**1. How do I launch the browser using WebDriver?**

**Answer:**

**Launching Firefox Browser**

Firefox is one of the most widely used browsers in automation. The following steps are required to launch the firefox browser.

1.Download geckodriver.exe from GeckoDriver Github Release Page. Make sure to download the right driver file based on your platform and OS version.

<https://github.com/mozilla/geckodriver/releases>

2.Set the System Property for “webdriver.gecko.driver” with the geckodriver.exe path – System.setProperty(“webdriver.gecko.driver”,”geckodriver.exe path”);

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

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

//Creating a driver object referencing WebDriver interface

WebDriver driver;

//Setting webdriver.gecko.driver property

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

//Instantiating driver object and launching browser

driver = **new** FirefoxDriver();

//Using get() method to open a webpage

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

//Closing the browser

driver.quit();

}

}

**Launching Chrome Browser**

1.Download the latest ChromeDriver binary from Chromium.org download page and place the executable on your local machine.

<https://chromedriver.chromium.org/downloads>

2.Set the webdriver.chrome.driver property to the chromeDriver.exe’s location as-

System.setProperty(“webdriver.chrome.driver”, “chromeDriver.exe path”);

**public** **class** **ChromeBrowserLaunchDemo** {

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

//Creating a driver object referencing WebDriver interface

WebDriver driver;

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

System.setProperty("webdriver.chrome.driver", "/lib/chromeDriver/chromedriver.exe");

//Instantiating driver object

driver = **new** ChromeDriver();

//Using get() method to open a webpage

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

//Closing the browser

driver.quit();

}

}

**Launching Internet Explorer Browser**

Like ChromeDriver, InternetExplorer driver also requires setting up the “webdriver.ie.driver” property with the location of IEDriverServer.exe. The IEDriverServer.exe can be downloaded from here.

<https://www.selenium.dev/downloads/>

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

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

//Creating a driver object referencing WebDriver interface

WebDriver driver;

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

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

//Instantiating driver object

driver = **new** InternetExplorerDriver();

//Using get() method to open a webpage

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

//Closing the browser

driver.quit();

}

}

**2.  How do you use findElement() and findElements()?**

**Answer:**  **findElement()**: findElement() is used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.

**Syntax:**

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

**findElements()**: findElements() is used to find all the elements in the current web page matching to the specified locator value. Take a note that all the matching elements would be fetched and stored in the list of WebElements.

**Syntax:**

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

**3. How can you find whether an element is displayed on the screen using Selenium?**

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

isDisplayed()

isSelected()

isEnabled()

**4. How can we get a text on a web element using Selenium?**

**Answer:**Get command is used to retrieve the inner text of the specified web element. The command doesn’t require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc displayed on the web pages.

Using getText() method.

String  eleText= driver.findElement(By.xpath(“ ”)).getText();

**5. How to type into a text box using Selenium?**

**Answer:**  User can use sendKeys(“String to be entered”) to enter the string in the textbox.

WebElement username= driver.findElement(By.id("uname"));

username.sendKeys("testusername");

**6. How to select a check box in Selenium?**

**Answer:**WebElement option1 = driver.findElement(By.id("checkbox123"));

// This will Toggle the Check box

option1.click();

**7. How to press enter key in selenium**

**Answer:**driver.findElement(By.id("Value")).sendKeys(Keys.ENTER);

**8. How to enter input in textbox without using sendkeys**

**Answer:**You can use Javascript Executer to input text into a text box without using sendKeys() method:

// Initialize JS object

JavascriptExecutor JS = (JavascriptExecutor)webdriver;

// Enter username

JS.executeScript("document.getElementById('User').value='admin'");

// Enter password

JS.executeScript("document.getElementById('Password').value='password123'");

**9. How to get no of frame present**

**Answer:**  
//First finding the elements using any of locator strategy  
 List<WebElement> iframList = driver.findElements(By.tagName("iframe"));

  int totalFrames = iframList.size();

  System.out.println("No of Frames:" + totalFrames);

**Complete Article:** <https://www.automationtestinginsider.com/2020/02/handling-iframes-in-selenium-webdriver.html>

**10. How to handle new tab?**

**Answer:**Please refer below article on how to handle multiple windows/tabs

<https://www.automationtestinginsider.com/2020/02/handle-multiple-windows-in-selenium.html>

**11. What are the three ways to initialize webdriver?**

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

**12. How to select all the check boxes in a page?**

**Answer:**Below is the sample script given:  
  
List<WebElement> list = driver.findElements(By.Xpath("//input[type='checkbox']"));

for(WebElement el : list){

    if(!el.isSelected()) // validate Checked property, otherwise you'll uncheck!

        el.click();

}

**13. How to verify whether the checkbox option or radio button is selected or not?**

**Answer:**Below is the sample script given:

String str = driver.findElement(By.id("26110162")).getAttribute("checked");

if (str.equalsIgnoreCase("true"))

{

    System.out.println("Checkbox selected");

}

**14. What is the alternative way to click on login button?**

**Answer:**use submit() method but it can be used only when attribute type=submit.

**15. How to handle a drop-down field and select a value from it using Selenium?**

**Answer:**Value in the drop down can be selected using WebDriver’s Select class.

Syntax:

**selectByValue**:

Select selectByValue = new Select(driver.findElement(By.id(“SelectID\_One”)));

selectByValue.selectByValue(“greenvalue”);

**selectByVisibleText**:

Select selectByVisibleText = new Select (driver.findElement(By.id(“SelectID\_Two”)));

selectByVisibleText.selectByVisibleText(“Lime”);

**selectByIndex**:

Select selectByIndex = new Select(driver.findElement(By.id(“SelectID\_Three”)));

selectByIndex.selectByIndex(2);

**16. How to check which option in the drop-down is selected?**

**Answer:**You should be able to get the text using getText() (for the option element you got using getFirstSelectedOption()

Select select = new Select(driver.findElement(By.xpath("//select")));

WebElement option = select.getFirstSelectedOption();

String defaultItem = option.getText();

System.out.println(defaultItem );

**17. How to select a third value from a drop-down field?**

**Answer:**  
 **Sample HTML**  
<html>

<body>

<select id = "designation">

<option value = "MD">MD</option>

<option value = "prog"> Programmer </option>

<option value = "CEO"> Chief Executive Officer </option>

</option>

</select>

<body>

</html>

Now to identify dropdown do

Select dropdown = new Select(driver.findElement(By.id("designation")));

To select its option say 'CEO' you can do

dropdown.selectByVisibleText("CEO");

or

dropdown.selectByIndex(2);

or

dropdown.selectByValue("CEO");

**18. How to find more than one web element in to a list?**

**Answer:**At times, we may come across elements of same type like multiple hyperlinks, images etc arranged in an ordered or unordered list. Thus, it makes absolute sense to deal with such elements by a single piece of code and this can be done using WebElement List.

Sample Code

// Storing the list

List <WebElement> elementList = driver.findElements(By.xpath("//div[@id='example']//ul//li"));

// Fetching the size of the list

int listSize = elementList.size();

for (int i=0; i<listSize; i++)

{

// Clicking on each service provider link

serviceProviderLinks.get(i).click();

// Navigating back to the previous page that stores link to service providers

driver.navigate().back();

}  
Please refer below link to get more details on multi select actions in selenium

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

**19. How to handle frames in WebDriver?**

**Answer**:  An inline frame acronym as iframe is used to insert another document with in the current HTML document or simply a web page into a web page by enabling nesting.

**Select iframe by id**

driver.switchTo().frame(“ID of the frame“);

**Locating iframe using tagName**

driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));

**Locating iframe using index**

frame(index)  
**Example:**

driver.switchTo().frame(0);

frame(Name of Frame)

driver.switchTo().frame(“name of the frame”);

frame(WebElement element)

**Select Parent Window**

driver.switchTo().defaultContent();  
Please refer below link to get more details on iframe in selenium

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

**20. How to click on a hyperlink using Selenium WebDriver?**

**Answer:**driver.findElement(By.linkText(“Google”)).click();

The command finds the element using link text and then click on that element and thus the user would be re-directed to the corresponding page.

The above mentioned link can also be accessed by using the following command.

driver.findElement(By.partialLinkText(“Goo”)).click();

The above command find the element based on the substring of the link provided in the parenthesis and thus partialLinkText() finds the web element with the specified substring and then clicks on it.

**21. How to read and verify the text on the tooltip using Selenium WebDriver?**

**Answer:**please refer below link

<https://www.automationtestinginsider.com/2020/01/how-to-handle-tooltip-in-selenium.html>

**22. How to execute JavaScript in Selenium?**

**Answer:**JavaScriptExecutor is an interface which provides mechanism to execute Javascript through selenium driver. It provides “executescript” & “executeAsyncScript” methods, to run JavaScript in the context of the currently selected frame or window

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

**23. How can we handle window based pop up using Selenium?**

**Answer:**Selenium is an automation testing tool which supports only web application testing. Therefore, windows pop up cannot be handled using Selenium.

But we can handle it using third party tool like AutoIT. Please refer below link.

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

**24. How can we handle web-based pop up using Selenium?**

**Answer:**WebDriver offers the users with a very efficient way to handle these pop ups using Alert interface. There are the four methods that we would be using along with the Alert interface.

• void dismiss() – The accept() method clicks on the “Cancel” button as soon as the pop up window appears.

• void accept() – The accept() method clicks on the “Ok” button as soon as the pop up window appears.

• String getText() – The getText() method returns the text displayed on the alert box.

• void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

**Syntax:**

// accepting javascript alert

                Alert alert = driver.switchTo().alert();

alert.accept();  
Please refer below link to get more details on popups in selenium

<https://www.automationtestinginsider.com/2020/01/alert-interface-and-handle-javascript.html>

**25. How to assert title of the web page?**

**Answer:**//verify the title of the web page

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

**26. How to mouse hover on a web element using WebDriver?**

**Answer:**WebDriver offers a wide range of interaction utilities that the user can exploit to automate mouse and keyboard events. Action Interface is one such utility which simulates the single user interactions.

Thus, In the following scenario, we have used Action Interface to mouse hover on a drop down which then opens a list of options.

