--limitations of Selenium:

Some limitations of Selenium Automation tool are as follows:

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

Its 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 in order 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.

--installing/configure selenium?

Step 1 - Install Java on your computer

Download and install the Java Software Development Kit (JDK)

This JDK version comes bundled with Java Runtime Environment (JRE), so you do not need to download and install the JRE separately.

Step 2 - Install Eclipse IDE

Download "Eclipse IDE for Java Developers". Be sure to choose correctly between Windows 32 Bit and 64 Bit versions.

You should be able to download an exe file named "eclipse-inst-win64"

Double-click on file to Install the Eclipse. A new window will open. Click Eclipse IDE for Java Developers.After that, a new window will open which click button marked 1 and change path to "C:\eclipse". Post that Click on Install button marked

After successful completion of the installation procedure, a window will appear. On that window click on Launch

This will start eclipse neon IDE for you.

Step 3 - Download the Selenium Java Client Driver

You can download the Selenium Java Client Driver [here](http://seleniumhq.org/download/). You will find client drivers for other languages there, but only choose the one for Java.This download comes as a ZIP file named "selenium-2.25.0.zip". For simplicity, extract the contents of this ZIP file on your C drive so that you would have the directory "C:\selenium-2.25.0\". This directory contains all the JAR files that we would later import on Eclipse.

Step 4 - Configure Eclipse IDE with WebDriver

1. Launch the "eclipse.exe" file inside the "eclipse" folder that we extracted in step 2. If you followed step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe.
2. When asked to select for a workspace, just accept the default location.

3. Create a new project through File > New > Java Project. Name the project as "newproject".

A new pop-up window will open enter details as follow

1. Project Name
2. Location to save project
3. Select an execution JRE
4. Select layout project option
5. Click on Finish button

4. In this step,

1. Right-click on the newly created project and
2. Select New > Package, and name that package as "newpackage".

A pop-up window will open to name the package,

1. Enter the name of the package
2. Click on Finish button
3. Create a new Java class under newpackage by right-clicking on it and then selecting- New > Class, and then name it as "MyClass".
4. When you click on Class, a pop-up window will open, enter details as
5. Name of the class
6. Click on Finish button

This is how it looks like after creating class.

Now selenium WebDriver's into Java Build Path

In this step,

1. Right-click on "newproject" and select Properties.
2. On the Properties dialog, click on "Java Build Path".
3. Click on the Libraries tab, and then
4. Click on "Add External JARs.."

When you click on "Add External JARs.." It will open a pop-up window. Select the JAR files you want to add.

After selecting jar files, click on OK button.

6. Add all the JAR files inside and outside the "libs" folder. Your Properties dialog should now look similar to the image below.

7. Finally, click OK and we are done importing Selenium libraries into our project.

--what are different ways of locating elements in selenium

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

* Firebug for Firefox
* Google Developer Tools for Chrome
* Web Inspector for Safari

Selenium WebDriver API supports different possibilities to identify elements: by ID, by CLASS, by NAME, by CSS selector, by XPath, by TAG name. Also you define your custom selector in order to interact with the elements.

It’s always a good practice to assign unique IDs to elements, also names and classes in order to be more usable for automatic UI tests. If that is not possible you’ll need to use advanced or XPath selector to interact with those elements. The most popular selectors are the CSS selectors due to performance and simplicity reasons.

To inspect an element you just have to open the desired web page, right-click the desired element and click on Inspect Element. A new panel will open showing the desired element. Also you can inspect other elements by clicking on the cursor in the top left side of the Developer Tools or Firebug panels and hovering page elements.

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

1. By ID:

|  |  |
| --- | --- |
| 1 | in Java: driver.findElement(By.id("element id")) |

2. By CLASS:

|  |  |
| --- | --- |
| 1 | in Java: driver.findElement(By.className("element class")) |

3. By NAME:

|  |  |
| --- | --- |
| 1 | in Java: driver.findElement(By.name("element name")) |

4. By TAGNAME:

|  |  |
| --- | --- |
| 1 | in Java: driver.findElement(By.tagName("element html tag name")) |

5. By CSS Selector:

|  |  |
| --- | --- |
| 1 | in Java: driver.findElement(By.cssSelector("css selector")) |

6. By Link:

|  |  |
| --- | --- |
| 1 | in Java: driver.findElement(By.link("link text")) |

7. By XPath:

|  |  |
| --- | --- |
| 1 | in Java: driver.findElement(By.xpath("xpath expression")) |

--which is fastest way to identify elements in web page?

