Selenium Interview Questions

**Javatpoint.com**

Selenium is based on automating web applications for testing purpose, but it is certainly not limited to just that. The web-based administration tasks can be automated as well. It automates browsers

Selenium has the support of some of the leading browser vendors who have adopted it to make Selenium an essential part of their browser. It is also the core technology in many other browser automation tools, APIs, and frameworks.

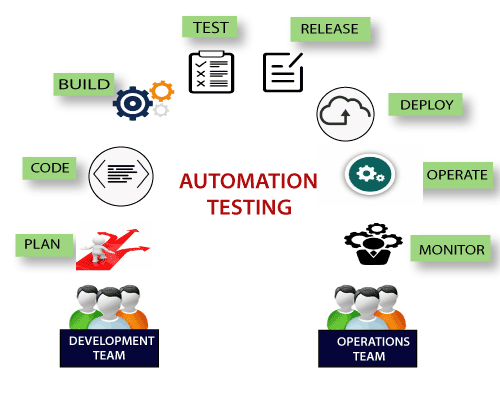
A list of most frequently asked **Selenium interview questions**, and their answers are given below.

Basic Level - 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.

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.

AD

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:

Selenium Integrated Development Environment (IDE)

Selenium Remote Control (Now Deprecated)

WebDriver

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** : Functional testing is **a type of testing that seeks to establish whether each application feature works as per the software requirements**. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations.

**Regression Testing:** Regression testing is **a software testing practice that ensures an application still functions as expected after any code changes, updates, or improvements**. Regression testing is responsible for the overall stability and functionality of the existing features.

**Sanity Testing**: Sanity testing is **a type of regression testing used in quality assurance**. The main purpose of sanity testing is to make sure that the proposed modifications or functionality works as intended. If the sanity test fails, the testing team will reject the software product to save time and money

**Smoke Testing**: Smoke Testing is done whenever the new functionalities of software are developed and integrated with existing build that is deployed in QA/staging environment. It ensures that all critical functionalities are working correctly or not.

**Responsive Testing**: Responsive Test **lets you assess your design on different screen widths for a 'true' test of 'adaptivity'**. The goal of responsive testing is **to check if all the website components are adaptive to different screen sizes**.

Cross Browser Testing: **Cross Browser testing** is a type of non-functional testing that lets you check whether your website works as intended when accessed through:

* **Different Browser-OS combinations**
* **Different devices** i.e., users can view and interact with your website on popular devices—smartphones, tablets, desktops and laptops etc.
* **Assistive Tools** i.e., the website is compatible with assistive technologies like screen readers for individuals who are differently abled.

UI testing (black box): UI Testing, also known as GUI Testing is basically **a mechanism meant to test the aspects of any software that a user will come into contact with**. This usually means testing the visual elements to verify that they are functioning according to requirements – in terms of functionality and performance.

Integration Testing: What is integration testing explain?

Integration testing -- also known as integration and testing (I&T) -- is **a type of software testing in which the different units, modules or components of a software application are tested as a combined entity**

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:

Implicit Wait

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.

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.

**public** **class** FirefoxBrowserLaunchDemo {

**public** **static** **void** main(String[] args) {

//Creating a driver object referencing WebDriver interface

WebDriver driver;

//Setting webdriver.gecko.driver property

System.setProperty("webdriver.gecko.driver", pathToGeckoDriver + "\\geckodriver.exe");

//Instantiating driver object and launching browser

driver = newFirefoxDriver();

//Using get() method to open a webpage

driver.get("http://javatpoint.com");

//Closing the browser

driver.quit();

    }

}

31) Write a code snippet to launch Chrome browser in WebDriver.

**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");

//Closing the browser

driver.quit();

    }

}

32) Write a code snippet to launch Internet Explorer browser in WebDriver.