**Sample Code:**

**// Instantiating Action Interface**

Actions actions=new Actions(driver);

**// howering on the dropdown**

actions.moveToElement(driver.findElement(By.id("id of the dropdown"))).perform();

**// Clicking on one of the items in the list options**

WebElement subLinkOption=driver.findElement(By.id("id of the sub link"));

subLinkOption.click();

**27. How to capture screen-shot in Selenium WebDriver ?**

**Answer:**Below is the sample script given:

**import** **org.junit.After**;

**import** **org.junit.Before**;

**import** **org.junit.Test**;

**import** **java.io.File**;

**import** **java.io.IOException**;

**import** **org.apache.commons.io.FileUtils**;

**import** **org.openqa.selenium.OutputType**;

**import** **org.openqa.selenium.TakesScreenshot**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.firefox.FirefoxDriver**;

**public** **class** **CaptureScreenshot** {

WebDriver driver;

**@Before**

**public** **void** **setUp**() **throws** Exception {

driver = **new** FirefoxDriver();

driver.get("https://google.com");

}

**@After**

**public** **void** **tearDown**() **throws** Exception {

driver.quit();

}

**@Test**

**public** **void** **test**() **throws** IOException {

// Code to capture the screenshot

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

// Code to copy the screenshot in the desired location

FileUtils.copyFile(scrFile, newFile("C:\\CaptureScreenshot\\google.jpg"));

}

}

**28. How to login into any site if it is showing an authentication pop-up for Username and Password?**

**Answer:**Please refer below link

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

**29. How to input text into the text box fields without calling the sendKeys()?**

**Answer:**In above example we have used javaScript to locate the Search Textbox and set the

“ATI” value in it.

WebDriver driver = new FirefoxDriver();

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

JavascriptExecutor myExecutor = ((JavascriptExecutor) driver);

myExecutor.executeScript("document.getElementsByName('q')[0].value='ATI'");

driver.quit();

**30. How to clear the text inside the text box fields using Selenium WebDriver?**

**Answer:**clear( ) predefined method of Selenium 'WebDriver' Class is used to clear the text entered or displayed in the text fields.

**Without Clear method:**

**public** **class** **Clear** {

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

Scanner in = **new** Scanner(System.in);

System.out.println("enter the string which you want to send and then clean");

String input = in.next();

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

driver.get("https://www.google.co.in/?gfe\_rd=cr&ei=zQmQU9D3OoyK8QfpxoDQBA&gws\_rd=ssl");

WebElement box = driver.findElement(By.name("q"));

box.sendKeys(input);

Thread.sleep(**5000**); //simply wait to see that input string has been sent to box

//this for block is used to clear the input from box

//this we can do by using clear method but clear sometime won't work so use this way

//box.clear();

box.sendKeys(Keys.chord(Keys.CONTROL,"a"));

box.sendKeys(Keys.BACK\_SPACE);

}

}

**31. How to get an attribute value of an element using Selenium WebDriver?**

**Answer:  Below is the sample script given:**

**Sample HTML**

<button name="btnK" id="gbqfba" aria-label="Google Search" class="gbqfba"><span id="gbqfsa">Google Search</span></button>  
**Selenium Script**

WebElement googleSearchBtn = driver.findElement(bySearchButton);

System.out.println("Name of the button is:- " +googleSearchBtn.getAttribute("name"));

**32. How to press Enter key on text box in Selenium WebDriver?**

**Answer:**For pressing Enter key over a textbox we can pass Keys.ENTER or Keys.RETURN to the sendKeys method for that textbox.

WebElement textbox = driver.findElement(By.id("idOfElement"));

textbox.sendKeys(Keys.ENTER);

Or

WebElement textbox = driver.findElement(By.id("idOfElement"));

textbox.sendKeys(Keys.RETURN);

**33. How to pause a text execution for 5 seconds at a specific point?**

**Answer:**Thread.sleep(5000);

**34. What is an alternative to driver.get() method to open a URL using Selenium WebDriver?**

**Answer:**Navigating to a URL is very common, then driver.get() is a convenient way. However, it does the same function as the driver.navigate().to(“url”)

**35. How to delete cookies in Selenium?**

**Answer:**Below is the sample script given:

**public** **void** **deleteCookieNamedExample**()

{

driver= **new** FirefoxDriver();

String URL="http://www.flipkart.com";

driver.navigate().to(URL);

//Delete particular cookie

driver.manage().deleteCookieNamed("\_\_utmb");

//Delete all cookies

driver.manage().deleteAllCookies();

}

**36. How to handle hidden elements in Selenium WebDriver?**

**Answer:**Please refer below link

<https://automation-seleniumtutorial.blogspot.com/2019/01/how-to-handle-hidden-elements-in-selenium-webdriver.html>

**37. How can you find broken links in a page using Selenium WebDriver?**

**Answer:**One of the key test case is to find broken links on a webpage. Due to existence of broken links, your website reputation gets damaged and there will be a negative impact on your business. It’s mandatory to find and fix all the broken links before release. If a link is not working, we face a message as 404 Page Not Found.

Let’s see some of the HTTP status codes.

200 – Valid Link

404 – Link not found

400 – Bad request

401 – Unauthorized

500 – Internal Error

Consider a test case to test all the links in the home page of “automationtestinginsider.com”

Below code fetches all the links of a given website (i.e., utomationtestinginsider.com) using WebDriver commands and reads the status of each href link with the help of HttpURLConnection class.

**import** **java.net.HttpURLConnection**;

**import** **java.net.URL**;

**import** **java.util.List**;

**import** **java.util.concurrent.TimeUnit**;

**import** **org.openqa.selenium.By**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.WebElement**;

**import** **org.openqa.selenium.firefox.FirefoxDriver**;

**public** **class** **BrokenLinks** {

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

//Instantiating FirefoxDriver

System.setProperty("webdriver.gecko.driver", "D:\\Selenium Environment\\Drivers\\geckodriver.exe");

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

//Maximize the browser

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

//Implicit wait for 10 seconds

driver.manage().timeouts().implicitlyWait(**10**, TimeUnit.SECONDS);

//To launch automationtestinginsider.com

driver.get("https://www.automationtestinginsider.com");

//Wait for 5 seconds

Thread.sleep(**5000**);

//Used tagName method to collect the list of items with tagName "a"

//findElements - to find all the elements with in the current page. It returns a list of all webelements or an empty list if nothing matches

List<WebElement> links = driver.findElements(By.tagName("a"));

//To print the total number of links

System.out.println("Total links are "+links.size());

//used for loop to

**for**(**int** i=**0**; i<links.size(); i++) {

WebElement element = links.get(i);

//By using "href" attribute, we could get the url of the requried link

String url=element.getAttribute("href");

//calling verifyLink() method here. Passing the parameter as url which we collected in the above link

//See the detailed functionality of the verifyLink(url) method below

verifyLink(url);

}

}

// The below function verifyLink(String urlLink) verifies any broken links and return the server status.

**public** **static** **void** **verifyLink**(String urlLink) {

//Sometimes we may face exception "java.net.MalformedURLException". Keep the code in try catch block to continue the broken link analysis

**try** {

//Use URL Class - Create object of the URL Class and pass the urlLink as parameter

URL link = **new** URL(urlLink);

// Create a connection using URL object (i.e., link)

HttpURLConnection httpConn =(HttpURLConnection)link.openConnection();

//Set the timeout for 2 seconds

httpConn.setConnectTimeout(**2000**);

//connect using connect method

httpConn.connect();

//use getResponseCode() to get the response code.

**if**(httpConn.getResponseCode()== **200**) {

System.out.println(urlLink+" - "+httpConn.getResponseMessage());

}

**if**(httpConn.getResponseCode()== **404**) {

System.out.println(urlLink+" - "+httpConn.getResponseMessage());

}

}

//getResponseCode method returns = IOException - if an error occurred connecting to the server.

**catch** (Exception e) {

//e.printStackTrace();

}

}

}

**38. How to find more than one web element in the list?**

**Answer:**findElements() command in WebDriver can be used to find more than one web elements and save them into a list.

1. If the provided value has the possibility to locate a single element, we can use findElement() command and save them into a variable of WebElement

2. If the provided locator has the possibility to locate multiple elements, we can use findElements() command and save them into a list of WebElement’s i.e. List<WebElement>

**39. How to scroll web page up and down using Selenium WebDriver?**

**Answer:**Please refer below article

<https://www.automationtestinginsider.com/2020/02/different-ways-of-scrolling-in-selenium.html>

**40. How to perform right click (Context Click) action in Selenium WebDriver?**

**Answer:**Please refer below link

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

**41. How to perform double click action in Selenium WebDriver?**

**Answer:**Please refer below link

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

**42. How to perform drag and drop action in Selenium WebDriver?**

**Answer:**Please refer below link

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

**43. How to highlight elements using Selenium WebDriver?**

**Answer:**Below is the sample script given

**public** **class** **HighlighterClass** {

**@Test**

**public** **void** **highlighterElement**() {

System.setProperty("webdriver.gecko.driver", "D:\\Selenium Environment\\Drivers\\geckodriver.exe");

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

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

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

WebElement ele = driver.findElement(By.xpath("//\*[@id='Email']"));

//Call the highlighterMethod and pass webdriver and WebElement which you want to highlight as arguments.

highLighterMethod(driver,ele);

ele.sendKeys("ATI");

}

//Creating a custom function

**public** **void** **highLighterMethod**(WebDriver driver, WebElement element){

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].setAttribute('style', 'background: yellow; border: 2px solid red;');", element);

}

}

**44. How to read a JavaScript variable in Selenium WebDriver?**

**Answer:**Its quite easy to access any JavaScript variable from the Selenium Webdriver test scripts. Just can simply use the below code snippet.

JavascriptExecutor JS = (JavascriptExecutor) webdriver;

// Get the current site title.

String sitetitle = (String)JS.executeScript("return document.title");

System.out.println("My Site Title: " + sitetitle);

**45. How to handle Ajax calls in Selenium WebDriver?**

**Answer:**Handling AJAX calls is one of the common issues when using Selenium WebDriver. We wouldn’t know when the AJAX call would get completed and the page has been updated. In this post, we see how to handle AJAX calls using Selenium.

AJAX stands for Asynchronous JavaScript and XML. AJAX allows the web page to retrieve small amounts of data from the server without reloading the entire page. AJAX sends HTTP requests from the client to server and then process the server’s response without reloading the entire page. To handle AJAX controls, wait commands may not work. It’s just because the actual page is not going to refresh.

When you click on a submit button, required information may appear on the web page without refreshing the browser. Sometimes it may load in a second and sometimes it may take longer. We have no control on loading time. The best approach to handle this kind of situations in selenium is to use dynamic waits (i.e. WebDriverWait in combination with ExpectedCondition)

titleIs() – The expected condition waits for a page with a specific title.

wait.until(ExpectedConditions.titleIs(“Deal of the Day”));

wait.until(ExpectedConditions.titleIs(“Deal of the Day”));

elementToBeClickable() – The expected condition waits for an element to be clickable i.e. it should be present/displayed/visible on the screen as well as enabled.

wait.until(ExpectedConditions.elementToBeClickable(By.xpath("xpath")));

wait.until(ExpectedConditions.elementToBeClickable(By.xpath("xpath")));

alertIsPresent() – The expected condition waits for an alert box to appear.

wait.until(ExpectedConditions.alertIsPresent()) !=null);

wait.until(ExpectedConditions.alertIsPresent()) !=null);

textToBePresentInElement() – The expected condition waits for an element having a certain string pattern.

wait.until(ExpectedConditions.textToBePresentInElement(By.id(“title’”), “text to be found”));

wait.until(ExpectedConditions.textToBePresentInElement(By.id(“title’”), “text to be found”));  
  
Please refer very good article on waits:  
<https://www.automationtestinginsider.com/2020/02/waits-in-selenium-webdriver.html>

**46. How to upload a file in Selenium WebDriver?**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2020/02/upload-and-download-file-in-selenium.html>

**47. How to download a file in Selenium WebDriver?**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2020/02/upload-and-download-file-in-selenium.html>

**48. How to run Selenium WebDriver tests from command line?**

**Answer:**Answer will be provided soon

**49. How to connect to a database in Selenium?**

**Answer:**As we know every application has to maintain a database like My SQL, Oracle or any other databases to store all its data. And where as Selenium Webdriver is used for testing web applications and we perform many operations like submitting information and some times retrieving information and validate them.

