**1. How do I launch the browser using WebDriver?**  
**Answer:**The following syntax can be used to launch the Browser:

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

**2. What is the difference between POI and jxl jar?**  
**Answer:**  
JXL supports the “.xls” format i.e. binary-based format. JXL doesn’t support Excel 2007 and “.xlsx” format i.e. XML based format  
JXL API was last updated in the year 2009  
The JXL documentation is not as comprehensive as that of POI  
JXL API doesn’t support rich text formatting  
JXL API is faster than POI API

POI jar supports all of these formats  
POI is regularly updated and released  
POI has a well prepared and highly comprehensive documentation  
POI API supports rich text formatting  
POI API is slower than JXL API

**3. How to assert the title of the web page?  
Answer:**//verify the title of the web page

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

**4. What is a data-driven framework?**  
**Answer:** The Data-Driven test design framework follows a design paradigm where test logic is fixed but varies the test data. The data itself can be in different repositories like a simple .csv file, .json file or .xls sheet, or database and can add the tests merely updating those external files or DB (instead of placing in test code itself).

**5. What is Object Repository? How can we create an Object Repository in Selenium?**  
**Answer**: Object Repository is a term used to refer to the collection of web elements belonging to Application Under Test (AUT) along with their locator values. Thus, whenever the element is required within the script, the locator value can be populated from the Object Repository. Object [Repository](https://en.wikipedia.org/wiki/Selenium_(software)) is used to store locators in a centralized location instead of hard coding them within the scripts.  
In Selenium, objects can be stored in an excel sheet that can be populated inside the script whenever required.  
That’s all for now.  
Hope in this article you will find answers to most frequently asked Selenium and WebDriver Interview questions. The answers provided here are also helpful for understanding the Selenium basics and advanced WebDriver topics.  
Do you have any Selenium Interview questions that are not answered here? Please let us know in the comments below and we will try to answer all.

**6. How a text written in a text field could be cleared?**  
**Answer:** A text written in a text field could be deleted by using the clear() method.

**7. Can WebDriver test Mobile applications?**  
**Answer:** WebDriver cannot test Mobile applications. WebDriver is a web-based testing tool, therefore applications on the mobile browsers can be tested.

**8. What is the FirefoxDriver, class or an interface? And which interface does it implement?**  
**Answer**: FirefoxDriver is a Java class. It implements all the methods available in the interface.

**9. Explain how you can capture server-side log Selenium Server?**  
**Answer**: To capture server-side log in Selenium Server, you can use the command: java –jar .jar –log selenium.log

**10. How to verify if the checkbox/radio is checked or not?**  
**Answer:** is Selected() method is used to verify if the checkbox/radio is checked or not.

**11. What is the difference between findElement () and find elements ()?**  
**Answer**: Both of them let the user find elements in the current web page matching the specified locator value. But if you use find Element(), only the first matching element would be fetched. ([Salesforce Training Online](https://svrtechnologies.com/selenium-training/))

**12. Can we use Selenium RC for tests driving on two different browsers on one operating system without Selenium Grid?**  
**Answer**: We can do it if the JAVA testing framework is not used. If we use Java client driver of Selenium, instead of using a Java testing framework, TestNG allows us not to use Selenium Grid.

**13. What is IntelliJ?**  
**Answer**: IntelliJ is an IDE that helps users to write code for Selenium better and faster. It could be used as an option to Java bean and Eclipse.

**14. How can I read test data from excels?**  
**Answer:**Test data can efficiently be read from excel using JXL or POI API. See the detailed tutorial here.

**15. How can we handle Web-based pop-up?**  
**Answe**r: There are four methods of effective Web-based pop-up handling:  
string getText() method returns the text displayed on the alert box  
void accept() method clicks on the “Ok” button as soon as the pop-up window appears  
void dismiss() method clicks on the “Cancel” button as soon as the pop-up window appears  
void sendKeys(String stringToSend) method enters the specified string pattern into the alert box.

**16. Which web driver implementation is the fastest?**  
**Answer:**The fastest Web Driver is Html Unit Driver. Differing of other drivers (Firefox Driver, Chrome Driver, etc), it’s non-GUI, while running no browser gets launched.

**17. How would you test your element locator?**  
**Answer**: “Find Button” of Selenium IDE is used to test the locator. Clicking on this button, you will see on-screen if your element locator is right or wrong.  
Also, you can use the “FirePath” plugin in Firefox. ([Company](https://www.seleniumhq.org/))

**18. How to set test case priority in TestNG?**  
**Answer:** TestNG “Priority” is used to schedule the test cases. To achieve, we need to add an annotation as @Test(priority=??). The default value will be zero for priority. If you don’t mention the priority, it will take all the test cases as “priority=0” and execute.  
The example below shows the usage of the priority for test cases.  
As we have not defined the priority for test case “Registration”, it will get executed first and then the other test cases based on priority.

