Advanced Python Programming / Next-Level Python (TTPS4850)

Category: Software Development & Programming | **Vendor:** Trivera Tech

Duration: 32.00 hours (4 days) **26.0 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Online

Course Overview

Advanced Python Programming®is a practical, hands-on Python training course that thoroughly explores intermediate to advanced level topics and skills, with a focus on enterprise development. ®Throughout the course, students will learn how to Leverage OS services, Code graphical interfaces for applications, create modules and run unit tests, define classes, interact with network services, query databases, process XML data, and much more.® This comprehensive, practical course provides an in-depth exploration of working with the programming language, not an academic overview of syntax and grammar.®

About This Course

Next Level Python Programming (Advanced Python) is a practical, hands-on Python training course that thoroughly explores intermediate to advanced-level topics and skills, with a focus on enterprise development. Throughout the course, students will learn how to Leverage OS services, Code graphical interfaces for applications, create modules and run unit tests, define classes, interact with network services, query databases, process XML data, and much more. This comprehensive, practical course provides an in-depth exploration of working with the programming language, not an academic overview of syntax and grammar.

Who Should Attend

This in an intermediate and beyond-level Python course geared for experienced Python programmers, focused on next-level skills for enterprise development.

Prerequisites & Entry Requirements

General Prerequisites:

This in an intermediate and beyond level Python course geared for students experienced with Python who want to use Python in
web development projects or automate or simplify common tasks with the use of Python scripts. Basic incoming practical
experience working with Python is required, along with a working, user-level knowledge of Unix/Linux, Mac, or Windows. This
course does not cover Python fundamentals.

Learning Outcomes

Upon successful completion of this course, participants will be able to:

This course is approximately 50% hands-on, combining 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.

Working in a hands-on learning environment led by our expert practitioner, attendees will learn advanced skills needed to:

- Leverage OS services
- Add enhancements to classes
- Code graphical interfaces for applications
- Understand advanced Python metaprogramming concepts
- Create easy-to-use and easy-to-maintain modules and packages
- Implement and run unit tests
- Create multithreaded and multi-process applications
- Interact with network services
- Design professional scripts
- Query databases
- Process XML, CSV, and JSON data
- Working with more data types if time permits
- Using type hints if time permits

Detailed Course Outline

Python Quick Refresher

Builtin data types

Lists and tuples

Dictionaries and sets

Program structure

Files and console I/O

If statement

for and while loops

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

Dates and Times

Basic date and time classes

Different time formats

Converting between formats

Formatting dates and times

Parsing date/time information

Binary Data

What is Binary Data

Binary vs text

Using the Struct module

Pythonic Programming

The Zen of Python

Tuples

Advanced unpacking

Sorting

Lambda functions

List comprehensions

Generator expressions

String formatting

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

Intermediate classes

Class/static data and methods

Inheritance (or composition)

Abstract base classes

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

Metaprogramming

Implicit properties

globals() and locals()

Working with object attributes

The inspect module

Callable classes

Decorators

Monkey patching

Developer Tools

Analyzing programs with pylint

Using the debugger

Profiling code

Testing speed with benchmarking

Unit testing with PyTest

What is a unit test

Writing tests

Working with fixtures

Test runners

Mocking resources

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

PyQt

Overview

Qt Architecture

Using designer

Standard widgets

Event handling

Extras

Network Programming

Builtin classes

Using requests

Grabbing web pages

Sending email

Working with binary data

Consuming RESTful services

Remote access (SSH)

Multiprogramming

The threading module

Sharing variables

The queue module

The multiprocessing module

Creating pools

About async programming

Scripting for System Administration

Running external programs

Parsing arguments

Creating filters to read text files

Logging

Serializing data €œ XML and JSON

Working with XML

XML modules in Python

Getting started with ElementTree

Parsing XML

Updating an XML tree

Creating a new document

About JSON

Reading JSON

Writing JSON

Reading/writing CSV files

YAML, other formats as time permits

Time Permitting Sessions

Advanced data handling

Discover the collections module

Use defaultdict, Counter, and namedtuple

Create dataclasses

Store data offline with pickle

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

Additional Course Details

Nexus Humans Advanced Python Programming / Next-Level Python (TTPS4850) training program is a workshop that presents an invigorating mix of sessions, lessons, and masterclasses meticulously crafted to propel your learning expedition forward.

This immersive bootcamp-style experience boasts interactive lectures, hands-on labs, and collaborative hackathons, all strategically designed to fortify fundamental concepts.

Guided by seasoned coaches, each session offers priceless insights and practical skills crucial for honing your expertise. Whether you're stepping into the realm of professional skills or a seasoned professional, this comprehensive course ensures you're equipped with the knowledge and prowess necessary for success.

While we feel this is the best course for the Advanced Python Programming / Next-Level Python (TTPS4850) course and one of our Top 10 we encourage you to read the course outline to make sure it is the right content for you.

Additionally, private sessions, closed classes or dedicated events are available both live online and at our training centres in Dublin and London, as well as at your offices anywhere in the UK, Ireland or across EMEA.

Frequently Asked Questions

Q: What delivery options are available for Advanced Python Programming / Next-Level Python (TTPS4850)?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: How many CPD hours does this course provide?

The 4-day Advanced Python Programming / Next-Level Python (TTPS4850) course provides up to 26.0 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the Advanced Python Programming / Next-Level Python (TTPS4850) training?

The training takes place over 4 day(s), with each day lasting approximately 32.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for Advanced Python Programming / Next-Level Python (TTPS4850)?

Yes, we provide corporate training, dedicated training, and closed classes for Advanced Python Programming / Next-Level Python (TTPS4850). Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: Python Intermediate Advanced

---

Q: Why choose Nexus Human for Advanced Python Programming / Next-Level Python (TTPS4850)?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your Advanced Python Programming / Next-Level Python (TTPS4850) training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 889 8700 (Ireland) | +44 20 8142 8961 (UK)

© 2026 Nexus Human. All rights reserved. This brochure was generated on 14/11/2026.