In selenium scripts, when there is a need of getting the Data from the database we may have to use APIs which helps to interact with database like JDBC.

Java Database Connectivity(JDBC) is a Java API which is used to connect and interact with Database.

**Why do we use JDBC instead ODBC?**

ODBC is developed and written in C, which is platform Independent. where as JDBC API is a specification provides set of interfaces which are written and developed in Java programming language.

**How to Connect to database using JDBC?**

The below are the Steps to Connect to database, before proceeding, you need to have MySQL Connector. You can download from here Download MySQL Connector Jar and add it the build path as we add selenium webdriver jar.

**1. Load and Registering the Driver**

For registering the Driver we Load the Driver class using forName() method.

forName() is the static factory method which is present in predefined class called "Class". This method loads the class which is mentioned as parameter.

Class.forName("com.mysql.jdbc.Driver");// class.forName load the Driver class

Internally this Driver class will register the driver by using static method called registerDriver().

**2. Establishing Connection.**

For establishing connection with database we call static method called getConnection(...) present in DriverManager Class. This method contains three arguments of string type. i.e., url, username and password

DriverManager.getConnection("jdbc:mysql://localhost:3306/Employee","root","root");

URL contains "jdbc(main protocol):mysql(sub protocol for mySql)://localhost:3306(sub name for mysql (host:prot))/Employee(database)" and this method return type is Connection Object ie.,

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Employee","root","root");

**3. Creating Statement Object**

For creating statement object we need to call a method called createStatement() which is present in Connection Interface.

con.createStatement();

And this method returns Statement object and it is no argument method.

Statement st= con.createStatement();

**4. Execute the Statement**

For executing queries there are different methods present in Statement Interface for retrieving records and for updating records.

Retrieving records:

for executing select queries(for fetching records) we call a method called executeQuery(String qry) by taking string as parameter.

st.executeQuery("Select \* from Employee");

This method returns ResultSet object.

Resultset rs= st.executeQuery("Select \* from Employee");// once executeQuery() executes the query and stores the records in to ResultSet object.

**5. Closing the connection**.

Once execution of all statements were completed we need to close all the connections by using method called close() present in Connection interface

con.close();

**public** **class** **TestDatabaseTesting** {

**@Test**

**public** **void** **TestVerifyDB**(){

**try** {

// This will load the driver

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

System.out.println("Driver Loaded");

// Specify the path of the database

String dblocation= "C:\\Users\\Desktop\\DB\\FacebookDB1.accdb";

// This will connect with Database , getConnection taking three argument

// first argument Test\_Oracle is the dsn which you can change as per your system,

Connection con=DriverManager.getConnection("jdbc:odbc:AscentAccess;DBQ="+dblocation);

System.out.println("Connection created");

// This will create statement

Statement smt=con.createStatement();

System.out.println("Statement created");

// Now execute the query, here facebook is the table which I have created in DB

ResultSet rs= smt.executeQuery("Select \* from Facebook");

System.out.println("Query Executed");

// Iterate the resultset now

**while**(rs.next()){

String uname= rs.getString("username");

String pass= rs.getString("password");

String email= rs.getString("email");

System.out.println("Uname is "+uname+" password is "+pass+" email id is "+email);

}

}

**catch** (Exception e) {

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

}

}

}

**50. How to resize browser window using Selenium WebDriver?**

**Answer:**Below is the sample script given

**public** **class** **ResizeBrowser** {

**@Test**

**public** **void** **launchBrowser**() {

System.setProperty("webdriver.gecko.driver","D://Selenium Environment//Drivers//geckodriver.exe");

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

driver.navigate().to("http://www.automationtestinginsider.com");

System.out.println(driver.manage().window().getSize());

//Create object of Dimensions class

Dimension d = **new** Dimension(**480**,**620**);

//Resize the current window to the given dimension

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

System.out.println(driver.manage().window().getSize());

}

}

**51. How to handle Chrome Browser notifications in Selenium?**

**Answer:**The pop-ups can be irritating but it hardly will effect your script. The pop-ups are generally notifications, ads etc.

Even though if you want to remove or disable to pop-up notifications then you can use the following code:-

ChromeOptions options = new ChromeOptions();

options.addArguments("disable-infobars");

WebDriver player = new ChromeDriver(options)

**52. How to assert text on a web page using Selenium?**

**Answer:**Below is the sample script given  
WebElement el = driver.findElement(By.id(“ElementID”))

//get text from element and stored in String variable

String text = el.getText();

//assert text from expected

Assert.assertEquals(“Element Text”, text);

**53. How to find broken images in a page using Selenium WebDriver?**

**Answer:**Below is the sample script given

**public** **class** **FindBrokenImages** {

**private** WebDriver driver;

**private** **int** invalidImageCount;

**@BeforeClass**

**public** **void** **setUp**() {

driver = **new** FirefoxDriver();

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

}

**@Test**

**public** **void** **validateInvalidImages**() {

**try** {

invalidImageCount = **0**;

List<WebElement> imagesList = driver.findElements(By.tagName("img"));

System.out.println("Total no. of images are " + imagesList.size());

**for** (WebElement imgElement : imagesList) {

**if** (imgElement != **null**) {

verifyimageActive(imgElement);

}

}

System.out.println("Total no. of invalid images are " + invalidImageCount);

} **catch** (Exception e) {

e.printStackTrace();

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

}

}

**@AfterClass**

**public** **void** **tearDown**() {

**if** (driver != **null**)

driver.quit();

}

**public** **void** **verifyimageActive**(WebElement imgElement) {

**try** {

HttpClient client = HttpClientBuilder.create().build();

HttpGet request = **new** HttpGet(imgElement.getAttribute("src"));

HttpResponse response = client.execute(request);

// verifying response code he HttpStatus should be 200 if not,

// increment as invalid images count

**if** (response.getStatusLine().getStatusCode() != **200**)

invalidImageCount++;

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

**54. How to handle colors in Selenium WebDriver?**

**Answer:**First of all, we have to get a value of link color using getCssValue method. It can be done by using below code. In the code, Products link’s CSS attribute ‘color’ is stored in a String variable called ‘color’.

We need to return value in RGB format such as “rgba(36, 93, 193, 1)”. We will convert it into more convenient Hex code using Java.

We can add an Assert statement to verify that the color is matching with the expected color.

**Please refer below program**

**public** **void** **GoogleAbout**() **throws** Exception{

String color = driver.findElement(By.xpath("//a[@href='products/']")).getCssValue("color");

String[] hexValue = color.replace("rgba(", "").replace(")", "").split(",");

**int** hexValue1=Integer.parseInt(hexValue[**0**]);

hexValue[**1**] = hexValue[**1**].trim();

**int** hexValue2=Integer.parseInt(hexValue[**1**]);

hexValue[**2**] = hexValue[**2**].trim();

**int** hexValue3=Integer.parseInt(hexValue[**2**]);

String actualColor = String.format("#%02x%02x%02x", hexValue1, hexValue2, hexValue3);

Assert.assertEquals("#245dc1", actualColor);

}

**55. Explain how you can switch back from a frame?**

**Answer:**We can even switch to the iframe using web element . We have to come out of the iframe. To move back to the parent frame, you can either use switchTo(). parentFrame() or if you want to get back to the main (or most parent) frame, you can use switchTo()

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

**56. How can you redirect browsing from a browser through some proxy?**

**Answer:** The way of setting it varies depending on Selenium Client Language Bindings and the browser you’re automating, here is an example for Java and Chrome (assuming ChromeDriver and org.openqa.selenium.Proxy)

Please refer below program:

**import** **org.openqa.selenium.Proxy**;

**import** **org.openqa.selenium.chrome.ChromeDriver**;

**import** **org.openqa.selenium.chrome.ChromeOptions**;

**public** **class** **ChromeWithProxy** {

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

Proxy proxy = **new** Proxy();

proxy.setHttpProxy("proxyHost:proxyPort");

proxy.setSslProxy("proxyHost:proxyPort");

ChromeOptions options = **new** ChromeOptions();

options.setProxy(proxy);

ChromeDriver driver = **new** ChromeDriver(options);

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

driver.quit();

}

}

**57. Write a code to wait for a particular element to be visible on a page.**

**Answer:**Below is the sample script given

**@Test**

**public** **void** **test**() **throws** InterruptedException, IOException

{

//To wait for element visible

WebDriverWait wait = **new** WebDriverWait(driver, **15**);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//input[@id='text3']")));

driver.findElement(By.xpath("//input[@id='text3']")).sendKeys("Text box is visible now");

System.out.print("Text box text3 is now visible");

}

**58. Write a code to wait for an alert to appear.**

**Answer:**Below is the sample script given

**public** **class** **Mytest1** {

WebDriver driver = **null**;

**@Before**

**public** **void** **beforetest**() {

System.setProperty("webdriver.gecko.driver", "D:\\Selenium Files\\geckodriver.exe");

driver = **new** FirefoxDriver();

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

driver.get("http://only-testing-blog.blogspot.in/2014/01/new-testing.html");

}

**@After**

**public** **void** **aftertest**() {

driver.quit();

}

**@Test**

**public** **void** **test** ()

{

driver.findElement(By.xpath("//input[@name='fname']")).sendKeys("My Name");

WebDriverWait wait = **new** WebDriverWait(driver, **15**);

wait.until(ExpectedConditions.alertIsPresent());

String alrt = driver.switchTo().alert().getText();

System.out.print(alrt);

}

}

**59. How to handle keyboard and mouse actions using Selenium?**

**Answer:**Please refer below link

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

**60. How can you fetch an attribute from an element?**

**Answer:**The getAttribute() method returns the value of the attribute with the specified name, of an element. Tip: Use the getAttributeNode() method if you want to return the attribute as an Attr object.

