**Selenium Question -105**

**1) What is Automation Testing?**

Automation testing is the process of Software testing using automation tools to find the defects

before it is released into production.

In automation testing Special software tools [like Selenium, Jenkins, Katalon /studio,Test Complete, HP,QFT, Tosca TestSuite etc ]are used to control the test execution, actual outcomes and predicted outcomes comparison, the test preconditions setting up, and other test control and test reporting functions.

**2) What are the main advantages of Automation Testing?**

**Save time and money, It is faster in execution, Easy reporting -it generates automatic reports after test execution using TestNG. It is cheaper compared to Manual Testing in a long run , Easy for compatibility testing, It enables parallel execution in the combination of different OS and browser environment. It helps to increase the test coverage.** reducing human-generated errors. consistency in testing, test cases reusability, reduced software maintenance cost, increased test effectiveness, reduction of the test interval.Regression testing coverage, test engineer productivity.

**3) what is Selenium?**

Selenium is a free and robust, open test automation suite to test web applications. It is an API to drive the browser. It is designed in a way to support and encourage automation testing of functional aspects of web-based applications and a wide range of browsers and platforms.

**4) What are the main advantages of Selenium?**

* Selenium is a free and open source.It supports multiple languages like java, rubi,phython etc.
* Cross Browser compatibility (Firefox, Chrome, Internet Explorer, Safari etc.)
* Compatibility with the main platform (Windows, Mac OS, Linux etc.)
* It is an API to drive the browser.
* Automation scripts creating ability for non-programmers as well as for programmers
* Testing distribution support
* Regular and fresh repository developments
* It is not working in case of Mobile testing. It uses APM for mobile testing
* It can not generate report. For reporting it takes help from TestNG and Junit.
* It can not read Bar code and Capvhst- in that case we need for z- ging

**5) What Selenium components do you know?**

Selenium is a free and robust, open test automation suite to test web applications. It has four components :

* **Selenium IDE (Integrated Development Environment)**. It is a tool for recording and playing back. It is a Firefox plugin. We can create scripts using selenium IDE. It works with both chrome and Firefox, but not IE.
* **Selenium RC**. It provides the APIs for a variety of languages like Java, .NET, PHP, etc. They work with most of the browsers. It supports in a maintenance mode. It needs a separate sever to start /run the program. Need to start server before execution the test scripts
* Selenium WebDriver-It is browser automation framework that accepts commands and send them to a browser. It is implemented through a browser -specific driver. It supports multiple languages like java, rubi, phython, etc. loop m if else, it has not reporting system. For reporting purpose we take need junit an test ng
* **Grid:** It is used to distribute tests on multiple machines so that test can be run parallel which helps cutting down the time .So It helps in parallel and distributes testing . it is used to distribute the test execution on multiple platforms and environments concurrently. [Own configuration. To configure -open everything to- webdriver = new Remote [hub -own .– node -child ]
* That why we use sauce lab.

**6) How many types of Webdriver APIs are available in Selenium?**

The list of driver classes could be used for the browser automation.

* AndroidDriver,
* ChromeDriver,
* **EventFiringWebDriver**,It is a JavaClass.extends java.lang.Object
* FirefoxDriver,
* HtmlUnitDriver,
* InternetExplorerDriver,
* iPhoneDriver,
* iPhoneSimulatorDriver,
* RemoteWebDriver

**AndroidDriver This driver class inherits from AppiumDriver,** but it adds in additional functions that are**:** useful in the context of a mobile automation test on Android devices through Appium. Only use this driver class if you want to start a test on an Android device or Android emulator.

WebDriver is an open source tool for automated testing of webapps across many browsers. It provides capabilities for navigating to web pages, user input, JavaScript execution, and more. It is available for Chrome on Android and Chrome on Desktop (Mac, Linux, Windows and ChromeOS).

public class **EventFiringWebDriver**

extends java.lang.Object

implements [WebDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html), [JavascriptExecutor](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/JavascriptExecutor.html), [TakesScreenshot](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/TakesScreenshot.html), [WrapsDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WrapsDriver.html), [HasInputDevices](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/interactions/HasInputDevices.html), [HasTouchScreen](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/interactions/HasTouchScreen.html), [Interactive](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/interactions/Interactive.html), [HasCapabilities](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/HasCapabilities.html)

A wrapper around an arbitrary [WebDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html) instance which supports registering of a [**WebDriverEventListener**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/events/WebDriverEventListener.html)**,** e.g. for logging purposes.

public class **FirefoxDriver**

extends [RemoteWebDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/remote/RemoteWebDriver.html)

implements [WebStorage](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/html5/WebStorage.html), [HasExtensions](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/firefox/HasExtensions.html)

**All Implemented Interfaces:**

[**HasExtensions**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/firefox/HasExtensions.html)**,**[**HasCapabilities**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/HasCapabilities.html)**,**[**WebStorage**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/html5/WebStorage.html)**,**[**HasInputDevices**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/interactions/HasInputDevices.html)**,**[**Interactive**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/interactions/Interactive.html)**,**[**JavascriptExecutor**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/JavascriptExecutor.html)**,**[**SearchContext**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/SearchContext.html)**,**[**TakesScreenshot**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/TakesScreenshot.html)**,**[**HasVirtualAuthenticator**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/virtualauthenticator/HasVirtualAuthenticator.html)**,**[**WebDriver**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html)

HtmlUnitDriver is a WebDriver compatible driver for the [HtmlUnit](http://htmlunit.sourceforge.net/) headless browser.

public class **InternetExplorerDriver in java.** extends [RemoteWebDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/remote/RemoteWebDriver.html).

