**Selenium Interview Question and Answer for Fresher and Experienced Part -1 (Basics of Selenium)**

**Question 1 - What is Selenium?**

Ans- Selenium is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers.

Selenium is a set of different software tools. Each tool has a different approach in supporting web-based automation testing.

It has four components namely,

i Selenium IDE (Integrated Development Environment)

ii Selenium RC (Remote Control) – selenium 1

iii Selenium WebDriver – selenium 2 &amp; 3

iv Selenium Grid

**Question 2- What is Selenium IDE?**

Ans- Selenium IDE (Integrated Development Environment) is a Firefox plugin. It is the simplest framework in the Selenium Suite. It allows user to record and playback the scripts. Even though we can create scripts using Selenium IDE, we need to use Selenium RC or Selenium WebDriver to write more advanced and robust test cases.

**Question 3- What is Selenese?**

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

**Question 4- Which browser supports Selenium IDE to be used?**

Ans- Firefox

**Question 5- What is Selenium RC?**

Ans- Selenium 1. Selenium RC was the main Selenium project for a long time before the

WebDriver merge brought up Selenium 2. Selenium 1 is still actively supported .It relies on JavaScript for automation. It supports Java, Javascript, Ruby, PHP, Python, Perl and C#. It supports almost every browser out there.

**Question 6- What is Selenium WebDriver?**

Selenium WebDriver (Selenium 2) is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it.

Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.

**Question 7- What is Selenium Grid?**

Selenium Grid is a tool used together with Selenium RC to run tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems.

it is used to distribute your test execution on multiple platforms and environments concurrently.

Use- Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently to achieve distributed test execution

**Question 8-What are the types of WebDriver APIs available in Selenium?**

Ans- WebDriver APIs are-

• Firefox Driver

• Gecko Driver

• InternetExplorer Driver

• Chrome Driver

• HTMLUNIT Driver

• Opera Driver

• Safari Driver

• Android Driver

• iPhone Driver

• EventFiringWebDriver

**Question 9- Which WebDriver implementation claims to be the fastest?**

Ans- The fastest implementation of WebDriver is the HTMLUnitDriver. It is because the HTMLUnitDriver does not execute tests in the browser.

**Question 10 - What are the Programming Languages supported by Selenium WebDiver?**

Ans-

• Java

• C#

• Python

• Ruby

• Perl

• PHP

**Question 11- What are the Operating Systems supported by Selenium WebDriver?**

Ans- Below list-

• Windows

• Linux

• Apple

**Question 12- What are the Open-source Frameworks supported by Selenium WebDriver?**

Ans – Below List

• JUnit

• TestNG

• CUCUMBER

• JBHEAVE

**Question 13- What are the Locators available in Selenium?**

Ans-Different types of locators are:

1 ID

2 ClassName

3 Name

4 TagName

5 LinkText

6 PartialLinkText

7 XPath

8 CSS Selector

**Question 14- What is a XPath?**

Ans - XPath is used to locate the elements. Using XPath, we could navigate through elements and attributes in an XML document to locate web elements such as textbox, button, checkbox, Image etc., in a web page.

**Question 15- How to submit a form using Selenium WebDriver?**

**Ans**- We use “submit” method on element to submit a form

driver.findElement(By.id(form\_1&quot;)).submit();

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

**Question 16**. **How to press ENTER key on text box In Selenium WebDriver?**

**Ans**- driver.findElement(By.xpath(&quot;xpath&quot;)).sendKeys(Keys.ENTER);

**Question 17. How to pause a test execution for 5 seconds at a specific point?**

**Ans**- By using Thread.sleep(long milliseconds) method we could pause the execution for a specific time.

Thread.sleep(5000)

**Question 18- Is Selenium Server needed to run Selenium WebDriver Scripts?**

**Ans**- When we are distributing our Selenium WebDriver scripts to execute using Selenium Grid, we need to use Selenium Server.

**Question 19- What are the different exceptions you have faced in Selenium WebDriver?**

**Ans-**

• WebDriverException

• TimeoutException

• NoAlertPresentException

• NoSuchWindowException

• NoSuchElementException

• StaleElementReferenceException

• IllegalStateException

**Question 20- How To Login Into Any Site If It Is Showing Any Authentication Pop-Up For Username And Password?**

**Ans**- We pass username and password with the URL

Syntax- http://username:password@url

e.g. http://admin:admin12@abc.com

**Question 21- What are the types of waits available in Selenium WebDriver?**

**Ans**- In Selenium we could see three types of waits such as Implicit Waits, Explicit Waits and Fluent Waits.

• Implicit Waits –

• Explicit Waits –

• Fluent Waits –

• PageLoadTimeOut

• Thread.sleep() –

**Question 22- What is Implicit Wait In Selenium WebDriver?**

Ans- Implicit waits tell to the WebDriver to wait for a certain amount of time before it throws an exception. Once we set the time, WebDriver will wait for the element based on the time we set before it throws an exception.