Finding elements by ID is usually going to be the fastest option, because at its root, it eventually calls down to document.getElementById(), which is optimized by many browsers.

Finding elements by XPath is useful for finding elements using very complex selectors, and is the most flexible selection strategy, but it has the potential to be very slow, particularly in IE. In IE 6, 7, or 8, finding by XPath can be an order of magnitude slower than doing the same in Firefox. IE provides no native XPath-over-HTML solution, so the project must use a JavaScript XPath implementation, and the JavaScript engine in legacy versions of IE really is that much slower.

If you have a need to find an element using a complex selector, I usually recommend using CSS Selectors, if possible. It's not quite as flexible as XPath, but will cover many of the same cases, without exhibiting the extreme performance penalty on IE that XPath can.

-- what is absolute path and relative path in xpath

Location path specifies the location of node in XML document. This path can be absolute or relative. If location path starts with root node or with '/' then it is an absolute path. Following are few of the example locating the elements using absolute path.

/class/student − select student nodes within class root node.

<xsl:for-each select = "/class/student">

/class/student/firstname − select firstname of a student node within class root node.

<p><xsl:value-of select = "/class/student/firstname"/></p>

## Example

In this example, we've created a sample XML document students.xml and its stylesheet document students.xsl which uses the XPath expressions.

Following is the sample XML used.

### students.xml

<?xml version = "1.0"?>

<?xml-stylesheet type = "text/xsl" href = "students.xsl"?>

<class>

<student rollno = "393">

<firstname>Dinkar</firstname>

<lastname>Kad</lastname>

<nickname>Dinkar</nickname>

<marks>85</marks>

</student>

<student rollno = "493">

<firstname>Vaneet</firstname>

<lastname>Gupta</lastname>

<nickname>Vinni</nickname>

<marks>95</marks>

</student>

<student rollno = "593">

<firstname>Jasvir</firstname>

<lastname>Singh</lastname>

<nickname>Jazz</nickname>

<marks>90</marks>

</student>

</class>

students.xsl

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

<xsl:stylesheet version = "1.0"

xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">

<xsl:template match = "/" >

<html>

<body>

<h3>Details of each Students. </h3>

<table border = "1">

<tr bgcolor = "#9acd32">

<th>Roll No</th>

<th>First Name</th>

<th>Last Name</th>

<th>Nick Name</th>

<th>Marks</th>

</tr>

<tr>

<td><xsl:value-of select = "/class/student[1]/@rollno"/></td>

<td><xsl:value-of select = "/class/student[1]/firstname"/></td>

<td><xsl:value-of select = "/class/student[1]/lastname"/></td>

<td><xsl:value-of select = "/class/student[1]/nickname"/></td>

<td><xsl:value-of select = "/class/student[1]/marks"/></td>

</tr>

<tr>

<td>

<xsl:value-of select = "/class/student/@rollno"/>

</td>

<td><xsl:value-of select = "/class/student[2]/firstname"/></td>

<td><xsl:value-of select = "/class/student[2]/lastname"/></td>

<td><xsl:value-of select = "/class/student[2]/nickname"/></td>

<td><xsl:value-of select = "/class/student[2]/marks"/></td>

</tr>

<tr>

<td>

<xsl:value-of select = "/class/student[3]/@rollno"/>

</td>

<td><xsl:value-of select = "/class/student[3]/firstname"/></td>

<td><xsl:value-of select = "/class/student[3]/lastname"/></td>

<td><xsl:value-of select = "/class/student[3]/nickname"/></td>

<td><xsl:value-of select = "/class/student[3]/marks"/></td>

</tr>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

-- different types of waits or synchronization in selenium webdriver

Generally in Test Automation, we have two components  
1. Application Under Test  
2. Test Automation Tool.

Both these components will have their own speed. We should write our scripts in such a way that both the components should move with same and desired speed, so that we will not encounter "Element Not Found" errors which will consume time again in debugging.