**All Implemented Interfaces:**

[**HasCapabilities**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/HasCapabilities.html)**,**[**HasInputDevices**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/interactions/HasInputDevices.html)**,**[**Interactive**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/interactions/Interactive.html)**,**[**JavascriptExecutor**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/JavascriptExecutor.html)**,**[**SearchContext**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/SearchContext.html)**,**[**TakesScreenshot**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/TakesScreenshot.html)**,**[**HasVirtualAuthenticator**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/virtualauthenticator/HasVirtualAuthenticator.html)**,**[**WebDriver**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html)

[Selenium WebDriver](https://www.browserstack.com/guide/selenium-webdriver-tutorial) is a tool used to execute automated test cases on various browsers. The object of the webdriver is a browser.

**Remotewebdriver**

* [Selenium](https://www.browserstack.com/selenium) remotewebdriver implements the webdriver interface to execute test cases.
* RemoteWebDriver consists of a server and a client.
* The server is a component that listens on a port for various requests from a RemoteWebDriver client. Once the request is received, it forwards the request to the browser driver: FirefoxDriver, [IEDriver](https://www.browserstack.com/guide/run-selenium-tests-using-ie-driver), or [ChromeDriver](https://www.browserstack.com/guide/run-selenium-tests-using-selenium-chromedriver).

**7) Does the Selenium have any limitations?**

Selenium supports only web based applications testing. So, here are the limitations of it:

* Mobile applications cannot be tested using Selenium
* Desktop applications cannot be tested using Selenium
* Captcha and Bar code readers cannot be tested using Selenium
* User should use third-party tools like TestNG or jUnit to write test scripts and generate reports
* Programming language knowledge is required to create robust scripts in Selenium WebDriver

**8) What is Selenium IDE?**

Selenium IDE is a plug-in used to record and replay tests in Firefox browser. Scripts may be automatically recorded and edited manually providing auto-completion support and the ability to move commands around quickly.

**9)** **What is Selenese?** Selenese is the language which is used to write test scripts in Selenium IDE.

**10)** **What kinds of test types are supported by Selenium?**

1. Functional Testing[**Checking the functionality of the particular application in a different way-login scenario]**
2. Regression Testing**[process of testing changes to computer program to make sure that the older program still works with new changes]**
3. Sanity Testing[**Software receive after build with little changes in code or functionality to assure that no issues introduces due to new changes**]
4. Smoke Testing[ **Software receive after build to find if critical [Major]functionality of the program are working fine or not]**
5. Responsive Testing[**testing** the website or URL from different devices]
6. Cross Browser Testing[**testing** refers to **testing** a website in multiple platform, version and multiple **browsers** like IE, Chrome, Firefox to check its efficacy on each.]
7. UI testing (black box)
8. Integration Testing ---are supported by Selenium.[]

**Black box testing involves testing** a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

**11) What are the different types of locators in Selenium?**

The different types of locators in Selenium are ID, ClassName, Name, TagName, LinkText, PartialLinkText, XPath, CSS Selector, DOM.

**12) What automation tools could be used for post-release validation with continuous integration?**

Automation tools could be used for post-release validation with continuous integration: CruiseCont, Hudson, Jenkins nighty basis and if any bug get then fix, Quick Build.

**13) Explain the meaning of assertion in Selenium and what are the types of assertion?**

Assertion is used as a verification point. Asserts helps to verify the conditions of the test and decide whether test has failed or passed. A test is considered successful only if it completed without throwing any exception. In TestNg we have two types exceptions, one is

1. **Hard Assertion and 2) Soft Assertion.**

A**)Hard Assertions** -  In hard assertion when we use it if a test has to stop immediately after this session/condition is failed. That means if the assert condition is true then the program will execute the next test steps but if the condition is false, the execution will stop and further test steps will not be executed.

**Statement– of Hard Assert**

In hard assertion we have three types of Assertion statement/ commands

* Assert.assertTrue()
* Assert.assertFalse()
* Assert.assertEquals()

B) Soft Assertions- In the soft assertion if the test statement or assertion or condition is failed then again it will proceed with the rest of the statement.

Statement of SoftAssert—

SoftAssert softAssert = new SoftAssert();

softAssert.assertTrue(Condition);

softAssert.assertAll();-------

 If you want to use software assertion in testNg, we need to create an object for soft asset class and through this object we can use asset true and asset all commands.

Difference between Hard assertion and Soft assertion – In hard assertion even though the assertion statement is failed rest of the commands will not be execute. But In case of soft assertion even though that the assertion statement is failed rest of the statements will be continued.

softAssert.assertAll();-------This is a command which will make rest of the statements will execute even though the previous statements have been failed. so that's the reason we must specify softAssert.assertAll();-at all at the end of the script.

**Question-**

**How written two test cases or multiple test cases both the test methods are using the same object ?**

If we want to pass all the test case then we use softassert. Becaue we know that all are manually ok and pass.

Here soft assert and softAssert object is created. so both test 1 and test 2 both the test methods I have used same object.

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

@Test

**void** DemoTest1() {

softAssert.assertEquals("wel", "Welcome"); //-false -failed

softAssert.assertAll(); }

@Test

**void** DemoTest2() {

softAssert.assertEquals("Welcome", "Welcome");

softAssert.assertAll();}

-----------------------------------------------

For example in the first test method this statement has been failed by default the second test method will also

Fail. even though the statement is true the second test method also will fail. why because we are using the same

Object across multiple test methods.

