

# Beginner Level Selenium Interview Questions

## 1. What is Selenium?

Selenium is an open-source tool used as a free automation testing suite of tools. Licensing is not required, and it has more benefits than other testing tools. The tests can be done in any operating system like Mac, Linux and Windows. Selenium can be combined with some tools, such as TESTNG in Selenium and JUNIT, for managing test cases and generating 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 is a server that 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 different 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 and has some 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?

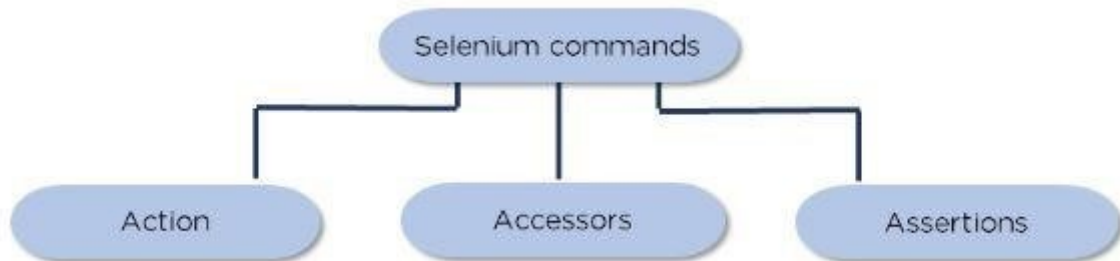
There are many advantages of automation testing. Perhaps the most obvious is that it can save you a lot of time and effort. Automation testing can help to speed up the process of testing by automating repetitive tasks, such as running the same test cases multiple times or across different browsers.

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 where there is a greater risk of errors.

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

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

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



Action: Commands which interact directly with the application

Accessors: Allow the user to store certain values to 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 the user queries.
- Tests web applications only: Selenium needs to be integrated with thirdparty 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.
- May require the knowledge of programming languages: Selenium WebDriver expects the user to have some basic knowledge about programming.

## 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 got deprecated since the merge, however, 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.



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

Selenium supports Regression testing and Functional testing.

Regression testing - It 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 -

1. Re-testing: All tests in the existing test suite are executed. It proves to be very expensive and time-consuming.
2. Regression test selection: Tests are classified as feature tests, integration tests, and the end to end tests. In this step, some of the tests are selected.
3. Prioritization of test cases: The selected test cases are prioritized based on business impact and critical functionalities.

Functional testing - Functional Testing involves the verification of every function of the application with the required specification.

The following are the steps involved:

1. Identify test input
2. Compute test outcome
3. Execute test
4. Compare the test outcome with the actual outcome

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

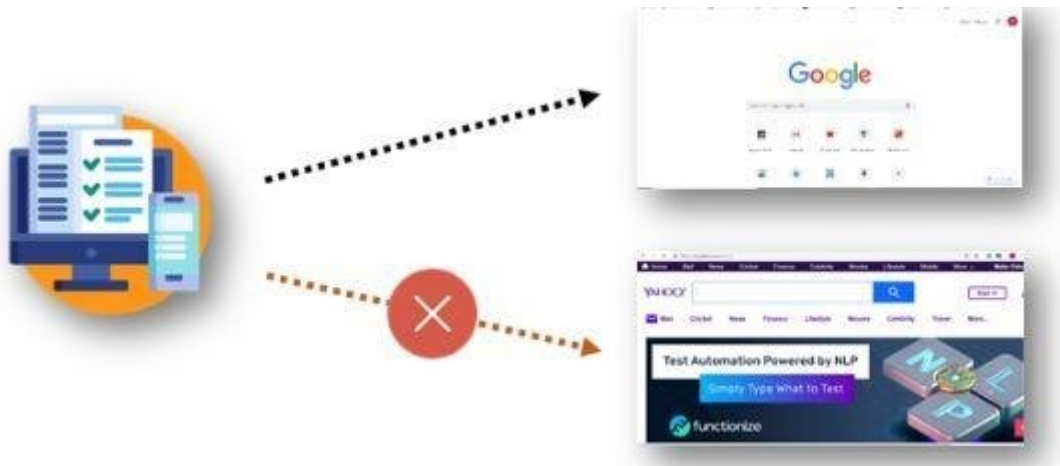
Different types of annotations that are used in Selenium include:

- `@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?

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 the 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 to access 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 ( 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 which returns the object to 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"));
```

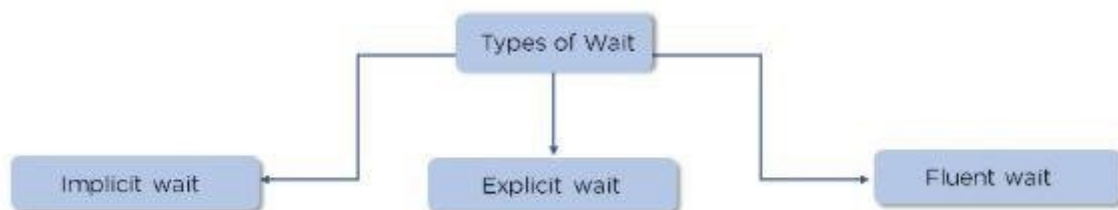
Locator by XPath: It takes a parameter of String which is a XPATH EXPRESSION and it returns an object to findElement() method.

```
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 - 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 - 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 - It is used to tell the web driver to wait for a condition, as well as 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 window of focus will be closed.

```
driver.quit()
```



When quit() is called on the driver instance and there are one or more browser windows 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 testing tool for web applications or systems?

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.

## 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 in such a way 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 to wait 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.

## 21. Is Selenium WebDriver a library?

No, Selenium WebDriver is not a library.

## 22. Which browsers/drivers are supported by Selenium Webdriver?

Selenium WebDriver supports several popular web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and Opera.

## 23. Explain Selenium 4 and why it is different from other Selenium versions?

Selenium 4 is the latest version of the Selenium web testing framework, which introduces several new features and improvements over previous versions.

## 24. What will happen if I execute this command? `driver.get`

This command will instruct the Selenium WebDriver to open the specified URL in the web browser.

## 25. What is an alternative option to `driver.get()` method to open an URL in Selenium Web Driver?

An alternative option to the `driver.get()` method to open an URL in Selenium Web Driver is to use the `driver.navigate().to()` method.

## 26. Is it possible to test APIs or web services using Selenium Webdriver?

No, we cannot test APIs or web services using Selenium Webdriver as it is designed to test web-based applications.

## 27. Mention different ways of locating an element in Selenium?

The various ways of locating an element in Selenium are: by ID, by name, by classname, by tagname, by link text, by partial link text, by CSS selector, and by XPath.

## 28. How can we move to the nth-child element using XPath?

We can move to the nth-child element using XPath by using the following expression:  
`(//parent-element/*)[n]`.

## 29. How can we type text in a textbox using Selenium?

To type text in a textbox using Selenium, first, we need to locate the textbox using any of the locating strategies in Selenium, and then use the `sendKeys()` method to enter text into the textbox.

## Intermediate Level Selenium Interview Questions

### 30. How to type text in an input box using Selenium?

`sendKeys()` is the method used to type text in input boxes

Consider the following example -

`WebElement email = driver.findElement(By.id("email"));` - Finds the "email" text using the ID locator

`email.sendKeys("abcd.efgh@gmail.com");` - Enters text into the URL field

`WebElement password = driver.findElement(By.id("Password"));` - Finds the "password" text using the ID locator

`password.sendKeys("abcdefgh123");` - Enters text into the password field

### 31. How to click on a hyperlink in Selenium?

`driver.findElement(By.linkText("Today's deals")).click();`

The command finds the element using link text and then clicks on that element, where after the user would be redirected to the corresponding page.

`driver.findElement(By.partialLinkText("Service")).click();`

The above command finds the element based on the substring of the link provided in the parenthesis and thus `partialLinkText()` finds the web element.

## 32. How to scroll down a page using JavaScript?

scrollBy() method is used to scroll down the webpage

General syntax: `executeScript("window.scrollBy(x-pixels,y-pixels)");`

First, create a JavaScript object

`JavascriptExecutor js = (JavascriptExecutor) driver;` Launch the  
desired application

`driver.get("https://www.amazon.com");` Scroll  
down to the desired location

`js.executeScript("window.scrollBy(0,1000)");`

The window is not scrolled vertically by 1000 pixels

## 33. How to assert the title of a webpage?

Get the title of the webpage and store in a variable

`String actualTitle = driver.getTitle();`

Type in the expected title

`String expectedTitle = "abcdefgh";` Verify if  
both of them are equal

`if(actualTitle.equalsIgnoreCase(expectedTitle))`

```
System.out.println("Title Matched");
```

else

```
System.out.println("Title didn't match");
```

Alternatively,

```
Assert.assertEquals(actualTitle, expectedTitle);
```

## 34. How to mouse hover over a web element?

Actions class utility is used to hover over a web element in Selenium WebDriver. Instantiate

Actions class.

```
Actions action = new Actions(driver);
```

In this scenario, we hover over search box of a website

```
actions.moveToElement(driver.findElement(By.id("id of the searchbox"))).perform();
```

## 35. How to retrieve CSS properties of an element?

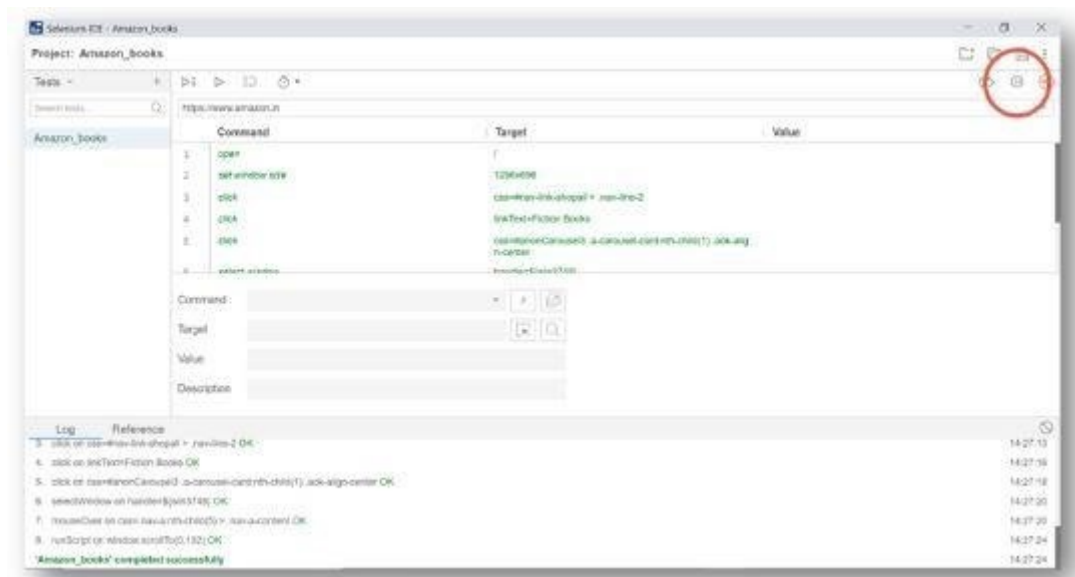
getCssValue() method is used to retrieve [CSS properties](#) of any web element

General Syntax: `driver.findElement(By.id("id")).getCssValue("name of css attribute");`

Example:

```
driver.findElement(By.id("email")).getCssValue("font-size");
```

## 36. What is POM (Page Object Model)?



Every webpage of the application has a corresponding page class that is responsible for locating the web elements and performing actions on them. Page Object Model is a design pattern that helps create object repositories for the web elements. POM improves code reusability and readability. Multiple test cases can be run on the object repository.

### 37. Can Captcha be automated?

No, Selenium cannot automate Captcha. Well, the whole concept of Captcha is to ensure that bots and automated programs don't access sensitive information - which is why, Selenium cannot automate it. The automation test engineer has to manually type the captcha while other fields can be filled automatically.

### 38. How does Selenium handle Windows-based popups?

Selenium was designed to handle web applications. Windows-based features are not natively supported by Selenium. However, third-party tools like AutoIT, Robot, etc can be integrated with Selenium to handle pop-ups and other Windows-based features.

### 39. How to take screenshots in WebDriver?

TakeScreenshot interface can be used to take screenshots in WebDriver. `getScreenshotAs()`

method can be used to save the screenshot

```
File scrFile = ((TakeScreenshot)driver).getScreenshotAs(outputType.FILE);
```

## 40. Why do testers choose Selenium over QTP?

Selenium is an open-source and free automation testing tool, while QTP is a licensed tool that requires a considerable investment. Selenium supports multiple programming languages and is cross-platform compatible.

## 41. What are the data-driven framework and keyworddriven framework?

A data-driven framework is a test automation framework that separates the test data from the test script. It allows testers to write automated tests in a way that is independent of the test data.

## 42. What is the difference between `getWindowHandles()` and `getWindowhandle()`?

`getWindowhandle()` method returns the unique identifier of the current browser window, while `getWindowHandles()` method returns a set of unique identifiers of all the browser windows opened by WebDriver.

## 43. What is a Selenium Maven project?

A Selenium Maven project is a software project that uses Maven to manage the project's dependencies and build process.

## 44. What is an Object Repository?



An Object Repository is a centralized location where testers can store all the web elements, such as buttons, text boxes, and links, used in the test automation framework.

## 45. What is exactly meant by a WebElement in Selenium, and how is it used?

The WebElement is an interface in Selenium which is used to represent an HTML element on a web page. It provides methods to interact with the web elements, such as clicking, entering text, and getting the value. It is used to identify and manipulate the web elements in the automation testing process.

## Advanced Level Selenium Interview Questions

### 46. Is there a way to type in a textbox without using sendKeys()?

Yes! Text can be entered into a textbox using JavaScriptExecutor JavascriptExecutor jse =

(JavascriptExecutor) driver;

```
jse.executeScript("document.getElementById('email').value='abc.efg@xyz.com'");
```

### 47. How to select a value from a dropdown in Selenium WebDriver?

Select class in WebDriver is used for selecting and deselecting options in a dropdown.

The objects of Select type can be initialized by passing the dropdown webElement as a parameter to its constructor.

```
WebElement testDrop = driver.findElement(By.id("testingDropdown"));
```

```
Select dropdown = new Select(testDrop);
```

WebDriver offers three ways to select from a dropdown: `selectByIndex`:

Selection based on index starting from 0

```
dropdown.selectByIndex(5);
```

`selectByValue`: Selection based on value

```
dropdown.selectByValue("Books");
```

`selectByVisibleText`: Selection of option that displays text matching the given argument

```
dropdown.selectByVisibleText("The Alchemist");
```



## 48. What does the `switchTo()` command do?

`switchTo()` command is used to switch between windows, frames or pop-ups within the application. Every window instantiated by the WebDriver is given a unique alphanumeric value called "Window Handle".

Get the window handle of the window you wish to switch to

```
String handle= driver.getWindowHandle(); Switch to
```

the desired window

```
driver.switchTo().window(handle); Alternatively
```

```
for(String handle= driver.getWindowHandles())
```

```
{ driver.switchTo().window(handle); }
```

## 49. How to upload a file in Selenium WebDriver?

You can achieve this by using `sendKeys()` or `Robot` class method. Locate the text box and set the file path using `sendKeys()` and click on submit button

Locate the browse button

```
WebElement browse =driver.findElement(By.id("uploadfile")); Pass the
```

path of the file to be uploaded using `sendKeys` method

```
browse.sendKeys("D:\\SeleniumInterview\\UploadFile.txt");
```

## 50. How to set browser window size in Selenium?

The window size can be maximized, set or resized To

maximize the window

```
driver.manage().window().maximize();
```

To set the window size

```
Dimension d = new Dimension(400,600);
```

```
driver.manage().window().setSize(d);
```

Alternatively,

The window size can be reset using JavaScriptExecutor

```
((JavascriptExecutor)driver).executeScript("window.resizeTo(1024, 768)");
```

## 51. When do we use findElement() and findElements()?

findElement() is used to access any single element on the web page. It returns the object of the first matching element of the specified locator.

General syntax:

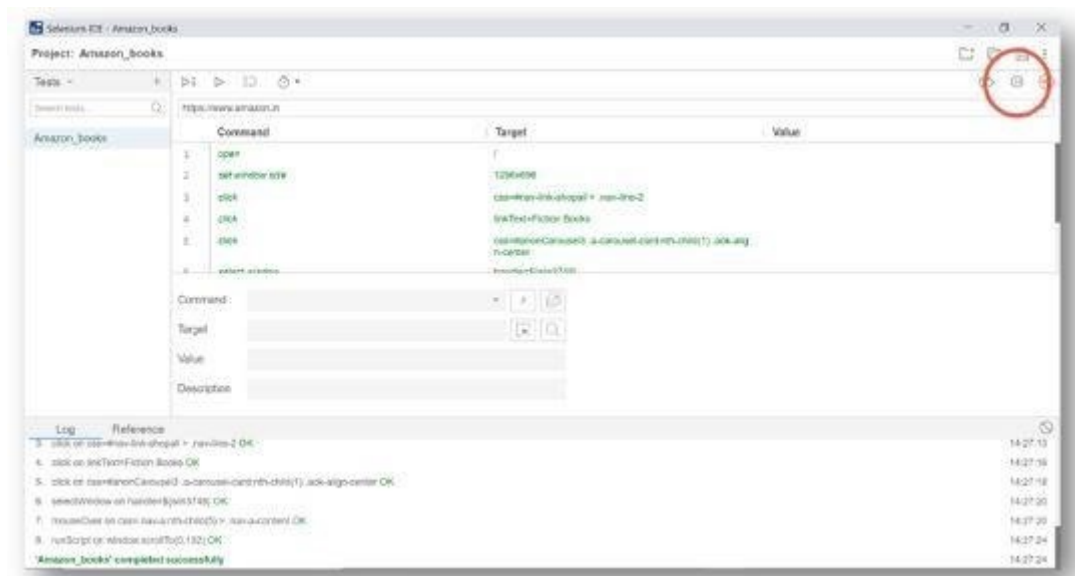
```
WebElement element = driver.findElement(By.id(example));
```

findElements() is used to find all the elements in the current web page matching the specified locator value. All the matching elements would be fetched and stored in the list of Web elements.

General syntax:

```
List <WebElement> elementList = driver.findElements(By.id(example));
```

## 52. What is a pause on an exception in Selenium IDE?



The user can use this feature to handle exceptions by clicking the pause icon on the top right corner of the IDE. When the script finds an exception it pauses at that particular statement and enters a debug mode. The entire test case does not fail and hence the user can rectify the error immediately.

### 53. How to login to any site if it is showing an Authentication Pop-Up for Username and Password?



To handle authentication pop-ups, verify its appearance and then handle them using an explicit wait command.

Use the explicit wait command

```
WebDriverWait wait = new WebDriverWait(driver, 10);
```

Alert class is used to verify the alert

```
Alert alert = wait.until(ExpectedConditions.alertIsPresent());
```

 Once

verified, provide the credentials

```
alert.authenticateUsing(new UserAndPassword(<username>, <password>));
```

### 54. What is the difference between single and double slash in Xpath?

Single slash is used to create Xpath with an absolute path i.e. the XPath would be created to start selection from the start node.

```
/html/body/div[2]/div[1]/div[1]/a
```

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

```
//div[class="qa-logo"]/a
```

## Top Selenium Interview Questions for Experienced

Here are some of the top Selenium interview questions and answers for experienced professionals which can help them prepare better for their interviews:

### 55. How do you find broken links in Selenium WebDriver?

When we use `driver.get()` method to navigate to a URL, it will respond with a status of 200-OK

200 – OK denotes that the link is working and it has been obtained. If any other status is obtained, then it is an indication that the link is broken.

Some of the HTTP status codes are :

- 200 – valid Link
- 404 – Link Not Found
- 400 – Bad Request
- 401 – Unauthorized
- 500 – Internal error

As a starter, obtain the links from the web application, and then individually get their status.

Navigate to the interested webpage for e.g. [www.amazon.com](http://www.amazon.com)

Collect all the links from the webpage. All the links are associated with the Tag 'a'

```
List<WebElement> links = driver.findElements(By.tagName("a"));
```

Create a list of type WebElement to store all the Link elements in it.

```
for(int i=0; i<links.size(); i++) {
```

```
    WebElement element = links.get(i);
```

```
        String url=element.getAttribute("href");
```

```
        verifyLink(url); }
```

Now Create a Connection using URL object( i.e ., link)

```
URL link = new URL(urlLink);
```

Connect using Connect Method

```
URLConnection httpConn =(URLConnection)link.openConnection(); Use
```

getResponseCode () to get response code

```
if(httpConn.getResponseCode()!= 200)
```

Through exception, if any error occurred

```
System.out.println("Broken Link");
```

With that we have come to the end of the article Selenium interview questions.

56. Name some of the commonly used automation testing tools that are used for functional automation.

QTP, Test Complete, RFT, and Silk Test.

57. Name some of the commonly used automation testing tools that are used for non-functional automation.

- LoadRunner
- JMeter
- WebLoad
- Neoload
- Silk Performer
- HP Performance Center
- Gatling
- Apache

58. List out some of the automation tools which could be integrated with Selenium to achieve continuous testing.

Some of the automation tools which could be integrated with Selenium to achieve continuous testing are:

- Jenkins
- Travis CI
- CircleCI
- AWS CodePipeline
- Azure DevOps
- Bitbucket Pipelines



## 59. What do you mean by the assertion in Selenium?

An assertion is a method of testing whether a particular condition is true or false. In Selenium, assertions are used to verify the state of elements on a page or the results of an action. Assertions can be used to check for the presence or absence of an element, the value of an element, or the text of an element. Assertions can also be used to check that an element is visible or hidden.

Assertions are an important part of testing with Selenium, as they enable you to verify that the state of your application meets your expectations. Without assertions, it would be difficult to know whether or not your tests are actually passing or failing.

## 60. Explain the difference between assert and verify commands.

The assert command is used to check if the given condition is true or not. If the condition is true, then the execution of the program will continue. If the condition is false, then the execution of the program will stop.

The verify command is used to check if the given condition is true or not. If the condition is true, then the execution of the program will continue. If the condition is false, then the execution of the program will not stop, but an error message will be displayed.

## 61. What do you mean by XPath?

XPath is a language for addressing parts of an XML document. XSLT and other XML-related technologies use it to access data within XML documents. XPath can be used to navigate through elements and attributes in an XML document. XPath is a major element in the XSLT standard and is crucial for processing XML documents.

## 62. Explain XPath Absolute and XPath attributes.

XPath has two main types of expressions: absolute and relative. Absolute expressions always start with a forward slash (/), which indicates the root element of the document.

Relative expressions do not start with a forward slash, and are relative to the current context.

Attributes are another important part of XPath. Attributes are added to elements and can contain valuable information about that element. In order to access an attribute, you must use the at sign (@) followed by the attribute name.

## 63. What is the difference between "/" and "/" in XPath?

The difference between "/" and "/" in XPath is that "/" is used to select an element based on its absolute location, while "/" is used to select an element based on its relative location.

For example, if you want to select the first <p> element on a page, you would use "/p". If you want to select all <p> elements on a page, regardless of their location, you would use "//p".

## 64. What are the WebDriver supported Mobile Testing Drivers?

There are four main WebDriver-supported mobile testing drivers: AndroidDriver, iPhoneDriver, RemoteWebDriver, and Selendroid.

1. AndroidDriver is used to test Android applications. It can be used on real devices or emulators.
2. iPhoneDriver is used to test iOS applications. It can only be used on real devices.
3. RemoteWebDriver is used to test web applications on mobile devices. It can be used with a variety of browsers, including Safari, Chrome, and Firefox.
4. Selendroid is used to test Android applications. It can be used on real devices or emulators.

## 65. What is the difference between type keys and type commands?

There is a difference between type keys and type commands in computer programming. Type keys are specific characters that you type on the keyboard, while type commands are instructions given to the computer.

## 66. What is the difference between "type" and "typeAndWait" command?

The "type" command is used to enter text into a field on a web page. The "typeAndWait" command also enters text into a field, but it waits for the page to load before proceeding to the next command. This can be useful if you are unsure whether or not the text you entered will cause the page to refresh.

The main difference between the "type" and "typeAndWait" command is that the "type" command does not wait for the page to reload, while the "typeAndWait" command does. If you are unsure whether or not the text you are entering will cause the page to refresh, it is best to use the "typeAndWait" command. This way, you can be sure that the next command will not be executed until the page has finished loading.

## 67. What is the main disadvantage of implicit wait?

The main disadvantage of implicit wait is that it can slow down your tests. This is because, by default, the implicit wait time is set to zero. As such, if an element is not found immediately, your test will keep trying to find it for the duration of the implicit wait time. This can add a significant amount of time to your test suite. Another disadvantage of implicit wait is that it can cause your tests to fail if the element you are waiting for takes longer to appear than the implicit wait time. Finally, implicit wait can make your tests less reliable because they can introduce flakiness.

## 68. How can we launch different browsers in Selenium WebDriver?

We can launch different browsers in Selenium WebDriver using several methods. For example, we can use the `setWebDriver()` method to specify the path to the browser's executable file. Alternatively, we can use the `addCustomProfilePreference()` method to add a custom profile preference for the browser. Finally, we can use the `launchBrowser()` method to launch the browser.

## 69. Write a code snippet to launch Firefox browser in WebDriver.

```
public class FirefoxDriver {  
  
    private WebDriver driver;  
  
    public FirefoxDriver() {  
  
        this.driver = new FirefoxDriver();  
  
    }  
  
    public void get(String url) {  
  
        this.driver.get(url);  
  
    }  
}
```

Assuming that you have already downloaded and installed Firefox, the next thing you need to do is write a code snippet to launch Firefox browser in WebDriver. The code snippet for launching Firefox browser is given below:

```
driver = new FirefoxDriver();  
  
driver.get("http://www.google.com");
```

## 70. Write a code snippet to launch Chrome browser in WebDriver.

```
WebDriver driver = new ChromeDriver();

WebDriverWait wait = new WebDriverWait(driver, 30); String url

= "https://www.google.com";

driver.get(url);

wait.until(ExpectedConditions.titleContains("Google")); System.out.println("Page title

is: " + driver.getTitle());

driver.quit();
```

This code snippet will launch the Chrome browser, navigate to Google.com, and then print the page title to the console.

## 71. Write a code snippet to launch Internet Explorer browser in WebDriver.

To launch Internet Explorer browser in WebDriver, you can use the following code snippet:

```
WebDriver driver = new InternetExplorerDriver();

driver.get("http://www.google.com");
```

This will launch the Internet Explorer browser and navigate to the Google homepage.

---

## 72. How do you perform drag and drop operations in WebDriver?

When using WebDriver, you can perform drag and drop operations using the Actions class. This class has a number of methods that can be used to perform various actions, such as clicking, dragging, and dropping. In order to use the Actions class, you first need to instantiate it with a WebDriver instance:

```
Actions actions = new Actions(driver);
```

Once you have an Actions instance, you can use the `dragAndDrop()` method to perform a drag and drop operation. This method takes two WebElements as arguments: the element to drag, and the element to drop. For example, to drag an element with the id "draggable" and drop it on an element with the id "droppable", you would do the following:

```
WebElement draggable = driver.findElement(By.id("draggable")); WebElement  
  
droppable = driver.findElement(By.id("droppable"));  
  
actions.dragAndDrop(draggable, droppable).perform();
```

You can also use the `clickAndHold()` and `release()` methods to perform a drag and drop operation. The `clickAndHold()` method takes a WebElement as an argument and "grabs" it, while the `release()` method releases the element. For example:

```
actions.clickAndHold(draggable).release(droppable).perform();
```

You can also chain together multiple Actions methods to create more complex interactions. For example, the following code will first move to the draggable element, then click and hold it, move to the droppable element, and finally release it:

```
actions.moveToElement(draggable).clickAndHold().moveToElement(droppable).rele  
ase().perform();
```

## 73. What are the different methods to refresh a web page in WebDriver?

There are a couple of ways to refresh a web page in WebDriver. The most common method is to use the "Refresh" button in the browser toolbar. Alternatively, you can also use the keyboard shortcut for refresh, which is typically F5. Finally, you can also right-click on the page and select "Refresh" from the context menu. All of these methods will cause the page to reload and any changes that have been made will be lost.

If you want to refresh the page without losing any changes, you can use the "Reload" button in the browser toolbar. This will reload the page from the server without losing any changes that have been made. Alternatively, you can also use the keyboard shortcut for reload, which is typically Shift+F5. Finally, you can also rightclick on the page and select "Reload" from the context menu. All of these methods will cause the page to reload without losing any changes that have been made.

## 74. How to invoke an application in WebDriver?

When using WebDriver, you can launch applications by either calling the "get" method on the driver instance, or by using the "navigate" method. Applications can also be invoked using a third-party tool such as Selenium IDE. However, doing so requires that you have the URL of the application to be launched beforehand. When using the "get" method, you simply need to pass in the URL of the application as a string.

## 75. What are the benefits of automation testing?

Automation testing can be a great way to speed up the software testing process. By automating certain tests, you can save time and resources that would otherwise be spent on manual testing. Additionally, automation can help to improve the accuracy of your tests, as well as provide detailed reports that can help you to identify any areas that need further attention. Overall, automation testing can be a valuable tool for any software development team.

## 76. Is there an HtmlUnitDriver for .NET?

Yes.

## 77. How can you redirect browsing from a browser through some proxy?

There are a few ways to redirect browsing from a browser through some proxy. One way is to use a web proxy. Web proxies can be used to access websites that may be blocked by your network administrator. Another way to redirect browsing is to use a Virtual Private Network (VPN). VPNs can be used to encrypt your traffic and route it through a proxy server. Finally, you can use a browser extension to redirect your traffic. Browser extensions are useful if you want to bypass proxy servers that are configured in your network settings.

## 78. Explain the pause feature in Selenium IDE.

The pause feature in Selenium IDE allows the tester to add a pause between the execution of two commands. It is used to slow down the test execution to allow for better observation of the test execution flow.

## 79. How do you handle a frame in WebDriver?

To handle a frame in WebDriver, we use the `switchTo()` method.

## 80. Mention the types of listeners in TestNG

TestNG supports three types of listeners:

- Test listeners: It listens to the test execution and allows us to perform actions before or after a test case is executed.
- Suite listeners: It listens to the suite execution and allows us to perform actions before or after the suite is executed.
- Method listeners: It listens to the execution of individual test methods and allows us to perform actions before or after a test method is executed.



## 81. Mention important details of different types of frameworks and also regarding the connection of Selenium with Robot Framework

Different sets of frameworks are available which can be used with Selenium:

- **Data-Driven Framework:** It uses external data sources such as CSV or Excel files to drive the test execution.
- **Keyword-Driven Framework:** It uses specific keywords to execute test cases.
- **Hybrid Framework:** It is a combination of both Data-Driven and KeywordDriven frameworks.

Selenium can be connected with Robot Framework using the Selenium2Library. The Selenium2Library is a Robot Framework test library that allows us to control web browsers using Selenium WebDriver.

## 82. Mention details of the basic steps of Selenium testing and also mention which are the widely used commands via a practical application.

The basic steps of Selenium testing are:

- Launch a web browser
- Navigate to a web page
- Locate web elements
- Perform actions on web elements
- Verify the results

Some widely used Selenium commands are:

- `get()` method: to launch the web page
- `findElement()` method: to locate the web element

- `click()` method: to click on the web element
- `sendKeys()` method: to enter text into the web element
- `getText()` method: to retrieve the text from the web element

### 83. What are exactly Jenkins and mention the advantages of using it with Selenium?

These are servers to automate the software development process. Using Jenkins with Selenium provides the following benefits:

- Automatic triggering of Selenium tests when code changes are committed to the repository.
- Parallel execution of Selenium tests on multiple machines to reduce the execution time.
- Integration with other tools such as JIRA, GitHub, and Slack to provide notifications and status updates.

## Selenium Tricky Interview Questions

### 84. Explain the methods used to handle dynamic web elements using Selenium?

Dynamic web elements can be handled using different methods such as xpath, CSS selectors, and the Explicit wait mechanism in Selenium. You can use these methods to locate the dynamic elements and perform actions on them.

### 85. How do you deal with stale element exceptions in Selenium?

Stale element exceptions occur when the element you are interacting with is no longer attached to the DOM or has been modified. To handle this exception, you can refresh the page or try to locate the element again using a different locator strategy.

## 86. How do you simulate a browser back button click in Selenium?

You can simulate the browser back button click in Selenium using the `navigate().back()` method. This method will navigate back to the previous page in the browser history.

## 87. How do you handle alerts in Selenium?

Alerts can be handled using the `Alert` interface in Selenium. You can switch to the alert using the `switchTo().alert()` method and perform actions such as accepting or dismissing the alert using the `accept()` or `dismiss()` methods.

## 90. Which of the following methods is used to clear the text in a text field using Selenium?

`element.clear()`

## 91. Which of the following methods is used to maximize the browser window in Selenium?

`driver.maximize_window()`