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();*](https://docs.oracle.com/javase/tutorial/essential/concurrency/sleep.html)

The main disadvantage for the above statements are, there is a chance of unnecessary waiting time even though the application is ready.

The advantages are like in a situation where we interact for third party systems like interfaces, it is not possible to write a condition or check for a condition. Here in this situations, we have to make the application to wait for certain amount of time by specifying the timeout value.

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 particular condition to satisfy.

1. Implicit Wait.

An implicit wait is to tell WebDriver to poll the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available.

The default setting is 0. Once when we define the implicit wait, it will set for the life of the WebDriver object instance.

It is a mechanism which will be written once and applied for entire session automatically. It should be applied immediately once we initiate the Webdriver.

Implicit wait will not work all the commands/statements in the application. It will work only for "FindElement" and "FindElements" statements.

If we set implicit wait, find element will not throw an exception if the element is not found in first instance, instead it will poll for the element until the timeout and then proceeds further. We should always remember to add the below syntax immediately below the Webdriver statement.

Syntax:

driver.manage.TimeOuts.implicitwait(6,Timeunit.SECONDS);

Example using implicit timeout

WebDriver driver = new FirefoxDriver();

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

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

Explicit Wait:

We need to define a wait statement for certain condition to be satisfied until the specified timeout period. If the Webdriver finds the element within the timeout period the code will get executed.

Explicit wait is mostly used when we need to Wait for a specific content/attribute change after performing any action, like when application gives AJAX call to system and get dynamic data and render on UI.

Example: Like there are drop-downs Country and State, based on the country value selected, the values in the state drop-down will change, which will take few seconds of time to get the data based on user selection.

    WebDriverWait wait = new WebDriverWait(driver, 10);

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

The above statement waits up to 10 seconds before throwing Exception (TimeoutException - Timed out after 10 seconds waiting for visibility of element) or if it finds the element, it will return in 0 - 10 seconds.

There are different waits that can be used based on the needs which we frequently come across when automating web applications.

Fluent Wait:

Using FluentWait we can define the maximum amount of time to wait for a condition, as well as the frequency with which to check for the condition.

And also the user can configure to ignore specific types of exceptions such as ["NoSuchElementExceptions"](http://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/NoSuchElementException.html) when searching for an element. NoSuchElement exception is thrown by findElement(By) and findElements(By). When ever it try to find any element it returns the first matching element on the current page else it throws NoSuchElementException - when no matching elements are found.

Syntax:

Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)

//Wait for the condition

       .withTimeout(30, TimeUnit.SECONDS)

         // which to check for the condition with interval of 5 seconds.

       .pollingEvery(5, TimeUnit.SECONDS)

     //Which will ignore the NoSuchElementException

       .ignoring(NoSuchElementException.class);

-- how to save screen shots using selenium webdriver

import java.io.File;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;

  public class takeScreenShotExample{

public WebDriver driver;

public void openBrowser() throws Exception {

driver = new FirefoxDriver();

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

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

  try{

driver.findElement(By.id("testing")).sendKeys("test");                                 }

catch (Exception e){

System.out.println("I'm in exception");

getscreenshot();

    }

}

  public void getscreenshot() throws Exception       {

     File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE FileUtils.copyFile(scrFile, new File("D:\\screenshot.png"));

}

}

-- how to handle multiple windows in selenium webdriver

public class MultiWindowHandle {

WebDriver driver;

public void setup() throws Exception {

driver=new FirefoxDriver();

String URL="https://www.abc.co.in/";

driver.get(URL);

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

}

public void test() throws Exception {

// Opening Calender

driver.findElement(By.xpath("//img[@alt='Calender']")).click();

// Storing parent window reference into a String Variable

String Parent\_Window = driver.getWindowHandle();

  // Switching from parent window to child window

for (String Child\_Window : driver.getWindowHandles())

{

driver.switchTo().window(Child\_Window);

// Performing actions on child window

driver.findElement(By.id("calendar\_month\_txt")).click();

List  Months=driver.findElements(By.xpath("//div[@id='monthDropDown']//div"));

int Months\_Size=Months.size();

System.out.println("Month size is:"+Months\_Size);

Months.get(1).click();

driver.findElement(By.xpath("//\*[@id='calendarDiv']/div/table/tbody/tr/td[contains(text(),'16')]")).click();

