Beginner-Level Selenium Interview Questions for 2025

**1. What is Selenium?**

Selenium is an open-source tool used as a free automation testing suite of tools. Licensing is not required and has more benefits than other testing tools. The tests can be done on any operating system, such as Mac, Linux, and Windows. Selenium can be combined with some tools, such as TestNG in Selenium and JUnit, to manage test cases and generate reports.

**2. What are the Selenium suite components?**

Selenium IDE

It is a Firefox/Chrome plug-in that was developed to speed up the creation of automation scripts. It records the user actions on the web browser and exports them as a reusable script.

Selenium Remote Control (RC)

RC server allows users to write application tests in various programming languages. The commands from the test script are accepted by this server and are sent to the browser as Selenium core JavaScript commands. The browser then behaves accordingly.

Selenium WebDriver

WebDriver is a programming interface that helps create and run test cases. It makes provision to act on web elements. Unlike RC, WebDriver does not require an additional server and interacts natively with the browser applications.

Selenium Grid

The grid was designed to distribute commands to different machines simultaneously. It allows the parallel execution of tests on different browsers and operating systems. It is exceptionally flexible and is integrated with other suite components for simultaneous execution.

**3. Mention the advantages of using Selenium as an automation tool.**

Selenium is an automation tool with unique benefits that give it a competitive edge over others, such as open source, multi-language support, platform support, multi-browser support, framework availability and flexibility, reusability, and integrated and parallel test execution.

**4. What is test automation or automation testing?**

Test automation or automation testing is the process of using specialized software to control the execution of tests and compare the results with expected outcomes. Automation testing can help reduce the time, cost, and effort required to test software applications by automating repetitive tasks and allowing testers to focus on more critical test cases.

**5. What are the advantages of automation testing?**

There are many advantages of automation testing. Perhaps the most obvious is that it can save you time and effort. Automation testing can help speed up the testing process by automating repetitive tasks, such as running the same test cases multiple times or across different browsers.

Another big advantage is that automation testing can improve the accuracy of your tests. By automating the process, you can eliminate human error and ensure that your tests are always carried out the same way. This can be particularly important when testing complex applications with a greater risk of errors.

Finally, automation testing can also help improve your test coverage. By automating the testing process more, you can increase the number of run test cases and cover a larger range of functionality. This can help ensure that your software is thoroughly tested and bugs-free.

**6. What is Selenese? How is it classified?**

Selenese is the set of Selenium commands used to test your web application. The tester can test the broken links, the existence of some objects on the UI, Ajax functionality, alerts, windows, list options, and a lot more using Selenese.

Action: Commands which interact directly with the application

Accessors: Allow the user to store certain values in a user-defined variable

Assertions: Verifies the current state of the application with an expected state

**7. What are the limitations of Selenium testing?**

Unavailability of reliable tech support: Since Selenium is an open-source tool, it does not have dedicated tech support to resolve user queries.

Tests web applications only: Selenium must be integrated with third-party tools like Appium and TestNG to test desktop and mobile applications.

Limited support for image testing.

No built-in reporting and test management facility: Selenium has to be integrated with tools like TestNG or JUnit, among others, to facilitate test reporting and management.

It may require knowledge of programming languages: Selenium WebDriver expects the user to have some basic programming knowledge.

**8. What is the difference between Selenium 2.0 and Selenium 3.0?**

Selenium 2.0 is a tool that makes the development of automated tests for web applications easier. It represents the merger of the original Selenium project with the WebDriver project. Selenium RC was deprecated since the merge but was used for backward compatibility.

Selenium 3.0 is the extended version of Selenium 2.0. It is inherently backward compatible and does not involve Selenium RC. The new version came along with several bug fixes and increased stability.

selenium3.

**9. What are the testing types supported by Selenium?**

Selenium supports Regression testing and Functional testing.

Regression testing is a full or partial selection of already executed test cases that are re-executed to ensure existing functionalities work fine.

The steps involved are:

Re-testing: All tests in the existing test suite are executed. It proves to be very expensive and time-consuming.

Regression test selection: Tests are classified as feature, integration, and end-to-end tests. In this step, some of the tests are selected.

Prioritization of test cases: The selected test cases are prioritized based on business impact and critical functionalities.

Functional testing involves verifying every function of the application with the required specifications.

The following are the steps involved:

Identify test input.

Compute test outcome.

Execute test.

Compare the test outcome with the actual outcome.

**10. What are the different types of annotations used in Selenium?**

Different types of annotations that are used in Selenium include:

@Test - This annotation is used to mark a method as a test method

@BeforeMethod - This annotation is used to execute a method before each test method

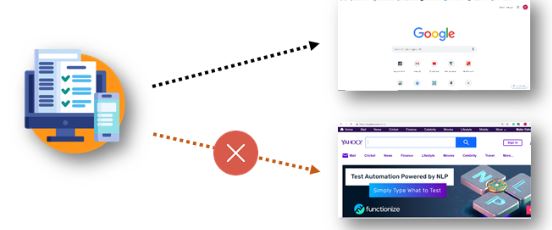
@AfterMethod - This annotation is used to execute a method after each test method

@BeforeClass - This annotation is used to execute a method before the first test method.

**11. What is the same-origin policy and how is it handled?**

The same-origin policy is a feature adopted for security purposes. According to this policy, a web browser allows scripts from one webpage to access the contents of another webpage, provided both pages have the same origin. The origin refers to a combination of the URL scheme, hostname, and port number.