**public** **class** IEBrowserLaunchDemo {

**public** **static** **void** main(String[] args) {

//Creating a driver object referencing WebDriver interface

WebDriver driver;

//Setting the webdriver.ie.driver property to its executable's location

System.setProperty("webdriver.ie.driver", "/lib/IEDriverServer/IEDriverServer.exe");

//Instantiating driver object

driver = newInternetExplorerDriver();

//Using get() method to open a webpage

driver.get("http://javatpoint.com");

//Closing the browser

driver.quit();

    }

}

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.

Actions action = newActions(driver);

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

action.contextClick(element).perform();

34) Write a code snippet to perform mouse hover in WebDriver.

Actions action = newActions(driver);

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

action.moveToElement(element).perform();

35) How do you perform drag and drop operation in WebDriver?

Code snippet to perform drag and drop operation:

//WebElement on which drag and drop operation needs to be performed

WebElementfromWebElement = driver.findElement(By Locator of fromWebElement);

//WebElement to which the above object is dropped

WebElementtoWebElement = driver.findElement(By Locator of toWebElement);

//Creating object of Actions class to build composite actions

Actions builder = newActions(driver);

//Building a drag and drop action

Action dragAndDrop = builder.clickAndHold(fromWebElement)

             .moveToElement(toWebElement)

             .release(toWebElement)

         .build();

//Performing the drag and drop action

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 -

driver.navigate().refresh();

2. Using driver.getCurrentUrl() with driver.get() command -

driver.get(driver.getCurrentUrl());

3. Using driver.getCurrentUrl() with driver.navigate() command -

driver.navigate().to(driver.getCurrentUrl());

4. Pressing an F5 key on any textbox using the sendKeys command -

driver.findElement(By textboxLocator).sendKeys(Keys.F5);

5. Passing ascii value of the F5 key, i.e., "\uE035" using the sendKeys command -

driver.findElement(By textboxLocator).sendKeys("\uE035");

37) Write a code snippet to navigate back and forward in browser history?

Navigate back in browser history:

driver.navigate().back();

Navigate forward in browser history:

driver.navigate().forward();

38) How to invoke an application in WebDriver?

driver.get("url"); or

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**

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:**

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");

Select selectByIndex = **new** Select(driver.findElement(By.id("SelectID\_Three")));

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**

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**

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**

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**

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**

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();

44) Is there an HtmlUnitDriver for .NET?

To use HtmlUnit first use the RemoteWebDriver and pass it in the desired capabilities.

IWebDriver driver

= **new** RemoteWebDriver(DesiredCapabilities.HtmlUnit())

For the Firefox implementation to run, use

IWebDriver driver

= **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**

String PROXY = "199.201.125.147:8080";

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);

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.

**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);

// paste the screenshot in the desired location

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

}

}

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**

WebElement username = drv.findElement(By.id("Email"));

// entering username

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:**

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();

50) How to click on a hyper link using linkText?

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.

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.

**Guru99.com**

Basic Selenium Interview Questions for Freshers

1) What is Selenium and what is composed of?

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

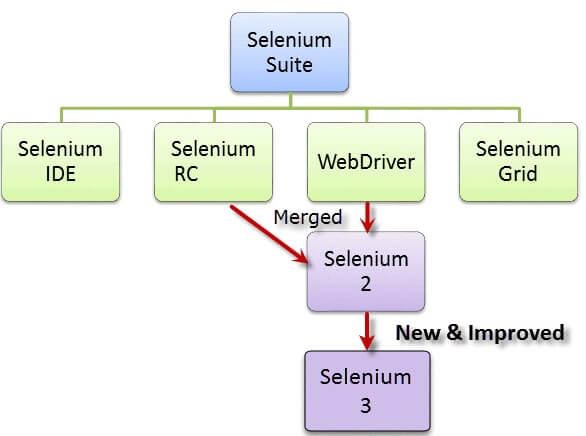
**Selenium IDE (Integrated Development Environment) :**It is a tool for recording and playing back.  It is a firefox plugin

**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

**AD**

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.

**AD**

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.