}

//Switching back to Parent Window

driver.switchTo().window(Parent\_Window);

//Performing some actions on Parent Window

driver.findElement(By.className("btn\_style")).click();

}

  public void close() {

  driver.quit();

  }

}

-- how to lanuch webpage using chrome driver

import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
public class Chrome {  
WebDriver driver;  
public void launchChrome()  
{  
System.setProperty("webdriver.chrome.driver", "E:\\DD MISHRA\\workspace\\chromedriver\_win\_26.0.1383.0\\chromedriver.exe");  
driver = new ChromeDriver();  
}

public void testChrome()  
{  
driver.get("http://www.google.co.in");  
driver.findElement(By.id("gbqfq")).sendKeys("Selenium");  
}  
public void kill()  
{  
driver.close();  
driver.quit();  
}  
}

-- what is desired capabilities in selenium webdriver

importorg.openqa.selenium.WebDriver;

importorg.openqa.selenium.ie.InternetExplorerDriver;

importorg.openqa.selenium.remote.DesiredCapabilities;

public class IEtestforDesiredCapabilities {

public static void main(String[] args) {

DesiredCapabilities capabilities = DesiredCapabilities.internetExplorer();

capabilities.setCapability(CapabilityType.BROWSER\_NAME, "IE");

capabilities.setCapability(InternetExplorerDriver.

INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS,true);

System.setProperty("webdriver.ie.driver", "C:\\IEDriverServer.exe");

WebDriver driver = newInternetExplorerDriver(capabilities);

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

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

driver.quit();

}

}

-- how to set language while opening website

Using Firefox Browser :

FirefoxProfile profile = new FirefoxProfile();  
//setting the locale french : ‘fr’  
profile.setPreference(“intl.accept\_languages”,”fr”);  
driver = new FirefoxDriver(profile);  
driver.get(“[http://google.co.in&#8221](#8221););

Using Chrome Browser :

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](#8221););

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

-- write code to use textbox, button click events

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 Facebook {

public static void main(String args[]){

WebDriver driver = new FirefoxDriver();

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

WebElement email= driver.findElement(By.id("email"));

Actions builder = new Actions(driver);

Actions seriesOfActions = builder.moveToElement(email).click().sendKeys(email, "gati.naveen@gmail.com");

seriesOfActions.perform();

WebElement pass = driver.findElement(By.id("pass"));

WebElement login =driver.findElement(By.id("u\_0\_b"));

Actions seriesOfAction = builder.moveToElement(pass).click().sendKeys(pass, "naveench").click(login);

seriesOfAction.perform();

driver.

}

}

-- how to select items from dropdown/select box

package automationFramework;

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;

import org.openqa.selenium.support.ui.Select;

public class DropDownCommands {

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

WebDriver driver = new FirefoxDriver();

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

driver.get("http://toolsqa.wpengine.com/automation-practice-form");

Select oSelect = new Select(driver.findElement(By.id("continents")));

oSelect.selectByVisibleText("Europe");

Thread.sleep(2000);

oSelect.selectByIndex(2);

Thread.sleep(2000);

List<WebElement> oSize = oSelect.getOptions();

int iListSize = oSize.size();

for(int i =0; i < iListSize ; i++){

String sValue = oSelect.getOptions().get(i).getText();

System.out.println(sValue);

if(sValue.equals("Africa")){

oSelect.selectByIndex(i);

break;

}

}

driver.quit();

}

}

-- how to know if checkbox is checked or not in webpage

WebDriver driver = new InternetExplorerDriver();  
WebDriverBackedSelenium rc = new WebDriverBackedSelenium(driver, "");  
driver.get("[http://www.thejekels.com/dojo/demo/checkboxtree/](http://www.google.com/url?q=http%3A%2F%2Fwww.thejekels.com%2Fdojo%2Fdemo%2Fcheckboxtree%2F&sa=D&sntz=1&usg=AFQjCNFg3O4m8F9ZpfaRGboJ48iI1XQCQw)");  
if (driver.getWindowHandles().size()==0){  
throw new RuntimeException("Hello!");  
}  
String xpath = "//INPUT[@type='checkbox' and @id='cbt\_CheckBox\_1']";  
new WebDriverWait(driver, 5000).until(ExpectedConditions.presenceOfElementLocated(By.xpath(xpath)));  
WebElement checkbox = driver.findElement(By.xpath(xpath));   
System.out.println("============Before click");  
System.out.println("isSelected via webdriver: "+checkbox.isSelected());  
System.out.println("isSelected via rc: "+rc.isChecked(xpath));  
System.out.println("isSelected via value attribute: "+checkbox.getAttribute("value"));  
checkbox.click();   
System.out.println("============After click");  
System.out.println("isSelected via webdriver: "+checkbox.isSelected());  
System.out.println("isSelected via rc: "+rc.isChecked(xpath));  
System.out.println("isSelected via value attribute: "+checkbox.getAttribute("value"));