**61. How to retrieve typed text from a text box?**

**Answer:**To get the typed text from a textbox, you can use getAttribute(“value”) method by passing arg as value. Value attribute stores the typed text of the textbox element. See the following example to know better:

String typedText = driver.findElement(By.xpath("xpath\_textbox")).getAttribute("value"));

**62. How to send alt or shift or control or enter or tab key in Selenium WebDriver?**

**Answer:**When we generally use ALT/SHIFT/CONTROL keys, we hold onto those keys and click other buttons to achieve the special functionality. So it is not enough just to specify keys.ALT or keys.SHIFT or keys.CONTROLfunctions.

For the purpose of holding onto these keys while subsequent keys are pressed, we need to define two more methods: keyDown(modifier\_key) and keyUp(modifier\_key)

Parameters: Modifier\_key (keys.ALT or Keys.SHIFT or Keys.CONTROL)

Purpose: Performs a modifier key press and does not release the modifier key. Subsequent interactions may assume it’s kept pressed.

Parameters: Modifier\_key (keys.ALT or Keys.SHIFT or Keys.CONTROL)

Purpose: Performs a key release.

Hence with a combination of these two methods, we can capture the special function of a particular key.  
**Below is the sample script given**

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

{

String baseUrl = “**https:**//www.facebook.com”;

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

driver.get("baseUrl");

WebElement txtUserName = driver.findElement(By.id(“Email”);

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

Action seriesOfActions = builder

.moveToElement(txtUerName)

.click()

.keyDown(txtUserName, Keys.SHIFT)

.sendKeys(txtUserName, “hello”)

.keyUp(txtUserName, Keys.SHIFT)

.doubleClick(txtUserName);

.contextClick();

.build();

seriesOfActions.perform();

}​

**63. How to switch to a new window (new tab) which opens up after you click on a link?**

**Answer:**If you click on a link in a web page, then for changing the WebDriver’s focus/ reference to the new window we need to use the switchTo() command. Look at the below example to switch to a new window:

String handle = driver.getWindowHandle();

//get the name of all the windows that were initiated by the WebDriver

for (String handle : driver.getWindowHandles())

{

    driver.switchTo().window(handle);

}

**65. How to check for 404 using Selenium WebDriver?**

**Answer:**Below is the sample script given

**public** **class** **Verif404** {

**public** **static** **void** **verifyLinkActive**(String linkUrl)

{

**try**

{

URL url = **new** URL(linkUrl);

HttpURLConnection httpURLConnect=(HttpURLConnection)url.openConnection();

httpURLConnect.setConnectTimeout(**3000**);

httpURLConnect.connect();

**if**(httpURLConnect.getResponseCode()==**400**)

{

System.out.println(linkUrl+" - "+httpURLConnect.getResponseMessage());

}

**else**

{

System.out.println(linkUrl+" - "+httpURLConnect.getResponseMessage()" );

}

} **catch** (Exception e) {

}

}

}

**66. How to get HTTP Response code using Selenium WebDriver?**

**Answer:**This is a Java library use for general HTTP network communication and includes supports sessions via cookies. Their fluent API makes it relatively simple to get the response code for a URL.

    public int getResponseCode(String url) {

        try {

            return Request.Get(url).execute().returnResponse().getStatusLine()

                    .getStatusCode();

        } catch (Exception e) {

            throw new RuntimeException(e);

        }

    }

**67. How to stop selenium from creating temporary Firefox profiles using web driver?**

**Answer:**

1. You can control how the Firefox driver chooses the profile. Set the webdriver.firefox.profile property to the name of the profile you want to use. Most folks think this is a bad idea, because you'll inherit all the cookies, cache contents, etc. of the previous uses of the profile, but it's allowed if you really want to do it.

System.setProperty("webdriver.firefox.profile", "MySeleniumProfile");

WebDriver driver = new FirefoxDriver(...);

2.

FirefoxProfile profile = new FirefoxProfile(new File("D:\\Selenium Profile"));

WebDriver driver = new FirefoxDriver(profile);

Then I changed settings in Firefox to clear all cookies and cache when exiting.

**68. How to close browser popup window in Selenium WebDriver?**

**Answer:**There're at least 3 ways to handle this case.

1. Refresh the page and then dismiss the dialog if the driver supports it :

driver.refresh();

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

driver.quit();

2. Setup the browser oriented driver to avoid the prompt :

// Chrome

ChromeOptions options = new ChromeOptions();

options.addArguments("--disable-popup-blocking");

driver = new ChromeDriver(options);

// Firefox

FirefoxOptions options = new FirefoxOptions();

options.addPreference("dom.disable\_beforeunload", true)

WebDriver driver = new FirefoxDriver(options)

3. Add some JavaScript code into the page to escape the prompt:

string JS\_DISABLE\_UNLOAD\_DIALOG = "Object.defineProperty(BeforeUnloadEvent.prototype, 'returnValue', { get:function(){}, set:function(){} });"

((JavascriptExecutor)driver).executeScript(JS\_DISABLE\_UNLOAD\_DIALOG);

driver.quit();

**69. How to select values in combo-box item using Selenium WebDriver?**

**Answer:**

**70. How to get the html source code of a particular web element using Selenium WebDriver?**

**Answer:**You can read innerHTML attribute to get source of the content of the element or outerHTML for source with the current element.

elem.getAttribute("innerHTML");

**71. How can you find the value of different attributes like name, class, value of an element?**

**72. How to verify whether a button is enabled on the page?**

**Answer:**Below is the sample script given

**public** **class** **ButtonTest**{

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

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver", "C:\\Selenuim\\chromedriver2.3.exe");

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

**try**{

driver.get("http://register.rediff.com/register/register.php");

Thread.sleep(**2000**);

WebElement e = driver.findElement(By.name("btnemail"));

**boolean** actualValue = e.isEnabled();

**if** (actualValue)

System.out.println("Button is enabled");

**else**

System.out.println("Button is disabled");

Thread.sleep(**2000**);

}

**catch**(Exception ex){

System.out.println("Exception " + ex.getMessage());

}

**finally**{

driver.close();

driver.quit();

}

}

}

**73. What kind of mouse actions can be performed using Selenium?**

**Answer:**Mouse Events Using Selenium Actions Class API:

doubleClick (): Double clicks onElement. contextClick() : Performs a context-click (right click) on an element. clickAndHold(): Clicks at the present mouse location (without releasing)

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

**74. What kind of keyboard operations can be performed in Selenium?**

**Answer:**

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

**75. How to locate a link using its text in Selenium?**

**Answer:**Below is the sample script given  
driver.get("<a href="https://twitter.com/">https://twitter.com/</a>");

driver.findElement(By.linkText("Sign Up")).click(); //linkText locator for links

driver.get("<a href="https://twitter.com/">https://twitter.com/</a>");

driver.findElement(By.partialLinkText("Sign")).click(); //partiallinkText locator for links

**76. Write the program to locate/fetch all the links on a specific web page?**

**Answer:** Below is the sample script given

**public** **class** **GetAllLinks** {

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

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

//Launching sample website

driver.get("https://www.automationtestinginsider.com");

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

//Get list of web-elements with tagName - a

List<WebElement> allLinks = driver.findElements(By.tagName("a"));

//Traversing through the list and printing its text along with link address

**for**(WebElement **link:**allLinks){

System.out.println(link.getText() + " - " + link.getAttribute("href"));

}

//Commenting driver.quit() for user to verify the links printed

//driver.quit();

}

}

**77. How do you get the height and width of a text box field using Selenium?**

**Answer:**You can use following lines of code to get the size of a textbox on a webpage using Selenium Webdriver:

driver.findElement(By.xpath(“xpath\_textbox”)).getSize().getWidth();

driver.findElement(By.xpath(“xpath\_textbox”)).getSize().getHeight();

**78. How to handle alerts in Selenium WebDriver?**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2020/01/alert-interface-and-handle-javascript.html>

**79. How can we submit a form in Selenium?**

**Answer:**Below is the sample script given

**//Enter username**

driver.findElement(By.id("usrnm")).sendKeys("contact.automateapps@gmail.com");

**//Enter password**

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

**//submit form**

driver.findElement(By.id("password")).submit();

**80. How can we fetch the title of the page in Selenium?**

**Answer:**Below is the sample script given

//Get and verify the title of the web page

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

**81. How can we fetch the page source in Selenium?**

**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.

WebDriver driver = new ChromeDriver();

driver.get("https://www.googel.com/");

String str = driver.getPageSource();

System.out.println(str);

**82. How to handle HTTPS websites in Selenium? Does Selenium support them?**

**Answer:**

**83. How to accept the SSL untrusted connection?**

**Answer:**Below is the sample script given

**public** **class** **SSLCertificate** {

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

//It create firefox profile

FirefoxProfile profile=**new** FirefoxProfile();

// This will set the true value

profile.setAcceptUntrustedCertificates(**true**);

// This will open firefox browser using above created profile

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

driver.get("pass the url as per your requirement");

}

}

**84. How to handle AJAX popup windows?**

**Answer:**Please refer below link

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

**85. How to get the number of frames on a page using Selenium WebDriver?**

**Answer:**Below is the sample script given

List<WebElement> iframList = driver.findElements(By.tagName("iframe"));

  int totalFrames = iframList.size();

  System.out.println("No of Frames:" + totalFrames);

**86. What is the command line we have to write inside a .bat file to execute a Selenium Project when we are using testng?**

**Answer:**

Step 1: Open notepad

Step 2: Paste the below lines of code - You may need to add your project location. In the example, project location is set as 'F:\Selenium\TestNGBatchExample'.

Step 3: Save the file as 'testNGBatchFile.bat' in location that you want to save.

set projectLocation=F:\Selenium\TestNGBatchExample

cd %projectLocation%

set classpath=%projectLocation%\bin;%projectLocation%\lib\\*

java org.testng.TestNG %projectLocation%\testng.xml

pause

**87. How to check if a text is highlighted on the page?**

**Answer:**You can use following code snippet to check whether a text is highlighted on a page using Selenium:

String color = driver.findElement(By.xpath("//a[text()='Shop']")).getCssValue("color");

String backcolor = driver.findElement(By.xpath("//a[text()='Shop']")).getCssValue("background-color");

System.out.println(color);

System.out.println(backcolor);

**if**(!color.equals(backcolor)){

System.out.println("Text is highlighted!")

}

**else**{

System.out.println("Text is not highlighted!")

