

Training information 2018

- Location and dates: check the training calendar
- Schedule: 9.15 - 17.00
- Fee: € 1.195, excluding VAT
- Coffee, tea, lunch and training material are included in the training fee
- Three months free e-coaching
- Information: phone +31 (0)40 – 202 1803 or e-mail opleidingen@improveqs.nl

Possibilities in-house training

Improve provides this training in house and tailored to your specific situation or needs upon request. For further information, please contact us at +31 40 - 202 1803 or opleidingen@improveqs.nl

REGISTRATION

You can register for this training on our website

Improve Quality Services BV
Prof. Dr. Dorgelolaan 30
5613 AM Eindhoven
The Netherlands
www.improveqs.nl

How Python can help you to save valuable testing time!

On all registrations and courses our general and additional conditions apply:
www.improveqs.nl/contact/algemene-voorwaarden

TEST AUTOMATION Python for testers

If you feel there are laborious and repetitive tasks that you spend your valuable test time on. If you want to pick up programming and scripting to help solve that problem. Then this course is for you!



consult.



train.



practice.



TEST AUTOMATION

Python for testers

If you feel there are laborious and repetitive tasks that you spend your valuable test time on. If you want to pick up programming and scripting to help solve that problem. Then this course is for you!

Description

Have you ever been faced with doing laborious and repetitive work as a tester? Of course you have. In our day-to-day work we have to perform many complex, laborious and repetitive tasks. Not only does it take time away from us, it is also mind numbingly boring at times, and it is easy to make mistakes when we have long lists of things to do. But who says it has to be done by you?

You may have heard of Python. "Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability. The language provides constructs intended to enable clear programs on both a small and large scale" (source: Wikipedia). Ideal for testers who want to go fast! Now you can learn more about it. You can see and learn how you can apply it in your day-to-day work.

After this training:

- You can generate test data
- You can monitor the system under test
- You can compress and send test data
- You can clean out test data after performing a test.
- You can delete temporary files and automate starting processes on the system
- You can control external processes.
- You can get started with performance measurements
- You can work independently with Python!

Who should attend this training?

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

Roles of participants in their organization

This training is meant for:

- Testers
- Test Developers
- Developers

Prior knowledge or training required

You must have basic computer knowledge and a genuine interest in learning about scripting and programming.

Program

Structure of this training

This two-day training is very hands-on and intense with a very experienced trainer: Kristoffer Nordström from Northern Test Consulting in Sweden. You won't waste a lot of time on theory, but spend most of the time practicing.

You are required to bring your own laptop to be able to complete the exercises. Please bring a computer that you use regularly. Almost all course material is cross platform and works equally well on Windows, Mac OS X and Linux!

Course topics

- What is Python?
- Basic syntax
- Built-in data types
- The REPL
- Basic constructs
- Input from command line
- Introduction to Object Oriented Python
- The standard library
- Text parsing and manipulation
- Regular expressions
- Generating test data
- Compressing and transferring test data
- Installing 3rd party libraries
- Calling and testing REST APIs
- Controlling external processes
- Monitor system under test
- Python 2 vs 3