-- tell me code to pass values from parent window to child window

Parent Window:

Our parent window contains a simple TextBox and a LinkButton. The TextBox will display the data which is sent from the child window and the LinkButton will simply opens the child window also known as the popup window. Below is the simple code that is used to attach the onclick attribute to the LinkButton.

protected void Page\_Load(object sender, EventArgs e)

{

string openWindow = @"window.open('Child.aspx')";

this.LinkButton1.Attributes.Add("onclick", openWindow);

}

Now let's see the child window. In the code below we attach the onclick event to the LinkButton which is on the Child page.

Child Window:

protected void Page\_Load(object sender, EventArgs e)

{

this.LinkButton1.Attributes.Add("onclick", "PassValues()");

}

Child Page also contains a TextBox. Whatever you type in the TextBox will be transferred to the parent page. Now all we need is JavaScript so that we can pass the information to the parent page.

function PassValues()

{

var txtValue = document.getElementById("TextBox1").value;

// Now pass the value to the Parent form

window.opener.form1.txtUserName.value = txtValue;

window.close();

}

This is it! Now when you type something in the TextBox of the child window and press the LinkButton that value is passed to the parent window and displayed in the TextBox of the parent window. After that the child window is automatically closed.

-- write code to find out if all links are working or not

import [org.openqa.selenium.By](http://org.openqa.selenium.by/);

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class FindBrokenLinks {

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

  WebDriver driver = new FirefoxDriver();

  driver.get("[https://www.xyz.com/](https://www.google.com/url?q=https%3A%2F%2Fwww.xyz.com%2F&sa=D&sntz=1&usg=AFQjCNG02baXRkxW4WuokCXBFj8cBmZugw)");

  Thread.sleep(5000L);

  int size = driver.findElements(By.tagName("a")).size();

  System.out.println(size);

  List<String> Linkarray = new ArrayList<String>();

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

      for (WebElement link : Linklist) {

       String links = link.getText();

       Linkarray.add(links );

      }

      for (String linkToTest : Linkarray){

       driver.findElement(By.linkText(linkToTest)).click();

       Thread.sleep(15000L);

    if(driver.getTitle().contains("Problem")) {

     System.out.println("Fail");

    }

    else

    {

     System.out.println("pass");

    }

    driver.navigate().back();

    Thread.sleep(5000L);

   }

  driver.close();

}

}

-- write code on how to use javascriptexecutor?

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;

public class JavaSE\_Test {

{

WebDriver driver= new FirefoxDriver();

JavascriptExecutor js = (JavascriptExecutor)driver;

Site. driver.get("[http://demo.guru99.com/V4/](http://www.google.com/url?q=http%3A%2F%2Fdemo.guru99.com%2FV4%2F&sa=D&sntz=1&usg=AFQjCNEJcoeaavmMJfDgPKK5fjwCWGsmuw)");

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

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

start\_time = System.currentTimeMillis();

executeAsyncScript() method to wait for 5 seconds js.executeAsyncScript("window.setTimeout(arguments[arguments.length - 1], 5000);"); System.out.println("Passed time: " + (System.currentTimeMillis() - start\_time));

}

}

up one javascript

diff b/w assert and verify

package stmTutorial;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.testng.Assert;  
import org.testng.annotations.Test;  
  
public class AssertionExample {  
  
    public void assertion(){  
        //Instantiation of driver object. To launch Firefox browser  
        WebDriver driver = new FirefoxDriver();  
  
        driver.get("[http://www.softwaretestingmaterial.com](http://www.google.com/url?q=http%3A%2F%2Fwww.softwaretestingmaterial.com&sa=D&sntz=1&usg=AFQjCNEOeqPQ8ldYCekTftAHOPBpNkwyeg)");  
              String Title = "Software Testing Material";  
        String GetTitle = driver.getTitle();  
        System.out.println("Assertion starts here...");  
        Assert.assertEquals(Title, GetTitle);  
        System.out.println("A blog for Software Testers");  
                driver.quit();          
    }  
}