The default setting is 0 (zero). We need to set some wait time to make WebDriver to wait for the required time.

**Question 23- What is WebDriver Wait In Selenium WebDriver?**

**Ans**- WebDriverWait is applied on a certain element with defined expected condition and time. This wait is only applied to the specified element. This wait can also throw an exception when an element is not found.

**Question 24- What is Fluent Wait In Selenium WebDriver?**

**Ans**- FluentWait can define the maximum amount of time to wait for a specific condition and frequency with which to check the condition before throwing an “ElementNotVisibleException” exception.

**Question 25- What is the difference between Absolute Path and Relative Path?**

**Ans**- Absolute XPath starts from the root node and ends with desired descendant element’s node. It starts with top HTML node and ends with input node. It starts with a single forward slash(/) as shown below.

/html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input

Relative XPath starts from any node in between the HTML page to the current element’s node(last node of the element). It starts with a double forward slash(//) as shown below.

Syntax- //tagname[@attribute=’value’]

**Question 26- What is the difference between Assert and Verify in Selenium?**

**Ans**- Assert: In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.

Verify: In simple words, there won’t be any halt in the test execution even though the verify condition is true or false.

For detailed post check the below link.

**Question 27- What are Soft Assert and Hard Assert in Selenium?**

**Ans**- Soft Assert: Soft Assert collects errors during @Test Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement

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

**Question 28- What are the verification points available in Selenium?**

In Selenium IDE, we use Selenese Verify and Assert Commands as Verification points

In Selenium WebDriver, there is no built-in features for verification points. It totally depends on our coding style. some of the Verification points are To check for page title To check for certain text

To check for certain element (text box, button, drop down, etc.)

**Question 29- How to launch a browser using Selenium WebDriver?**

**Ans**- WebDriver is an Interface. We create Object of a WebDriver Interface.

To launch Firefox Driver: WebDriver driver = new FirefoxDriver();

To launch Chrome Driver: WebDriver driver = new ChromeDriver();

To launch Internet Explorer Driver: WebDriver driver = new InternetExplorerDriver();

**Question 30- Is the FirefoxDriver a Class or an Interface?**

**Ans**- FirefoxDriver is a Java class, and it implements the WebDriver interface.

**Question 31- What is the super interface of WebDriver?**

**Ans**- SearchContext.

**Question 32- Explain the line of code Webdriver driver = new FirefoxDriver(); ?**

**Ans**- ‘WebDriver‘ is an interface and we are creating an object reference of type WebDriver instantiating an object

of FirefoxDriver class.

**Question 33- We do create a reference variable ‘driver’ of type WebDriver**

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

instead of creating

FirefoxDriver driver = new FirefoxDriver();

1 WebDriver driver = new FirefoxDriver();

………FirefoxDriver driver = new FirefoxDriver();

**Question 34- How to input text in the text box using Selenium WebDriver?**

**Ans**- By using sendKeys() method

WebDriver driver = new FirefoxDriver();

driver.get(https://www.gmail.com&quot;);

driver.findElement(By.xpath));

**Question 35- How to input text in the text box without calling the sendKeys()?**

// To initialize js object

JavascriptExecutor JS = (JavascriptExecutor)driver;

// To enter username

JS.executeScript(&quot;document.getElementById(‘User&#39;).value=test.com&#39;&quot;);

**Question 36- How to clear the text in the text box using Selenium WebDriver?**

**Ans** - By using clear () method

WebDriver driver = new FirefoxDriver();

driver.get(&quot;https://www.gmail.com&quot;);

driver. findElement(By.xpath(&quot;xpath\_of\_element1&quot;)).sendKeys(&quot;Software Testing Material Website&quot;);

driver.findElement(By.xpath(&quot;xpath\_of\_element1&quot;)).clear();

**Question 37- How to get a text of a web element?**

**Ans**- By using getText() method

**Question 38- How to get an attribute value using Selenium WebDriver?**

By using getAttribute(value);

**Question 39- How to click on a hyperlink using Selenium WebDriver?**

**Ans**- We use click() method in Selenium to click on the hyperlink

driver.findElement(By.linkText(“Software Testing Material Website”)).click();

1 driver.findElement(By.linkText(“Software Testing Material Website”)).click();

**Question 40-How to handle hidden elements in Selenium WebDriver?**

**Ans** - It is one of the most important selenium interview questions.

We can handle hidden elements by using JavaScriptExecutor

(JavascriptExecutor(driver)).executeScript(document.getElementsByClassName(ElementLocator).click());

