Advanced Python Programming - TTPS4850

Master next-level Python skills to build efficient, scalable solutions for real-world automation, data integration, and application development.

Duration: 4 Days
Skill Level: Intermediate
Available Format: Instructor-Led Online; Instructor-Led, Onsite In Person ; Blended; On Public Schedule

Take your Python skills to the next level with this expert-led, hands-on course designed for professionals who want to write cleaner, more efficient, and more powerful code. You will learn how to automate system tasks, work with databases and APIs, debug and optimize performance, and handle advanced programming concepts like concurrency, multiprocessing, and meta-programming. Whether you are building scalable applications, improving workflows, or tackling complex data processing, this course provides the practical skills and real-world experience needed to write better Python code with confidence. With 50% hands-on exercises and live guidance from an expert instructor, you will not just learn—you will apply what you learn in meaningful ways.

What You'll Learn

Overview

Advanced Python Programming is a four-day, expert-led course designed for professionals with basic Python experience who want to move beyond simple scripts and develop real-world programming expertise. Python is a powerful and versatile language, but to fully harness its potential, you need to understand how to write more efficient, scalable, and maintainable code. This course helps bridge that gap by focusing on practical skills that make your Python development faster, more organized, and better suited for professional applications.

Guided by our expert instructor, you will gain hands-on experience with Python's advanced features, learning how to work with OS services, automate tasks, manage files and directories, and apply best practices in structuring and optimizing your code. These next-level skills are essential for anyone looking to build robust applications, streamline workflows, or handle large-scale data processing. You will learn how to debug and profile code effectively, implement unit tests using PyTest, and interact with databases using SQL and ORM tools. You will also explore network programming, work with APIs, and leverage concurrency techniques like threading, multiprocessing, and asynchronous programming to improve performance. For those interested in application development, the course includes practical instruction on GUI development with PyQt, as well as working with structured data formats like JSON, XML, and YAML.

With 50% of the course dedicated to hands-on exercises, you will not just learn these skills—you will apply them in real scenarios, giving you the confidence and experience to use Python effectively in professional settings.

Objectives

By the end of this expert-led, hands-on course, you will have the skills to write more efficient, organized, and professional Python code. You will build confidence in using Python for real-world applications, from automating tasks to handling complex data and building scalable programs.

Here are a few of the key skills you will gain:

Write More Efficient and Readable Code – Use Python's advanced data structures, functional programming techniques, and best practices to make your code cleaner, faster, and easier to maintain.

Automate System and File Operations – Work with OS services, process files and directories, launch external programs, and manage system tasks with Python scripts.

Work with Databases and APIs – Connect to SQL databases, execute queries, and integrate with web services using Python's database libraries and HTTP request tools.

Improve Debugging and Code Performance – Use debugging tools, performance profiling, and unit testing with PyTest to identify issues and optimize your programs.

Handle Concurrency and Parallel Processing – Write more responsive and scalable programs using threading, multiprocessing, and asynchronous programming techniques.

Build Applications and Process Data Effectively – Develop GUI applications with PyQt, process structured data formats like JSON, XML, and YAML, and serialize data for storage and exchange.

With 50% hands-on practice and expert guidance, you will not only learn these skills—you will apply them, ensuring you leave the course ready to take on more complex Python projects with confidence.

If your team requires different topics, additional skills or a custom approach, our team will collaborate with you to adjust the course to focus on your specific learning objectives and goals.

Audience

This intermediate and beyond level course is designed for professionals who have basic Python experience and want to deepen their skills for more advanced programming, automation, and application development. It is ideal for software developers, data analysts, system administrators, and engineers looking to write more efficient code, work with real-world data, and build scalable applications.

Pre-Requisites

To get the most out of this expert-led, hands-on course, you should have a basic understanding of Python and some experience writing simple scripts. This course builds on foundational Python skills, so you will be ready to take on more advanced programming concepts.