--dif b/w driver and close

public class BrowserClose()

{

public static String baseUrl;

public void setup(){

driver = new FirefoxDriver();

baseUrl="[http://www.wikishown.com](http://www.google.com/url?q=http%3A%2F%2Fwww.wikishown.com&sa=D&sntz=1&usg=AFQjCNGrxVdqqZU37URV6sEsIbyibjQKjA)";

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

}

public void wikiShown(){

driver.manage().timeouts().implecitlyWait(30,TimeUnit.SSECONDS);

}

public void tearDown()

{

driver.quit();

}

}

--ajax cal using selinum

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.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.Assert;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class Ajaxdemo { private String URL = "http://google.com /selenium/ajax.html"; WebDriver driver;

WebDriverWait wait;

public void setUp()

{

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

driver = new ChromeDriver(); driver.manage().window().maximize(); driver.navigate().to(URL);

}

public void test\_AjaxExample() {

By container = By.cssSelector(".container");

wait = new WebDriverWait(driver, 5); wait.until(ExpectedConditions.presenceOfElementLocated(container));

WebElement noTextElement = driver.findElement(By.className("radiobutton"));

String textBefore = noTextElement.getText().trim();

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

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

TextElement = driver.findElement(By.className("radiobutton")); wait.until(ExpectedConditions.visibilityOf(TextElement));

String textAfter = TextElement.getText().trim();

Assert.assertNotEquals(textBefore, textAfter);

System.out.println("Ajax Call Performed");

String expectedText = "Radio button is checked and it's value is Yes";

Assert.assertEquals(textAfter, expectedText); driver.close();

}

}

--Write code for right click in selenium

package com.pack.rightclick;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.NoSuchElementException;

import org.openqa.selenium.StaleElementReferenceException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.Assert; import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class RightClickExample { WebDriver driver;

public void Setup() { driver = new FirefoxDriver();

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

public void rightClickTest()

{

driver.navigate().to(URL);

By locator = By.cssSelector(".context-menu-one.box");

WebDriverWait wait = new WebDriverWait(driver, 5);

WebElement element=driver.findElement(locator);

rightClick(element);

WebElement elementEdit =driver.findElement(By.cssSelector(".context-menu-item.icon.icon-edit>span")); elementEdit.click();

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

String textEdit = alert.getText();

Assert.assertEquals(textEdit, "clicked: edit", "Failed to click on Edit link");

}

public void rightClick(WebElement element)

{

{

Actions action = new Actions(driver).contextClick(element);

action.build().perform();

System.out.println("Sucessfully Right clicked on the element");

} catch (StaleElementReferenceException e) {

System.out.println("Element is not attached to the page document " + e.getStackTrace());

} catch (NoSuchElementException e) { System.out.println("Element " + element + " was not found in DOM " + e.getStackTrace());

} catch (Exception e)

}

}

-- Write code for drag/drop in selenium

   
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;  
   
import org.openqa.selenium.interactions.Action;  
   
import org.openqa.selenium.interactions.Actions;  
   
public class DragAndDrop {  
   
public static void main(String[] args) throws InterruptedException {  
   
  WebDriver driver = new FirefoxDriver();  
   
  String URL = "[http://sandbox.checklist.com/account/](http://www.google.com/url?q=http%3A%2F%2Fsandbox.checklist.com%2Faccount%2F&sa=D&sntz=1&usg=AFQjCNGjYNEE_7XDf1xAru0lYS4R6BPJRQ)";  
   
  driver.get(URL);  
   
  driver.findElement(By.name("j\_username")).sendKeys("Username");  
   
  driver.findElement(By.name("j\_password")).sendKeys("Password);  
   
  driver.findElement(By.name("login")).click();  
   
  driver.manage().window().maximize();  
   
  driver.manage().timeouts().implicitlyWait(10000, TimeUnit.MILLISECONDS);  
   