In the first test method I have said two different strings here and use two set equals method by using this object. now basically these two are not equal. so that it will fail and it will return false and finally it is failed.

and if you see the second method test method both are exactly the same String. So it should return true and you should also pass . but finally it will failed because both for test cases we are using same object.

So to overcome this problem we have to create two different objects for the two different test cases/ methods.

**14) Explain the difference between assert and verify commands?**

Both assert and verify command are used to find whether a given input is present or not on web page. That means both of them check if the given condition is true or false. In TestNG we use only Assert statements, we can use verify statements in terms of if -else and try catch.

**15)** **What is a XPath?**

XPath ore XML path is a query language for selecting nodes from XML documents. It is the locator and it is used to locate the elements which is supported by selenium web driver. Using Xpath we could navigate through elements and attributes in an XML documents to locate web elements such as texrbox, button, checkbox , image etc in a web page.

**16) What is an Absolute XPath?**

**It is called complete or full xpath. It starts with single forward slash.**Absolute XPath is the direct way to find the element. It identifies the element very fast. **XPath gets failed if there are any changes or added or removed made in between in the path of the element.**  Just Immediate tag.

html/body/div/div/div[@id=’Email’]– Absolute XPath example.

**17)** **What is a Relative XPath?**

Relative XPath means that user can start from the current location or middle of the HTML DOM structure. It starts with double forward slash.Example of**Relative XPath – //input[@id=’email’]**.

--No need to write long xpath. This is a safer approach. And this xpath will select all the elements present in the DOM

**18)** **What is the difference between single slash (/) and a double slash ( //) in XPath?**

A single slash (/) is used for creating absolute XPaths . it begins with a root path and it is prefixed with a “/”.

Double slash (//) is used for creating relative XPath. It starts from anywhere / current location and it prefixed with “//”.

**19)** **How could the web element attributes be inspected in order to use them in different locators?**

**Firebug is a Firefox plugin** that provides various development tools for debugging applications. In case of automation, Firebug is used specifically for inspecting web-elements in order to use their attributes like id, class, name etc. in different locators.

**20)** **Give an example of the languages supported by WebDriver.**

Java, C#, Python, and Ruby, are all supported directly by the development team. There are also PHP and Perl WebDriver implementations.

**21) How can we launch different browsers in Selenium WebDriver?**

We should create **an instance of a driver of a particular browser**:

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

**22) What the WebDriver supported Mobile Testing Drivers do you know?**

Mobile Testing Drivers supported by the WebDriver are: AndroidDriver, IphoneDriver, OperaMobileDriver.

**23)** **Explain the fundamental difference between XPath and CSS selector.**

Using **XPath** we can traverse both forward and backward whereas **CSS** selector only moves forward.

**24) How can you find if an element is displayed on the screen?**

There are different methods, which help user to check the visibility of the web elements: **isDisplayed(), isEnabled(), isSelected()**.These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

**25)** **What is the difference between “type” and “typeAndWait” command?**

If you need to type keyboard key values into a text field of the web application, “type” command will be used. Another reason for its usage is selecting values of the combo box. “typeAndWait” command is used when your typing is completed and software web page start reloading.

**"type" command** is useful for typing keyboard key values into text box of software web application. you can also use it for selecting values of combo box.

"**typeAndWait" command** will be useful when your typing completed, software web page start reloading. This command will wait for software application page to reload. If there is not page reload event on typing, then you have to use simple "type" command.

**26) How can the user get a text of a web element?**

User can retrieve the text of the specified web element by using get command. It doesn’t require any parameter but returns a string value. Exmple

**String Text = driver.findElement(By.id(“Some Text”)).getText()**

**27)** **How a text written in a text field could be cleared?**

A text written in a text field could be deleted by using the **clear()** method.

**28)** **How to check a checkBox in Selenium?**

The same **click()** method could be used for checking checkbox as well as for clicking buttons or radio buttons.

**29)** **How to verify if the checkbox/radio is checked or not?**

**isSelected()** method is used to verify if the checkbox/radio is checked or not. An example –

**driver.findElement(By.xpath(“XPath of the checkbox/radio button”)).isSelected();**

**30)** **What is the alternate way to click on login button?**

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

**31)** **How to select a value in a dropdown?** WebDriver’s Select class is used to select value in the drop down.

selectByVisibleText:

Select selectByVisibleText = new Select (driver.findElement(By.id(“id\_of\_some\_element”)));  
selectByVisibleText.selectByVisibleText(“some\_visible\_text”);

**32)** **Explain the difference between close and quit command.**

If you need to close the current browser having focus **driver.close()** is used. If you need to close all the browser instances **driver.quit()** is used.

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

Both of these methods delay the speed of execution. The main difference between them is setSpeed sets a speed while will apply delay time before every Selenium operation takes place. thread.sleep() will set up wait only for once.

For Example:

* **sleep(5000)**– It will wait for 5 seconds. It is executed only once, where the command is written.
* **setSpeed(“5000”)**– It also will wait for 5 seconds. It runs each command after setSpeed delay by the number of milliseconds mentioned in set Speed.

**34) What are the different types of navigation commands?**

* **navigate().back()** command takes user back to the previous webpage in the web browser’s history. An example: driver.navigate().back();
* **navigate().forward()** lets the user to navigate to the next web page with reference to the browser’s history. An example: driver.navigate().forward();
* According to **navigate().refresh()** command user can refresh the current web page there by reloading all the web elements. An example: driver.navigate().refresh();
* User can launch a new web browser window and navigate to the specified URL by executing **navigate().to()** An example: **driver.navigate().to(“https:// thinkmobiles.com/”);**

**35)** **What is the difference between findElement () and findElements ()?**

Both of them let user to find elements in the current web page matching to the specified locator value. But if you use **findElement()**, only the first matching element would be fetched. An example:

**WebElement element = driver.findElements(By.xpath(“//div[@id=’example’]//ul//li”))**

If you use **findElements()**, the all matching elements would be fetched and stored in the WebElements list. An example:

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

**36)** **Can Selenium handle Windows based pop–up? ----**Windows pop-ups cannot be handled by using Selenium. Because it supports only web application testing.

**37) How can we handle Web-based pop-up?**There are four methods of the 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

**38) Do you know a way to refresh browser by using Selenium?** The list of commands to refresh a page in Selenium:

* navigate().refresh()
* getCurrentUrl()
* navigate().to(driver.getCurrentUrl())
* sendKeys(Keys.F5)

**39) How can we maximize browser window in Selenium?**

**driver.manage().window().maximize(); //command is used to maximize browser window in Selenium**

**40) How can we find the value of different attributes like name, class, value of an element?**

**getAttribute(“{attributeName}”) //method is used find the value of different attributes of an element**

**41) Could cookies be deleted in Selenium?**

**driver.manage().deleteAllCookies(); //command is used for deleting all cookies**

**42) How to perform right click using Selenium WebDriver?**

The next Actions class is used to perform right click:

**Actions act = new Actions(driver); // where driver is WebDriver type**

**act.moveToElement(webElement).perform();**

**act.contextClick().perform();**

**43)  How do perform drag and drop using Selenium WebDriver?**

The next Actions class is used to perform drag and drop:

**Actions builder = new Actions(driver);**

**Action dragAndDrop = builder.clickAndHold(SourceElement)**

**moveToElement(TargetElement)**

**release(TargetElement)**

**build();**

**dragAndDrop.perform();**

**44) How to check if an element is visible on the page?**

The return method type is logical. If it returns true then element is visible otherwise it is not.**isDisplayed()** method could be used for it:

**driver.findElement(By.id(“id\_of\_element”)).isDisplayed();**

**45)** **How to check if a button is enabled on the page? isEnabled()** method could be used for it:

**driver.findElement(By.id(“id\_of\_element”)).isEnabled();**

**46) What kind of mouse actions can be performed in Selenium?** Selenium supports different mouse actions, such as:

* click(WebElement element)
* contextClick(WebElement element)
* doubleClick(WebElement element)
* mouseUp(WebElement element)
* mouseDown(WebElement element)
* mouseMove(WebElement element)
* mouseMove(WebElement element, long xOffset, long yOffset)

47) Can you write the code to double click an element in Selenium?

Code to double click an element in Selenium:

Actions action = new Actions(driver);

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

action.doubleClick(element).perform();

**48) How to mouse hover an element in Selenium?**

Code to mouse hover over an element in Selenium:

Actions action = new Actions(driver);  
WebElement element=driver.findElement(By.id(“elementId”));

action.moveToElement(element).perform();

**49) What kind of keyboard operations can be performed in Selenium?**

Selenium lets to perform different kinds of keyboard operations, such as:

* **.pressKey(“non-text keys”)** is used for keys like control, function keys etc that are non-text
* **.releaseKey(“non-text keys”)** is used in conjunction with key press event to simulate releasing a key from keyboard event
* **.sendKeys(“sequence of characters”)** is used for passing character sequence to an input or textbox element.

**50) What is JUnit? And what is JUnit Annotation?**

JUnit is an open source Java applications testing framework, introduced by Apache. A process of adding a special form of syntactic metadata to Java source code is called annotation. JUnit Annotations are: variables, parameters, packages, methods and classes.

**51) What is TestNG and why is it better than JUnit?**

TestNG is a testing framework inspired from JUnit and NUnit in a way to use the merits by both the developers and testers.  Here are some new functionalities that make it more powerful and easier to use, such as:

* test that your code is multithread safe
* support for data-driven testing
* support for parameters
* a variety of tools and plug-ins support (Eclipse, IDEA, Maven, etc…)
* default JDK functions for runtime and logging
* dependent methods for application server testing
* flexible test configuration

**58) Can we find all links on a web page? --**As all  links are of anchor tag ‘a’, so we can find all of them on a web page by locating elements of tagName ‘a’:

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

**59) How can we capture screenshots in Selenium?**

We can take the screenshots in selenium by using getScreenshotAs method of **TakesScreenshot** interface:

**File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);**

**FileUtils.copyFile(scrFile, new File(“C:\\screenshot1.jpg”));**

**60) Explain how colors could be handled in Selenium WebDriver?**

To handle colors could be handled in Selenium WebDriver by using **Use getCssValue(arg0)** function to get the colors by sending ‘color’ string as an argument.

**61) How many exceptions do you know in Selenium WebDriver?** There are 5 different exceptions Selenium WebDriver:

* NoAlertPresentException,
* NoSuchElementException
* NoSuchWindowException
* TimeoutException
* WebDriverException

**62) How will you use Selenium to upload a file?**

File uploading action could be performed by using **element.sendKeys("path of file")** on the webElement of input tag and type file: **< name="fileUpload" type="file" />**

**52) What kinds of annotations are used in TestNG?** The following kinds of annotations are used in **TestNG**:

* Test
* BeforeSuite
* AfterSuite
* BeforeTest
* AfterTest
* BeforeClass
* AfterClass
* BeforeMethod
* AfterMethod

**53) How to set test case priority in TestNG?**

TestNG “Priority” is used to schedule the test cases. In order 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.

**import org.testng.annotations.Test;  
public class testNGPriorityExample {**

**@Test  
public void registerAccount() {**

**System.out.println(“Create an account”); }**

**@Test(priority=2)  
public void sendEmail(){**

**System.out.println(“Confirm your email”); }**

**@Test(priority=1)  
public void login () {**

**System.out.println(“Execute login after confirmation”); }}  
------------------------------------------------------------------------**

**import org.testng.annotations.Test;  
public class testNGPriorityExample {**

**@Test**

**public void registerAccount() {**

**System.out.println(“Create an account”); }**

**@Test(priority=2)  
public void sendEmail() {**

**System.out.println(“Confirm your account”); }**

**@Test(priority=1)  
public void login() {**

**System.out.println(“Execute login after confirmation”); } }**

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

You have to follow the next steps to find broken images in a page using Selenium Web driver:

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

**55) Can captcha and bar code reader be automated by using Selenium?**

Neither captcha, no bar code reader can be automated by using Selenium.

**56) How to verify tooltip text using Selenium?**

The tooltip text in Selenium could be verified by fetching the value of ‘title’ attribute. An example:

**String toolTipText = element.getAttribute(“title”);**

**57) How to locate a link using its text in Selenium?**

**linkText()** and **partialLinkText()** are used for link location.

The examples:

**WebElement link1 = driver.findElement(By.linkText(“some\_link\_test”));**

**WebElement link2 = driver.findElement(By.partialLinkText(“some\_link\_part\_text”));**

**65) How to execute JavaScript in Selenium?**

**JavaScriptExecuter** is used for JavaScript execution in Selenium. A simple example:

**WebDriver driver = new FireFoxDriver();**

**if (driver instanceof JavascriptExecutor) {**

**((JavascriptExecutor)driver).executeScript(“{JavaScript Code}”); }**

**74)** **Explain how you can use recovery scenario with Selenium?**

You can use recovery scenario in accordance with the programming language.

If it is Java then you can use exception handling to overcome same.

**63) What is Robot API?**

Robot API is used to control keyboard or mouse to interact with OS windows like Download pop-up, Alerts, Print Pop-ups, etc. or native Operation System applications like Notepad, Skype, Calculator, etc.

**64) What methods of Robot Class do you know?** Some commonly and popularly used methods of Robot Class during web automation:

* **keyPress():** method with press down arrow key of Keyboard

Example:

**keyPress(KeyEvent.VK\_DOWN)**

* **keyRelease():**method with release down arrow key of Keyboard:

Example:

**robot.keyRelease(KeyEvent.VK\_DOWN)**

* **mouseRelease()**method will release the right click of your mouse

Example:

**mouseRelease(InputEvent.BUTTON3\_DOWN\_MASK)**

* **mouseMove()** method will move mouse pointer to the specified X and Y coordinates.

Example:

**robot.mouseMove(point.getX(), point.getY())**

* **mousePress()** method will press the right click of your mouse. Example :

**robot.mousePress(InputEvent.BUTTON3\_DOWN\_MASK)**

**66) Which package can be imported while working with WebDriver?**

**org.openqa.selenium java -cp bin;jars/\* org.testng.TestNG testng.xml**

**67) How do you get the width of the textbox?**

You can get the width of the textbox by using the following command:

driver.findElement(By.xpath(“xpath of textbox ”)).getSize().getWidth();

driver.findElement(By.xpath(“xpath of textbox ”)).getSize().getHeight()

**68) Which web driver implementation is the fastest?**

The fastest **WebDriver** is **HtmlUnitDriver**. Differing of other drivers (FireFoxDriver, ChromeDriver etc), it’s non-GUI, while running no browser gets launched.

**69) What is the purpose of deSelectAll() method?**

It is used to deselect all the options which have been selected from the drop-down list.

**70) How can you switch back from a frame?**

**defaultContent()** method is used to switch back from a frame.

**71) How to login into any site if it’s showing any authentication pop-up for username and password?**

You should pass the username and password with URL:

https://username:password@url

https://creds:test@www.test.com

**72) What is the purpose of getOptions() method?**

**getOptions()** is used to get the selected option from the drop-down list.

**73) What is the difference between getWindowHandles() and getWindowHandle()?**

You can get the browser address using these commands. But if you use getWindowHandle(), you’ll get the address of the current browser where the control is and return type is a string. So, if you use getWindowHandles(), you will get the address of all the open browser and its return type is an iterator.

**75)** **How do you send ENTER/TAB keys in WebDriver?**

use **click()** or submit() methods are used for ENTER. But, don’t forget that **submit()** method is used only if type=’submit’.

You can use Actions class **act.sendKeys(Keys.ENTER) for** TAB.

**76) How to Handle Alerts in Selenium WebDriver?**

Here are a few methods of Alerts handling which are widely used in Selenium Webdriver.

* **void dismiss()** is used to click on the ‘Cancel’ button of the alert.

**driver.switchTo().alert().dismiss();**

* **void accept()** is used to click on the ‘OK’ button of the alert.

driver.switchTo().alert().accept();

* **String getText()** is used to capture the alert message.

**driver.switchTo().alert().getText();**

* **void sendKeys(String stringToSend)**is used to send some data to alert box.

**driver.switchTo().alert().sendKeys(“Text”);**

**77) What is a data-driven framework?**

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).

**78) What is a keyword-driven framework?**

The keyword driven framework is a methodology where actions or steps are treated as keywords. These keywords (like click, move, type etc.,) are stored in some external repositories along just like data **(in .csv/.json/.xls/DB).**

**79) What is the hybrid framework?**

The combination of data driven and keyword driven framework is called the hybrid. Here the operations/instructions/keywords in a separate repository (.csv/.xls/.json/DB) and data is in separate (.csv/.xls/.json/db from data provider) and the tests/driver would read both and perform the actual tests automatically. In this design, we get the best of both methodologies, and it is kind of practical in most of the automation cases.

**80) What are the main advantages of Selenium Grid?**

Selenium Grid has following advantages: multi-browser testing, parallel test case execution, multi-platform testing.

**81) What is a hub in Selenium Grid?**

Selenium Grid hub is a central point or a server that controls the test executions on the different machines.

**82) What is a node in Selenium Grid?**

Selenium Grid node is a hub attached machine, which has instances running the test scripts. Unlike a hub, there can be more than one nodes in Selenium Grid.

**83) Could you explain the line of code Webdriver driver = new FirefoxDriver();.**

‘WebDriver’ is an interface and Firefox driver is a class which implements Web driver interface. That why we can write in this way **Web driver driver = new FirefoxDriver();.**

-we are creating an object of type WebDriver instantiating an object of Firefox Driver class.

**84) What is the purpose of creating a reference variable- ‘driver’ of type WebDriver instead of directly creating a FireFoxDriver object or any other driver’s reference in the statement Webdriver driver = new FirefoxDriver();?**

We can use the same variable to work with multiple browsers like ChromeDriver, IEDriver by creating a reference variable of type WebDriver.

**85) How can you create HTML test report from your test script?**

There are three ways of HTML test report creation:

* using inbuilt default.html to get the HTML report in TestNG
* with the ANT help in JUnit
* using XSL jar for converting XML content to HTML in own customized reports

**86) What could be the cause of Selenium WebDriver test to fail?**

There are some causes of Selenium WebDriver test to fail: because of time out.

* SeleniumWebDriver element waiting to access did not appear on the web page and the operation timed out.
* SeleniumWebDriver is trying to access not created element
* SeleniumWebDriver cannot locate the element, because the locator has been changed

**87) Explain how can you debug the tests in Selenium IDE?**

The tests could be debugged in such way:

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

**88) What is the testng.xml file used for?**

testng.xml file is used to configure the whole a 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.

**89) What is the difference between @Factory and @DataProvider annotation?**

**@DataProvider** is concerned to 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.

**90) In which format does source view show the script in Selenium IDE?**

The script is shown by Selenium IDE source view in XML format.

**91) How could AJAX controls be handled in WebDriver?** 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:

* ThreadSleep- java
* Implicit Wait- full page
* Explicit Wait-
* WebdriverWait
* Fluent Wait

AJAX -Communication between data and server without refreshing the page.

Explicit wait –

//Explicit wait-

WebDriverWait wait = **new** WebDriverWait(dr, 20);// dr= this class driver

wait.until(ExpectedConditions.*elementToBeClickable*(By.*xpath*("")));// for single element

// wait for expected condition - for this elememt - until clickable

* Fluent Wait

**92) What is the FirefoxDriver, class or an interface? And which interface does it implement?**

FirefoxDriver is a Java class which implement webdriver interface.  It implements all the methods available in the interface.

**93) How can we make one test method dependent on other using TestNG?**

We can make one test method run only after successful execution of dependent test method by using dependsOnMethods parameter inside @Test annotation in TestNG: @Test(dependsOnMethods = { “preTests” })

**94) How could you** **explain the main difference between WebDriver and RC?**

Selenium WebDriver drives the browser using built-in support. RC injects JavaScript function into browsers when the page is loaded.

**95) What is IntelliJ?**

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.

**96) What are the advantages of Using Git Hub For Selenium?**

* Members of multiple people team working on the same project can update its details and inform other team members simultaneously
* You can build the project from the remote repository regularly by using Jenkins. This helps you to keep track of failed builds.

**97) Can we use Selenium RC for tests driving on two different browsers on one operating system without Selenium Grid?**

We can do it if JAVA testing framework is not used. If we use Java client driver of Selenium, instead of using Java testing framework, TestNG allows us not to use Selenium Grid.

**98) How can we run test cases in parallel using TestNG?**

You should just to add these two key value pairs in the suite to run the tests in parallel:

**parallel="{methods/tests/classes}"  
thread-count="{number of threads you want to run simultaneously}".**

**99) How would you test your own element locator?**

**“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 **“FirePath”** plugin in FireFox

**100) When AutoIT is used?**

AutoIT is used to handle window GUI and non-HTML popups in the application.

**101) What API is required** **for Database Testing in Selenium WebDriver?**

JDBC (Java Database Connectivity) API is **required** **for Database Testing in Selenium WebDriver.**

**102)** What Java API is required for generating pdf reports?

Java API IText is required for generating pdf reports.

**103) Explain why to choose Python over Java in Selenium.**

Here are some points that favor Python over Java to use with Selenium:

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

**104) How can you handle network latency using Selenium?**

You can use **driver.manage().timeouts().pageLoadTimeout();** for network latency

**105) How can you run Selenium Server other than the default port 4444?**

Selenium server could be run on java-jar selenium-server.jar-port other than its default port.

**106) Explain how you can capture server side log Selenium Server?**

To capture server side log in Selenium Server, you can use command  
java –jar .jar –log selenium.log,

**Java Jar -SeeLinux command**

Question 54

-Explain how you can find Broken Images in a page using selenium WebDriver?

a) get Xpath and get the all links on the page using the tag name

b)Click on every link on the page

c) perform a get response on the image and grabs its responds status code.

d) Assert that the status code is equal 400 or more / 500. Then it would be Broken image.

**Coding-**

package brokenImg;

import java.io.IOException;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.net.URLConnection;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

public class BrokenImage {

WebDriver dr;

public void img() throws IOException, InterruptedException {

List<WebElement> ele = dr.findElements(By.tagName("//img"));

for(WebElement e:ele) {

String link=e.getAttribute("src");//src - page link // take each attribute

// URL - class

URL url = new URL(link); // in every page seprate url address.I get all url in this class

// HttpURLConnection is interface and openConnection(); method

// to open url / link

URLConnection con=(HttpURLConnection)url.openConnection();

con.connect(); // to connect

Thread.sleep(4000);

//we need status code for broken image // below 400 is good

int code =((HttpURLConnection) con).getResponseCode(); // combine with get respose code

**if(code>= 400 )** {

System.out.println("broken img");

}else {

System.out.println("No broken img");}}}}

=======================================================

**Questions 2**

**Can captcha and bar code reader can be automated by using selenium?**

Neither captcha, no bar code reader can be automated by using Selenium. By 3rd party tool by z-xing.

**Questions—3 How to verify tooltip text using selenium?**

**The tooltip text in selenium could be verified by fetching the value of ‘title’ attribute. [element.get ]**

**Example -toolTipText = element.getAttribute(“title”);**

Questions 4- How to locate a link using its text in selenium?

linkText() and PartialLinkText() are used for link location.

Example –

Link1=dr.findElement(By.linkText(“some\_link\_test”));

Link2= dr.findElement(By.partialLinkText(“some\_link\_ partialLinkText\_ test”));

**Ques- 5**

**Can we find all links on a web page?**

**As all links are of anchor tag ‘a’, So we can find all of them on a web page by locator tagName ‘a’.**

**Links= dr.findElements(By.tagName(“a”));**

**Ques- 6 How we can capture screenshorts in selenium?**

1. **Covert webDriver object to TakeScreenshot**
2. **Call getScreenshotsAs method to create image file**
3. **Create object of TakesScreenshots interface with WebDriver instance**
4. **Call getScreenshotsAs() using ref of TakesScreenshots interface to take screenshots.**

**we can capture screenshorts in selenium by using getStringshotsAs().**

**Ques- 7 How do perform drag and drop using selenium webdriver?**

Question -8

How to check if an element is visible on the page?

The return method type is logical. If it returns true then element is visible otherwise it is not. isDisplayed() method could be used for it.

**Ques-9**

How to check if a button is enabled on the page?

isEnabled() method could be used for it:

dr.findElement(By.id(“id\_of\_element”)).isEnabled();

Question-10

**What kind of mouse actions can be performed in selenium?**

**Selenium supports different mouse actions, such as :**

* **Click(WebElement element)**
* **contextClick(WebElement element)**
* **doubleClick(WebElement element)**
* **mouseUp(WebElement element)**
* **mouseDown(WebElement element)**
* **mouseMove(WebElement element)**
* **mouseMove(WebElement element, long xOffset,long yOffset)**

**Ques-11 -Can you write the code to double click an element in selenium?**

Code to double click an element in selenium. Create an Actions Class object.

Actions ac = new Actions();

**Ques-12 -Explain how colors could be handled in selenium webDriver?**

**Handle colors could be handled in selenium by using getCssValue(args)**

**Function to get the colors by sending ‘color ’ string as an argument.**

Ques-13

How many Exceptions do you know in selenium webdriver?

1. **ElementNotSelectableException**: An element is disabled (can not be clicked/selected) in spite of being present in the DOM
2. **NoSuchElementException**: Webdriver is not able to determine the elements during runtime, i.e., the *FindBy* method cannot find a particular component
3. **NoSuchFrameException**: Webdriver attempts to switch to an invalid frame, which is unavailable
4. **NoAlertPresentException**: Webdriver is trying to switch to an invalid alert, which is unavailable
5. **NoSuchWindowException**: Webdriver is trying to switch to an invalid window, which is unavailable
6. **StaleElementReferenceException**: The referenced element is no longer present on the DOM page (a reference to a component is now Stale). For example, the item belongs to a different frame than the current one or the user has navigated away to another page
7. **SessionNotFoundException**: Webdriver is acting immediately after ‘quitting’ the browser
8. **TimeoutException**: The command did not complete in the specified time. For example, the element didn’t display at the specified time. This is especially encountered when working with waits
9. **WebDriverException**: Webdriver is acting immediately after ‘closing’ the browser

**Ques**-14

**How will you use selenium to upload a file ?**

**File uploading action could be performed by using element.sendKeys(“path of file”) on the Web Element of input tag and type file:< name = “fileUpload” type = “file”/>**

**Ques 15- What is Robot API?**

**Robot API is used to control keyboard or mouse to interact with OS windows like Download pop-up, Alerts, Print Pop-ups, etc or native Operation system application like Notepad,skype, Calculator, etc.**