Here are a few key prerequisites:

Basic Python Syntax and Data Structures – You should be comfortable with variables, loops, conditionals, functions, and common data types like lists, dictionaries, and tuples.

Writing and Running Simple Python Scripts – Experience creating and executing Python scripts, whether for basic automation, simple data processing, or small projects.

Some Exposure to Modules and File Handling – A general understanding of how to import and use Python modules, as well as read and write to files, will be helpful.

Take Before: Advanced Python programming requires incoming basic experience with Python. Each course below teaches basic Python experience, so each course serve as a standalone, solid prerequisite to attend the advanced class. Any ONE of the classes below would apply.

Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We'll work with you to tune this course and level of coverage to target the skills you need most. Topics, agenda and labs are subject to change and may adjust during live delivery based on audience skill level, interests and participation.

1. Python refresher

Builtin data types

Lists and tuples

Dictionaries and sets

Program structure

Files and console I/O

If statement

for and while loops

2. OS Services

The os and os.path modules

Environment variables

Launching external commands with subprocess

Walking directory trees

Paths, directories, and filenames

Working with file systems

3. Dates and Times

Basic date and time classes

Different time formats

Converting between formats

Formatting dates and times

Parsing date/time information

4. Binary Data

What is Binary Data?

Binary vs text

Using the Struct module

5. Pythonic Programming

The Zen of Python

Tuples

Advanced unpacking

Sorting

Lambda functions

List comprehensions

Generator expressions

String formatting

6. Functions, modules, and packages

Four types of function parameters

Four levels of name scoping

Single/multi dispatch

Relative imports

Using init effectively

Documentation best practices

7. Intermediate classes

Class/static data and methods

Inheritance (or composition)

Abstract base classes

Implementing protocols (context, iterator, etc.) with special methods

8. Metaprogramming

Implicit properties

globals() and locals()

Working with object attributes

The inspect module

Callable classes

Decorators

Monkey patching

9. Developer Tools

Analyzing programs with pylint

Using the debugger

Profiling code

Testing speed with benchmarking

10. Unit testing with PyTest

What is a unit test?

Writing tests

Working with fixtures

Test runners

Mocking resources

11. Database access

The DB API

Available Interfaces

Connecting to a server

Creating and executing a cursor

Fetching data

Parameterized statements

Using Metadata

Transaction control

ORMs and NoSQL overview

12. PyQt

Overview

Qt Architecture

Using designer

Standard widgets

Event handling

Extras

13. Network Programming

Builtin classes

Using requests

Grabbing web pages

Sending email

Working with binary data

Consuming RESTful services

Remote access (SSH)

14. Multiprogramming

The threading module

Sharing variables

The queue module

The multiprocessing module

Creating pools

About async programming

15. Scripting for System Administration

Running external programs

Parsing arguments

Creating filters to read text files

Implementing logging

16. Serializing data

Parsing XML with ElementTree

Updating the XML tree

Creating new XML documents

Reading/Writing JSON data

Reading/writing CSV data

YAML, other formats

Time Permitting Chapters

17. Virtual Environments

Use case for

Creating an environment

Activating and deactivating

Replicating an environment

Useful tools

18. Type hinting

Annotate variables

Learn what type hinting does NOT do

Use the typing module for detailed type hints

Understand union and optional types

Write stub interfaces

19. Advanced data handling

Discover the collections module

Use defaultdict, Counter, and namedtuple

Create dataclasses

Store data offline with pickle

Our core Python courses are available for either local installation, where all course items are supplied for setup directly onto your students machines (or server), or can be run on our engaging online Learning Experience Platform (LXP) that hosts all course software, content, labs and solutions on our “easy access, single source, no install required” cloud-based hands-on learning environment. Either way, we’ll collaborate with you to ensure your team is set up and ready to go well in advance of the class.

Please inquire about set up details and options for your specific course of interest.

For More Information

Please contact us or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.