**Software Testing Material**

<https://www.softwaretestingmaterial.com/selenium-interview-questions/>

**Basic & Advanced Selenium WebDriver Interview Questions:**

No matter you are experienced or freshers, we have covered all levels based on the requests we got from our readers. Here we have covered Selenium Interview Questions asked in companies.

**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.

Saves time and money. Automation testing is faster in execution.

Reusability of code. Create one time and execute multiple times with less or no maintenance.

Easy reporting. It generates automatic reports after test execution.

Easy for compatibility testing. It enables parallel execution in the combination of different OS and browser environments.

Low-cost maintenance. It is cheaper compared to manual testing in a long run.

Automated testing is more reliable.

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.,

It is mostly used for regression testing. Supports execution of repeated test cases.

Minimal manual intervention. Test scripts can be run unattended.

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

We cannot test the windows application

We cannot test mobile apps

Limited reporting

Handling dynamic Elements

Handling page load

Handling pop up windows

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?**

Reusability of code.

Easy reporting.

Low-cost maintenance.

Maximum Coverage

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/).

# How To Explain Test Automation Framework To The Interviewer

By[Rajkumar](https://www.softwaretestingmaterial.com/)Updated onJuly 18, 2020

Before going to see how to explain **test automation framework** to the interviewer, I would highly recommend you go through the below link to learn more about the following

* [Types of Test Automation Frameworks](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/)
* [Test Automation Framework Interview Questions](https://www.softwaretestingmaterial.com/test-automation-framework-interview-questions/)

Coming back to this ultimate guide to explain the automation framework in the interview. Here I will explain to you every component of the architecture based on the below-mentioned screenshot. Frame your answer in your own words while explaining the framework to the interviewer.

We need to specify in and out of our Test Automation Framework such as programming **language** used, **Type of framework** used, Test Base Class (Initializing WebDriver, Implicit Waits), How we separate Element locators and tests (Page Objects, Page Factory), Utility functions file, Property files, TestNG annotations, How we parameterize tests using Excel files, How we capture error screenshots, Generating reports(Extent Reports), Emailing reports, Version Control System used and Continues Integration Tool used.

***Must Read:***[***Where you have applied OOPs in your Automation Framework?***](https://www.softwaretestingmaterial.com/oops-concept-in-automation-framework/)

Check below video to see “Selenium Automation Framework Explanation”

If you liked this video, then please subscribe to our YouTube Channel for more video tutorials.

**Language:**In our Selenium Project we are using Java language. Even though Selenium supports multiple languages, we are using Java language is just because most of the automation developers have knowledge on Selenium with Java.

[***Java For Selenium***](https://www.softwaretestingmaterial.com/java-tutorial/)

**Type of Framework:** In our project, we are using [Data-driven Framework](https://www.softwaretestingmaterial.com/data-driven-framework-selenium-webdriver/) by using [Page Object Model design pattern](https://www.softwaretestingmaterial.com/page-object-model/) with Page Factory.

***Must Read:***[***Types of Test Automation Framework***](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/)

**POM:**As per the Page Object Model, we have maintained a class for every web page. Each web page has a separate class and that class holds the functionality and members of that web page. Separate classes for every individual test.

[***Difference between Page Object Model and Page Factory***](https://www.softwaretestingmaterial.com/page-object-model/#Difference-Between-Page-Object-Model-And-Page-Factory)

**Packages:** We have separate packages for Pages and Tests. All the web page related classes come under the **Pages** package and all the tests related classes come under **Tests** package.

For example, Home Page and Login Page have separate classes to store element locators. For the login test, there would be a separate class which calls the methods from the Home Page class and Login Page class.

I will explain based on the below-mentioned test automation framework structure.

The above screenshot illustrates a standardized maven project. As per the above maven project, all the tests are kept in the ‘***src/test/java***‘ and remaining files (such as config.properties, element locators (POM classes), utility files, test data, etc.,) kept under ‘***src/main/java***‘.

