Mastering Python Programming (TTPS4820)

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

Duration: 40.00 hours (5 days) **32.5 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Online

Course Overview

Mastering Python Programming is an introductory and beyond-level practical, hands-on Python training course that leads the student from the basics of writing and running Python scripts to more advanced features such as file operations, regular expressions, working with binary data, and 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 an in-depth exploration of working with the programming language, not an academic overview of syntax and grammar. Students will immediately be able to use Python to complete tasks in the real world.

About This Course

Mastering Python Programming is an introductory and beyond-level practical, hands-on Python training course that leads the student from the basics of writing and running Python scripts to more advanced features such as file operations, regular expressions, working with binary data, and 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 an in-depth exploration of working with the programming language, not an academic overview of syntax and grammar. Students will immediately be able to use Python to complete tasks in the real world.

Who Should Attend

This introductory-level Python course is 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.

Prerequisites & Entry Requirements

General 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.

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. Throughout the hands-on course students, will learn to write essential Python scripts using the most current and efficient skills and techniques.

Working in a hands-on learning environment, guided by our expert team, attendees will learn 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

Detailed Course Outline

An overview of Python

- What is python
- Python Timeline
- Advantages/Disadvantages of Python
- Getting help with pydoc

The Python Environment

- Starting Python
- Using the interpreter
- Running a Python script
- Python scripts on Unix/Windows
- Editors and IDEs

Getting Started

- Using variables
- Builtin functions
- Strings
- Numbers
- Converting among types
- Writing to the screen
- Command line parameters

Flow Control

- About flow control
- White space
- Conditional expressions
- Relational and Boolean operators
- While loops
- Alternate loop exits

Array types

- About array types (AKA sequences)
- Lists and list methods
- Tuples
- Indexing and slicing
- Iterating through a sequence
- Nested sequences
- Sequence functions, keywords, and operators
- List comprehensions
- Generator Expressions

Working with files

- File overview
- Opening a text file
- Reading a text file
- Writing to a text file
- Reading and writing raw (binary) data
- Converting binary data with struct

Dictionaries and Sets

- About dictionaries
- Creating dictionaries
- Iterating through a dictionary
- About sets
- Creating sets
- Working with sets

Functions

- Defining functions
- Parameters
- Global and local scope
- Nested functions
- Returning values

Sorting

- The sorted() function
- Alternate keys
- Lambda functions
- Sorting collections
- Using operator.itemgetter()
- Reverse sorting

Errors and Exception Handling

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

Modules and Packages

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

An Introduction to Python Classes

- About o-o programming
- Defining classes
- Constructors
- Methods
- Instance data
- Properties
- Class methods and data

Regular Expressions

- RE syntax overview
- RE Objects
- Searching and matching
- Compilation flags
- Groups and special groups
- Replacing text
- Splitting strings

Using the Standard Library

- The sys module
- Launching external programs
- Math functions
- Random numbers
- Reading CSV data

Dates and Times

- Working with dates and times
- Translating timestamps
- Parsing dates from text
- Formatting dates
- Calendar data

Working with the File System

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

Advanced Data Handling

- Defaultdict and Counter
- Prettyprinting 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

, 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, 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, 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, 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

Additional Course Details

Nexus Humans Mastering Python Programming (TTPS4820) 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 Mastering Python Programming (TTPS4820) 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 Mastering Python Programming (TTPS4820)?

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 5-day Mastering Python Programming (TTPS4820) course provides up to 32.5 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the Mastering Python Programming (TTPS4820) training?

The training takes place over 5 day(s), with each day lasting approximately 40.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for Mastering Python Programming (TTPS4820)?

Yes, we provide corporate training, dedicated training, and closed classes for Mastering Python Programming (TTPS4820). 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

Q: Why choose Nexus Human for Mastering Python Programming (TTPS4820)?

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 Mastering Python Programming (TTPS4820) 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.