**Question 41- What are the different types of navigation commands?**

**Ans- Below commands are navigation commands-**

1. driver.navigate().to("URL");
2. driver.navigate().forward();
3. driver.navigate().back();
4. driver.navigate().refresh();

**Question 42. How can we maximize browser window in Selenium?**

**Ans**- To maximize browser window in selenium we use maximize() method. This method maximizes the current window if it is not already maximized

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

**Question 43-What is the difference between driver.get() and driver.navigate.to(“url”)?**

**Ans**- driver.get(): To open an URL and it will wait till the whole page gets loaded

driver.navigate.to(): To navigate to an URL and It will not wait till the whole page gets loaded

**Question 44- What happens if I run this command. driver.get(“www.facebook.com”);**

**Ans**- An exception is thrown. We need to pass HTTP protocol within driver.get() method.

driver.get(&http://www.softwaretestingmaterial.co);

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

**Question 45- What is the alternative to driver.get() method to open an URL using Selenium WebDriver?**

**Ans**- Alternative method to driver.get(“url”) method is driver.navigate.to(“url”)

**Question 46- Can I navigate back and forth in a browser in Selenium WebDriver?**

**Ans**- We use Navigate interface to do navigate back and forth in a browser. It has methods to move back, forward as well as to refresh a page.

driver.navigate().forward(); – to navigate to the next web page with reference to the browser’s history

driver.navigate().back(); – takes back to the previous webpage with reference to the browser’s history

driver.navigate().refresh(); – to refresh the current web page thereby reloading all the web elements

driver.navigate().to(“url”); – to launch a new web browser window and navigate to the specified URL

**Question 47- How to delete cookies in Selenium?**

**Ans**- To delete cookies we use deleteAllCookies() method

driver.manage().deleteAllCookies();

**Question 48- What are the ways to refresh a browser using Selenium WebDriver?**

**Ans**- There are multiple ways to refresh a page in selenium

• Using driver.navigate().refresh() command as mentioned in the question

• Using driver.navigate().to(“URL”) on the current URL or driver.navigate().to(driver.getCurrentUrl());

• Using sendKeys(Keys.F5) on any textbox on the webpage

**Question 49- What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in Selenium WebDriver?**

**Ans**-

driver.getWindowHandle() – It returns a handle of the current page (a unique identifier)

driver.getWindowHandles() – It returns a set of handles of the all the pages available.

**Question 50- What is the difference between driver.close() and driver.quit() methods?**

**Ans**- Purpose of these two methods (driver.close and driver.quit) is almost same. Both allow us to close a browser but still, there is a difference.

driver.close(): To close current WebDriver instance

driver.quit(): To close all the opened WebDriver instances

**Question 51- What is the difference between driver.findElement() and driver.findElements() commands?**

**Ans**- The difference between driver.findElement() and driver.findElements() commands is-

• findElement() returns a single WebElement (found first) based on the locator passed as parameter.

Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.

• Syntax of findElement()-

• WebElement textbox = driver.findElement(By.id(“textBoxLocator”));

• Syntax of findElements()-

• List <WebElement> elements = driver.findElements(By.id(“value”));

• Another difference between the two is- if no element is found then findElement() throws

NoSuchElementException whereas findElements() returns a list of 0 elements.

List&lt;WebElement&gt; list = driver.findElements(By.tagName(“a”));

**Question 52- How to find whether an element is displayed on the web page?**

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

1 isDisplayed()

2 boolean elePresent = driver.findElement(By.xpath(&quot;xpath&quot;)).isDisplayed();

3 isSelected()

4 boolean eleSelected= driver.findElement(By.xpath(&quot;xpath&quot;)).isSelected();

5 isEnabled()

6 boolean eleEnabled= driver.findElement(By.xpath(&quot;xpath&quot;)).isEnabled();

**Question 53 How to select a value in a dropdown?**

**Ans**- By using Select class

WebElement mySelectElement = driver.findElement(By.name(“value”));

Select dropdown = new Select(mySelectElement);

dropdown.selectByVisibleText(Text);

dropdown.selectByIndex(Index);

dropdown.selectByValue(Value);

**Question 54- How to capture Screenshot in Selenium WebDriver?**

**Ans**- By using TakesScreenshot Interface

TakesScreenshot scrShot =((TakesScreenshot)webdriver);

File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

**Question 55-How to mouse hover on a web element using WebDriver?**

**Ans**- By using Actions class

WebElement ele = driver.findElement(By.xpath(&quot;xpath&quot;));

//Create object action of an Actions class

Actions action = new Actions(driver);

//Mouseover on an element

action.moveToElement(ele).build().perform();

.

**Question 56- How can we handle web based pop-up?**

Ans- To handle alerts popups we need to do switch to the alert window and call Selenium WebDriver Alert API methods.