[***Where you have applied OOPS in Automation Framework***](https://www.softwaretestingmaterial.com/oops-concept-in-automation-framework/)

**Test Base Class:**Test Base class (TestBase.java) deals with all the common functions used by all the pages. This class is responsible for loading the configurations from properties files, Initializing the WebDriver, Implicit Waits, Extent Reports, and also to create the object of FileInputStream which is responsible for pointing towards the file from which the data should be read.

**Utility Class (AKA Functions Class):**Utility class (TestUtil.java) stores and handles the functions (The code which is repetitive in nature such as waits, actions, capturing screenshots, accessing excels, sending email, etc.,) which can be commonly used across the entire framework. The reason behind creating a utility class is to achieve reusability. This class extends the TestBase class to inherit the properties of TestBase in TestUtil.

**Properties file:**This file (**config.properties**) stores the information that remains static throughout the framework such as browser-specific information, application URL, screenshots path, etc.

All the details which change as per the environment and authorization such as URL, Login Credentials are kept in the config.properties file. Keeping these details in a separate file makes it easy to maintain.

**Screenshots:** Screenshots will be captured and stored in a separate folder and also the screenshots of failed test cases will be added to the extent reports.

[***How To Capture Screenshots of Failed Test Cases***](https://www.softwaretestingmaterial.com/capture-screenshot-of-failed-test-cases-using-selenium-webdriver-2/)

**Test Data:** All the historical test data will be kept in an excel sheet (controller.xlsx). By using ‘controller.xlsx’, we pass test data and handle data-driven testing. We use [Apache POI](https://www.softwaretestingmaterial.com/handling-excel-files-using-apache-poi/) to handle excel sheets.

**TestNG:** Using TestNG for Assertions, Grouping, and Parallel execution.

Here you could find [TestNG Complete Tutorial](https://www.softwaretestingmaterial.com/testng-tutorial/) and also you could find [TestNG Interview Questions](https://www.softwaretestingmaterial.com/testng-interview-questions/)

**Maven:** Using Maven for build, execution, and dependency purpose. Integrating the TestNG dependency in the POM.xml file and running this POM.xml file using Jenkins.

[***How To Create Maven Project***](https://www.softwaretestingmaterial.com/create-selenium-maven-project/)

**Version Control Tool:** We use Git as a repository to store our test scripts.

**Jenkins:** By using Jenkins CI (Continuous Integration) Tool, we execute test cases on a daily basis and also for nightly execution based on the schedule. Test Results will be sent to the peers using Jenkins.

[***Maven Project with Jenkins***](https://www.softwaretestingmaterial.com/execute-maven-project-using-jenkins/)

**Extent Reports:** For the reporting purpose, we are using Extent Reports. It generates beautiful HTML reports. We use the extent reports for maintaining logs and also to include the screenshots of failed test cases in the Extent Report.

Here you could find [How To Generate Extent Reports](https://www.softwaretestingmaterial.com/generate-extent-reports/) and also find [How To Add Screenshots In Extent Reports](https://www.softwaretestingmaterial.com/screenshots-extent-reports/).

You have to explain all these when you are asked to explain the test automation framework in the interview. If you have any other thoughts on how to explain the test automation framework, comment below in the comments section.

**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,

Selenium IDE (Selenium Integrated Development Environment)

Selenium RC (Selenium Remote Control)

Selenium WebDriver

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/" \t "_blank). 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:

ID – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-id-locator/)

findElement(By.id("inputtext"))

ClassName – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-class-name-locator/)

findElement(By.className("inputtext"))

Name – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-name-locator/)

findElement(By.name("inputtext"))

TagName – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-tag-name-locator/)

findElement(By.tagName("a”))

LinkText – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-link-text-and-partial-link-text-locator/)

findElement(By.linkText("Create account"))

PartialLinkText – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-link-text-and-partial-link-text-locator/)

findElement(By.partialLinkText("Create accout"))

XPath – [Practical example](https://www.softwaretestingmaterial.com/how-to-locate-element-by-xpath-locator/)

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

ElementNotVisibleException

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

The element has been deleted entirely.