  WebElement From = driver.findElement(By.xpath(".//\*[@id='userChecklists']/li[1]/a/span"));  
   
  WebElement To = driver.findElement(By.xpath(".//\*[@id='userChecklists']/li[4]/a/span"));  
   
  Actions builder = new Actions(driver);  
   
      Action dragAndDrop = builder.clickAndHold(From)  
   
     .moveToElement(To)  
   
    .release(To)  
   
   .build();  
   
  dragAndDrop.perform();  
   
}  
   
}

\* package javaj;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class app {

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

// TODO Auto-generated method stub

System.setProperty("webdriver.gecko.driver","C:\\Users\\Leela Rathod\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

driver.get("https://npuosc.npu.edu");

//Thread.sleep(5000);

driver.findElement(By.cssSelector(".tile-content>img")).click();

Thread.sleep(5000);

//driver.findElement(By.linkText("#main>fieldset"));

driver.findElement(By.cssSelector("#UserName")).sendKeys("15276rs");

driver.findElement(By.cssSelector("#Password")).sendKeys("slrathod");

//driver.switchTo().defaultContent(driver.findElement("By.cssSelector("[id='UserName']")).sendKeys("15276rs");

driver.findElement(By.cssSelector(".left > form:nth-child(4) > p:nth-child(8) > input:nth-child(3)")).click();

}

}

\* import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class auto1 {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.gecko.driver","C:\\Users\\Leela Rathod\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

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

driver.findElement(By.id("email")).sendKeys("leelarathod5@gmail.com");

driver.findElement(By.id("pass")).sendKeys("leelavenky@143");

driver.findElement(By.xpath(".//\*[@id='u\_0\_n']")).click();

\* package selenium;

public class basic {

public static void main(String args[]){

int a,b;

a=3;

b=3;

int x=a+b;

System.out.println(x);

}

}

\* package selenium;

import org.openqa.selenium.firefox.FirefoxProfile;

public class BrowserInvocation {

public static void main(String args[]){

FirefoxProfile =new FirefoxProfile();

}

}

\* package selenium;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class browserinvoking {

public static void main(String args[]){

System.setProperty("webdriver.gecko.driver", "C:\\Users\\Leela Rathod\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

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

}

}

\*Button

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class button {

public static void main(String args[]){

System.setProperty("webdriver.gecko.driver", "C:\\Users\\Leela Rathod\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

driver.get("http://www.echoecho.com/htmlforms10.htm");

driver.findElement(By.xpath("//input[@value='Milk']")).click();

System.out.println(driver.findElement(By.xpath(".//input[@type='radio']")).getSize());

}

}

\*css

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class css {

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

// TODO Auto-generated method stub

System.setProperty("webdriver.gecko.driver","C:\\Users\\Leela Rathod\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

driver.get("https://npuosc.npu.edu");

//Thread.sleep(5000);

driver.findElement(By.cssSelector(".tile-content>img")).click();

Thread.sleep(5000);

//driver.findElement(By.linkText("#main>fieldset"));

driver.findElement(By.cssSelector("#UserName")).sendKeys("15276rs");

driver.findElement(By.cssSelector("#Password")).sendKeys("vkolluru1");

//driver.switchTo().defaultContent(driver.findElement("By.cssSelector("[id='UserName']")).sendKeys("15276rs");

driver.findElement(By.cssSelector(".left > form:nth-child(4) > p:nth-child(8) > input:nth-child(3)")).click();

}

\* import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Dropdown {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.gecko.driver","C:\\Users\\Leela Rathod\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

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

//driver.findElement(By.xpath(".//\*[@class='signupclose icon ic-cancel-fill']")).click();

driver.findElement(By.cssSelector(".signupClose")).click();

driver.findElement(By.xpath(".//input[@id='ember707']")).click();

driver.findElement(By.xpath(".//input[@id='ember707']")).sendKeys("sjc");

driver.findElement(By.xpath(".//input[@id='ember718']")).click();

driver.findElement(By.xpath(".//input[@id='ember718']")).sendKeys("sfo");

}

}

\*B2

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.Select;

public class Dropdown2 {

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

System.setProperty("webdriver.gecko.driver","C:\\Users\\Leela Rathod\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