}

**88. How to check whether a text is underlined or not?**

**Answer:**You can use following code snippet to check whether a text is underlined or not. But first you need to identify the underlined property. You can Identify by getCssValue(“border-bottom”) or sometime getCssValue(“text-decoration”) method if the cssValue is 'underline' for that WebElement or not.  
Below is the sample script given:

**public** **class** **UnderLine** {

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

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

driver.manage().timeouts().implicitlyWait(**10**, TimeUnit.SECONDS);

driver.get("https://www.google.co.in/?gfe\_rd=ctrl&amp;ei=bXAwU8jYN4W6iAf8zIDgDA&amp;gws\_rd=cr");

String cssValue= driver.findElement(By.xpath("//a[text()='Hindi']")).getCssValue("text-decoration");

System.out.println("value"+cssValue);

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

act.moveToElement(driver.findElement(By.xpath("//a[text()='Hindi']"))).perform();

String cssValue1= driver.findElement(By.xpath("//a[text()='Hindi']")).getCssValue("text-decoration");

System.out.println("value over"+cssValue1);

driver.close();

}

}

**89. How to change the URL on a web page using Selenium WebDriver?**

**Answer:**You can use some wait method and then invoke the driver again.

 //open browser

    driver = new FirefoxDriver();

    //login

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

    //set implicit wait

    driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

    //Then invoke method again for your second request(I am not try this code maybe you need to create new driver object)

    driver.get("https://www.automationtestinginsider.com");

**90. How to press Shift+Tab?**

**Answer:**Using chord method

String press = Keys.chord(Keys.SHIFT,Keys.ENTER);

webelement.sendKeys(press);

String press = Keys.chord(Keys.SHIFT,Keys.ENTER);

webelement.sendKeys(press);

**91. Count the number of links in a page?**

**Answer:**Below is the sample script given  
List<WebElement> links = driver.findElements(By.xpath("//a"));    //Identify the number of Link on webpage and assign into Webelement List

int linkCount = links.size();     // Count the total Link list on Web Page

System.out.println("Total Number of link count on webpage = "  + linkCount);    //Print the total count of links on webpage

**92. Write a code to make use of assert if my username is incorrect.**

**Answer:**Below is the sample script given  
try{

Assert.assertEquals(expUserName, actUserName);

}catch(Exception e){

Syste.out.println(“name is invalid”);

}

**93. What are the different methods which can be used to verify the existence of an element on a web page?**

**Answer:**isEnabled() is the method used to verify if the web element is enabled or disabled within the webpage. isEnabled() is primarily used with buttons. isSelected() is the method used to verify if the web element is selected or not. isSelected() method is predominantly used with radio buttons, dropdowns and checkboxes.

**94. How do you retrieve the text displayed on an Alert?**

**Answer:**There is a method in Alert interface which gives you the text of the alert box message. As below:

Alert alert = driver.switchTo().alert();

alert.getText();

**95. How do you type text into the text box on an Alert?**

**Answer:**You can use sendKeys() method send some data to a Prompt Alert Box in Selenium Webdriver. sendKeys() method takes String as an input and is mostly used to send keyboard inputs in Selenium Webdriver. Checkout the following command to send data to alert box:

driver.switchTo().alert().sendKeys("Dummy\_text");

**96. How do you capture screen-shots in Selenium and what is the best place to have the screen-shot code?**

**Answer:**Below is the sample script given

**public** **class** **ScreenshootGoogle** {

**@Test**

**public** **void** **TestJavaS1**()

{

// Open Firefox

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

// Maximize the window

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

// Pass the url

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

// Take screenshot and store as a file format

File src= ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

**try** {

// now copy the screenshot to desired location using copyFile //method

FileUtils.copyFile(src, **new** File("C:/selenium/error.png"));

}

**catch** (IOException e)

{

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

}

**97. Write a code using JavascriptExecutor to scroll the web page?**

**Answer:**Below is the sample script given

**public** **class** **ScrollUpDown** {

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

WebDriver driver;

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

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

driver = **new** ChromeDriver();

driver.get("https://www.automationtestinginsider.com/");

Thread.sleep(**2000**);

JavascriptExecutor js= (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(1000,0)");

}

}

**98. How to switch to another window using Selenium?**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2020/02/handle-multiple-windows-in-selenium.html>

**99. How to set the Firefox Profile in Selenium WebDriver?**

**Answer:**Below is the sample script given

// import the package

**import** **java.io.File**;

**import** **java.util.concurrent.TimeUnit**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.firefox.FirefoxDriver**;

**import** **org.openqa.selenium.firefox.FirefoxProfile**;

**import** **org.openqa.selenium.firefox.internal.ProfilesIni**;

**public** **class** **FirefoxProfile** {

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

ProfilesIni profile = **new** ProfilesIni();

FirefoxProfile myprofile = profile.getProfile("abcProfile");

// Initialize Firefox driver

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

//Maximize browser window

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

//Go to URL which you want to navigate

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

//Set timeout for 5 seconds so that the page may load properly within that time

driver.manage().timeouts().implicitlyWait(**5**, TimeUnit.SECONDS);

//close firefox browser

driver.close();

}

}

Below is the explanation of code line by line.

Code line 2-7: First of all we need to import the package required to run the selenium code.

Code line 8: Make a public class "FirefoxProfile."

Code line 9: Make an object (you need to have basic knowledge of oops concepts).

Code line 10-11: We need to initialize Firefox profile with the object of myprofile .

Code line 13: Create object for Firefox

Code line 15: Maximize window.

Code line 17:Driver.get use to navigate to given URL .

Code line 19: Set timeout is used to wait for some time so that browser may load the page before proceeding to next page.

Code line 21:Close Firefox.

**100. How to read data from the XML files?**

**Answer:**

Sample XML

<?xml version = "1.0"?>

<bank>

<account acn = "001">

<firstname>Alvin</firstname>

<lastname>Joseph</lastname>

<balance>**50000**</balance>

</account>

<account acn = "002">

<firstname>Mark</firstname>

<lastname>dsouza</lastname>

<balance>**500**</balance>

</account>

<account acn = "003">

<firstname>David</firstname>

<lastname>dsouza</lastname>

<balance>**1000**</balance>

</account>

</bank>

Let’s understand the code which is used for parsing the XML file.

**import** **java.io.File**;

**import** **java.util.Iterator**;

**import** **java.util.ListIterator**;

**import** **javax.xml.parsers.DocumentBuilder**;

**import** **javax.xml.parsers.DocumentBuilderFactory**;

**import** **org.w3c.dom.Document**;

**import** **org.w3c.dom.Element**;

**import** **org.w3c.dom.Node**;

**import** **org.w3c.dom.NodeList**;

**public** **class** **XMLFileReader** {

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

**try**{

String filePath = "XMLFilePath";

File file = **new** File(filePath);

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbf.newDocumentBuilder();

Document doc = dBuilder.parse(file);

doc.getDocumentElement().normalize();

System.out.println(doc.getDocumentElement().getNodeName());

NodeList nodeList = doc.getElementsByTagName("account");

**int** tLength = nodeList.getLength();

**for**(**int** i=**0**; i<tLength; i++){

Node node = nodeList.item(i);

**if**(node.getNodeType()==Node.ELEMENT\_NODE){

Element element = (Element)node;

System.out.println("Account No: "+element.getAttribute("acn"));

System.out.println("First Name: "+element.getElementsByTagName("firstname").item(**0**).getTextContent());

System.out.println("Last Name: "+element.getElementsByTagName("lastname").item(**0**).getTextContent());

System.out.println("Balance: "+element.getElementsByTagName("balance").item(**0**).getTextContent());

}

}

}**catch** (Exception e){

e.printStackTrace();

}

}

}

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbf.newDocumentBuilder();

These two lines of code create the instance of the DOM architecture of the XML file.

Document doc = dBuilder.parse(file);

doc.getDocumentElement().normalize();

These codes parse the XML file and normalize it for test data retrieval.

We use JavaScript methods to get the texts So here is the entire program to read the test data from an XML file.

**Output:**

bank

Account **No:** **001**

First **Name:** Alvin

Last **Name:** Joseph

**Balance:** **50000**

Account **No:** **002**

First **Name:** Mark

Last **Name:** dsouza

**Balance:** **500**

Account **No:** **003**

First **Name:** David

Last **Name:** dsouza

**Balance:** **1000**

**101. If we write FileInputStream statements in try block, what are the statements we will write in finally block?**

**Answer:**Below is the sample script given

close() statement is used to close all the open streams in a program. Its a good practice to use close() inside **finally** block. Since **finally** block executes even **if** exception occurs so you can be sure that all input and output streams are closed properly regardless of whether the exception occurs or not.

**public** **class** **SimpleExceptionHandling** {

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

FileInputStream fileInputStream = **null**;

**try** {

fileInputStream = **new** FileInputStream("C:\\SomeTempFile");

} **catch** (FileNotFoundException e) {

System.out.println("An exception occured :: " + e.getMessage());

System.out.println("Need to close FileStream");

} **finally** {

System.out.println("In Finally Block - Thank God we are here :) ");

**try** {

**if** (fileInputStream != **null**) {

fileInputStream.close();

}

} **catch** (IOException e) {

}

}

}

}

**Output:**

An exception occured :: **C:**\SomeTempFile (The system cannot find the file specified)

Need to close FileStream

In Finally Block - Thank God we are here :)

**102. How to capture the bitmaps in Selenium?**

**Answer:**

**103. How to select a date in a Calendar on a web page using Selenium?**

**Answer:**To handle this type of control we will fill date without separating with delimiter, i.e. if date is 03/24/2020, then we will pass 03242020 to the input box

**import** **java.util.concurrent.TimeUnit**;

**import** **org.openqa.selenium.By**;

**import** **org.openqa.selenium.Keys**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.WebElement**;

**import** **org.openqa.selenium.chrome.ChromeDriver**;

**import** **org.testng.annotations.Test**;

**public** **class** **DateTimePicker** {

WebDriver driver;

**@Test**

**public** **void** **dateTimePicker**(){

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

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

driver = **new** ChromeDriver();

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

driver.manage().timeouts().implicitlyWait(**10**, TimeUnit.SECONDS);

driver.get("https://www.automationtestinginsider.com/2019/08/textarea-textarea-element-defines-multi.html");

//Find the date time picker control

WebElement dateBox = driver.findElement(By.id("start"));

//Fill date as mm/dd/yyyy as 03/24/2020

dateBox.sendKeys("03242020");

//Press tab to shift focus to time field

dateBox.sendKeys(Keys.TAB);

//Fill time as 02:45 PM

//dateBox.sendKeys("0245PM");

driver.close();

}

}

**104. How to get columns from a table?**

**Answer:**

**105. How to send emails from official mail id?**

**Answer:**

**106. How to achieve Singleton Design in Selenium WebDriver?**

**Answer:**

What is Singleton Design Pattern?

When we develop a class in such a way that it can have only instance at any time, is called Singleton design pattern. It is very useful when you need to use same object of a class across all classes or framework. Singleton class must return the same instance again, if it is instantiated again.

To create a singleton class, we need to do following steps:

Declare constructor of class as private so that no one instantiate class outside of it.

Declare a static reference variable of class. Static is needed to make it available globally.

Declare a static method with return type as object of class which should check if class is already instantiated once.

**class** **SingletonClassExample**

{

// declaring an instance of class SingletonClassExample which is null initially means not initialized.