Alert al=driver.switchTo.alert();

al.dismiss();

al.accept();

al.getText();

al.sendKeys("Text");

**Question 57- How can we handle windows-based pop up?**

Ans- Selenium doesn’t support windows-based applications. It is an automation testing tool which supports only web application testing. We could handle windows-based popups in Selenium using some third party tools such as AutoIT, SIKULI, Robot class etc.

**Question 58- How to find more than one web element in the list?**

Ans- List<WebElement> list= driver.findElements(By.xpath(“ ”));

// To fetch the size of the list

int listSize = list.size();

//for loop

for (int i=0; i&lt; list Size; i++)

{

**Question 59- How to read a JavaScript variable in Selenium WebDriver?**

Ans- By using JavaScriptExecutor

// To initialize the JS object.

JavascriptExecutor JS = (JavascriptExecutor) webdriver;

// To get the site title.

String title = (String)JS.executeScript(&quot;return document.title&quot;);

System.out.println(“Title of the webpage : “ + title);

**Question 60- How do you read test data from excels?**

Ans- Test data can efficiently be read from excel using JXL or POI API. POI API has many advantages.

**Question 61- Is it possible to automate the captcha using Selenium?**

Ans- No, It’s not possible to automate captcha and bar code reader.

**Question 62- List some scenarios which we cannot automate using Selenium WebDriver?**

Ans-

1. Bitmap comparison Is not possible using Selenium WebDriver

2. Automating Captcha is not possible using Selenium WebDriver

3. We cannot read bar code using Selenium WebDriver

4. windows OS-based pop ups

5. third party calendars/element

6. Image

7. Word/PDF- we can use a java library called PDFBox. Apache PDFBox is an open-source library that exclusively helps in handling the PDF documents.

**Question 63- What is Object Repository in Selenium WebDriver?**

Ans- Object Repository is used to store element locator values in a centralized location instead of hard coding them within the scripts. We do create a property file (.properties) to store all the element locators and these property files act as an object repository in Selenium WebDriver.

**Question 64- How can you use the Recovery Scenario in Selenium WebDriver?**

Ans- By using “Try Catch Block” within Selenium WebDriver Java tests.

try {

driver.get(www.xyz.com);

}catch(Exception e){

System.out.println(e.getMessage());

}

**Question 65- How to Upload a file in Selenium WebDriver?**

Ans- There are two cases which are majorly used to upload a file in Selenium WebDriver such as using

SendKeys Method and using AutoIT Script.

Browser Button – type =“file”

SendKeys (c:\\test\\naveen.jpg);

**Question 66- How to Download a file in Selenium WebDriver?**

Ans- By using AutoIT script, we could download a file in Selenium WebDriver.

**Question 67- How to run Selenium WebDriver Test from the command line?**

Ans- Class A{}

cd c

c: javac A.java

c: java A.java

java org.testng.TestNG C:\Users \Desktop\ \workspace\testing\testng.xml

**Question 68- How to switch between frames in Selenium?**

Ans- By using the following code, we could switch between frames.

driver.switchTo().frame( name );

By using- name

Index

value

**Question 69- How to connect a Database in selenium?**

Ans- As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.

We use JDBC Driver to connect the Database in Selenium (While using Java Programming Language).

**Question 70- How To Resize Browser Window Using Selenium WebDriver?**

Ans- To resize the browser window to dimensions, we use ‘Dimension’ class to resize the browser window.

//Create object of Dimensions class

        Dimension d = new Dimension(480,620);

        //Resize the current window to the given dimension

        driver.manage().window().setSize(d);

**Question 71- How To Scroll Web Page Down Or UP Using Selenium WebDriver?**

Ans- JavaScript scrollBy() method scrolls the document by the specified number of pixels.

**Question 72- How To Perform Right Click Action (Context Click) In Selenium WebDriver?**

Ans- We use Actions class in Selenium WebDriver to do Right-Click (Context Click) action.

action.contextClick(driver.findElement(By.xpsjht()).build().perform();

**Question 73- How To Perform Double Click Action In Selenium WebDriver?**

Ans- We use Actions class to do Double click action in selenium.

**Question 74- How To Perform Drag And Drop Action in Selenium WebDriver?**

Ans- We use Actions class to do Drag And Drop Action

**Question 75- How To Highlight Element Using Selenium WebDriver?**

By using JavascriptExecutor interface we could highlight the specified element

**Question 76- How to fetch the current page URL in Selenium?**

Ans- To fetch the current page URL, we use driver.getCurrentUrl();

**Question 78- How can you find Broken Links in a page using Selenium WebDriver?**

**List<WebElement> list=driver.findElements(By.id(“a”));**

**Thank You!!**