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/)

Implicit Waits:

Implicit waits tell to the WebDriver to wait for 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.

Note: Implicit Wait is in place for the entire time the browser is open. Time taken to search all the elements are based on the time fixed for the implicit wait.

Syntax:

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

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

Implicit Wait time is applied to all the elements in the script.

Implicit wait will accept 2 parameters, the first parameter will accept the time as an integer value and the second parameter will accept the time measurement in terms of SECONDS, MINUTES, MILISECOND, MICROSECONDS, NANOSECONDS, DAYS, HOURS, etc.

Test script with an explanation – Implicit Waits in Selenium:

Find the sample script (using Java) mentioned below. Execute it to see the functionality of Implicit Wait in Selenium.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | package waits;  import java.util.concurrent.TimeUnit;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.firefox.FirefoxDriver;  public class ImplicitWaits {  public static void main(String[] args) {  //To create a new instance of Firefox Driver  WebDriver driver = new FirefoxDriver();  //Implicit Wait - Here the specified Implicit Wait time frame is 15 seconds.  //It waits 15 seconds of time frame for the element to load.  //It throws an exception, if the element is not loaded within the specified time frame  driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);  //To open a website "Software Testing Material"  driver.get("https://www.softwaretestingmaterial.com");  //To maximize the browser window  driver.manage().window().maximize();  //To close the browser  driver.close();  }  } |

There are some instances when a particular element takes more time (eg. 1 min) to load. In such cases setting a huge time to Implicit wait makes the browser to wait for the same time for every element. To avoid this, we need to implement Explicit Waits.

Selenium FluentWait is one of the Explicit waits.

Explicit waits are confined to a particular web element. Explicit Wait is code you define to wait for a certain condition to occur before proceeding further in the code.

Types of Explicit Waits:

1. WebDriverWait
2. FluentWait

Here we discuss about Selenium FluentWait. Click on this link for [WebDriverWait](https://www.softwaretestingmaterial.com/webdriverwait-selenium-webdriver).

## **Selenium FluentWait:**

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.

To say in effortless manner, it tries to find the web element repeatedly at regular intervals of time until the timeout or till the object gets found.

We use FluentWait commands mainly when we have web elements which sometimes visible in few seconds and some times take more time than usual. Mainly in Ajax applications. We could set the default pooling period based on our requirement. We could ignore any exception while polling an element.

***Must Read:***[***Waits in Selenium – Implicit, Explicit, FluentWait***](https://www.softwaretestingmaterial.com/selenium-wait-commands)

**Syntax:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Wait wait = new FluentWait(WebDriver reference)  .withTimeout(timeout, SECONDS)  .pollingEvery(timeout, SECONDS)  .ignoring(Exception.class);    WebElement foo=wait.until(new Function<WebDriver, WebElement>() {  public WebElement applyy(WebDriver driver) {  return driver.findElement(By.id("foo"));  }  }); |

**Example:**

|  |  |
| --- | --- |
| 1  2  3  4 | Wait wait = new FluentWait<WebDriver>(driver)  .withTimeout(45, TimeUnit.SECONDS)  .pollingevery(5, TimeUnit.SECONDS)  .ignoring(NoSuchElementException.class); |

FluentWait uses two parameters mainly – timeout value and polling frequency. In the above syntax we took time out value as 45 seconds and polling frequency as 5 seconds. The maximum amount of time (45 seconds) to wait for a condition and the frequency (5 seconds) to check the success or failure of a specified condition. If the element is located with in this time frame it will perform the operations else it will throw an “ElementNotVisibleException”