**private** **static** SingletonClassExample instanceOfSingletonClassExample = **null**;

// Declaring constructor as private to restrict object creation outside of class

**private** **SingletonClassExample**()

{

System.out.println("Object created");

}

// static method to create instance of class SingletonClassExample

**public** **static** SingletonClassExample **getInstanceOfSingletonClassExample**()

{

**if** (instanceOfSingletonClassExample == **null**)

instanceOfSingletonClassExample = **new** SingletonClassExample();

**return** instanceOfSingletonClassExample;

}

}

// A normal class where we will try to instantiate class SingletonClassExample

**class** **LetsTryToInstaltiateSingletonClassExample**

{

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

{

// instantiating Singleton class first time

SingletonClassExample first= SingletonClassExample.getInstanceOfSingletonClassExample();

// instantiating Singleton class second time

SingletonClassExample second= SingletonClassExample.getInstanceOfSingletonClassExample();

}

}

**Output:**

Object created

When you run above program, you will get print “Object created.” only once while you have instantiated class twice. This is because of singleton pattern. It will not create object of class again if already initialized once.

How does Singleton pattern help in Selenium Webdriver?

Keep track of same driver instance throughout execution.

DBMS connectivity.

Loading external files like properties, excel etc once rather than loading again and again.

Logger.

So wherever you feel, you should have single instance of any class, you should use singleton design pattern.  For example: If database connection is already established, you should not create new connection.

We will see implementation of singleton design pattern for keep track of driver.

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.chrome.ChromeDriver**;

**public** **class** **SingletonBrowserClass** {

// instance of singleton class

**private** **static** SingletonBrowserClass instanceOfSingletonBrowserClass=**null**;

**private** WebDriver driver;

// Constructor

**private** **SingletonBrowserClass**(){

System.setProperty("webdriver.chrome.driver","./exefiles/chromedriver.exe");

driver= **new** ChromeDriver();

}

// TO create instance of class

**public** **static** SingletonBrowserClass **getInstanceOfSingletonBrowserClass**(){

**if**(instanceOfSingletonBrowserClass==**null**){

instanceOfSingletonBrowserClass = **new** SingletonBrowserClass();

}

**return** instanceOfSingletonBrowserClass;

}

// To get driver

**public** WebDriver **getDriver**()

{

**return** driver;

}

}

**import** **org.openqa.selenium.WebDriver**;

**public** **class** **LuanchURL** {

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

SingletonBrowserClass sbc1= SingletonBrowserClass.getInstanceOfSingletonBrowserClass();

WebDriver driver1 = sbc1.getDriver();

SingletonBrowserClass sbc2= SingletonBrowserClass.getInstanceOfSingletonBrowserClass();

WebDriver driver2 = sbc2.getDriver();

driver2.get("https://www.google.com");

}

}

**Explanation:** When you run LoadURL.java, you will see browser will be launched and url will be opened in same browser. We have instantiated two instance of class SingletonBrowserClass, but both give the same instance of driver.

**107. How to select an option in list box?**

**Answer:**Please refer below link

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

**108. How do you count the total number of rows in a web table?**

**Answer:**Below is the sample script given

**import** **java.text.ParseException**;

**import** **java.util.List**;

**import** **org.openqa.selenium.By**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.WebElement**;

**import** **org.openqa.selenium.chrome.ChromeDriver**;

**public** **class** **Noofrowsandcols** {

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

WebDriver wd;

System.setProperty("webdriver.chrome.driver","G://chromedriver.exe");

wd= **new** ChromeDriver();

wd.get("URL");

//No.of Columns

List col = wd.findElements(By.xpath(".//\*[@id=\"leftcontainer\"]/table/thead/tr/th"));

System.out.println("No of cols are : " +col.size());

//No.of rows

List rows = wd.findElements(By.xpath(".//\*[@id='leftcontainer']/table/tbody/tr/td[1]"));

System.out.println("No of rows are : " + rows.size());

wd.close();

}

}

**109. How to select a checkbox present in a grid?**

**Answer:**

**110. How to get text from hidden elements?**

**Answer:**

**111. How do you attach screenshot to test, which tool is using for screenshot attachment?**

**Answer:**A task to take screenshot

Access http://google.com

Search for cars

Verify if there is any link containing the word “car”

Take screenshot

Adding screenshots to the report

String filePath = screenShotName.toString();

String path = "<img src="\"file://"" alt="\"\"/" />";

Reporter.log(path);

**Complete Working Code**

**public** **class** **TestNGDefaultReport** {

**static** WebDriver driver;

**@BeforeSuite**

**public** **void** **setup**(){

System.setProperty("webdriver.chrome.driver","D:\\MyTest\\chromedriver.exe");

System.setProperty("org.uncommons.reportng.escape-output", "false");

driver = **new** ChromeDriver();

}

**@BeforeMethod**

**public** **void** **beforeEachMethod**(){

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

}

//Test case 1

**@Test**

**public** **void** **cars**() **throws** Exception {

System.out.println("I am Test method and I am searching for cars");

driver.findElement(By.name("q")).sendKeys("Cars");

driver.findElement(By.name("btnG")).click();

//Wait for the results to appear

Thread.sleep(**2000**);

takeScreenshot();

**if**(driver.findElement(By.partialLinkText("car")).isDisplayed()){

Assert.assertTrue(**true**);

}

**else**{

Assert.assertTrue(**false**);

}

}

**@AfterSuite**

**public** **void** **endOfSuite**(){

System.out.println("I am the end of suite");

driver.quit();

}

**public** **static** **void** **takeScreenshot**() **throws** Exception {

String timeStamp;

File screenShotName;

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

//The below method will save the screen shot in d drive with name "screenshot.png"

timeStamp = **new** SimpleDateFormat("yyyyMMdd\_HHmmss").format(Calendar.getInstance().getTime());

screenShotName = **new** File("D:\\MyTest\\Screenshots\\"+timeStamp+".png");

FileUtils.copyFile(scrFile, screenShotName);

String filePath = screenShotName.toString();

String path = "<img src="\"file://"" alt="\"\"/" />";

Reporter.log(path);

}

}

You make you reports more meaningful and presentable by adding screenshots to your reports.

**112. How will you read CSV file and write some sample code?**

**Answer:**Below is the sample script given

**package** Test;

**import** **java.io.FileReader**;

**import** **java.io.IOException**;

**import** **java.io.Reader**;

**import** **java.util.List**;

**import** **au.com.bytecode.opencsv.CSVReader**;

**public** **class** **CSVReaderDemo** {

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

String path = "C:\\Users\\Avinash\\Desktop\\csvtest.csv";

Reader reader = **new** FileReader(path);

CSVReader csvreader = **new** CSVReader(reader);

List<String[]> data = csvreader.readAll();

**for**(String[] d : data){

**for**(String c : d ){

System.out.println(c);

}

}

}

}

**113. How to read data from properties file in Selenium?**

**Answer:**  
**Properties File**

URL ::http://gmail.com

User name::testuser

Password::password123

Below is the sample script given

**public** **class** **ReadFileData** {

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

File file = **new** File("D:/ReadData.properties");

FileInputStream fileInput = **null**;

**try** {

fileInput = **new** FileInputStream(file);

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

Properties prop = **new** Properties();

//load properties file

**try** {

prop.load(fileInput);

} **catch** (IOException e) {

e.printStackTrace();

}

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

driver.get(prop.getProperty("URL"));

driver.findElement(By.id("Email")).sendKeys(prop.getProperty("username"));

driver.findElement(By.id("Passwd")).sendKeys(prop.getProperty("password"));

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

System.out.println("URL ::" + prop.getProperty("URL"));

System.out.println("User name::" +prop.getProperty("username"));

System.out.println("Password::" +prop.getProperty("password"));

}

}

**114. How to handle internationalization using Selenium WebDriver?**

**Answer:**

**115. How to verify a particular image out of many images on a web page and verify its size using Selenium WebDriver?**

**116. How do you find out Active elements using Selenium WebDriver?**

**Answer:**Below is the sample script given

**public** **class** **ActiveElements** {

WebDriver driver;

**@Test**

**public** **void** **activeElementsTest**() **throws** InterruptedException {

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

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

driver = **new** ChromeDriver();

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

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

List<WebElement>list = driver.findElements(By.xpath("//\*"));

System.out.println("Total Elements: "+list.size());

**int** enableCount=**0**;

**int** disableCount=**0**;

**for**(WebElement **ele:**list) {

**boolean** actualValue = ele.isEnabled();

**if** (actualValue) {

//System.out.println("Element is enabled: "+ele.getText());

enableCount=enableCount+**1**;

}

**else** {

//System.out.println("Element is disabled: "+ele.getText());

Thread.sleep(**2000**);

disableCount=disableCount+**1**;

}

}

System.out.println("Enable Element Count: "+enableCount);

System.out.println("Disable Element Count: "+disableCount);

driver.close();

}

}

**Output:**

Total **Elements:** **205**

Enable Element **Count:** **204**

Disable Element **Count:** **1**

**117. Write a code to click on a button which is inside a nested iframe?**

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

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

**118. How to handle the dynamic alerts which don’t always appear?**

**Answer:**Below given webdriver code Is just for example. It Is just explaining the way of using try catch block to handle unexpected alert. You can use such try catch block In that area where you are facing unexpected alerts very frequently.

Below is the sample script given:

**public** **class** **unexpected\_alert** {

WebDriver driver = **null**;

**@BeforeTest**

**public** **void** **setup**() **throws** Exception {

System.setProperty("webdriver.gecko.driver", "D:\\Selenium Files\\geckodriver.exe");

driver = **new** FirefoxDriver();

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

driver.manage().timeouts().implicitlyWait(**5**, TimeUnit.SECONDS);

driver.get("URL");

}

**@AfterTest**

**public** **void** **tearDown**() **throws** Exception {

driver.quit();

}

**@Test**

**public** **void** **Text**() **throws** InterruptedException {

//To handle unexpected alert on page load.

**try**{

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

}**catch**(Exception e){

System.out.println("unexpected alert not present");

}

driver.findElement(By.xpath("//input[@name='fname']")).sendKeys("fname");

driver.findElement(By.xpath("//input[@name='lname']")).sendKeys("lname");

driver.findElement(By.xpath("//input[@type='submit']")).click();

}

}

**119. How to verify whether the size of a div is 320px or not?**

**Answer:**

**120. How to check the cursor type changes on hovering on a link?**

**Answer:**Below is a situation where I had to automate the test which checks that the checkbox is read-only and the cursor does not change its type to 'pointer' after hovering the mouse over that checkbox.

Here's how I achieved it with Selenium.

Solution : There is a CSS value to the element which may change after the hover of the mouse on the element. We can capture the CSS value with selenium's WebElement's method getCssValue().

