

## Selenium 3 Web Driver Scripting with Python

---

### Course Summary

#### Description

This is an intensive workshop on writing Selenium scripts using Python. The course covers all of the core functionality of the Selenium API from finding elements using different selector types to manipulating widgets and working with the various other Web artifacts like pop-ups, and AJAX. The advanced features of Selenium are covered including complex mouse operations and the use of event listeners.

However, the course goes beyond the mechanics of writing code by exploring the internals of what is happening in the API so that students can learn how to both optimize their scripts and troubleshoot issues that may arise in their test environments.

Because Selenium is generally not used as a stand alone tool but tends to be used in conjunction with other test frameworks and tools, students also learn how to integrate Selenium code into the popular unit test frameworks (unittest for example) and behaviour driven test frameworks (behave).

The best practices of the Selenium community are reviewed, including the essential Page Object Pattern.

#### Objective

At the end of the course, students will be able to:

- Write a high quality Selenium script in Python
- Use all of the core functionality of the API in manipulating widgets
- Use waits, timeouts and other features to tune their scripts
- Work with pop-ups, alerts, JavaScript code and AJAX applications
- Integrate Selenium into different test frameworks and other testing tools
- Effectively use the Page Object Pattern
- Work with complex mouse operations
- Create and use event listeners
- Work with advanced features like HTML5 canvas and sessions

#### Audience

The course is targeted at Python developers who will be or are writing Selenium Web Driver scripts in Python.

#### Topics

- The Selenium Python Binding
- Web Driver Navigation Commands
- Locating Elements
- Working with Widgets
- Mouse Actions and Listeners
- The Page Object Pattern
- Selenium Integration
- Advanced Topics
- Best Practices

#### Prerequisite

Students must be able to program at an intermediate level in Python; students that do not meet this requirement will not be able to follow the course.

Students must also have knowledge of Selenium concepts equivalent to 'PT20158 Web Testing with Selenium 3: Concepts and Techniques'. It is recommended that students take PT20158 before this course.

#### Duration

Two Days

## Selenium 3 Web Driver Scripting with Python

---

### Course Outline

- I. ***The Selenium Python Binding***
  - A. The Web Driver API and architecture
  - B. Setting up the Selenium environment for Python
  - C. Basic scripting workflows
  - D. Web Driver command categories
  - E. Advantages of using scripting over the Selenium IDE
  - F. Selenium 3 differences from previous versions
  - G. Review of the Web Driver architecture
- II. ***Web Driver Navigation Commands***
  - A. Page navigation commands
  - B. Working with windows and frames
  - C. Working with pop-ups and alerts
  - D. Managing HTTPS
  - E. Working with cookies
  - F. Using implicit and explicit waits and timeouts
- III. ***Locating Elements***
  - A. Concept of selectors: single return value versus multiple return values
  - B. How the "by" clause functions
  - C. Selecting by HTML attributes
  - D. Using CSS selectors
  - E. Using XPATH selectors
  - F. Deprecated selectors
  - G. Comparison of selector efficiency
- IV. ***Working with Widgets***
  - A. Reading and writing text based widgets
  - B. Manipulating widget attributes
  - C. Working with visibility and other issues
  - D. Radio buttons, check boxes and drop down lists
  - E. Working with IFRAMES
  - F. Working with forms and form submissions
  - G. Executing JavaScript
- V. ***Mouse Actions and Listeners***
  - A. Moving the mouse
  - B. Click & hold, drag & drop operations
  - C. The Builder technique
  - D. Implementing a Web Driver event listener
  - E. Event types: navigation and element change events
  - F. Listening for script execution and exceptions
  - G. Using multiple listeners
- VI. ***The Page Object Pattern***
  - A. Encapsulating Web pages – insulation Selenium code from HTML changes
  - B. Creating and using page objects
  - C. Decoupling test logic from imperative code
    - D. Best practices for using the page object pattern
    - E. Page object factories
- VII. ***Selenium Integration***
  - A. Integrating Selenium with testunit
  - B. Integrating Selenium with PyTest
  - C. Integrating Selenium with Behave
- VIII. ***Advanced Topics***
  - A. Deep dive into working with AJAX
  - B. Data driven testing
  - C. Dealing with file system I/O
  - D. Taking screen shots
  - E. Recording videos
  - F. Reporting tools
  - G. HTML5 canvas and video elements
  - H. HTML5 sessions and local storage
- IX. ***Best Practices***
  - A. Best practices for writing Python Web Driver code
  - B. Common pitfalls and mistakes
  - C. Migrating older Selenium Python code to Selenium 3

- D. Best practices for the workflow of developing Python scripts