### ****Test script with an explanation – Selenium FluentWait:****

Find the sample script (using Java) mentioned below. Execute it to see the functionality of a Selenium FluentWait.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | package waits;  import java.util.concurrent.TimeUnit;  import org.openqa.selenium.By;  import org.openqa.selenium.NoSuchElementException;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.WebElement;  import org.openqa.selenium.firefox.FirefoxDriver;  import org.openqa.selenium.support.ui.FluentWait; //FluentWait is a Class and it is a part of this package  import org.testng.annotations.Test;    import com.google.common.base.Function;    public class FluentWaitClass {  *@Test*  public static void fluentWaitMethod(){  System.setProperty("webdriver.gecko.driver","D://Selenium Environment//Drivers//geckodriver.exe");  WebDriver driver = new FirefoxDriver();  driver.get("http://softwaretestingplace.blogspot.com/2017/02/selenium-fluent-wait.html");  driver.findElement(By.xpath("//\*[@id='post-body-5280210221385817166']/div[1]/button")).click();  FluentWait<WebDriver> wait = new FluentWait<WebDriver>(driver)         .withTimeout(30, TimeUnit.SECONDS)         .pollingEvery(5, TimeUnit.SECONDS)         .ignoring(NoSuchElementException.class);    WebElement element = wait.until(new Function<WebDriver, WebElement>() {  public WebElement apply(WebDriver driver) {  WebElement element = driver.findElement(By.xpath("//\*[@id='softwareTestingMaterial']"));  String getTextOnPage = element.getText();  if(getTextOnPage.equals("Software Testing Material - DEMO PAGE")){  System.out.println(getTextOnPage);  return element;  }else{  System.out.println("FluentWait Failed");  return null;  }  }  });  }  } |

If

**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.

isDisplayed()



|  |  |
| --- | --- |
| 1 | boolean elePresent = driver.findElement(By.xpath("xpath")).isDisplayed(); |

isSelected()



|  |  |
| --- | --- |
| 1 | boolean eleSelected= driver.findElement(By.xpath("xpath")).isSelected(); |

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

[Simplilearn.com](https://hackr.io/blog/selenium-interview-questions)

**Table of Contents**

[Beginner Level Selenium Interview Questions](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-interview-questions-and-answers#beginner_level_selenium_interview_questions)

[Intermediate Level Selenium Interview Questions](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-interview-questions-and-answers#intermediate_level_selenium_interview_questions)

[Advanced Level Selenium Interview Questions](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-interview-questions-and-answers#advanced_level_selenium_interview_questions)

Testing is a crucial phase in the software development life cycle (SDLC). Manual testing, in particular, involves the physical execution of test cases against the applications to detect bugs and error. However, manual testing posed a lot of challenges, and a method to automate the testing process was in demand. As a result, Selenium, a powerful [automated testing tool](https://www.simplilearn.com/what-is-automation-testing-article), was introduced. This is why grabbing a selenium-related job is definitely a plus point, and cracking the selenium interview questions is the key!

And if you’re looking forward to a Selenium interview, here are the top 30 frequently asked selenium interview questions.

**Full Stack Java Developer Course**

In Partnership with HIRIST and HackerEarth[EXPLORE COURSE](https://www.simplilearn.com/java-full-stack-developer-certification-training-course?source=GhPreviewCTABanner)



**Beginner Level Selenium Interview Questions**

**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?**

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

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

Limited support for image testing.

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

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)



**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 -

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

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

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

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

The following are the steps involved:

Identify test input

Compute test outcome

Execute test

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



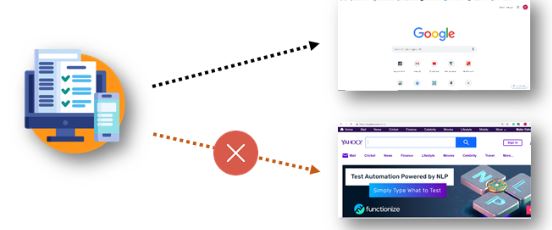
Selenium 3.0 is the extended version of Selenium 2.0. It is inherently backward compatible and does not involve Selenium RC. The new version came along with several bug fixes and increased stability.

