UI Automation - Selenium Questions

**What is Selenese?**

Selenese is the language which is used to write test scripts in Selenium IDE.

**What is Selenium? What are the different Selenium components?**

Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web-based applications and a wide range of browsers and platforms. Due to its existence in the open source community, it has become one of the most accepted tools amongst the testing professionals.

Selenium is not just a single tool or a utility, rather a package of several testing tools and for the same reason, it is referred to as a Suite. Each of these tools is designed to cater different testing and test environment requirements.

The suite package constitutes the following sets of tools:

* [**Selenium Integrated Development Environment (IDE)**](https://www.softwaretestinghelp.com/selenium-ide-download-and-installation-selenium-tutorial-2/) – Selenium IDE is a record and playback tool. It is distributed as a Firefox Plugin.
* **Selenium Remote Control (RC)** – Selenium RC is a server that allows a user to create test scripts in the desired programming language. It also allows executing test scripts within the large spectrum of browsers.
* [**Selenium WebDriver**](https://www.softwaretestinghelp.com/selenium-webdriver-selenium-tutorial-8/) – WebDriver is a different tool altogether that has various advantages over Selenium RC. WebDriver directly communicates with the web browser and uses its native compatibility to automate.
* [**Selenium Grid**](https://www.softwaretestinghelp.com/selenium-grid-selenium-tutorial-29/) – Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

**What is Locator?**

The locator can be termed as an address that identifies a web element uniquely within the webpage. Locators are the HTML properties of a web element which tells the Selenium about the web element it needs to perform the action on.

There is a diverse range of web elements. **The most common amongst them are:**

* Text box
* Button
* Drop Down
* Hyperlink
* Check Box
* Radio Button

**Types of Locators**

Identifying these elements has always been a very tricky subject and thus it requires an accurate and effective approach. Thereby, we can assert that more effective the locator, more stable will be the automation script. Essentially every Selenium command requires locators to find the web elements. Thus, to identify these web elements accurately and precisely we have different types of locators.

[A screenshot of a cell phone

Description automatically generated](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Types-of-Locators-in-Selenium-1.jpg)

**What is the difference between assert and verify commands?**

**Assert:** Assert command checks whether the given condition is true or false. Let’s say we assert whether the given element is present on the web page or not. If the condition is true then the program control will execute the next test step but if the condition is false, the execution would stop and no further test would be executed.

**Verify:** Verify command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn’t halt i.e. any failure during verification would not stop the execution and all the test steps would be executed.

**What is an XPath?**

[XPath](https://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/) is used to locate a web element based on its XML path. XML stands for Extensible Markup Language and is used to store, organize and transport arbitrary data. It stores data in a key-value pair which is very much similar to HTML tags. Both being markup languages and since they fall under the same umbrella, XPath can be used to locate HTML elements.

The fundamental behind locating elements using XPath is the traversing between various elements across the entire page and thus enabling a user to find an element with the reference of another element.

**What is the difference between “/” and “//” in Xpath?**

**Single Slash “/” –** Single slash is used to create Xpath with absolute path i.e. the xpath would be created to start selection from the document node/start node.

**Double Slash “//” –** Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the document.

**What is Same origin policy and how it can be handled?**

The problem of same origin policy disallows to access the DOM of a document from an origin that is different from the origin we are trying to access the document.

Origin is a sequential combination of scheme, host, and port of the URL. For example, for a URL https://www.softwaretestinghelp.com/resources/, the origin is a combination of http, softwaretestinghelp.com, 80 correspondingly.

Thus the Selenium Core (JavaScript Program) cannot access the elements from an origin that is different from where it was launched. For Example, if I have launched the JavaScript Program from “https://www.softwaretestinghelp.com”, then I would be able to access the pages within the same domain such as “https://www.softwaretestinghelp.com/resources” or “https://www.softwaretestinghelp.com/istqb-free-updates/”. The other domains like google.com, seleniumhq.org would no more be accessible.

So, In order to handle the same origin policy, Selenium Remote Control was introduced.

**When should I use Selenium Grid?**

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

**What do we mean by Selenium 1 and Selenium 2?**

Selenium RC and WebDriver, in a combination, are popularly known as Selenium 2. Selenium RC alone is also referred to as Selenium 1.

**How do I launch the browser using WebDriver?**

The following syntax can be used to launch Browser: *WebDriver driver = new FirefoxDriver();* *WebDriver driver = new ChromeDriver();* *WebDriver driver = new InternetExplorerDriver();*

**What are the different types of waits available in WebDriver?**

There are two types:

1. Implicit Wait
2. Explicit Wait

**Implicit Wait:** Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, the subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

**Explicit Wait:** Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

**What are the different types of frameworks?**

**Below are the different types of frameworks:**

1. **Module Based Testing Framework:** The framework divides the entire “Application Under Test” into the number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts have taken together builds a larger test script representing more than one module.
2. **Library Architecture Testing Framework:** The basic fundamental behind the framework is to determine the common steps and group them into functions under a library and call those functions in the test scripts whenever required.
3. Data Driven Testing Framework: Data Driven Testing Framework helps the user segregate the test script logic and the test data from each other. It lets the user store the test data into an external database. The data is conventionally stored in “Key-Value” pairs. Thus, the key can be used to access and populate the data within the test scripts.
4. **Keyword Driven Testing Framework:** The Keyword Driven testing framework is an extension to Data-driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file.
5. **Hybrid Testing Framework:** Hybrid Testing Framework is a combination of more than one above mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.
6. **Behavior Driven Development Framework:** Behavior Driven Development framework allows automation of functional validations in an easily readable and understandable format to Business Analysts, Developers, Testers, etc.

**What is a framework?**

**Answer:** A framework is a set of the structure of the entire automation suite. It is also a guideline, which if followed can result in a structure that is easy to maintain and enhance.

**These guidelines include:**

* Coding standards
* Handling the test data
* Maintaining and handling the elements (object repository in QTP)
* Handling of environment files and properties file
* Reporting of data
* Handling logs

**What are the attributes of a good framework?**

**Answer: The characteristics include:**

* **Modular:** The framework should be adaptable to change. Testers should be able to modify the scripts as per the environment or login information change.
* **Reusable:** The commonly used methods or utilities should be written in a common file that is accessible to all the scripts.
* **Consistent:** The suite should be written in a consistent format by following all the accepted coding practices.
* **Independent:** The scripts should be written in such a way that they are independent of each other. In case one test fails, it should not hold back the remaining test cases (unless it is a login page)
* **Logger:** It is good to have implemented the logging feature in the framework. This would help in case our scripts run for longer hours (say nightly mode), if the script fails at any point of time, having the log file will help us to detect the location along with the type of the error.
* **Reporting:** It is good to have the reporting feature automatically embedded into the framework. Once the scripting is done, we can have the results and reports sent via email.
* **Integration:** Automation Framework should be such that it is easy to integrate with other applications like continuous integration or triggering the automated script as soon as the build is deployed.

**What factors determine the effectiveness of Automation testing?**

**Answer: Some of the factors that determine the effectiveness of automation testing are:**

* Time saved by running scripts over the manual execution of test cases.
* Defects found
* Test Coverage or code coverage
* Maintenance time or development time
* Stability of the scripts
* Test Reusability
* Quality of the software under test