**19. How do you send ENTER/TAB keys in WebDriver?**  
**Answer:** use click() or submit() methods are used for ENTER. But, don’t forget that the submit() method is used only if type=’ submit’.  
You can use Actions class act.sendKeys(Keys.ENTER) for TAB.

**20. How could AJAX controls be handled in WebDriver?**  
**Answer:**AJAX allows the Web page to retrieve small amounts of data from the server without reloading the entire page.  
**The different wait methods should be applied for testing Ajax application:**

Thread Sleep

Implicit Wait

Explicit Wait

WebdriverWait

Fluent Wait

**21. Could you explain the line of code Webdriver driver = new FirefoxDriver()?**  
**Answer:**‘WebDriver’ is an interface and we are creating an object of type WebDriver instantiating an object of FirefoxDriver class.

**22. What is the alternate way to click on the login button?**  
**Answer:**submit() method could be used as the alternate way to click on the login button, but only if attribute type=submit.

**23. What is the testng.xml file used for?**  
**Answer:**testng.xml file is used to configure the whole test suite. Here we can create a test suite, create test groups, mark tests for parallel execution, add listeners and pass parameters to test scripts. It can be used for the further test suite triggering. ([Selenium Online Training](https://svrtechnologies.com/selenium-training/best-selenium-online-training))

**24. What is an Exception?**  
**Answer:**Exceptions are events due to which java program ends abruptly without giving expected output. Java provides a framework where a user can handle exceptions.

The process of handling Exceptions is called Exception Handling.  
Exceptions need to be handled because they break the normal flow of execution of a program.

One of the important intentions of exception handling is to prevent this break and continue program execution. Sometimes, you might want to perform some series of actions on occurring of a certain exception.  
When an exception occurs, an exception object is created which is technically referred to as ‘Throwing an Exception’ and we add Try/Catch blocks like,  
try {  
// Protected code  
catch (ExceptionName e) {  
// Catch block  
1. The piece of code that might throw an exception is added inside the Try block.  
2. The Catch statement catches the exception and takes it as a parameter.  
3. When no exception is thrown, the try statement is executed and not the catch statement.  
**Example**: When the selenium script fails due to the wrong locator, then the developer should be able to understand the reason for failure and this can be achieved easily if the exception is handled properly in the program.  
In my experience, it is best to try to avoid WebDriver exceptions whenever possible and catch truly exceptional cases. Use try/catch to handle things that go wrong and are outside my control.

**25. Do you know a way to refresh the browser by using Selenium?**  
**Answer:**

**The list of commands to refresh a page in Selenium:**

navigate().refresh()

getCurrentUrl()

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

sendKeys(Keys.F5)

**26. What are the limitations of Selenium?  
Answer:**

**Following are the limitations of Selenium:**

Selenium supports testing of only web-based applications

Mobile applications cannot be tested using Selenium

Captcha and Barcode readers cannot be tested using Selenium

Reports can only be generated using third-party tools like TestNG or JUnit.

As Selenium is a free tool, thus there is no ready vendor support

the user can find numerous helping communities.

The user is expected to possess prior programming language knowledge.

**27. How will you use Selenium to upload a file?**  
**Answer:**  
File uploading action could be performed by using elements.sendKeys(“path of file”) on the web element of input tag and type file: < name=”fileUpload” type=”file” ([Interview Questions And Answers](https://svrtechnologies.com/interview-questions-and-answers/))

**28. What methods of Robot Class do you know?**  
**Answer:**Some commonly and popularly used methods of Robot Class during web automation

**29. What is the purpose of deselecting all() method?**  
**Answer:**It is used to deselect all the options which have been selected from the drop-down list.

**30. What is the difference between @Factory and @Data Provider annotation?**  
**Answer:**@DataProvider is concerned with individual test methods and run the specific methods for many times. @Factory method creates test class instances and runs all the test methods in that class with different data. sets.

**31. What is the purpose of the getOptions() method?**  
**Answer:**getOptions() is used to get the selected option from the drop-down list.

**36. How can you handle network latency using Selenium?**  
**Answer:**  
You can use driver.manage().timeouts().pageLoadTimeout(); for network latency.

**37. What is the difference between driver.close() and driver.quit command?**  
**Answer:**Object repository is an essential entity in any UI automation 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.

