**1.What do we mean by Selenium 1, Selenium 2 and Selenium 3?**

**Answer:** Earlier Selenium 1 used to be the major project of Selenium.

Selenium 1 = Selenium IDE + Selenium RC + Selenium Grid

Later Selenium Team has decided to merge both Selenium WebDriver and Selenium RC to form a more powerful Selenium tool.

They both got merged to form “Selenium 2”

“Selenium WebDriver” was the core of “Selenium 2” and “Selenium RC” used to run in maintenance mode.

Hence Selenium 2 = Selenium IDE + Selenium WebDriver 2.x + Selenium Grid.

“Selenium 2” released on July 8, 2011.

Selenium team has decided to completely remove the dependency for Selenium RC.

After 5 years, “Selenium 3" was released on October 13, 2016 with a major change, which is the original Selenium Core implementation and replacing it with one backed by WebDriver and lot more improvements.

Hence Selenium 3 = Selenium IDE + Selenium WebDriver 3.x + Selenium Grid.

After 3 years from it’s a major release, now Selenium has put out its first alpha version of Selenium 4 on Apr 24, 2019. Still, there is no official announcement about the release date of Selenium 4, but we are expecting it around October 2019. Till that there can be several alpha or beta versions released time to time with stabilization.

**2. When should I use Selenium Grid?**

**Answer:**Selenium Grid is a part of the Selenium Suite that specializes in running multiple tests across different browsers, operating systems, and machines in parallel.

You should use Selenium Grid when you want to do either one or both of following:

1.For Parellel Testing - Run your tests against different browsers, operating systems, and machines all at the same time. This will ensure that the application you are Testing is fully compatible with a wide range of browser-O.S combinations.

2. Save time in the execution of your test suites. If you set up Selenium Grid to run, say, 4 tests at a time, then you would be able to finish the whole suite around 4 times faster.

**3. What are the different types of drivers available in WebDriver?**

**Answer:** Drivers are required for different types of browsers.

Here is the list of different types of drivers available in Selenium WebDriver.

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

**4. Is WebDriver a class or interface?**

**Answer:** WebDriver is an Interface,and we are defining a reference variable(driver) whose type is an interface. Now any object we assign to it must be an instance of a class(FireFoxDriver) that implements the interface. Whenever we use Selenium for automation ,the first line of our Program starts with the code to invoke Fire Fox Driver

WebDriver driver = new FireFoxDriver();

**5. What is the super interface of WebDriver ?**

**Answer:** SearchContext is the super most interface in selenium, which is extended by another interface called WebDriver. All the abstract methods of SearchContext and WebDriver interfaces are implemented in RemoteWebDriver class.

**6. Is FirefoxDriver a class or interface?**

**Answer:** FirefoxDriver is a class that has been written specifically for the Firefox browser. It has methods that are implemented and it can be instantiated. It can perform all functions (or methods) on the Firefox browser as defined in the interface WebDriver.

**7. Why do we create a reference variable ‘driver’ of type WebDriver and what is the purpose of its creation?**

**Answer:**  Having a reference variable of type WebDriver allows us to assign the driver object to different browser specific drivers. Thus allowing multi-browser testing by assigning the driver object to any of the desired browser.

**8. Explain the line of code WebDriver driver = new FirefoxDriver();?**

**Answer:**WebDriver driver = new FirefoxDriver();

We can create Object of a class FirefoxDriver by taking reference of an interface (WebDriver). In this case, we can call implemented methods of WebDriver interface.

As per the above statement, we are creating an instance of the WebDriver interface and casting it to FirefoxDriver Class. All other Browser Drivers like ChromeDriver, InternetExplorerDriver, PhantomJSDriver, SafariDriver etc implemented the WebDriver interface (actually the RemoteWebDriver class implements WebDriver Interface and the Browser Drivers extends RemoteWebDriver). Based on this statement, you can assign Firefox driver and run the script in Firefox browser (any browser depends on your choice).

**9. What are the different types of navigation commands in WebDriver?**

**Answer:**  Browser Navigation Commands:

WebDriver provides some basic Browser Navigation Commands that allows the browser to move backwards or forwards in the browser's history.

**1. Navigate To:**

Method - String to(String arg0)

In WebDriver, this method loads a new web page in the existing browser window. It accepts String as parameter and returns void. The respective command to load/navigate a new web page can be written as:

Example:

driver.navigate().to("www.automationtestinginsider.com");

**2. Backward:**

Method - void back()

This method enables the web browser to click on the back button in the existing browser window. It neither accepts anything nor returns anything. The respective command that takes you back by one page on the browser's history can be written as:

Example:

driver.navigate().back();

**3. Forward:**

Method - void forward()

This method enables the web browser to click on the forward button in the existing browser window. It neither accepts anything nor returns anything. The respective command that takes you forward by one page on the browser's history can be written as:

Example:

driver.navigate().forward();

**4. Refresh:**

Method - void refresh()

In WebDriver, this method refresh/reloads the current web page in the existing browser window. It neither accepts anything nor returns anything. The respective command that takes you back by one page on the browser's history can be written as:

Example:

driver.navigate().refresh();

Complete article:  https://www.automationtestinginsider.com/2019/10/webdriver-commands.html

**10. What are the different types of waits available in WebDriver?**

**Answer:** Please refer complete article here

<https://www.automationtestinginsider.com/2020/02/waits-in-selenium-webdriver.html>

**11. What is the difference between driver.close() and driver.quit() commands?**

**Answer:**driver. quit() is used to exit the browser, end the session, tabs, pop-ups etc. But the when you driver. close(), only the window that has focus is closed.

**12. Can Selenium Automate Desktop Applications?**

**Answer:**  Selenium does not have the capability to automate the desktop applications.It cannot recognize the objects in a desktop application. Selenium drives the testing using the driver object that identifies the elements on screen using id, cssselector, xpath etc. which are not present in a desktop app.

**13. What is the difference between Assert and Verify commands?**

**Answer:** In case of the “Assert” command, as soon as the validation fails the execution of that particular test method is stopped and the test method is marked as failed. Whereas, in case of “Verify”, the test method continues execution even after the failure of an assertion statement.

Assert

We use Assert when we have to validate critical functionality, failing of which makes the execution of further statements irrelevant. Hence, the test method is aborted as soon as failure occurs.

**@Test**

**public** **void** **assertionTest**(){

//Assertion Passing

Assert.assertTrue(**1**+**2** == **3**);

System.out.println("Passing 1");

//Assertion failing

Assert.fail("Failing second assertion");

System.out.println("Failing 2");

}

**Output-**

Passing **1**

**FAILED:** assertionTest

java.lang.AssertionError: Failing second assertion

**Verify**

At times, we might require the test method to continue execution even after the failure of the assertion statements. In TestNG, Verify is implemented using SoftAssert class.

In case of SoftAssert, all the statements in the test method are executed (including multiple assertions).

**@Test**

**public** **void** **softAssertionTest**(){

//Creating softAssert object

SoftAssert softAssert = **new** SoftAssert();

//Assertion failing

softAssert.fail("Failing first assertion");

System.out.println("Failing 1");

//Assertion failing

softAssert.fail("Failing second assertion");

System.out.println("Failing 2");

//Collates the assertion results and marks test as pass or fail

softAssert.assertAll();

}

**Output-**

Failing **1**

Failing **2**

**FAILED:** softAssertionTest

java.lang.AssertionError: The following asserts **failed:**

Failing first assertion,

Failing second assertion

Here, we can see that even though both the test methods are bound to fail, still the test continues to execute.

**14. Can Captcha be automated using Selenium?**

**Answer:** No. CAPTCHA can be automated if you are able to decode the image using OCR (Optical Character Recognition). For that, one will need to write complex algorithm to sort out the image pattern and has to be an expert in image pattern mapping as well. But images with the passage of time have become progressively more unreadable, therefore reducing the very chances of CAPTCHA automation.