The same Origin Policy prevents a malicious script on one page from accessing sensitive data on another webpage.



Consider a JavaScript program used by google.com. This test application can access all Google domain pages like google.com/login, google.com/mail, etc. However, it cannot access pages from other domains like yahoo.com

Selenium RC was introduced to address this. The server acts as a client-configured HTTP proxy and "tricks" the browser into believing that Selenium Core and the web application being tested come from the same origin.

**12. Mention the types of Web locators.**

Locator is a command that tells Selenium IDE which [GUI elements](https://www.simplilearn.com/tutorials/python-tutorial/python-graphical-user-interface-gui) ( say Text Box, Buttons, Check Boxes, etc) it needs to operate on. Locators specify the area of action.

Locator by ID: It takes a string parameter, which is a value of the ID attribute that returns the object to the findElement() method.

**driver.findElement(By.id(“user”));**

Locator by the link: If your targeted element is a link text, then you can use the by.linkText locator to locate that element.

**driver.findElement(By.linkText(“Today’s deals”)).click();**

Locator by Partial link: The target link can be located using a portion of text in a link text element.

**driver.findElement(By.linkText(“Service”)).click();**

Locator by Name: The first element with the name attribute value matching the location will be returned.

**driver.findElement(By.name(“books”).click());**

Locator by TagName: Locates all the elements with the matching tag name

**driver.findElement(By.tagName(“button”).click());**

Locator by classname: This finds elements based on the value of the CLASS attribute. If an element has many classes, then this will match against each of them.

**driver.findElement(By.className(“inputtext”));**

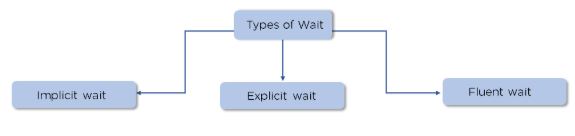
Locator by XPath: It takes a parameter of String, an XPATHEXPRESSION, and returns an object to findElement() method.

**driver.findElement(By.xpath(“//span[contains(text(),’an account’)]”)).getText();**

Locator by CSS Selector: Locates elements based on the driver’s underlying CSS selector engine.

**driver.findElement(By.cssSelector(“input#email”)).sendKeys(“myemail@email.com”);**

**13. What are the types of waits supported by WebDriver?**



Implicit wait commands Selenium to wait for a certain amount of time before throwing a “No such element” exception.

driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

Explicit wait is used to tell the Web Driver to wait for certain conditions before throwing an "ElementNotVisibleException" exception.

WebDriverWait wait = new WebDriverWait(WebDriver Reference, TimeOut);

Fluent wait is used to tell the web driver to wait for a condition and the frequency with which we want to check the condition before throwing an "ElementNotVisibleException" exception.

Wait wait = new FluentWait(WebDriver reference).withTimeout(timeout, SECONDS).pollingEvery(timeout, SECONDS).ignoring(Exception.class);

**14. Mention the types of navigation commands**

* driver.navigate().to("https://www.ebay.in/"); - Navigates to the provided URL
* driver.navigate().refresh(); - This method refreshes the current page
* driver.navigate().forward(); - This method does the same operation as clicking on the Forward Button of any browser. It neither accepts nor returns anything.
* driver.navigate().back(); - This method does the same operation as clicking on the Back Button of any browser. It neither accepts nor returns anything.

**15. What is the major difference between driver.close() and driver.quit()?**

driver.close()

This command closes the browser’s current window. If multiple windows are open, the current focus window will be closed.

driver.quit()

When quit() is called on the driver instance and one or more browser windows are open, it closes all the open browser windows.

**16. What makes Selenium such a widely used testing tool? Give reasons.**

1. Selenium is easy to use since it’s essentially developed in JavaScript.
2. Selenium can test web applications against browsers like Firefox, Opera, Chrome, and Safari, to name a few.
3. The test code can be written in various programming languages like Java, Perl, Python, and PHP.
4. Selenium is platform-independent and can be deployed on different Operating systems like Windows, Linux, and Macintosh.
5. Selenium can be integrated with third-party tools like JUnit and TestNG for test management.

**17. Why is it advised to select Selenium as a web application or system testing tool?**

1. Selenium is an open-source, portable tool and freeware.
2. It supports many operating systems like Linux, Unix, Macintosh and Windows.
3. Selenium supports Chrome, opera, safari, and Internet Explorer.
4. It supports many languages like Perl, Java, Python, Ruby, Groovy, JavaScript, VB Script, etc.
5. Selenium can be used for iPhone, Android, and Blackberry based on application testing.
6. It helps integrate it with ANT or Maven framework for source code compilation.
7. It requires less CPU and RAM consumption for script execution.

Relevant Read: [How to Read Selenium With Java?](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-with-java)

**18. What is an exception test in Selenium?**

The exception test in Selenium is an exception that you expect will be thrown inside a test class. If you have written a test case so that it should be thrown as an exception, then you can use the test annotation and specify the exception in the parameters.

**19. How to wait until a web page has been loaded completely in Selenium?**

One approach is to use the "implicit wait" command in Selenium, which instructs the web driver to wait a certain amount of time before throwing an error if the element is not found or loaded. Another option is to use the "explicit wait" command for a specific element to appear on the page before proceeding with the script.

**20. What is Selenium WebDriver?**

Selenium WebDriver is a popular tool for automating web browsers and has a program interface.