**Ques-16 What methods of Robot class do you know ?**

**The following methods of Robot class is used during web automation –**

* keyPress(): **Example:** **robot.keyPress(KeyEvent.VK\_DOWN)** : This method with press down arrow key of Keyboard
* mousePress() : **Example** : **robot.mousePress(InputEvent.BUTTON3\_DOWN\_MASK)** : This method will press the right click of your mouse.
* mouseMove() : **Example: robot.mouseMove(point.getX(), point.getY())** : This will move mouse pointer to the specified X and Y coordinates.
* keyRelease() : **Example: robot.keyRelease(KeyEvent.VK\_DOWN)** : This method with release down arrow key of Keyboard
* mouseRelease() : **Example: robot.mouseRelease(InputEvent.BUTTON3\_DOWN\_MASK)** : This method will release the right click of your mouse

**Question -17 How to execute java script in selenium?**

**JavaScriptExecuter is used for javascript execution in selenium.**

**SoftAssert – always give pass the test cases. Two weeks behind 6 ,7 and 13. If we test mannully then this test are passed, then we use soft assert. Because I did functional test before. If ‘I confused we use assertAll().**

**If actual and expected is not equal then it will be failed . then stoped every things. One line problem then it will no go the next case.**

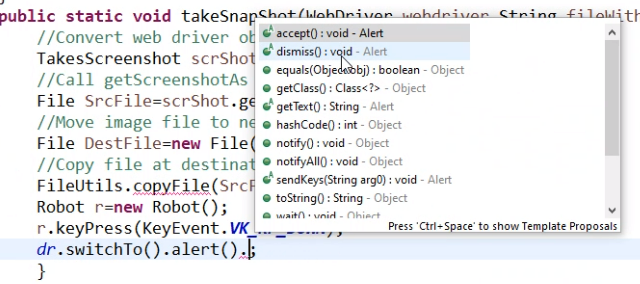
**Verify it will go to next execution.**

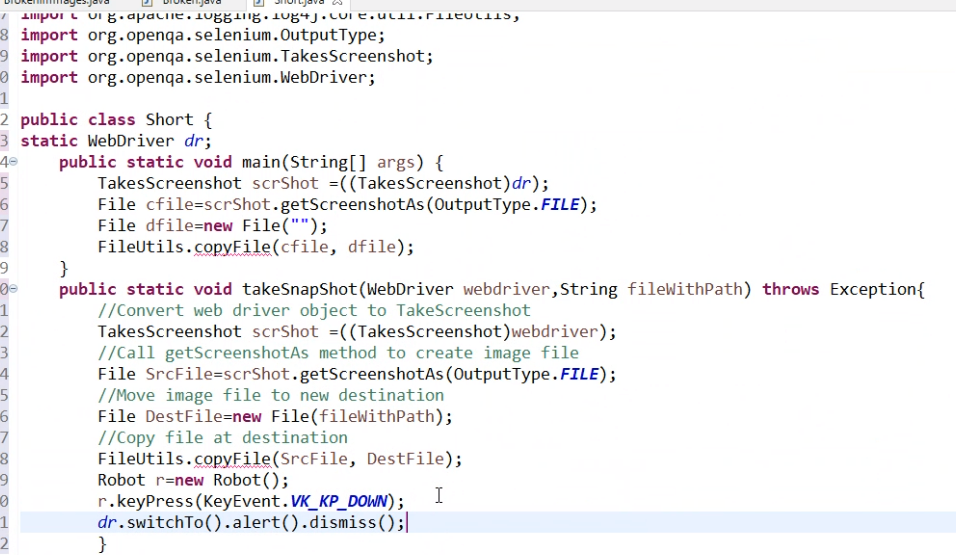
**How can I handle web base pop up ?**

**Three types pop up.**

**1) If it is build usding HTMLpop up-use click()**

**2)If it is build -Java script pop up-**

1. **dr,switchTo.allert()**
2. 
3. **Windows base pop -use Robot class in java to handle the pop up . in this case selenium is not working.**



**Refresh –in case of re**

**Dr.navogae ()**

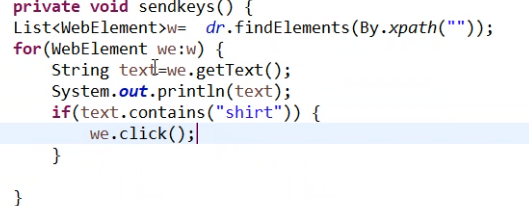
**Dr.current ()**

**Selenium can delete cookies using two mehtos**

**Delete single bowser**

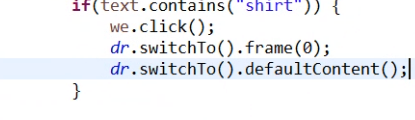
**Delete all opwn browser**

**In Dynamic within list<webElements> to check that this element is available or not / I want to click on the specific shirts then I will use this**



**Back from frame to page**

**Dr. switchTodefault conta**

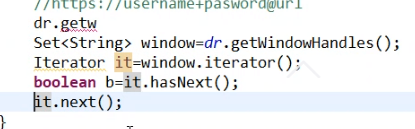


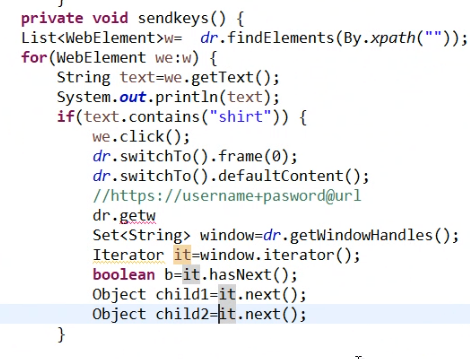
**Authencation pop up in the URL**



**For single window handle- use windowhandle() it return -string**

**For multiple window handle- use windowhandle() it return -set<string>**





**Q- How you recovery sec with sele**

**I forgot to give value like that string s; here is no value .**