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.

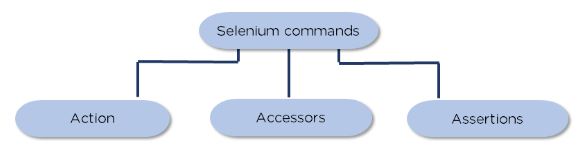
**Free Course: Introduction to Selenium**

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)



**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”));

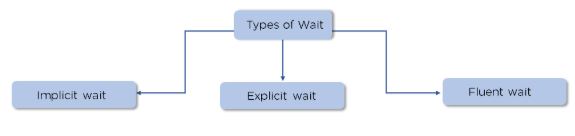
**Locator by XPath:** It takes a parameter of String which is a XPATHEXPRESSION and it returns an object to findElement() method.

  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.

**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.**

Selenium is easy to use since it’s essentially developed in JavaScript.

Selenium can test web applications against browsers like Firefox, Opera, Chrome, and Safari, to name a few.

The test code can be written in various programming languages like Java, Perl, Python, and PHP.

Selenium is platform-independent, and can be deployed on different Operating systems like Windows, Linux, and Macintosh.

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.

**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

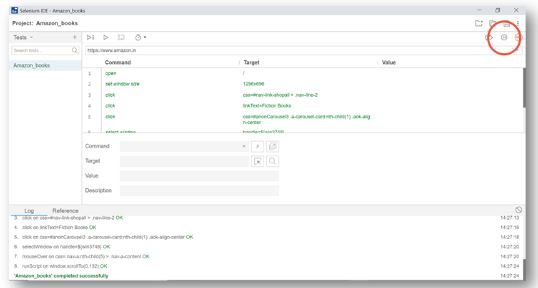
**General Syntax:**

  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)



**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:

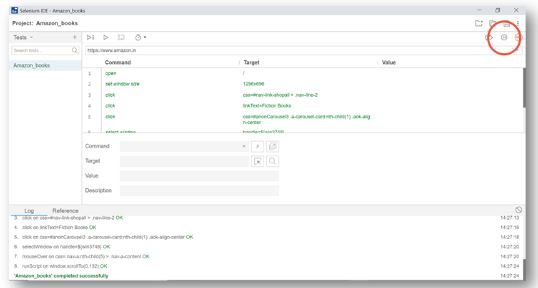
   WebElement element = driver.findElement(By.id(example));

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:

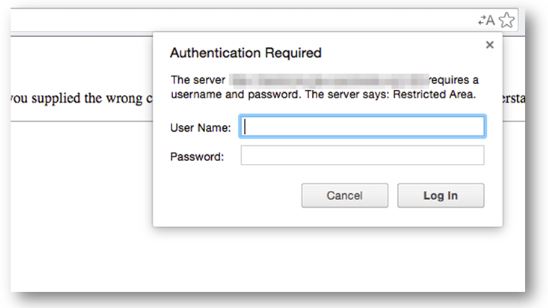
   List <WebElement> elementList = driver.findElements(By.id(example));

**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”);

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

**Your Next Step to Success**

We have an industry expert who illustrates certain pieces of code while explaining how to answer the selenium interview questions.

Apart from the above mentioned interview questions, it is also important to have a fair understanding of frequently asked [DevOps interview questions](https://www.simplilearn.com/tutorials/devops-tutorial/devops-interview-questions).

If you wish to embark on your journey as a test automation engineer, and add another shining star to your selenium interview, then here’s a certification course that will come in handy. Check out Simplilearn’s [Selenium training course](https://www.simplilearn.com/selenium-certification-training?referrer=search&tag=Selenium). This training is designed to train developers and manual testers to learn how to automate web applications with a robust framework, and integrate it within the DevOps processes of an organization, and help you master important concepts such as TestNG, Selenium IDE, and Selenium Grid.

Hope you liked the article Selenium Interview Questions, in case of any doubts you can ask your questions in the below section.