roles might include developers, software engineers, data analysts who want to enhance data processing, system administrators and web site administrators who want to use Python to support their server installations, developers who want more efficient web solutions, as well as anyone else who wants to automate or simplify common tasks with the use of Python scripts.

Prerequisites:

To ensure a smooth learning experience and maximize the benefits of attending this course, you should have the following prerequisite skills:

- At least some prior hands-on experience with scripting or programming. You don't need to be an expert in either, but you should have had some exposure and should be coming from a technical background.
- Working with Unix or Linux, and familiarity with using the command line interface for simple tasks, such as file navigation and executing commands.

- Basic familiarity working with text editors like Notepad, or IDEs, would be helpful as the course includes hands-on lab sessions requiring code editing.

Course Outline:

The Python Environment

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

Variables and Values

- Using variables
- Builtin functions
- String data
- Numeric data
- Converting types

Basic input and output

- Writing to the screen
- String formatting
- Command line arguments
- Reading the keyboard

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
- 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()

Exception handling and logging

- Exceptions
- Using try/catch/else/finally
- Handling multiple exceptions
- Logging setup
- Basic logging

Modules and Packages

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

Introduction to 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

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

Effective Scripts

- Reading input files a la Unix

- Parsing command-line options
- Detecting the current platform
- Implementing logging

Virtual Environments

- Why are virtual environments needed
- Creating a virtual env
- Replicating an environment
- Virtual environment issues

Credly Badge:



Display your Completion Badge And Get The Recognition You Deserve.

Add a completion and readiness badge to your Linkedin profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

Find Out More or See List Of Badges