



Advanced Python Programming / Next-Level Python Programming

Intermediate and Beyond Level Skills for Python Experienced Developers

www.triveratech.com

Course Snapshot

- **Course:** Next Level Python | Advanced Python Programming (TTPS4850)
- **Duration:** 4 days
- **Audience & Skill-Level:** This is an **intermediate and beyond-level** Python course geared for experienced Python programmers, focused on next-level skills for enterprise development.
- **Hands-on Learning:** This course is approximately **50% hands-on**, combining expert lecture, real-world demonstrations and group discussions with machine-based practical labs and exercises. Student machines are required.
- **Delivery Options:** This course is available for **onsite private classroom presentation, live online virtual presentation**, or can be presented in a **flexible blended learning format** for combined onsite and remote attendees. Please also ask about our **Self-Paced / Video / QuickSkills or Mini-Camp Flex Hours / Short Course** options.
- **Public Schedule:** This course has active dates on our live-online open enrollment **Public Schedule**.
- **Customizable:** This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please inquire for details.

Overview

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.

Learning Objectives

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*

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate. We offer additional python, web development, data science, machine learning and other related topics that may be blended with this course for a track that best suits your needs. Our team will collaborate with you to understand your needs and will target the course to focus on your specific learning objectives and goals.

Audience & Pre-Requisites

This is 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.

Take Before: Students should have practical skills equivalent to or should have received training in the following topic(s) as a pre-requisite:

- TTPS4800: Introduction to Python | Python Programming Basics (3 days)

Follow On Courses: Our Python tracks include a wide variety of follow-on courses and learning paths for leveraging Python for next-level web development, data science / machine learning, networking, task automation, security and other topics. Please see the attached **Python Training Suite** list of courses, or inquire for recommendations based on your specific role and goals.

Enhanced Learning Services: Please also ask about our **Pre-Training Class OnRamp & Prep / Primer** offerings, **Skills Gap Assessment Services, Case Studies, Knowledge Check Quizzes, Skills Immersion Programs & Camps, Collaborative Mentoring Services** and **Extended Learning Support & Post Training** services.

Course Topics / 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 needs and skill-level.

- | | | |
|--|--|--|
| <p>1. Python refresher</p> <ul style="list-style-type: none"> • Builtin data types • Lists and tuples • Dictionaries and sets • Program structure • Files and console I/O • If statement • <i>for</i> and <i>while</i> loops <p>2. OS Services</p> <ul style="list-style-type: none"> • The os and os.path modules • Environment variables • Launching external commands with subprocess • Walking directory trees • Paths, directories, and filenames • Working with file systems <p>3. Dates and Times</p> <ul style="list-style-type: none"> • Basic date and time classes • Different time formats • Converting between formats • Formatting dates and times • Parsing date/time information <p>4. Binary Data</p> <ul style="list-style-type: none"> • What is Binary Data? • Binary vs text | <ul style="list-style-type: none"> • Using the Struct module <p>5. Pythonic Programming</p> <ul style="list-style-type: none"> • The Zen of Python • Tuples • Advanced unpacking • Sorting • Lambda functions • List comprehensions • Generator expressions • String formatting <p>6. Functions, modules, and packages</p> <ul style="list-style-type: none"> • Four types of function parameters • Four levels of name scoping • Single/multi dispatch • Relative imports • Using <code>__init__</code> effectively • Documentation best practices <p>7. Intermediate classes</p> <ul style="list-style-type: none"> • Class/static data and methods • Inheritance (or composition) • Abstract base classes • Implementing protocols (context, iterator, etc.) with special | <p>methods</p> <p>8. Metaprogramming</p> <ul style="list-style-type: none"> • Implicit properties • <code>globals()</code> and <code>locals()</code> • Working with object attributes • The inspect module • Callable classes • Decorators • Monkey patching <p>9. Developer Tools</p> <ul style="list-style-type: none"> • Analyzing programs with pylint • Using the debugger • Profiling code • Testing speed with benchmarking <p>10. Unit testing with PyTest</p> <ul style="list-style-type: none"> • What is a unit test? • Writing tests • Working with fixtures • Test runners • Mocking resources <p>11. Database access</p> <ul style="list-style-type: none"> • 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
- Logging

16. Serializing data – 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

17. Advanced data handling

- Discover the **collections** module
- Use **defaultdict**, **Counter**, and **namedtuple**
- Create dataclasses
- Store data offline with **pickle**

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

Appendix A: Bibliography

Appendix B: Python virtual environments

Student Materials: Each student will receive a **Student Guide** with course notes, code samples, software tutorials, diagrams and related reference materials and links (as applicable). Our courses also include step by step hands-on lab instructions and solutions, clearly illustrated for users to complete hands-on work in class, and to revisit to review or refresh skills at any time. Students will also receive related (as applicable) project files, code files, data sets and solutions required for the hands-on work.

Classroom Setup Made Simple: Our dedicated tech team will work with you to **ensure your classroom and lab environment is setup, tested and ready to go** well in advance of the course delivery date, ensuring a smooth start to class and seamless hands-on experience for your students. We offer several flexible student machine setup options including **guided manual set up** for simple installation directly on student machines, or **cloud based / remote hosted lab solutions** where students can log in to a complete separate lab environment minus any installations, or we can supply **complete turn-key, pre-loaded equipment** to bring ready-to-go student machines to your facility. Please inquire for details, options and pricing.

For More Information

Need dedicated training? All courses can be presented **onsite** or **online**, or in a **blended learning format**, tailored to target your specific audience, needs and learning goals. In addition to **full day courses**, we also offer **flex hours, short courses, self-paced options** and more. We train beginner to advanced skills in all areas we cover, and offer **New Hire / Cohort Training, Boot Camps, Skills Immersion Programs, Coaching & Mentoring, Reskilling Programs, Skills Migration & Transition Programs**, and more. We collaborate with you to ensure all courses are truly targeted to meet your specific needs and learning skills, maximizing your valuable training time, as well as your critical budget.

Please visit our extensive **Public Training Schedule** for training for smaller groups or individuals. Please contact us for course details, **Corporate Rates** and **Special Discount Offers**. Our pricing and services are always satisfaction guaranteed.

For more information about our dedicated training services, collaborative mentoring services, courseware licensing options, courseware development services, public course schedule, training management services, partner and reseller programs, or to see our complete list of course offerings and special offers please visit us at www.triveratech.com, email Info@triveratech.com or call us toll free at **844-475-4559**. Our pricing and services are always satisfaction guaranteed.