**3. what are limitations in selenium web driver?**

* It does not support and non-web-based applications, it only supports web based applications.

It’s an open source tool so in case of any technical issues you need to rely on the selenium community forums to get your issue resolved. You need to know at least one of the supported language very well to automate your application successfully. No inbuilt reporting capability so you need plugins like JUnit and TestNG for test reports. Lot of challenges with IE browser.

**6. installing/configure selenium?**

* Step 1 - Install Java on your computer

Step 2 - Install Eclipse IDE

Step 3 - Download the Selenium Java Client Driver

Step 4 - Configure Eclipse IDE with WebDriver

**9. what are different ways of locating elements in selenium?**

* There are some browser tools that you can use to identify web elements in the DOM easier. These are:

Firebug for Firefox

Google Developer Tools for Chrome

Web Inspector for Safari

Locating Elements with Selenium WebDriver, findElement() method returns and WebElement and findElements() returns a list of WebElements.

1. By ID:

in Java: driver.findElement(By.id("element id"))

2. By CLASS:

in Java: driver.findElement(By.className("element class"))

3. By NAME:

name=name\_of\_the\_element filter=value\_of\_filter

**12. which is fastest way to identify elements in web page?**

* Finding web elements with Selenium WebDriver by ID is usually the fastest option, but here is the list of the best and fastest selectors Selenium WebDriver Tests run faster:

**ID selectors (By.ID – Matches by @id attribute):**

IDs are the safest, fastest locator option and should always be your first choice

IDs should be unique in every page according to W3C website

even if the DOM changes, if the ID is still there, then WebDriver can still locate it

fastest locator as it uses the document.getElementById() javascript command which is optimised by many browsers.

**CSS and Name selectors (Matches by CSS selector or @name attribute)**

**XPath locators (Matches with arbitrary XPath expression):** Most flexible to build reliable web element locators

very slow locator (particularly in IE) since in order to locate the element it needs to traverse the whole DOM of the page which is a time consuming operation

check here how to identify and validate XPath locators inside Chrome Developer Tool, also you can install a Firebug extension to be able to identify and validate XPaths easily with FirePath from FireBug (adds a development tool to edit, inspect and generate XPath 1.0 expressions, CSS 3 selectors and JQuery selectors).

**15. what is absolute path and relative path in xpath?**

**Absolute Xpath:** It uses Complete path from the Root Element to the desire element.

**Relative Xpath:** You can simply start by referencing the element you want and go from there.

Always Relative Xpaths are preferred as they are not the complete paths from the Root element. (//html//body). Because in future any of the web element when added/Removed then Absolute Xpath changes. So Always use Relative Xpaths in your Automation.

**18. different types of waits or synchronization in selenium webdriver?**

* Synchronization can be classified into two categories:

1. Unconditional

2. Conditional Synchronization

**Unconditional:** In this we just specify timeout value only. We will make the tool to wait until certain amount of time and then proceed further.

Examples: Wait() and Thread.Sleep();

**Conditional Synchronization:** We specify a condition along with timeout value, so that tool waits to check for the condition and then come out if nothing happens.

It is very important to set the timeout value in conditional synchronization, because the tool should proceed further instead of making the tool to wait for a condition to satisfy.

In Selenium we have implicit Wait and Explicit Wait conditional statements.

**Implicit Wait:** driver.manage.TimeOuts.implicitwait(6,Timeunit.SECONDS);

WebDriver driver = new FirefoxDriver();

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

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

**Explicit Wait:**

/\*Explicit wait for state dropdown field\*/

WebDriverWait wait = new WebDriverWait(driver, 10);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("statedropdown")));

**21. how to save screen shots using selenium webdriver?**

* In selenium webdriver, we can take the screen shot using the below command.

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

**24. how to handle multiple windows in selenium webdriver?**

* when we are likely to deal with multiple windows we need to first consider a few situations

Filling forms may require to select the date from a separately opened window. Clicking on some link/button can kick-off yet another window. Handling Advertisement windows. Hence, we can come up with various scenarios depending upon the application.

* When we have multiple windows in test automation, all we need to do is switching the focus from one window to another. Let us understand the same in the following way:

Window A has a link "Link1" and we need to click on the link (click event).

Window B displays and we perform some actions.

The entire process can be fundamentally segregated into following steps:

Step 1 : Clicking on Link1 on Window A

A new Window B is opened.

Step 2 : Save reference for Window A

Step 3 : Create reference for Window B

Step 3 : Move Focus from Window A to Window B

Window B is active now

Step 3 : Perform Actions on Window B

Complete the entire set of Actions

Step 4 : Move Focus from Window B to Window A …Window A is active now

**30. what is desired capabilities in selenium webdriver?**

Here we will see a different type of desired capabilities methods and see how to use one of this method "setCapability Method".

getBrowserName()

public java.lang.String getBrowserName()

setBrowserName()

public void setBrowserName(java.lang.String browserName)

getVersion()

public java.lang.String getVersion()

setVersion()

public void setVersion(java.lang.String version)

getPlatform()

public Platform getPlatform()

setPlatform()

public Platform getPlatform()

getCapability Method

The getCapability method of the Desired Capabilities class can be used to get the capability that is in use currently in the system.

public java.lang.Object getCapability(java.lang.String capabilityName)

**33. how to set language while opening website?**

System.setProperty(“webdriver.chrome.driver”,”D:/DollarArchive/chromedriver.exe”);

ChromeOptions options = new ChromeOptions();

options.addArguments(“–lang= sl”);

ChromeDriver driver = new ChromeDriver(options);

driver.get(“http://google.co.in&#8221;);

Unfortunately it wont work for IE browser, We need to change it manually.

**36. how to handle windows based popups (upload and dropdown)?**

* 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.

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

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

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

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

**39. write code to verify any application login page is working or not?**

In my database I have the username = user@javachap.com and password = javachap

If I run the code below, it passes the test although the username and password does not exist in my database.

@Test

public void testLogin()

{

String username="abc";

String password="123";

boolean valueFound=false;

// Check the db

try

{

pstmt=conn.prepareCall("select \* from user where USR\_EMAIL=? and USD\_PASSWORD=?");

pstmt.setString(1,username);

pstmt.setString(2,password);

rs=pstmt.executeQuery();

valueFound = rs.next();

}

catch(Exception e)

{

// report some error

}

**43. how to select items from dropdown/select box?**

WebElement select = driver.findElement(By.id("gender"));

List<WebElement> options = select.findElements(By.tagName("Male"));

for (WebElement option : options) {

if("Germany".equals(option.getText()))

option.click();

}

**46. how to know if checkbox is checked or not in webpage?**

Find out if a single checkbox is checked or not, returns true or false:

$('#checkBox').attr('checked');

Find all checked checkboxes:

$('input[type=checkbox]:checked');

**49. tell me code to pass values from parent window to child window?**

Inside the Parent Page I have placed a TextBox which is read-only and a button to open the child popup window using the SelectName JavaScript function.

<table border="0" cellpadding="0" cellspacing="0">

<tr>

<td>

Name:&nbsp;

</td>

<td>

<input type="text" id="txtName" readonly="readonly" />

</td>

<td>

<input type="button" value="Select Name" onclick="SelectName()" />

</td>

</tr>

</table>

<script type="text/javascript">

var popup;

function SelectName() {

popup = window.open("Popup.htm", "Popup", "width=300,height=100");

popup.focus();

}

</script>

The Popup Child Window Page has a DropDownList to choose the name of the person and a button which when clicked calls the SetName JavaScript function.

<select name="ddlNames" id="ddlNames">

<option value="Mudassar Khan">Mudassar Khan</option>

<option value="John Hammond">John Hammond</option>

<option value="Mike Stanley">Mike Stanley</option>

</select>

<br />

<br />

<input type="button" value="Select" onclick="SetName();" />

<script type="text/javascript">

function SetName() {

if (window.opener != null && !window.opener.closed) {

var txtName = window.opener.document.getElementById("txtName");

txtName.value = document.getElementById("ddlNames").value;

}

window.close();

}

</script>

**55. write code on how to use javascriptexecutor?**

JavaScriptExecutor is an Interface that helps to execute JavaScript through Selenium Webdriver. JavaScriptExecutor provides two methods "executescript" & "executeAsyncScript" to run javascript on the selected window or current page.

The basic syntax for JavascriptExecutor is given below:

**Syntax:**

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

**58. difference between assert and verify?**

* When an “assert” fails, the test will be aborted. Where if a “verify” fails, the test will continue executing and logging the failure. So when Assertion fails all test steps after that line of code are skipped . To resolve this we write this command in try catch block.

**61. difference between driver.close and driver.quit methods?**

* driver.close and driver.quit are two different methods for closing the browser session in Selenium WebDriver. Understanding both of them and knowing when to use which method is important in your test execution. Therefore, in this article, we have tried to throw light on both these methods.

driver.close – It closes the the browser window on which the focus is set.

driver.quit – It basically calls driver.dispose method which in turn closes all the browser windows and ends the WebDriver session gracefully.

**64. common exceptions in selenium?**

* NoSuchElementException

This exception occurs when WebDriver is unable to identify the elements during run time. Due to wrong selector or selector, which is, not exist.

Example:- driver.findElement(By.id("invalidid")).sendKeys("Mukesh");

* ElementNotVisibleException

This Exception occurs when the element presence is in DOM, it is not visible.

Example:- Hidden Elements, which has presence in DOM and it, is not visible. Visibility means the height and width should be greater than zero. Hidden Elements are defined in HTML using of type=”hidden”.

driver.findElement(By.id("hiddenid")).sendKeys("Mukesh");

driver.findElement(By.id("hiddenid")).sendKeys("Mukesh");

* NoSuchFrameException

This Exception occurs when the driver is switching to an invalid frame, which is not available.

Example:-

driver.switchTo().frame(invalidindex);

(or)

driver.switchTo().frame("frame\_z");

driver.switchTo().frame(invalidindex);

(or)

driver.switchTo().frame("frame\_z");

//frame\_z is the name of the invalid frame

For frames indexing starts from Zero. Try to access the frame by providing invalid index.

* NoAlertPresentException

This Exception occurs when the driver is switching to an invalid Alert, which is not available.

Example:-

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

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

//Execute this command on browser without invoking the alert.

* NoSuchWindowException

This Exception occurs when the driver is switching to an invalid Window, which is not available.

Example:-

driver.switchTo().window("invalidwindowname");

driver.switchTo().window("invalidwindowname");

* WebDriverException

This Exception occurs when the driver is performing the action after immediately closing the browser.

Example:-

driver.close();

driver.findElement(By.id("username")).sendKeys("Mukesh");

driver.close();

driver.findElement(By.id("username")).sendKeys("Mukesh");

* SessionNotFoundException

This Exception occurs when the driver is performing the action after immediately quitting the browser.

Example:-

driver.quit();

driver.findElement(By.id("username")).sendKeys("Mukesh");

driver.quit();

driver.findElement(By.id("username")).sendKeys("Mukesh");

* StaleElementReferenceException

This Exception occurs when the Element belongs to a different frame than the current one. The user has navigated away to another page.

Example:-

WebElement element=driver.findElement(By.id("username"));// Element is available in parent window

driver.switchTo().window(Child\_Window);//Switch to Child Window

element.sendKeys("Mukesh");//perform the action on the element which is not visible in the child window

**67. how to handle Ajax calls in selenium?**

* AJAX stands for Asynchronous JavaScript & XML, and it allows the Web page to retrieve small amounts of data from the server without reloading the entire page. The biggest challenge in handling Ajax call is knowing the loading time for the web page. Since the loading of the web page will last only for a fraction of seconds, it is difficult for the tester to test such application through automation tool. For that, Selenium Webdriver has to use the wait method on this Ajax Call.

Thread.Sleep()

Implicit Wait()

Explicit Wait()

WebdriverWait

Fluent Wait

**75. How to assign the value to textbox other than sendkeys method?**

* I have a long string to test and sendKeys() takes too long. If I try to set the value the program crashes. I know the Selenium sendKeys() is the best way to test the actual user input, but for my application it takes too much time.

var webdriver = require('selenium-webdriver');

var driver = new webdriver.Builder().

withCapabilities(webdriver.Capabilities.chrome()).

build();

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

// find the search input field on google.com

inputField = driver.findElement(webdriver.By.name('q'));

var longstring = "test"; // not really long for the sake of this quick example

// this works but is slow

inputField.sendKeys(longstring);

// no error but no values set

inputField.value = longstring;

// Output: TypeError: Object [object Object] has no method 'setAttributes'

inputField.setAttributes("value", longstring);

**79. Write code for drag/drop in selenium?**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

public class DemoDragDrop {

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

// Initiate browser

WebDriver driver=new FirefoxDriver();

// maximize browser

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

// Open webpage

driver.get("http://jqueryui.com/resources/demos/droppable/default.html");

// Add 5 seconds wait

Thread.sleep(5000);

// Create object of actions class

Actions act=new Actions(driver);

// find element which we need to drag

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

// find element which we need to drop

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

// this will drag element to destination

act.dragAndDrop(drag, drop).build().perform();

}

}</span>

**81. Write code for right click in seelnium?**

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.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

public class Right\_click {

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

// Initialize WebDriver

WebDriver driver = new FirefoxDriver();

// Wait For Page To Load

driver.manage().timeouts().implicitlyWait(120,TimeUnit.SECONDS);

// Go to Myntra Page

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

// Maximize Window

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

WebElement R1 = driver.findElement(By.xpath("//a[@href='/shop/men?src=tn&nav\_id=5']"));

// Initialize Actions class object

Actions builder = new Actions(driver);

// Perform Right Click on MEN and Open "Men" content in a new tab

builder.contextClick(R1).sendKeys(Keys.ARROW\_DOWN).sendKeys(Keys.ENTER).perform();

**83. Write code fro scroll to specific element?**

This will scroll until the element is in view :

WebElement element = driver.findElement(By.id("id\_of\_element"));

((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);", element);

Thread.sleep(500);