

QA

ADVANCED

4 DAYS

UI Automation Testing with Selenium (Python)

Build a strong foundation in UI automation testing using Selenium with Python by mastering core concepts and environment setup. Learn how to write robust automation scripts, handle dynamic elements, apply advanced Selenium features, and design scalable, maintainable test frameworks.

Why Selenium (Python)?

Faster and Efficient Testing

Automation allows you to execute repetitive UI tests quickly and accurately, saving valuable time and effort.

Improved Test Coverage

UI automation testing with Selenium enables you to cover a broader range of scenarios and test cases by automating repetitive and tedious tasks. This ensures a higher level of test coverage, reducing the chances of critical defects slipping through.

Consistent and Reliable Results

By using Selenium with Python, you can create reliable UI automation scripts that deliver consistent results across different environments and platforms.

What you'll be able to do

- ✓ Gain a solid understanding of UI automation testing concepts, the role of Selenium, and the benefits of using Python for automation.
- ✓ Get familiarised with the process of setting up the Selenium environment.
- ✓ Get insights on Web Element Identification and Interaction.
- ✓ Develop proficiency in writing effective automation scripts using Selenium with Python.
- ✓ Learn how to handle dynamic web elements and synchronisation issues.
- ✓ Explore advanced Selenium features.
- ✓ Learn to design scalable and maintainable test frameworks.

Who this is for

- Software testers
- Quality assurance (QA) professionals
- Test automation engineers
- Developers
- Agile Development Teams

Curriculum

01 Foundations

- Apply foundations using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

02 Environment Setup

- Apply environment setup using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

03 Locators and Interactions

- Apply locators and interactions using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

04 Waits and Dynamic Elements

- Apply waits and dynamic elements using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

05 Alerts, Frames, and Windows

- Apply alerts, frames, and windows using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

06 POM and Design Patterns

- Apply pom and design patterns using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

07 Data-Driven Tests

- Apply data-driven tests using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

08 Frameworks and Reporting

- Apply frameworks and reporting using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

09 Cross-Browser and Parallel

- Apply cross-browser and parallel using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

10 CI/CD Integration

- Apply ci/cd integration using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

11 Debugging and Best Practices

- Apply debugging and best practices using Selenium (Python) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks