# InterviewBIT 31Q: <https://www.interviewbit.com/selenium-interview-questions/>

### 1. What is meant by Selenium Suite and what are its different components?

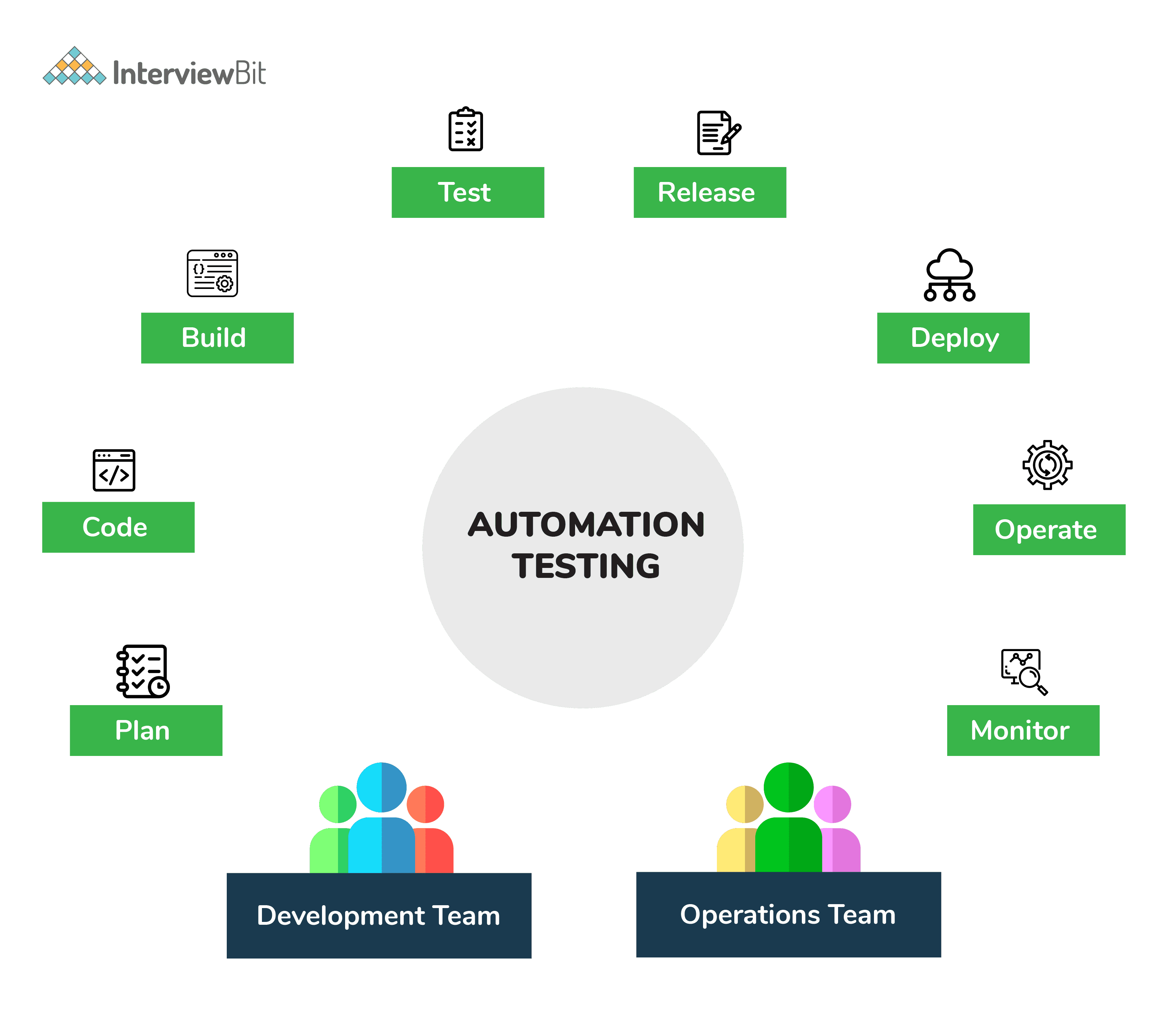
Selenium is a package of several testing tools and is therefore often referred to as a Selenium Suite with each of these tools designed to cater to a different testing requirement.

Following are the different components of Selenium Suite:

* **Selenium Integrated Development Environment (IDE)**: It is a Firefox/Chrome plug-in that is developed to speed up the creation of automation scripts by recording the user actions on the web browser and exporting them as a reusable script.
* **Selenium Remote Control (RC)**: It is a server that enables users to generate test scripts in their preferred programming language. It accepts commands from the test scripts and sends them to the browser as Selenium core JavaScript commands, for the browser to behave accordingly.
* [**Selenium WebDriver**](https://www.interviewbit.com/selenium-webdriver-interview-questions/)**:**It is a programming interface that helps create and run test cases by directly communicating with the web browser and using its native compatibility to automate. Unlike RC, it doesn’t require an additional server to create and run test cases.
* **Selenium Grid:**It allows parallel execution of tests on different browsers and operating systems by distributing commands to different machines simultaneously.

### 2. What is automation testing, and what are its advantages?

Automation testing or Test Automation is a process of automating the manual testing process of an application or a system by using testing tools that allow you to create scripts that can be executed repeatedly, generating detailed test reports of the application or system under test.



Advantages of Automated Testing are:

* It supports both the performance and functional testing of an application or system.
* It facilitates the execution of repeated test cases.
* It allows the parallel execution of the test cases.
* It improves the accuracy and efficiency of the system by reducing the manual intervention of humans to generate test cases.
* It helps in testing a large scale test matrix.
* It saves valuable time and money for the testing team involved in the project.

### 3. What are the advantages of using Selenium as an automation tool?

Following are the advantages of using Selenium for automated testing :

* **Open-Source:** Selenium's greatest strength, as previously said, is that it is a freeware and portable tool. There are no out-of-pocket expenses. The utility can be downloaded for free, and community-based help is also accessible.
* **Language assistance:**Java, Perl, Python, C#, Ruby, Groovy, JavaScript, and other languages are supported by Selenium. It has its own script, yet it is not constrained by it. It can work with a variety of languages, depending on the developers' and testers' preferences.
* **Compatible with a variety of operating systems:** Selenium may run on a variety of operating systems, including Windows, Mac OS X, Linux, and UNIX. A customized testing suite can be constructed on any platform and then executed on another using the Selenium suite of products. For example, you may write test cases in Windows and run them on a Linux system with ease.
* **Browser compatibility:** Selenium is compatible with a variety of web browsers, including Internet Explorer, Chrome, Firefox, Opera, and Safari. When running tests and testing them across multiple browsers at the same time, this becomes really useful.
* **Programming languages and framework support:** Selenium works with a variety of programming languages and frameworks. For source code compilation, it can, for example, integrate with ANT or Maven frameworks. It may also be used to test apps and generate reports using the TestNG framework. Continuous Integration (CI), can integrate with Jenkins or Hudson, and it can also integrate with other open-source tools to offer other functionalities.
* **Tests on a variety of devices:** On Android, iPhone, and Blackberry, Selenium Test Automation can be used to automate mobile web applications testing. This can aid in the generation of necessary results and the ongoing resolution of bugs present in the application.
* **Regular updates:** Selenium support is based on a community, which allows for frequent updates and upgrades. These upgrades are simple to install and don't require any special training. Selenium is thus both resourceful and cost-effective.
* **Selenium suites with a lot of content:** Selenium is more than just a single tool or utility; it's a full set of numerous testing tools that's why it's called a Suite. Each tool is tailored to specific testing needs and test environment constraints. Selenium also includes features such as Selenium IDE, Selenium Grid, and Selenium Remote Control (RC).
* **Ease with which it can be implemented:** Selenium has a user-friendly interface that makes it simple to develop and perform tests. Its open-source capabilities allow users to script their own extensions, making them simple to create, alter actions, and manipulate at a high level. Selenium's reporting features are also one of the reasons for its popularity, as it allows testers to extract results and take action based on them.

**You can download a PDF version of Selenium Interview Questions.**

[**Download PDF**](javascript:void(0))

### 4. What are the disadvantages of using Selenium as a testing tool?

The following are the disadvantages of using Selenium as a testing tool:

* **Tests web applications only**: Selenium supports the testing of only web-based applications. Mobile applications, Captcha, and Barcode readers cannot be tested using Selenium unless integrated with third-party tools like Appium and TestNG.
* **No built-in reporting and test management facility**: Selenium can generate reports only using third-party tools like TestNG or JUnit.
* **Unavailability of reliable tech support**: Since Selenium is an open-source tool, no dedicated support for user issues is available.
* **May require the knowledge of programming languages**: Some prior programming knowledge is required to use Selenium.

### 5. Why should Selenium be selected as a testing tool for web applications or systems?

Selenium provides the following advantages, which make it an excellent automated testing framework:

* It is free and open-source software with a large user base and supports providing community.
* It has cross-browser compatibility and supports multiple browsers like Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
* It supports multiple operating systems such as Windows, Linux, macOS, etc.
* It facilitates the usage of multiple programming languages including Scala, Ruby, Python, PHP, Perl, Java, Groovy, C#, etc.
* It provides support for distributed testing as well.

### 6. Can selenium be used to launch web browsers?

Yes, Selenium provides good support to launch web browsers like Google Chrome, Mozilla Firefox, Internet Explorer, etc.

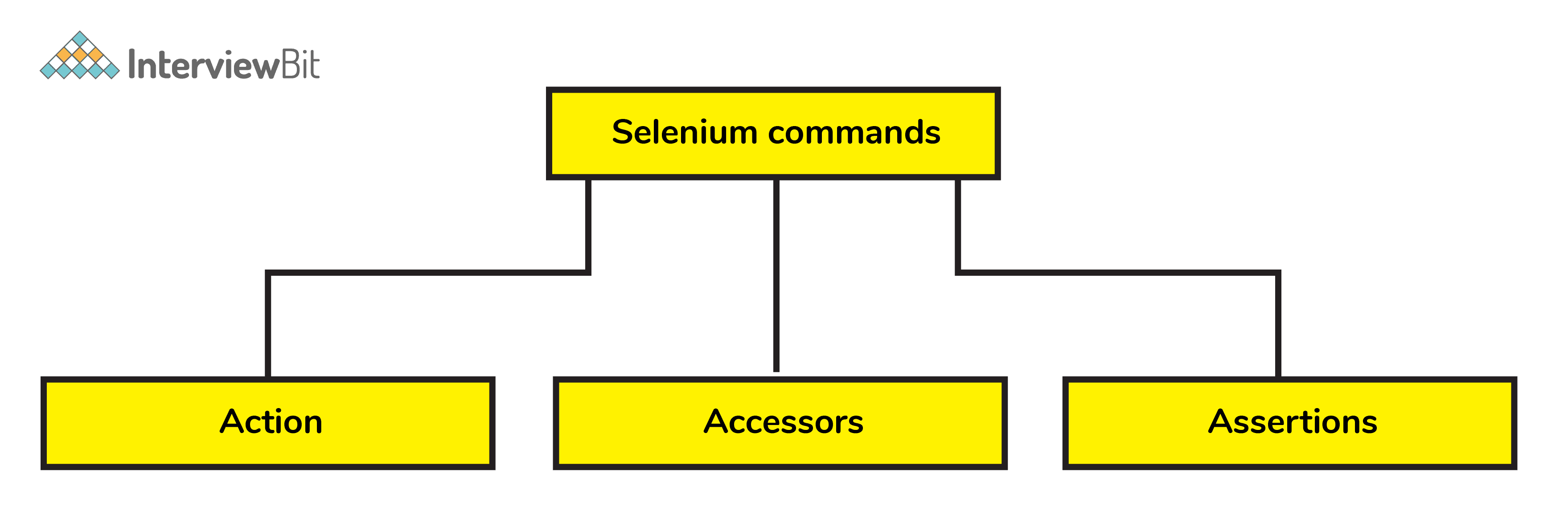
The following commands can be used to launch web browsers using Selenium:

* WebDriver driver = new FirefoxDriver();
* WebDriver driver = new ChromeDriver();
* WebDriver driver = new InternetExplorerDriver();

### 7. What is meant by Selenese? Explain different types of Selenium commands.

The language used for writing test scripts in Selenium IDE is called Selenese. It is a set of commands used to test your web application or system. Selenium commands could be divided into 3 major categories:

* **Actions:** These are the commands interacting directly with web applications.
* **Accessors:** These are the commands which allow users to store values to a user-defined variable.
* **Assertions:** They enable a comparison of the current state of the application with its expected state.



### 8. What is meant by a locator and name a few different types of locators present in Selenium.

A locator is an address for uniquely identifying web elements within a web page. There are different types of locators present in Selenium to identify web elements uniquely and accurately like:

* ID
* ClassName
* Name
* TagName
* LinkText
* PartialLinkText
* Xpath
* CSS Selector
* DOM.

### 9. State the major difference between “assert” and “verify” commands in Selenium.

Both “assert” and “verify” commands check whether the given condition is true or false and the only difference between them is that:

* **Assert:** Assert condition stops the execution of the testing if the given condition is false else would continue with the further tests.
* **Verify:** Verify the condition doesn’t stop the flow of execution irrespective of the condition being true or false.

### 10. What is meant by an exception test in Selenium?

An exception test is a test that expects an exception to be thrown inside a test class. It expects a @Test annotation followed by the expected exception name in the brackets.

Eg: @Test(expectedException = NoSuchElementException.class) is an exception test for missing elements in Selenium.

### 11. What is meant by XPath in Selenium. Explain XPath Absolute and XPath Relative.

XPath, also defined as XML-Path (Extensible Markup Language Path), is a language used to query XML documents and provide functionalities like locating elements in Selenium by iterating through each element in a webpage. In XPath, data is stored in a key-value pair format similar to an HTML tag. It uses a single slash, i.e. ‘ / ’ for creating an absolute path, and a double slash, i.e. ‘ // ’ for creating a relative path for an element to be located on a webpage.

### 12. In Xpath, what is the difference between "/" and "//"?

**Single Slash "/"** - A single slash is used to create an Xpath with an absolute path, i.e. the XPath will begin with the document node/start node.

For example,

Absolute XPath: /html/body/div/div/form/input

Here, /html is the root html node.

**Double Slash "//"** - The double slash is used to construct an Xpath with a relative path, which means the XPath can start selection from anywhere on the page.

For example,

Relative XPath: //input[@id = 'email']

Here, we can locate an input having id = ‘email’ present anywhere in the document object model (DOM).

### 13. What is the difference between the commands "type" and "typeAndWait" in the context of Selenium?

The "type" command is used to enter keyboard key values into a software web application's text box. It can also be used to choose values from a combo box, whereas the "typeAndWait" command is used when you finish typing and the software web page begins to reload. This command will wait for the page of the software program to reload before proceeding. You must use a basic "type" command if there is no page reload event when typing.

### 14. Differentiate between findElement() and findElements() in the context of Selenium with proper examples.

Following table lists the differences between findElement() and findElements() in Selenium :

| **findElement()** | **findElements()** |
| --- | --- |
| The first web element that matches the locator is returned. | It gives you a list of all the web items that match the locator. |
| If there are no matching web elements, a NoSuchElementException is produced. | If there are no matching elements, an empty list is returned. |
| Syntax − WebElement button = webdriver.findElement(By.name("<<Name value>>")); | Syntax − List<WebElement> buttons = webdriver.findElements(By.name("<<Name value>>")); |

* Using findElements():-

// JAVA

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class findElements {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "C:\\Users\\vaibhav\\Desktop\\Java\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String url = "https://www.exampleurl.com/example.htm";

driver.get(url);

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

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

List<WebElement> rows = driver.findElements(By.xpath("//table/tbody/tr[2]/td")); // xpath with index appended to get the values from the row 1of table using findElements(), which returns a list

System.out.println("The number of values in row 2 is "+ rows.size());

driver.close();

}

}

**Explanation -** In the above code, first of all, we import all the necessary headers and then set up the driver for the chrome browser. We use the findElements() method to find all the values present in the 2nd row of a table in the given URL web page using the XPath of the element.

* Using findElement() :-

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class findTagname {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "C:\\Users\\vaibhav\\Desktop\\Java\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String url = "https://www.exampleurl.com/example.htm";

driver.get(url);

driver.manage().timeouts().implicitlyWait(12, TimeUnit.SECONDS);

driver.findElement(By.cssSelector("input[id='search']")).sendKeys("Selenium"); //Using id tagname attribute combination for css expression and get the element from findElement()

driver.close();

}

}

**Explanation** - In the above code, first of all, we import all the necessary headers and then set up the driver for the chrome browser. We use the findElement() method to find an input element having an id attribute set as search.

### 15. In Selenium, how will you wait until a web page has been loaded completely?

There are two methods of making sure that the web page has been loaded completely in Selenium.

They are as follows :

1. Immediately after creating the webdriver instance, set an implicit wait:

temp = driver.Manage().Timeouts().ImplicitWait;

On every page navigation or reload, this will try to wait until the page is fully loaded.

2. Call JavaScript return document.readyState till "complete" is returned after page navigation. As a JavaScript executor, the web driver instance can be used.

Code example:

new WebDriverWait(firefoxDriver, pageLoadTimeout).until(

webDriver -> ((JavascriptExecutor) webDriver).executeScript("return document.readyState").equals("complete"));

## Selenium Interview Questions for Experienced

### 16. Explain the difference between driver.close() and driver.quit() command in Selenium?

Following is the major difference between both the commands:

* driver.close() command closes the currently active window on which the user is working or the window being currently accessed by the web driver.
* driver.quit() command, unlike the driver.close() command it closes all the windows opened by the program and hence should be used with care.

Both the commands don’t take any parameter and don’t return any value either.

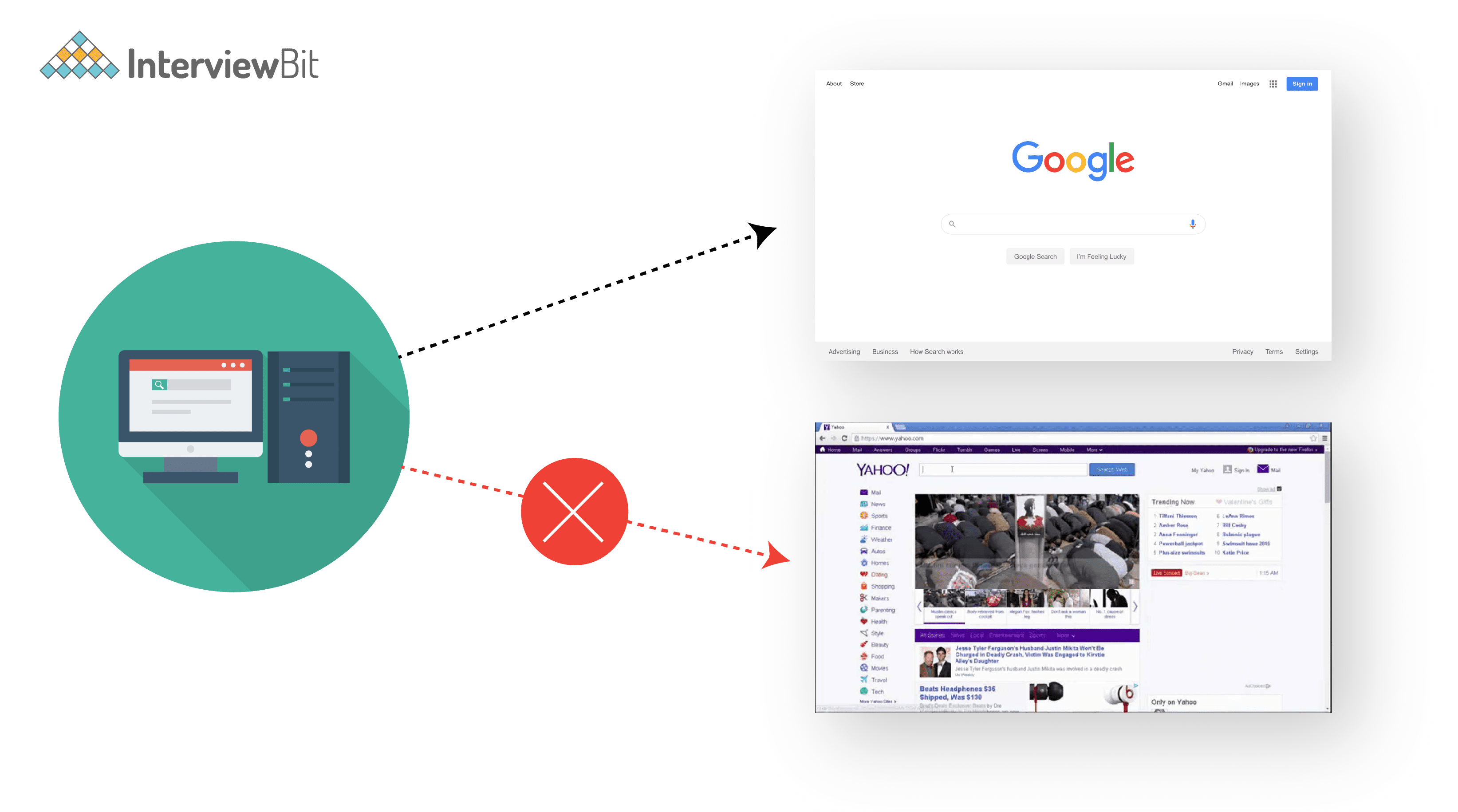
### 17. Explain the various navigation commands supported by Selenium?

Selenium has the support of majorly 4 navigation commands:

* navigate().back(): This command is used for taking the user to the last webpage of the browser history.
* navigate().forward(): This command is used for taking the user to the next web page of the browser history.
* navigate().refresh(): This command is used for reloading the web components of a webpage by refreshing it.
* navigate().to(): This command is used for navigating to a particular URL in a new web browser. It takes the URL to be migrated to, as a parameter.

### 18. Explain the same-origin policy and how Selenium handles it?

Same Origin policy is a feature adopted for security purposes that allows a web browser to run scripts from one webpage to access the contents of another webpage provided both the pages have the same origin. The URL scheme, hostname, and port number combo are referred to as the origin. This policy was introduced to prevent access to sensitive data on one webpage by another for ill purposes. Consider a Java program used by scaler.com, the program can access domain pages like scaler.com/mentors, scaler.com/courses but none from different domains like facebook.com.



The Selenium Server (Selenium RC) 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.

### 19. Explain the difference between findElement() and findElements() in Selenium.

Following is the major difference between the **two commands**:

1.findElement(): command is used for finding a particular element on a web page, it is used to return an object of the first found element by the locator. Eg:

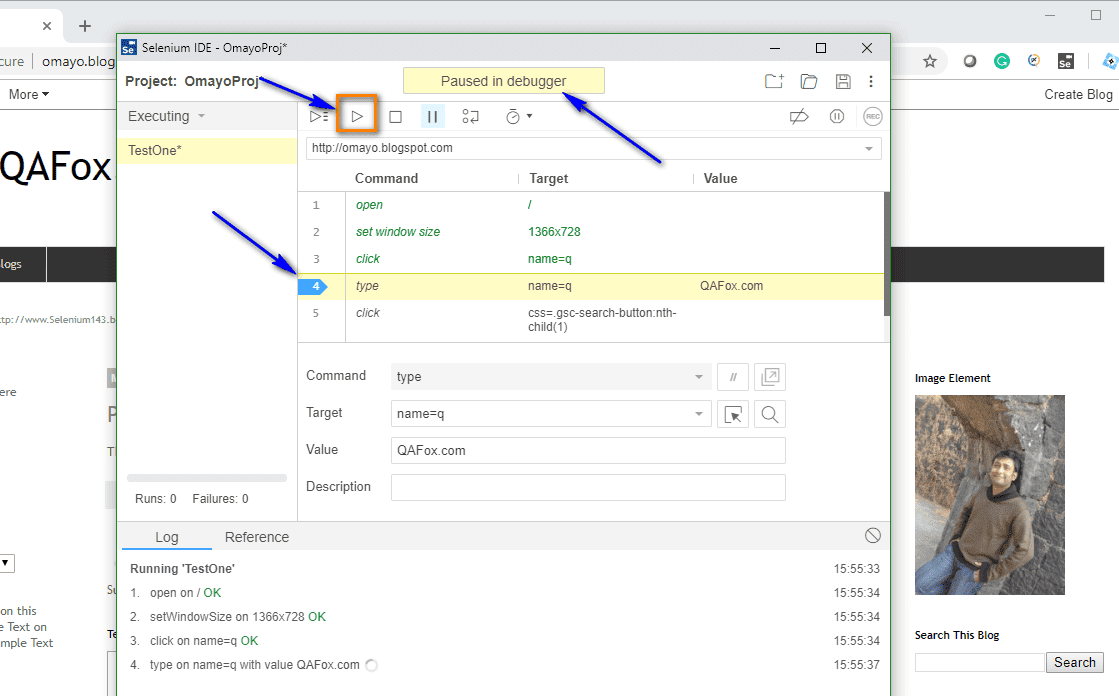
WebElement element = driver.findElement(By.id(example));

2. findElements(): command is used for finding all the elements in a web page specified by the locator value. The return type of this command is the list of all the matching web elements. Eg:

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

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

The pause feature is built to handle exceptions in the test script by allowing the user to pause at the statement causing the exception and enter the debug mode by clicking on the pause icon on the top right corner of the IDE. This feature prevents the entire test case's failure and gives the user a chance to correct the error instantly.



### 21. With the help of code snippets, explain how we can create right-click and mouse hover actions in Selenium.

The following code can replicate right-click action:

actions action = newActions(driver);

WebElement element = driver.findElement(By.id("elementId"));

action.contextClick(element).perform();

The following code can replicate mouse hover action:

actions action = newActions(driver);

WebElement element = driver.findElement(By.id("elementId"));

action.moveToElement(element).perform();

### 22. Can we handle a windows-based pop-up in Selenium, and if not, then what are the alternatives?

No, Selenium doesn’t support windows based pop-ups as it’s an automated testing tool built for web application based testing. However, with the support of third-party tools like AutoIT, Robot class, etc., windows-based pop-ups can be handled in selenium.

### 23. Can you capture a screenshot using Selenium? If yes, write a simple code to illustrate the same.

Yes, using a web driver in Selenium, we can capture the screenshot. Following is the code to do the same:

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import java.io.File;

import java.io.IOException;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class TakeScreenshot {

WebDriver drv;

@ Before

public void setUp() throws Exception {

driver = new FirefoxDriver();

drv.get("https://google.com");

}

@ After

public void tearDown() throws Exception {

drv.quit();

}

@ Test

public void test() throws IOException {

// Capture the screenshot

File scrFile = ((TakeScreenshot)drv).getScreenshotAs(OutputType.FILE);

// Code for pasting screenshot to a user-specified location

FileUtils.copyFile(scrFile, new File("C:\\Screenshot\\Scr.jpg"))

}

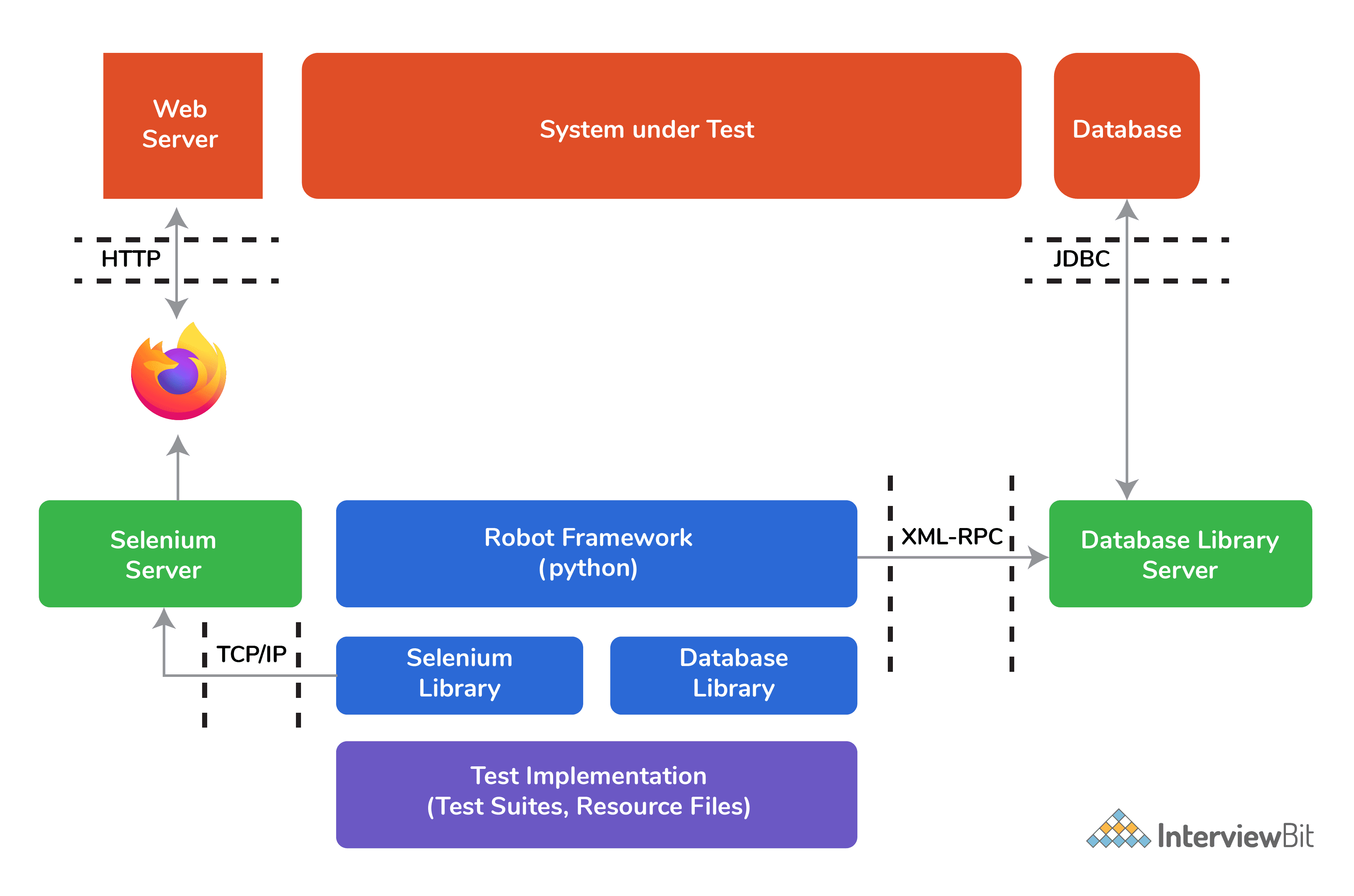
}

### 24. Explain different types of framework and connection of Selenium with Robot Framework.

Following are the different types of frameworks:

* **Behavior-Driven Development Framework:** This type of framework provides a readable and easily understandable format to Business Analysts, Developers, Testers, etc.
* **Data-Driven Testing Framework:**This type of framework helps separate test data from the test-script logic by storing test data in some external database in the form of key-value pairs. These keys can be used for accessing as well as populating data into the test scripts.
* **Keyword-Driven Testing Framework:**This type of framework is an extension of the data-driven testing framework. In addition to separating test data and the test-script logic, it also separates a part of the test script code by storing it in an external data file.
* **Library Architecture Testing Framework:**This type of framework groups common steps into functions under a library and calls these functions as and when required.
* **Module-Based Testing Framework:** This type of framework divides each test application into several isolated and logical modules, with each module having its distinct test script.
* **Hybrid Testing Framework:** This type of framework is a combination of the above-mentioned frameworks leveraging all their good features.
* Robot Framework is a modular open-source automation framework that can interact with 3rd party libraries and functions. To execute a web testing library such as Selenium, a test automation runner or an automation wrapper is required, which is provided to it in the form of Robot Framework. Other popular test runners to serve the same purpose are MSTest, TestNG, Nunit, Junit, etc.

The below diagram shows the connection of the Robot framework to the Selenium library:



### 25. Demonstrate usage of Selenium through a test application.

You need the following prerequisites to run a demo Selenium test script:

* **Java SDK**in your respective Operating System.
* A **Java-based IDE** such as Eclipse or IntelliJ.
* A **Selenium WebDriver** to be added as a dependency to Java IDE.

package scalerAcademy;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.WebDriver;

public class MyFirstTestClass {

public static void main(String[] args) throws InterruptedException {

//It sets the system property to the given value.

System.setProperty("webdriver.gecko.driver","D:\\Softwares\\geckodriver.exe”);

WebDriver driver = new FirefoxDriver();

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

//Launch website in the browser

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

//The sleep pauses the execution of the thread for 5000 ms.

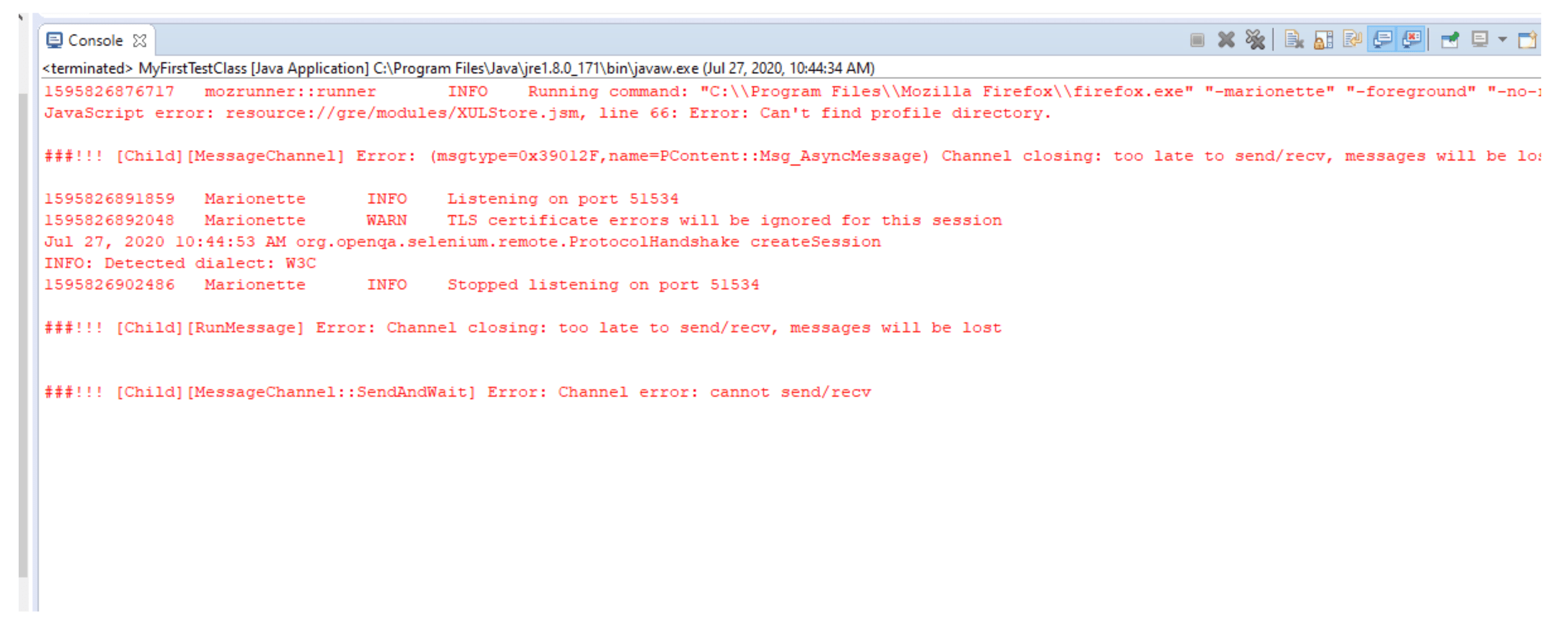
Thread.sleep(5000);

driver.quit();

}

}

Once you run the above script in a Java IDE, you’ll get the following execution logs displayed in your IDE window.



### 26. Explain basic steps of Selenium testing and its widely used commands via a practical application.

Selenium testing can be divided into the following seven basic elements:

**1.** **Creating an instance of a Webdriver**: This is the first step for all the usages of a Selenium webdriver API. An instance of a webdriver interface is created using a constructor of a particular browser. This webdriver instance is used to invoke methods and to access other interfaces. Following are the most commonly used commands for initialising a web driver:

Firefox:

WebDriver driver = new FirefoxDriver();

Chrome:

WebDriver driver = new ChromeDriver();

Safari Driver:

WebDriver driver = new SafariDriver();

Internet Explorer:

WebDriver driver = new InternetExplorerDriver();

**2. Navigating to a webpage:** The second step after initializing an instance of a webdriver, to navigate to a particular webpage you want to test. Following are the most commonly used commands for webpage navigation:

Navigate to URL:

driver.get(“https://www.interviewbit.com”)

driver.navigateo.to(“https://www.interviewbit.com”)

Refresh page:

driver.navigate().refresh()

Navigate forward in browser history:

driver.navigate().forward()

Navigate backward in browser history:

driver.navigate().backward()

**3. Locating an HTML element on the webpage:** To interact with a web element and perform actions on it like clicking a button or entering text, we first need to locate the desired elements such as the button or the textbox on the web page. Following are the most commonly used commands for web element navigation:

Locating by ID:

driver.findElement(By.id("q")).sendKeys("Selenium 3");

Location by Name:

driver.findElement(By.name("q")).sendKeys ("Selenium 3");

Location by Xpath:

driver.findElement(By.xpath("//input[@id==’q’])).sendKeys("Selenium 3");

Locating Hyperlinks by Link Text:

driver.FindElement(By.LinkText("edit this page")).Click();

Locating by ClassName

driver.findElement(By.className("profileheader"));

Locating by TagName

driver.findElement(By.tagName("select')).click();

Locating by LinkText

driver.findElement(By.linkText("NextPage")).click();

Locating by PartialLinkText

driverlindElement(By.partialLinkText(" NextP")).click();

**4. Performing actions on an HTML element:**Once we have located the HTML element, the next step is interacting with it. Following are the most commonly used commands for performing actions on HTML element:

Entering a username

usernameElement.sendKeys("InterviewBit");

Entering a password

passwordElement.sendKeys("Raw");

Submitting a text input element

passwordElement.submit();

Submitting a form element:

formElement.submit();

**5. Anticipating browser response from the action:** Once an action is performed, anticipating a response from the browser to test comes under this step. It takes a second or two for the action to reach the browser, and hence wait is often required for this step. There are two main types of wait conditions:

**Implicit Wait:** It sets a fixed, definite time for all the webdriver interactions. It’s slightly unreliable as web driver response times are usually unpredictable. Eg:

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

**Explicit Wait:** This type of wait condition sets an expected condition to occur on the web page or a maximum wait time for all the webdriver interactions. Eg:

WebElement messageElement = wait.until( ExpectedConditions.presenceOfElementLocated(By.id("loginResponse")) );

**6. Running tests and recording their results using a test framework:**in this step, we run tests in an automated test script to evaluate an application's function and performance. Various test frameworks are used for this step, such as:

1. JUnit for Java
2. NUnit for C#
3. Unittest or Pyunit for Python
4. RUnit for Ruby

   Most frameworks use some sort of asset statement to verify their test results from the expected results. Eg:

assertEquals (expectedMessage, actualMessage);

**7. Concluding a test:** In this step, we conclude a test by invoking a quit method on the driver variable. This step closes all the webpages, quits the WebDriver server, and releases the driver. Eg:

driver.quit();

The following is an example of an app that covers all the steps mentioned above:

import org.openqa.selenium.By,

import org.openqa.selenium.WebElement,

import org.openqa.selenium.support.ni.ExpectedConditiof, import org.openqa.selenium.support.ni.WebOriverWait,

import org.junit.Assert;

public class Example {

public static void main(String[] args) {

// Creating a driver instance

WebDriver driver = new FirefoxDriver(),

// Navigate to a web page

­driver.get("http://www.foo.com");

// Enter text to submit the form

WebElement usernameElement = driver.findElement( By.name("username"));

WebElement passwordElement = driver.findElement(By.name(”password"));

WebElement formElement = driver.findElement(By.id(”loginForm"));

usernameElement.sendKeys("Scaler Academy");

passwordElement.sendKeys("Raw");

formElement.submit(); // submit by form element

//Putting an explicit wait

WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement messageElement = wait.until(

ExpectedConditions.presenceofElementLocated(By.id(”loginResponse"))

) ;

// Run a test

String message = messageElement.getrept();

String successMsg = "Welcome to foo. You logged in successfully.”;

Assert.assertEquals (message, successMsg);

// Conclude a test

driver.quit();

}

}

### 27. What do you understand about the Page Object Model in the context of Selenium? What are its advantages?

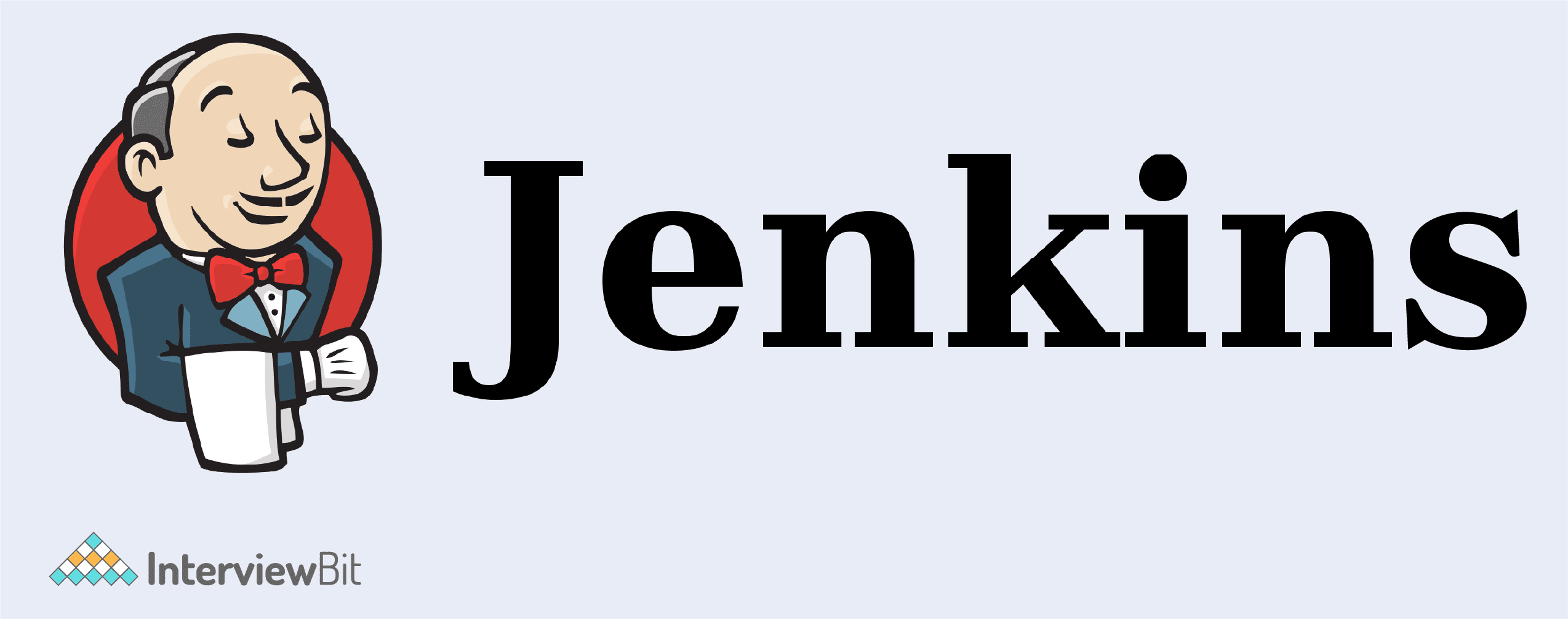
Page Object Model (POM) is a design pattern that generates an Object Repository for web UI elements and is widely used in test automation. The paradigm has the advantage of reducing code duplication and improving test maintenance. According to this paradigm, each web page in the application should have its own Page Class. This Page class will identify the web page's WebElements and also has Page methods that operate on those WebElements. The names of these methods should correspond to the tasks they perform, for example, if a loader is waiting for the payment gateway to appear, the POM method name could be waitForPaymentScreenDisplay().

Following are the advantages of the Page Object Model (POM) :

* According to the Page Object Design Pattern, user interface activities and flows should be separated from verification. Our code is clearer and easier to understand as a result of this notion.
* The second advantage is that the object repository is independent of test cases, allowing us to reuse the same object repository with different tools. For example, we can use Selenium to combine Page Object Model with TestNG/JUnit for functional testing and JBehave/Cucumber for acceptability testing.
* Because of the reusable page methods in the POM classes, code gets less and more efficient.
* Methods are given more realistic names that can be easily associated with the UI operation. If we land on the home page after clicking the button, the function name will be 'gotoHomePage()'.

### 28. What do you understand about Jenkins? Why are the benefits of using it with Selenium?

Hudson Lab's Jenkins is the most popular open-source continuous integration technology. It's cross-platform, meaning it can run on Windows, Linux, Mac OS, and Solaris. Jenkins is a Java application. Jenkin's main purpose is to keep track of any job, such as SVN (Apache Subversion) checkouts, cron jobs, or application states. When a specific event in an application occurs, it triggers pre-configured actions.



Following are the **features of Jenkins**:

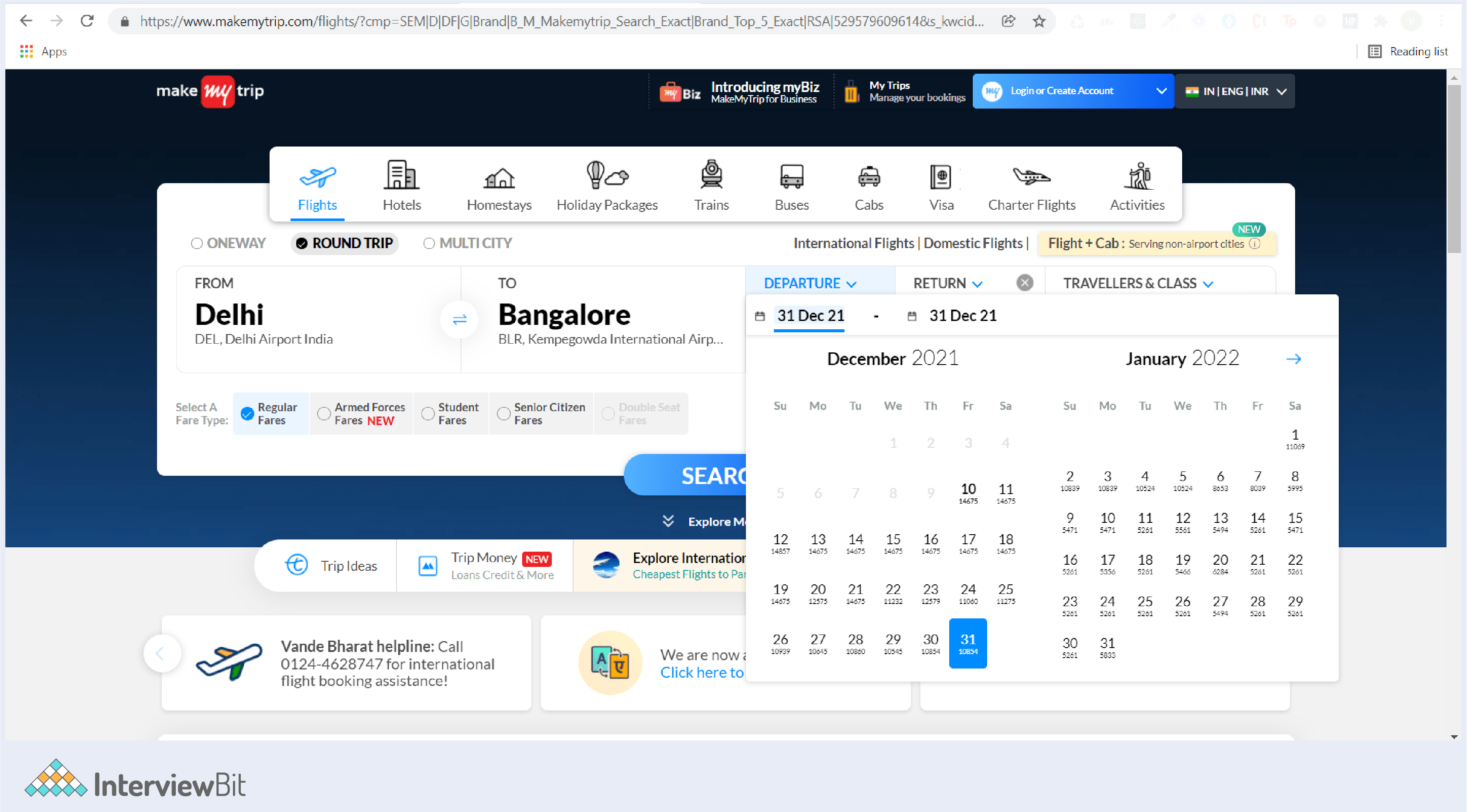
* Jenkins generates a list of all changes made in SVN repositories, for example.
* Jenkins gives permanent links to the most recent build or failed build, which can be utilised for convenient communication.
* Jenkins is simple to install using either a direct installation file (exe) or a war file for deployment via the application server.
* Jenkins can be set up to send the content of the build status to an email address.
* Simple Configuration: Jenkins makes it simple to set up multiple tasks.
* Jenkins can be configured to run the automated test build on TestNg following every SVN build.
* Jenkins documents the details of the jar, its version, and the mapping of build and jar numbers.
* Plugins: Jenkins can be set to utilise features and additional functionality provided by third-party plugins.

Following are the reasons we **use Jenkins with Selenium** :

* When you run Selenium tests in Jenkins, you can run them every time your program changes, and when the tests pass, you may deploy the software to a new environment.
* Jenkins may execute your tests at a predetermined time.
* The execution history as well as the Test Reports can be saved.
* Jenkins allows you to develop and test a project in continuous integration using Maven.

### 29. How will you select a date from a datepicker in a webpage using Selenium for automated testing? Explain with a proper code.

In such types of questions, the interviewer wants to assess how clear your understanding is about the framework. It is a good practice to explain the code while you write it so that the interviewer is engaged at all points and does not feel left out. We will be considering an example on MakeMyTrip.



Here, we will be using the chrome browser and so we will be implementing the code for the chrome browser only. You can implement similar code for firefox and other browsers as well.

First of all, we create a package named browserSelection which contains a class defined for handling different types of browsers such as chrome, firefox that we may want to use.

package browserSelection;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class SelectBrowser

{

static WebDriver driver;

public static WebDriver useChrome()

{

System.setProperty("webdriver.chrome.driver", "E:\\SeleniumLibs\\\\chromedriver\_win32\\chromedriver.exe");

driver = new ChromeDriver();

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

return driver;

}

}

Next, we create a package named datepicker which will contain a class containing methods defined for selecting a specific date on the website of MakeMyTrip. We need to import this package into our driver class and call the respective methods.

package datepicker;

import java.awt.AWTException;

import java.awt.Robot;

import java.awt.event.KeyEvent;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebDriverException;

import org.openqa.selenium.WebElement;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

import browserSelection.SelectBrowser;

public class DatePick

{

WebDriver driver;

@BeforeMethod

public void startBrowser()

{

driver = SelectBrowser.useChrome();

}

@Test

public void selectDateUtil() throws InterruptedException, AWTException

{

//Modify Wait time as per the Network Ability in the Thread Sleep method

driver.get("https://www.makemytrip.com/");

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

Thread.sleep(5000);

try

{

driver.findElement(By.xpath("//input[@id='hp-widget\_\_depart']")).click();

Thread.sleep(2000);

Date sampleDate = new Date(); // initialising the date object with the current date

SimpleDateFormat formatter = new SimpleDateFormat("dd-MMM yyyy");

String date = formatter.format(sampleDate); // formatting the date object in dd-MMM yyyy format

String splitter[] = date.split("-");

String monthYear = splitter[1]; // storing the month and year concatenated string excluding the day number

String day = splitter[0]; // storing the day number in the current date

System.out.println(monthYear);

System.out.println(day);

selectDate(monthYear,day); // function invocation

Thread.sleep(3000);

public void selectDate(String monthYear, String select\_day) throws InterruptedException

{

List<WebElement> elements = driver.findElements(By.xpath("//div[@class='ui-datepicker-title']/span[1]"));

for (int i=0; i<elements.size();i++)

{

System.out.println(elements.get(i).getText());

//Selecting the month

if(elements.get(i).getText().equals(monthYear))

{

//Selecting the date

List<WebElement> days = driver.findElements(By.xpath("//div[@class='ui-datepicker-inline ui-datepicker ui-widget ui-widget-content ui-helper-clearfix ui-corner-all ui-datepicker-multi ui-datepicker-multi-2']/div[2]/table/tbody/tr/td/a"));

for (WebElement d:days)

{

System.out.println(d.getText());

if(d.getText().equals(select\_day))

{

d.click();

Thread.sleep(10000);

return;

}

}

}

}

// if we do not find the matching month and year, we click on the arrow button to load new months.

driver.findElement(By.xpath("//div[@class='ui-datepicker-inline ui-datepicker ui-widget ui-widget-content ui-helper-clearfix ui-corner-all ui-datepicker-multi ui-datepicker-multi-2']/div[2]/div/a/span")).click();

selectDate(monthYear,select\_day); // function invocation

}

@AfterMethod

public void endBrowser()

{

driver.quit();

}

}

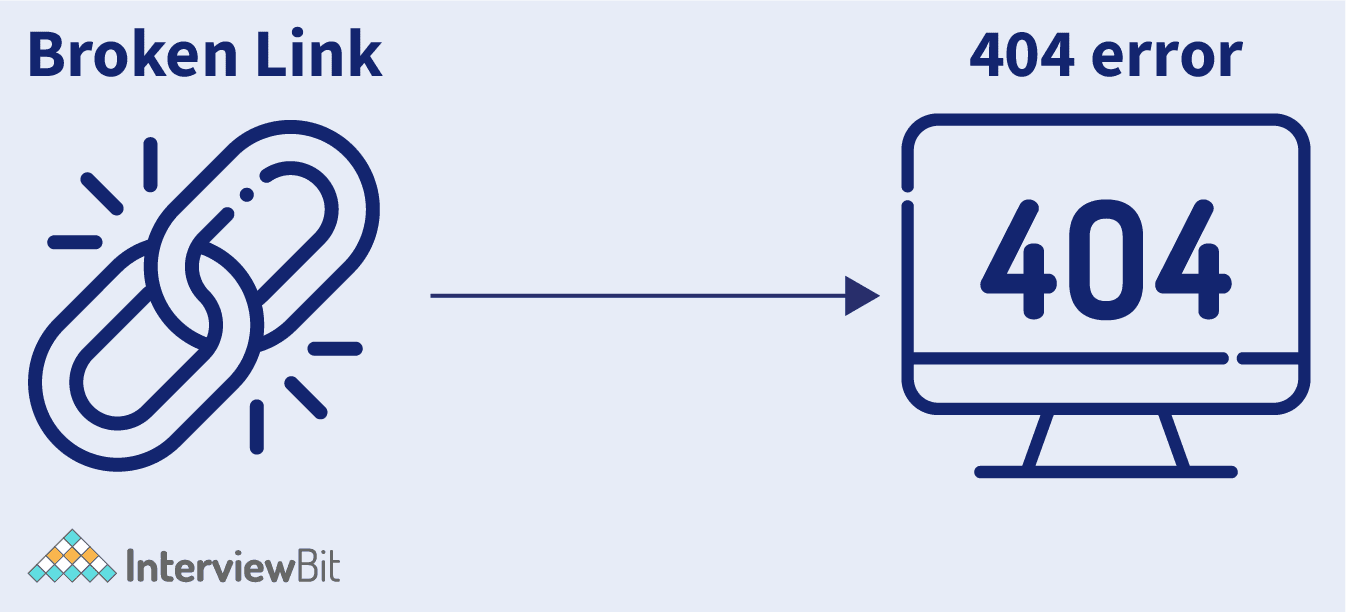
}

}

In the above code, the function startBrowser() is used to invoke the useChrome() method from the imported package browserSelection. The function selectDateUtil() is used to select the current date from the date picker of the sample web page. The endBrowser() function is used to close the driver connections by invoking the quit() method.

### 30. What do you understand about broken links? How can you detect broken links in Selenium? Explain properly with code.

Links or URLs that are not reachable are known as broken links. They may be unavailable or inoperable due to a server issue. A URL's status will always be 2xx, indicating that it is legitimate. There are a variety of HTTP status codes, each with its own set of functions. HTTP status 4xx and 5xx indicate an invalid request. The 4xx class of status codes is used for client-side errors, while the 5xx class is used for server response errors.



You should always check for broken links on your site to ensure that the user does not end up on an error page. If the rules aren't updated appropriately, or the requested resources aren't available on the server, the error will occur. Manual link checking is a time-consuming task because each web page may have a huge number of links, and the process must be performed for each page.

To find broken links in Selenium, follow the instructions below.

* Using the <a> (anchor) tag, collect all of the links on a web page.
* For each link, send an HTTP request.
* Make that the HTTP response code is correct.
* Based on the HTTP response code, determine whether the link is genuine or not.
* Repeat the procedure for all of the links that were captured in the first step.

package SeleniumPackage;

import java.io.IOException;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.Iterator;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

public class BrokenLinks {

public static void main(String[] args) {

String pageURL = "http://www.interviewbit.com";

String url = "";

HttpURLConnection huc = null;

int responseCode = 200;

System.setProperty("webdriver.chrome.driver", "C:\\Users\\user\\Downloads\\selenium\\chromedriver\_win32\\chromedriver.exe");

ChromeOptions options = new ChromeOptions();

options.addArguments("--headless", "--disable-gpu", "--window-size=1920,1200","--ignore-certificate-errors", "--silent");

WebDriver driver = new ChromeDriver(options);//Creating an instance of the WebDriver class

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

driver.get(pageURL);

List<WebElement> links = driver.findElements(By.tagName("a")); // getting hold of all the elements having the anchor tag

Iterator<WebElement> it = links.iterator();

// Iterating over the obtained list of elements and checking them one by one

while(it.hasNext()){

url = it.next().getAttribute("href");

System.out.println(url);

if(url == null || url.isEmpty()){

System.out.println("The linked element has invalid href url.");

continue;

}

if(!url.startsWith(pageURL)){

System.out.println("URL belongs to another domain, skipping it.");

continue;

}

try {

huc = (HttpURLConnection)(new URL(url).openConnection());

huc.setRequestMethod("HEAD");

huc.connect(); // connecting to the url

responseCode = huc.getResponseCode(); // reading the response code on firing the url

if(responseCode >= 400){

System.out.println(url+" is a broken link");

}

else{

System.out.println(url+" is a valid link");

}

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

driver.quit();

}

}

**Explanation** - In the above code, we first set up the system properties and then initialise a webdriver object. We find all the elements in the web page having the anchor tag with the help of the findElements() method. Then, we iterate over the list obtained one by one and fire up the URL and read the response code received to check if it is a broken link or not.

### 31. What do you understand about window handle in the context of automated testing? How can you handle multiple windows in Selenium?

Window handle is a one-of-a-kind identifier that contains the addresses of all of the windows. Consider it a window pointer that returns the string value. Each browser will presumably have its own window handle. This window handle function aids in the retrieval of all window handles.  
Syntax :

* get.windowhandle(): This function is used to retrieve the current window's handle.
* get.windowhandles(): This function is useful for retrieving the handles of all the windows that have been opened.
* set: This method allows you to set the window handles as a string.   
  set<string> set= driver.get.windowhandles()
* switch to: This method aids in the switching of windows.
* action: This method aids in the execution of specific window actions.

Let us consider an example code to understand better. We will open the website of InterviewBit and then click on all the links available on the web page. Then, we will switch from the parent window to multiple different child windows and then switch back to the parent window at last.

package SeleniumPackage;

import java.util.Iterator;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class WindowHandle\_Demo {

public static void main(String[] args) throws Exception {

System.setProperty("webdriver.chrome.driver", "C:\\Users\\user\\Downloads\\selenium\\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

// Loading the website

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

String parent=driver.getWindowHandle(); // storing the parent window name as a string

List<WebElement> links = driver.findElements(By.tagName("a")); // storing the list of all the elements having an anchor tag

Iterator<WebElement> it = links.iterator();

// Iterating over the list elements one by one and clicking all the links to open new child windows

while(it.hasNext()){

it.next().click();

}

Set<String> s = driver.getWindowHandles(); // Storing the list of all the child windows

Iterator<String> I1= s.iterator();

// Iterating over the list of child windows

while(I1.hasNext())

{

String child\_window=I1.next();

if(!parent.equals(child\_window))

{

driver.switchTo().window(child\_window);

System.out.println(driver.switchTo().window(child\_window).getTitle());

driver.close();

}

}

//switch to the parent window

driver.switchTo().window(parent);

}

}

In the above code, we open the landing page of interviewbit and then find all the elements having the anchor tag and click them to open multiple child windows. Then, we iterate over each of the child windows and print them as  a string. Finally, having traversed over the entire list, we break from the loop and switch back to the parent window.

# Guru99 100Q: <https://www.guru99.com/top-100-selenium-interview-questions-answers.html>

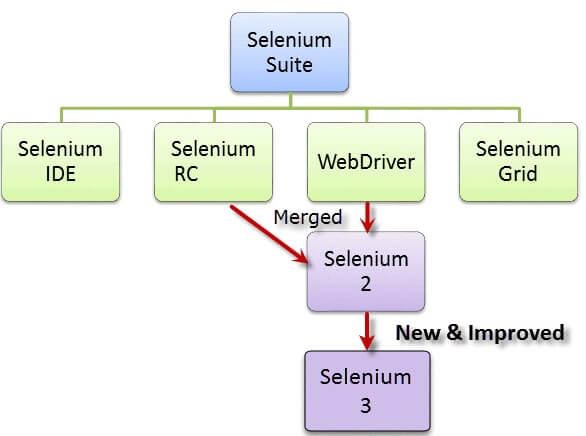
#### 1) What is Selenium and what is composed of?

Selenium is a suite of tools for automated web testing.  It is composed of

* **Selenium IDE (Integrated Development Environment) :**It is a tool for recording and playing back.  It is a firefox plugin
* **WebDriver and RC:**  It provide the APIs for a variety of languages like Java, .NET, PHP, etc. With most of the browsers Webdriver and RC works.
* **Grid:**With the help of Grid you can distribute tests on multiple machines so that test can be run parallel which helps in cutting down the time required for running in browser test suites

#### 2) What is Selenium 2.0?

Web[Testing](https://www.guru99.com/software-testing.html)tools Selenium RC and WebDriver are consolidated in single tool in Selenium 2.0



What is Selenium 2.0

#### 3) Mention what is Selenium 3.0?

Selenium 3.0 is the latest version of Selenium. It has released 2 beta versions of selenium 3.0 with few of the below changes:

Here are few new features added to Selenium 3.0

|  |  |
| --- | --- |
| **Beta 1 updates** | **Beta 2 updates (Only for Java)** |
| * Minimum[Java](https://www.guru99.com/java-tutorial.html)version is now 8+ | * System property webdriver.gecko.driver now forces the server in marionette or legacy firefox driver mode, ignoring any related Desired Capability |
| * It will support for Firefox Via Mozilla’s geckodriver | * Grid fixes NPE’s on registration when -browser not specified |
| * Support for Edge is provided by MS * It now supports Safari on MacOS via Apple’s own Safari driver | * Update GeckoDriver –port argument in all bindings |

#### 4) How will you find an element using Selenium?

In Selenium every object or control in a web page is referred as an elements, there are different ways to find an element in a web page they are

* ID
* Name
* Tag
* Attribute
* CSS
* Linktext
* PartialLink Text
* Xpath etc

#### 5) List out the test types that are supported by Selenium?

For web based application testing selenium can be used

The test types can be supported are

a) Functional, Learn More about [Functional Testing.](https://www.guru99.com/functional-testing.html)

b) Regression

For post release validation with continuous integration [automation tool](https://www.guru99.com/automated-testing-tools.html) could be used

a) Jenkins

b) Hudson

c) Quick Build

d) CruiseCont

#### 6) Explain what is assertion in Selenium and what are the types of assertion?

Assertion is used as a  verification point. It verifies that the state of the application conforms to what is expected.  The types of assertion are “assert” , “verify” and “waitFor”.

#### 7) Mention what is the use of X-path?

X-Path is used to find the WebElement in web pages. It is also useful in identifying the dynamic elements.

Refer Complete Guide on [XPath](https://www.guru99.com/xpath-selenium.html)

#### 8) Explain the difference between single and double slash in X-path?

Single slash ‘/ ’

* Single slash ( / ) start selection from the document node
* It allows you to create ‘absolute’ path expressions

Double Slash ‘// ’

* Double slash ( // ) start selection matching anywhere in the document
* It enables to create ‘relative’ path expressions

#### 9) List out the technical challenges with Selenium?

Technical challenges with Selenium are

* Selenium supports only web based applications
* It does not support the Bitmap comparison
* For any reporting related capabilities have to depend on third party tools
* No vendor support for tool compared to commercial tools like HP UFT
* As there is no object repository concept in Selenium, maintainability of objects becomes difficult

#### 10) What is the difference between type keys and type commands ?

TypeKeys() will trigger[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)event in most of the cases whereas .type() won’t. Type key populates the value attribute using JavaScript whereas .typekeys() emulates like actual user typing

#### 11) What is the difference between verify and assert commands?

**Assert:**  Assert allows to check whether an element is on the page or not. The test will stop on the step failed, if the asserted element is not available. In other words, the test will terminated at the point where check fails.

**Verify:** Verify command will check whether the element is on the page, if it is not then the test will carry on executing.  In verification, all the commands are going to run guaranteed even if any of test fails.

#### 12) What is JUnit Annotations and what are different types of annotations which are useful ?

In JAVA a special form of syntactic meta-data can be added to Java source code, this is know as Annotations.  Variables, parameters, packages, methods and classes are annotated some of the[Junit](https://www.guru99.com/junit-tutorial.html)annotations which can be useful are

* Test
* Before
* After
* Ignore
* BeforeClass
* AfterClass
* RunWith

#### 13) While using click command can you use screen coordinate?

To click on specific part of element, you would need to use clickAT command.  ClickAt command accepts element locator and x, y co-ordinates as arguments-

clickAt (locator, cordString)

#### 14)  What are the advantages of Selenium?

* It supports C#, PHP, Java, Perl, Phython
* It supports different OS like Windows,[Linux](https://www.guru99.com/unix-linux-tutorial.html)and Mac OS
* It has got powerful methods to locate elements (Xpath, DOM , CSS)
* It has highly developer community supported by Google

#### 15) Why testers should opt for Selenium and not QTP?

Selenium is more popular than[QTP](https://www.guru99.com/quick-test-professional-qtp-tutorial.html)as

* Selenium is an open source whereas QTP is a commercial tool
* Selenium is used specially for testing web based applications while QTP can be used for testing client server application also
* Selenium supports Firefox, IE, Opera, Safari  on operating systems like Windows, Mac, Linux etc. however QTP is limited to Internet Explorer on Windows.
* Selenium supports many programming languages like Ruby, Perl,[Python](https://www.guru99.com/python-tutorials.html)whereas QTP supports only VB script

## Advanced Selenium Automation Testing Interview Questions for 3/5/10 Years Experience

Following are the top Selenium interview questions for experience candidates:

#### 16) What are the four parameter you have to pass in Selenium?

Four parameters that you have to pass in Selenium are

* Host
* Port Number
* Browser
* URL

#### 17) What is the difference between setSpeed() and sleep() methods?

Both will delay the speed of execution.

Thread.sleep () :  It will stop the current (java) thread for the specified period of time.  Its done only once

* It takes a single argument in integer format

Ex: thread.sleep(2000)- It will wait for 2 seconds

* It waits only once at the command given at sleep

SetSpeed () :  For specific amount of time it will stop the execution for every selenium command.

* It takes a single argument in integer format

Ex: selenium.setSpeed(“2000”)- It will wait for 2 seconds

* Runs each command  after setSpeed delay by the number of milliseconds mentioned in set Speed

This command is useful for  demonstration purpose or if you are using a slow web application

#### 18) What is same origin policy? How you can avoid same origin policy?

The **“Same Origin Policy”** is introduced for security reason, and it ensures that content of your site will never be accessible by a script from another site.  As per the policy, any code loaded within the browser can only operate within that website’s domain.

To avoid “Same Origin Policy” proxy injection method is used, in proxy injection mode the Selenium Server acts as a client configured **HTTP proxy** , which sits between the browser and application under test and then masks the AUT under a fictional URL

#### 19) What is heightened privileges browsers?

The purpose of heightened privileges is similar to Proxy Injection, allows websites to do something that are not commonly permitted.  The key difference is that the browsers are launced in a special mode called heightened privileges.  By using these browser mode, Selenium core can open the AUT directly and also read/write its content without passing the whole AUT through the Selenium RC server.

#### 20) How you can use “submit” a form using Selenium ?

You can use “submit” method on element to submit form-

element.submit () ;

Alternatively you can use click method on the element which does form submission

#### 21) What are the features of TestNG and list some of the functionality in TestNG which makes it more effective?

TestNG is a testing framework based on JUnit and NUnit to simplify a broad range of testing needs, from [Unit Testing](https://www.guru99.com/unit-testing-guide.html) to [Integration Testing](https://www.guru99.com/integration-testing.html). And the functionality which makes it efficient testing framework are

* Support for annotations
* Support for data-driven testing
* Flexible test configuration
* Ability to re-execute failed test cases

#### 22) Mention what is the difference between Implicit wait and Explicit wait?

Implicit Wait: Sets a timeout for all successive Web Element searches. For the specified amount of time it will try looking for element again and again before throwing a NoSuchElementException.  It waits for elements to show up.

Explicit Wait :  It is a one-timer, used for a particular search.

#### 23) Which attribute you should consider throughout the script in frame for “if no frame Id as well as no frame name”?

You can use…..driver.findElements(By.xpath(“//iframe”))….

This will return list of frames.

You will need to  switch to each and every frame and search for locator which we want.

Then break the loop

#### 24) Explain what is the difference between find elements () and find element () ?

find element ():

It finds the first element within the current page using the given “locating mechanism”.  It returns a single WebElement

findElements () : Using the given “locating mechanism” find all the elements within the current page.  It returns a list of web elements.

#### 25) Explain what are the JUnits annotation linked with Selenium?

The JUnits annotation linked with Selenium are

* @Before public void method() – It will perform the method () before each test, this method can prepare the test
* @Test public void method() – Annotations @Test identifies that this method is a test method environment
* @After public void method()- To execute a method before this annotation is used, test method must start with test@Before

#### 26) Explain what is Datadriven framework and Keyword driven?

**Datadriven framework:**  In this framework, the test data is separated and kept outside the Test Scripts, while[Test Case](https://www.guru99.com/test-case.html)logic resides in Test Scripts.  Test data is read from the external files ( Excel Files) and are loaded into the variables inside the Test Script.  Variables are used for both for input values and for verification values.

**Keyworddriven framework:** The keyword driven frameworks requires the development of data tables and keywords, independent of the test automation.  In a keyword driven test, the functionality of the application under test is documented in a table as well as step by step instructions for each test.

#### 27) Explain how you can login into any site if it’s showing any authentication popup for password and username?

Pass the username and password with url

* Syntax-http://username:password@url
* ex- http://creyate:tom@www.gmail.com

#### 28) Explain how to assert text of webpage using selenium 2.0 ?

WebElement el = driver.findElement(By.id(“ElementID”))

//get test from element and stored in text variable

String text = el.getText();

//assert text from expected

Assert.assertEquals(“Element Text”, text);

#### 29) Explain what is the difference between Borland Silk and Selenium?

|  |  |
| --- | --- |
| **Silk Test Tool** | **Selenium Test Tool** |
| * Borland Silk test is not a free testing tool | * Selenium is completely free test automation tool |
| * Silk test supports only Internet Explorer and Firefox | * Selenium supports many browsers like Internet Explorer, Firefox, Safari, Opera and so on |
| * Silk test uses test scripting language | * Selenium suite has the flexibility to use many languages like Java, Ruby,Perl and so on |
| * Silk test can be used for client server applications | * Selenium can be used for only web application |

#### 30) What is Object Repository ?

An object repository is an essential entity in any UI automations which allows a tester to store all object that will be used in the scripts in one or more centralized locations rather than scattered all over the test scripts.

#### 31) Explain how Selenium Grid works?

[Selenium Grid](https://www.guru99.com/introduction-to-selenium-grid.html) sent the tests to the hub. These tests are redirected to Selenium Webdriver, which launch the browser and run the test.  With entire test suite, it allows for running tests in parallel.

#### 32) Can we use Selenium grid for performance testing?

Yes. But not as effectively as a dedicated [Performance Testing](https://www.guru99.com/performance-testing.html) tool like Loadrunner.

#### 33)  List the advantages of Webdriver over Selenium Server?

* If you are using Selenium-WebDriver, you don’t need the Selenium Server as it is using totally different technology
* Selenium Server provides Selenium RC functionality which is used for Selenium 1.0 backwards compatibility
* Selenium Web driver makes direct calls to browser using each browsers native support for automation, while Selenium RC requires selenium server to inject Javascript into the browser

#### 34)  Mention what are the capabilities of Selenium WebDriver or Selenium 2.0 ?

WebDriver should be used when requiring improvement support for

* Handling multiple frames, pop ups , multiple browser windows and alerts
* Page navigation and drag & drop
* Ajax based UI elements
* Multi browser testing including improved functionality for browser not well supported by Selenium 1.0

#### 35)  While injecting capabilities in webdriver to perform tests on a browser which is not supported by a webdriver what is the limitation that one can come across?

Major limitation of injecting capabilities is that “findElement” command may not work as expected.

#### 36)  Explain how you can find broken images in a page using Selenium Web driver ?

To find the broken images in a page using Selenium web driver is

* Get XPath and get all the links in the page using tag name
* In the page click on each and every link
* Look for 404/500 in the target page title

#### 37)  Explain how you can handle colors in web driver?

To handle colors in web driver you can use

Use getCssValue(arg0) function to get the colors by sending ‘color’ string as an argument

#### 38)  Using web driver how you can store a value which is text box?

You can use following command to store a value which is text box using web driver

driver.findElement(By.id(“your Textbox”)).sendKeys(“your keyword”);

#### 39)  Explain how you can switch between frames?

To switch between frames webdrivers **[ driver.switchTo().frame() ]** method takes one of the three possible arguments

* A number:  It selects the number by its (zero-based) index
* A name or ID: Select a frame by its name or ID
* Previously found WebElement: Using its previously located WebElement select a frame

#### 40)  Mention 5 different exceptions you had in Selenium web driver?

The 5 different exceptions you had in Selenium web drivers are

* WebDriverException
* NoAlertPresentException
* NoSuchWindowException
* NoSuchElementException
* TimeoutException

#### 41)  Explain using Webdriver how you can perform double click ?

You can perform double click by using

* **Syntax- Actions act = new Actions (driver);**
* **act.doubleClick(webelement);**

#### 42) How will you use  Selenium to upload a file ?

You can use “type”command to type in a file input box of upload file. Then, you have to use “Robot” class in JAVA to make file upload work.

#### 43) Which web driver implementation is fastest?

HTMLUnit Driver implementation is fastest, HTMLUnitDriver does not execute tests on browser but plain http request, which is far quick than launching a browser and executing tests

#### 44) Explain how you can handle frames using Selenium 2.0 ?

To bring control on HTML frame you can use “SwitchTo” frame method-

driver.switchTo().frame(“frameName”);

To specify a frame you can use index number

driver.switchTo().frame(“parentFrame.4.frameName”);

This would bring control on frame named- “frameName” of the 4th sub frame names “parentFrame”

#### 45) What is the difference between getWindowhandles() and getwindowhandle() ?

getwindowhandles(): It is used to get the address of all the open browser and its return type is Set<String>

getwindowhandle(): It is used to get the address of the current browser where the control is and return type is string

#### 46) Explain how you can switch back from a frame?

To switch back from a frame use method defaultContent()

Syntax-driver.switchTo().defaultContent();

#### 47) List out different types of locators?

Different types of locators are

* By.id()
* By.name()
* By.tagName()
* By.className()
* By.linkText()
* By.partialLinkText()
* By.xpath
* By.cssSelector()

#### 48) What is the command that is used in order to display the values of a variable into the output console or log?

* In order to display a constant string, command can be used is echo <constant string>
* If order to display the value of a variable you can use command like echo ${variable name>>

Above is using PHP. If you are using Java, replace echo with System.out.println

#### 49)   Explain how you can use recovery scenario with Selenium?

Recovery scenarios depends upon the programming language you use.  If you are using Java then you can use exception handling to overcome same.  By using “Try Catch Block” within your Selenium WebDriver Java tests

#### 50) Explain how to iterate through options in test script?

To iterate through options in test script you can loop features of the programming language, for example to type different test data in a text box you can use “for” loop in Java

// test data collection in an array

String[ ] testData = { “test1” , “test2” , “test3” } ;

// iterate through each test data

For  (string s: test data) { selenium.type ( “elementLocator”, testData) ; }

#### 51) How can you prepare customized html report using TestNG in hybrid framework ?

There are three ways

* Junit: With the help of ANT
* TestNG: Using inbuilt default.html to get the HTML report. Also XST reports from ANT, Selenium,[Testng](https://www.guru99.com/all-about-testng-and-selenium.html)combinations
* Using our own customized reports using XSL jar for converting XML content to HTML

#### 52) From your test script how you can create html test report?

To create html test report there are three ways

* TestNG:  Using inbuilt default.html to get the HTML report. Also XLST reports from ANT, Selenium, TestNG combination
* JUnit: With the help of ANT
* Using our own customized reports using XSL jar for converting XML content to HTML

#### 53)  Explain how you can insert a break point in Selenium IDE ?

In [Selenium IDE](https://www.guru99.com/introduction-selenuim-ide.html) to insert a break point

* Select “Toggle break point” by right click on the command in Selenium IDE
* Press “B” on the keyboard and select the command in Selenium IDE
* Multiple break points can be set in Selenium IDE

#### 54)  Explain in Selenium IDE how can you debug the tests?

* Insert a break point from the location from where you want to execute test step by step
* Run the test case
* At the given break point execution will be paused
* To continue with the next statement click on the blue button
* Click on the “Run” button to continue executing all the commands at a time

#### 55)  What is Selenese and what are the types of Selenese ?

Selenese is a selenium set of command which are used for running the test

There are three types of Selenese

* Actions: It is used for performing the operations and interactions with the target elements
* Assertions: It is used as a check points
* Accessors: It is used for storing the values in a variable

#### 56)  Explain what are the limitations of Selenium IDE?

The limitations of Selenium IDE

* Exceptional handling is not present
* Selenium IDE uses only HTML languages
* External databases reading is not possible with IDE
* Reading from the external files like .txt, .xls is not possible
* Conditional or branching statements execution like if,else,  select statements is not possible

#### 57)  What are the two modes of views in Selenium IDE ?

Either Selenium IDE can be opened as a pop up window or in side bar

#### 58)  In selenium IDE what are the element locators that can be used to locate elements on web page?

In selenium there are mainly 4 locators that are used

* X-path locators
* Css locators
* Html id
* Html name

#### 59)  In Selenium IDE how you can generate random numbers and dates for test data ?

In Selenium IDE you can generate random numbers by using Java Script

type

css=input#s

javascript{Math.random()}

And for

type

css=input#s

javascript{new Date()}

#### 60)  How you can convert any Selenium IDE tests from Selenese to another language?

You can use the format option of Selenium IDE to convert tests into another programming language

#### 61)  Using Selenium IDE is it possible to get data from a particular html table cell ?

You can use the “storeTable” command

Example store text from cell 0,2 from an html table

storeTable

Css=#table 0.2

textFromCell

#### 62)  Explain what can cause a Selenium IDE test to fail?

* When a locator has changed and Selenium IDE cannot locate the element
* When element Selenium IDE waiting to access did not appear on the web page and the operation timed out
* When element Selenium IDE was trying to access was not created

#### 63)  Explain how you can debug the tests in Selenium IDE ?

* Insert a break point from the location where you want to execute step by step
* Run the test case
* At the given break point execution will be paused
* To continues with the next step click on the Blue button
* To run commands at a time click on run button

#### 64)  From Selenium IDE how you can execute a single line?

From Selenium IDE single line command can be executed in two ways

* Select “Execute this command” by right clicking on the command in Selenium IDE
* Press “X” key on the keyboard after selecting the command in Selenium IDE

#### 65)  In which format does source view shows your script in Selenium IDE ?

In Selenium IDE source view shows your script in XML format

#### 66)  Explain how you can insert a start point in Selenium IDE?

In two ways selenium IDE can be set

* Press “S” key on the keyboard and select the command in Selenium IDE
* In Seleniun IDE right click on the command and the select  “Set / Clear Start Point”

#### 67)  What if you have written your own element locator and how would you test it?

To test the locator one can use “Find Button” of Selenium IDE, as you click on it, you would see on screen an element being highlighted provided your element locator is right or or else an error message will be displayed

#### 68)  What is regular expressions? How you can use regular expressions in Selenium ?

A regular expression is a special text string used for describing a search pattern. In Selenium IDE regular expression can be used with the keyword- **regexp:**as a prefix to the value and patterns needs to be included for the expected values.

#### 69)  What are core extension ?

If you want to “extend” the defualt functionality provided by Selenium Function Library , you can create a Core Extension. They are also called “User Extension”. You can even download ready-made Core Extension created by other Selenium enthusiats.

#### 70)  How will you handle working with multiple windows in Selenium ?

We can use the command **selectWindow**to switch between windows. This command uses the title of Windows to identify which window to switch to.

#### 71)  How will you verify the specific position of an web element

You can use verifyElementPositionLeft & verifyElementPositionTop. It does a pixel comparison of the position of the element from the Left and Top of page respectively

#### 72)  How can you retrive the message in an alert box ?

You can use the storeAlert command which will fetch the message of the alert pop up and store it in a variable.

#### 73) What is selenium RC (Remote Control)?

Selenium IDE have limitations in terms of browser support and language support. By using Selenium RC limitation can be diminished.

* On different platforms and  different web browser for automating web application  selenium RC is used with languages like Java, C#, Perl, Python
* Selenium RC is a java based and using any language it can interact with the web application
* Using server you can bypass the restriction and run your automation script running against any web application

#### 74) Why Selenium RC is used?

Selenium IDE does not directly support many functions like condition statements, Iteration, logging and reporting of test results, unexpected error handling and so on as IDE supports only HTML language.  To handle such issues Selenium RC is used  it supports the language like Perl, Ruby, Python,[PHP](https://www.guru99.com/php-tutorials.html)using these languages we can write the program to achieve the IDE issues.

#### 75) Explain what is the main difference between web-driver and RC ?

The main difference between Selenium RC and Webdriver is that, selenium RC injects javascript function into browsers when the page is loaded. On the other hand, Selenium Webdriver drives the browser using browsers built in support

#### 76) What are the advantages of RC?

Advantages of RC are

* Can read or write data from/ to .xls, .txt, etc
* It can handle dynamic objects and Ajax based UI elements
* Loops and conditions can be used for better performance and flexibility
* Support many Programming languages and Operating Systems
* For any JAVA script enabled browser Selenium RC can be used

#### 77) Explain what is framework and what are the frameworks available in RC?

A collection of libraries and classes is known as Framework and they are helpful when testers has to automate test cases. NUnit, JUnit, TestNG, Bromine, RSpec, unittest are some of the frameworks available in RC .

#### 78) How can we handle pop-ups in RC ?

To handle pop-ups in RC , using selectWindow method, pop-up window will be selected and windowFocus method will let the control from current window to pop-up windows and perform actions according to script

#### 79) What are the technical limitations while using Selenium RC?

Apart from “same origin policy” restriction from js, Selenium is also restricted from exercising anything that is outside browser.

#### 80) Can we use Selenium RC to drive tests on two different browsers on one operating system without Selenium Grid?

Yes, it is possible when you are not using JAVA testing framework.  Instead of using Java testing framework if you are using java client driver of selenium then TestNG allows you to do this.  By using “parallel=test” attribute you can set tests to be executed in parallel and can define two different tests, each using different browser.

#### 81) Why to use TestNG with Selenium RC ?

If you want full automation against different server and client platforms, You need a way to invoke the tests from a command line process, reports that tells you what happened and flexibility in how you create your test suites. TestNG gives that flexibility.

#### 82) Explain how you can capture server side log Selenium Server?

To capture server side log in Selenium Server, you can use command

* java –jar .jar –log selenium.log

#### 83) Other than the default port 4444 how you can run Selenium Server?

You can run Selenium server on java-jar selenium-server.jar-port other than its default port

#### 84) How Selenium grid hub keeps in touch with RC slave machine?

At predefined time selenium grid hub keeps polling all RC slaves to make sure it is available for testing.  The deciding parameter is called “remoteControlPollingIntervalSeconds” and is defined in “grid\_configuration.yml”file

#### 85) Using Selenium how can you handle network latency ?

To handle network latency you can use driver.manage.pageloadingtime for network latency

#### 86) To enter values onto text boxes what is the command that can be used?

To enter values onto text boxes we can use command **sendkeys()**

#### 87) How do you identify an object using selenium?

To identify an object using Selenium you can use

isElementPresent(String locator)

isElementPresent takes a locator as the argument and if found returns a Boolean

#### 88) In Selenium what are Breakpoints and Startpoints?

* **Breakpoints:**When you implement a breakpoint in your code, the execution will stop right there. This helps you to verify that your code is working as expected.
* **Startpoints**Startpoint indicates the point from where the execution should begin. Startpoint can be used when you want to run the testscript from the middle of the code or a breakpoint.

#### 89) Mention why to choose Python over Java in Selenium?

Few points that favor[Python](https://www.guru99.com/python-tutorials.html)over Java to use with Selenium is,

* Java programs tend to run slower compared to Python programs.
* Java uses traditional braces to start and ends blocks, while Python uses indentation.
* Java employs static typing, while Python is dynamically typed.
* Python is simpler and more compact compared to Java.

#### 90) Mention what are the challenges in Handling Ajax Call in Selenium Webdriver?

The challenges faced in Handling Ajax Call in Selenium Webdriver are

* Using “pause” command for handling Ajax call is not completely reliable. Long pause time makes the test unacceptably slow and increases the testing time. Instead, “waitforcondition” will be more helpful in testing Ajax applications.
* It is difficult to assess the risk associated with particular Ajax applications
* Given full freedom to developers to modify Ajax application makes the testing process challenging
* Creating automated test request may be difficult for testing tools as such AJAX application often use different encoding or serialization technique to submit POST data.

#### 91) Mention what is IntelliJ?

Intellij is an IDE that helps you to write better and faster code for Selenium. Intellij can be used in the option to Java bean and Eclipse.

#### 92) Mention in what ways you can customize TestNG report?

You can customize TestNG report in two ways,

* Using ITestListener Interface
* Using IReporter Interface

#### 93) To generate pdf reports mention what Java API is required?

To generate pdf reports, you need Java API IText.

#### 94) Mention what is Listeners in Selenium WebDriver?

In Selenium WebDriver, Listeners “listen” to the event defined in the selenium script and behave accordingly. It allows customizing TestNG reports or logs. There are two main listeners i.e. WebDriver Listeners and TestNG Listeners.

#### 95) Mention what are the types of [Listeners in TestNG](https://www.guru99.com/listeners-selenium-webdriver.html)?

The types of Listeners in TestNG are,

* IAnnotationTransformer
* IAnnotationTransformer2
* IConfigurable
* IConfigurationListener
* IExecutionListener
* IHookable
* IInvokedMethodListener
* IInvokedMethodListener2
* IMethodInterceptor
* IReporter
* ISuiteListener
* ITestListener

#### 96) Mention what is desired capability? How is it useful in terms of Selenium?

The desired capability is a series of key/value pairs that stores the browser properties like browser name, browser version, the path of the browser driver in the system, etc. to determine the behavior of the browser at run time.

For Selenium,

* It can be used to configure the driver instance of Selenium WebDriver.
* When you want to run the test cases on a different browser with different operating systems and versions.

#### 97) For Database Testing in Selenium Webdriver what API is required?

For [Database Testing](https://www.guru99.com/data-testing.html) in Selenium Webdriver, you need JDBC (Java Database Connectivity) API. It allows you to execute[SQL](https://www.guru99.com/sql.html)statements.

#### 98) Mention when to use AutoIT?

Selenium is designed to automate web-based applications on different browsers. But to handle window GUI and non-HTML popups in the application you need AutoIT. know more about [How to use AutoIT with Selenium](https://www.guru99.com/use-autoit-selenium.html)

#### 99) Mention why do you need Session Handling while working with Selenium?

While working with Selenium, you need Session Handling. This is because, during test execution, the Selenium WebDriver has to interact with the browser all the time to execute given commands. At the time of execution, it is also possible that, before current execution completes, someone else starts execution of another script, in the same machine and in the same type of browser. So to avoid such situation you need Session Handling.

#### 100) Mention what are the advantages of Using Git Hub For Selenium?

The advantages of Using Git Hub for Selenium are

* Multiple people when they work on the same project they can update project details and inform other team members simultaneously.
* Jenkins can help you to build the project from the remote repository regularly. This helps you to keep track of failed builds.

# SimpliLearn: 31Q

### 1. 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.

### 2. What are the limitations of Selenium testing?

1. Unavailability of reliable tech support: Since Selenium is an open-source tool, it does not have dedicated tech support to resolve the user queries.
2. Tests web applications only: Selenium needs to be integrated with third-party tools like Appium and TestNG to test desktop and mobile applications.
3. Limited support for image testing.
4. 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.
5. May require the knowledge of programming languages: Selenium WebDriver expects the user to have some basic knowledge about programming.

#### Free Course: JavaScript for Beginners

Learn the Basics of JavaScript[ENROLL NOW](https://www.simplilearn.com/learn-javascript-basics-free-course-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-tutorial-skillup-course-promotion)

Free Course: JavaScript for Beginners

### 3. 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

### 4. 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

https://www.simplilearn.com/ice9/free_resources_article_thumb/selenium2.JPG

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.

### 5. 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.

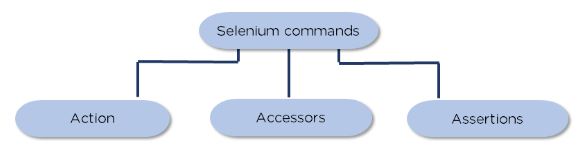
#### Free Course: Introduction to Selenium

Master the Basics of Selenium[ENROLL NOW](https://www.simplilearn.com/learn-selenium-basics-free-course-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-tutorial-skillup-course-promotion)

Free Course: Introduction to Selenium

### 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. 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”));

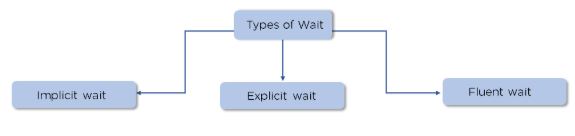
**Locator by XPath:** It takes a parameter of String which is a XPATHEXPRESSION and it returns an object to findElement() method.

  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”);

### 8. 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);

### 9. 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.

#### Free Course: Intro to RPA

Get closer to your dream role with the FREE course[START LEARNING](https://www.simplilearn.com/learn-rpa-basics-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-tutorial-skillup-course-promotion)

Free Course: Intro to RPA

### 10. 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.

### 11. 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.

## Intermediate Level Selenium Interview Questions

### 12. 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](mailto: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

### 13. 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.

#### Robotic Process Automation using UiPath Training

Build RPA Expertise Using the UiPath Platform[VIEW COURSE](https://www.simplilearn.com/robotic-process-automation-using-uipath-training-course?source=GhPreviewCTABanner)

Robotic Process Automation using UiPath Training

### 14. 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

### 15. 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);

### 16. 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();

Master important testing concepts such as TestNG, Selenium IDE, Selenium Grid, Selenium WebDriver. Check out [Selenium Certification Training](https://www.simplilearn.com/selenium-certification-training). Enroll now!

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

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

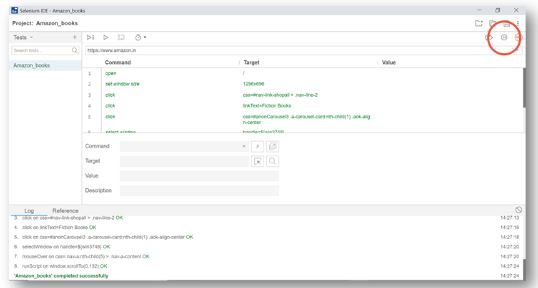
**General Syntax:**

  driver.findElement(By.id(“id“)).getCssValue(“name of css attribute”);

**Example:**

   driver.findElement(By.id(“email“)).getCssValue(“font-size”);

### 18. 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.

### 19. 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.

### 20. How does Selenium handle Windows-based pop-ups?

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.

### 21. 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);

#### Stand Out From Your Peers this Appraisal Season

Start Learning With Our FREE Courses[ENROLL NOW](https://www.simplilearn.com/skillup-free-online-courses?utm_source=frs&utm_medium=skillup-home-banner&utm_campaign=frs-tutorial-skillup-home-promotion)

Stand Out From Your Peers this Appraisal Season

## Advanced Level Selenium Interview Questions

### 22. 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”);

### 23. 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”);

### 24. 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); }

### 25. 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");

### 26. 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)");

### 27. 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:

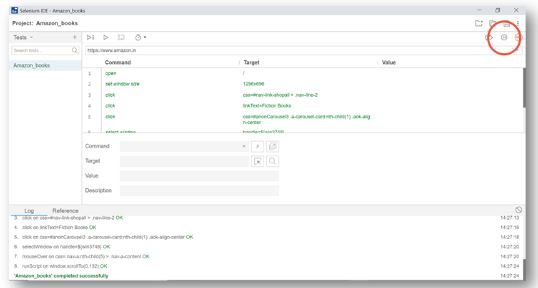
   WebElement element = driver.findElement(By.id(example));

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:

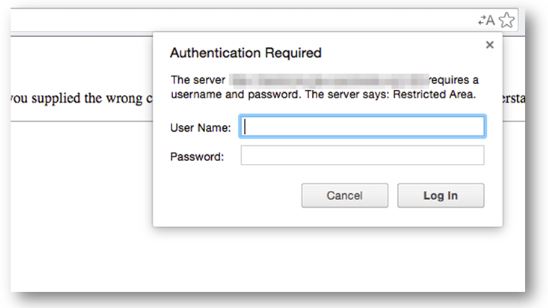
   List <WebElement> elementList = driver.findElements(By.id(example));

### 28. 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.

### 29. 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>));

### 30. 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

### 31. 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

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

   HttpURLConnection httpConn =(HttpURLConnection)link.openConnection();

Use getResponseCode () to get response code

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

Through exception, if any error occurred

    System.out.println(“Broken Link”);

# 4.MyGreatLearning 50Q: <https://www.mygreatlearning.com/blog/selenium-interview-questions/>

### ****1. What is Automation Testing?****

The use of automation tools for writing and executing the test cases with no manual involvement for executing the Automated Test Suite is called Automation Testing. The Automation tools are preferred by Testers to write the test cases or test scripts and then group them all into Test Suites. These tools are designed to execute manually designed test cases without any human intervention.

### ****2. What are the advantages/benefits of Automation Testing?****

This is one of the common interview questions. Some of the advantages of Automation Testing include:

* Save Money and Time.
* Automation Testing increases the accuracy as there is no chance for human errors.
* Code Reusability. Create once and execute multiple times with Less or no maintenance.
* Easy Reporting. Automatically generates the reports after the execution of the test cases.
* Easy for Compatibility Testing by enabling the parallel execution in combination with OS and browser environments.
* More Reliable, Powerful and Versatile.
* Automation Tools allows us to integrate with the Jenkins, GitHub etc.,
* Mostly used for [Regression Testing](https://www.mygreatlearning.com/blog/regression-testing/).
* Helps to monitor the Test Coverage and increase the test cases accordingly.

### ****3. How many Test cases have you automated per day?****

This is one of the tricky Selenium interview questions. Actually, it depends on the complexity and length of the Test case scenario. When the complexity is limited, you can automate two to five test cases and one or fewer test cases when the complexity is high in a day.

## ****Basic Selenium Interview Questions for Freshers****

### ****4. What is Selenium?****

Selenium is one of the most popular automated Testing Suite. It’s a playback tool to perform functional testing without any knowledge of a test scripting language. Selenium is the open-source, web UI-based automation testing suite. Jason Huggins developed this tool in 2004 at ThoughtWorks.

### ****5. What are the different Selenium suite Components?****

Selenium is not just a single tool or utility, rather a package of several testing tools and that’s why it’s called Selenium Suite. Each of these tools caters to different testing and testing environments.

It comprises four major components as follows,

1. **Selenium Integrated Development Environment (IDE):**

It is distributed as a Firefox plugin that acts as a record and playback tool. It is developed to speed up the creation of automation scripts.

1. **Selenium Remote Control (RC):**

Selenium Remote Control is a server with which the user can create the Test cases in any desired [programming language](https://www.mygreatlearning.com/academy/learn-for-free/courses/programming-fundamentals/?gl_blog_id=25305). The commands present in the test scripts are accepted by the server and sent to the browsers as the Selenium core Java script commands to maintain the browser behavior accordingly.

1. **Selenium Web driver:**

Selenium Web Driver is a programming interface to create and run test cases. It provides a way to act on various web elements. This Web driver does not require another server, unlike Selenium Remote Control.

1. **Selenium Grid:**

The Selenium Grid is used to distribute the commands to the different machines simultaneously. It paves a way for the parallel execution of the test cases in different browsers and different environments. It achieves concurrency and integrity in the Test Suite Execution.

### ****6. Why should I use Selenium?****

* Open-source
* Large support from the community and the user base is huge
* Cross-browser compatibility (Firefox, Chrome, Edge etc.,)
* Multiple programming language support ([Python](https://www.mygreatlearning.com/academy/learn-for-free/courses/python-fundamentals-for-beginners?gl_blog_id=25305), [Java](https://www.mygreatlearning.com/blog/java-tutorial-for-beginners/), Perl etc.,)
* Every day or regular repository developments
* Distributed Testing is also supported.

### ****7. What is the major difference between Selenium 3.0 and Selenium 2.0?****

Selenium 2.0 represents the merger of the Original Selenium project with the Web Driver project. The Selenium RC got deprecated but was used for backward compatibility.

Selenium 3.0 often called an extension of the Selenium 2.0 is inherently backward compatible with increased stability and several bug fixes and does not involve the Selenium RC at all.

### ****8. What do you mean by Selenese?****

Selenese which is called the Selenium command is the set of the selenium commands that run your test cases. For example, open (URL) is a Selenium command which opens the specified URL in the specific browser. A test script is the sequence of all these commands put together. There are three types of Selenese. They are,

1. Actions: We can use it to perform interactions and operations on the target elements
2. Accessors: These are used for storing the values in the variables.
3. Assertions: These can be used as checkpoints to keep track of the sequential flow of commands.

### ****9. How many parameters do selenium commands have at a minimum?****

There are four parameters that you must pass in Selenium. They are as follows,

* **Host**

We can bind the Selenium to a specific IP address instead of a Local host in our local machine.

* **Port Number**

We can specify the port number too i.e., TCP/IP port to connect the Selenium tests to the Grid hub. We need to make sure that there is no other application with the same port number up in any other location within the same machine.

* **Browser**

The browser in which the selenium scripts have to be executed.

* **URL**

The application URL triggers the test cases within that.

### ****10. What is the difference between Absolute path and Relative Path?****

**Absolute Path:**

* This path starts from the root node and ends with the desired elements node.
* It starts with a single slash (/)
* Example: /html.body/div/td/input

**Relative Path:**

* This path starts from any node in between and to the desired elements node.
* It starts with a double slash (//)
* Example: //input/example[@id=name]

### ****11. What is the difference between Assert and Verify statements in Selenium?****

**Assert:**

* We can check whether an element is on the page or not.
* The test fails and gets terminated whenever there is a fail in the check. That is the program control flow stops.

**Verify:**

* It’s only to specify whether the given condition is true or false.
* The program control flow does not stop due to this.

Overall, the Verify statement does not halt the program execution whereas the Assert statement does halt the execution.

### ****12. What is the difference between findElement() and findElements()?****

**findElement() :**

It uses the given locating mechanism to find the first element within the current page and return a single web element.

**findElements():**

It also uses the given locating mechanism to find all the elements within the current page and return the list of all web elements.

### ****13. What are the verification points that are available in Selenium?****

* Selenium IDE: Verify and Assert commands
* Selenium Web Driver: This does not have any built-in features for verification and always depends on our coding style. Some examples can be like checking the page title, checking for certain text in the page, checking for a certain element like textbox, dropdown, field etc.,

### ****14. Explain Implicit wait, Explicit wait and Fluent wait?****

* **Implicit wait:**

We can set the timeout for a specific amount of time for all the successive web elements. In this specified time, the web driver searches for all the web elements before throwing the NoSuchElementException.

* **Explicit wait:**

We can tell the Web Driver to wait for certain conditions before throwing the ElemntNotVisibleException.

* **Fluent wait:**

It’s a slight extension for the explicit wait. Apart from waiting for certain conditions to be met, we can also set the frequency with which we check the condition before throwing the ElementNotVisibleException.

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

**driver.close():**

It’s a command by which the current window can be closed. For example, if we have multiple browser windows open then, by using this command we can close the window with which the focus is.

**driver.quit():**

This command closes all the browser windows which are open. For example, if we have multiple browser windows open then, this command can close all the windows at once.

### ****16. Can Selenium handle windows-based pop-up?****

As we already know that Selenium is an automation testing tool that can work with only web applications. Therefore, the pop-up in windows cannot be handled by Selenium. But, by integrating with the third-party tools we can overcome this problem.

### ****17. What is an Object Repository and Explain how we can create the Object Repository in Selenium?****

The collection of web elements that belongs to the application under test along with their locator values. This is often referred to as Object Repository. The only thing is the locators are stored in a centralized location than hard coding them in the test scripts.

While coming to Selenium, the objects can be stored in Excel also and can be used to populate them in the script whenever required.

### ****18. What are the different kinds of frameworks?****

* **Data-driven framework:**

The test data of this kind is generated from external files like excel, CSV, XML, etc. The data relating to the test cases rely outside on the input and verification values maintained using the variables.

* **Keyword-driven framework:**

The data tables and the keywords are developed independently of the automation test cases. In this framework, the functionality of the test scenarios is documented in the table step by step.

* **Module-based testing framework:**

Isolated and Logical modules are made out of the application that is under test. Independent test scripts are written for each module. These scripts are then clubbed together as a whole.

* **Behaviour driven testing framework:**

The test scripts are written in such a way that the automation is done on the functionality or features to ensure the readability, understanding, and reliability to either the business experts or developers, or other testers.

* **Hybrid testing framework:**

The combination of more than one of the above-mentioned frameworks by leveraging the benefits of each is called a Hybrid testing framework.

### ****19. What is the same-origin policy? How can we avoid it?****

This policy is introduced for security reasons.

* By this policy, it ensures that we cannot access the script from another site.
* The source code which is loaded within the browser can only be operated within that website’s domain.
* This policy can be avoided by using the proxy-injection method. In this method, the browser is tricked using the Selenium’s server to be a real URL (fictional URL).

### ****20. What is the difference between the getwindowhandle() and getwindowhandles()?****

getwindowhandles(): This is used to get the address of all the open browsers and returns the data type of Set<String>.

getwindowhandle(): This is used to get the address of the current browser window where it’s focused on and returns the data type of String.

### ****21. What are the different types of Annotations used in Selenium?****

Syntactic metadata can be added to the Java source code which is known as annotations. They can be added on variables, parameters, packages, classes etc., Some of them which are used in Selenium include,

@Before, @Test, @After

### ****22. What is the difference between setSpeed() and sleep() methods?****

These methods will delay the speed of the execution.

**Thread.sleep():** The current thread will stop for a specified period of time. It only waits once when the command is given. It takes only a single argument that is Integer format.

**set sleep():** This command will stop the execution for every selenium command. This command is used only for demonstration purposes and also for a slow web application.

### ****23. List the advantages of Web driver over Selenium server?****

* If Selenium web driver is used, then no need to use the dependency of the selenium server.
* Selenium server provides us with the functionality of Selenium RC for the backward compatibility.
* Selenium server acts as a middleware for the communication between browsers and the application. Whereas, the Selenium web driver directly calls the browser to support automation.

## ****Selenium Interview Questions for Experienced Professionals****

### ****24. How can you store a value in a text box?****

A command which can store the value from the text box using a web driver.

For example,

driver.findElement(By.id(“your Textbox”)).sendKeys(“your keyword”);

### ****25. How can you debug the tests in the Selenium IDE?****

* We need to first insert the breakpoints where we need to debug and execute the test step by step.
* Later, Run the test case.
* After this, the execution stops at the point where the breakpoint is kept.
* You can click on the Run button to continue the execution of all the commands sequentially.

### ****26. What are regular expressions and How can you use regular expressions in Selenium?****

The Search patterns are represented using the special Text string called the regular expressions. Regexp is the keyword that can be used as a prefix to treat a text in Selenium as a regular expression.

### ****27. How can you handle working with multiple windows in Selenium?****

selectWindow() command in Selenium can be used to switch among the working windows. The distinguishing factor among all the windows is the title of the window.

### ****28. How can the message in the alert box be retrieved?****

storeAlert() command can be used to retrieve the message from the alert pop-up and store it in a variable.

### ****29. How can you verify the specific position of a web element?****

verifyElementPositionLeft and verifyElementPositionTop commands are used. These use the pixel comparison by identifying the position of the element from the left and top of the web page respectively.

### ****30. Why do we use Selenium RC?****

* Selenium IDE has the limitations which Selenium RC can diminish in terms of browser support and language support.
* IDE does not support many functions like Logging, iterations, conditional statements, test case results reporting etc., as it supports only HTML Language.
* That’s why the Selenium RC is used to communicate with the other languages like Perl, Python etc.,

### ****31. What are the advantages or Benefits of Selenium RC?****

Advantages or Benefits of RC include,

* Data can be read or write to or from .xlsx, .csv, .txt etc.,
* Dynamic objects and Ajax based UI elements can be handled.
* Conditional statements and Iterations functionalities can be used to improve and enhance performance and flexibility.
* Support for any programming languages and Operating systems.
* Selenium RC can be used on any browser which is enabled with [Java script](https://www.mygreatlearning.com/blog/javascript-tutorial/).

### ****32. Can you list the Technical limitations while using the Selenium RC?****

* Same origin policy
* Anything outside the browser is also restricted to access in the test scenario execution.

### ****33. What is the need to use the TestNG with Selenium RC?****

If the requirement of the automation test suite is supposed to work against the different servers and client platforms then, we need to invoke the test cases from the command line. Also, we do need the test case results reports periodically to make estimates and also implement the flexibility on how to create the tests. Such kind of flexibility is provided by the TestNG.

### ****34. Which Language do you use to implement test case suites in Selenium?****

The most preferred language to implement Selenium is [Python](https://www.mygreatlearning.com/academy/learn-for-free/courses/selenium-with-python/?gl_blog_id=25305) over Java. This is because,

* Java programs do run slower when compared to the [Python programs](https://www.mygreatlearning.com/blog/data-types-in-python-programming/).
* The implementation and the complexity of both are quite different and Python works well.
* Python is dynamically typed whereas Java is statically typed.
* Java is not simpler and compact to write test cases when compared to python.

### ****35. What are Breakpoints and Start points in Selenium?****

* **Break points:**

The flow of execution is stopped using the breakpoints. By doing this we can verify that the code is working in an expected manner or not.

* **Start points:**

The point from where the execution should begin is determined by the start points. When you want to run a test script from the middle of the code or a breakpoint, we use these.

### ****36. How is the desired capability useful in terms of Selenium?****

A series of key-value pairs can be used to store information like the browser name, browser version, the path of the browser driver in the system, etc., to determine the behavior of the browser at the run-time. This feature or functionality is commonly referred to as desired capability.

For Selenium,

The driver instance can be configured using this.

It comes into practice when you want to run the test cases in different [operating systems](https://www.mygreatlearning.com/blog/what-is-operating-system/), browsers, and versions.

### ****37. When do you use AutoIT?****

As we already know that Selenium is used only to automate web applications. But if we want to handle or manage or maintain the GUI, [HTML](https://www.mygreatlearning.com/blog/html-tutorial/) pop-ups then, we need the use of AutoIT.

### ****38. Can you mention why do you need the Session handling in Selenium?****

Session handling is very important while working with Selenium. This is because while working with the test case scenarios we need to establish the communication between the browser and the application while executing the commands. There can be an issue where while running a particular test script, another test script can be triggered within the same host and same type of browser. This is the most important part and the reason why we need the session handling.

### ****39. Can you automate CAPTCHA?****

The answer is No. We cannot automate CAPTCHA in Selenium. The concept of CAPTCHA is to ensure that the bots or automated test scripts should not have access to sensitive information, and that is why we cannot automate it. The captcha should be manually typed in order to continue the flow of execution of the test suite.

### ****40. How do we launch different browsers in Selenium?****

* A driver instance should be created for a particular browser which we need to work upon.
* WebDriver driver = new FireFoxDriver();
* WebDriver driver = new ChromeDriver();
* Here, the WebDriver is the interface where the Firefox, chrome and all other browser driver implementations are made.

### ****41. What is Locator and How do you locate an element in Selenium?****

The elements of the web page are found and matched using locators to interact with. There are different kinds of locators to identify different types of elements on the web page. They are as follows,

* ID
* Name
* Class
* Partial Link
* XPath
* CSS Selector
* Link Text

### ****42. Why do you choose Selenium over QTP (Quick Test Professional)?****

|  |  |  |
| --- | --- | --- |
| **Feature** | **Selenium** | **Quick Test Professional (QTP)** |
| Browser Compatibility | Almost all popular browsers are supported. Firefox, Chrome, Safari, Internet Explorer, Edge, etc. | This supports only Chrome, Firefox, and Internet explorer. Also, it supports only the Windows operating system. |
| Distribution | Distributed as open-source and it is freely available. | Distributed as a licensed tool and is commercialized. |
| Application Under Test (AUT) | Web-based applications are only supported in this kind of automated testing. | This provides support for both webs- based as well as windows applications for automated testing. |
| Object Repository | The separate entity in Selenium is used to create the object repository. | Automatic creation of the Object repository happens and is maintained too |
| Language Support | Multiple Programming languages like Java, Python, Perl, Ruby, C#, etc., are supported by this Selenium. | Only VBScript language is supported by this QTP. |
| Vendor Support | As we already know that Selenium is a free tool, we do not get vendor support in troubleshooting the issues we face. | Vendor support is guaranteed for the users using the QTP and they do get quick responses as QTP is commercialized and licensed. |

## ****Selenium Webdriver Interview Questions****

### ****43. What are the challenges and Limitations of the Selenium Web Driver?****

As we all know that the Selenium Web Driver is used to mimic the real user actions on the web in the browser. Some of the challenges of Selenium are as follows,

1. Not compatible to test the Windows or Desktop applications.
2. Also, cannot be used to test the Mobile applications.
3. No Built-in reporting and it must be integrated with the other tools to obtain the reports.
4. Support for Image Testing is limited.
5. The user should have prior programming knowledge.
6. Since it is open-source, there is no vendor support.

### ****44. What are the Testing types that are supported by Selenium?****

Regression Testing and Functional Testing are majorly supported by Selenium.

1. **Regression Testing:**

All the Test cases are executed repeatedly either partially or as a full to ensure that the existing functionalities are working as designed. It includes the following steps,

* **Re-testing:** All the test cases in the Test Suite are executed which proves to be time-consuming and expensive.
* **Regression Test case Selection:** The test cases are classified or categorized as features tests, integration tests etc., and some test cases are selected.
* Prioritization of Test cases: The test cases that are selected are prioritized based on the critical functionalities and business impact.

1. **Functional Testing:**

In this, all the functions in the application are verified with a specific requirement. This is primarily the black-box testing as it is not concerned about the source code of the application. It includes the following steps,

* Test Input is identified.
* Test Outcome is computed.
* The test case is executed
* The actual outcome and the test outcome are compared accordingly.

### ****45. Explain different types of Exceptions in Selenium Web driver?****

The Exceptions in the Selenium web driver are also similar to the exceptions in any other programming language like Java, C#, etc., Some of the most common exceptions are as follows,

* **TimeOutException:** This exception can be thrown when a particular operation cannot be completed in a specified or stipulated time.
* **NoSuchElementException:** When an element in the given set of attributes is not present on the web page then, this exception can be thrown.
* **ElementNotVisibleException:** There can be cases where a particular element can be present in the DOM (Document Object Model) but not visible on the web page. In such cases, this exception can be thrown.
* **StaleElementException:** When an element is either deleted or no longer present in the DOM then, this exception can be thrown. We generally face this exception when the element we are interacting with is either destroyed or recreated again.

### 46. Which API is used for database testing in the Selenium web driver?

JDBC (Java Database connectivity) is used for database testing in Selenium. This allows us to write the [SQL](https://www.mygreatlearning.com/blog/sql-tutorial-for-beginners/) (Structured Query Language) queries and execute them.

### ****47. Can you write a small code snippet to launch a Chrome browser in Web Driver?****

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | public class ChromeBrowserLaunchDemo {    public static void main(String[] args) {    //Creating a driver object referencing WebDriver interface  WebDriver driver;    //Setting the webdriver.chrome.driver property to its executable's location  System.setProperty("webdriver.chrome.driver", "/lib/chromeDriver/chromedriver.exe");    //Instantiating driver object  driver = newChromeDriver();    //Using get() method to open a webpage  driver.get("[http://javatpoint.com](http://javatpoint.com/)");    //Closing the browser  Driver.quit();  }    } |

### ****48. Can you mention the challenges you face when handling the Ajax calls in Selenium Web driver?****

The following are the challenges we face while handling the AJAX calls,

* Pause command in Ajax calls is not conventional and reliable in handling. Long pauses do cause the test case execution increases the time of execution. We can use the waitForCondition instead to test the [AJAX](https://www.mygreatlearning.com/blog/ajax-interview-questions/) applications.
* Assessing the risk associated with these applications is difficult.
* Encoding and Serialization in the AJAX applications make it difficult in developing the automation test case scenarios.

### ****49. Can you explain how can you handle colours in web driver?****

We can use the command getCssValue(arg0) to fetch the colors by sending the color as the argument.

### ****50. Which implementation of the web driver is the fastest?****

The web driver implementation HTML Unit Driver is the fastest of all. This is because this driver does not execute the tests in the browser but rather, executes only the plain HTTP which is quick faster as expected.

For a more detailed course, visit Great Learning Academy where you will find various [**courses on Machine Learning**](https://www.mygreatlearning.com/academy/learn-for-free/courses/basics-of-machine-learning-1?gl_blog_id=25305) for free.

Also Read [Top Common Interview Questions](https://www.mygreatlearning.com/blog/common-interview-questions/)

## ****Frequently Asked Questions on Selenium****

#### ****What are the important topics in selenium?****

* Selenium Webdriver 3. x.
* Difference between selenium versions like IDE, RC, WEB DRIVER
* Selenium classes/interface/methods hierarchy from top to bottom
* Selenium GRID implementation for local and remote nodes.
* Java Concepts in detail.
* TestNG Framework.
* Advanced Reporting.
* Page Object Model.
* Page Factory Framework.
* Data-Driven Framework.
* Using reflection in the Page Object framework for creating pages dynamically.
* Selenium usage with [Docker](https://www.mygreatlearning.com/academy/learn-for-free/courses/docker-for-intermediate-level).
* Selenium JSON wire protocol details.
* The Selenium integration with different tools like JMeter, cucumber, AutoIt, Jenkins.
* Selenium frameworks like page factory, hybrid etc.

#### ****What can Selenium be used for?****

Selenium is used for automation testing on various web browsers such as  Chrome, Mozilla, Firefox, Safari, and IE. With the help of a selenium web driver, one can easily automate browser testing.

#### ****What are selenium commands?****

|  |  |
| --- | --- |
| **WebDriver command** | **Usage** |
| get() | Launches a new browser and opensthe specified URLTakes a single string type parameter driver.get(); |
| getClass() | This command is used to retrieve the Class object driver.getClass(); |
| getCurrentUrl() | • This command is used to retrieve the URL of the webpage and doesn’t require any parameter. driver.getCurrentUrl(); |
| getPageSource() | • This command is used to retrieve the page source and doesn’t require any parameter and returns a string value boolean result = driver.getPageSource().contains(“String to find”); |
| getTitle() | • The command is used to retrieve the title of the webpage the user is currently working on.• The command doesn’t require any parameter and returns a trimmed string value String title = driver.getTitle(); |
| getText() | • This command is used to retrieve the inner text of the specified web element  String Text = driver.findElement(By.id(“Text”)).getText(); |
| getAttribute() | • The command is used to retrieve the value of the specified attribute and requires a single string parameterdriver.findElement(By.id(“findID”)).getAttribute(“value”); |
| getWindowHandle() | • This command is used in the situation when we have more than one window to deal with and the user can also switch back to the previous window if he/she desires. private String winHandleBefore;winHandleBefore = driver.getWindowHandle();driver.switchTo().window(winHandleBefore); |
| getWindowHandles() | • The command is similar to that of “getWindowHandle()” but here they deal with more than 2 windows only. |

#### ****What are the basics of selenium?****

Selenium was originally developed by Jason Huggins in 2004 as an internal tool as an open-source Web UI for automation testing. Selenium supports automation across different browsers such as  Internet Explorer, Mozilla Firefox, Google Chrome, and Safari, platforms such as Windows, [Linux](https://www.mygreatlearning.com/blog/linux-tutorial/), Solaris, and Macintosh, and programming languages such as C#, Java, Perl, PHP, Python and Ruby and now it is also popular with Java and c#.

Read More: [*Basics of Selenium*](https://www.mygreatlearning.com/academy/learn-for-free/courses/selenium-basics/?gl_blog_id=25305)

#### ****What are the disadvantages of selenium?****

* Can’t rely on technical Support – As it is open-source software, it is hard to rely on technical support.
* It supports Web-based applications only.
* Selenium takes more time to create test cases as it has a programming interface only.
* Difficult to set up Test Environment with tools like UFT, RFT, SilkTest etc.
* Limited support for Image Testing.
* New features may not work properly.
* No Test Tool integration for Test Management.
* No Built-in reporting facility.

#### ****Why is selenium so popular?****

* Transparency
* Platform Independent
* Quickens TTM and TTD
* Fosters Continuous Integration Efforts
* Visibility in End-to-End Testing
* Reduces Turnaround Time
* Integration With Other Tools

#### ****Is Selenium a framework?****

Yes, selenium is a software testing framework for web applications that allows you to automate web app testing. With languages like Java, Python, Ruby, C#, you can write test scripts to run against browsers and VMs.

#### ****Which language is best for selenium?****

Python and Ruby are the best binding languages for selenium as software may automatically load the driver if it is present in the same folder as your system or in the python path.  
**Others:**

* **Java:** Java remains the most popular programming language for test automation but not good as it is not a scripting language
* **JavaScript:** JavaScript is used for test automation by 35% of our users writing their tests in this language.

# 5.STH 50Q: <https://www.softwaretestinghelp.com/selenium-interview-questions-answers/>

**Q #1) What is Automation Testing?**

Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves the use of a separate testing tool which lets you create test scripts which can be executed repeatedly and doesn’t require any manual intervention.

**Q #2) What are the benefits of Automation Testing?**

Benefits of Automation testing are:

1. Supports execution of repeated test cases
2. Aids in testing a large test matrix
3. Enables parallel execution
4. Encourages unattended execution
5. Improves accuracy thereby reducing human-generated errors
6. Saves time and money

**Q #3) Why should Selenium be selected as a test tool?**

Selenium

1. is a free and open source
2. have a large user base and helping communities
3. have cross Browser compatibility (Firefox, Chrome, Internet Explorer, Safari etc.)
4. have great platform compatibility (Windows, Mac OS, Linux etc.)
5. supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
6. has fresh and regular repository developments
7. supports distributed testing

**Q #4) 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.

**Q #5) What are the testing types that can be supported by Selenium?**

Selenium supports the following types of testing:

1. Functional Testing
2. Regression Testing

**Q #6) What are the limitations of Selenium?**

Following are the limitations of Selenium:

* Selenium supports testing of only web-based applications
* Mobile applications cannot be tested using Selenium
* Captcha and Barcode readers cannot be tested using Selenium
* Reports can only be generated using third-party tools like TestNG or JUnit.
* As Selenium is a free tool, thus there is no ready vendor support through the user can find numerous helping communities.
* The user is expected to possess prior programming language knowledge.

**Q #7)** **What is the difference between Selenium IDE, Selenium RC, and WebDriver?**

| **Feature** | **Selenium IDE** | **Selenium RC** | **WebDriver** |
| --- | --- | --- | --- |
| Browser Compatibility | Selenium IDE comes as a Firefox plugin, thus it supports only Firefox | Selenium RC supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera. | WebDriver supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera. Also supports HtmlUnitDriver which is a GUI less or headless browser. |
| Record and Playback | Selenium IDE supports record and playback feature | Selenium RC doesn't supports record and playback feature. | WebDriver doesn't support record and playback feature |
| Server Requirement | Selenium IDE doesn't require any server to be started before executing the test scripts | Selenium RC requires server to be started before executing the test scripts. | WebDriver doesn't require any server to be started before executing the test scripts |
| Architecture | Selenium IDE is a Javascript based framework | Selenium RC is a JavaScript based Framework. | WebDriver uses the browser's native compatibility to automation |
| Object Oriented | Selenium IDE is not an object oriented tool | Selenium RC is semi object oriented tool. | WebDriver is a purely object oriented tool |
| Dynamic Finders (for locating web elements on a webpage) | Selenium IDE doesn't support dynamic finders | Selenium RC doesn't support dynamic finders. | WebDriver supports dynamic finders |
| Handling Alerts, Navigations, Dropdowns | Selenium IDE doesn't explicitly provides aids to handle alerts, navigations, dropdowns | Selenium RC doesn't explicitly provides aids to handle alerts, navigations, dropdowns. | WebDriver offers a wide range of utilities and classes that helps in handling alerts, navigations, and dropdowns efficiently and effectively. |
| WAP (iPhone/Android) Testing | Selenium IDE doesn't support testing of iPhone/Andriod applications | Selenium RC doesn't support testing of iPhone/Android applications. | WebDriver is designed in a way to efficiently support testing of iPhone/Android applications. The tool comes with a large range of drivers for WAP based testing. For example, AndroidDriver, iPhoneDriver |
| Listener Support | Selenium IDE doesn't support listeners | Selenium RC doesn't support listeners. | WebDriver supports the implementation of Listeners |
| Speed | Selenium IDE is fast as it is plugged in with the web-browser that launches the test. Thus, the IDE and browser communicates directly | Selenium RC is slower than WebDriver as it doesn't communicates directly with the browser; rather it sends selenese commands over to Selenium Core which in turn communicates with the browser. | WebDriver communicates directly with the web browsers. Thus making it much faster. |

**Q #8) When should I use Selenium IDE?**

Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. Selenium IDE is an ideal tool for a naïve user.

**Q #9) What is Selenese?**

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

**Q #10)** **What are the different types of locators in Selenium?**

The locator can be termed as an address that identifies a web element uniquely within the webpage. Thus, to identify web elements accurately and precisely we have [different types of locators in Selenium](https://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/):

* ID
* ClassName
* Name
* TagName
* LinkText
* PartialLinkText
* Xpath
* CSS Selector
* DOM

**Q #11)** **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.

**Q #12) 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.

**Q #13) 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.

**Q #14) 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.

**Q #15)** **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.

**Q #16) 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.

**Q #17) Which is the latest Selenium tool?**

WebDriver

**Q #18) 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();

**Q #19) What are the different types of Drivers available in WebDriver?**

The different drivers available in WebDriver are:

* FirefoxDriver
* InternetExplorerDriver
* ChromeDriver
* SafariDriver
* OperaDriver
* AndroidDriver
* IPhoneDriver
* HtmlUnitDriver

**Q #20) What are the different types of waits available in WebDriver?**

There are two [types of waits available in WebDriver](https://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):

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.

**Q #21)** **How to type in a textbox using Selenium?**

The user can use sendKeys(“String to be entered”) to enter the string in the textbox.

**Syntax:**  
WebElement username = drv.findElement(By.id(“Email”));  
// entering username  
username.sendKeys(“sth”);

**Q #22)** **How can you find if an element in displayed on the screen?**

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. isDisplayed()
2. isSelected()
3. isEnabled()

**Syntax:**

**isDisplayed():**  
**boolean** buttonPresence = driver.findElement(By.id(“gbqfba”)).isDisplayed();

**isSelected():**  
**boolean** buttonSelected = driver.findElement(By.id(“gbqfba”)).isSelected();

**isEnabled():**  
**boolean** searchIconEnabled = driver.findElement(By.id(“gbqfb”)).isEnabled();

**Q #23)** **How can we get a text of a web element?**

Get command is used to retrieve the inner text of the specified web element. The command doesn’t require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc displayed on the web pages.

**Syntax:**  
String Text = driver.findElement(By.id(“Text”)).getText();

**Q #24) How to select value in a dropdown?**

The value in the dropdown can be selected using WebDriver’s Select class.

**Syntax:**

**selectByValue:**  
Select selectByValue = **new** Select(driver.findElement(By.id(“SelectID\_One”)));  
selectByValue.selectByValue(“greenvalue”);

**selectByVisibleText:**  
Select selectByVisibleText = **new** Select (driver.findElement(By.id(“SelectID\_Two”)));  
selectByVisibleText.selectByVisibleText(“Lime”);

**selectByIndex:**  
Select selectByIndex = **new** Select(driver.findElement(By.id(“SelectID\_Three”)));  
selectByIndex.selectByIndex(2);

**Q #25) What are the different types of navigation commands?**

Following are the [navigation commands](https://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):  
**navigate().back()** – The above command requires no parameters and takes back the user to the previous webpage in the web browser’s history.

**Sample code:**  
driver.navigate().back();

**navigate().forward()** – This command lets the user to navigate to the next web page with reference to the browser’s history.

**Sample code:**  
driver.navigate().forward();

**navigate().refresh()** – This command lets the user to refresh the current web page there by reloading all the web elements.

**Sample code:**  
driver.navigate().refresh();

**navigate().to()** – This command lets the user to launch a new web browser window and navigate to the specified URL.

**Sample code:**  
driver.navigate().to(“https://google.com”);

**Q #26) How to click on a hyper link using linkText?**

driver.findElement(By.linkText(“Google”)).click();

The command finds the element using link text and then click on that element and thus the user would be re-directed to the corresponding page.

The above-mentioned link can also be accessed by using the following command.

driver.findElement(By.partialLinkText(“Goo”)).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 with the specified substring and then clicks on it.

**Q #27)** **How to**[**handle frame in WebDriver**](https://www.softwaretestinghelp.com/selenium-tutorial-18/)**?**

An inline frame acronym as iframe is used to insert another document within the current HTML document or simply a web page into a web page by enabling nesting.

**Select iframe by id**  
driver.switchTo().frame(“ID of the frame“);

**Locating iframe using tagName**  
driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));

**Locating iframe using index**

**frame(index)**  
driver.switchTo().frame(0);

**frame(Name of Frame)**  
driver.switchTo().frame(“name of the frame”);

**frame(WebElement element)**  
**Select Parent Window**  
driver.switchTo().defaultContent();

**Q #28) When do we use findElement() and findElements()?**

**findElement():**findElement() is used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.

**Syntax:**

WebElement element = driver.findElements(By.xpath(“//div[@id=’example’]//ul//li”));  
**findElements():**findElements() is used to find all the elements in the current web page matching to the specified locator value. Take a note that all the matching elements would be fetched and stored in the list of WebElements.

**Syntax:**  
List <WebElement> elementList = driver.findElements(By.xpath(“//div[@id=’example’]//ul//li”));

**Q #29)** **How to find more than one web element in the list?**

At times, we may come across elements of the same type like multiple hyperlinks, images etc arranged in an ordered or unordered list. Thus, it makes absolute sense to deal with such elements by a single piece of code and this can be done using WebElement List.

**Sample Code**

|  |
| --- |
| // Storing the list  List <WebElement> elementList = driver.findElements(By.xpath("//div[@id='example']//ul//li"));  // Fetching the size of the list  int listSize = elementList.size();  for (int i=0; i<listSize; i++)  {  // Clicking on each service provider link  serviceProviderLinks.get(i).click();  // Navigating back to the previous page that stores link to service providers  driver.navigate().back();  } |

**Q #30) What is the difference between driver.close() and driver.quit command?**

**close()**: WebDriver’s close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver. The command neither requires any parameter nor does it return any value.

**quit()**: Unlike close() method, quit() method closes down all the windows that the program has opened. Same as close() method, the command neither requires any parameter nor does is return any value.

**Q #31) Can Selenium handle windows based pop up?**

Selenium is an automation testing tool which supports only web application testing. Therefore, windows pop up cannot be handled using Selenium.

**Q #32) How can we handle web-based pop-up?**

WebDriver offers the users a very efficient way to [handle these pop-ups using Alert interface](https://www.softwaretestinghelp.com/handle-alerts-popups-selenium-webdriver-selenium-tutorial-16/). There are the four methods that we would be using along with the Alert interface.

* void dismiss() – The dismiss() method clicks on the “Cancel” button as soon as the pop-up window appears.
* void accept() – The accept() method clicks on the “Ok” button as soon as the pop-up window appears.
* String getText() – The getText() method returns the text displayed on the alert box.
* void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

**Syntax:**  
// accepting javascript alert  
                Alert alert = driver.switchTo().alert();  
alert.accept();

**Q #33) How can we handle windows based pop up?**

Selenium is an automation testing tool which supports only web application testing, that means, it doesn’t support testing of windows based applications. However Selenium alone can’t help the situation but along with some third-party intervention, this problem can be overcome. There are several third-party tools available for handling window based pop-ups along with the selenium like AutoIT, Robot class etc.

**Q #34) How to assert the title of the web page?**

//verify the title of the web page  
assertTrue(“The title of the window is incorrect.”,driver.getTitle().equals(“Title of the page”));

**Q #35) How to mouse hover on a web element using WebDriver?**

WebDriver offers a wide range of interaction utilities that the user can exploit to automate mouse and keyboard events. Action Interface is one such utility which simulates the single user interactions.

Thus, In the following scenario, we have used Action Interface to mouse hover on a drop down which then opens a list of options.

**Sample Code:**

|  |
| --- |
| // Instantiating Action Interface  Actions actions=new Actions(driver);  // howering on the dropdown  actions.moveToElement(driver.findElement(By.id("id of the dropdown"))).perform();  // Clicking on one of the items in the list options  WebElement subLinkOption=driver.findElement(By.id("id of the sub link"));  subLinkOption.click(); |

**Q #36) How to retrieve CSS properties of an element?**

The values of the css properties can be retrieved using a get() method:

**Syntax:**  
driver.findElement(By.id(“id“)).getCssValue(“name of css attribute”);  
driver.findElement(By.id(“id“)).getCssValue(“font-size”);

**Q #37) How to capture screenshot in WebDriver?**

|  |
| --- |
| import org.junit.After;  import org.junit.Before;  import org.junit.Test;  import java.io.File;  import java.io.IOException;  import org.apache.commons.io.FileUtils;  import org.openqa.selenium.OutputType;  import org.openqa.selenium.TakesScreenshot;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.firefox.FirefoxDriver;    public class CaptureScreenshot {  WebDriver driver;  @Before  public void setUp() throws Exception {  driver = new FirefoxDriver();  driver.get("https://google.com");  }  @After  public void tearDown() throws Exception {  driver.quit();  }    @Test  public void test() throws IOException {  // Code to capture the screenshot  File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);  // Code to copy the screenshot in the desired location  FileUtils.copyFile(scrFile, new File("C:\\CaptureScreenshot\\google.jpg"))  }  } |

**Q #38) What is Junit?**

[Junit](https://www.softwaretestinghelp.com/selenium-junit-framework-selenium-tutorial-11/) is a unit testing framework introduced by Apache. Junit is based on Java.

**Q #39) What are Junit annotations?**

Following are the JUnit Annotations:

* **@Test:**Annotation lets the system know that the method annotated as @Test is a test method. There can be multiple test methods in a single test script.
* **@Before:**Method annotated as @Before lets the system know that this method shall be executed every time before each of the test methods.
* **@After:**Method annotated as @After lets the system know that this method shall be executed every time after each of the test method.
* **@BeforeClass:**Method annotated as @BeforeClass lets the system know that this method shall be executed once before any of the test methods.
* **@AfterClass:**Method annotated as @AfterClass lets the system know that this method shall be executed once after any of the test methods.
* **@Ignore:**Method annotated as @Ignore lets the system know that this method shall not be executed.

**Q #40)** **What is TestNG and how is it better than Junit?**

[TestNG](https://www.softwaretestinghelp.com/testng-framework-selenium-tutorial-12/) is an advanced framework designed in a way to leverage the benefits by both the developers and testers. With the commencement of the frameworks, JUnit gained enormous popularity across the Java applications, Java developers and Java testers with remarkably increasing the code quality. Despite being easy to use and straightforward, JUnit has its own limitations which give rise to the need of bringing TestNG into the picture. TestNG is an open source framework which is distributed under the Apache Software License and is readily available for download.

TestNG with WebDriver provides an efficient and effective test result format that can, in turn, be shared with the stakeholders to have a glimpse on the product’s/application’s health thereby eliminating the drawback of WebDriver’s incapability to generate test reports. TestNG has an inbuilt exception handling mechanism which lets the program to run without terminating unexpectedly.

There are various advantages that make TestNG superior to JUnit. Some of them are:

* Added advance and easy annotations
* Execution patterns can set
* Concurrent execution of test scripts
* Test case dependencies can be set

**Q #41)** **How to set test case priority in TestNG?**

**Setting Priority in TestNG**

**Code Snippet**

|  |
| --- |
| package TestNG;  import org.testng.annotations.\*;  public class SettingPriority {  @Test(priority=0)  public void method1() {  }  @Test(priority=1)  public void method2() {  }  @Test(priority=2)  public void method3() {  }  } |

**Test Execution Sequence:**

1. Method1
2. Method2
3. Method3

**Q #42) What is a framework?**

The framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing.

**Q #43)** **What are the advantages of the Automation framework?**

**The advantage of**[**Test Automation framework**](https://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/)

* Reusability of code
* Maximum coverage
* Recovery scenario
* Low-cost maintenance
* Minimal manual intervention
* Easy Reporting

**Q #44) 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.

**Q #45) How can I read test data from excels?**

Test data can efficiently be read from excel using JXL or POI API. [See detailed tutorial here](https://www.softwaretestinghelp.com/selenium-framework-design-selenium-tutorial-21/).

**Q #46) What is the difference between POI and jxl jar?**

| **#** | **JXL jar** | **POI jar** |
| --- | --- | --- |
| 1 | JXL supports “.xls” format i.e. binary based format. JXL doesn’t support Excel 2007 and “.xlsx” format i.e. XML based format | POI jar supports all of these formats |
| 2 | JXL API was last updated in the year 2009 | POI is regularly updated and released |
| 3 | The JXL documentation is not as comprehensive as that of POI | POI has a well prepared and highly comprehensive documentation |
| 4 | JXL API doesn’t support rich text formatting | POI API supports rich text formatting |
| 5 | JXL API is faster than POI API | POI API is slower than JXL API |

**Q #47)** **What is the difference between Selenium and QTP?**

| **Feature** | **Selenium** | **Quick Test Professional (QTP)** |
| --- | --- | --- |
| Browser Compatibility | Selenium supports almost all the popular browsers like Firefox, Chrome, Safari, Internet Explorer, Opera etc | QTP supports Internet Explorer, Firefox and Chrome. QTP only supports Windows Operating System |
| Distribution | Selenium is distributed as an open source tool and is freely available | QTP is distributed as a licensed tool and is commercialized |
| Application under Test | Selenium supports testing of only web based applications | QTP supports testing of both the web based application and windows based application |
| Object Repository | Object Repository needs to be created as a separate entity | QTP automatically creates and maintains Object Repository |
| Language Support | Selenium supports multiple programming languages like Java, C#, Ruby, Python, Perl etc | QTP supports only VB Script |
| Vendor Support | As Selenium is a free tool, user would not get the vendor’s support in troubleshooting issues | Users can easily get the vendor’s support in case of any issue |

**Q #48) Can WebDriver test Mobile applications?**

WebDriver cannot test Mobile applications. WebDriver is a web-based testing tool, therefore applications on the mobile browsers can be tested.

**Q #49) Can captcha be automated?**

No, captcha and barcode reader cannot be automated.

**Q #50) What is Object Repository? How can we create an Object Repository in Selenium?**

Object Repository is a term used to refer to the collection of web elements belonging to Application Under Test (AUT) along with their locator values. Thus, whenever the element is required within the script, the locator value can be populated from the Object Repository. Object Repository is used to store locators in a centralized location instead of hardcoding them within the scripts.

In Selenium, objects can be stored in an excel sheet which can be populated inside the script whenever required.

# 6.Intellipat:48Q

### ****1. What is automation testing?****

[Automation](https://intellipaat.com/blog/what-is-automation-testing/) testing is the process of automating the manual testing of the application under test (AUT) or the system under test (SUT). It includes the involvement of testing tools that help us create test scripts, which can be used repeatedly without any human intervention.

***Know more about the***[***Difference between Automation Testing and Manual Testing***](https://intellipaat.com/blog/automation-vs-manual-testing/)***from our blog.***

### ****2. Is Selenium 2.0 different from Selenium 3.0? If so, how?****

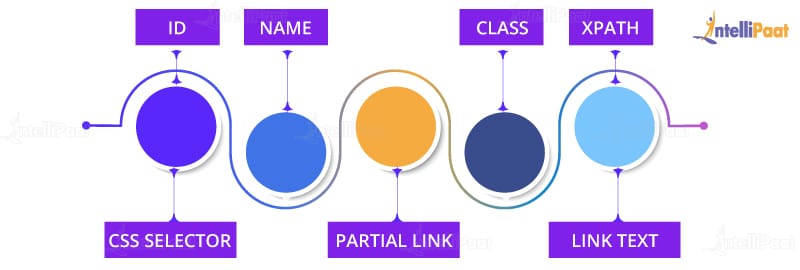
Selenium RC is called Selenium 2.0, and the integration of Selenium RC and WebDriver as a single tool has formed Selenium 3.0.

### ****3. Mention some of the popular tools used for automation testing.****

* Selenium
* Rational Robot
* HP Unified Functional Testing (UFT)
* IBM Rational Functional Tester

### ****4. What is a locator? How can you find elements in Selenium?****

Selenium uses locators to find and match the elements of a web page that it needs to interact with. There are different types of Selenium locators to identify various web elements on a web page:



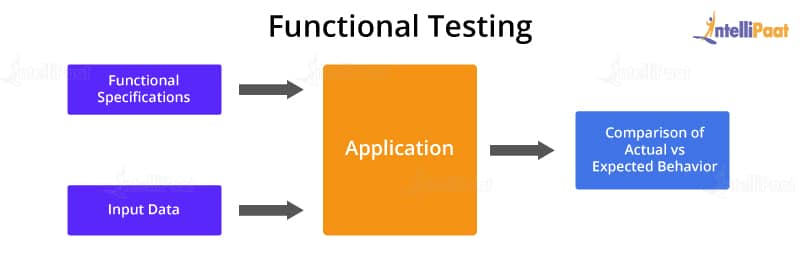
* ID
* Name
* Class
* Partial Link
* XPath
* CSS Selector
* Link Text

### ****5. What are the test types supported by Selenium?****

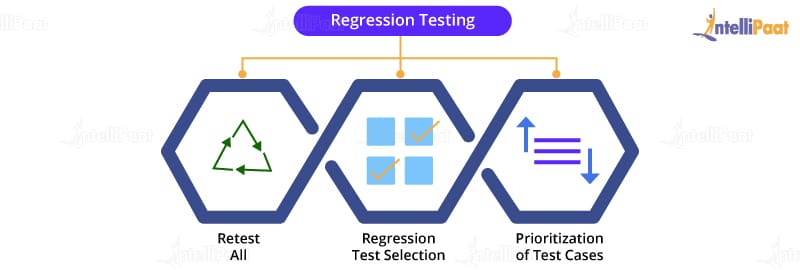
For testing web-based applications, Selenium can be used.

The test types supported by Selenium are:

* **Functional testing**: It verifies if each function of a software application performs in accordance with specific requirements. This testing primarily involves black-box testing, and it is not concerned about the source code of the application.



* **Regression testing**: It is nothing but a full or partial selection of the already executed test cases to be re-executed to ensure whether the existing functionalities work fine.



***Learn more about Selenium from this insightful blog on***[***How would you test Web Applications using Selenium?***](https://intellipaat.com/blog/how-to-test-web-application-using-selenium/)

### Get 100% Hike!

Master Most in Demand Skills Now !

Top of Form





Bottom of Form

### ****6. What is XPath?****

While DOM is the recognized standard way for navigating through an HTML element tree, **XPath** is the navigation tool 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 that is very much similar to HTML tags. Both being markup languages and falling under the same umbrella, XPath can be used to locate HTML elements.

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

### ****7. Explain the difference between single slash and double slash in XPath.****

* **Single slash (/)**: Single slash is used to create an XPath with an absolute path. In this case, the XPath would start selection from the document’s start node.
* **Double slash (//)**: Double slash is used to create an XPath with a relative path. In this case, the XPath would start selection from anywhere within the document.

### ****8. Why should you use Selenium for test automation?****

**Selenium should be used for test automation as it:**

* Is a free and open-source tool
* Has a large user base and community support
* Has cross-browser compatibility (Firefox, Chrome, Internet Explorer, Safari, etc.)
* Has great platform compatibility (Windows, Mac OS, Linux, etc.)
* Supports multiple programming languages (Java, C#, Ruby, Python, Perl, etc.)
* Has fresh and regular repository developments
* Supports distributed testing

[](https://intellipaat.com/test-architect-masters-program-training/)

### ****9. Does Selenium have any technical limitations? If so, what are those limitations?****

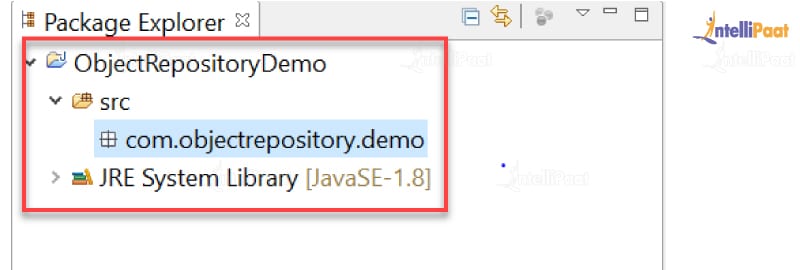
Yes, Selenium has a few limitations:

* Testing of only web applications is possible using Selenium.
* Testing of mobile applications or desktop applications is not possible.
* Captcha and barcode readers cannot be tested using Selenium.
* A third-party tool like [**TestNG**](https://intellipaat.com/blog/tutorial/selenium-tutorial/testng-in-selenium/) or JUnit should be used to generate reports.
* As Selenium is a free tool, there is no ready vendor support through which users can find various helping communities.
* Prior programming language knowledge is expected from users.

***Enroll today in our***[***Selenium Course***](https://intellipaat.com/selenium-training/)***to know more about Selenium!***

### ****10. What is an object repository?****

An object repository allows testers to accumulate web elements of the application under test (AUT), along with their locator values, in one or more centralized locations as restricted to hard-coding them within the test scripts.



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

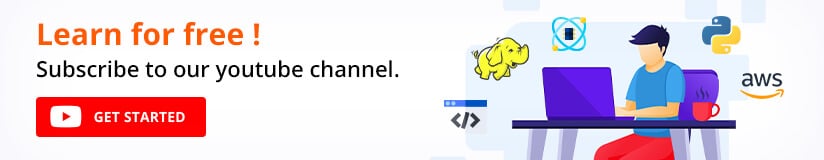
TypeKeys() will trigger JavaScript events, while type() won’t. TypesKeys collects different value attributes using JavaScript. Whereas, the type commands imitate an actual user typing.

### ****12. What are the advantages of Selenium?****

* Selenium is a purely open-source and portable automation testing tool.
* It supports different languages such as C#, PHP, Java, Perl, Python, JS, and Groovy.
* It also supports different OS, including Windows, Linux, UNIX, and Mac OS.
* It provides powerful methods such as Xpath, DOM, and CSS to locate elements.
* Since it is an open-source tool, developers can customize the code. Also, the developer community is supported by Google.

### Watch this video on Selenium automation testing tutorial:



[](https://www.youtube.com/user/intellipaaat?sub_confirmation=1)

### ****13. Define automation testing, and list down its advantages.****

Automation testing or test automation is the process of automating the testing process by using tools to write and execute test cases without any human intervention. It allows us to create scripts that can be executed frequently and generate detailed test reports of the application.

Its advantages are as given below:

* It helps in the performance and functional testing of an application.
* It makes the execution of repeated test cases easy.
* It supports the parallel execution of multiple test cases.
* It boosts the accuracy and efficiency of the application by cutting down the chances of human error.
* It easily performs testing on a large-scale test matrix.
* It saves time and money by reducing the burden of arbitrary tasks.

### ****14. What are the significant changes/upgrades made to various Selenium versions?****

Selenium’s first version included only three sets of tools: Selenium IDE, Selenium RC, and Selenium Grid. There was no WebDriver included in the first version. Later, Selenium WebDriver was introduced and hence included in Selenium V2. However, as WebDriver got included, the use of Selenium RC was discouraged with time and is not much in use ever since. Selenium 3 is in use. There are some newly added [**Selenium features**](https://intellipaat.com/blog/tutorial/selenium-tutorial/features-of-selenium/) such as IDE and WebDriver. Selenium 4 is the latest released version.

### ****15. How many types of WebDriver APIs are available in Selenium?****

The following is the list of WebDriver APIs:

* AndroidDriver
* ChromeDriver
* EventFiringWebDriver
* FirefoxDriver
* HTMLUnitDriver
* InternetExplorerDriver
* iPhoneDriver
* iPhoneSimulatorDriver
* RemoteWebDriver

**Career Transition**

[[](javascript:void(0);)](javascript:void(0);)

[[](javascript:void(0);)](javascript:void(0);)

[[](javascript:void(0);)](javascript:void(0);)

### ****16. What is an exception test in Selenium?****

An exception test is a test that looks forward to an exception to be thrown inside a test class. It anticipates the @Test annotation followed by the expected exception name. For example, **@Test(expectedException = NoSuchElementException.class)** is an exception test for missing elements in Selenium.

**Note**: Keep in mind the syntax, where the exception is suffixed with **.class**.

### ****17. What is POM (Page Object Model)? What are its advantages?****

Page Object Model is a design pattern used to create object repositories for the web UI elements. Every web page of an application has a corresponding page class that is responsible for locating the web elements and performing actions on them.

Its advantages are as follows:

* It provides support to separate operations and flows on the UI from verification, hence improving code readability.
* As the object repository is independent of test cases, multiple tests can use the same object repository.
* It increases the reusability of the code

That’s all for the basic Selenium Java interview questions. Let’s move on to the next section of intermediate Selenium WebDriver interview questions.

## ****Intermediate Selenium Interview Questions and Answers****

### ****18. What are the different types of annotations used in Selenium? Explain the JUnit annotation linked with Selenium.****

In Java, a special form of syntactic metadata can be added to Java source code, which is known as ‘annotations’. Variables, parameters, packages, methods, and classes are annotated. Some of the JUnit annotations are:

* Test
* Before
* After
* Ignore
* BeforeClass
* AfterClass
* RunWith

JUnit annotations linked with Selenium are:

JUnit AnnotationsJUnit Annotations

* @Test: The @Test annotation finds a method to be a test method. When used before a test method, it is mentioned as ‘@Test’; it informs the JUnit framework that the following method is a test method.
* @Before: The @Before annotation is used to find the method that is executed before executing the test method. This method can be used to set up the test environment.
* @After: The @After annotation is a method that is executed after executing the test method. This method can be used to do a teardown, i.e., it is a method used to delete all temporary data, set up default values, clean up the test environment, etc.
* @BeforeClass: The @BeforeClass method is used only once before the start of all tests. Basically, this is used to perform cumbersome activities, like connecting to a database.
* @AfterClass: The @AfterClass method is used only once after executing all tests. This is used to carry out clean-up activities, like disconnecting from a database.

***Download the***[***Selenium Cheat Sheet***](https://intellipaat.com/blog/tutorial/selenium-tutorial/selenium-cheat-sheet/)***and use it whenever required especially during your interviews.***

### ****19. Why do testers choose Selenium over QTP?****

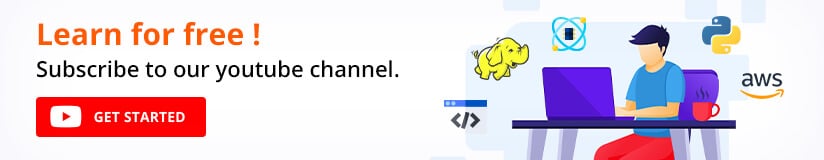
Selenium is more widely used than QTP since:

* Selenium is an open-source tool, whereas QTP is a profitable tool
* Selenium is used specifically for testing web-based applications, while QTP can be used for testing client–server applications too
* Selenium supports multiple browsers like Firefox, IE, Opera, Safari, etc. and has multiple operating systems compatibility too. Selenium-supported OS platforms are Windows, Mac, Linux, etc. On the other hand, QTP is limited to Internet Explorer on Windows
* Selenium supports multi-programming language compatibility. Languages supported by Selenium are Python, Ruby, Perl, etc. But, QTP supports only VBScript

***You can find more on Selenium by visiting our***[***Selenium Community***](https://intellipaat.com/community/selenium)***!***

### ****Watch this video on Selenium tutorial:****



[](https://www.youtube.com/user/intellipaaat?sub_confirmation=1)

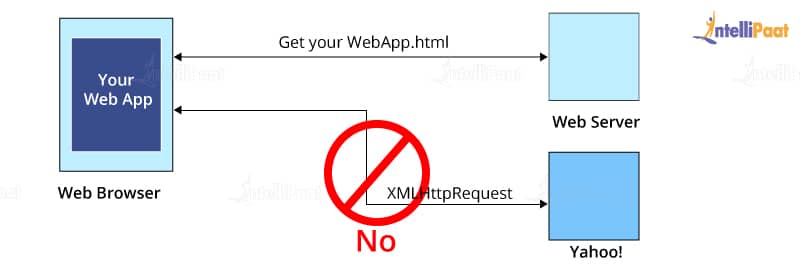
### ****20. What are the four elements that you have to pass in Selenium?****

Four parameters that need to be passed in Selenium are:

* Host
* Port number
* Browser
* URL

### ****21. What is Same-origin Policy? How can we avoid it?****

The ‘Same-origin Policy’ is introduced for security reasons.

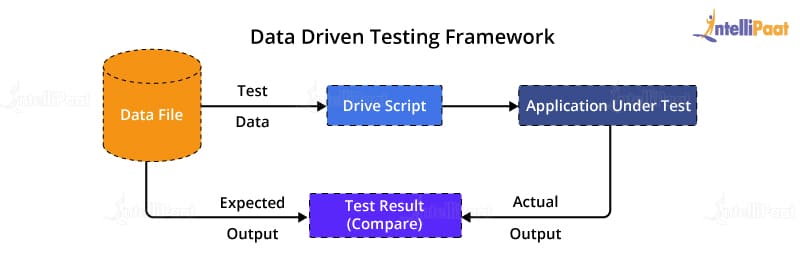


* It ensures that the content of our site will never be accessible by a script from another site.
* As per the policy, any code loaded within the browser can only operate within that website’s domain.

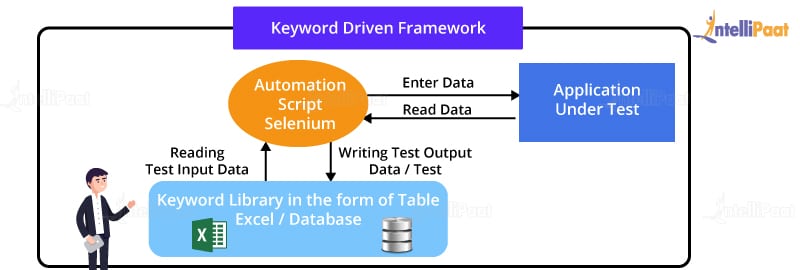
To avoid this same-origin policy, the proxy injection method is used. In the proxy injection mode, Selenium Server tricks the browser to be a real HTTP URL, i.e., it acts as a client-configured HTTP proxy, which sits between the browser and the application under test (AUT) and then masks the AUT under a fictional URL.

### ****22. What are data-driven framework and keyword-driven framework?****

A data-driven framework in Selenium is an approach of separating a ‘dataset’ from the actual ‘test case’ (code). This framework is completely dependent on the input test data. The test data is inserted from external sources, such as from an Excel file, a CSV file, or from any database. It also allows us to easily control how much data needs to be tested. We can easily increase the number of test parameters by adding more username and password fields to the Excel file (or other sources).



A keyword-driven framework is an extension to the data-driven testing framework in the sense that it not only isolates the test data from the scripts but also keeps the particular section of the code belonging to the test script in an external data file. These sets of code are known as keywords, and hence the framework is so named. Keywords are self-guiding and work based on what actions need to be performed on the application.



### ****23. How will you use Selenium to upload a file?****

If the file is on the same machine or in a mapped network drive, it is really straightforward: We have to just type the ‘path’ of the file in the FileUpload control.

Example:

driver = webdriver.Firefox()

element = driver.find\_element\_by\_id("fileUpload")

element.send\_keys("C:\myfile.txt")

***Learn more about Selenium from this informative***[***Selenium Tutorial***](https://intellipaat.com/blog/tutorial/selenium-tutorial/)***!***

### ****24. What is the difference between getwindowhandles() and getwindowhandle()?****

* **getwindowhandles():** It is used to get the address of all open browsers, and its return data type is Set<String>.
* **getwindowhandle()**: It is used to get the address of the current browser where the control is, and its return type is a string data type.

### ****25. What is Selenese, and what are the three types of Selenese?****

Selenese is a set of commands in Selenium used for running a test.

**Three types of Selenese are as follows:**

* **Actions**: They are used for performing interactions and operations with the target elements.
* **Accessors**: They are used for storing values in a variable.
* **Assertions**: They are used as a checkpoint.

### ****26. If you want to insert a breakpoint in Selenium IDE, how can you do that?****

To insert a breakpoint:

* First, select ‘Toggle Breakpoint’ by right-clicking on the command in [Selenium IDE](https://intellipaat.com/blog/tutorial/selenium-tutorial/selenium-ide/)
* Then, press ‘B’ on the keyboard and select the command
* The same step should be repeated for deselecting a breakpoint

### ****27. How do you launch the web browser using WebDriver?****

The following syntax can be used to launch the browser corresponding to the system’s operating system:

WebDriver driver = new FirefoxDriver();

Or

WebDriver driver = new InternetExplorerDriver();

Or

WebDriver driver = new ChromeDriver();

### ****28. List down some of the technical challenges with Selenium.****

* **Testing a Windows application**: Selenium is just a web-based driver. It does not support Windows-based apps and only supports web apps.
* **Testing mobile apps**: With the help of Selenium, we can test web apps on any OS and browser that run on desktops. But, we cannot test mobile apps with Selenium because it does not work with OS such as Android and iOS. However, there is an alternative for this, i.e., Appium. It is an open-source automation testing tool that uses the WebDriver protocol to drive native, hybrid, and iOS and Android, which is built specifically for testing mobile apps.
* **Limited reporting**: It is one of the key challenges. In Selenium, we cannot generate efficient and accurate reports. Accurate reports help developers fix all bugs and errors. We can create reports using [TestNG](https://intellipaat.com/blog/tutorial/selenium-tutorial/testng-in-selenium/) or ExtentReports.
* **Handling dynamic elements**: With the surge in the use of web apps, the management of dynamic elements should be as much efficient as possible. When a web page loads, the content present on the page changes depending on the user, location, and other factors. Most of today’s web apps are dynamic in nature for better user experience, e.g., e-commerce websites. In Selenium automation, the handling of dynamic web content is a major challenge. However, Selenium provides an explicit wait feature, where we can set a time interval for the automation testing process to hold the process for the new content to load. Also, another alternative is to utilize the implicit wait feature.
* **Handling page load**: Some of the web pages in a web app are user-specific. They load elements depending on the user. Also, some elements may be loaded depending on the user’s previous activities. During background processes, the Selenium script might not be able to identify a specific element. To overcome this, we can use explicit waits to provide sufficient time to load and discover the element.
* **Handling pop-up windows**: Whenever any simple, prompt, or confirmation alert pops up, it is difficult to automate it. Windows-based OS alerts are beyond Selenium’s capabilities as they are part of the OS instead of the browser. However, Selenium WebDriver can utilize multiple windows, and the web-based alerts can easily be handled with the help of the switchTo method. It manages the OS-based pop-ups while keeping the browser running in the background.

***Checkout our intuitive blog on the***[***Future Scope of Selenium***](https://intellipaat.com/blog/selenium-career-opportunities/)

### ****29. What is the difference between setSpeed() and sleep() methods?****

Both setSpeed() and Sleep() in Selenium are used to delay the speed of execution.

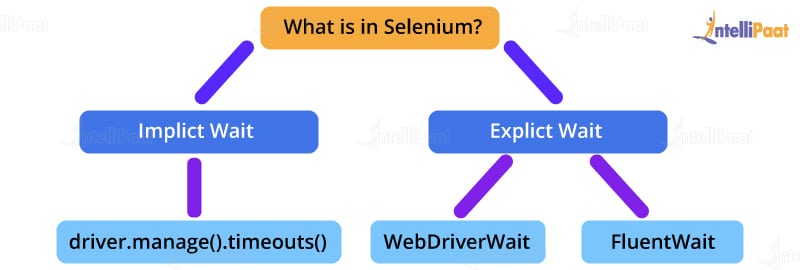
* **setSpeed**: Sets the execution speed with a delay of milliseconds, followed by the Selenium operation. By default, the delay is 0 milliseconds.
* **sleep**: Causes the suspension of execution of the current thread for a specified period.

That is all in the section of intermediate Selenium framework interview questions. Let’s move on to the next section of Selenium advanced interview questions.

## ****Advanced Selenium Interview Questions for Experienced Professional****

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

There are two types of waits available in WebDriver:



* **Implicit wait**: These waits are used to provide a default waiting time (say, 30 seconds) between the consecutive test steps across the entire test script. Hence, the subsequent test step would only be executed when the 30 seconds are over after executing the previous test step.
* **Explicit wait**: These waits are used to halt the execution until a particular condition is met or the maximum time has elapsed.

Explicit waits are instantiated for a particular instance only, whereas implicit waits are not.

***Want to learn about Selenium with Python! Check out our blog on***[***Selenium Python Tutorial for beginners***](https://intellipaat.com/blog/selenium-python-tutorial-for-beginners/)***.***

### ****31. How do you handle a frame in WebDriver?****

An **iframe** (an acronym for ‘inline frame’) is used to insert another document within the current HTML document.

**Selecting iframe by ID:**

driver.switchTo().frame(“ID of the frame“);

**Locating iframe using the tagName:**

driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));

**Locating iframe using index:**

* **frame(index)**

driver.switchTo().frame(0);

* **frame(“Name of the Frame”)**

driver.switchTo().frame(“name of the frame”);

* **frame(WebElement element)**  
  **Select Parent Window**

driver.switchTo().defaultContent();

***Go through the***[***Selenium Course in London***](https://intellipaat.com/selenium-training-london/)***to get a clear understanding of Selenium!***

### ****32. How do you set the test case priority in TestNG?****

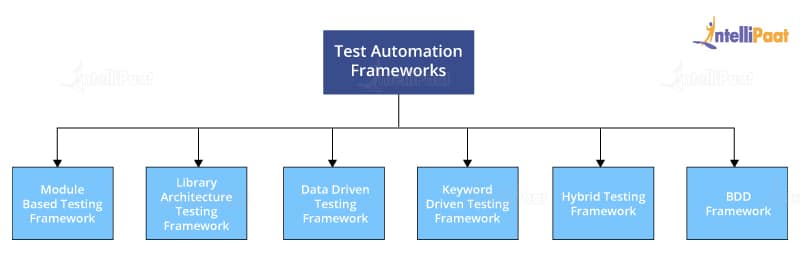
Setting Priority in TestNG:

**Test Execution Sequence:**

* Method1
* Method2
* Method3

### ****33. What are the different types of frameworks?****

Different types of frameworks are as follows:



* **Module-based testing framework**: This framework divides the entire application under test (AUT) into a number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts are taken together, it builds a larger test script representing more than one module.
* **Library architecture testing framework**: Instead of dividing AUT into test scripts, with this framework, we segregate the application into functions or rather common functions that can be used by the other parts of the application as well. Thus, we create a common library constituting common functions for AUT. Therefore, these libraries can be called from the test scripts whenever required.
* **Data-driven testing framework**: The data-driven testing framework helps us segregate the test script logic and the test data from each other. It lets us store the test data into an external database. The data is conventionally stored in ‘key–value’ pairs. Keys can be used to access and populate the data within the test scripts.
* **Keyword-driven testing framework**: The keyword-driven testing framework is an extension to the data-driven testing framework in the sense that it not only segregates the test data from the scripts but also keeps a certain set of codes belonging to the test script in an external data file.
* **Hybrid testing framework**: A hybrid testing framework is a combination of more than one of the above-mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.
* **Behavior-driven development framework**: The behavior-driven development framework allows the automation of functional validations in an easily readable and understandable format for Business Analysts, Developers, Testers, etc.

### ****34. Describe the difference between Selenium and QTP.****

|  |  |  |
| --- | --- | --- |
| **Feature** | **Selenium** | **Quick Test Professional (QTP)** |
| Browser compatibility | It supports almost all popular browsers: Firefox, Chrome, Safari, Internet Explorer, Opera, etc. | QTP supports Internet Explorer, Firefox, and Chrome. It only supports Windows operating system |
| Distribution | It is distributed as an open-source tool and is freely available | It is distributed as a licensed tool and is commercialized |
| Application under test (AUT) | It supports the testing of web-based applications only | It supports the testing of both web-based and Windows-based applications |
| Object repository | Object repository needs to be created as a separate entity in Selenium | QTP automatically creates and maintains the object repository |
| Language support | It supports multiple programming languages like Java, C#, Ruby, Python, Perl, etc. | It supports only VBScript |
| Vendor support | As Selenium is a free tool, users would not get the vendor’s support in troubleshooting issues | Users can easily get the vendor’s support if they face any issues |

***Learn more about Selenium in this***[***Selenium Training in New York***](https://intellipaat.com/selenium-training-new-york/)***to get ahead in your career!***

### ****35. In Selenium, what are breakpoints and start points?****

**Breakpoints**: Breakpoints are used to stall the execution of the test. The execution will stop whenever a breakpoint is implemented, and this will help us check whether the code is working properly or not.

**Start points**: Start points are the points from where the execution should begin. Start points can be used when we want to run the test script from the middle of the code or after a breakpoint.

### ****36. Mention the need for session handling while working with Selenium.****

While working with Selenium, we need session handling. This is because, during test execution, Selenium WebDriver has to interact with the browser all the time to execute the given commands. It is also possible that, before the current execution completes, someone else starts the execution of another script in the same machine and in the same type of browser. So, to avoid such a situation, we need session handling.

### ****37. Mention the types of listeners in TestNG.****

The types of listeners in TestNG are:

1. IAnnotationTransformer
2. IConfigurable
3. IConfigurationListener
4. IExecutionListener
5. IHookable
6. IInvokedMethodListener
7. IInvokedMethodListener2
8. IMethodInterceptor
9. IReporter
10. ISuiteListener
11. ITestListener

### ****38. How can we handle Windows-based popups with Selenium?****

Selenium solely supports web application testing. It does not support the testing of Windows-based applications or mobile applications. To handle Windows-based popups, third-party intervention is required. AutoIt and Robot Class are examples of third-party tools that we can use alongside Selenium to handle Windows-based popups.

Interested in learning Selenium? Check out our [Selenium Training in Sydney](https://intellipaat.com/selenium-training-sydney/)!

### ****39. For the database testing in Selenium WebDriver, what API is required?****

For the database testing in Selenium WebDriver, we need the JDBC (Java Database Connectivity) API. It allows us to execute SQL statements.

### ****40. How can you identify an object in Selenium?****

We can use isElementPresent (string locator) to find an object in Selenium. It takes a locator as the argument and, if found, returns a Boolean.

Become a master of Selenium by taking up this online [Selenium Course in Toronto](https://intellipaat.com/selenium-training-toronto/)!

### ****41. Mention the use of XPath in Selenium testing.****

XPath is used to define web elements on a web page. The major advantage of XPath is that it helps us in identifying the elements dynamically.

**There are two types of XPath:**

* Absolute XPath
* Relative XPath

### ****42. What is Selenium?****

Selenium is a popular open-source software that is used to automate web-based applications. It is a set of multiple software tools, and each tool has a different approach to automated testing.

**Selenium has four major components, namely:**

* Selenium Integrated Development Environment
* Selenium Remote Control
* Selenium WebDriver
* Selenium Grid

### ****43. Can WebDriver test mobile applications?****

No, WebDriver is a testing tool used for web-based applications. So, we cannot test mobile applications with Selenium WebDriver.

### ****44. Explain how Selenium Grid works.****

Selenium Grid creates a test suite that works by forwarding test cases to the hub, and from there, the test cases are redirected to Selenium WebDriver. WebDriver will then execute them in the browser. The test suite allows for running tests in parallel.

Hope you find our comprehensive blog on Selenium testing interview questions useful. Do let us know in the comments section if you could make use of these Selenium topics for the interview.

***Check out our blog to learn about***[***Types of Software Testing***](https://intellipaat.com/blog/types-of-software-testing/)***!***

### ****45. Explain the difference between driver.close() and driver.quit() commands in Selenium.****

The following is the major difference between both commands:

* **close()** command closes the currently active browser window, which is being used by the user or which is currently accessed by the web driver.
* **quit()** command closes all the windows opened by the program, unlike the driver.close () command.

Both the commands do not take any value and also do not accept any parameter.

### ****46. Explain the difference between findElement() and findElements() in Selenium.****

The difference between findElement() and findElements() is as follows:

* **findElement()**: It finds one particular element within the current page using the locating mechanism. It returns the first element located by the locator.
* **findElements()**: It finds all the elements within the current page with the help of the locating mechanism. Hence, it returns a list of matching web elements found by the locator.

***Get ready for a Manual Testing job by going through these Top***[***Manual Testing Interview Questions and Answers***](https://intellipaat.com/blog/interview-question/manual-testing-interview-questions/)***prepared by Industry Experts!***

### ****47. Mention the types of navigation commands.****

The following are the navigation commands provided by Selenium:

* **navigate().back()**: It takes the user back to the previous or the last-used web page, according to the history.
* **navigate().forward()**: It takes the user to the next web page, according to the browser history.
* **navigate().refresh()**: It allows the user to refresh the current web page by reloading all the web elements.
* **navigate().to()**: It takes the user to a new web page in a new window, depending on the URL specified.

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

We can detect whether the given links are broken or not by using the following process:

1. First, accumulate all the links present on a web page using the <a> anchor tag. For each <a> tag, use the attribute ‘href’ value to obtain the hyperlink
2. Send HTTP requests for each link and verify the HTTP response code
3. Based on the HTTP response code, determine if the link is valid or broken. Then, use the driver.get() method to navigate to a URL, which will respond with a status of 200 – OK (200 – OK indicates that the link is working). If we get any other status, then it indicates that the link is broken
4. Repeat the same process for all the links captured

# STM 134Q: <https://www.softwaretestingmaterial.com/selenium-interview-questions/>

### ****1. What is Automation Testing?****

Automation testing is the process of testing a software or application using an automation testing tool to find the defects. In this process, executing the test scripts and generating the results are performed automatically by automation tools. It is required when we have a huge amount of [regression test cases](https://www.softwaretestingmaterial.com/regression-testing/). Some most popular tools to do automation testing are HP QTP/UFT, [Selenium WebDriver](https://www.softwaretestingmaterial.com/install-selenium-webdriver/), etc.,

### ****2. What are the benefits of Automation Testing?****

This is one of the common interview questions in any Automation testing job.

1. Saves time and money. Automation testing is faster in execution.
2. Reusability of code. Create one time and execute multiple times with less or no maintenance.
3. Easy reporting. It generates automatic reports after test execution.
4. Easy for compatibility testing. It enables parallel execution in the combination of different OS and browser environments.
5. Low-cost maintenance. It is cheaper compared to manual testing in a long run.
6. Automated testing is more reliable.
7. Automated testing is more powerful and versatile. Automation tools allow us to integrate with [Cross Browser Testing](https://www.softwaretestingmaterial.com/run-selenium-tests-on-browserstack/) Tools, [Jenkins](https://www.softwaretestingmaterial.com/setup-integration-jenkins-ci-tools/), [Github](https://www.softwaretestingmaterial.com/selenium-continuous-integration/), etc.,
8. It is mostly used for regression testing. Supports execution of repeated test cases.
9. Minimal manual intervention. Test scripts can be run unattended.
10. Maximum coverage. It helps to increase the test coverage.

### ****3. What are the challenges and limitations of Selenium WebDriver?****

As we all know Selenium WebDriver is a tool that automates the browser to mimic real user actions on the web. Selenium is a free open source testing tool. Some of the challenges with Selenium WebDriver are as follows

1. We cannot test the windows application
2. We cannot test mobile apps
3. Limited reporting
4. Handling dynamic Elements
5. Handling page load
6. Handling pop up windows
7. Handling captcha

Read the detailed explanation on the [challenges and limitations of Selenium WebDriver](https://www.softwaretestingmaterial.com/challenges-and-limitations-of-selenium-webdriver/)

### ****4. What type of tests have you automated?****

Our main focus is to automate test cases to do [*Regression testing*](https://www.softwaretestingmaterial.com/regression-testing/), [*Smoke & Sanity testing*](https://www.softwaretestingmaterial.com/smoke-testing-vs-sanity-testing/). Sometimes based on the project and the test time estimation, we do focus on End to End testing.

### ****5. How many test cases you have automated per day?****

It is one of the Selenium Tricky Interview Questions.

Actually, it depends on Test case scenario complexity and length. I did automate 2-5 test scenarios per day when the complexity is limited. Sometimes just 1 or fewer test scenarios in a day when the complexity is high.

### ****6. What is a Framework*?*****

A framework defines a set of rules or best practices that we can follow in a systematic way to achieve the desired results. There are different types of automation frameworks and the most common ones are:

* [Data-Driven Testing Framework](https://www.softwaretestingmaterial.com/data-driven-framework-selenium-webdriver/)
* Keyword Driven Testing Framework
* Hybrid Testing Framework
* Behavioural Driven Framework

[Detailed Explanation: Types of Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/)

### ****7. What type of test cases to be automated?****

Types of Test Cases To Automate are

* Data-driven test cases
* Test cases with higher complexity
* Test case with many database updates
* The test execution rate is high
* Smoke/Critical tests
* Tests with several combinations
* Graph test cases
* Higher manual execution time

Read in detail explanation on [types of test cases to be automated](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/) here

### ****8. What type of test cases not to be automated?****

Types of Test Cases Not To Be Automated are

* Subjective Validation
* New Functionalities
* Strategic Development
* User Experience
* Complex Functionality
* Quality Control
* Low return on investment
* Installation and setup testing

Read in detail explanation on [types of test cases not to be automated](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/) here

### ****9. What are the advantages of the Test Automation Framework?****

1. Reusability of code.
2. Easy reporting.
3. Low-cost maintenance.
4. Maximum Coverage
5. Minimal manual intervention

### ****10. Have you created any Framework?****

**If you are a beginner:**You can say “No, I didn’t get a chance to create a framework from the scratch. I have used the framework which is already available. My contribution is mostly in creating test cases by using the existing framework.”

**If you are a beginner but have good knowledge of creating framework:** You can say “Yes, I have involved in developing framework along with other automation testers in my company.”

**If you are an experienced tester:**You can say “I have contributed to developing framework.” or You can say “Yes, I have created a framework from the scratch. There was no automation process in my previous company. I designed the framework from the scratch.”

### ****11. How would you explain the Selenium test automation framework in the interview?****

Here we have clearly explained each component of the Framework. Check this post to learn more about [how to explain the selenium test automation framework to the interviewer](https://www.softwaretestingmaterial.com/explain-test-automation-framework/).

### ****12. Why do you prefer Selenium Automation Tool?****

I prefer Selenium Automation Tool because some of the benefits of Selenium to do automation testing are

* **Free and open source –** It is a free open source tool. There is no need to allot budget for this tool
* **Help –** Have large user base and helping communities.
* [**Cross-browser compatibility**](https://www.softwaretestingmaterial.com/what-is-cross-browser-testing/)**–** It works on almost all popular browsers such as Chrome, Firefox, Internet Explorer, and Safari.
* **Cross Platform compatibility –** It works on platforms such as Windows, Linux, Mac.
* **Multiple programming languages –** It supports programming languages such as [Java](https://www.softwaretestingmaterial.com/java-tutorial/), Phyton, Perl, Php, C#, Ruby, etc.,
* **Parallel Execution –** Selenium Grid supports parallel execution of Selenium Scripts.
* **Continuous Integration –** We can achieve nightly execution using Jenkins.

### ****13. What is Selenium?****

Selenium is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers. It has gained a lot of popularity in terms of web-based automated testing and giving a great competition to the famous commercial tool HP QTP (Quick Test Professional) AKA HP UFT (Unified Functional Testing).

Selenium is a set of different software tools. Each tool has a different approach in supporting web based automation testing.

It has four components namely,

1. Selenium IDE (Selenium Integrated Development Environment)
2. Selenium RC (Selenium Remote Control)
3. Selenium WebDriver
4. Selenium Grid

### ****14. What is Selenium IDE?****

Selenium IDE (Integrated Development Environment) is a Firefox plugin. It is the simplest framework in the Selenium Suite. It allows us to record and playback the scripts. Even though we can create scripts using Selenium IDE, we need to use Selenium RC or Selenium WebDriver to write more advanced and robust test cases.

### ****15. What is Selenese?****

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

### ****16. Which is the only browser that supports Selenium IDE to be used?****

Firefox and Chrome. However, as Selenium IDE is community-powered, regular updates and compatibility with new browser versions cannot be ensured.

Back in 2017 when it no longer worked with Firefox’s latest version, users switched to [Katalon Recorder](https://www.softwaretestingmaterial.com/go/katalon-selenium-interview-questions/). It supports the same commands, extension scripts, data-driven testing, and advanced test reporting platform with TestOps.

***Check out***[***Best Chrome Extensions for Software Testers***](https://www.softwaretestingmaterial.com/chrome-extensions-for-software-testers/)

### ****17. What is Selenium RC?****

Selenium RC AKA Selenium Remote control / Selenium 1. Selenium Remote Control was the main Selenium project for a long time before the WebDriver merge brought up Selenium 2. Selenium 1 is still actively supported (in maintenance mode). It relies on JavaScript for automation. It supports Java, Javascript, Ruby, PHP, Python, Perl, and C#. It supports almost every browser out there.

### ****18. What is Selenium WebDriver?****

Selenium WebDriver AKA Selenium 2 is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it. Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.

Learn [Selenium WebDriver Architecture](https://www.softwaretestingmaterial.com/selenium-webdriver-architecture/)

### ****19. What is the difference between Selenium 3 and Selenium 4?****

We all know that Selenium 4 was released as a stable version on October 13, 2021. So here in this post, we have covered Selenium 4 Interview Questions & Answers. First Let’s see the difference between Selenium 3 and Selenium 4.

**Selenium 3 –** JSON wire protocol was used to communicate between the Selenium Webdriver APIs and the browser native APIs. All the requests and responses communicated across the protocol were encoded & decoded.

**Selenium 4 –** Follows the W3C standard protocol. Due to this request and the response communicated across the protocol doesn’t require the encoding and decoding API.

***Checkout***[***new features of Selenium 4***](https://www.softwaretestingmaterial.com/selenium-4/)

### ****20. What is Selenium Grid?****

Selenium Grid is a tool used together with Selenium RC to run tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems.

In simple words, it is used to distribute your test execution on multiple platforms and environments concurrently.

### ****21. When do you 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

### ****22. What are the advantages of Selenium Grid?****

It allows running test cases in parallel thereby saving test execution time.  
It allows multi-browser testing  
It allows us to execute test cases on multi-platform

### ****23. What is a hub in Selenium Grid?****

A hub is a server or a central point that controls the test executions on different machines.

### ****24. What is a node in Selenium Grid?****

Node is the machine which is attached to the hub. There can be multiple nodes in Selenium Grid.

### ****25. What are the types of WebDriver APIs available in Selenium?****

* Firefox Driver
* Gecko Driver
* InternetExplorer Driver
* Chrome Driver
* HTMLUnit Driver
* Opera Driver
* Safari Driver
* Android Driver
* iPhone Driver
* EventFiringWebDriver

### ****26. Which WebDriver implementation claims to be the fastest?****

The fastest implementation of WebDriver is the HTMLUnitDriver. It is because the HTMLUnitDriver does not execute tests in the browser. Starting a browser and running test cases took more time compared to running the scripts without a browser. HTMLUnitDriver took a simple HTTP request-response mechanism for test case execution.

Learn more on [How To Do Headless Browser Testing using Selenium WebDriver](https://www.softwaretestingmaterial.com/headless-browser-testing-using-selenium-webdriver/)

### ****27. What are the Programming Languages supported by Selenium WebDiver?****

* [Java](https://www.softwaretestingmaterial.com/java-tutorial/)
* C#
* Python
* Ruby
* Perl
* PHP

### ****28. Which language is not supported by selenium?****

Selenium supports all major programming languages such as Java, C#, Perl, Python, Ruby, PHP, Scala and Groovy. As of today, others are not compatible.

### ****29. What are the Operating Systems supported by Selenium WebDriver?****

* Windows
* Linux
* Mac OS X
* iOS
* Android

### ****30. What are the testing types that can be supported by selenium?****

Testing types that can be supported by Selenium are as follows:

* Functional Testing
* Regression Testing
* Retesting
* Acceptance Testing
* End-to-End Testing
* Smoke Testing
* Sanity Testing
* Responsive Testing
* Cross Browser Testing
* UI Testing
* Integration Testing

### ****31. How many parameters can selenium commands have at minimum?****

There are four parameters that you have to pass in Selenium are

* Host
* Port Number
* Browser
* URL

**Host:** It is the parameter which we use to bind Selenium to a specific IP. Usually, we run selenium tests on our local machine so the value will be ‘localhost’. You can sepcify IP address instead of localhost.

java -jar <selenium server standalone jar name> -host <Your IP Address>

**Port Number:** TCP/IP port which is used to connect selenium tests to the selenium grid hub. Default port hub is 4444.  
java -jar <selenium server standalone jar name> -role hub -port 4444

Make sure no other application in your system is using this port. You may face an exception like Exception in thread “main” java.net.BindException: Selenium is already running on port 4444. Or some other service is.

If this occurs you can either shutdown the other process that is using port 4444, or you can tell Selenium-Grid to use a different port for its hub. Use the -port option for changing the port used by the hub.

java -jar <selenium server standalone jar name> -role hub -port 4441

**Browser:** To pass the browser which has to execute our selenium scripts

**URL:** To pass the application URL

### ****32. What are the Open-source Frameworks supported by Selenium WebDriver?****

* JUnit
* TestNG

***Read:***[***TestNG Complete Tutorial***](https://www.softwaretestingmaterial.com/testng-tutorial/)

### ****33. What are the Locators available in Selenium?****

In Selenium WebDriver, there are 8 different types of locators:

1. ID – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-id-locator/)
2. ClassName – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-class-name-locator/)
3. Name – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-name-locator/)
4. TagName – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-tag-name-locator/)
5. LinkText – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-link-text-and-partial-link-text-locator/)
6. PartialLinkText – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-link-text-and-partial-link-text-locator/)
7. XPath – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-xpath-locator/)
8. CSS Selector – [Practical example](https://www.softwaretestingmaterial.com/css-selector-selenium-webdriver-tutorial/)

Click here to see the detailed post on [Locators](https://www.softwaretestingmaterial.com/locators-in-selenium/).

### ****34. What is an XPath?****

XPath is used to locate the elements. Using XPath, we could navigate through elements and attributes in an XML document to locate web elements such as textbox, button, checkbox, Image etc., in a web page.

[Learn How To Write Dynamic XPath](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/)

### ****35. When you use these locators ID, Name, XPath, Or CSS Selector?****

**ID** & **Name** locators will be used when there are unique identifiers & unique names available on the web page.  
**CSS Selector** can be used for performance and when ID & Name locators are not unique.  
**XPath** is used when there is no preferred locators.

### ****36. What is the difference between “/” and “//”****

**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.

### ****37. What is the difference between Absolute Path and Relative Path?****

Absolute XPath starts from the root node and ends with desired descendant element’s node. It starts with top HTML node and ends with input node. It starts with a single forward slash(/) as shown below.



|  |  |
| --- | --- |
| 1 | /html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input |

Relative XPath starts from any node in between the HTML page to the current element’s node(last node of the element). It starts with a double forward slash(//) as shown below.



|  |  |
| --- | --- |
| 1 | //input[@id='email'] |

### ****38. What should you do when even XPath functions can’t identify the web element?****

In the early stages of software developement, developers change identifiers and elements quite often. During the execution, the web elements may change dynamically and we cannot identify the web elements. To overcome this we use XPath axes along with XPath functions.

### ****39. What are XPath Axes?****

XPath axes are used to search for the multiple nodes in the XML document from the context (current) node.

XPath axes are used to find dynamic elements that would otherwise be impossible using standard locators.

### ****40. What is a Context Node?****

The context node is the node the XPath processor is currently looking at.

### ****41. What is the difference between Assert and Verify in Selenium?****

**Assert:** In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.

**Verify:** In simple words, there won’t be any halt in the test execution even though the verify condition is true or false.

Read this detailed post on [Assert vs Verify with practical example](https://www.softwaretestingmaterial.com/difference-between-assert-and-verify/) here detailed post check the below link.

### ****42. What are Soft Assert and Hard Assert in Selenium?****

Soft Assert: Soft Assert collects errors during *@Test* Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

Hard Assert: Hard Assert throws an AssertException immediately when an assert statement fails and test suite continues with next *@Test*

***Detailed Post:***[***Soft Assert***](https://www.softwaretestingmaterial.com/soft-assert/)

### ****43. What is the difference between setSpeed () and sleep () methods?****

Both sleep() and setSpeed() are used to delay the execution speed.

**setSpeed():** It set up speed that will apply a delay time before every Selenium operation.

***Example:*** setSpeed(“5000”) – It waits for 5 seconds

**sleep():** It set up wait only for once when called in our Selenium script.

***Example:*** sleep(5000) – It waits for 5 seconds

**Note:**setSpeed method is applicable to Selenium IDE and Selenium RC. We cannot use setSpeed in Selenium WebDriver.

### ****44. What are the verification points available in Selenium?****

In Selenium IDE, we use Selenese Verify and Assert Commands as Verification points  
In Selenium WebDriver, there is no built-in features for verification points. It totally depends on our coding style. some of the Verification points are

* To check for page title
* To check for certain text
* To check for certain element (text box, button, drop down, etc.)

### ****45. How to launch a browser using Selenium WebDriver?****

WebDriver is an Interface. We create an Object of a required driver class such as FirefoxDriver, ChromeDriver, InternetExplorerDriver etc.,

To launch Firefox Driver:WebDriver driver = new FirefoxDriver();

Note: If you use geckodriver with Selenium, you must upgrade to Selenium 3.3. Here we have to set the property as follows



|  |  |
| --- | --- |
| 1 | System.setProperty("webdriver.gecko.driver", "D:\\Selenium Environment\\Drivers\\geckodriver.exe"); |

To launch Chrome Driver:WebDriver driver = new ChromeDriver();

To launch Internet Explorer Driver:WebDriver driver = new InternetExplorerDriver();

To launch Safari Driver:  
WebDriver driver = new SafariDriver();

### ****46. Is the FirefoxDriver a Class or an Interface?****

*FirefoxDriver* is a Java class, and it implements the *WebDriver* interface.

### ****47. What is the super interface of WebDriver?****

SearchContext acts as the super interface of Web Driver.

### ****48. Explain the line of code****Webdriver driver = new FirefoxDriver(); ****?****



|  |  |
| --- | --- |
| 1 | Webdriver driver = new FirefoxDriver(); |

‘WebDriver‘ is an interface and we are creating an object of type WebDriver instantiating an object of FirefoxDriver class.

Read more on why [WebDriver driver = new FirefoxDriver();](https://www.softwaretestingmaterial.com/webdriver-driver-new-firefoxdriver/)

### ****49. We do create a reference variable ‘driver’ of type WebDriver as shown below. What is the purpose of doing this way?****



|  |  |
| --- | --- |
| 1  2  3  4  5 | WebDriver driver = new FirefoxDriver();    instead of creating    FirefoxDriver driver = new FirefoxDriver(); |

If we create a reference variable driver of type WebDriver then we could use the same driver variable to work with any browser of our choice such as IEDriver, SafariDriver etc.,

### ****50. What is WebElement selenium?****

WebElement in Selenium represents an HTML element. It basically represents a DOM element in a HTML document.

### ****51. What are the different exceptions you have faced in Selenium WebDriver?****

Some of the exceptions I have faced in my current project are

1. ElementNotVisibleException
2. StaleElementReferenceException

**Element Not visible Exception:**

This exception will be thrown when you are trying to locate a particular element on webpage that is not currently visible eventhough it is present in the DOM. Also sometimes, if you are trying to locate an element with the xpath which associates with two or more element.

**Stale Element Reference Exception:**

A [stale element reference exception](https://www.softwaretestingmaterial.com/stale-element-reference-exception-selenium-webdriver/) is thrown in one of two cases, the first being more common than the second.

The two reasons for Stale element reference are

1. The element has been deleted entirely.
2. The element is no longer attached to the DOM.

We face this stale element reference exception when the element we are interacting is destroyed and then recreated again. When this happens the reference of the element in the DOM becomes stale. Hence we are not able to get the reference to the element.

Some other exceptions we usually face are as follows:

* WebDriverException
* IllegalStateException
* TimeoutException
* NoAlertPresentException
* NoSuchWindowException
* NoSuchElementException

### ****52. How to handle STALEELEMENTREFERENCEEXCEPTION?****

Before looking how to handle Stale Element Reference Exception through Page Object Model. Let’s see what is Stale Element Reference Exception first.

Stale means old, decayed, no longer fresh. Stale Element means an old element or no longer available element. Assume there is an element that is found on a web page referenced as a WebElement in WebDriver. If the DOM changes then the WebElement goes stale. If we try to interact with an element which is staled then the StaleElementReferenceException is thrown.

Here we have given solutions to [handle StaleElementReferenceException](https://www.softwaretestingmaterial.com/stale-element-reference-exception-selenium-webdriver/) in detail.

### ****53. What are the types of waits available in Selenium WebDriver?****

In Selenium we could see three types of waits such as Implicit Waits, Explicit Waits and Fluent Waits.

* Implicit Waits – [Click to view detailed post](https://www.softwaretestingmaterial.com/implicit-waits-selenium-webdriver/)
* Explicit Waits – [Click to view detailed post](https://www.softwaretestingmaterial.com/webdriverwait-selenium-webdriver/)
* Fluent Waits – [Click to view detailed post](https://www.softwaretestingmaterial.com/selenium-fluentwait/)

### ****54. What is Implicit Wait In Selenium WebDriver?****

Implicit waits tell to the WebDriver to wait for a certain amount of time before it throws an exception. Once we set the time, WebDriver will wait for the element based on the time we set before it throws an exception. The default setting is 0 (zero). We need to set some wait time to make WebDriver to wait for the required time.

[Practical example](https://www.softwaretestingmaterial.com/implicit-waits-selenium-webdriver/)

### ****55. What is WebDriver Wait In Selenium WebDriver?****

WebDriverWaitis applied on a certain element with defined expected condition and time. This wait is only applied to the specified element. This wait can also throw an exception when an element is not found.

[Practical example](https://www.softwaretestingmaterial.com/webdriverwait-selenium-webdriver/)

### ****56. What is Fluent Wait In Selenium WebDriver?****

FluentWait can define the maximum amount of time to wait for a specific condition and frequency with which to check the condition before throwing an “ElementNotVisibleException” exception.

[Practical example](https://www.softwaretestingmaterial.com/selenium-fluentwait/)

### ****57. What happen if you mix both implicit wait and explicit wait in a Selenium Script?****

As per the official Selenium documentation, it is suggested not to mix both Implicit waits and Explicit Waits. Mixing both of them can cause unpredictable wait times.

Implicit wait is defined only once in the code. It will remain same throughout the driver object instance.

Explicit wait is defined whenever it is necessary in the code. This wait will call at the time of execution. It is a conditional wait.

Explicit wait will overwrite the implicit wait where ever explicit wait is applied. So, Explicit Wait gets first preference then Implicit Wait.

### ****58. What happen if you mix both Thread.Sleep and WebDriver Waits in a Selenium Script?****

Thread.sleep() method can be used to pause the execution for specified time in milliseconds

If we use WebDriver waits along with Thread.sleep() method then webdriver will hold the execution for specified time and then will follow other wait. Test execution time will become more, if we mix both waits.

### ****59. How to Login into any site if it is showing an Authentication Pop-Up for Username and Password?****

To do this we pass username and password with the URL



|  |  |
| --- | --- |
| 1  2 | http://username:password@url  e.g. http://myUserName:myPassword@softwaretestingmaterial.com |

### ****60. How to input text in the text box using Selenium WebDriver?****

By using sendKeys() method



|  |  |
| --- | --- |
| 1  2  3 | WebDriver driver = new FirefoxDriver();  driver.get("https://www.gmail.com");  driver.findElement(By.xpath("xpath")).sendKeys("Software Testing Material Website"); |

### ****61. How to input text in the text box without calling the sendKeys()?****



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // To initialize js object  JavascriptExecutor JS = (JavascriptExecutor)webdriver;  // To enter username  JS.executeScript("document.getElementById('User').value='SoftwareTestingMaterial.com'");  // To enter password  JS.executeScript("document.getElementById('Pass').value='tester'"); |

[***Learn JavaScriptExecutor in Selenium with Examples***](https://www.softwaretestingmaterial.com/javascriptexecutor-selenium-webdriver/)

### ****62. How to clear the text in the text box using Selenium WebDriver?****

By using clear() method



|  |  |
| --- | --- |
| 1  2  3  4 | WebDriver driver = new FirefoxDriver();  driver.get("https://www.gmail.com");  driver.findElement(By.xpath("xpath\_of\_element1")).sendKeys("Software Testing Material Website");  driver.findElement(By.xpath("xpath\_of\_element1")).clear(); |

### ****63. How to get a text of a web element?****

By using getText() method



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | package softwareTestingMaterial;  import org.openqa.selenium.By;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.chrome.ChromeDriver;  import org.testng.annotations.Test;  public class TestTestTest {    @Test  public void testmethod(){  System.setProperty("webdriver.chrome.driver", "D:\\Selenium Environment\\Drivers\\chromedriver.exe");     WebDriver driver = new ChromeDriver();     driver.get("https://www.google.com");     String availableText = driver.findElement(By.xpath("//\*[@id='gbw']/div/div/div[1]/div[1]/a")).getText();     System.out.println("Text Available is :"+availableText);  }  } |

### ****64. How to get an attribute value using Selenium WebDriver?****

By using getAttribute(value);

It returns the value of the attribute passed as a parameter.

HTML:



|  |  |
| --- | --- |
| 1 | <input name="nameSelenium" value="valueSelenium">SoftwareTestingMaterial</input> |

Selenium Code:



|  |  |
| --- | --- |
| 1  2  3 | String attributeValue = driver.findElement(By.name("nameSelenium")).getAttribute("value");  System.out.println("Available attribute value is :"+attributeValue);  Output: valueSelenium |

### ****65. How to click on a hyperlink using Selenium WebDriver?****

We use click() method in Selenium to click on the hyperlink



|  |  |
| --- | --- |
| 1 | driver.findElement(By.linkText(“Software Testing Material Website”)).click(); |

### ****66. How to submit a form using Selenium WebDriver?****

We use “submit” method on element to submit a form



|  |  |
| --- | --- |
| 1 | driver.findElement(By.id("form\_1")).submit(); |

Alternatively, you can use click method on the element which does form submission

### ****67. How to press ENTER key on text box In Selenium WebDriver?****

To press ENTER key using Selenium WebDriver, We need to use Selenium Enum Keys with its constant ENTER.



|  |  |
| --- | --- |
| 1 | driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER); |

### ****68. How to pause a test execution for 5 seconds at a specific point?****

By using **java.lang.Thread.sleep**(long milliseconds) method we could pause the execution for a specific time. To pause 5 seconds, we need to pass parameter as 5000 (5 seconds)



|  |  |
| --- | --- |
| 1 | Thread.sleep(5000) |

### ****69. Is Selenium Server needed to run Selenium WebDriver Scripts?****

When we are distributing our Selenium WebDriver scripts to execute using Selenium Grid, we need to use Selenium Server.

### ****70. What happens if I run this command.****driver.get(“www.softwaretestingmaterial.com”) ;

If the URL doesn’t contain http or https prefix then an exception is thrown. So, we need to pass HTTP protocol within driver.get() method.



|  |  |
| --- | --- |
| 1 | driver.get("https://www.softwaretestingmaterial.com"); |

### ****71. What is the alternative to****driver.get()****method to open an URL using Selenium WebDriver?****

Alternative method to driver.get(“url”) method is driver.navigate.to(“url”)

### ****72. What is the difference between****driver.get() ****and**** driver.navigate.to(“url”)****?****

driver.get(): To open an URL and it will wait till the whole page gets loaded  
driver.navigate.to(): To navigate to an URL and It will not wait till the whole page gets loaded

### ****73. Can I navigate back and forth in a browser in Selenium WebDriver?****

We use Navigate interface to do navigate back and forth in a browser. It has methods to move back, forward as well as to refresh a page.

**driver.navigate().forward();** – to navigate to the next web page with reference to the browser’s history  
**driver.navigate().back();** – takes back to the previous webpage with reference to the browser’s history  
**driver.navigate().refresh();**– to refresh the current web page thereby reloading all the web elements  
**driver.navigate().to(“url”);**– to launch a new web browser window and navigate to the specified URL

### ****74. What are the different types of navigation commands?****

Refer above question (Can I navigate back and forth in a browser)

### ****75. How to fetch the current page URL in Selenium?****

To fetch the current page URL, we use getCurrentURL()



|  |  |
| --- | --- |
| 1 | driver.getCurrentUrl(); |

### ****76. How can we maximize browser window in Selenium?****

To maximize browser window in selenium we use maximize() method. This method maximizes the current window if it is not already maximized



|  |  |
| --- | --- |
| 1 | driver.manage().window().maximize(); |

### ****77. How to delete cookies in Selenium?****

To delete cookies we use deleteAllCookies() method



|  |  |
| --- | --- |
| 1 | driver.manage().deleteAllCookies(); |

### ****78. What are the ways to refresh a browser using Selenium WebDriver?****

There are multiple ways to refresh a page in selenium

* Using *driver.navigate().refresh()* command as mentioned in the question 45
* Using driver.get(“URL”) on the current URL or using *driver.getCurrentUrl()*
* Using driver.navigate().to(“URL”) on the current URL or *driver.navigate().to(driver.getCurrentUrl());*
* Using *sendKeys(Keys.F5)* on any textbox on the webpage

### ****79. What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in Selenium WebDriver?****

driver.getWindowHandle() – It returns a handle of the current page (a unique identifier)  
driver.getWindowHandles() – It returns a set of handles of the all the pages available.

### ****80. What is the difference between driver.close() and driver.quit() methods?****

Purpose of these two methods (driver.close and driver.quit) is almost same. Both allow us to close a browser but still, there is a difference.

driver.close(): To close current WebDriver instance  
driver.quit(): To close all the opened WebDriver instances

### ****81. What is the difference between****driver.findElement() and driver.findElements() commands?

The difference between driver.findElement() and driver.findElements() commands is-

* findElement() returns a single WebElement (found first) based on the locator passed as parameter. Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.
* Syntax of findElement()-  
  WebElement textbox = driver.findElement(By.id(“textBoxLocator”));  
  Syntax of findElements()-  
  List <WebElement> elements = element.findElements(By.id(“value”));
* Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of 0 elements.

### ****82. What Is The Difference Between MaxSessions Vs. MaxInstances Properties in Selenium Grid?****

**MaxInstances** is the no. of browser instances of the same version of the browser that can run on the remote machine.

Let’s see an example below:



|  |  |
| --- | --- |
| 1  2 | -browser browserName=InternetExplorer,version=6,maxInstances=2,platform=WINDOWS  -browser browserName=firefox,version=11,maxInstances=2,platform=WINDOWS |

As per the above example, it will allow us to run 4 instances of both IE and Firefox at the same time (in parallel) in a remote machine.

**MaxSession** says how many browsers, independent of the type & version, can run in parallel on the remote machine.

It supersedes the “MaxInstances” setting.

If maxSession=1 then no more than a single browser would run. If maxSession=2 then any of the below combinations can run at a time irrespective of what MaxInstances we have defined.  
2 Internet Explorer  
2 Firefox  
1 Internet Explorer + 1 Firefox

### ****83. How to find whether an element is displayed on the web page?****

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. isDisplayed()



|  |  |
| --- | --- |
| 1 | boolean elePresent = driver.findElement(By.xpath("xpath")).isDisplayed(); |

1. isSelected()



|  |  |
| --- | --- |
| 1 | boolean eleSelected= driver.findElement(By.xpath("xpath")).isSelected(); |

1. isEnabled()



|  |  |
| --- | --- |
| 1 | boolean eleEnabled= driver.findElement(By.xpath("xpath")).isEnabled(); |

### ****84. How to select a value in a dropdown?****

By using Select class



|  |  |
| --- | --- |
| 1  2  3  4  5 | WebElement mySelectElement = driver.findElement(By.name("dropdown"));  Select dropdown = new Select(mySelectElement);  dropdown.selectByVisibleText(Text);  dropdown.selectByIndex(Index);  dropdown.selectByValue(Value); |

[Practical Example:](https://www.softwaretestingmaterial.com/handle-drop-down-and-multi-select-list-using-selenium/)

### ****85. How to capture Screenshot in Selenium WebDriver?****

Test cases may fail while executing the test scripts. While we are executing the test cases manually we just take a screenshot and place in a result repository. The same can be done by using Selenium WebDriver.

Some of the scenarios we may need to capture a screenshot using Selenium WebDriver are

i. Application issues  
ii. Assertion Failure  
iii. Difficulty to find Webelements on the web page  
iv. Timeout to find Webelements on the web page

Selenium provides an interface called TakesScreenshot which has a method getScreenShotAs which can be used to take a screenshot of the application under test.

In Selenium 3, we may face few issues while capturing Screenshots. To overcome we use aShot utility. Click on below links to see posts related to the normal way of capturing a screenshot and capturing a screenshot using aShot utility.

[Capture screenshot using Selenium WebDriver](https://www.softwaretestingmaterial.com/capture-screenshot-using-selenium-webdriver/)

[Full Page Screenshot using aShot utility](https://www.softwaretestingmaterial.com/how-to-capture-full-page-screenshot-using-selenium-webdriver/)

[Failed Test Cases Screenshot](https://www.softwaretestingmaterial.com/capture-screenshot-of-failed-test-cases-using-selenium-webdriver-2/)

### ****86 How to mouse hover on a web element using WebDriver?****

By using Actions class



|  |  |
| --- | --- |
| 1  2  3  4  5 | WebElement ele = driver.findElement(By.xpath("xpath"));  //Create object 'action' of an Actions class  Actions action = new Actions(driver);  //Mouseover on an element  action.moveToElement(ele).perform(); |

[Practical Example.](https://www.softwaretestingmaterial.com/mouse-hover-actions-using-selenium/)

### ****87. How can we handle Web-based Pop-ups or Alerts in Selenium?****

To handle Web-based alerts or popups, we need to do switch to the alert window and call Selenium WebDriver Alert API methods.

dismiss(): To click on Cancel button.  
accept(): To Click on OK button.  
getText(): To get the text which is present on the Alert.  
sendKeys(): To enter the text into the alert box.

[Practical Example.](https://www.softwaretestingmaterial.com/javascript-alerts-popups-selenium/)

### ****88. How can we handle windows based pop up?****

Selenium doesn’t support windows based applications. It is an automation testing tool which supports only web application testing. We could handle windows based popups in Selenium using some third party tools such as AutoIT, Robot class etc.

### ****89. How to handle hidden elements in Selenium WebDriver?****

We can handle hidden elements by using javaScript executor



|  |  |
| --- | --- |
| 1 | (JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();"); |

### ****90. How can you find Broken Links in a page using Selenium WebDriver?****

[Practical Example.](https://www.softwaretestingmaterial.com/broken-links-using-selenium/)

### ****91. How to find more than one web element in the list?****



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // To store the list  List <WebElement> eleList = driver.findElements(By.xpath("xpath"));  // To fetch the size of the list  int listSize = eleList.size();  //for loop  for (int i=0; i<listSize; i++)  {  // Clicking on each link  links.get(i).click();  // Navigating back to the previous page that stores the links  driver.navigate().back();  } |

### ****92. How to read a JavaScript variable in Selenium WebDriver?****

By using JavascriptExecutor



|  |  |
| --- | --- |
| 1  2  3  4  5 | // To initialize the JS object.  JavascriptExecutor JS = (JavascriptExecutor) webdriver;  // To get the site title.  String title = (String)JS.executeScript("return document.title");  System.out.println("Title of the webpage : " + title); |

### ****93. What is JavaScriptExecutor and in which cases JavaScriptExecutor will help in Selenium automation?****

In general, we click on an element using click() method in Selenium.

For example:



|  |  |
| --- | --- |
| 1 | driver.findElement(By.id("Id Value")).click(); |

Sometimes web controls don’t react well against selenium commands and we may face issues with the above statement (click()). To overcome such kind of situation, we use JavaScriptExecutor interface.

It provides a mechanism to execute Javascript through Selenium driver. It provides “executescript” & “executeAsyncScript” methods, to run JavaScript in the context of the currently selected frame or window.

There is no need to write a separate script to execute JavaScript within the browser using Selenium WebDriver script. Just we use predefined interface named ‘Java Script Executor’. We need to import the JavascriptExecutor package in the script.

Package:



|  |  |
| --- | --- |
| 1 | import org.openqa.selenium.JavascriptExecutor; |

Syntax:



|  |  |
| --- | --- |
| 1  2 | JavascriptExecutor js = (JavascriptExecutor) driver;  js.executeScript(Script,Arguments); |

**Script** – The JavaScript to execute***Arguments*** – The arguments to the script(Optional). May be empty.***Returns*** – One of Boolean, Long, String, List, WebElement, or null.

Let’s see some scenarios we could handle using this Interface:

1. To type Text in Selenium WebDriver without using sendKeys() method  
2. To click a Button in Selenium WebDriver using JavaScript  
3. To handle Checkbox  
4. To generate Alert Pop window in selenium  
5. To refresh browser window using Javascript  
6. To get innertext of the entire webpage in Selenium  
7. To get the Title of our webpage  
8. To get the domain  
9. To get the URL of a webpage  
10. To perform Scroll on an application using  Selenium  
11. To click on a SubMenu which is only visible on mouse hover on Menu  
12. To navigate to different page using Javascript

### ****94. How do you read test data from excels?****

Test data can efficiently be read from excel using JXL or POI API. POI API has many advantages than JXL.  
Click here to see a practical example of using [Apache POI](https://www.softwaretestingmaterial.com/handling-excel-files-using-apache-poi/).

### ****95. Is it possible to automate the captcha using Selenium?****

No, It’s not possible to automate captcha and bar code reader.

### ****96. Can You Use Selenium For Rest API Testing Or Web Services Testing?****

Simple answer for this is Selenium is not a tool for API Testing. It automates web browsers. Rest API & Web Services contains no UI. So we cannot automate using Selenium.

**Don’t miss:** [API Testing Interview Questions](https://www.softwaretestingmaterial.com/api-testing-interview-questions/)

### ****97. How to handle Ajax calls in Selenium WebDriver?****

Handling AJAX calls is one of the common issues when using Selenium WebDriver. We wouldn’t know when the AJAX call would get completed and the page has been updated. In this post, we see how to handle AJAX calls using Selenium.

AJAX stands for Asynchronous JavaScript and XML. AJAX allows the web page to retrieve small amounts of data from the server without reloading the entire page. AJAX sends HTTP requests from the client to server and then process the server’s response without reloading the entire page. To handle AJAX controls, wait commands may not work. It’s just because the actual page is not going to refresh.

When you click on a submit button, the required information may appear on the web page without refreshing the browser. Sometimes it may load in a second and sometimes it may take longer. We have no control over loading time. The best approach to handle this kind of situations in selenium is to use dynamic waits (i.e. WebDriverWait in combination with ExpectedCondition)

Some of the methods which are available are as follows:

1. titleIs() – The expected condition waits for a page with a specific title.



|  |  |
| --- | --- |
| 1 | wait.until(ExpectedConditions.titleIs(“Deal of the Day”)); |

2. elementToBeClickable() – The expected condition waits for an element to be clickable i.e. it should be present/displayed/visible on the screen as well as enabled.



|  |  |
| --- | --- |
| 1 | wait.until(ExpectedConditions.elementToBeClickable(By.xpath("xpath"))); |

3. alertIsPresent() – The expected condition waits for an alert box to appear.



|  |  |
| --- | --- |
| 1 | wait.until(ExpectedConditions.alertIsPresent()) !=null); |

4. textToBePresentInElement() – The expected condition waits for an element having a certain string pattern.



|  |  |
| --- | --- |
| 1 | wait.until(ExpectedConditions.textToBePresentInElement(By.id(“title’”), “text to be found”)); |

### ****98. List some scenarios which we cannot automate using Selenium WebDriver?****

1. Bitmap comparison is not possible using Selenium WebDriver  
2. Automating Captcha is not possible using Selenium WebDriver  
3. We can not read bar code using Selenium WebDriver

### ****99. What is Object Repository in Selenium WebDriver?****

Object Repository is used to store element locator values in a centralized location instead of hard coding them within the scripts. We do create a property file (.properties) to store all the element locators and these property files act as an object repository in Selenium WebDriver.

### ****100. How you build Object Repository in your project?****

In QTP, there is an Object Repository concept. When a user records a test, the objects and its properties are captured by default in an Object Repository. QTP uses this Object Repository to play back the scripts. Coming to Selenium, there is no default Object Repository concept. It doesn’t mean that there is no Object Repository in Selenium. Even though there is no default one still we could create our own. In Selenium, we call objects as locators (such as ID, Name, Class Name, Tag Name, Link Text, Partial Link Text, XPath, and CSS). Object repository is a collection of objects. One of the ways to create Object Repository is to place all the locators in a separate file (i.e., properties file). But the best way is to use Page Object Model. In the Page Object Model Design Pattern, each web page is represented as a class. All the objects related to a particular page of a web application are stored in a class.

### ****101. What is Page Object Model in Selenium?****

[Page Object Model](https://www.softwaretestingmaterial.com/page-object-model/) is a Design Pattern which has become popular in Selenium Test Automation. It is widely used design pattern in Selenium for enhancing test maintenance and reducing code duplication. Page object model (POM) can be used in any [kind of framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/) such as modular, [data-driven](https://www.softwaretestingmaterial.com/data-driven-framework-selenium-webdriver/), keyword driven, hybrid framework etc.  A page object is an object-oriented class that serves as an interface to a page of your Application Under Test(AUT). The tests then use the methods of this page object class whenever they need to interact with the User Interface (UI) of that page. The benefit is that if the UI changes for the page, the tests themselves don’t need to change, only the code within the page object needs to change. Subsequently, all changes to support that new UI is located in one place.

### ****102. What is Page Factory?****

We have seen that ‘Page Object Model’ is a way of representing an application in a test framework. For every ‘page’ in the application, we create a Page Object to reference the ‘page’ whereas a ‘Page Factory’ is one way of implementing the ‘Page Object Model’.

### ****103. What is the difference between Page Object Model (POM) and Page Factory?****

Page Object is a class that represents a web page and hold the functionality and members.  
Page Factory is a way to initialize the web elements you want to interact with within the page object when you create an instance of it.

### ****104. What are the advantages of Page Object Model Framework?****

**Code reusability** – We could achieve code reusability by writing the code once and use it in different tests.

**Code maintainability** – There is a clean separation between test code and page specific code such as locators and layout which becomes very easy to maintain code. Code changes only on Page Object Classes when a UI change occurs. It enhances test maintenance and reduces code duplication.

**Object Repository** – Each page will be defined as a java class. All the fields in the page will be defined in an interface as members. The class will then implement the interface.

**Readability** – Improves readability due to clean separation between test code and page specific code

### ****105. How can you use the Recovery Scenario in Selenium WebDriver?****

By using “Try Catch Block” within Selenium WebDriver Java tests.



|  |  |
| --- | --- |
| 1  2  3  4  5 | try {       driver.get("www.softwaretestingmaterial.com");  }catch(Exception e){       System.out.println(e.getMessage());  } |

### ****106. How to Upload a file in Selenium WebDriver?****

There are two cases which are majorly used to upload a file in Selenium WebDriver such as using SendKeys Method and using AutoIT Script.

[Practical Example.](https://www.softwaretestingmaterial.com/upload-file-using-autoit/)

### ****107. How to Download a file in Selenium WebDriver?****

By using AutoIT script, we could download a file in Selenium WebDriver.

[Practical Example](https://www.softwaretestingmaterial.com/download-file-using-autoit/)

### ****108. How to run Selenium WebDriver Test from the command line?****

[Run Java Program using Command Prompt](https://www.softwaretestingmaterial.com/run-java-program-using-command-prompt/)

[Run TestNG using Command Prompt](https://www.softwaretestingmaterial.com/run-testng-using-command-prompt/)

### ****109. How to switch between frames in Selenium?****

By using the following code, we could switch between frames.



|  |  |
| --- | --- |
| 1 | driver.switchTo().frame(); |

### ****110. How to connect a Database in selenium?****

As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.

We use JDBC Driver to connect the Database in Selenium (While using Java Programming Language).

[Practical Example](https://www.softwaretestingmaterial.com/mssql-database-testing-using-selenium/)

### ****111. How To Resize Browser Window Using Selenium WebDriver?****

To resize the browser window to particular dimensions, we use ‘Dimension’ class to resize the browser window.

[Practical Example](https://www.softwaretestingmaterial.com/resize-browser-window-using-selenium-webdriver/)

### ****112. How To Scroll Web Page Down Or UP Using Selenium WebDriver?****

JavaScript **scrollBy**() method scrolls the document by the specified number of pixels.

[Practical Example](https://www.softwaretestingmaterial.com/scroll-web-page-using-selenium-webdriver/)

### ****113. How To Perform Right Click Action (Context Click) In Selenium WebDriver?****

We use Actions class in Selenium WebDriver to do Right-Click (Context Click) action.

[Practical Example](https://www.softwaretestingmaterial.com/selenium-right-click-action/)

### ****114. How To Perform Double Click Action In Selenium WebDriver?****

We use Actions class to do Double click action in selenium.

[Practical Example](https://www.softwaretestingmaterial.com/double-click-action-selenium/)

### ****115. How To Perform Drag And Drop Action in Selenium WebDriver?****

In some applications, we may face a situation to automate drag and drop an item from one location to another location. We could not achieve these using basic elements. Selenium has provided an “Actions” class to handle this kind of scenarios. We overcome this kind of scenarios such as drag and drop using Actions Class.

To achieve this, we use Actions class in Selenium WebDriver.

[Practical Example](https://www.softwaretestingmaterial.com/drag-and-drop-using-actions-class-in-selenium/)

### ****116. How To Highlight Element Using Selenium WebDriver?****

By using JavascriptExecutor interface, we could highlight the specified element

[Practical Example](https://www.softwaretestingmaterial.com/highlight-element-using-selenium/)

### ****117. Have you used any crossbrowsertesting tool to run selenium scripts on cloud?****

I have used BrowserStack to run selenium tests on multiple browsers & Multiple operating systems in parallel. Earlier we have made a video on how to use BrowserStack to run selenium scripts on the cloud. Find the link in the description below.

### ****118. What is desired capabilities?****

In Selenium we use desired capabilities to handle SSL certificates in chrome browser

We need to create an instance of DesiredCapabilities



|  |  |
| --- | --- |
| 1 | DesiredCapabilities desiredCapability = DesiredCapabilities.chrome(); |

### ****119. What is Continuous Integration?****

Continuous Integration is abbreviated as CI. Continuous Integration is a development practice that aims to make sure the correctness of software. After each commit, a suite of tests run automatically and test the software to ensure whether the software is running without any breaks. If any test fails, we will get immediate feedback say “build is broken”.

In simple words, continuous integration is a process of verifying the correctness of a software.

Some of the continuous integration tools are Jenkins, TeamCity, Bamboo, Travis, Circle Ci, Bitbucket.

We can schedule the test suite execution using these CI Tools.

Learn how [Continuous Integration with Jenkins in Selenium works](https://www.softwaretestingmaterial.com/selenium-continuous-integration/)

### ****120. Name some CI tools available in the Market?****

Some of the best continuous testing softwares to use in your project.

* Selenium
* Katalon Studio
* Appium
* Unified Functional Testing
* Travis CI
* Egg Plant
* Watir
* Tricentis Tosca
* Test Sigma
* IBM Rational Functional Tester
* Test Complete
* QuerySurge
* JMeter
* Jenkins
* Bamboo
* Docker
* PagerDuty
* JIRA
* GitHub

Read the detailed explanation on [Continuous Testing Tools](https://www.softwaretestingmaterial.com/continuous-testing-tools/)

### ****121. How to achieve Database testing in Selenium?****

As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.

Sometimes, we may face a situation to get the data from the Database or to modify (update/delete) the data from the Database.  If we plan to automate anything outside the vicinity of a browser, then we need to use other tools to achieve our task. To achieve the Database connection and work on it, we need to use JDBC API Driver.

The Java Database Connectivity (JDBC) API provides universal data access from the Java programming language. Using the JDBC API, you can access virtually any data source, from relational databases to spreadsheets and flat files. It lets the user connect and interact with the Database and fetch the data based on the queries we use in the automation script. JDBC is a SQL level API that allows us to execute SQL statements. It creates a connectivity between Java Programming Language and the database.

Using JDBC Driver we could do the following

i. Establish a Database connection  
ii. Send SQL Queries to the Database  
iii. Process the results

### ****122. How to delete Browser Cookies with Selenium Web Driver?****



|  |  |
| --- | --- |
| 1 | driver.Manage().Cookies.DeleteAllCookies(); |

### TestNG Interview Questions:

Here we have dealt with some important TestNG interview questions. If you want to learn more interview questions related to TestNG then here you go. We have a special post on [TestNG Interview Questions](https://www.softwaretestingmaterial.com/testng-interview-questions/). Also, you could find [TestNG Complete Tutorial](https://www.softwaretestingmaterial.com/testng-tutorial/) here

### ****123. What is TestNG?****

TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing to [integration testing](https://www.softwaretestingmaterial.com/integration-testing/).

### ****124. What are the types of annotations available in TestNG?****

@BeforeTest  
@AfterTest  
@BeforeClass  
@AfterClass  
@BeforeMethod  
@AfterMethod  
@BeforeSuite  
@AfterSuite  
@BeforeGroups  
@AfterGroups  
@Test

### ****125. What is TestNG Assert and list out some common Assertions supported by TestNG?****

TestNG Asserts help us to verify the condition of the test in the middle of the test run. Based on the TestNG Assertions, we will consider a successful test only if it is completed the test run without throwing any exception.

Some of the common assertions supported by TestNG are

* assertEqual(String actual,String expected)
* assertEqual(String actual,String expected, String message)
* assertEquals(boolean actual,boolean expected)
* assertTrue(condition)
* assertTrue(condition, message)
* assertFalse(condition)
* assertFalse(condition, message)

[For Complete Post](https://www.softwaretestingmaterial.com/testng-asserts/)

### ****126. How to create and run TestNG.xml?****

In TestNG framework, we need to create T**estNG XML** file to create and handle multiple test classes. We do configure our test run, set test dependency, include or exclude any test, method, class or package and set priority etc in the XML file.

[For Complete Post](https://www.softwaretestingmaterial.com/create-testng-xml-file/)

### ****127. How to set test case priority in TestNG?****

We use priority attribute to the @Test annotations. In case priority is not set then the test scripts execute in alphabetical order.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | package TestNG;  import org.testng.annotations.\*;  public class PriorityTestCase{  @Test(priority=0)  public void testCase1() {  system.out.println("Test Case 1");  }  @Test(priority=1)  public void testCase2() {  system.out.println("Test Case 2");  }  } |

Output:



|  |  |
| --- | --- |
| 1  2 | Test Case 1  Test Case 2 |

### ****128. What is Parameterized testing in TestNG?****

*Parameterized tests* allow developers to run the same test over and over again using different values.

There are two ways to set these parameters:

* with *testng.xml -*[Practical Example](https://www.softwaretestingmaterial.com/testng-parameterization-using-xml/)
* with Data Providers  – [Practical Example](https://www.softwaretestingmaterial.com/testng-parameterization-using-dataproviders/)

### ****129. How to run a group of test cases using TestNG?****

TestNG allows you to perform sophisticated groupings of test methods. Not only can you declare that methods belong to groups, but you can also specify groups that contain other groups. Then TestNG can be invoked and asked to include a certain set of groups (or regular expressions) while excluding another set.  This gives you maximum flexibility in how you partition your tests and doesn’t require you to recompile anything if you want to run two different sets of tests back to back.

Groups are specified in your testng.xml file and can be found either under the <test> or <suite> tag. Groups specified in the <suite> tag apply to all the <test> tags underneath.



|  |  |
| --- | --- |
| 1  2  3  4 | @Test (groups = { "smokeTest", "functionalTest" })  public void loginTest(){  System.out.println("Logged in successfully");  } |

[View Complete Post](https://www.softwaretestingmaterial.com/testng-groups/)

### ****130. What is the use of @Listener annotation in TestNG?****

Ans. TestNG listeners are used to configure reports and logging. One of the most widely used listeners in TestNG is ITestListener interface. It has methods like onTestStart, onTestSuccess, onTestFailure, onTestSkipped etc. We should implement this interface creating a listener class of our own. Next, we should add the listeners annotation (@Listeners) in the Class which was created.

[Practical Example](https://www.softwaretestingmaterial.com/testng-listeners/)

### ****131. How can we create a data-driven framework using TestNG?****

By using @DataProvider annotation,  we can create a [Data-Driven Testing Framework](https://www.softwaretestingmaterial.com/data-driven-framework-selenium-webdriver/).



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | @DataProvider(name="getData")  public Object[][] getData(){  //Object [][] data = new Object [rowCount][colCount];  Object [][] data = new Object [2][2];  data [0][0] = "FirstUid";  data [0][1] = "FirstPWD";  data[1][0] = "SecondUid";  data[1][1] = "SecondPWD";  return data;  } |

[Practical Example](https://www.softwaretestingmaterial.com/testng-parameterization-using-dataproviders/)

### ****132. Where you have applied OOPS in Automation Framework?****

Here we have given a clear explanation of the application of [OOPs in Automation Framework](https://www.softwaretestingmaterial.com/oops-concept-in-automation-framework/)

### ****133. How to handle browser (chrome) notifications in Selenium?****

In Chrome, we can use ChromeOptions as shown below.



|  |  |
| --- | --- |
| 1  2  3 | ChromeOptions options = new ChromeOptions();  options.addArguments("disable-infobars");  WebDriver player = new ChromeDriver(options); |

I would like to conclude this post here. Final words, Bookmark this post “Selenium Testing Interview Questions” for future reference. We keep on updating this post based on user requests.

### ****134. Mention types of data you have handled in Selenium?****

* Excel
* CSV
* XML
* JSON
* YAML
* SQL

# 8.ArtOfLearning 100Q: <https://artoftesting.com/selenium-interview-questions>

### ****Ques.1. What is Selenium?****

Ans. Selenium is a robust test automation suite that is used for automating web-based applications. It supports multiple browsers, programming languages, and platforms.

### ****Ques.2. What are the different forms of Selenium?****

Ans. Selenium comes in four forms-

1. Selenium WebDriver – Selenium WebDriver is used to automate web applications by directly calling the browser’s native methods.
2. The Selenium IDE Plugin – Selenium IDE is an open-source test automation tool that works on record and playback principles.
3. Selenium RC component – Selenium Remote Control(RC) is officially deprecated by Selenium and it used to work using javascript to automate the web applications.
4. Selenium Grid – Allows Selenium tests to run in parallel across multiple machines.

### ****Ques.3. What are some advantages of Selenium?****

Ans. Following are the advantages of Selenium-

1. Selenium is open source and free to use without any licensing cost.
2. It supports multiple languages like Java, Ruby, Python, etc.
3. Selenium supports multi-browser testing.
4. It has vast resources and helping community over the internet.
5. Using Selenium IDE component, non-programmers can also write automation scripts.
6. Using the Selenium Grid component, distributed testing can be carried out on remote machines.

### ****Ques.4. What are some limitations of Selenium?****

Ans. Following are the limitations of Selenium-

1. We cannot test desktop applications using Selenium.
2. We cannot test web services using Selenium.
3. For creating robust scripts in Selenium Webdriver, programming language knowledge is required.
4. Also, we have to rely on external libraries and tools for performing tasks like – logging(log4J), testing framework-(TestNG, JUnit), reading from external files(POI for excels), etc.

### ****Ques.5. Which browsers/drivers are supported by Selenium Webdriver?****

Ans. Some commonly used browsers supported by Selenium are-

1. Google Chrome – ChromeDriver
2. Firefox – FireFoxDriver
3. Internet Explorer – InternetExplorerDriver
4. Safari – SafariDriver
5. HtmlUnit (Headless browser) – HtmlUnitDriver
6. Android – Selendroid/Appium
7. IOS – ios-driver/Appium

### ****Ques.6. Can we test APIs or web services using Selenium Webdriver?****

Ans. No. Selenium WebDriver uses the browser’s native method to automate the web applications. So, there is no support for testing web services using Selenium WebDriver.

### ****Ques.7. What are the various ways of locating an element in Selenium?****

Ans. The different locators in Selenium are-

1. Id
2. XPath
3. CSS selector
4. className
5. tagName
6. name
7. link text
8. partialLinkText

### ****Ques.8. How can we inspect the web element attributes in order to use them in different locators?****

Ans. In order to locate web elements, we can use the Developer tool and plugins like Firebug.  
The developer tool can be launched by pressing F12 on the browser. Users can easily hover over any element and find its different HTML properties.  
  
Firebug is a plugin of Firefox that provides various development tools for debugging applications. From an automation perspective, Firebug is used specifically for inspecting web elements in order to find their attributes like id, class, name, etc. in different locators.

### ****Ques.9. What is an XPath?****

Ans. Xpath or XML path is a query language that is used for selecting nodes from XML documents. Also, it is one of the locators supported by Selenium Webdriver.

### ****Ques.10. What is an absolute XPath?****

Ans. An absolute XPath is a way of locating an element using an XML expression, beginning from the root node i.e. HTML node in the case of web pages.  
  
The main disadvantage of absolute XPath is that even if there is a slight change in the UI or any element then also whole XPath will fail.  
Example – html/body/div/div[2]/div/div/div/div[1]/div/input

### ****Ques.11. What is a relative XPath?****

Ans. A relative XPath is a way of locating an element using an XML expression, starting from anywhere in the HTML document.  
  
In this way, there are different ways of creating robust relative XPaths that are unaffected by changes in other UI elements.  
Example – //input[@id=’username’]  
  
**For details, check –**[XPath Tutorial](https://artoftesting.com/xpath-in-selenium-tutorial)

### ****Ques.12. What is the difference between single slash(/) and a double slash(//) in XPath?****

Ans. In XPath, a single slash is used for creating absolute XPaths, beginning from the root node. Whereas double slash is used for creating relative XPaths.

### ****Ques.13. How can we locate an element by only partially matching the value of its attributes in Xpath?****

Ans. Using contains() method we can locate an element by partially matching its attribute’s value. This is particularly helpful in scenarios where the attributes have dynamic values with a certain constant part.

xPath expression = //\*[contains(@name,'user')]

Basically, the above statement will match all the values of the name attribute containing the word ‘user’ in them.

### ****Ques.14. How can we locate elements using their text in XPath?****

Ans. Using the text() method –

xPathExpression = //\*[text()='username']

### ****Ques.15. How can we move to the parent of an element using XPath?****

Ans. Using ‘/..’ after the XPath expression of the child element, we can move to the parent of an element.  
For example, the locator **//div[@id=”childId”]/..** will move to the parent of the div element with id value as ‘childId’.

### ****Ques.16. How can we move to the nth-child element using XPath?****

Ans. Basically, there are two ways of navigating to the nth element using XPath-

* Using square brackets with index position-  
  Example – div[2] will find the second div element.
* Using position()-  
  Example – div[position()=3] will find the third div element.

### ****Ques.17. What is the syntax of finding elements by class using CSS Selector?****

Ans. By using **.className** in the CSS locator, we can select all the elements belonging to a particular class e.g. ‘.red’ will select all elements having class ‘red’.

### ****Ques.18. What is the syntax of finding elements by id using CSS Selector?****

Ans. By using **#idValue** in the CSS locator, we can select all the elements belonging to a particular class e.g. ‘#userId’ will select the element having an id – userId.

### ****Ques.19. How can we select elements by their attribute value using the CSS Selector?****

Ans. Using **[attribute=value]** in the CSS locator, we can select all the elements belonging to a particular class e.g. ‘[type=small]’ will select the element having attribute type of value ‘small’.

### ****Ques.20. How can we move to the nth-child element using the CSS selector?****

Ans. Using**:nth-child(n)** in the CSS locator, we can move to the nth child element e.g. div:nth-child(2) will locate 2nd div element of its parent.

### ****Ques.21. What is the fundamental difference between XPath and CSS selectors?****

Ans. The fundamental difference between XPath and CSS selector is – using XPaths we can traverse up in the document i.e. we can move to parent elements. Whereas using the CSS selector, we can only move downwards in the document.

## Selenium Java Interview Questions

### ****Ques.22. How can we launch different browsers in Selenium WebDriver?****

Ans. By creating an instance of the desired browser driver e.g. below command will initialize the Firefox browser.

WebDriver driver = new FirefoxDriver();

### ****Ques.23. What is the use of driver.get(“URL”) and driver.navigate().to(“URL”) commands? Is there any difference between the two?****

Ans. Both driver.get(“URL”) and driver.navigate().to(“URL”) commands are used to navigate to a URL passed as parameter.  
There is a minor difference between the two commands-

1. driver.navigate() allows moving back and forward in browser history with the help of driver.navigate().forward() and driver.navigate().back() commands.
2. In the case of single-page applications (where the URL is appended by ‘#’ to navigate to different sections of the page), driver.navigate().to() navigates to a particular section by changing the URL without refreshing the page whereas driver.get() refreshes the page also.  
     
   This refreshing of the page is also the primary reason because of which history is not maintained in the case of the driver.get() command.  
   **Reference –**[Stack overflow](https://stackoverflow.com/questions/33865618/difference-between-webdriver-get-and-webdriver-navigate-to-in-the-case-of-ur/33868976#33868976)

### ****Ques.24. How can we type text in a textbox element using Selenium?****

Ans. With the help of sendKeys() method we can type text in a textbox-

WebElement searchTextBox = driver.findElement(By.id("srch"));

searchTextBox.sendKeys("searchTerm");

### ****Ques.25. How can we clear a text written in a textbox?****

Ans. In order to delete the text written in a textbox, we can use the clear() method.

driver.findElement(By.id("elementLocator")).clear();

### ****Ques.26. How to check a checkBox in Selenium?****

Ans. The same click() method used for clicking buttons or radio buttons can be used for checking the checkbox as well.

### ****Ques.27. How can we submit a form in Selenium?****

Ans. Using the submit() method we can submit a form in selenium.

driver.findElement(By.id("form1")).submit();

Also, the click() method can be used for the same purpose.

### ****Ques.28. Explain the difference between close and quit command.****

Ans. The difference between close and quit command is-  
driver.close() – Used to close the current browser having a focus.  
driver.quit() – Used to close all the browser instances.

### ****Ques.29. How to switch between multiple windows in Selenium?****

Ans. Selenium has **driver.getWindowHandles()** and **driver.switchTo().window(“{windowHandleName}”)** commands to work with multiple windows.  
  
The getWindowHandles() command returns a list of ids corresponding to each window. If we pass a particular window handle to the **driver.switchTo().window(“{windowHandleName}”)** command then we can switch control/focus to that particular window.

for (String windowHandle : driver.getWindowHandles()) {

driver.switchTo().window(handle);

}

### ****Ques.30. What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in Selenium?****

Ans. The driver.getWindowHandle() returns a handle of the current window (a single unique identifier).  
Whereas driver.getWindowHandles() returns a set of handles of all the windows available.

### ****Ques.31. How can we move to a particular frame in Selenium?****

Ans. The driver.switchTo() commands can be used for switching to a particular iframe.

driver.switchTo().frame("{frameIndex/frameId/frameName}");

For locating a frame, we can either use the index (starting from 0), its name, or its Id.

### ****Ques.32. Can we move back and forward in the browser using Selenium?****

Ans. Yes, using driver.navigate().back() and driver.navigate().forward() commands, we can move backward and forward in a browser.

### ****Ques.33. What are the different ways to refresh a browser?****

Ans. There a multiple ways to refresh a page in Selenium-

* Using driver.navigate().refresh() command.
* Using sendKeys(Keys.F5) on any textbox on the webpage.
* Using driver.get(“URL”) on the current URL or using driver.getCurrentUrl().
* Using driver.navigate().to(“URL”) on the current URL or driver.navigate().to(driver.getCurrentUrl());

### ****Ques.34. How can we maximize the browser window in Selenium?****

Ans. We can maximize the browser window using the following command-

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

### ****Ques.35. How can we fetch a text written over an element?****

Ans. Using the getText() method we can fetch the text over an element.

String text = driver.findElement("elementLocator").getText();

### ****Ques.36. How can we find the value of different attributes like name, class, value of an element?****

Ans. Using getAttribute(“{attributeName}”) method, we can find the value of different attributes of an element e.g.-

String valueAttribute =

driver.findElement(By.id("locator")).getAttribute("value");

### ****Ques.37. How to delete cookies in Selenium?****

Ans. Using deleteAllCookies() method.

driver.manage().deleteAllCookies();

### ****Ques.38. What is an implicit wait in Selenium?****

Ans. An implicit wait is a type of wait that waits for a specified time while locating an element before throwing NoSuchElementException. By default, Selenium tries to find web elements immediately when required without any wait. So, it is good to use implicit wait. This wait is applied to all the elements of the current driver instance.

driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

### ****Ques.39. What is an explicit wait in Selenium?****

Ans. An explicit wait is a type of wait that is applied to a particular web element until the expected condition specified is met.

WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("elementId")));

It is advisable to use explicit waits over implicit waits because higher timeout value of implicit wait (set for handling only some of the elements) gets applied to all the web elements. Thus increasing the overall execution time of the script. On the other hand, we can apply different timeouts to the different elements in case of explicit waits.

**Check our detailed tutorial here –**[Implicit & Explicit Waits in Selenium](https://artoftesting.com/waits-in-selenium-webdriver).

### ****Ques.40. What are some expected conditions that can be used in Explicit waits?****

Ans. Some of the commonly used expected conditions of an element that can be used with explicit waits are-

* elementToBeClickable(WebElement element or By locator)
* stalenessOf(WebElement element)
* visibilityOf(WebElement element)
* visibilityOfElementLocated(By locator)
* invisibilityOfElementLocated(By locator)
* attributeContains(WebElement element, String attribute, String value)
* alertIsPresent()
* titleContains(String title)
* titleIs(String title)
* textToBePresentInElementLocated(By, String)

### ****Ques.41. What is a fluent wait?****

Ans. A fluent wait is a type of wait in which we can also specify polling interval (the time intervals after which driver will try to find the elements when not located) along with the maximum timeout value.

Wait wait = new FluentWait(driver)

.withTimeout(20, SECONDS)

.pollingEvery(5, SECONDS)

.ignoring(NoSuchElementException.class);

WebElement textBox = wait.until(new Function<webdriver,webElement>() {

public WebElement apply(WebDriver driver) {

return driver.findElement(By.id("textBoxId"));

}

}

);

### ****Ques.42. What are the different keyboard operations that can be performed in Selenium?****

Ans. The different keyboard operations that can be performed in Selenium are-

1. **.sendKeys(“sequence of characters”)** – Used for passing character sequence to an input or textbox element.
2. **.pressKey(“non-text keys”)** – Used for keys like control, function keys etc that are non-text.
3. **.releaseKey(“non-text keys”)** – Used in conjunction with keypress event to simulate releasing a key from keyboard event.

### ****Ques.43. What are the different mouse actions that can be performed using Selenium?****

Ans. The different mouse events supported in Selenium are-

1. click(WebElement element)
2. doubleClick(WebElement element)
3. contextClick(WebElement element)
4. mouseDown(WebElement element)
5. mouseUp(WebElement element)
6. mouseMove(WebElement element)
7. mouseMove(WebElement element, long xOffset, long yOffset)

### ****Ques.44. Write the code to double-click an element****.

Ans. Code to double click an element-

Actions action = new Actions(driver);

WebElement element=driver.findElement(By.id("elementId"));

action.doubleClick(element).perform();

### ****Ques.45. Write the code to right-click an element****.

Ans. Code to right-click an element in selenium-

Actions action = new Actions(driver);

WebElement element=driver.findElement(By.id("elementId"));

action.contextClick(element).perform();

### ****Ques.46. How to mouse hover an element in Selenium?****

Ans. Code to mouse hover over an element-

Actions action = new Actions(driver);

WebElement element=driver.findElement(By.id("elementId"));

action.moveToElement(element).perform();

### ****Ques.47. How to fetch the current page URL in Selenium?****

Ans. In order to fetch the current page URL, we can use the getCurrentURL() command.

driver.getCurrentUrl();

### ****Ques.48. How can we fetch the title of the page in Selenium?****

Ans. Using **driver.getTitle()** command, we can fetch the page title in Selenium. This method returns a string containing the title of the webpage.

### ****Ques.49. How can we fetch the page source in Selenium?****

Ans. Using the **driver.getPageSource()** command, we can fetch the page source in selenium. This method returns a string containing the page source.

### ****Ques.50. How to verify tooltip text using Selenium?****

Ans. Tooltips web elements have an attribute of type ‘title’. By fetching the value of the ‘title’ attribute, we can verify the tooltip text in selenium.

String toolTipText = element.getAttribute("title");

**Also, check –**[Top SQL Queries Asked in Interviews](https://artoftesting.com/sql-queries-for-interview)

### ****Ques.51. How to locate a link using its text in Selenium?****

Ans. Using linkText() and partialLinkText() methods, we can locate a link. The difference between the two is – linkText() matches the complete string passed as a parameter to the link texts. Whereas partialLinkText() only matches the string parameter partially.

WebElement link1 = driver.findElement(By.linkText("artOfTesting"));

WebElement link2 = driver.findElement(By.partialLinkText("artOf"));

### ****Ques.52. What are DesiredCapabilities in Selenium WebDriver?****

Ans. Desired capabilities are a set of key-value pairs that are used for storing or configuring browser-specific properties. For example – browser’s version, platform, etc in the browser instances.

### ****Ques.53. How can we find all the links on a web page?****

Ans. All the links are of anchor tag ‘a’. So by locating elements of tagName ‘a’ we can find all the links on a webpage.

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

### ****Ques.54. What are some commonly encountered exceptions in Selenium?****

Ans. Some of the commonly seen exceptions in Selenium are-

* NoSuchElementException – When no element could be located by the locator provided.
* ElementNotVisibleException – When an element is present in the DOM but is not visible.
* NoAlertPresentException – When we try to switch to an alert box but the targetted alert is not present.
* NoSuchFrameException – When we try to switch to a frame but the targetted frame is not present.
* NoSuchWindowException – When we try to switch to a window but the targetted window is not present.
* UnexpectedAlertPresentException – When an unexpected alert blocks the normal interaction of the driver.
* TimeoutException – When a command execution gets a timeout.
* InvalidElementStateException – When the state of an element is not appropriate for the desired action.
* NoSuchAttributeException – When we are trying to fetch an attribute’s value but the attribute is not correct.
* WebDriverException – When there is some issue with the driver instance preventing it from getting launched.

### ****Ques.55. How can we capture screenshots using Selenium?****

Ans. In order to take screenshots in Selenium, we can use the getScreenshotAs method of the TakesScreenshot interface.

File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(scrFile, new File("D:\\testScreenShot.jpg"));

### ****Ques.56. How to handle dropdowns in Selenium?****

Ans. Using Select class-

Select countriesDropDown = new Select(driver.findElement(By.id("countries")));

dropdown.selectByVisibleText("India");

//or using index of the option starting from 0

dropdown.selectByIndex(1);

//or using its value attribute

dropdown.selectByValue("Ind");

### ****Ques.57. How to check which option in the dropdown is selected?****

Ans. Using is Selected() method, we can check the state of a dropdown’s option.

Select countriesDropDown = new Select(driver.findElement(By.id("countries")));

dropdown.selectByVisibleText("India");

//returns true or false value

System.out.println(driver.findElement(By.id("India")).isSelected());

### ****Ques.58. How can we check if an element is getting displayed on a web page?****

Ans. Using the isDisplayed() method we can check if an element is getting displayed on a web page.

driver.findElement(By locator).isDisplayed();

### ****Ques.59. How can we check if an element is enabled for interaction on a web page?****

Ans. Using the isEnabled method, we can check if an element is enabled or not.

driver.findElement(By locator).isEnabled();

### ****Ques.60. What is the difference between driver.findElement() and driver.findElements() commands?****

Ans. The difference between driver.findElement() and driver.findElements() commands is-

* findElement() returns a single WebElement (found first) based on the locator passed as a parameter. Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.
* Syntax of findElement()-  
  WebElement textbox = driver.findElement(By.id(“textBoxLocator”));  
    
  Whereas the syntax of findElements()-  
  List <WebElement> elements = driver.findElements(By.id(“value”));
* Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of 0 elements.

### ****Ques.61. How can we handle window UI elements and window POP ups using selenium?****

Ans. Selenium is used for automating web-based applications only(or browsers only). If we want to handle window GUI elements then we can use tools like AutoIT.  
  
AutoIT is a freeware used for automating window GUI. The AutoIt scripts follow simple BASIC language like syntax. Also, it can be easily integrated with Selenium tests.

### ****Ques.62. What is Robot API?****

Ans. Robot API is used for handling Keyboard or mouse events.

Robot robot = new Robot();

//Simulate enter key action

robot.keyPress(KeyEvent.VK\_ENTER);

### ****Ques.63. How to do file upload in Selenium?****

Ans. File upload action can be performed in multiple ways-

1. Using element.sendKeys(“path of file”) on the webElement of input tag and type file i.e. the elements should be like –  
   <input type=”file” name=”fileUpload”>
2. With the help of Robot API.
3. Using the AutoIT API.

### ****Ques.64. How to handle the HTTPS website in Selenium or how to accept the SSL untrusted connection?****

Ans. Using profiles, we can handle accept the SSL untrusted connection certificate. Profiles are basically set of user preferences stored in a file.

FirefoxProfile profile = new FirefoxProfile();

profile.setAcceptUntrustedCertificates(true);

profile.setAssumeUntrustedCertificateIssuer(false);

WebDriver driver = new FirefoxDriver(profile);

### ****Ques.65. How to do drag and drop in Selenium?****

Ans. Using Action class, [drag and drop can be performed in Selenium](https://artoftesting.com/drag-and-drop-in-selenium-webdriver). Sample code-

Actions builder = new Actions(driver);

Action dragAndDrop = builder.clickAndHold(SourceElement)

.moveToElement(TargetElement)

.release(TargetElement)

.build();

dragAndDrop.perform();

### ****Ques.66. How to execute JavaScript code in Selenium?****

Ans. JavaScript code can be executed in Selenium using JavaScriptExecuter. Sample code for javascript execution-

WebDriver driver = new FireFoxDriver();

if (driver instanceof JavascriptExecutor) {

((JavascriptExecutor)driver).executeScript("{JavaScriptCode}");

}

### ****Ques.67. How to handle alerts in Selenium?****

Ans. In order to accept or dismiss an alert box, the alert class is used. This requires first switching to the alert box and then using accept() or dismiss() command as the case may be.

Alert alert = driver.switchTo().alert();

//To accept the alert

alert.accept();

Alert alert = driver.switchTo().alert();

//To cancel the alert box

alert.dismiss();

### ****Ques.68. What is HtmlUnitDriver?****

Ans. HtmlUnitDriver is the fastest WebDriver. Unlike other drivers (FireFoxDriver, ChromeDriver, etc), the HtmlUnitDriver is non-GUI. On executing test scripts, no browser gets launched.

### ****Ques.69. How to handle hidden elements in Selenium WebDriver?****

Ans. Using javaScript executor we can handle hidden elements-

(JavascriptExecutor(driver))

.executeScript("document.getElementsByClassName(locator).click();");

## Selenium Framework Interview Questions

### ****Ques.70. What is Page Object Model or POM?****

Ans. Page Object Model(POM) is a design pattern in Selenium. A design pattern is a solution or a set of standards that are used for solving commonly occurring software problems.  
  
Now coming to POM – POM helps to create a framework for maintaining selenium scripts. In POM for each page of the application, a class is created having the web elements belonging to the page and methods handling the events on that page. The test scripts are maintained in separate files and the methods of the page object files are called from the test scripts file.  
  
In this way, we can create a robust automation framework using POM.

### ****Ques.71. What are the advantages of POM?****

Ans. The advantages are POM are-

1. Using POM, we can create an Object Repository i.e. a set of web elements in separate files along with their associated functions. In this way, keeping the code clean.
2. For any change in UI(or web elements) only page object files are required to be updated leaving test files unchanged.
3. It makes code reusable as well as maintainable.

### ****Ques.72. What is Page Factory?****

Ans. Page factory is an implementation of the Page Object Model in Selenium. It provides @FindBy annotation to find web elements. In addition, there is a PageFactory.initElements() method to initialize all web elements defined with @FindBy annotation.

public class SamplePage {

WebDriver driver;

@FindBy(id="search")

WebElement searchTextBox;

@FindBy(name="searchBtn")

WebElement searchButton;

//Constructor

public samplePage(WebDriver driver){

this.driver = driver;

//initElements method to initialize all elements

PageFactory.initElements(driver, this);

}

//Sample method

public void search(String searchTerm){

searchTextBox.sendKeys(searchTerm);

searchButton.click();

}

}

### ****Ques.73. What is an Object repository?****

Ans. An object repository is the centralized location of all the objects or WebElements of the test scripts. In Selenium, we can implement an object repository using the Page Object Model as well as Page Factory design patterns.

### ****Ques.74. What is a data-driven framework?****

Ans. A data-driven framework is one in which the test data is put in external files like CSV, Excel, etc. Basically, the test data is separated from the test logic that is written in test script files. The test data drives the test cases, i.e. the test methods run for each set of test data values.  
  
TestNG provides inherent support for data-driven testing using @dataProvider annotation.

**Also, check –**[Test Automation Framework](https://artoftesting.com/test-automation-framework)

### ****Ques.75. What is a keyword-driven**** ****framework?****

Ans. A keyword-driven framework is one in which the normal set of actions are associated with keywords and are kept in external files usually in tabular form.  
  
For example, an action of launching a browser will be associated with keyword – launchBrowser(), action to write in a textbox with keyword – writeInTextBox(webElement, textToWrite), etc.  
  
The code to perform the action based on a keyword specified in the external file is implemented in the framework itself.  
  
In this way, the test steps can be written in a file by a person of a non-programming background also (provided all the used keywords are implemented in the framework).

### ****Ques.76. What is a hybrid framework?****

Ans. A hybrid framework is a combination of two or more frameworks. For example, a combination of data-driven and keyword-driven frameworks can be considered as a hybrid framework.

### ****Ques.77. What is Selenium Grid?****

Ans. Selenium Grid is a tool that helps in the distributed testing. Using Grid, we can run test scripts in different machines having different browsers, browser versions, platforms, etc in parallel. In the Selenium grid, there is a hub that is a central server managing all the distributed machines known as nodes.

### ****Ques.78. What are some advantages of the Selenium grid?****

Ans. The advantages of the Selenium grid are-

1. It allows running test cases in parallel thereby saving test execution time.
2. Multi-browser testing is possible using the Selenium grid by running the test on machines having different browsers.
3. Also, we can do multi-platform testing by configuring nodes having different operating systems.

### ****Ques.79. What is a hub in the Selenium Grid?****

Ans. A hub is a server or a central point in the Selenium grid that controls the test executions on the different machines.

### ****Ques.80. What is a node in the Selenium Grid?****

Ans. Nodes are the machines that are attached to the selenium grid hub and have selenium instances running the test scripts. Unlike a hub, there can be multiple nodes in the selenium grid.

### ****Ques.81. Explain the line of code Webdriver driver = new FirefoxDriver();.****

Ans. In the line of code **Webdriver driver = new FirefoxDriver();** ‘WebDriver’ is an interface and we are creating an object of type WebDriver instantiating an object of FirefoxDriver class.

### ****Ques.82 What is the purpose of creating a reference variable- ‘driver’ of type WebDriver instead of directly creating a FireFoxDriver object or any other driver’s reference in the statement Webdriver driver = new FirefoxDriver();?****

Ans. By creating a reference variable of type WebDriver, we can use the same variable to work with multiple browsers like ChromeDriver, IEDriver, etc.

### ****Ques.83. Name an API used for reading and writing data to excel files.****

Ans. Apache POI API and JXL(Java Excel API) can be used for reading, writing, and updating excel files.

### ****Ques.84. Name an API used for logging in Java.****

Ans. Log4j is an open-source API widely used for logging in Java. It supports multiple levels of logging like – ALL, DEBUG, INFO, WARN, ERROR, TRACE, and FATAL.

### ****Ques.85. What is the use of logging in automation?****

Ans. Logging helps in debugging the tests when required and also provides storage of the test’s runtime behavior.

### ****Ques.86. What is TestNG?****

Ans. TestNG(NG for Next Generation) is a testing framework that can be integrated with Selenium or any other automation tool. Moreover, it provides multiple capabilities like assertions, reporting, parallel test execution, etc.

### ****Ques.87. What are some advantages of TestNG?****

Ans. Following are the advantages of TestNG-

1. TestNG provides different assertions that help in checking the expected and actual results.
2. It provides parallel execution of test methods.
3. We can define the dependency of one test method over others in TestNG.
4. Also, we can assign priority to test methods in selenium.
5. It allows the grouping of test methods into test groups.
6. It allows data-driven testing using @DataProvider annotation.
7. TestNG has inherent support for reporting.
8. It has support for parameterizing test cases using @Parameters annotation.

### ****Ques.88. What are commonly used TestNG annotations?****

Ans. The commonly used TestNG annotations are-

* @Test – The @Test annotation **marks a method as a test method**.
* @BeforeSuite – The annotated method will run only **once before all tests in this suite have run**.
* @AfterSuite -The annotated method will run only once **after all tests in this suite have run**.
* @BeforeClass – The annotated method will run **only once before the first test method in the current class is invoked**.
* @AfterClass – The annotated method will run **only once after all the test methods in the current class have been run**.
* @BeforeTest – The annotated method will run **before any test method belonging to the classes inside the <test> tag is run**.
* @AfterTest – The annotated method will run **after all the test methods** **belonging to the classes** **inside the <test> tag have run**.
* **@**BeforeMethod – The annotated method will run **before each test method** marked by @Test annotation.
* **@**AfterMethod – The annotated method will run **after each test method** marked by @Test annotation.
* @DataProvider-The @DataProvider annotation is used to **pass test data to the test method**. The test method will run as per the number of rows of data passed via the data provider method.

### ****Ques.89. What are some common assertions provided by TestNG?****

Ans. Some of the common assertions provided by testNG are-

1. assertEquals(String actual, String expected, String message) – (and other overloaded data type in parameters)
2. assertNotEquals(double data1, double data2, String message) – (and other overloaded data type in parameters)
3. assertFalse(boolean condition, String message)
4. assertTrue(boolean condition, String message)
5. assertNotNull(Object object)
6. fail(boolean condition, String message)
7. true(String message)

### ****Ques.90. What is the use of the testng.xml file?****

Ans. A testng.xml file is used for configuring the whole test suite. In this file, we can create a test suite, create test groups, mark tests for parallel execution, add listeners, and pass parameters to test scripts. Later, this testng.xml file can be used for triggering the test suite.

### ****Ques.91. How can we pass the parameter to test script using TestNG?****

Ans. Using @Parameter annotation and ‘parameter’ tag in testng.xml we can pass parameters to the test script.  
Sample testng.xml –

<suite name="sampleTestSuite">

<test name="sampleTest">

<parameter name="sampleParamName" value="sampleValue"/>

<classes>

<class name="TestFile" />

</classes>

</test>

</suite>

Sample test script-

public class TestFile {

@Test

@Parameters("sampleParamName")

public void parameterTest(String paramValue) {

System.out.println("Value of sampleParamName is - " + sampleParamName);

}

### ****Ques.92. How can we create a data-driven framework using TestNG?****

Ans. Using @DataProvider we can create a data-driven framework. Basically, we can pass test data to the associated test method and then multiple iterations of the test run for the different test data values passed from the @DataProvider method. The method annotated with @DataProvider annotation return a 2D array of object.

//Data provider returning 2D array of 3\*2 matrix

@DataProvider(name = "dataProvider1")

public Object[][] dataProviderMethod1() {

return new Object[][] {{"kuldeep","rana"}, {"k1","r1"},{"k2","r2"}};

}

//This method is bound to the above data provider returning 2D array of 3\*2 matrix

//The test case will run 3 times with different set of values

@Test(dataProvider = "dataProvider1")

public void sampleTest(String s1, String s2) {

System.out.println(s1 + " " + s2);

}

### ****Ques.93. What is the use of @Listener annotation in TestNG?****

Ans. Listeners are used for performing some action in case an event gets triggered. Usually, testNG listeners are used for configuring reports and logging. One of the most widely used listeners in testNG is ITestListener interface.  
  
It has methods like onTestSuccess, onTestFailure, onTestSkipped, etc. We need to implement this interface creating a listener class of our own. After that using the @Listener annotation, we can specify that for a particular test class, a customized listener class should be used.

@Listeners(PackageName.CustomizedListenerClassName.class)

public class TestClass {

WebDriver driver= new FirefoxDriver();

@Test

public void testMethod(){

//test logic

}

}

### ****Ques.94. How can we make one test method dependent on others using TestNG?****

Ans. Using the dependsOnMethods parameter inside @Test annotation in TestNG, we can make one test method run only after the successful execution of the dependent test method.

@Test(dependsOnMethods = { "preTests" })

### ****Ques.95. How can we set the priority of test cases in TestNG?****

Ans. Using the priority parameter in @Test annotation in TestNG we can define the priority of test cases. The default priority of the test when not specified is integer value 0. Example-

@Test(priority=1)

### ****Ques.96. What is the default priority of a test method in TestNG?****

Ans. The default priority of a test when not specified is integer value 0. So, if we have one test case with priority 1 and one without any priority then the test without any priority value will get executed first.

### ****Ques.97. How to prevent a test case from running using TestNG?****

Ans. A Test method can be disabled from getting executed by setting the “enabled” attribute as false.

//In case of a test method

@Test(enabled = false)

public void testMethod1() {

//Test logic

}

//In case of test method belonging to a group

@Test(groups = {"NegativeTests"}, enabled = false)

public void testMethod2() {

//Test logic

}

### ****Ques.98. How can we run test cases in parallel using TestNG?****

Ans. In order to run the tests in parallel just add these two key-value pairs in the suite-

* parallel=”{methods/tests/classes}”
* thread-count=”{number of thread you want to run simultaneously}”.

<suite name="ArtOfTestingSuite" parallel="methods" thread-count="5">

Check [Running Selenium Tests in parallel](https://artoftesting.com/run-selenium-test-in-parallel) for details.

### ****Ques.99. What is the use of @Factory annotation in TestNG?****

Ans. @Factory annotation helps in the dynamic execution of test cases. Using @Factory annotation, we can pass parameters to the whole test class at run time. The parameters passed can then be used by one or more test methods of that class.  
  
For example – there are two classes TestClass and the TestFactory class. Because of the @Factory annotation, the test methods in class TestClass will run twice with the data “k1” and “k2”.

public class TestClass{

private String str;

//Constructor

public TestClass(String str) {

this.str = str;

}

@Test

public void TestMethod() {

System.out.println(str);

}

}

public class TestFactory{

//The test methods in class TestClass will run twice with data "k1" and "k2"

@Factory

public Object[] factoryMethod() {

return new Object[] { new TestClass("K1"), new TestClass("k2") };

}

}

### ****Ques.100. What is the difference between @Factory and @DataProvider annotation?****

Ans. @Factory method creates instances of test class and runs all the test methods in that class with a different set of data.  
Whereas, @DataProvider is bound to individual test methods and run the specific methods multiple times.

# 9. JavaTPoint 50Q: <https://www.javatpoint.com/selenium-interview-questions>

### 1) What is test automation or automation testing?

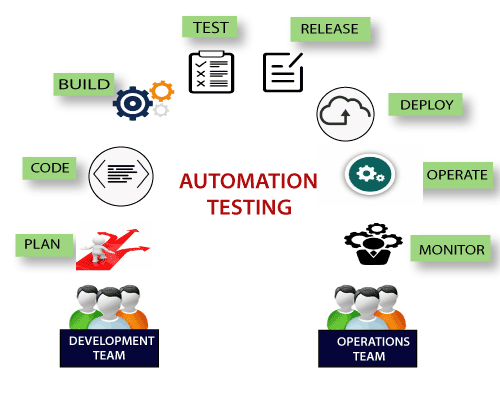
Automation testing uses automation tools to write and execute test cases, no manual involvement is necessary for executing an automated test suite. Testers prefer automation tools to write test scripts and test cases and then group into test suites.

6M

65

Facebook Temporarily Allows Violent Anti-Russian Posts

Automation testing enables the use of specialized tools to automate the execution of manually designed test cases without any human intervention. Automation testing tools can access the test data, controls the execution of tests and compares the actual result against the expected result. Consequently, generating detailed test reports of the system under test.



### 2) What are the advantages of automation testing?

Some basic Advantages of automation testing are as follows.

* Automation testing supports both functional and performance test on an application.
* It supports the execution of repeated test cases.
* It facilitates parallel execution.
* It aids in testing a large test matrix.
* It improves accuracy because there are no chances of human errors.
* It saves time and money.

### 3) Name some of the commonly used Automation Testing tools that are used for Functional Automation.

Lists of top 10 used automation testing tools for Functional Automation are as follows.

* Teleric Test Studio, Developed by Teleric.
* TestingWhiz
* HPE Unified Functional Testing (HP - UFT formerly QTP)
* Tosca Testsuite
* Watir
* Quick Test Professional, provided by HP.
* Rational Robot, provided by IBM.
* Coded UI, provided by Microsoft.
* Selenium, open source.
* Auto It, Open Source.

### 4) Name some of the commonly used Automation Testing tools that are used for Non-Functional Automation.

Lists of some commonly used Automation Testing tools for Non-Functional Automation are as follows.

* Load Runner, provided by Hp.
* JMeter, provided by Apache.
* Burp Suite, provided by PortSwigger.
* Acunetix, provided by Acunetix.

### 5) What is Selenium?

Selenium is a portable framework for software testing. Selenium tool facilitates with a playback tool for authoring functional tests without the need to learn a test scripting language.

Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. **Jason Huggins** developed Selenium in 2004 as an internal tool at **Thought Works**. Selenium supports automation across different browsers, platforms, and programming languages.

### 6) What are the different components of Selenium?

Selenium is not just a single tool but a suite of software's, each having a different approach to support automation testing. It comprises of four major components which include:

1. Selenium Integrated Development Environment (IDE)
2. Selenium Remote Control (Now Deprecated)
3. WebDriver
4. Selenium Grid

### 7) List out the names of programming languages, browsers and operating systems that are supported by Selenium.

Selenium supports various operating systems, browsers and programming languages. Following is the list:

* **Programming Languages**: C#, Java, Python, PHP, Ruby, Perl, JavaScript.
* **Operating Systems**: Android, iOS, Windows, Linux, Mac, Solaris.
* **Browsers**: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.

### 8) What are the significant changes/upgrades in various Selenium versions?

**Selenium v1.0:**

* Version 1.0 was the initial release of Selenium.
* It included three tools: Selenium IDE, Selenium RC, and Selenium Grid.

**Selenium v2.0:**

* Selenium WebDriver was introduced replacing Selenium RC in version "2.0".
* With the onset of WebDriver, RC got deprecated and moved to the legacy package.

**Selenium v3:**

* The latest release Selenium 3 has new added features and functionalities.
* It includes Selenium IDE, Selenium WebDriver, and Selenium Grid.

### 9) List some of the test types that are supported by Selenium.

Different types of testing's that we can achieve through Selenium are.

* Functional Testing
* Regression Testing
* Sanity Testing
* Smoke Testing
* Responsive Testing
* Cross Browser Testing
* UI testing (black box)
* Integration Testing

### 10) What is Selenium IDE?

Selenium IDE is implemented as Firefox extension which provides record and playback functionality on test scripts. It allows testers to export recorded scripts in many languages like HTML, Java, Ruby, RSpec, Python, C#, JUnit and TestNG.

Selenium IDE has limited scope, and the generated test scripts are not very robust, and portable.

### 11) What do you mean by Selenese?

Selenium commands, also known as "Selenese" are the set of commands used in Selenium that run your tests. For example, command - open (URL); launches the desired URL in the specified browser and it accept both relative and absolute URLs.

A sequence of Selenium commands (Selenese) together is known as a test script.

### 12) What are the different ways of locating a web element in Selenium?

In Selenium, web elements are identified and located with the help of Locators. Locators specify a target location which uniquely defines the web element in the context of a web application. Thus, to identify web elements accurately and precisely we have different types of locators in Selenium:

* ID
* ClassName
* Name
* TagName
* LinkText
* PartialLinkText
* Xpath
* CSS Selector
* DOM

### 13) How many types of WebDriver API's are available in Selenium?

The list of WebDriver API's which are used to automate browser include:

* AndroidDriver
* ChromeDriver
* EventFiringWebDriver
* FirefoxDriver
* HtmlUnitDriver
* InternetExplorerDriver
* iPhoneDriver
* iPhoneSimulatorDriver
* RemoteWebDriver

### 14) List out some of the Automation tools which could be integrated with Selenium to achieve continuous testing.

Selenium can be used to automate functional tests and can be integrated with automation test tools such as **Maven, Jenkins, &Docker** to achieve continuous testing. It can also be integrated with tools such as **TestNG, &JUnit** for managing test cases and generating reports.

### 15) What do you mean by the assertion in Selenium?

The assertion is used as a verification point. It verifies that the state of the application conforms to what is expected. The types of assertion are "assert", "verify" and "waitFor".

### 16) Explain the difference between assert and verify commands?

**Assert**: Assert command checks if the given condition is true or false. If the condition is true, the program control will execute the next phase of testing, and if the condition is false, execution will stop, and nothing will be executed.

**Verify**: Verify command also checks if the given condition is true or false. It doesn't halt program execution, i.e., any failure during verification would not stop the execution, and all the test phases would be executed.

### 17) What do you mean by XPath?

XPath is also defined as XML Path. It is a language used to query XML documents. It is an important approach to locate elements in Selenium. XPath consists of a path expression along with some conditions. Here, we can easily write XPath script/query to locate any element in the webpage. It is developed to allow the navigation of XML documents. The key factors that it considered while navigating are selecting individual elements, attributes, or some other part of an XML document for specific processing. It also produces reliable locators. Some other points about XPath are as follows.

* XPath is a language used for locating nodes in XML documents.
* XPath can be used as a substitute when you don't have a suitable id or name attribute for the element you want to locate.
* XPath provides locating strategies like:
  + XPath Absolute
  + XPath Attributes

### 18) Explain XPath Absolute and XPath attributes.

**XPath Absolute:**

* XPath Absolute enables users to mention the complete XPath location from the root HTML tag to the specific elements.
* Syntax: //html/body/tag1[index]/tag2[index]/.../tagN[index]
* Example: //html/body/div[2]/div/div[2]/div/div/div/fieldset/form/div[1]/input[1]

**XPath Attributes:**

* XPath Attributes is always recommended when you don't have a suitable id or name attribute for the element you want to locate.
* Syntax: //htmltag[@attribute1='value1' and @attribute2='value2']
* Example: //input[@id='passwd' and @placeholder='password']

### 19) What is the difference between "/" and "//" in XPath?

**Single Slash "/":** Single slash is used to create XPath with absolute path.

**Double Slash "//":** Double slash is used to create XPath with the relative path.

### 20) What are the different types of annotations which are used in Selenium?

JUnit annotations which can be used are:

* Test
* Before
* After
* Ignore
* BeforeClass
* AfterClass
* RunWith

### 21) What are the WebDriver supported Mobile Testing Drivers?

WebDriver supported "mobile testing drivers" are:

* AndroidDriver
* IphoneDriver
* OperaMobileDriver

### 22) What are the popular programming languages supported by Selenium WebDriver to write Test Cases?

Selenium WebDriver supports the below languages to write Test Cases.

* JAVA
* PHP
* Python
* C#
* Ruby
* Perl

### 23) What is the difference between type keys and type commands?

TypeKeys() will trigger JavaScript event in most of the cases whereas .type() won't.

### 24) What is the difference between "type" and "typeAndWait" command?

"type" command is used to type keyboard key values into the text box of software web application. It can also be used for selecting values of combo box whereas "typeAndWait" command is used when your typing is completed and software web page start reloading. This command will wait for software application page to reload. If there is no page reload event on typing, you have to use a simple "type" command.

### 25) What is the difference between findElement() and findElements()?

**findElement():** It is used to find the first element within the current page using the given "locating mechanism". It returns a single WebElement.

**findElements():** It uses the given "locating mechanism" to find all the elements within the current page. It returns a list of web elements.

### 26) What is the wait? How many types of waits in selenium?

Selenium Webdriver introduces the concept of waits for the AJAX-based application. There are two types of waits:

1. Implicit Wait
2. Explicit Wait

### 27) What is the main disadvantage of implicit wait?

The main disadvantage of implicit wait is that it slows down test performance.

Another disadvantage of implicit wait is:

Suppose, you set the waiting limit to be 10 seconds, and the elements appear in the DOM in 11 seconds, your tests will be failed because you told it to wait a maximum of 10 seconds.

### 28) What is Selenium Grid?

Selenium Grid facilitates you to distribute your tests on multiple machines and all of them at the same time. So, you can execute tests on Internet Explorer on Windows and Safari on Mac machine using the same text script. It reduces the time of test execution and provides quick feedback.

### Advance Level - Selenium Interview Questions

### 29) How can we launch different browsers in Selenium WebDriver?

We have to create an instance of a driver of that particular browser.

1. WebDriver driver =newFirefoxDriver();

Here, "WebDriver" is an interface, and we are creating a reference variable "driver" of type WebDriver, instantiated using "FireFoxDriver" class.

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

1. **public** **class** FirefoxBrowserLaunchDemo {
3. **public** **static** **void** main(String[] args) {
5. //Creating a driver object referencing WebDriver interface
6. WebDriver driver;
8. //Setting webdriver.gecko.driver property
9. System.setProperty("webdriver.gecko.driver", pathToGeckoDriver + "\\geckodriver.exe");
11. //Instantiating driver object and launching browser
12. driver = newFirefoxDriver();
14. //Using get() method to open a webpage
15. driver.get("http://javatpoint.com");
17. //Closing the browser
18. driver.quit();
20. }
22. }

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

1. **public** **class** ChromeBrowserLaunchDemo {
3. **public** **static** **void** main(String[] args) {
5. //Creating a driver object referencing WebDriver interface
6. WebDriver driver;
8. //Setting the webdriver.chrome.driver property to its executable's location
9. System.setProperty("webdriver.chrome.driver", "/lib/chromeDriver/chromedriver.exe");
11. //Instantiating driver object
12. driver = newChromeDriver();
14. //Using get() method to open a webpage
15. driver.get("http://javatpoint.com");
17. //Closing the browser
18. driver.quit();
20. }
22. }

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

1. **public** **class** IEBrowserLaunchDemo {
3. **public** **static** **void** main(String[] args) {
5. //Creating a driver object referencing WebDriver interface
6. WebDriver driver;
8. //Setting the webdriver.ie.driver property to its executable's location
9. System.setProperty("webdriver.ie.driver", "/lib/IEDriverServer/IEDriverServer.exe");
11. //Instantiating driver object
12. driver = newInternetExplorerDriver();
14. //Using get() method to open a webpage
15. driver.get("http://javatpoint.com");
17. //Closing the browser
18. driver.quit();
20. }
22. }

### 33) Write a code snippet to perform right-click an element in WebDriver.

We will use **Action class** to generate user event like right-click an element in WebDriver.

1. Actions action = newActions(driver);
2. WebElement element = driver.findElement(By.id("elementId"));
3. action.contextClick(element).perform();

### 34) Write a code snippet to perform mouse hover in WebDriver.

1. Actions action = newActions(driver);
2. WebElement element = driver.findElement(By.id("elementId"));
3. action.moveToElement(element).perform();

### 35) How do you perform drag and drop operation in WebDriver?

Code snippet to perform drag and drop operation:

1. //WebElement on which drag and drop operation needs to be performed
2. WebElementfromWebElement = driver.findElement(By Locator of fromWebElement);
4. //WebElement to which the above object is dropped
5. WebElementtoWebElement = driver.findElement(By Locator of toWebElement);
7. //Creating object of Actions class to build composite actions
8. Actions builder = newActions(driver);
10. //Building a drag and drop action
11. Action dragAndDrop = builder.clickAndHold(fromWebElement)
12. .moveToElement(toWebElement)
13. .release(toWebElement)
14. .build();
16. //Performing the drag and drop action
17. dragAndDrop.perform();

### 36) What are the different methods to refresh a web page in WebDriver?

There are multiple ways of refreshing a page in Webdriver.

1. Using driver.navigate command -

1. driver.navigate().refresh();

2. Using driver.getCurrentUrl() with driver.get() command -

1. driver.get(driver.getCurrentUrl());

3. Using driver.getCurrentUrl() with driver.navigate() command -

1. driver.navigate().to(driver.getCurrentUrl());

4. Pressing an F5 key on any textbox using the sendKeys command -

1. driver.findElement(By textboxLocator).sendKeys(Keys.F5);

5. Passing ascii value of the F5 key, i.e., "\uE035" using the sendKeys command -

1. driver.findElement(By textboxLocator).sendKeys("\uE035");

### 37) Write a code snippet to navigate back and forward in browser history?

Navigate back in browser history:

1. driver.navigate().back();

Navigate forward in browser history:

1. driver.navigate().forward();

### 38) How to invoke an application in WebDriver?

1. driver.get("url"); or
2. driver.navigate().to("url");

### Misc. Questions - Selenium Interview Question.

### 39) What are the benefits of Automation Testing?

Benefits of Automation testing are as follows.

* It allows execution of repeated test cases
* It enables parallel execution
* Automation Testing encourages unattended execution
* It improves accuracy. Thus, it reduces human-generated errors
* It saves time and money.

### 40) How can we get a text of a web element?

**Get command** is used to get the inner text of the specified web element. The **get** command doesn't require any parameter, but it returns a string type value. It is also one of the widely used commands for verification of messages, labels, and errors,etc.,from web pages.

**Syntax**

1. String Text = driver.findElement(By.id("Text")).getText();

### 41) How to select value in a dropdown?

We use the WebDriver's Select class to select the value in the dropdown.

**Syntax:**

**selectByValue:**

1. Select selectByValue = **new** Select(driver.findElement(By.id("SelectID\_One")));
2. selectByValue.selectByValue("greenvalue");

**selectByVisibleText:**

1. Select selectByVisibleText = **new** Select (driver.findElement(By.id("SelectID\_Two")));
2. selectByVisibleText.selectByVisibleText("Lime");
3. Select selectByIndex = **new** Select(driver.findElement(By.id("SelectID\_Three")));
4. selectByIndex.selectByIndex(2);

### 42) What are the different types of navigation commands?

The navigation commands are as follows.

**navigate().back()**

The above command needs no parameters and takes back the user to the previous webpage.

**Example**

1. driver.navigate().back();

**navigate().forward()**

The above command allows the user to navigate to the next web page with reference to the browser's history.

**Example**

1. driver.navigate().forward();

**navigate().refresh()**

The **navigate().refresh()** command allows the user to refresh the current web page by reloading all the web elements.

**Example**

1. driver.navigate().refresh();

**navigate().to()**

The **navigate().to()** command allows the user to launch a new web browser window and navigate to the specified URL.

**Example**

1. driver.navigate().to("https://google.com");

### 43) How to deal with frame in WebDriver?

An inline frame abbreviates as an iframe. It is used to insert another document within the current document. These document can be HTML document or simply web page and nested web page.

**Select iframe by id**

1. driver.switchTo().frame("ID of the frame");

**Locating iframe using tagName**

1. driver.switchTo().frame(driver.findElements(By.tagName("iframe").get(0));

**Locating iframe using index**

**frame(index)**

1. driver.switchTo().frame(0);

**frame(Name of Frame)**

1. driver.switchTo().frame("name of the frame");

**frame(WebElement element)**

**Select Parent Window**

1. driver.switchTo().defaultContent();

### 44) Is there an HtmlUnitDriver for .NET?

To use HtmlUnit first use the RemoteWebDriver and pass it in the desired capabilities.

1. IWebDriver driver
2. = **new** RemoteWebDriver(DesiredCapabilities.HtmlUnit())

For the Firefox implementation to run, use

1. IWebDriver driver
2. = **new** RemoteWebDriver(DesiredCapabilities.HtmlUnitWithJavaScript())

### 45) How can you redirect browsing from a browser through some proxy?

Selenium facilitates with a **PROXY class** to redirect browsing from a proxy. Look at the example below.

**Example**

1. String PROXY = "199.201.125.147:8080";
2. org.openqa.selenium.Proxy proxy = **new**.org.openqa.selenium.Proxy();
3. proxy.setHTTPProxy(Proxy)
4. .setFtpProxy(Proxy)
5. .setSslProxy(Proxy)
6. DesiredCapabilities cap = **new** DesiredCapabilities();
7. cap.setCapability(CapabilityType.PROXY, proxy);
8. WebDriver driver = **new** FirefoxDriver(cap);

### 46) What is POM (Page Object Model)? What are its advantages?

Page Object Model is a design pattern for creating an Object directory for web UI elements. Each web page is required to have its page class. The page class is responsible for finding the WebElements in web pages and then perform operations on WebElements.

The benefits of using POM are as follows.

* It facilitates with separate operations and flows in the UI from Verification - improves code readability
* Multiple tests can use the same Object Repository because the Object Repository is independent of Test Cases.
* Reusability of code

### 47) How to capture screenshot in WebDriver?

Below is the program to capture screenshot in WebDriver.

1. **import** org.junit.After;
2. **import** org.junit.Before;
3. **import** org.junit.Test;
4. **import** java.io.File;
5. **import** java.io.IOException;
6. **import** org.apache.commons.io.FileUtils;
7. **import** org.openqa.selenium.OutputType;
8. **import** org.openqa.selenium.TakesScreenshot;
9. **import** org.openqa.selenium.WebDriver;
10. **import** org.openqa.selenium.firefox.FirefoxDriver;
12. **public** **class** TakeScreenshot {
13. WebDriver drv;
14. @Before
15. **public** **void** setUp() **throws** Exception {
16. driver = **new** FirefoxDriver();
17. drv.get("https://google.com");
18. }
19. @After
20. **public** **void** tearDown() **throws** Exception {
21. drv.quit();
22. }
24. @Test
25. **public** **void** test() **throws** IOException {
26. //capture the screenshot
27. File scrFile = ((TakeScreenshot)drv).getScreenshotAs(OutputType.FILE);
28. // paste the screenshot in the desired location
29. FileUtils.copyFile(scrFile, **new** File("C:\\Screenshot\\Scr.jpg"))
30. }
31. }

### 48) How to type text in a textbox using Selenium?

The sendKeys("String to be entered") is used to enter the string in a textbox.

**Syntax**

1. WebElement username = drv.findElement(By.id("Email"));
2. // entering username
3. username.sendKeys("sth");

### 49) How can you find if an element is displayed on the screen?

WebDriver allows user to check the visibility of the web elements. These web elements can be buttons, radio buttons, drop, checkboxes, boxes, labels etc. which are used with the following methods.

* isDisplayed()
* isSelected()
* isEnabled()

**Syntax:**

1. isDisplayed():
2. **boolean** buttonPresence = driver.findElement(By.id("gbqfba")).isDisplayed();
3. isSelected():
4. **boolean** buttonSelected = driver.findElement(By.id("gbqfba")).isSelected();
5. isEnabled():
6. **boolean** searchIconEnabled = driver.findElement(By.id("gbqfb")).isEnabled();

### 50) How to click on a hyper link using linkText?

1. driver.findElement(By.linkText("Google")).click();

The above command search the element using a link text, then click on that element and thus the user will be re-directed to the corresponding page.

The following command can access the link mentioned earlier.

1. driver.findElement(By.partialLinkText("Goo")).click();

The above-given command searches the element based on the substring of the link provided in the parenthesis. And after that **partialLinkText()** finds the web element with the specified substring and then clicks on it.

# 10.Edureka 50Q: <https://www.edureka.co/blog/interview-questions/selenium-interview-questions-answers/>

## ****A. Basic Level – Selenium Interview Questions****

### ****1. What are the advantages and disadvantages of Selenium over other testing tools like QTP and TestComplete?****

The differences are listed below.

***Selenium vs HP QTP vs TestComplete***

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **Selenium** | **HP QTP** | **TestComplete** |
| **License** | Open Source | Required | Required |
| **Cost** | Free | High | High |
| **Customer support** | Yes; Open source community | Yes | Yes |
| **Release Cycles/ Development Sprints** | Smaller release cycles with immediate feedback | Smaller release cycles | Agility only |
| **Coding skills** | Very High | Low | High |
| **Environment support** | Windows, Linux, Mac | Only Windows | Windows only (7, Vista, Server 2008 or later OS) |
| **Language support** | Language support | VB Script | VB Script, JS Script, Delphi Script, C++ & C# |

### ****2. What are the significant changes in upgrades in various Selenium versions?****

**Selenium v1** included only three suite of tools: Selenium IDE, Selenium RC and Selenium Grid. Note that there was no WebDriver in Selenium v1. Selenium WebDriver was introduced in **Selenium v2**. With the onset of WebDriver, Selenium RC got deprecated and is not in use since. Older versions of RC is available in the market though, but support for RC is not available. Currently, **Selenium v3** is in use, and it comprises of IDE, WebDriver and Grid. **Selenium 4** is actually the latest version.

IDE is used for recording and playback of tests, WebDriver is used for testing dynamic web applications via a programming interface and Grid is used for deploying tests in remote host machines.

### ****3. Explain the different exceptions in Selenium WebDriver.****

Exceptions in Selenium are similar to exceptions in other programming languages. The most common exceptions in Selenium are:

* **TimeoutException:** This exception is thrown when a command performing an operation does not complete in the stipulated time
* **NoSuchElementException:** This exception is thrown when an element with given attributes is not found on the web page
* **ElementNotVisibleException:** This exception is thrown when the element is present in DOM (Document Object Model), but not visible on the web page
* **StaleElementException:** This exception is thrown when the element is either deleted or no longer attached to the DOM

### ****4. What is exception test in Selenium?****

An exception test is an exception that you expect will be thrown inside a test class. If you have written a test case in such way that it should throw an exception, then you can use the **@Test** annotation and specify which exception you will be expecting by mentioning it in the parameters. Take a look at the example below: **@Test(expectedException = NoSuchElementException.class)**

Do note the syntax, where the exception is suffixed with .class

### ****5. Why and how will you use an Excel Sheet in your project?****

The reason we use Excel sheets is because it can be used as data source for tests. An excel sheet can also be used to store the data set while performing DataDriven Testing. These are the two main reasons for using Excel sheets.

When you use the excel sheet as **data source**, you can store the following:

* **Application URL for all environments**: You can specify the URL of the environment in which you want to do the testing like: development environment or testing environment or QA environment or staging environment or production/ pre-production environment.
* **User name and password credentials of different environments:**You can store the access credentials of the different applications/ environments in the excel sheet. You can store them in encoded format and whenever you want to use them, you can decode them instead of leaving it plain and unprotected.
* **Test cases to be executed**: You can list down the entire set of test cases in a column and in the next column, you can specify either Yes or No which indicates if you want that particular test case to be executed or ignored.

When you use the excel sheet for **DataDriven Test**, you can store the data for different iterations to be performed in the tests. For example while testing a web page, the different sets of input data that needs to be passed to the test box can be stored in the excel sheet.

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

Selenium provides a PROXY class to redirect browsing from a proxy. Look at the example below:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | String PROXY = &ldquo;199.201.125.147:8080&rdquo;;    org.openqa.selenium.Proxy proxy = new.org.openqa.selenium.Proxy();  proxy.setHTTPProxy(Proxy)   .setFtpProxy(Proxy)   .setSslProxy(Proxy)  DesiredCapabilities cap = new DesiredCapabilities();  cap.setCapability(CapabilityType.PROXY, proxy);  WebDriver driver = new FirefoxDriver(cap); |

### ****7. What is POM (Page Object Model)? What are its advantages?****

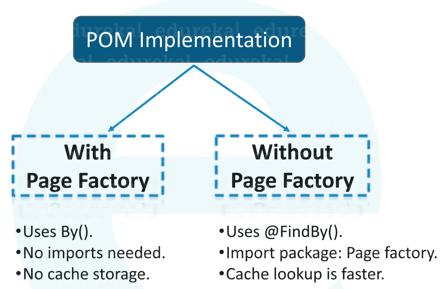
[*Page Object Model*](https://www.edureka.co/blog/page-object-model-in-selenium/) is a design pattern for creating an Object Repository for web UI elements. Each web page in the application is required to have it’s own corresponding page class. The page class is thus responsible for finding the WebElements in that page and then perform operations on those WebElements.

The advantages of using POM are:

* Allows us to separate operations and flows in the UI from Verification – improves code readability
* Since the Object Repository is independent of Test Cases, multiple tests can use the same Object Repository
* Reusability of code

### ****8. What is Page Factory?****

Page Factory gives an optimized way to implement Page Object Model. When we say it is optimized, it refers to the fact that the memory utilization is very good and also the implementation is done in an object oriented manner.



Page Factory is used to initialize the elements of the Page Object or instantiate the Page Objects itself. Annotations for elements can also be created (and recommended) as the describing properties may not always be descriptive enough to differentiate one object from the other.

The concept of separating the Page Object Repository and Test Methods is followed here also. Instead of having to use ‘FindElements’, we use annotations like: **@FindBy** to find WebElement, and **initElements** method to initialize web elements from the Page Factory class.

**@FindBy** can accept **tagName**, **partialLinkText**, **name**, **linkText**, **id**, **css**, **className**& **xpath**as attributes.

### ****9. What are the different types of WAIT statements in Selenium WebDriver? *Or the question can be framed like this:* How do you achieve synchronization in WebDriver?****

There are basically two types of wait statements: **Implicit Wait** and **Explicit Wait**.

Implicit wait instructs the WebDriver to wait for some time by polling the DOM. Once you have declared implicit wait, it will be available for the entire life of the WebDriver instance. By default, the value will be 0. If you set a longer default, then the behavior will poll the DOM on a periodic basis depending on the browser/ driver implementation.

Explicit wait instructs the execution to wait for some time until some condition is achieved. Some of those conditions to be attained are:

* elementToBeClickable
* elementToBeSelected
* presenceOfElementLocated

### ****10. Write a code to wait for a particular element to be visible on a page. Write a code to wait for an alert to appear.****

We can write a code such that we specify the [*XPath*](https://www.edureka.co/blog/xpath-in-selenium/) of the web element that needs to be visible on the page and then ask the WebDriver to wait for a specified time. Look at the sample piece of code below:

|  |  |
| --- | --- |
| 1  2 | WebDriverWait wait=new WebDriverWait(driver, 20);  Element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( &ldquo;<xpath&rdquo;))); |

Similarly, we can write another piece of code asking the WebDriver to wait until an error appears like this:

|  |  |
| --- | --- |
| 1  2 | WebDriverWait wait=new WebDriverWait(driver, 20);  Element = wait.until(ExpectedConditions.alertIsPresent()); |

### ****11. What is the use of JavaScriptExecutor?****

**JavaScriptExecutor** is an interface which provides a mechanism to execute Javascript through the Selenium WebDriver. It provides “**executescript**” and “**executeAsyncScript**” methods, to run JavaScript in the context of the currently selected frame or window. An example of that is:

|  |  |
| --- | --- |
| 1  2 | JavascriptExecutor js = (JavascriptExecutor) driver;  js.executeScript(Script,Arguments); |

### ****12. How to scroll down a page using JavaScript in Selenium?****

We can scroll down a page by using window.scrollBy() function. Example:

|  |  |
| --- | --- |
| 1 | ((JavascriptExecutor) driver).executeScript("window.scrollBy(0,500)"); |

### ****13. How to scroll down to a particular element?****

To scroll down to a particular element on a web page, we can use the function **scrollIntoView()**. Example:

|  |  |
| --- | --- |
| 1 | ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView();", element); |

### ****14. How to handle keyboard and mouse actions using Selenium?****

We can handle special keyboard and mouse events by using **Advanced User Interactions API**. The Advanced User Interactions API contains the Actions and the Action Classes that are needed for executing these events. Most commonly used keyboard and mouse events provided by the Actions class are in the table below:

[[](https://www.edureka.co/selenium-certification-training)](https://www.edureka.co/selenium-certification-training" \t "_blank)

### [Selenium Certification Training Course](https://www.edureka.co/selenium-certification-training" \t "_blank)

* *[Instructor-led Sessions](https://www.edureka.co/selenium-certification-training" \t "_blank)*
* *[Assessments](https://www.edureka.co/selenium-certification-training" \t "_blank)*
* *[Assignments](https://www.edureka.co/selenium-certification-training" \t "_blank)*
* *[Lifetime Access](https://www.edureka.co/selenium-certification-training" \t "_blank)*

[Explore Curriculum](https://www.edureka.co/selenium-certification-training" \t "_blank)

***Selenium functions and their explanation***

|  |  |
| --- | --- |
| **Method** | **Description** |
| **clickAndHold()** | Clicks (without releasing) the current mouse location. |
| **dragAndDrop()** | Performs click-and-hold at the location of the source element, moves. |
| **source, target()** | Moves to the location of the target element, then releases the mouse. |

### ****15. What are different types of frameworks?****

The different types of [*frameworks*](https://www.edureka.co/blog/selenium-framework-data-keyword-hybrid-frameworks) are:

* **Data Driven Framework**:-  
  When the entire test data is generated from some external files like Excel, CSV, XML or some database table, then it is called Data Driven framework.
* **Keyword Driven Framework**:-  
  When only the instructions and operations are written in a different file like an Excel worksheet, it is called Keyword Driven framework.
* **Hybrid Framework**:-  
  A combination of both the Data Driven framework and the Keyword Driven framework is called Hybrid framework.

### ****16. Which files can be used as data source for different frameworks?****

Some of the file types of the dataset can be: excel, xml, text, csv, etc.

### ****17. How can you fetch an attribute from an element? How to retrieve typed text from a textbox?****

We can fetch the attribute of an element by using the **getAttribute()** method. Sample code:

|  |  |
| --- | --- |
| 1  2 | WebElement eLogin = driver.findElement(By.name(&ldquo;Login&rdquo;);  String LoginClassName = eLogin.getAttribute("classname"); |

Here, I am finding the web page’s login button named ‘Login’. Once that element is found, getAttribute() can be used to retrieve any attribute value of that element and it can be stored it in string format. In my example, I have retrieved ‘classname’ attribute and stored it in LoginClassName.

Similarly, to retrieve some text from any textbox, we can use getText() method. In the below piece of code I have retrieved the text typed in the ‘Login’ element.

|  |  |
| --- | --- |
| 1  2 | WebElement eLogin = driver.findElement(By.name(&ldquo;Login&rdquo;);  String LoginText = Login.getText (); |

In the below Selenium WebDriver tutorial, there is a detailed demonstration of locating elements on the web page using different element locator techniques and the basic methods/ functions that can be applied on those elements.

## Selenium WebDriver Tutorial | Selenium Tutorial For Beginner | Selenium WebDriver Training | Edureka



This Selenium WebDriver tutorial talks about the drawbacks of Selenium RC and what was the need for Selenium WebDriver.

### ****18. What is Selenese?****

Selenese is the set of selenium commands which are used to test your web application.

You can even make use of:

* **Actions**: Used for performing operations
* **Assertions**: Used as checkpoints
* **Accessors:** Used for storing a value in a particular variable

### ****19. What is the difference between Page Object Model (POM) and Page Factory?****

## Pom vs Page factory-Edureka

### ****20. Can Selenium handle window pop-ups?****

Selenium does not support handling pop-ups. [*Alert*](https://www.edureka.co/blog/alerts-popups-in-selenium) is used to display a warning message. It is a pop-up window that comes up on the screen.

A few methods using which this can be achieved:

* **Void dismiss():** This method is called when the ‘Cancel’ button is clicked in the alert box.
* **Void accept():** This method is called when you click on the ‘OK’ button of the alert.
* **String getText():** This method is called to capture the alert message.
* **Void sendKeys(String stringToSed)**: This is called when you want to send some data to alert box.

### ****21. What is a Robot class?****

This [*Robot class*](https://www.edureka.co/blog/robot-class-selenium-webdriver/) provides control over the mouse and keyboard devices.

The methods include:

* **KeyPress():**This method is called when you want to press any key.
* **KeyRelease():**This method is used to release the pressed key on the keyboard.
* **MouseMove():**This method is called when you want to move the mouse pointer in the X and Y co-ordinates.
* **MousePress():**This is used to press the left button of the mouse.
* **MouseMove():**This method helps in releasing the pressed button of the mouse.

### ****22. How to handle multiple windows in Selenium?****

A [*window handle*](https://www.edureka.co/blog/handle-multiple-windows-in-selenium/) is a unique identifier that holds the address of all the windows. This is basically a pointer to a window, which returns the string value.

* **get.windowhandle()**: helps in getting the window handle of the current window.
* **get.windowhandles():** helps in getting the handles of all the windows opened.
* **set:** helps to set the window handles which is in the form of a string.
* **switch to:** helps in switching between the windows.
* **action**: helps to perform certain [*actions*](https://www.edureka.co/blog/keyboard-mouse-events-actions-class) on the windows.

### ****23. What are Listeners in Selenium?****

It is defined as an interface that modifies the behavior of the system. [*Listeners*](https://www.edureka.co/blog/listeners-in-selenium/) allow customization of reports and logs.

Listeners mainly comprise of two types, namely

1. WebDriver listeners
2. TestNG listeners

### ****24. What are Assert and Verify commands?****

* **Assert:** An assertion is used to compare the actual result of an application with the expected result.
* **Verify**: There won’t be any halt in the test execution even though the verify condition is true or false.

### ****25. Can you navigate back and forth the webpage in Selenium?****

Yes. You can navigate in the browser. A few methods using which you can achieve it are:

* driver.navigate.forward
* driver.manage.back
* driver.manage.navigate
* driver.navigate.to(“url”)

## ****B. Advanced Level – Selenium Interview Question****

From here on, we’ll be looking at the most important advanced level interview questions for Selenium testers.

### ****26. How to send ALT/SHIFT/CONTROL key in Selenium WebDriver?****

When we generally use ALT/SHIFT/CONTROL keys, we hold onto those keys and click other buttons to achieve the special functionality. So it is not enough just to specify **keys.ALT** or **keys.SHIFT** or **keys.CONTROL** functions.

For the purpose of holding onto these keys while subsequent keys are pressed, we need to define two more methods: **keyDown(modifier\_key)** and **keyUp(modifier\_key)**

Parameters: **Modifier\_key (keys.ALT or Keys.SHIFT or Keys.CONTROL)**Purpose: Performs a modifier key press and does not release the modifier key. Subsequent interactions may assume it’s kept pressed.

Parameters: **Modifier\_key (keys.ALT or Keys.SHIFT or Keys.CONTROL)**  
Purpose: Performs a key release.  
Hence with a combination of these two methods, we can capture the special function of a particular key.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | public static void main(String[] args)  {  String baseUrl = &ldquo;https://www.facebook.com&rdquo;;  WebDriver driver = new FirefoxDriver();    driver.get("baseUrl");  WebElement txtUserName = driver.findElement(By.id(&ldquo;Email&rdquo;);    Actions builder = new Actions(driver);  Action seriesOfActions = builder   .moveToElement(txtUerName)   .click()   .keyDown(txtUserName, Keys.SHIFT)   .sendKeys(txtUserName, &ldquo;hello&rdquo;)   .keyUp(txtUserName, Keys.SHIFT)   .doubleClick(txtUserName);   .contextClick();   .build();  seriesOfActions.perform();  } |

### ****27. How to take screenshots in Selenium WebDriver?****

You can take a [*screenshot*](https://www.edureka.co/blog/how-to-take-a-screenshot-in-selenium-webdriver/) by using the **TakeScreenshot** function. By using **getScreenshotAs()** method you can save that screenshot. Example:

|  |  |
| --- | --- |
| 1 | File scrFile = ((TakeScreenshot)driver).getScreenshotAs(outputType.FILE); |

### ****28. How to set the size of browser window using Selenium?****

To maximize the size of browser window, you can use the following piece of code:  
driver.manage().window().maximize(); – To maximize the window

To resize the current window to a particular dimension, you can use the **setSize()** method. Check out the below piece of code:

|  |  |
| --- | --- |
| 1  2  3 | System.out.println(driver.manage().window().getSize());  Dimension d = new Dimension(420,600);  driver.manage().window().setSize(d); |

To set the window to a particular size, use **window.resizeTo()** method. Check the below piece of code:

|  |  |
| --- | --- |
| 1 | ((JavascriptExecutor)driver).executeScript("window.resizeTo(1024, 768);"); |

To witness a demonstration on setting custom sizes for the browser window and finding various elements on the web page, see the video below.

### ****29. How to handle a dropdown in Selenium WebDriver? How to select a value from dropdown?****

Questions on dropdown and selecting a value from that dropdown are very common Selenium interview questions because of the technicality involved in writing the code.

The most important detail you should know is that to work with a [*dropdown in Selenium*](https://www.edureka.co/blog/selenium-select-class/), we must always make use of this html tag: **‘select’**. Without using ‘select’, we cannot handle dropdowns. Look at the snippet below in which I have written a code for a creating a dropdown with three options.

|  |  |
| --- | --- |
| 1  2  3  4  5 | <select id="mySelect">  <option value="option1">Cars</option>  <option value="option2">Bikes</option>  <option value="option3">Trains</option>  </select> |

In this code we use **‘select’**tag to define a dropdown element and the **id**of the dropdown element is ‘myselect’. We have 3 options in the dropdown: Cars, Bikes and Trains. Each of these options, have a ‘value’ attribute also assigned to them. First option from dropdown has value assigned as ‘option1’, second option has value = ‘option2’ and similarly third option has value assigned as ‘option3’.

If you are clear with the concept so far, then you can proceed to the next aspect of choosing a value from the dropdown. This is a 2 step process:

1. Identify the ‘select’ html element (Because dropdowns must have the ‘select’ tag)
2. Select an option from that dropdown element

To identify the ‘select’ html element from the web page, we need to use findElement() method. Look at the below piece of code:

|  |  |
| --- | --- |
| 1  2 | WebElement mySelectElement = driver.findElement(By.id("mySelect"));  Select dropdown = new Select(mySelectElement); |

Now to select an option from that dropdown, we can do it in either of the three ways:

1. dropdown.selectByVisibleText(“Bikes”); → Selecting an option by the text that is visible
2. dropdown.selectByIndex(“1”); → Selecting, by choosing the Index number of that option
3. dropdown.selectByValue(“option2”); → Selecting, by choosing the value of that option

Note that from the above example, in all the three cases, “Bikes” will be chosen from the dropdown. In the first case, we are choosing by visible text on the web page. When it comes to selection by index, 1 represents “Bikes” because indexing values start from 0 and then get incremented to 1 and 2. Finally in case of selection by value attribute, ‘option2’ refers to “Bikes”. So, these are the different ways to choose a value from a dropdown.

### Software Testing Training

### [SELENIUM CERTIFICATION TRAINING COURSE](https://www.edureka.co/selenium-certification-training" \o "Selenium Certification Training Course" \t "_blank)

### [Selenium Certification Training Course](https://www.edureka.co/selenium-certification-training" \o "Selenium Certification Training Course" \t "_blank)

*[Reviews](https://www.edureka.co/selenium-certification-training" \o "Selenium Certification Training Course" \t "_blank)*

**[5](https://www.edureka.co/selenium-certification-training" \o "Selenium Certification Training Course" \t "_blank)**[(37070)](https://www.edureka.co/selenium-certification-training" \o "Selenium Certification Training Course" \t "_blank)

### [MANUAL TESTING CERTIFICATION TRAINING COURSE ONLINE](https://www.edureka.co/software-testing" \o "Manual Testing Certification Training Course Online" \t "_blank)

### [Manual Testing Certification Training Course Online](https://www.edureka.co/software-testing" \o "Manual Testing Certification Training Course Online" \t "_blank)

*[Reviews](https://www.edureka.co/software-testing" \o "Manual Testing Certification Training Course Online" \t "_blank)*

**[5](https://www.edureka.co/software-testing" \o "Manual Testing Certification Training Course Online" \t "_blank)**[(6069)](https://www.edureka.co/software-testing" \o "Manual Testing Certification Training Course Online" \t "_blank)

### [SOFTWARE TESTING FUNDAMENTALS COURSE](https://www.edureka.co/software-testing-fundamentals-training" \o "Software Testing Fundamentals Course" \t "_blank)

### [Software Testing Fundamentals Course](https://www.edureka.co/software-testing-fundamentals-training" \o "Software Testing Fundamentals Course" \t "_blank)

*[Reviews](https://www.edureka.co/software-testing-fundamentals-training" \o "Software Testing Fundamentals Course" \t "_blank)*

**[5](https://www.edureka.co/software-testing-fundamentals-training" \o "Software Testing Fundamentals Course" \t "_blank)**[(1779)](https://www.edureka.co/software-testing-fundamentals-training" \o "Software Testing Fundamentals Course" \t "_blank)

### [PERFORMANCE TESTING USING JMETER](https://www.edureka.co/jmeter-training-performance-testing" \o "Performance Testing Using JMeter" \t "_blank)

### [Performance Testing Using JMeter](https://www.edureka.co/jmeter-training-performance-testing" \o "Performance Testing Using JMeter" \t "_blank)

*[Reviews](https://www.edureka.co/jmeter-training-performance-testing" \o "Performance Testing Using JMeter" \t "_blank)*

**[5](https://www.edureka.co/jmeter-training-performance-testing" \o "Performance Testing Using JMeter" \t "_blank)**[(2476)](https://www.edureka.co/jmeter-training-performance-testing" \o "Performance Testing Using JMeter" \t "_blank)

### [MOBILE APP TESTING USING APPIUM](https://www.edureka.co/appium-training-mobile-automation-testing" \o "Mobile App Testing Using Appium" \t "_blank)

### [Mobile App Testing Using Appium](https://www.edureka.co/appium-training-mobile-automation-testing" \o "Mobile App Testing Using Appium" \t "_blank)

*[Reviews](https://www.edureka.co/appium-training-mobile-automation-testing" \o "Mobile App Testing Using Appium" \t "_blank)*

**[5](https://www.edureka.co/appium-training-mobile-automation-testing" \o "Mobile App Testing Using Appium" \t "_blank)**[(2653)](https://www.edureka.co/appium-training-mobile-automation-testing" \o "Mobile App Testing Using Appium" \t "_blank)

### [CONTINUOUS TESTING IN DEVOPS](https://www.edureka.co/continuous-testing-devops-training" \o "Continuous Testing in DevOps" \t "_blank)

### [Continuous Testing in DevOps](https://www.edureka.co/continuous-testing-devops-training" \o "Continuous Testing in DevOps" \t "_blank)

*[Reviews](https://www.edureka.co/continuous-testing-devops-training" \o "Continuous Testing in DevOps" \t "_blank)*

**[5](https://www.edureka.co/continuous-testing-devops-training" \o "Continuous Testing in DevOps" \t "_blank)**[(1782)](https://www.edureka.co/continuous-testing-devops-training" \o "Continuous Testing in DevOps" \t "_blank)

### [AUTOMATION TESTING USING TESTCOMPLETE 11.0](https://www.edureka.co/automation-testing-testcomplete" \o "Automation Testing using TestComplete 11.0" \t "_blank)

### [Automation Testing using TestComplete 11.0](https://www.edureka.co/automation-testing-testcomplete" \o "Automation Testing using TestComplete 11.0" \t "_blank)

*[Reviews](https://www.edureka.co/automation-testing-testcomplete" \o "Automation Testing using TestComplete 11.0" \t "_blank)*

**[5](https://www.edureka.co/automation-testing-testcomplete" \o "Automation Testing using TestComplete 11.0" \t "_blank)**[(2264)](https://www.edureka.co/automation-testing-testcomplete" \o "Automation Testing using TestComplete 11.0" \t "_blank)

Next

## Selenium Tutorial For Beginners | What Is Selenium? | Selenium Automation Testing Tutorial | Edureka



This Edureka Selenium tutorial video will give you an introduction to software testing.

### ****30. How to switch to a new window (new tab) which opens up after you click on a link?****

If you click on a link in a web page, then for changing the WebDriver’s focus/ reference to the new window we need to use the **switchTo()** command. Look at the below example to switch to a new window:  
**driver.switchTo().window();**

Here, ‘windowName’ is the name of the window you want to switch your reference to.

In case you do not know the name of the window, then you can use the **driver.getWindowHandle()** command to get the name of all the windows that were initiated by the WebDriver. Note that it will not return the window names of browser windows which are not initiated by your WebDriver.

Once you have the name of the window, then you can use an enhanced for loop to switch to that window. Look at the piece of code below.

|  |  |
| --- | --- |
| 1  2  3  4  5 | String handle= driver.getWindowHandle();  for (String handle : driver.getWindowHandles())  {  driver.switchTo().window(handle);  } |

### ****31. How do you upload a file using Selenium WebDriver?****

To upload a file we can simply use the command **element.send\_keys(file path).**But there is a prerequisite before we upload the file. We have to use the html tag: **‘input’**and attribute type should be **‘file’**. Take a look at the below example where we are identifying the web element first and then uploading the file.

|  |  |
| --- | --- |
| 1  2  3 | <input type="file" name="uploaded\_file" size="50" class="pole\_plik">  element = driver.find\_element\_by\_id(&rdquo;uploaded\_file")  element.send\_keys("C:myfile.txt") |

### ****32. Can we enter text without using sendKeys()?****

Yes. We can enter/ send text without using **sendKeys()** method. We can do it using JavaScriptExecutor.

How do we do it?  
Using DOM method of, identification of an element, we can go to that particular document and then get the element by its ID (here login) and then send the text by value. Look at the sample code below:

|  |  |
| --- | --- |
| 1  2 | JavascriptExecutor jse = (JavascriptExecutor) driver;  jse.executeScript("document.getElementById(&lsquo;Login').value=Test text without sendkeys"); |

### ****33. Explain how you will login into any site if it is showing any authentication popup for username and password?****

Since there will be popup for logging in, we need to use the explicit command and verify if the alert is actually present. Only if the alert is present, we need to pass the username and password credentials. The sample code for using the explicit wait command and verifying the alert is below:

|  |  |
| --- | --- |
| 1  2  3 | WebDriverWait wait = new WebDriverWait(driver, 10);  Alert alert = wait.until(ExpectedConditions.alertIsPresent());  alert.authenticateUsing(new UserAndPassword(\*\*username\*\*, \*\*password\*\*)); |

### ****34. Explain how can you find broken links in a page using Selenium WebDriver?****

This is a tricky question which the interviewer will present to you. He can provide a situation where in there are 20 links in a web page, and we have to verify which of those 20 links are working and how many are not working (broken).

Since you need to verify the working of every link, the workaround is that, you need to send HTTP requests to all of the links on the web page and analyze the response. Whenever you 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.

But how will you do that?  
First, we have to use the anchor tags <a> to determine the different hyperlinks on the web page. For each <a> tag, we can use the attribute ‘href’ value to obtain the hyperlinks and then analyze the response received for each hyperlink when used in **driver.get()** method.

### ****35. Which technique should you consider using throughout the script “if there is neither frame id nor frame name”?****

If neither frame name nor frame id is available, then we can use **frame by index**.

Let’s say, that there are 3 frames in a web page and if none of them have frame name and frame id, then we can still select those frames by using frame (zero-based) index attribute. Each frame will have an index number. The first frame would be at index “0”, the second at index “1” and the third at index “2”. Once the frame has been selected, all subsequent calls on the WebDriver interface will be made to that frame.

|  |  |
| --- | --- |
| 1 | driver.switchTo().frame(int arg0); |

## ****C. TestNG Framework For Selenium – Selenium Interview Questions****

From here on, you’ll read all the TestNG and Selenium Webdriver interview questions for experienced professionals.

### ****36. What is the significance of testng.xml?****

I’m pretty sure you all know the importance of [*TestNG*](https://www.edureka.co/blog/testng-annotations-in-selenium/). Since Selenium does not support report generation and test case management, we use TestNG framework with Selenium. TestNG is much more advanced than [*JUnit*](https://www.edureka.co/blog/junit-tutorial/), and it makes implementing annotations easy. That is the reason TestNG framewrok is used with Selenium WebDriver.

But have you wondered where to define the test suites and grouping of test classes in TestNG?

It is by taking instructions from the testng.xml file. We cannot define a test suite in testing source code, instead it is represented in an XML file, because suite is the feature of execution. The test suite, that I am talking about is basically a collection of test cases.

So for executing the test cases in a suite, i.e a group of test cases, you have to create a testng.xml file which contains the name of all the classes and methods that you want to execute as a part of that execution flow.

Other advantages of using testng.xml file are:

* It allows execution of multiple test cases from multiple classes
* It allows parallel execution
* It allows execution of test cases in groups, where a single test can belong to multiple groups

### ****37. What is parameterization in TestNG? How to pass parameters using testng.xml?****

[*Parameterization*](https://www.edureka.co/blog/dataprovider-in-testng/) is the technique of defining values in testng.xml file and sending them as parameters to the test class. This technique is especially useful when we need to pass multiple login credentials of various test environments. Take a look at the code below, in which “myName” is annotated as a parameter.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | public class ParameterizedTest1{   @Test   @Parameters("myName")   public void parameterTest(String myName) {   System.out.println("Parameterized value is : " + myName);   }  } |

To pass parameters using testng.xml file, we need to use ‘parameters’ tag. Look at the below code for example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE suite SYSTEM "<http://testng.org/testng-1.0.dtd>" >   <suite name=&rdquo;CustomSuite">    <test name=&rdquo;CustomTest&rdquo;>     <parameter name="myName" value=&rdquo;John"/>      <classes>       <class name="ParameterizedTest1" />      </classes>    </test>   </suite> |

To extensively understand the working of TestNG and it’s benefit when used with Selenium, watch the below Selenium tutorial video.

## Selenium Training | TestNG Framework For Selenium | Selenium Tutorial For Beginners | Edureka



This Edureka Selenium Training video will take you through the in-depth details of Selenium WebDriver.

### ****38. Explain DataProviders in TestNG using an example. Can I call a single data provider method for multiple functions and classes?****

[*DataProvider*](https://www.edureka.co/blog/dataprovider-in-testng/) is a TestNG feature, which enables us to write DataDriven tests. When we say, it supports DataDriven testing, then it becomes obvious that the same test method can run multiple times with different data-sets. DataProvider is in fact another way of passing parameters to the test method.

**@DataProvider** marks a method as supplying data for a test method. The annotated method must return an Object[] where each Object[] can be assigned to parameter list of the test method.

To use the DataProvider feature in your tests, you have to declare a method annotated by **@DataProvider** and then use the said method in the test method using the ‘dataProvider‘ attribute in the Test annotation.

As far as the second part of the question is concerned, Yes, the same DataProvider can be used in multiple functions and classes by declaring DataProvider in separate class and then reusing it in multiple classes.

### ****39. How to skip a method or a code block in TestNG?****

If you want to skip a particular test method, then you can set the ‘enabled’ parameter in test annotation to false.  
@Test(enabled = false)

By default, the value of ‘enabled’ parameter will be true. Hence it is not necessary to define the annotation as true while defining it.

### ****40. What is soft assertion in Selenium? How can you mark a test case as failed by using soft assertion?****

Soft Assertions are customized error handlers provided by TestNG. Soft Assertions do not throw exceptions when assertion fails, and they simply continue to the next test step. They are commonly used when we want to perform multiple assertions.

To mark a test as failed with soft assertions, call **assertAll()** method at the end of the test.

### ****41. Explain what is Group Test in TestNG?****

In TestNG, methods can be categorized into groups. When a particular group is being executed, all the methods in that group will be executed. We can execute a group by parameterizing it’s name in group attribute of **@Test** annotation. Example: @Test(groups={“xxx”})

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | @Test(groups={&ldquo;Car&rdquo;})  public void drive(){  system.out.println(&ldquo;Driving the vehicle&rdquo;);  }    @Test(groups={&ldquo;Car&rdquo;})  public void changeGear() {  system.out.println("Change Gears&rdquo;);  }    @Test(groups={&ldquo;Car&rdquo;})  public void accelerate(){  system.out.println(&ldquo;Accelerating&rdquo;);  } |

### ****42. How does TestNG allow you to state dependencies? Explain it with an example.****

**Dependency**is a feature in TestNG that allows a test method to depend on a single or a group of test methods. Method dependency only works if the “depend-on-method” is part of the same class or any of the inherited base classes (i.e. while extending a class).

**Syntax:**@Test(dependsOnMethods = { “initEnvironmentTest” })

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | @Test(groups={&ldquo;Car&rdquo;})  public void drive(){  system.out.println(&ldquo;Driving the vehicle&rdquo;);  }    @Test(dependsOnMethods={&ldquo;drive&rdquo;},groups={cars})  public void changeGear() {  system.out.println("Change Gears&rdquo;);  }    @Test(dependsOnMethods={&ldquo;changeGear&rdquo;},groups={&ldquo;Car&rdquo;})  public void accelerate(){  system.out.println(&ldquo;Accelerating&rdquo;);  } |

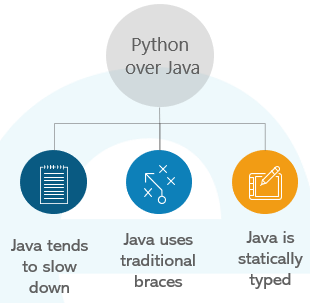
### ****43. Explain what does @Test(invocationCount=?) and @Test(threadPoolSize=?) indicate.****

**@Test(invocationCount=?)** is a parameter that indicates the number of times this method should be invoked.  
**@Test(threadPoolSize=?)** is used for executing suites in parallel. Each suite can be run in a separate thread.

To specify how many times @Test method should be invoked from different threads, you can use the attribute **threadPoolSize** along with **invocationCount**. Example:

|  |  |
| --- | --- |
| 1  2  3 | @Test(threadPoolSize = 3, invocationCount = 10)  public void testServer() {  } |

### ****44. Mention why to choose Python over Java****



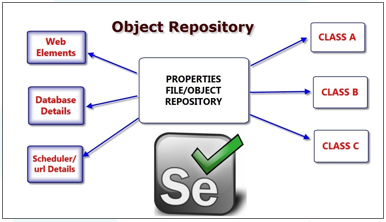
### ****45. How to build the object repository?****

An object repository is a common storage location for all objects. In the Selenium WebDriver context, objects would typically be the locators used to uniquely identify web elements.

[[](https://www.edureka.co/selenium-certification-training)](https://www.edureka.co/selenium-certification-training" \t "_blank)

### [Selenium Certification Training Course](https://www.edureka.co/selenium-certification-training" \t "_blank)

[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/selenium-certification-training" \t "_blank)

**

### ****46. How do you achieve synchronization in WebDriver?****

It is a mechanism that involves more than one component to work parallel with each other.

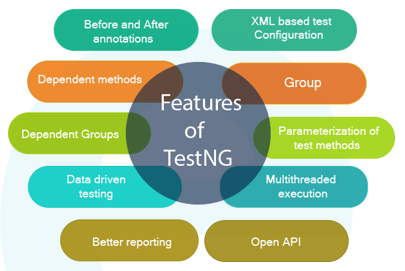
Synchronization can be classified into two categories:

* **Unconditional:** In this we just specify timeout value only. We will make the tool to wait until a certain amount of time and then proceed further.
* **Conditional**: It specifies a condition along with timeout value, so that tool waits to check for the condition and then comes out if nothing happens.

### ****47. What are the different types of TestNG Listeners in Selenium?****

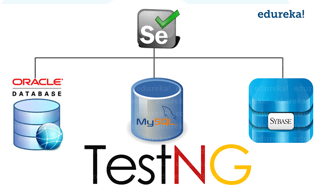
* IAnnotationTransformer
* IAnnotationTransformer2
* IConfigurable
* IConfigurationListener
* IExecutionListener
* IHookable
* IInvokedMethodListener
* IInvokedMethodListener2
* IMethodInterceptor
* IReporter
* ISuiteListener
* ITestListener

### ****48. What are the features of TestNG?****



### ****49. How can you achieve Database Testing using Selenium?****

Selenium does not support [*Database Testing*](https://www.edureka.co/blog/database-testing/), still, it can be partially done using JDBC and ODBC.



### ****50. How can you prepare a customised HTML report in TestNG using Hybrid framework?****

1. [*Junit*](https://www.edureka.co/blog/junit-tutorial/) with the help of an Ant.
2. TestNG using inbuild default file.
3. Also use XSL file.

# BesantTech 300Q: <https://www.besanttechnologies.com/selenium-interview-questions-and-answers>

# MindMajix 26Q: https://mindmajix.com/selenium-interview-questions

### 1. What is Selenese? And explain its types?

A test scripting language specially used by Selenium for creating test cases is called Selenese. It's a cross-platform language used for representing Selenium commands. By using Selenese, we can perform various actions like checking links, different UI elements, dropdown lists, etc. Besides we can also perform actions like testing Ajax functionality, scrolling through a page, and a lot more web application features.

**[ Related Article:**[**What is Selenium**](https://mindmajix.com/what-is-selenium)**? ]**

Mainly selenium commands are of three types:

1. Actions: Manipulates web application state.
2. Accessors: Checks the web application state and sorts the results in some variables.
3. Assertions: Verifies application state corresponding to an expected state specified by the user.

### 2. What are the limitations of Selenium?

* Doesn't supports desktop applications testing
* Unavailability of reliable tech support, as Selenium requires a high level of expertise and resources to manage.
* Since Selenium is open-source software, we need to rely on community forums to get technical issues resolved.
* Doesn’t support automation tests on REST and SOAP platforms.
* Limited support for image testing
* No built-in reposting and test management feature
* Need to rely on third-party tools for testing mobile and desktop applications.

**[ Related Article:**[**Selenium Career Opportunities**](https://mindmajix.com/career-scope-in-selenium)**]**

### 3. List the testing types supported by Selenium?

Types of testing supported by Selenium are as follows:

* Acceptance testing: Determines the feature or system that meets customer expectations and requirements.
* Functional testing: Determines if a feature or system functions well without issues.
* Performance testing: They are performed to measure how well an application is performing
* Regression testing: Generally, this test is done after a fix, change, or feature is added.
* Test-driven development (TDD): It's an iterative development methodology in which test drives the feature design.
* Behavior-driven development (BDD): It's also an iterative development methodology, in which the objective is to include all the parties in the development of an application.

### 4. What is the latest version of Selenium?

Selenium's latest version is Selenium 4.0.0 Alpha 5, which is released in March 2020.

Selenium 4.0 comes with new standardization and offers a seamless experience.

The key features of Selenium 4.0 Alpha 5 are:

* Improved docker support using domain sockets and the DOCKER\_HOST env variable
* Replaces OpenTracing with OpenTelemetry.
* Instead of creating a client per session, reuses the same HTTP client.

|  |
| --- |
| If you want to become a Selenium Certified Specialist, then visit Mindmajix - A Global online training platform: “[**Selenium Training**](https://mindmajix.com/selenium-training)”.  This course will help you to achieve excellence in this domain. |

### 5. What's new in Selenium 4.0?

* W3C WebDriver Standardization
* Selenium 4 IDE TNG
* Refreshed Documentation
* Improved Selenium Grid
* Better Window and Tab management
* Relative locators

### 6. How can you click on a hyperlink in Selenium?

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

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

The below command finds the element based on the substring of the link in the parenthesis and partial link text() finds the web element.

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

### 7. How to add text in the text box without using sendkeys()?

Using JavaScriptExecutor, we can enter text in the text box.

JavascriptExecutor jse = (JavascriptExecutor) driver;

jse.executeScript("document.getElementById('email').value="123.ab@xyz.com");

### 8. What is StaleElementReferenceException, and how do you handle it?

Stale means old and no longer new. A stale element means an old element. Any element present on the webpage is considered as a web element in Web Driver. If the document object model (DOM) changes, then the web element goes stale. If we try to connect with any element that is staled, then StaleElementReferenceException is thrown.

The reasons for StaleElementReferenceException thrown are, one is a web element deleted entirely and the next id element no longer attached to DOM.

To handle StaleElementReferenceException, follow the below-mentioned ways:

* **Solution 1:** Refresh the page and try again for the same element

Sample code to overcome the issue:

driver.navigate().refresh();

driver.findElement(By.xpath("xpath here")).click();

* **Solution 2:** Use try-catch block within for loop, if the element is not found in DOM.

// Using for loop, it tries for 4 times.

// If the element is located for the first time then it breaks from the for loop nad comeout of the loop

for(int i=0; i<=3;i++){

try{

driver.findElement(By.xpath("xpath here")).click();

break;

}

catch(Exception e){

Sysout(e.getMessage());

}

}

* **Solution 3:** Wait for element till it gets available

wait.until(ExpectedConditions.presenceOfElementLocated(By.id("table")));

Use ExpectedConditions.refreshed to avoid StaleElementReferenceException and retrieve the element again. This element updates the element by redrawing it accessing the referenced element.

wait.until(ExpectedConditions.refreshed(ExpectedConditions.stalenessOf("table")));

* **Solution 4:** Handle StaleElementElement Exception by POM.

### 9. What is the same-origin policy? And how can you avoid it?

The same-origin policy is a critical security mechanism that defines how the script or document loaded from one origin can associate with a resource from another origin. It helps to isolate potential malicious documents and rescues from attack vectors.

For example, for a URL http:// http://www.google.com/resources/, the origin is a combination of a google.com, http, 80 correspondingly. Selenium Core (JavaScript program) cannot obtain the elements from an origin that is different from where it is launched.

If we have launched a JavaScript program from "http://www.google.com", then it would be easy to access pages. To handle the same-origin policy, Selenium RC was introduced. In this, the server acts as a client-configured HTTP proxy and tricks the browser into believing Selenium Core and Web application tested are from the same origin.

### 10. What are locators in Selenium?

Locator in Selenium is a command that tells Selenium IDE about which [GUI](https://www.computerhope.com/jargon/g/gui.htm) elements need to operate. There are various types of locators available in Selenium WebDriver, and its choice depends largely on the application under test.

**[ Related Article:**[**Learn Selenium IDE**](https://mindmajix.com/what-is-selenium-ide)**]**

* **Locating by ID:** It takes a string parameter which is an ID attribute value and returns the object to the findElement() method.

driver.findElement(By.id("user"));

* **Locating by TagName:** It locates all elements with a matching tag name.

driver.findElement(By.tagName("button").click());

* L**ocating by Link:** The target link can be located using a by.link text locator.

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

* **Locating by Name:** The first element with the name attribute value will return the location matched.

driver.findElement(By.name("books").click());

* **Locating by XPath:** It takes a string parameter which is XPathExpression and returns it to an object to findElement() method.

driver.findElement(By.xpath("//span[contains(text(),'an account')]")).getText();

* **Locating by ClassName:** It finds elements based on class attribute value.

driver.findElement(By.className("inputtext"));

* **Locating by CSS selector:** It locates elements based on the drivers underlying the CSS selector engine.

driver.findElement(By.cssSelector("input#email")).sendKeys("myemail@email.com");

## Advanced Selenium Interview Questions

### 11. How many types of waits are there in Selenium?

Waits are commands in Selenium that are important for executing test scripts.

There are three types of waits in Selenium, such as:

* **Implicit Wait Type:** Implicit wait commands direct Selenium WebDriver to wait for a certain measure of time before throwing a "no such element" exception.

**Syntax:**

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

* **Explicit Wait Type:** Explicit wait command directs WebDriver to wait until a certain condition occurs before proceeding with executing the code.

**Syntax:**

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

* **Fluent Wait Type:** Fluent wait command defines Selenium WebDriver to wait for a specific condition to appear. And, also determines the frequency with which Selenium Webdriver checks the state that appeared before by throwing "ElementNotVisibleException".

**Syntax:**

Wait wait = new FluentWait(WebDriver reference)

.withTimeout(timeout, SECONDS)

.pollingEvery(timeout, SECONDS)

.ignoring(Exception.class);

### 12. What are the most commonly used Browser navigation commands for Selenium WebDriver?

* **Navigate To Command:** This method loads a new web page in the current browser window and returns a string as a parameter.

driver.navigate().to(appUrl);

* **Forward Command:** This method does the same operation as the forward button of any browser. It neither returns nor accepts anything

driver.navigate().forward();

* **Back Command:** This method does the same operation like the back button of any browser. Neither returns nor accepts anything.

driver.navigate().back();

* **Refresh Command:** This method is used for refreshing the current page.

driver.navigate().refresh();

### ****[ Related Article:****[Selenium Online Tutorial](https://mindmajix.com/selenium-tutorial)****]****

### 13. How is the Page Object Model(POM) different from Page Factory?

Page Object Model is a class that represents a web page and its functionality and members. Page Factory is a way of initializing web elements you want to interact with the page object while creating an instance of it.

### 14. How to handle pop-ups in Selenium?

The robot class in Selenium is used to handle mouse and keyboard functions. It closes the pop windows. Using the WindowHandle() function, we can handle the pop-up window. By using third-party tools also we can handle pop-ups and other window-based features.

### 15. What is the use of switchTo() command in Selenium?

switchTo() Command is used to switch the focus to a new window browser, by supplying the Window Handle or Window Name as an argument to the Command.

driver.SwitchTo().Window(WindowHandle);

driver.SwitchTo().Window(WindowName);

To get the current Window Handle, use the following command:

String currentWindowHandle = driver.CurrentWindowHandle;

### 16. How is findElement() different from findElements()?

* findElement() is used for accessing a single web element on a page. It returns the order of the first matching element of the specified order. It throws a NoSuchElementException when it fails to send the element.

**Syntax:**

driver.findElement(By.xpath("Value of Xpath"));

* findElements() returns the list of all the matching elements. findElements method returns an empty list when an +element is not available or doesn't exist on the page.

**Syntax:**

List link = driver.findElements(By.xpath("Value of Xpath"));

### 17. What is the major difference between "/" and "//" in Xpath?

* **Single Slash** “/”: It is used to create Xpath with an absolute path that is Xpath would be created to start selection from document node/start node.

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

* **Double Slash** "//": It is used to create Xpath with a relative path that is Xpath would be created to create start selection from anywhere within the document

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

### 18. How can you launch a browser using Selenium WebDriver?

The following syntax is used to launch a browser using Selenium WebDriver:

* WebDriver driver = new InternetExplorerDriver();
* WebDriver driver = new ChromeDriver();
* WebDriver driver = new FirefoxDriver();

### 19. How to verify whether an element is displayed or not on screen in Selenium?

To verify the visibility of web elements like checkbox, edit box, radio button, etc., use the following methods:

* isdisplayed(): Checks if a web element is present on the screen.

**Syntax:**

Boolean result = driver.findElement(By.xpath("//span[text()='Coding Ground']")).isDispayed();

* **isSelected():** Checks the status of the check box, radio button, and options in the static dropdown.

**Syntax:**

Boolean btnresult = driver.findElement(By.xpath("//xpath[contains(@class,'gsc-search-button')]")).isSelected();

* **isEnabled():** Checks if an element is enabled or not.

**Syntax:**

Boolean btnresult = driver.findElement(By.xpath("//xpath[contains(@class,'gsc-search-button')]")).isEnabled();

### 20. How to select a value from DropDown using Selenium Webdriver?

The value in the dropdown can be selected using WebDriver's Select class.

**Syntax:**

**select By VisibleText:**

Select selectByVisibleText = new Select (driver.findElement(By.id("SelectID\_Two")));

selectByVisibleText.selectByVisibleText("Lime");

**select By Value:**

Select selectByValue = new Select(driver.findElement(By.id("SelectID\_One")));

selectByValue.selectByValue("greenvalue");

**select By Index:**

Select selectByIndex = new Select(driver.findElement(By.id("SelectID\_Three")));

selectByIndex.selectByIndex(2);

### 21. How to handle frames using WebDriver?

The iFrame is an inline frame or web page used for inserting another document within the current HTML document. It is often used to add content from other sources like an advertisement on a web page. It is defined with the "iframe" tag.

Select iframe by id

driver.switchTo().frame("ID of the frame");

* **Locating iframe using tagName**

driver.switchTo().frame(driver.findElements(By.tagName("iframe").get(0));

* **Locating iframe using index**

frame(index)

driver.switchTo().frame(0);

* **Switch to Frame by WebElement**

frame(WebElement element)

### 22. What is the difference between driver.quit() command and driver.close()?

driver. quit() is used for exiting the browser, tabs, pop-ups, session, etc., whereas the driver.close() is used to close the web browser window that the user is currently working on.

### 23. How to perform mouse hover functions on a web element using WebDriver?

WebDriver offers a wide range of utilities that users can exploit to automate mouse and keyboard events. Action interface is one such utility that simulates single user interactions.

Here are the methods Actions class has provided for Mouse Hover action:

* moveToElement(WebElement target)
* moveToElement(WebElement target, int xOffset, int yOffset)

### 24. What are Junit annotations?

Junit annotations in Selenium are used for identifying method types defined in test code. To execute Selenium WebDriver testing with JUnit, it's necessary to add Junit annotation in the script.

Commonly used JUnit Annotations in Selenium are listed below:

* @BeforeClass
* @Before
* @Test
* @After
* @AfterClass
* @Ignore

### 25. Can Captcha be automated?

No, captcha and barcode readers cannot be automated. Captcha functionality is to ensure that automated programs and bots don't get access to sensitive information - which is why Selenium cannot automate it.

### 26. How to minimize and maximize browsers in Selenium with python?

We can minimize and maximize the browsers while we are testing an application in Selenium.

maximize() method is used for maximizing the browser, and minimize() method is used for minimizing the browser.

For example,

from selenium import webdriver

#browser exposes an executable file

#Through Selenium test we will invoke the executable file which will then #invoke actual browser

driver = webdriver.Firefox(executable\_path="C:geckodriver.exe")

# to maximize the browser window

driver.maximize\_window()

#get method to launch the URL

driver.get("https://www.mindmajix.com/index.htm")

#to refresh the browser

driver.refresh()

# to minimize the browser window

driver.minimize\_window()

#to close the browser

driver.close()

# KnowledgeHut 101Q: <https://www.knowledgehut.com/interview-questions/selenium>