Here's the test code :

**@Test**

**public** **void** **testCursorDoesntChangeOnHover**(){

//Identify the element on the hover of which you want to verify the pointer type

WebElement e = driver.findElement("Element Locator"));

System.out.println("Cursor before hovering on: " + e.getCssValue("cursor"));

System.out.println("Hovering on the element...");

//Hover the mouse over that element

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

builder.moveToElement(e);

builder.build().perform();

//Check that the cusrsor does not change to pointer

String cursorTypeAfter = e.getCssValue("cursor");

System.out.println("Cursor after hovering on: " + cursorTypeAfter);

//Verify that the cursor type has not changed to 'pointer'

Assert.assertFalse(cursorTypeAfter.equalsIgnoreCase("pointer"), "Cursor type changed !");

}

**121. How to verify that the font-size of a text is 12px?**

**122. How to verify the presence of tooltips on a web page?**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2020/01/how-to-handle-tooltip-in-selenium.html>

**123. How to enter :(colon using web driver) ?**

**Answer:**Please refer below code

**public** **class** **Colon** {

WebDriver driver;

**@Test**

**public** **void** **colonTest**() {

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

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

driver = **new** ChromeDriver();

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

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

WebElement ele=driver.findElement(By.name("q"));

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

//act.keyDown(Keys.SHIFT);

act.moveToElement(ele).keyDown(Keys.SHIFT).perform();

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

act.moveToElement(ele).keyUp(Keys.SHIFT).perform();

}

}

**124. How to check whether on click of an element ,a new tab will be opened before clicking on that element?**

**125. How to type text in a new line inside a text area?**

**Answer:**'\n' is usually used to type text in a new line inside a textarea. So with Selenium Webdriver, we can achieve this in following way:

WebElement webelement = driver.findElement(By.id("ele1"));

webelement.sendKeys(“This is line one.\n This is line two.”);

**126. How to add the screenshot to result window?**

**127. How to enter date into date field using JavaScript in WebDriver?**

**Answer:**In old applications, the datepicker was in old style, that is, it was a kind of dropdowns, by using select class. But in new applications now, Datepicker are not Select element.

Datepicker are in fact table with set of rows and columns. To select a date, you just have to navigate to the cell where our desired date is present.

So, to handle this datePicker, we will write a java script, in which firstly we will enable the datebox (which is restricted by default to enter the value manually rather than selecting it from calendar pop-up). Then We will clear that filed & then enter the date by using sendKeys() method.  
Below is the sample script given

**import** **java.util.concurrent.TimeUnit**;

**import** **org.openqa.selenium.By**;

**import** **org.openqa.selenium.JavascriptExecutor**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.WebElement**;

**import** **org.openqa.selenium.firefox.FirefoxDriver**;

**public** **class** **HandleDatePcikerUsingJavaScript** {

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

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

driver.navigate().to("https://www.redbus.in/");

driver.manage().timeouts().implicitlyWait(**10**, TimeUnit.SECONDS);

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

//This will select From date

WebElement elementFromDate = driver.findElement(By.id("onward\_cal"));

((JavascriptExecutor)driver).executeScript ("document.getElementById('onward\_cal').removeAttribute('readonly',0);"); // Enables the from date box

elementFromDate.clear();

elementFromDate.sendKeys("18-Aug-2017"); //Enter date in required format

}

}

**128. How to take complete screen shot of the application?**

**Answer:**What is aShot?

aShot is a WebDriver Screenshot utility. It takes a screenshot of the WebElement on different platforms (i.e. desktop browsers, iOS Simulator Mobile Safari, Android Emulator Browser).

aShot might be configured to handle browsers with the viewport problem. This gives a screenshot of the entire page even for Chrome, Mobile Safari, etc

Below mentioned script shows how to capture full page screenshot using Selenium WebDriver:

**import** **java.io.File**;

**import** **javax.imageio.ImageIO**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.firefox.FirefoxDriver**;

**import** **ru.yandex.qatools.ashot.AShot**;

**import** **ru.yandex.qatools.ashot.Screenshot**;

**import** **ru.yandex.qatools.ashot.shooting.ShootingStrategies**;

**public** **class** **FullPageScreenshot** {

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

//Modify the path of the GeckoDriver in the below step based on your local system path

System.setProperty("webdriver.gecko.driver","D://Selenium Environment//Drivers//geckodriver.exe");

// Instantiation of driver object. To launch Firefox browser

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

// To oepn URL "https://automationtestinginsider.com"

driver.get("https://www.automationtestinginsider.com");

Thread.sleep(**2000**);

Screenshot fpScreenshot = **new** AShot().shootingStrategy(ShootingStrategies.viewportPasting(**1000**)).takeScreenshot(driver);

ImageIO.write(fpScreenshot.getImage(),"PNG",**new** File("D:///FullPageScreenshot.png"));

}

}

**129. How to take the screen shot of required element?**

**Answer:**We can get the element screenshot by cropping entire page screenshot as below:

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

WebElement ele = driver.findElement(By.id("hplogo"));

// Get entire page screenshot

File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

BufferedImage  fullImg = ImageIO.read(screenshot);

// Get the location of element on the page

Point point = ele.getLocation();

// Get width and height of the element

int eleWidth = ele.getSize().getWidth();

int eleHeight = ele.getSize().getHeight();

// Crop the entire page screenshot to get only element screenshot

BufferedImage eleScreenshot= fullImg.getSubimage(point.getX(), point.getY(),

    eleWidth, eleHeight);

ImageIO.write(eleScreenshot, "png", screenshot);

// Copy the element screenshot to disk

File screenshotLocation = new File("C:\\images\\GoogleLogo\_screenshot.png");

FileUtils.copyFile(screenshot, screenshotLocation);

**130. How to get the partial server message using Selenium WebDriver?**

**131. How to scroll the web page by (30%,80%) using Selenium WebDriver?**

**Answer:**Refer to the Selenium script below, for performing a scroll down action on the Firefox browser.

**import** **org.openqa.selenium.JavascriptExecutor**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.firefox.FirefoxDriver**;

**import** **org.testng.annotations.Test**;

**public** **class** **HandleScroll**

{

**@Test**

**public** **void** **scrollDown**()

{

System.setProperty("webdriver.gecko.driver","D://Selenium Environment//Drivers//geckodriver.exe");

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

driver.navigate().to("Website URL");

//to perform Scroll on application using Selenium

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,350)", "");

}

}

**Output:** The code initializes Gecko driver for Firefox. Then the Firefox browser is launched, and it navigates to the specified website URL. Once the website loads, the browser window is vertically scrolled down by 350 pixels.

If a user needs to scroll up, they just need to modify the pixel value of the second parameter (in this case 350) to negative value (-350).

**132. How to handle Confirmation Pop-Up?**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2020/01/alert-interface-and-handle-javascript.html>

**133. How to specify some delay in loading WebPage?**

**134. How to automate videos?**

**Answer:**Testing a video is always a challenging task. Most of the video based web sites are manually tested & are out of scope of automation.

However, the javascript executor in selenium webdriver will enable us to play around with the video components. Thus, few of the test scenarios w.r.t video testing can be automated using the below approach.

Below is the sample script given

System.setProperty("webdriver.chrome.driver", "ChromeDriver Path");

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

//Open the page containing video component.

driver.get("URL");

driver.manage().timeouts().implicitlyWait(**15**, TimeUnit.SECONDS);

//If video is placed in embed tag or iframe, navigate to the source.

//However, we can even switch to frame. [Else, skip this step]

WebElement elm = driver.findElement(By.id("video\_iframe"));

String urlStr = elm.getAttribute("src");

System.out.println("Video Url : " + urlStr);

driver.navigate().to(urlStr);

cdrv.manage().timeouts().implicitlyWait(**15**, TimeUnit.SECONDS);

JavascriptExecutor jse = (JavascriptExecutor) driver;

//Click on play button

jse.executeScript("jwplayer().play();");

Thread.sleep(**2000**);

//Pause

jse.executeScript("jwplayer().pause()");

Thread.sleep(**2000**);

//Play

jse.executeScript("jwplayer().play();");

Thread.sleep(**2000**);

// Set Volume

Thread.sleep(**2000**);

jse.executeScript("jwplayer().setVolume(50);");

Thread.sleep(**2000**);

//Mute Player

jse.executeScript("jwplayer().setMute(true);");

Thread.sleep(**2000**);

//UnMute Player

jse.executeScript("jwplayer().setMute(false);");

Thread.sleep(**2000**);

//Stop the player

jse.executeScript("jwplayer().stop()");

Thread.sleep(**2000**);

driver.quit();

**135. How do you manage re-running only failed test cases?**

**Answer:**

**136. Write a program to find all options from a drop-down & then Sort them?**

**Answer:**Please refer below program

**import** **java.util.ArrayList**;

**import** **java.util.Collections**;

**import** **java.util.List**;

**import** **org.openqa.selenium.By**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.WebElement**;

**import** **org.openqa.selenium.chrome.ChromeDriver**;

**import** **org.openqa.selenium.support.ui.Select**;

**public** **class** **ValidationOfSortedDropdownAscending** {

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

// Launching browser

System.setProperty("webdriver.chrome.driver","./exefiles/chromedriver.exe");

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

// Launching URL

driver.get("URL");

// Locating select tag web element

WebElement singleSelectTagDropdownWebElement= driver.findElement(By.id("SingleDD"));

Select dropdown = **new** Select(singleSelectTagDropdownWebElement);

// Get all options

List allOptionsElement = dropdown.getOptions();

// Creating a list to store drop down options

List options = **new** ArrayList();

// Storing options in list

**for**(WebElement optionElement : allOptionsElement)

{

options.add(optionElement.getText());

}

// Removing "Select" option as it is not actual option

options.remove("Select");

// Default order of option in drop down

System.out.println("Options in dropdown with Default order :"+options);

// Creating a temp list to sort

List tempList = **new** ArrayList&lt;&gt;(options);

// Sort list ascending

Collections.sort(tempList);

System.out.println("Sorted List "+ tempList);

// equals() method checks for two lists if they contain the same elements in the same order.

