# **Advanced Selenium:**

# **Module – 2 (Selenium IDE).**

1. **What is Automation Testing?**

* Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps. Automation testing is both practical and cost-effective. It involves automating the testing process and the management and application of test data and results to improve software. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

1. **Which Are the Browsers Supported by Selenium Ide?**

* Google Chrome

Mozilla Firefox

Microsoft Edge (Legacy version, based on the Edge HTML engine)

Safari

1. **What are the benefits of Automation Testing?**

* Automation testing offers several benefits, which contribute to improved software development processes and the overall quality of the product. Here are some key advantages of automation testing:
* Faster and Efficient Testing: Automated tests can execute repetitive tasks and perform complex calculations much faster than manual testing. This leads to quicker test execution, allowing for faster feedback on the software's quality. Additionally, automation enables parallel testing, where multiple tests can be run simultaneously, reducing overall testing time.
* Increased Test Coverage: Automation allows for extensive test coverage by executing a large number of tests with different input combinations, configurations, and data sets. It helps identify defects, errors, and inconsistencies that might be missed during manual testing. With broader coverage, the automation testing process can ensure higher software quality.
* Improved Accuracy: Automation eliminates the possibility of human errors or inconsistencies during test execution, ensuring accurate and reliable results. Automated tests follow predefined steps precisely and compare actual results against expected outcomes, reducing the chances of overlooking defects or false positives.
* Reusability and Scalability: Automated test scripts can be reused across different versions or iterations of the software, saving time and effort in test creation. As the application evolves, existing test cases can be easily adapted and extended, making automation testing scalable and flexible.
* Cost and Time Savings: Although initial setup and maintenance of automation frameworks require some investment, in the long run, automation testing proves cost-effective. The reduction in manual effort, increased test coverage, and faster execution contribute to significant time savings and ultimately reduce project costs.
* Continuous Integration and Delivery: Automation testing fits well within the continuous integration and continuous delivery (CI/CD) pipelines. Automated tests can be integrated with build systems, allowing for regular and frequent execution of tests in an automated manner. This facilitates the early detection of issues and faster feedback, ensuring that software updates are delivered quickly and reliably.
* Regression Testing: Automation is particularly valuable for regression testing, which involves retesting existing functionality to ensure that changes or additions to the software haven't introduced new bugs or broken existing features. Automated regression tests can be executed quickly and repeatedly, allowing for efficient verification of system stability and preventing regression issues.

1. **What are the advantages of Selenium?**

* Selenium is a popular open-source automation tool widely used for testing web applications. It offers several advantages that make it a preferred choice for web testing and automation. Here are some of the key advantages of Selenium:
* Cross-browser and cross-platform compatibility: Selenium supports multiple web browsers such as Chrome, Firefox, Safari, Internet Explorer, and Opera. It also works on various operating systems like Windows, macOS, Linux, and more. This cross-browser and cross-platform compatibility ensure that your web application behaves consistently across different environments.
* Language and framework support: Selenium supports multiple programming languages such as Java, C#, Python, Ruby, and JavaScript. This allows you to choose a language that you are comfortable with or that aligns with your existing development stack. Selenium can be integrated with various testing frameworks like TestNG and JUnit, making it flexible and adaptable to different testing needs.
* Wide range of testing capabilities: Selenium offers a rich set of features for automating web testing. It allows you to interact with web elements, simulate user actions like clicking buttons, entering text, selecting options, and more. You can also perform advanced actions such as handling pop-ups, alerts, frames, and multiple windows. Selenium supports automated testing of complex scenarios and workflows, enabling comprehensive testing of web applications.
* Integration with other tools and frameworks: Selenium can be easily integrated with other tools and frameworks, making it highly extensible. It can be combined with testing frameworks like TestNG or JUnit for test management and reporting. It can also be integrated with build tools like Maven or Gradle for continuous integration and deployment processes. Furthermore, Selenium can be integrated with popular frameworks like Cucumber or Spec Flow for behaviour-driven development (BDD) testing.
* Large and active community: Selenium has a large and active community of developers and testers. This means you can find ample resources, documentation, tutorials, and support forums to help you get started and resolve any issues you may encounter. The community also contributes to the development and improvement of Selenium, ensuring that it stays up-to-date with the latest web technologies and best practices.
* Cost-effective and open source: Selenium is an open-source tool, which means it is freely available and can be customized according to your requirements. This makes it a cost-effective solution compared to proprietary automation tools. Additionally, the open-source nature of Selenium ensures transparency, allowing users to inspect the source code and contribute to its development.
* Overall, the advantages of Selenium, such as cross-browser compatibility, language support, extensive testing capabilities, integration options, community support, and cost-effectiveness, make it a powerful and popular choice for web application testing and automation.

1. **Why testers should opt for Selenium and not QTP?**

* Testers often choose Selenium over QTP (now known as UFT) for several reasons. Here are some key advantages of Selenium that make it a popular choice:
* Open-source and cost-effective: Selenium is an open-source tool, which means it is freely available for use. Testers and organizations can save on licensing costs compared to QTP, which is a commercial tool requiring a license. This makes Selenium an attractive option, particularly for small teams or projects with budget constraints.
* Language and platform support: Selenium supports multiple programming languages like Java, C#, Python, Ruby, and more. Testers can choose the language they are most comfortable with or that aligns with their project requirements. QTP primarily supports VBScript, which may limit the tester's language flexibility.
* Cross-browser and cross-platform compatibility: Selenium allows testers to automate web applications across different browsers like Chrome, Firefox, Safari, and Internet Explorer. It also supports multiple operating systems like Windows, macOS, and Linux. QTP has limitations in terms of cross-browser and cross-platform support.
* Robust community and extensive documentation: Selenium has a large and active community of developers and testers who actively contribute to its development and provide support. There are numerous online resources, forums, and documentation available, making it easier to find solutions to problems or seek guidance. QTP has a smaller community and may have limited resources available in comparison.
* Integration with various tools and frameworks: Selenium integrates well with popular tools and frameworks like TestNG, JUnit, Maven, Jenkins, and more. This enables testers to leverage existing tools and frameworks for test management, reporting, continuous integration, and automation infrastructure. QTP may have limited integrations and might require additional effort for seamless integration with other tools.

1. **To validate the tops technologies website Contact us page and enter your friend detail at last “Guest Call Back” https://www.topsint.com/contact-us/ 1. To use assert 2. To use click and wait 3. To use locators 4. To use other command.**

* ****