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

//Thread.sleep(5000);

driver.findElement(By.cssSelector("#ctl00\_mainContent\_ddl\_originStation1\_CTXTaction")).click();

driver.findElement(By.cssSelector("#dropdownGroup1 > div:nth-child(2) > ul:nth-child(2) > li:nth-child(7) > a:nth-child(1)")).click();

driver.findElement(By.cssSelector("#ctl00\_mainContent\_ddl\_destinationStation1\_CTXTaction")).click();

driver.findElement(By.cssSelector("#glsctl00\_mainContent\_ddl\_destinationStation1\_CTNR > table:nth-child(2) > tbody:nth-child(1) > tr:nth-child(2) > td:nth-child(2) > div:nth-child(3) > div:nth-child(1) > div:nth-child(2) > ul:nth-child(2) > li:nth-child(1) > a:nth-child(1)")).click();

/\*Select dropdown=new Select(driver.findElement(By.cssSelector("#ctl00\_mainContent\_ddl\_Adult")));

dropdown.selectByIndex(2);

dropdown.selectByValue("6");

dropdown.selectByVisibleText("3adults");

//Thread.sleep(5000);\*/

/\*driver.findElement(By.cssSelector("#ctl00\_mainContent\_ddl\_Child")).click();

Select child=new Select(driver.findElement(By.cssSelector("#ctl00\_hmainContent\_ddl\_Child")));

child.selectByIndex(1);

child.selectByValue("1");

child.selectByVisibleText("1Child");

Thread.sleep(5000);

//Select dropdown2=new Select(driver.findElement(By.cssSelector("#ctl00\_mainContent\_ddl\_Infant")));

driver.findElement(By.cssSelector("driver.findElement(By.cssSelector")).click();

Select infant=new Select(driver.findElement(By.cssSelector("#ctl00\_mainContent\_ddl\_Infant")));

infant.selectByIndex(2);

infant.selectByValue("2");

infant.selectByVisibleText("2Infants");

Thread.sleep(5000); \*/

driver.findElement(By.cssSelector("#ctl00\_mainContent\_DropDownListCurrency")).click();

Select currency=new Select(driver.findElement(By.cssSelector("#ctl00\_mainContent\_DropDownListCurrency")));

currency.selectByIndex(1);

System.out.println(driver.findElement(By.cssSelector("#ctl00\_mainContent\_chk\_IndArm")).isSelected());

driver.findElement(By.cssSelector("#ctl00\_mainContent\_chk\_IndArm")).click();

System.out.println(driver.findElement(By.cssSelector("#ctl00\_mainContent\_chk\_IndArm")).isSelected());

driver.findElement(By.cssSelector("#ctl00\_mainContent\_chk\_IndArm")).click();

System.out.println(driver.findElement(By.cssSelector("#ctl00\_mainContent\_chk\_IndArm")).isSelected());

}

}

\*gmail

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class gmail {

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

// TODO Auto-generated method stub

System.setProperty("webdriver.gecko.driver","C:\\Users\\Leela Rathod\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

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

driver.findElement(By.xpath(".//\*[@id='Email']")).sendKeys("leelarathod6@gmail.com");

driver.findElement(By.xpath(".//\*[@id='next']")).click();

Thread.sleep(5000);

//new WebDriverWait(driver,30).until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[@id='password-shown']/div[1]/input[@id='Passwd']"))).sendKeys("vkolluru1");

driver.findElement(By.xpath(".//\*[@id='Passwd']")).sendKeys("vkolluru1");

driver.findElement(By.xpath(".//\*[@id='signIn']")).click();

}

}

\* scroll

package javaj;

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 mousemovement {

public static void main(String args[]){

System.setProperty("webdriver.gecko.driver", "C:\\Users\\leela\\Desktop\\Selenium Driver Files\\geckodriver-v0.13.0-win64\\geckodriver.exe");

WebDriver driver=new FirefoxDriver();

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

Actions abc=new Actions(driver);

WebElement element=driver.findElement(By.xpath(".//\*[@id='nav-link-accountList']"));

abc.moveToElement(element).build().perform();

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

abc.keyDown(keys.SHIFT)=moveToElement(xyz).sendkeys("smallletter").build().perform();

abc.contextClick(xyz).build().perform();

}

}