**boolean** ifSortedAscending = options.equals(tempList);

**if**(ifSortedAscending)

{

System.out.println("List is sorted");

}

**else**

System.out.println("List is not sorted.");

driver.quit();

}

}

**137. How to click a button without using click() and without using CSS & XPath selectors?**

**Answer:**Using the non-native Javascript Executor:

((JavascriptExecutor) driver).executeScript("arguments[0].click();", yourelement);

or by using Javascript Library:

JavascriptLibrary jsLib = new JavascriptLibrary();`

jsLib.callEmbeddedSelenium(driver, "triggerMouseEventAt", we, "click", "0,0");

**138. How to run Selenium WebDriver script in Firefox browser using DesiredCapabilities?**

**Answer:**Below is the sample script given

**public** **class** **Second** {

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

// System Property for Gecko Driver

System.setProperty("webdriver.gecko.driver","D:\\GeckoDriver\\geckodriver.exe" );

// Initialize Gecko Driver using Desired Capabilities Class

DesiredCapabilities capabilities = DesiredCapabilities.firefox();

capabilities.setCapability("marionette",**true**);

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

// Launch Website

driver.navigate().to("https://www.google.com/");

// Click on the Search text box and send value

driver.findElement(By.name("q")).sendKeys("Java");

}

}

**139. How to handle Untrusted SSL certificate error in IE browser?**

**140. How to refresh a page without using context click?**

**Answer:**

1. Using sendKeys method

driver.findElement(By.id("firstname-placeholder")).sendKeys(Keys.F5);

2. Using navigate.refresh() method

driver.navigate().refresh();

3. Using navigate.to() method

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

4. Using get() method

driver.get(driver.getCurrentUrl());

**141. Can you send a code for printing in selenium?**

**Answer:**

**142. How to get the name of browser using Web Driver?**

**Answer:**You can use below code to know browser name, version and OS details:

Capabilities cap = ((RemoteWebDriver) driver).getCapabilities();

String browserName = cap.getBrowserName().toLowerCase();

System.out.println(browserName);

String os = cap.getPlatform().toString();

System.out.println(os);

String v = cap.getVersion().toString();

System.out.println(v);

**143. How to get text from captcha image?**

**Answer:**

driver.findElement(By.xpath(".//\*[@id='SkipCaptcha']")).click();

String attr = ie.findElement(By.xpath(".//\*[@id='SkipCaptcha']")).getAttribute("value"); System.out.println("The value of the attribute 'Name' is " + attr);

**144. Is there a way to click hidden LINK in web driver?**

**Answer:**First store that element in object, let's say element and then write following code to click on that hidden element:

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("arguments[0].click();", element);

**145. How to disable cookies in browser.?**

**Answer:**Firefox

FirefoxProfile profile = new ProfilesIni().getProfile("default");

profile.setPreference("network.cookie.cookieBehavior", 2);

driver = new FirefoxDriver(profile);

Chrome

Map prefs = new HashMap();

prefs.put("profile.default\_content\_settings.cookies", 2);

ChromeOptions options = new ChromeOptions();

options.setExperimentalOptions("prefs", prefs);

driver = new ChromeDriver(options);

**146. How to change user agent in Firefox by selenium web driver.?**

**147. How to ZIP files in Selenium with an Example?**

**Answer:**How to ZIP a file in SELENIUM

Steps to ZIP(Compare with Code and understand)

1. Create an File object for the SOURCE FOLDER, in which files are present.

2. Create another File object for the destination folder, for ZIP file.

3. Create a ZipOutputStream Object.

4. Create a BufferedInputStream Object.

5. Create a Byte Array Object of size 1000.

6. Create a String Array, and collect all files from SOURCE FOLDER as

    LIST  using File Object for source folder.

7. Loop through String Array and for each element perform below action.

8. INSIDE FOR LOOP

   1. Get the SOURCE FOLDER path and append each file to the path

       with default size, and put it in BufferedInputStream Object.

   2. Through ZipOutputStream Object, make an ZIP Entry for each File.

   3. Through While Loop read the data bit by bit untill 1000 bytes from

       BufferedInputStream Object and write to ZipOutputStream Object.

      9. INSIDE WHILE LOOP

       1. Read the BufferedInputStream Object.

       2. Write to ZipOutputStream Object.

4. After writing make a Close Entry using ZipOutputStream Object.

5. Flush the data using ZipOutputStream Object.

6. Close the ZipOutputStream Object.

———————————————————————–

**try**

{

File inFolder=**new** File(“**D:**\\sel\\framework\\tests”);

File outFolder=**new** File(“**D:**\\sel\\framework\\results\\results.zip”);

ZipOutputStream out = **new** ZipOutputStream(**new** BufferedOutputStream(

**new** **FileOutputStream**(outFolder)));

BufferedInputStream in = **null**;

**byte**[] data = **new** **byte**[**1000**];

String files[] = inFolder.list();

**for** (**int** i=**0**; i<files.length; i++)

{

in = **new** BufferedInputStream(**new** FileInputStream

(inFolder.getPath() + “/” + files[i]), **1000**);

out.putNextEntry(**new** ZipEntry(files[i]));

**int** count;

**while**((count = in.read(data,**0**,**1000**)) != -**1**)

{

out.write(data, **0**, count);

}

out.closeEntry();

}

out.flush();

out.close();

}

**catch**(Exception e)

{

e.printStackTrace();

}

**148. Write program for checking mails and deleting them?**

**Answer:**

**149. Write a program to return the number of rows and columns in a web-table?**

**Answer:**

**Image**

We need to get WebElement of <tbody> tag, so that we can get the count of <tr> tags & ultimately total number of rows in the web table.

So, Below is the sample code, with which we can get total number of rows in the web table.

WebElement TogetRows = driver.findElement(By.xpath("//table[@id='users\_table']/tbody"));

List<WebElement>TotalRowsList = TogetRows.findElements(By.tagName("tr"));

System.out.println("Total number of Rows in the table are : "+ TotalRowsList.size());

In order to get total number of columns in the web table, what we need to do is, count total number of <td> tags in first <tr> tagname.

Below is the sample code, with which we can get total number of columns easily.

WebElement ToGetColumns = driver.findElement(By.xpath("//table[@id='users\_table']/tbody/tr"));

List<WebElement> TotalColsList = ToGetColumns.findElements(By.tagName("td"));

System.out.println("Total Number of Columns in the table are: "+TotalColsList.size());

**150. 3 frames one upon another, how to switch without passing index, web-element, string and integer?**

**Answer:**Please refer below link

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

Some times when there are multiple Frames (Frame in side a frame), we need to first switch to the parent frame and then we need to switch to the child frame. below is the code snippet to work with multiple frames.

**public** **void** **switchToFrame**(String ParentFrame, String ChildFrame) {

**try** {

driver.switchTo().frame(ParentFrame).switchTo().frame(ChildFrame);

System.out.println("Navigated to innerframe with id " + ChildFrame

+ "which is present on frame with id" + ParentFrame);

} **catch** (NoSuchFrameException e) {

System.out.println("Unable to locate frame with id " + ParentFrame

+ " or " + ChildFrame + e.getStackTrace());

} **catch** (Exception e) {

System.out.println("Unable to navigate to innerframe with id "

+ ChildFrame + "which is present on frame with id"

+ ParentFrame + e.getStackTrace());

}

}

After working with the frames, main important is to come back to the web page. **if** we don't **switch** back to the **default** page, driver will **throw** an exception. Below is the code snippet to **switch** back to the **default** content.

**public** **void** **switchtoDefaultFrame**() {

**try** {

driver.switchTo().defaultContent();

System.out.println("Navigated back to webpage from frame");

} **catch** (Exception e) {

System.out

.println("unable to navigate back to main webpage from frame"

+ e.getStackTrace());

}

}

**151. How to handle security questions -- 3 security questions and changes every time.**

**152. How to switch from frame to main window? With syntax.**

**Answer:**After working with the frames, main important is to come back to the web page. if we don't switch back to the default page, driver will throw an exception. Below is the code snippet to switch back to the default content.

**public** **void** **switchtoDefaultFrame**() {

**try** {

driver.switchTo().defaultContent();

System.out.println("Navigated back to webpage from frame");

} **catch** (Exception e) {

System.out

.println("unable to navigate back to main webpage from frame"

+ e.getStackTrace());

}

}

**153. I have 10 window, how to switch 5th window?**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2020/02/handle-multiple-windows-in-selenium.html>

**154. How to show the output of the two commands - eco %path%,  set**

**155. How to handle frames when there is no frame id or name present in the frame source?**

**Answer:**You can use cssSelector,

driver.switchTo().frame(driver.findElement(By.cssSelector("iframe[title='Fill Quote']")));

**156. How to download file using selenium**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2020/02/upload-and-download-file-in-selenium.html>

**157. How to close multiple windows in reverse order?**

**158. How to select the element from dropdown without using select class?**

**Answer:**Below is the sample script given

**import** **java.util.concurrent.TimeUnit**;

**import** **org.openqa.selenium.By**;

**import** **org.openqa.selenium.Keys**;

**import** **org.openqa.selenium.WebDriver**;

**import** **org.openqa.selenium.firefox.FirefoxDriver**;

**import** **org.openqa.selenium.interactions.Actions**;

**public** **class** **SelectFromDropDown** {

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

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

driver.get("https://makemytrip.com/");

driver.manage().timeouts().implicitlyWait(**10**, TimeUnit.SECONDS);

driver.findElement(By.xpath("//div//span[2][contains(@class,'flL travelers rght\_space room\_sec1')]")).click();

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

act.sendKeys(Keys.chord(Keys.DOWN,Keys.DOWN)).perform(); //press down key two times to select 3

}

}

**159. How to perform double click without using selenium?**

**160. How to handle dynamic web pages?**

**Answer:**A page that has dynamic content has elements of the page that are hidden when you first get to the website. For example, if you were to press a button on a page and it loaded new content, like a pop-up form, without going to another page, that’s a dynamic website. It can also be a user-specific web page that changes day-to-day depending on when it’s accessed, where it’s accessed, user preferences, etc.

Selenium’s Solution

Selenium actually has two built-in solutions for testing dynamic loading content that we recommend you should use: the explicit and the implicit wait.

Please refer below very good article on waits

[**https://www.automationtestinginsider.com/2020/02/waits-in-selenium-webdriver.html**](https://www.automationtestinginsider.com/2020/02/waits-in-selenium-webdriver.html)

**162. How to skip a particular cell value?**

**163. How to find text/values in the webtable using single line of code in selenium?**

**164. How to click last element of a webpage**

**165. How to handle drop down when drop down not present in select tag**

**Answer:**Please refer below link

<https://www.automationtestinginsider.com/2019/12/handle-bootstrap-dropdown-in-selenium.html>

**166. How to verify text sent through senkeys correct or not?**

**167. How to verify an object on multiple pages**

**Answer:**It depends on the framework structure you're following to verify an object. If you're following Page object model (POM), then create a base class and define the locator of an object you want to verify. Method to verify its presence looks like (in Java):

 driver.findElement(by)

 or using dynamic wait as:

 WebDriverWait wait = new WebDriverWait(webDriver, 60);

 wait.until(new ExpectedCondition<Boolean>() {

 @Override

 public Boolean apply(WebDriver driver) {

 if(driver.findElement(by) != null)

 return true;

 return false;

 }

 });

**168. How to Find particular cell in excell sheet**

**169. Want to click 5 times.on first click stale element how to handle?**

**170. How to read excel from xml?**

**171. How to parse the data to xml?**

**172. How to scroll bar without java script?**

**173. How to handle prompt msg which disappears in 2 sec?**

**174. How to handle ascending drop down?**