**15. What is Object Repository and how can we create an Object Repository in Selenium?**

**Answer:**  An object repository is a common storage location for all objects. In Selenium WebDriver context, objects would typically be the locators used to uniquely identify web elements.

The major advantage of using object repository is the segregation of objects from test cases. If the locator value of one webelement changes, only the object repository needs to be changed rather than making changes in all test cases in which the locator has been used. Maintaining an object repository increases the modularity of framework implementation.

An object repository is a common storage location for all objects. In Selenium WebDriver context, objects would typically be the locators used to uniquely identify web elements.

The major advantage of using object repository is the segregation of objects from test cases. If the locator value of one webelement changes, only the object repository needs to be changed rather than making changes in all test cases in which the locator has been used. Maintaining an object repository increases the modularity of framework implementation.

A property file stores information in a Key-Value pair. Key value pair is represented by two string values separated by the equal to sign.

There are two ways to create object repository in Selenium:

1. any companies also use Page Object Model to store all locators which also make the test in a readable format. Depends on your current framework style you can adopt any of these. Personally, I use Page Object Model using PageFactory which make my test more readable and in terms of performance as well.

2. Using Properties file, Ex: ObjectRepo.peroperties

**16. What are the types of WebDriver API’s that are supported/available in Selenium?**

**Answer:** Please refer answer of question#108

**17. Which WebDriver implementation claims to be the fastest?**

**Answer:** The fastest implementation of WebDriver is the HTMLUnitDriver. It is because the HTMLUnitDriver does not execute tests in the browser and also called as a Headless browser.

**18. What is the difference between Soft Assert and Hard Assert in Selenium?**

**Answer:** Please refer answer of question#13

**19. What are the verification points available in Selenium?**

**Answer:**  Here some examples of verification

To check if element is enabled or not

driver.findElement(By.id("<ID>")).isEnabled()

To check if element is displayed or not

driver.findElement(By.id("<ID>")).isDisplayed()

To check if element is selected or not

driver.findElement(By.id("<ID>")).isSelected()

To check if element is currently active or not on the page

driver.findElement(By.id("<ID>")).equals(driver.switchTo().activeElement());

**20. Is Selenium Server needed to run Selenium WebDriver scripts?**

**Answer:** In case of Selenium WebDriver, it does not require to start Selenium Server for executing test scripts. Selenium WebDriver makes the calls between browser & automation script. Selenium WebDriver has native support for each browser to supports Test Automation; on the same machine (both Selenium WebDriver Automation tests & browsers are on same machine.

**21. What are the different ways for refreshing the page using Selenium WebDriver?**

**Answer:** 1) **Refresh command:**The most commonly used and simple command for refreshing a webpage.

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

driver.navigate().refresh();

2) **SendKeys command:** Second most commonly used command for refreshing a webpage. As it is using a send keys method, we must use this on any Text box on a webpage.

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

// Element "q" is a Seach Text box on gogle website

driver.findElement(By.name("q")).sendKeys(Keys.F5);

3) **Get command:** This is a tricky one, as it is using another command as an argument to it. If you look carefully, it is just feeding get command with a page URL.

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

driver.get(driver.getCurrentUrl());

4) **To command:** This command is again using the same above concept. navigate( ).to( ) is feeding with a page URL and an argument.

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

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

5) **SendKeys command**: This is the same SendKeys command but instead of using Key, it is using ASCII code.

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

driver.findElement(By.name("s")).sendKeys("\uE035");

**22. What is the difference between driver.getWindowHandle() and driver.getWinowHandles() in Selenium WebDriver and their return type?**

**Answer:** The main difference between driver.getWindowHandle() and driver.getWindowHandles() are :-

driver.getWindowHandle() is used to handle single window i.e. main window and driver.getWindowHandles() is used to handle multiple windows.

driver.getWindowHandle() return type is string and driver.getWindowHandles() return type is Set<string>.

**23. What is JavascriptExecutor and in which case JavascriptExecutor will help in Selenium automation?**

**Answer:**  JavaScriptExecutor is used when Selenium Webdriver fails to click on any element due to some issue. JavaScriptExecutor provides two methods "executescript" & "executeAsyncScript" to handle. Executed the JavaScript using Selenium Webdriver.

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

**24. List some scenarios which we cannot automate using Selenium WebDriver?**

**Answer:**  **1.Pop-up Windows**

When a simple, prompt, or confirmation alert pops-up, it can be difficult to automate it to either accept or close.

**2.Dynamic Content**

A web page that has dynamically loaded content has elements that may not be at first visible when you visit the site.

**3.Flakiness**

Sometimes Selenium will give you flaky tests, which means they’ll come back as positive or negative when they actually are the opposite.

**4. Mobile Testing**

While Selenium WebDriver can test on any operating system and browser on desktops, it’s limited when it comes to mobile testing in that it cannot run on native operating systems like iOS and Android.

**5. Limited Reporting**

While Selenium will exponentially increase your automated testing capabilities, because it’s an open source tool it is limited and features and does not support much reporting on its own.

**25. How can use Recovery Scenario in Selenium WebDriver?**

**Answer:** 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.

**26. Have you used any cross browser testing tool to run Selenium Scripts on cloud?**

**Answer:** We can perform cross browser using TestNG framework.

Some other popular tools for cross browser testing are:

**1) LambdaTest**

LambdaTest is a cloud based platform that helps you perform cross browser compatibility testing of your web app or websites. You can run automated selenium scripts on LambdaTest's scalable cloud grid, or can even perform live interactive testing on real browser environments.

Key Features:

Run Selenium automation tests on a scalable Selenium grid having 2000+ browser environments

Execute automated screenshot and responsive testing of your website

Test your locally or privately hosted website using SSH Tunnel

One click bug logging to your favorite bug tracking tool like Asana, BitBucket, GitHub, JIRA, Microsoft VSTS, Slack, Trello etc.

**2) CrossBrowser Testing** - Crossbrowser testing tool has wide range of different browsers and their versions. It is available for multiple OS. It supports over 1000 combinations of different browsers and O.S including mobile browsers.

**3) Browser-Stack-** With browser stack it is possible to do web based browser testing on desktop and mobile browser. It is cloud based and so it does not require any installation, and the pre-installed developer tools are useful for quick cross-browser testing and debugging. With browser-stack you can set up a comprehensive testing environment with support for proxies, firewalls and Active Directory. It supports opera mobile, Android, Windows (XP, 7 and 8), iOS, OSX snow leopard, lion and mountain lion and so on. Browser stack allows you to test your pages remotely.

**4) Sauce Labs-** It is the leading cloud based web and mobile app testing platform. It allows you to run tests in the cloud on more than 260 different browser platform and devices. There is no VM set up or maintenance required.

**27. What are the DesiredCapabitlies in Selenium WebDriver and their use?**

**Answer:** Desired Capabilities is a class used to declare a set of basic requirements such as combinations of browsers, operating systems, browser versions, etc. to perform automated cross browser testing of a web application.

When we try to automate our test scripts through Selenium automation testing, we need to consider these combinations to declare a specific test environment over which our website or web application should render seamlessly.

Desired Capabilities class is a component of the org.openqa.selenium.remote.DesiredCapabilities package. It helps Selenium WebDriver set the properties for the browsers. So using different capabilities from Desired Capabilities class we can set the properties of browsers. For example, the name of the browser, the version of the browser, etc. We use these capabilities as key-value pairs to set them for browsers.

**28. 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?'**

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

**29. List out the test types that are supported by Selenium?**

**Answer:** Using Selenium type of testing can be done are:

* Functional Testing.
* Regression Testing.
* Sanity Testing.
* Smoke Testing.
* Responsive Testing.
* Cross Browser Testing.
* UI testing (black box)
* Integration Testing.

**30. Explain what is assertion in Selenium and what are the different types of assertions?**

**Answer:** Please refer answer of question#13

**31. List out the technical challenges with Selenium?**

**Answer:** The Most Common Selenium Challenges

Pop-up Windows. When a simple, prompt, or confirmation alert pops-up, it can be difficult to automate it to either accept or close. ...

* Dynamic Content.
* Flakiness.
* Mobile Testing.
* Limited Reporting.
* Multi-tab Testing.
* Manual Testing.
* Scalability.

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

**Answer:** 1. The sendKeys" command does not replace the existing text content in the text box whereas the "type" command replaces the existing text content of the text box. 2. It will send explicit key events like a user pressing a key on the keyboard.

**33. While using click() command, can you use screen coordinates?**

**Answer:**To click on specific part of element, you would need to use clickAT command.  ClickAt command accepts element locator and x, y coordinates as arguments- clickAt (locator, cordString)

**34. What are the four parameters you have to pass in Selenium?**

**Answer:** In total, there are four conditions (parameters) for Selenium to pass a test. These are as follows: URL, host, browser and port number.

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

**Answer:**Both sleep() and setSpeed() are used to delay the speed of execution. The main difference between them is that: setSpeed sets a speed that will apply a delay time before every Selenium operation. ... sleep() will set up wait only for once when called.Both sleep() and setSpeed() are used to delay the speed of execution. The main difference between them is that: setSpeed sets a speed that will apply a delay time before every Selenium operation. ... sleep() will set up wait only for once when called.

**36. What is heightened privileged browsers?**

**Answer:**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 launched 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.

**37. Which attribute you should consider throughout the script in frame for (if no frame id as well as no frame name”)?**

**Answer:** 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.

**38. List the advantages of Selenium WebDriver over Selenium Server?**

**Answer:** WebDriver is faster than Selenium RC because of its simpler architecture. WebDriver directly talks to the browser while Selenium RC needs the help of the RC Server in order to do so. WebDriver's API is more concise than Selenium RC's. WebDriver can support HtmlUnit while Selenium RC cannot.

**39. What are the challenges in handling AJAX calls in Selenium WebDriver?**

**Answer:** Please refer below article on waits

<https://www.automationtestinginsider.com/2020/02/waits-in-selenium-webdriver.html>

**40. What is IntelliJ and how it is different from Eclipse IDE?**

**Answer:**

IntelliJ IDEA is the most powerful, popular, and fully-featured IDE for Java Developers, which was released for the public in 2001. It is developed and maintained by Jet Brains Company. It is licensed by Apache 2.0.

**Eclipse**

Eclipse is an open-source IDE for developing applications using the Java, Python, Ruby, C, C++, etc. The IBM released it in 2001 under the Eclipse Public License (EPL). It became popular soon for developing free and commercial projects. Today, it became the most popular Java IDE.

System Requirements

We can install IntelliJ Idea on Windows, macOS and Linux with the following hardware:

2 GB RAM minimum, 4 GB RAM recommended

1.5 GB hard disk space + at least 1 MB for caches

1024×768 minimum screen resolution

We can run Eclipse IDE on any platform that supports JVM including Windows, macOS, Linux and Solaris. It demands the following hardware:

0.5 GB RAM minimum, 1+ GB RAM recommended

300 MB hard disk space minimum, 1+ GB recommended

Processor speed of 800 MHz minimum, 1.5 GHz or faster recommended.

**41. What are Selenium WebDriver Listeners?**

**Answer:** Listeners “listen” to the event defined in the selenium script and behave accordingly. The main purpose of using listeners is to create logs and reports. There are many types of listeners such as WebDriver Event Listeners and TestNG Listeners.

We need to know the following class and interface when we talk about listeners in Selenium.

WebDriverEventListener: This WebDriver Event Listener interface allows us to implement the methods

Once the script is executed, Selenium WebDriver does perform activities such as Type, Click, Navigate etc., To keep track of these activities we use WebDriver Event Listeners interface.

EventFiringWebDriver: This EventFiringWebDriver class actually fire WebDriver event

Lets see how to implement Listeners in Selenium WebDriver Script.

**42. What are the different types of Listeners in TestNG?**

**Answer:** What is Listeners in TestNG?

Listener is defined as interface that modifies the default TestNG's behavior. As the name suggests Listeners "listen" to the event defined in the selenium script and behave accordingly. It is used in selenium by implementing Listeners Interface. It allows customizing TestNG reports or logs. There are many types of TestNG listeners available.

There are many types of listeners which allows you to change the TestNG's behavior.

Below are the few TestNG listeners:

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

ITestListener is one of the most important listener. ITestListener has following methods

OnStart- OnStart method is called when any Test starts.

onTestSuccess- onTestSuccess method is called on the success of any Test.

onTestFailure- onTestFailure method is called on the failure of any Test.

onTestSkipped- onTestSkipped method is called on skipped of any Test.

onTestFailedButWithinSuccessPercentage- method is called each time Test fails but is within success percentage.

onFinish- onFinish method is called after all Tests are executed.

**43. What is the API that is required for implementing Database Testing using Selenium WebDriver?**

**Answer:**  Selenium WebDriver alone is ineligible to perform database testing but this can be done using Java Database Connectivity API (JDBC). The API lets the user connect and interact with the data source and fetch the data with the help of automated queries.

**44. When to use AutoIt?**

**Answer:** AutoIt v3 is also freeware. It uses a combination of mouse movement, keystrokes and window control manipulation to automate a task which is not possible by selenium webdriver.

Why Use AutoIt?

Selenium is an open source tool that is designed to automate web-based applications on different browsers but to handle window GUI and non HTML popups in application. AutoIT is required as these window based activity are not handled by Selenium.

**45. Why do we need Session Handling while using Selenium WebDriver?**

**Answer:** During test execution, the Selenium WebDriver has to interact with the browser all the time to execute given commands.This can be achieved using Session Handling in Selenium.

**46. What is exception test in Selenium?**

**Answer:** TestNG provides an option of tracing the exception handling of code. You can test whether a code throws a desired exception or not. Here the expectedExceptions parameter is used along with the @Test annotation.

**47. How do you achieve synchronization in WebDriver?**

**Answer:** Please refer below article on waits

https://www.automationtestinginsider.com/2020/02/waits-in-selenium-webdriver.html

**48. Can Bar Code Reader be automated using Selenium?**

**Answer:**Selenium has limitation to automate Bar code but by using third party API we can automate Bar codes. So, ZXing is one the third party API will be used to automate Bar Codes.

**49. What is Robot API? What methods of Robot class do you know?**

**Answer:** As per the class description, this class is used to generate native system input events. This class uses native system events to control the mouse and keyboard.

java.awt.Robot class provides various methods needed for controlling mouse and keyboard.

Following are some of the methods commonly used in browser test automation:

Keyboard methods:

keyPress(int keycode): This method presses a given key. For Example, keyPress(KeyEvent.VK\_SHIFT) method presses ”SHIFT’ key

keyRelease(int keycode): This method releases a given key. For Example, keyRelease(KeyEvent.VK\_SHIFT) method releases ”SHIFT” key

Mouse Methods:

mousePress(int buttons): This method presses one or more mouse buttons.For Example, mousePress(InputEvent.BUTTON1\_DOWN\_MASK) method is used left click mouse button

mouseRelease(int buttons): This method releases one or more mouse buttons. For Example, mouseRelease(InputEvent.BUTTON1\_DOWN\_MASK) method is used to release the left mouse button click

mouseMove(int x, int y): This method moves the mouse pointer to given screen coordinates specified by x and y values. For Example, mouseMove(100, 50) will move the mouse pointer to the x coordinate 100 and y coordinate 50 on the screen.

**50. Which package can be imported while working with WebDriver?**

**Answer:**  org.openqa.selenium

**51. What is the purpose of deselectAll() method?**

**Answer:** deselectAll() Method

deselectAll() method is useful to remove selection from all selected options of select box. It will works with multiple select box when you need to remove all selections. Syntax for deselectAll() method is as bellow.

Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));

listbox.deselectAll();

**52. What is the purpose of getOptions() method?**

**Answer:**getOptions() is used to get all options from the dropdown list. This gets the all options belonging to the Select tag. It takes no parameter and returns List<WebElements>. Sometimes you may like to count the element in the dropdown and multiple select box, so that you can use the loop on Select element. Syntax for get method is:

dropdownElement.getOptions();

Example : Select optionSelect = new Select(driver.findElement(By.id("dropdown\_cities")));

List <WebElement> elementCount = optionSelect.getOptions();

System.out.println(elementCount.size());

**53. What could be the cause for Selenium WebDriver test to fail?**

**Answer:** **1. Asynchronous websites**

Selenium tests are developed keeping a particular order in mind and when there are websites that are built using asynchronous architecture. The order of response cannot be fixed at all times. In such scenarios, sometimes the responses from the website are not in order. As a result, the test case fails. Such failures are tough to debug as well because they are not reproducible at all times.

**2. Dynamic websites** that change without refresh

Now websites are built that do a page reload or refresh rarely and, all the changes happen dynamically using JavaScript on the same page. In some scenarios, if there is a small error in one step, the whole test case fails because there is no opportunity to reload or refresh the page.

**3. Alerts, Pop-ups, Nested IFrames**

Selenium Users faced problems while automating alerts, pop-ups and Nested IFrames related functionalities on a website but with the latest version of Selenium, this problem has been handled quite well.

**4. Selenium WebDriver version mismatch**

Selenium releases WebDrivers for supported browsers to execute tests on them. If the installed browser on the system does not match the versions supported by the web drivers, an exception like this is reported: “This version of ChromeDriver only supports Chrome version 74”.

**5. Selectors move on screen**

Probably the single greatest cause of Selenium failures is when a selector moves or subtly changes on screen.

**54. Can we test APIs or web services using Selenium WebDriver?**

**Answer:** To do API testing, one should only use – Postman/Rest Assured (which is a Java based library which possess inbuilt API methods)/Jmeter & Soap Ui. Selenium is an API for automation of browsers. The API testing can be included in Selenium framework. In postman, code for any API request can be generated.

**55. What are some expected conditions that can be used in Explicit Waits?**

**Answer:**  In order to declare explicit wait, one has to use “ExpectedConditions”. The following Expected Conditions can be used in Explicit Wait.

* alertIsPresent()
* elementSelectionStateToBe()
* elementToBeClickable()
* elementToBeSelected()
* frameToBeAvaliableAndSwitchToIt()
* invisibilityOfTheElementLocated()
* invisibilityOfElementWithText()
* presenceOfAllElementsLocatedBy()
* presenceOfElementLocated()
* textToBePresentInElement()
* textToBePresentInElementLocated()
* textToBePresentInElementValue()
* titleIs()
* titleContains()
* visibilityOf()
* visibilityOfAllElements()
* visibilityOfAllElementsLocatedBy()
* visibilityOfElementLocated()

**Syntax:**

WebDriverWait wait= new WebDriverWait(driver, 5);

        WebElement ele=wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(“xpath")));

**56. What is HtmlUnitDriver?**

**Answer:**  HtmlUnit is a headless web browser written in Java. It allows high-level manipulation of websites from other Java code, including filling and submitting forms and clicking hyperlinks. It also provides access to the structure and the details within received web pages.

A headless browser is a web-browser without a graphical user interface. This program will behave just like a browser but will not show any GUI.

Some of the examples of Headless Drivers include

* HtmlUnit
* Ghost
* PhantomJS
* ZombieJS
* Watir-webdriver

**57. Name an API used for logging in Java?**

**Answer:** util. logging. Logger is the class used to log application messages in java logging API

**58. What is the use of logging in Automation?**

**Answer:** Advantages of Logging in Selenium Scripts:

Grants a complete understanding of test suites execution. Log messages can be stored in external files for post-execution scrutiny. Logs are an exceptional assistant in debugging the program execution issues and failures.

**59. Can Selenium Test an application on Android Browser?**

**Answer:** Selenium should be able to handle Android browser. There is a Selenium Android Driver for running tests in Android browser.

You can use Selendroid or Appium framework to test native apps or web apps in Android browser

**60. Give the example for method overload in Selenium?**

**Answer:** Method overloading example

You want to have your own Listbox class to interact with dropdown lists.

The Listbox class is just a wrapper around the Select class.

You want it to have simpler method names for selecting list options:

public class Listbox

{

Select list;

public Listbox(Select list) {

this.list = list;

 }

public void select(int i) {

 this.list.selectByIndex(i);

}

public void select(String text) {

this.list.selectByVisibleText(text);

}

public void deSelect(int i) {

 this.list.deselectByIndex(i);

}

public void deSelect(String text) {

this.list.deselectByVisibleText(text);

 }

//other methods }

**61. What is WebDriverBackedSelenium?**

**Answer:**  WebDriverBackedSelenium is an implementation of the Selenium-RC API by Selenium Webdriver, which is primarily provided for backwards compatibility. It allows to test existing test suites using the Selenium-RC API by using WebDriver under the covers

WebDriver Backed Selenium is used for migrating Selenium 1.0(Selenium RC) to Selenium WebDriver tests.

**62. What is the use of contextClick()?**

**Answer:** This method is from Actions class.

ContextClick Method. Right-clicks the mouse at the last known mouse coordinates. Right-clicks the mouse on the specified element.

Please refer below article:

<https://www.automationtestinginsider.com/2020/01/perform-double-click-and-right-click.html>

**63. How do you accommodate project specific method in your framework?**

**Answer:**1st go through all the manual test cases and identify the steps which are repeating. Note down such steps and make them as methods and write into ProjectSpecificLibrary.

**64. What is Actions class in WebDriver and its methods?**

**Answer:**Handling special keyboard and mouse events are done using the Advanced User Interactions API. It contains the Actions and the Action classes that are needed when executing these events.

Some Examples: Mouse double click, Drag and Drop, Handling tooltip, Mouse hover, entering uppercase letters in the textbook using Shift+Letters .

Actions Class: Actions class is an API for performing complex user web interactions like double-click, right-click, etc. and it is the only choice for emulating Keyboard and Mouse interactions.

Please refer below article:

https://www.automationtestinginsider.com/2020/01/actions-class-in-selenium.html

**65. What are different versions of Selenium available you have used and what are the additional features you have seen from the previous versions?**

**Answer:**

|  |  |  |
| --- | --- | --- |
| Version | Version | Comparison |
| Selenium 1 | Selenium RC | Essentially the same thing.  Selenium 1 has never been an official name, but is commonly used in order to distinguish between versions. |
| Selenium 2 | Selenium WebDriver | Essentially the same thing.  The term "Selenium WebDriver" is now more commonly used. |
| Selenium RC | Selenium WebDriver | Selenium RC is the predecessor of Selenium WebDriver.  It has been deprecated and now released inside Selenium WebDriver for backward compatibility. |
| Selenium IDE | Selenium RC/WebDriver | Selenium IDE is a recording tool for automating Firefox, with the ability to generate simple RC/WebDriver code.  Selenium RC/WebDriver are frameworks to automate browsers programmatically. |
| Selenium Grid | Selenium WebDriver | Selenium Grid is a tool to execute Selenium tests in parallel on different machines.  Selenium WebDriver is the core library to drive web browsers on a single machine. |

**66. What are the challenges have you faced with Selenium and how did you overcome them?**

**Answer:**  Some of the challenges given below:

* We cannot test windows application.
* We cannot test mobile apps.
* Limited reporting.
* Handling dynamic Elements.
* Handling page load.
* Handling pop up windows.
* Handling captcha.

**67. Which of the WebDriver APIs is the fastest and why?**

**Answer:**Selenium is a GUI automated testing tool therefore execution speed depends on how fast a particular browser can respond to action events. PhantomJS or HtmlUnit for that matter would be the fastest in terms of execution speed as both are headless browsers.

**68. Give different examples for method overloading and overriding in Selenium Project?**

**Answer:**  Please refer last section in below article

<https://www.automationtestinginsider.com/2020/02/selenium-data-driven-framework-with-pom.html>

**69. How do you debug your automation code when it is not working as expected?**

**Answer:** Please refer below article

http://total-qa.com/selenium-webdriver-debugging-code-breakpoints-java-debugging-eclipse/

**70. What are the end methods you use for verifying whether the end result is achieved by our Selenium automation scripts?**

**Answer:**Answer will be provided soon

**71. What is the use of property file in Selenium?**

**Answer:** properties' files are mainly used in Java programs to maintain project configuration data, database config or project settings etc. Each parameter in properties file are stored as a pair of strings, in key and value format, where each key is on one line.

It can also be used as object repository.

**72. How will you ensure that the page has been loaded completely?**

**Answer:**  Example for using the above specified JavaScript code in Selenium for ensuring the completeness of page loading: The below reusable user defined method can be called anytime you want to check the completeness of page loading:

public void waitForPageLoad(WebDriver driver, int timeout) {

ExpectedCondition<Boolean> pageLoadCondition = new ExpectedCondition<Boolean>() {

public Boolean apply(WebDriver driver) {

return ((JavascriptExecutor)driver).executeScript(“return document.readyState”).equals(“complete”)}

};

WebDriverWait wait = new WebDriverWait(driver, timeout);

wait.until(pageLoadCondition);

}

We can call the above method using the following method calling statement:

waitForPageLoad(driver, 30); //waits till page load gets completed by setting a time limit of 30 seconds

**73. Class.forName(X). What will you write in place of X?**

**Answer:**Class.forName("X") loads the class if it not already loaded. The JVM keeps track of all the classes that have been previously loaded. This method uses the classloader of the class that invokes it. The "X" is the fully qualified name of the desired class.

**74. Explain about typecasting?**

**Answer:**Type casting in Java is to cast one type, a class or interface, into another type i.e. another class or interface. Since Java is an Object oriented programming language and supports both Inheritance and Polymorphism, It’s easy that Super class reference variable is pointing to SubClass object but the catch here is that there is no way for Java compiler to know that a Superclass variable is pointing to SubClass object. Which means you can not call a method which is declared in the subclass. In order to do that, you first need to cast the Object back into its original type. This is called type casting in Java.

**75. What is the default timeout of Selenium WebDriver?**

**Answer:**The default WebDriver setting for timeouts is never. WebDriver will sit there forever waiting for the page to load. Apparently there is a timeout. It is 30 minutes long.

**76. What is the difference between isDisplayed() and isEnabled() functions in Selenium WebDriver?**

**Answer:**  isDisplayed() is capable to check for the presence of all kinds of web elements available. isEnabled() is the method used to verify if the web element is enabled or disabled within the webpage. isEnabled() is primarily used with buttons.

**77. Can you test flash images in Selenium?**

**Answer:** Yes

Why flash object capturing is difficult? How is it resolved?

Flash is an outdated technology. It is difficult to capture a flash object as it is different from HTML. Also, Flash is an embedded SWF file (Small Web Format). It is also difficult to access Flash object on a mobile device.

 You use to write a script using any automation tool like Selenium, SoapUI, TestComplete, etc. and execute the script.

**Difference between the Flash and other element**.

As mentioned above, the main difference between flash and other element is that Flash is embedded in SWF files, while other elements are embedded in HTML files. That's why HTML is easy to capture compared to flash.

Below are the requirements in order to test the flash application

Flash Application.

Support web browser.

Adobe Flash player plugins.

**78. How to verify whether an object is present on the multiple pages?**

**Answer:**answer will be provided soon.

**79. Which driver implementation will allow headless mode?**

**Answer:**HTML UnitDriver is the most light weight and fastest implementation headless browser for of WebDriver. It is based on HtmlUnit. It is known as Headless Browser Driver. It is same as Chrome, IE, or FireFox driver, but it does not have GUI so one cannot see the test execution on screen.

**80. What are important Classes in WebDriver?**

**Answer:** The major implementation classes of WebDriver interface are ChromeDriver, EdgeDriver, FirefoxDriver, InternetExplorerDriver etc. Each driver class corresponds to a browser. We simply create the object of the driver classes and work with them. It helps you to execute Selenium Scripts on Chrome browser.

**81. What is Sikuli, its purpose and explain more about it?**

**Answer:** Sikuli is a scripting language which helps us to automate Software testing of Graphical User Interface(GUI). ... It basically uses image recognition technology as it helps us to identify and control GUI components. It is robust and powerful GUI automation tool.

**82. What is Robot Class and explain more about it?**

**Answer:** Answer given in question#49

**83. What is AutoIt and explain more about it?**

**Answer:** AutoIt is an open source automation language for Windows operating system. AutoIt uses the combination of simulated keystrokes, mouse movement, and window manipulation in order to perform user interface testing.

Article: <https://www.automationtestinginsider.com/2020/01/autoit-tool-and-two-ways-to-handle.html>

**84. What is the difference between AutoIt, Sikuli and Robot Class?**

**Answer:**answer will be provided soon.

**85. Which is best in AutoIt, Sikuli and Robot Class along with reasons?**

**Answer:**answer will be provided soon.

**86. What classes you will use for reading the PDF files?**

**Answer:** We will use PDFBox API to read PDF file using Java code.

**87. What is the use of pdfstripper class?**

**Answer:**  Class PDFTextStripper. This class will take a pdf document and strip out all of the text and ignore the formatting and such. Please note; it is up to clients of this class to verify that a specific user has the correct permissions to extract text from the PDF document.

**88. What is verbose?**

**Answer:**TestNG - Verbose Attribute [Selenium Users] Verbose Attribute lets you obtain clear reports through IDE console. This attribute will be placed inside the <Suite> tag of testng.xml as shown below: <suite name="Suite" parallel="tests" verbose="2">

**89. What is thread count?**

**Answer:** It's the number of tests that are run at the same time. It takes up more slots in the grid because is uses one grid slot per test run. It's like saying how many cars in a fleet that you want on the highway at the same time. If you say you want them to take up three lanes, then they will take up no more than three lanes, but it will take longer to get all cars through. If you say five lanes, then they will take up no more than five lanes, but it will take less time to get all cars through.

**90. What is the difference between build and perform methods in Actions Class?**

**Answer:** build() method in Actions class is use to create chain of action or operation you want to perform. perform() this method in Actions Class is use to execute chain of action which are build using Action build method.

**91. How will you install ReportNG in your project?**

**Answer:**  ReportNG is a simple plug-in for the TestNG unit-testing framework to generate HTML reports as a replacement for the default TestNG HTML reports. You can also customize html report with the help of TestNG listeners.

To use ReportNG reports we need to follow the below three steps:

Step 1: Add the below Jars Files to your project.

reportng-1.1.4.jar

velocity-dep-1.4.jar

guice-3.0.jar

Step 2: To make sure reportNG reports, we need to disable the default TestNG Listeners.

It can be done by following the below steps:

1. Right Click on Properties

2. Click on TestNG

3. You will find an option as "Disable default listeners", check the checbox

4. Click on "Apply" button, it will show as message as "Project preferences are saved".

5. Now Click on "OK" button.

Step 3: We need to add the below two listeners to testng.xml file.

<listeners>

      <listener class-name="org.uncommons.reportng.HTMLReporter"/>

      <listener class-name="org.uncommons.reportng.JUnitXMLReporter"/>

  </listeners>

Finally testng.xml file should look as the below for the given example :

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >

<suite name="Suite1" verbose="1" >

<listeners>

      <listener class-name="org.uncommons.reportng.HTMLReporter"/>

      <listener class-name="org.uncommons.reportng.JUnitXMLReporter"/>

  </listeners>

  <test name="Regression Test Suite"   >

    <packages>

      <package name="packOne" />

      <package name="packTwo" />

   </packages>

 </test>

</suite>

**92. Why do we go for Apache POI API? What is its purpose?**

**Answer:** Apache POI (Poor Obfuscation Implementation) is an API written in Java to support read and write operations – modifying office files. This is the most common API used for Selenium data driven tests to read/write excel file.

**93. Why do we provide “//” in java while fetching a path of excel?**

**Answer:** In a string a single backslash is a so-called 'escape' character. This is used to include special characters like tab (\t) or a new line (\n). In order to specify a backslash in the case of a path you will have to 'espace' the single slash using the escape character, which means that you will have to write a double backslash.

**94. What is an object array and why do we use it for data provider?**

**Answer:** An array of array of objects (Object[][]) where the first dimension's size is the number of times the test method will be invoked and the second dimension size contains an array of objects that must be compatible with the parameter types of the test method

The reason is quite simple as we get test data in forms of rows and columns so that we need 2D object (object type because we are free to pass any type data String, int etc.) array.

An important features provided by TestNG is the testng DataProvider feature. It helps you to write data-driven tests which essentially means that same test method can be run multiple times with different data-sets.

A Data Provider is a method on your class that returns an array of array of objects.

//This method will provide data to any test method that declares that its Data Provider

//is named "test1"

@DataProvider(name = "test1")

public Object[][] createData1() {

 return new Object[][] {

   { "Cedric", new Integer(36) },

   { "Anne", new Integer(37)},

 };

}

//This test method declares that its data should be supplied by the Data Provider

//named "test1"

@Test(dataProvider = "test1")

public void verifyData1(String n1, Integer n2) {

 System.out.println(n1 + " " + n2);

}

**95. In how many ways, can I take keyboard inputs?**

**Answer:**There are many ways to read data from the keyboard in java:

* InputStreamReader Class.
* Scanner Class.
* Using Command Line Arguments.
* Console Class.

**96. Is it possible to compare two images with Sikuli?**

**Answer:** answer will be provided soon.

**97. What is the difference between “type” and “typeAndWait” command?**

**Answer:**"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.

**98. How can you run Selenium Server other than the default port 4444?**

**Answer:** The Hub will listen to port 4444 by default. You can view the status of the hub by opening a browser window and navigating to http://localhost:4444/grid/console. To change the default port, you can add the optional -port flag with an integer representing the port to listen to when you run the command.

**99. Explain how you can capture server side log Selenium Server?**

**Answer:**   Selenium RC Server Logging has two options: Server-Side Logs and Browser-Side Logs.

Server-Side Logs:

When launching Selenium server with the -log option, the server can record valuable debugging information reported by the Selenium Server to a text file.

For example: this command below will create a file named "selenium.log" under c:\SeleniumTestCase folder.

C:\selenium-java-2.39.0>java -jar selenium-server-standalone-2.39.0.jar -log c:\SeleniumTestCase\selenium.log

Another example: this example below creates a debug.log file with debug information and nnn.log file

C:\selenium-java-2.39.0>java -jar selenium-server-standalone-2.39.0.jar -debug -log c:\SeleniumTestCase\debug.log -DSelenium.LOGGER=nnn.log

Setting system property Selenium.LOGGER to nnn.log

**Browser-Side Logs:**

JavaScript on the browser side also logs important messages. To access browser-side logs, pass the -browserSideLog argument to the Selenium Server.

For example: this command below creates a log on the browser side under c:\SeleniumTestCase folder. If -log option is not defined, the server console will display all the debug information.

C:\selenium-java-2.39.0>java -jar selenium-server-standalone-2.39.0.jar -browserSideLog -log c:\SeleniumTestCase\browserside.log

**100. Can you handle flash using web driver?**

**Answer:** Answer given in question#77

**101. What is the difference between dragAndDrop() and dragAndDropBy()?**

**Answer:**  The Actions class has two methods that support Drag and Drop.

Actions.dragAndDrop(Sourcelocator, Destinationlocator)

In dragAndDrop method, we pass the two parameters -

First parameter "Sourcelocator" is the element which we need to drag

Second parameter "Destinationlocator" is the element on which we need to drop the first element

Actions.dragAndDropBy(Sourcelocator, x-axis pixel of Destinationlocator, y-axis pixel of Destinationlocator)

dragAndDropBy method we pass the 3 parameters -

First parameter "Sourcelocator" is the element which we need to drag

The second parameter is x-axis pixel value of the 2nd element on which we need to drop the first element.

The third parameter is y-axis pixel value of the 2nd element on which we need to drop the first element.

https://www.automationtestinginsider.com/2020/01/how-to-handle-drag-and-drop-in-selenium.html

**102. What is the use of getPageSource()?**

**Answer:** There is a method called getPageSource() in selenium webdriver. It returns string, so you can either store it in a file or can print it in the console.

**103. What is bitmap Comparison? Why we use it in Selenium WebDriver?**

**Answer:**   Please refer below link

<https://www.telerik.com/forums/bitmap-comparison>

**104. Tell me some of the tools name which is used to store the script in common place?**

**Answer:**  SVN, GitHub

**105. What is an assertion? What is its drawback? How to overcome it?**

**Answer:** Assertions verify that the state of the application is same to what we are expecting. Selenium Assertions can be of three types: “assert”, “verify”, and ” waitFor”. When an “assert” fails, the test is aborted. When a “verify” fails, the test will continue execution, logging the failure.

Please refer detail answer in question#13

**106. What is a headless browser?**

**Answer:** Please refer answer from question#17

**107. Have you ever done profiling of a web page?**

**Answer:** profile is the collection of settings, customization, add-ons and other personalization settings that can be done on the Firefox Browser. You can customize profile to suit your Selenium automation requirement.

**108. What are the different methods available in selenium webdriver?**

**Answer:** **1. get ()**.

It is used to open specified url browser in windows.

Syntax:

//to launch the browser

driver. get(http://google.com);

**2. getCurrentUrl().**

Its Returns title of the Browser

Syntax:

//to launch the browser

Driver.get(http://google.com);

String url=driver.getCurrentUrl();

System.out.println(url);

**3. getTitle().**

It is used to get the title of current web page

Syntax:

//to launch the browser

driver.get(“http://www.google.com”);

String title=driver.getTitle();

**4. getPageSource().**

It is used to get the source of current load page

Syntax:

//to launch the browser

driver.get(“http://www.google.com”);

String pagesource=driver.getPageSource();

System.out.println(pagesource);

**5. findElement().**

It is used to find the first WebElement using the given method.

Syntax:

//to launch the browser

driver.get(“http://www.gmail.com”);

WebElement gmaillink=driver.findElement(By.id());

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

**6. findElements().**

It is used to find all elements within the current page

Syntax:

//to launch the browser

driver.get(“http://www.facebook.com”);

//to findelements

List links=driver.findElements(By.TagName(“a”));

//Counting no of links in result page

System.out.println(links.size());

**7. close().**

Close the current window, if there are multiple windows, it will close the current window which is active and quits the browser if it’s the last window opened currently.

Syntax:

driver.get(“http://www.etestinghub.com”);

driver.close();

**8. quit().**

It is used to close every associated window which is opened.

Syntax:

driver.get(“http://www.etestinghub.com”);

driver.quit();

**9. getWindowHandle().**

Whenever the web driver launches the browser it assigns the unique id to that browser which is called as window handler. This can be captured through the method.

Syntax: driver.getWindowhandle().

**10. getWindowHandles().**

Whenever multiple windows are opened by webdriver and we want to capture all their ids. We use this method.

Syntax: getWindowHandles().

**11. switchTo().**

Used to switch from one window to another window (or) window to a frame (or) frame to a window (or) window to an alert

Syntax:

driver.switchTo().window();

driver.switchTo().frame();

driver.switchTo().alert();

**12. navigate().**

The driver to access the browser’s history and to navigate to a given URL&Refresh page.

Syntax:

driver.get(“http://gmail.com”);

//navigate to page

driver.navigate().to(“http://estestinghub.com “);

//navigate to back

driver.navigate().back();

//navigate to forward

driver.navigate().forward();

//navigate to refresh page

driver.navigate().refresh();

**13. manage().**

This is used to perform maximize the size of the window.

driver.get(“http://gmail.com”);

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

**Web operations on web Elements**

**1.click()**

This is used to click on webelements like link, button, radio group, checkbox, images…etc.

**2.sendKeys()**

Purpose: This is used to sending inputs into text fields and text areas, and also used to select value from the drop down box.

**3.clear()**

**Purpose:** This is used to clear the input from existing data.

**4.getText()**

Purpose: This is used to capture text of the webElement.

**5. getTagName()**

Purpose: This is used to capture html tag of the webElement.

**6.getLocation()**

This is used to capture X and Y co-ordinates of webelement in the application.

**7.isSelected()**

This is used to check, is the check-box is currently checked or unchecked to checked Radio buttons are selected or not.

**8. isDisplayed()**

This is a Boolean condition. It is used to either an element is visible or not.

If an element is displayed it gives true and an element is not displayed it gives false.

**9. IsEnabled()**

This is a Boolean condition. It is used to either an element is enable or not.

If an element is enable it gives true and an element is disable it gives false.

**10.getAttribute ()**

This is used to capture the attributes which are present in web applications.

**109. Which time unit we provide In time test? minutes? seconds? milliseconds? or hours? Give Example?**

**Answer:**  SECONDS

**110. Can we use implicitly wait() and explicitly wait() together in the test case?**

**Answer:** Implicit wait destroys meaning of using explicit wait when using together. So it is advised not to use implicit wait and explicit wait together. Actually when we use both waits together, both waits will be applied at the same time and it get messed up.

Now we will see reason behind these using below scenarios:

1. Explicit Wait= Implicit Wait (Say 10 seconds)

Both waits get activated at same time to locate element. Explicit wait keeps searching for an element till it is found and implicit wait allows webdriver to search till timeout. When explicit wait starts and looks for element, because of implicit wait it needs to wait for 10 seconds because element is not found. So both waits completes 10 seconds wait time.

2. Implicit wait(20) > Explicit Wait(10)

When explicit wait stars looking for element, it needs to wait for 20 seconds because of implicit wait time.

3. Implicit wait(10) < Explicit Wait(20)

When explicit wait starts looking for element, it needs to wait for 10 seconds because of implicit wait. After that implicit wait throws exception because of not able to locate element.  Exception stops explicit wait to search further and does not allow to reach its timeout.

Few more points to remember:

1. The most widely used waits are implicit and explicit waits. Fluent waits are not preferable for real-time projects.

2. We use FluentWait commands mainly when we have web elements which sometimes visible in few seconds and some times take more time than usual. Mostly in Ajax applications.

3. We use mostly explicit wait because it is for specific condition/element and we have more flexibility in using explicit rather than implicit.

**111. Does selenium support https protocols ?**

**Answer:**  Yes

**112. What is object repository and explain page factory technique?**

**Answer:**Page Object Model is an Object Repository design pattern in Selenium WebDriver. POM creates our testing code maintainable, reusable. Page Factory is an optimized way to create object repository in POM concept.

Page Object Model is an Object Repository design pattern in Selenium WebDriver. POM creates our testing code maintainable, reusable. Page Factory is an optimized way to create object repository in POM concept.

**What is Page Factory?**

Page Factory is an inbuilt Page Object Model concept for Selenium WebDriver but it is very optimized.

we follow the concept of separation of Page Object Repository and Test Methods. Additionally, with the help of PageFactory class, we use annotations @FindBy to find WebElement. We use initElements method to initialize web elements

@FindBy can accept tagName, partialLinkText, name, linkText, id, css, className, xpath as attributes.

Let's look at the below example using Page Factory.

package PageFactory;

import org.openqa.selenium.WebDriver

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class HomePage {

    WebDriver driver;

    @FindBy(xpath="//table//tr[@class='TestUsername']")

    WebElement homePageUserName;

    public Guru99HomePage(WebDriver driver){

        this.driver = driver;

        //This initElements method will create all WebElements

        PageFactory.initElements(driver, this);

    }

    //Get the User name from Home Page

        public String getHomePageDashboardUserName(){

         return    homePageUserName.getText();

        }

}

**113. What is log4j and how did you use in web driver?**

**Answer:**

Log4j is an open source logging framework. With log4j – logging behavior can be controlled by editing a configuration file only without touching the application binary and can be used to store the Selenium Automation flow logs. It equips the user with detailed context for application failures.

log4j has 3 major components:

Loggers – It is used for logging information. To use loggers we need to take care of things mentioned below:

Create object of logger class: Logger class is a Java-based utility which has all the generic methods in it to use log4j (To use Logger class we need to Import org.apache.log4j.Logger)

Define the log level: We can define the log levels in multiple forms and levels available are:

All – This level of logging will log everything, it is intended to turn on all logging.

DEBUG – It saves the debugging information and is most helpful to debug an application.

INFO – It prints informational message that highlights the progress of the application.

WARN – It designates potentially harmful situations.

ERROR – It designates error events that might still allow the application to continue running.

FATAL – It designates very severe error events that will presumably lead the application to crash

OFF – It is intended to turn off logging.

Appenders – In log4j, an output destination is called an appender. It allows the destination where the logs would get saved. It supports the following types of appenders:-

ConsoleAppender – It logs to some standard output.

File appender – It prints logs to some file at a particular destination.

Rolling file appender – It is used to for a log file with maximum size.

Layouts – It is used to format the logging information in different style.

Link - <https://logging.apache.org/log4j/1.2/manual.html>

**114. Why we use selenium grid and jenkins and advantages of it?**

**Answer:**  Selenium Grid is a part of the Selenium Suite that specializes in running multiple tests across different browsers, operating systems, and machines in parallel.

 Selenium Grid has 2 versions - the older Grid 1 and the newer Grid 2. We will only focus on Grid 2 because Grid 1 is gradually being deprecated by the Selenium Team.

Selenium Grid uses a hub-node concept where you only run the test on a single machine called a hub, but the execution will be done by different machines called nodes.

When to Use Selenium Grid?

You should use Selenium Grid when you want to do either one or both of following:

Run your tests against different browsers, operating systems, and machines all at the same time. This will ensure that the application you are Testing is fully compatible with a wide range of browser-O.S combinations.

Save time in the execution of your test suites. If you set up Selenium Grid to run, say, 4 tests at a time, then you would be able to finish the whole suite around 4 times faster.

Jenkins

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

**115. How many types of reports you will generate in your project?**

**Answer:** What are the essential qualities of a good test report?

**Brevity**–

A report should be short and concise.

It should reveal the total no. of successes as well as failures.

**Trackability**–

Captures all the footprints that could lead to the root cause of a failure.

**Traceability**–

It must provide the ability to review the following.

Historical data for test cases and failures

Age of a particular defect

**Sharable**–

It should support a format that you can share through email or integrate with CI tools like Jenkins/Bamboo.

Test coverage –

It should highlight the test coverage for the following.

Test coverage of the module under test.

Test coverage of the application under test.

Selenium Webdriver doesn’t have a built-in reporting feature, but there are plugins like the TestNG and JUnit which can add this functionality.

**TestNG HTML Report Generation.**

**Generating Extent HTML Reports.**

**116. Which is fastest web browser**

**Answer:** Opera is the fastest browser in a recent test. PC World, one of the world's leading technology publications, recently took a more in-depth look at the top 5 web browsers for computers: Opera, Firefox, Chrome, Microsoft Edge and Internet Explorer

**117. For a simple single page how to start automation testing?**

**Answer:**answer will be provided soon

**118. Diff between findby, findbys, findAll**

**Answer:** @FindBys will return the elements depending upon how @FindBy specified inside it. to put it in simple words, @FindBys have AND conditional relationship among the @FindBy whereas @FindAll has OR conditional relationship.

**119. Challenges in Selenium and how to overcome?**

**Answer:**Please refer answer from question#31, 66

**120. what are static members in selenium?**

**Answer:**answer will be provided soon

**121. which version of selenium?**

**Answer:**  Mention the version of selenium you are using.  Like Selenium 3.

**122. which version of Eclipse?**

**Answer:**Mention the version of eclipse you are using

**123. How do you use action class?**

**Answer:** Refer below link

<https://www.automationtestinginsider.com/2020/01/actions-class-in-selenium.html>

**124. what is assert class?**

**Answer:**The assert methods are provided by the class org. junit. Assert which extends java. ... Object class. There are various types of assertions like Boolean, Null, Identical etc.

**125. Difference between pom.xml and testng.xml**

**Answer:** testng. xml is the configuration for TestNG testing framework (e.g. defining test suites, test listeners, etc.) pom. xml is the configuration for Maven build tool (e.g. defining build plugins, compile and test dependencies, build profiles, etc.)

**126. Can we give priority as –ve**

**Answer:** Yes

Priority is an element applicable only for @Test annotated methods. Priority should be an integer value. It can be negative , zero or positive number. ... TestNG will execute test methods from lowest to highest priority.

**127. Diff between selenium 1, 2, 3 ,4?**

**Answer:**Please refer below article

https://www.automationtestinginsider.com/2019/07/selenium-webdriver-part1-selenium-and.html

**128. When to use explicit wait?**

**Answer:** Please refer below article

<https://www.automationtestinginsider.com/2020/02/waits-in-selenium-webdriver.html>

**129. What is by and what it returns?**

**Answer:** By is a class and Mechanism used to locate elements within a document. In order to create your own locating mechanisms, it is possible to subclass this class and override the protected methods as required, though it is expected that all subclasses rely on the basic finding mechanisms provided through static methods of this class.

**130. Diff among implicit, explicit and fluent wait**

**Answer:** Please refer below article

<https://www.automationtestinginsider.com/2020/02/waits-in-selenium-webdriver.html>

**131. How to handle null pointer exception after importing a project in eclipse**

**Answer:**answer will be provided soon

**132. When to use soft assertion**

**Answer:** When an assert fails the test script stops execution unless handled in some form. We call general assert as Hard Assert

**Hard Assert**– Hard Assert throws an AssertException immediately when an assert statement fails and test suite continues with next @Test

The disadvantage of Hard Assert – It marks method as fail if assert condition gets failed and the remaining statements inside the method will be aborted.

To overcome this we need to use Soft Assert. Let’s see what is Soft Assert.

**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.

If there is any exception and you want to throw it then you need to use assertAll() method as a last statement in the @Test and test suite again continue with next @Test as it is.

**133. How to login any application using 5 different users at the same time?**

**Answer:** Using parallel testing by passing parameters from testng.xml

Here I have given example of two user login:

**Test Class:**

**public** **class** **Login** {

WebDriver driver;

**@Parameters**({"Username","Password"})

**@Test**()

**public** **void** **loginTest**(String userName, String password) {

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\Hitendra\\Downloads\\chromedriver\_win32\\chromedriver.exe");

driver = **new** ChromeDriver();

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

driver.get("https://opensource-demo.orangehrmlive.com/index.php/auth/validateCredentials");

driver.findElement(By.id("txtUsername")).sendKeys(userName);

driver.findElement(By.id("txtPassword")).sendKeys(password);

driver.findElement(By.id("btnLogin")).click();

}

}

**Testng.xml**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Suite" parallel="tests">

<test name="Test1">

<parameter name="Username" value="admin"></parameter>

<parameter name="Password" value="admin123"></parameter>

<classes>

<**class** **name**="com.package1.Login"/>

</classes>

</test> <!-- Test -->

<test name="Test2">

<parameter name="Username" value="admin1"></parameter>

<parameter name="Password" value="admin1234"></parameter>

<classes>

<**class** **name**="com.package1.Login"/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

**Output:**

**Output:**

[Graphical user interface, text, application

Description automatically generated](https://1.bp.blogspot.com/-GAr0KT8DHNA/XneCEMpROpI/AAAAAAAAAdw/ANpbbiIRJDgmjtR20AhuIR-ISiExBdBygCLcBGAsYHQ/s1600/Question134%2B%25281%2529.webp)

**134. driver.get or driver.navigate.get which one is faster ?**

**Answer:**main difference between get() and navigate() is, both are performing the same task but with the use of navigate() you can move back() or forward() in your session's history.

navigate() is faster than get() because navigate() does not wait for the page to load fully or completely.

**135. How to do Parellel testing and cross browser testing**

**Answer:**answer will be provided soon

**136. Diff between seleniumatandalone and selenium-jar files**

**Answer:**Selenium server standalone jar is a library that provides you the classes of Selenium automation framework.

Just for additional info, selenium-standalone-server.jar is a bundled jar that contains both API and selenium server.

Selenium server is required to run older selenium RC tests or to run WebDriver tests in remote machines through selenium Grid.

**137. Diff B/W java script click and normal click?**

**Answer:**click is a function on HTML elements you can call to trigger their click handlers: element. ... onclick is a property that reflects the onclick attribute and allows you to attach a "DOM0" handler to the element for when clicks occur: element.

**138. What is the diff between profiles vs options vs desiredcapabilities**

**Answer:** Capabilities are options that you can use to customize and configure a ChromeDriver session. This page documents all ChromeDriver supported capabilities and how to use them.

There are two ways to specify capabilities.

The first is to use the ChromeOptions class.

If your client library does not have a ChromeOptions class (like the selenium ruby client), you can specify the capabilities directly as part of the DesiredCapabilities.

**139. What are the new things in geckodriver**

**Answer:**answer will be provided soon

**140. What is the diff b/w POM with and without page factory?**

**Answer:** Page Object Model (POM) and Page Factory has following differences:

A Page Object Model is a test design pattern which says organize page objects as per pages in such a way that scripts and page objects can be differentiated easily. A Page Factory is one way of implementing PageObject Model which is inbuilt in selenium.

In POM, you define locators using ‘By’ while in Page Factory, you use FindBy annotation to define page objects.

Page Object Model is a design approach while PageFactory is a class which provides implementation of Page Object Model design approach.

POM is not optimal as it does not provide lazy initialization while Page Factory provides lazy initialization.

Plain POM will not help in StaleElementReferecneException while Page Factory takes care of this exception by relocating web element every time whenever it is used.

In plain page object model, you need to initialize every page object individually otherwise you will encounter NullPointerException while In PageFactory all page objects are initialized (Lazily) by using initElements() method.

**141. Diff between pause and thread.sleep?**

**Answer:** The major difference is that wait() releases the lock or monitor while sleep() does

**142. What is the usage of remote web driver?**

**Answer:**RemoteWebDriver is an implementation class of the WebDriver interface that a test script developer can use to execute their test scripts via the RemoteWebDriver server on a remote machine.

**143. Is selenium supports angular jar?**

**Answer:** There is a library called ngWebDriver that is designed to automate AngularJS and Angular Web Applications using Selenium with Java. ... No need to write extra JavaScript for Angular Requests Waiting. It provides new locating techniques to use Angular Specific Attributes.

**144. Diff between isDisplayed and isPresent?**

**Answer:** There is a major difference between isDisplayed() and isPresent(). isDisplayed() - Your element is present on the page but it is displayed. isPresent() - Your element is present in entire DOM of the page. Probably it can be hidden or not disabled, but present.