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 activities. Participants will gain a solid understanding of the Python language, its philosophy, and the code syntax.

- 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.

Participant Recommendations

This course assumes no prior programming knowledge or background and will work itself up from beginner to more advanced concepts. However, it will be helpful if you are familiar with one or more concepts such as how computers work, operating systems, networks, and how concepts such as file systems, files and folders work. In other words, a good technical knowledge as a tester will help you greatly in this class. Any prior experience writing scripts, programming, and using some sort of Unix system (Unix, Solaris, Mac, etc) and a terminal will help as well.

None of the above are hard requirements, but please be aware this is a very hands-on class with a rapid pace, and the less prior "general IT-knowledge" you may have will make it more challenging, albeit not impossible by any means. If you are looking for a class to sit back and relax and listen to only, then this class is not for you. You will be working very hands-on from the start and a great learning experience is guaranteed by writing the code and solving the problems posed by yourself.

Computer and Audio Requirements

Attendees are required to bring their own laptop to be able to complete the exercisesAdditionally, connecting audio via computer (using an internal or external mic or headset) is required for the live virtual class. Please ensure your computer is capable of the following:

- Download a zip-file from the internet and extract it to the hard drive. If the class is taking place in a physical classroom then it is also acceptable to copy the zip-file from a USB-stick.
- Run an OS with a file system that supports larger than 4Gb files (i.e not FAT32 on Windows)
- Install and run VirtualBox. (Learn more at virtualbox.org [1].)
- Support VT-x/AMX-V hardware acceleration in the BIOS, ie run a virtual machine through VirtualBox.

The requirements above will be satisfied for the majority of computers. Please note if you have an older model computer, we highly recommend double checking that your computer is capable of the above requirements. Additionally, if your employer has strict security restrictions in place, please double check that you able to install software, access external USB sticks, etc. VirtualBox 5.1 or later.

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

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.

Price: $1495