**38. What are the main advantages of Selenium?**  
**Answer:**Selenium is free and open-source. You don’t need to spend any licensing cost to use it.  
Cross Browser compatibility (Firefox, Chrome, Internet Explorer, Safari, etc.)  
Multiple programming languages (Java, C#, Ruby, Python, Pearl, etc.) support  
Compatibility with the main platform (Windows, Mac OS, Linux, etc.)

**39. How can we find the value of different attributes like name, class, the value of an element?**  
**Answer:**Using getAttribute(“{attributeName}”) method we can find the value of different attrbutes of an element e.g.-  
String valueAttribute =  
driver.findElement(By.id(“elementLocator”)).getAttribute(“value”);

**40. How to delete cookies in selenium?**  
**Answer:**Using deleteAllCookies() method-  
driver.manage().deleteAllCookies();

**41. What is an implicit wait in selenium?**  
**Answer:**An implicit wait is a type of wait that waits for a specified time while locating an element before throwing NoSuchElementException. As by default selenium tries to find elements immediately when required without any wait. So, it is good to use implicit wait. This wait is applied to all the elements of the current driver instance.

**42. What is an explicit wait in selenium?**  
**Answer:**An explicit wait is a type of wait that is applied to a particular web element until the expected condition specified is met.

WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id(“elementId”);

**43. What are some expected conditions that can be used in Explicit waits?**  
**Answer:**Some of the commonly used expected conditions of an element that can be used with expicit waits are-  
elementToBeClickable(WebElement element or By locator)  
stalenessOf(WebElement element)  
visibilityOf(WebElement element)  
visibilityOfElementLocated(By locator)  
invisibilityOfElementLocated(By locator)  
attributeContains(WebElement element, String attribute, String value)  
alertIsPresent()  
titleContains(String title)  
titleIs(String title)  
textToBePresentInElementLocated(By, String)

**44. What is fluent wait in selenium?**  
**Answer:**A fluent wait is a type of wait in which we can also specify polling interval(intervals after which driver will try to find the element) along with the maximum timeout value.  
Wait wait = new FluentWait(driver)  
.withTimeout(20, SECONDS)  
.pollingEvery(5, SECONDS)  
.ignoring(NoSuchElementException.class);  
WebElement textBox = wait.until(new Function() {  
public WebElement apply(WebDriver driver) {  
return driver.findElement(By.id(“textBoxId”));

**45. How to mouse hover an element in selenium?**  
**Answer:**Code to mouse hover over an element in selenium-  
Actions action = new Actions(driver);  
Web Element element=driver. find Element(By.id(“element Id”));  
action. move To Element(element).perform();

**46. What are DesiredCapabilities in selenium web driver?**  
**Answer:**Desired capabilities are a set of key-value pairs that are used for storing or configuring browser-specific properties like its version, platform, etc in the browser instances.

**47. What are some commonly encountered exceptions in selenium?**  
**Answer:**Some of the commonly seen exceptions in selenium are-  
NoSuchElementException – When no element could be located from the locator provided.  
ElementNotVisibleException – When an element is present in the dom but is not visible.  
NoAlertPresentException – When we try to switch to an alert but the targetted alert is not present.  
NoSuchFrameException – When we try to switch to a frame but the targetted frame is not present.  
NoSuchWindowException – When we try to switch to a window but the targetted window is not present.  
UnexpectedAlertPresentException – When an unexpected alert blocks the normal interaction of the driver.  
TimeoutException – When a command execution gets a timeout.  
InvalidElementStateException – When the state of an element is not appropriate for the desired action.  
NoSuchAttributeException – When we are trying to fetch an attribute’s value but the attribute is not correct  
WebDriverException – When there is some issue with driver instance preventing it from getting launched.

**48. How to check which option in the dropdown is selected?**  
**Answer:**Using the isSelected() method we can check the state of a dropdown’s option.  
Select countriesDropDown = new Select(driver.findElement(By.id(“countries”)));  
dropdown.selectByVisibleText(“India”);  
//returns true or false value  
System.out.println(driver.findElement(By.id(“India”)).isSelected());

**49. How to locate a link using its text in selenium?**  
**Answer:**  
Using link text() and partial link text() we can locate a link. The difference between the two is link text matches the complete string passed as a parameter to the link texts. Whereas partial link text matches the string parameter partially with the link texts.  
WebElement link1 = driver.findElement(By.linkText(“artOfTesting”));  
WebElement link2 = driver.findElement(By.partialLinkText(“artOf”));

**50. What are the benefits of Automation Testing?**  
**Answer:**

**Benefits of Automation testing are:**

Supports execution of repeated test cases

Aids in testing a large test matrix

Enables parallel execution

Encourages unattended execution

Improves accuracy thereby reducing human-generated errors

Saves time and money