

Python® for Testers

This is a course for testers who are frustrated by the laborious and manual work that makes up day-to-day test work, anyone who has tried or wants to try scripting and programming in order to help them focus more on the sapient test activities and let the computer do the repetitive work. During the course we will work with a strong focus on practical knowledge and learning by doing with much hands-on coding. This so that attendees can work independently with Python after the course. A large amount of exercises are built to give the opportunity to build simple but powerful tools using Python, which gives a deeper understanding of the opportunities that the language offers.

- Explore examples covering topics such as generating test data, monitoring system under test, compressing and sending test data
- Gain a solid understanding of the Python language, its philosophy, and the code syntax
- Get a broad introduction to the vast standard library that Python comes installed with
- What is Pythonic code, good practices and common pitfalls to avoid
- Learn how to install third party Python libraries to extend the power of the language
- Understand how to control external processes
- Get started with performance measurements
- Learn how transfer results to remote computers

Learn by Doing

This is a course for testers who are frustrated by the laborious and manual work that makes up day-to-day test work, anyone who has tried or wants to try scripting and programming in order to help them focus more on the sapient test activities and let the computer do the repetitive work.

During the course we will work with a strong focus on practical knowledge and learning by doing with much hands-on coding so that attendees can work independently with Python after the course. A large amount of exercises are built to give the opportunity to build simple but powerful tools using Python, which gives a deeper understanding of the opportunities

that the language offers.

Why should I invest time in learning Python?

Because Python is powerful, efficient, and very fun to work in. You feel productive from the start and produce results with surprising speed. It is the perfect choice for a tester who wishes to use a programming language to assist them in their testing, and it's not a language that ends up standing in your way. Python is well established with over 20 years of history, included out of the box in most modern operating systems. It has a diverse, huge, and active community and ecosystem.

Who Should Attend

This course should be attended by engineers in a testing role or those seeking an introduction to programming concepts to develop their skills and learn more about the benefits and power of using Python for testing. The attendees are required to have basic computer knowledge and a genuine interest in learning about scripting and programming.

Laptop Required

Attendees are also required to bring their own laptop to be able to complete the exercises. The laptop must be able to run VirtualBox 5.1 or later. (Learn more at [virtualbox.org](https://www.virtualbox.org) ^[1].)

Course Outline:

Introduction

What is Python?

Basic syntax, structure of Python code

Importing modules

The REPL Basic constructs

The standard library

Python 2 vs 3

Generating Test Data

Built-in data types and objects

Control statements and control flows

Writing data into files

Gathering Test Artifacts

Python Methods

Working with the file systems and operating systems

Manipulating file paths

Compressing and transferring test data

Real Time Monitoring of System Under Test

Generating Test Data Continued

CSV - Comma Separated Values

Compressing data continued

Traversing and mapping file systems

Input from command line

Web Calls, REST APIs, and Monitoring Systems

Installing third party libraries

Working the data format JSON

HTTP Requests

Calling and testing REST APIs

Monitor system under test

Controlling External Processes

PExpect - The Python implementation of Expect

Calling and testing SOAP APIs

Attendees Choice

There will be the opportunity to do ad-hoc exercises depending on the attendees wishes.

Text parsing and manipulation
Manipulating dates and timestamps
Formatting output in terminal
Reading data from files

Class Schedule:

Sign-In/Registration 7:30 - 8:30 a.m.

Morning Session 8:30 a.m. - 12:00 p.m.

Lunch 12:00 - 1:00 p.m.

Afternoon Session 1:00 - 5:00 p.m.

Times represent the typical daily schedule. Please confirm your schedule at registration.

Price:

\$1545

Course Fee Includes:

- Tuition
 - Course notebook
 - Continental breakfasts and refreshment breaks
 - Lunches
 - Letter of completion
-