**Selenium**

**What is Selenium?**

Selenium is an open-source tool that is used for test automation. It is licensed under Apache License 2.0. Selenium is a suite of tools that helps in automating only web applications.

\*validate the application

2 types,

1. Manual Testing

2. Automation Testing

**1.Manual:**

\* using excel sheet we test the java application.

\* it takes more time to test much applications. so we go for automation

**2.Atomation Testing:**

**Tools:**

1. QTP--Paid

2. Selenium-open source

Selenium Modules:

selenium IDE

selenium RC

Web driver

selenium GRID

Appium(Mobile testing)

Now we used web driver

**Web driver:**

\* it is Interface

**1.Overview of selenium:**

\* Browser configuration(IE,Chrome, firefox, safari,opera)

**2.Web Driver Commands:**

\* Browser commands

\* Browser navigation commands

\* web element commands

\* Find element commands

\* Check box & radio buttons

\* Dropdown and multiple select

\* WebTable Handling

3. Locators and Xpath:

4. Switches alert and window popup

**Selenium Versions?**

Selenium 3.13.0

Selenium 3.11.0

Selenium 3.5.0

**What is Selenium IDE?**

Selenium Integrated Development Environment (IDE) is a Firefox plugin that lets testers to record their actions as they follow the workflow that they need to test.

**What is Selenium RC?**

Selenium Remote Control (RC) was the flagship testing framework that allowed more than simple browser actions and linear execution. It makes use of the full power of programming languages such as Java, C#, PHP, Python, Ruby and PERL to create more complex tests.

**What is Selenium WebDriver?**

Selenium WebDriver is the successor to Selenium RC which sends commands directly to the browser and retrieves results.

**What is Selenium Grid?**

Selenium Grid is a tool used to run parallel tests across different machines and different browsers simultaneously which results in minimized execution time.

**Explain Browser ,Driver capability?**

**Chrome:**

[ChromeDriver 2.40](https://chromedriver.storage.googleapis.com/index.html?path=2.40/) 🡪Supports Chrome v66-68

ChromeDriver 2.39 🡪Supports Chrome v66-68

ChromeDriver 2.38 🡪Supports Chrome v65-67

ChromeDriver 2.37 🡪Supports Chrome v64-66

ChromeDriver 2.36 🡪Supports Chrome v63-65

ChromeDriver 2.35 🡪Supports Chrome v62-64

ChromeDriver 2.34 🡪Supports Chrome v61-63

ChromeDriver 2.33 🡪Supports Chrome v60-62

**Firefox:**

Driver : [v0.21.0](https://github.com/mozilla/geckodriver/releases/tag/v0.21.0)

Mozilla Firefox 57 (and greater)

Selenium 3.11 (and greater)

Driver : [v0.19.0](https://github.com/mozilla/geckodriver/releases/tag/v0.21.0)

Mozilla Firefox 55 (and greater)

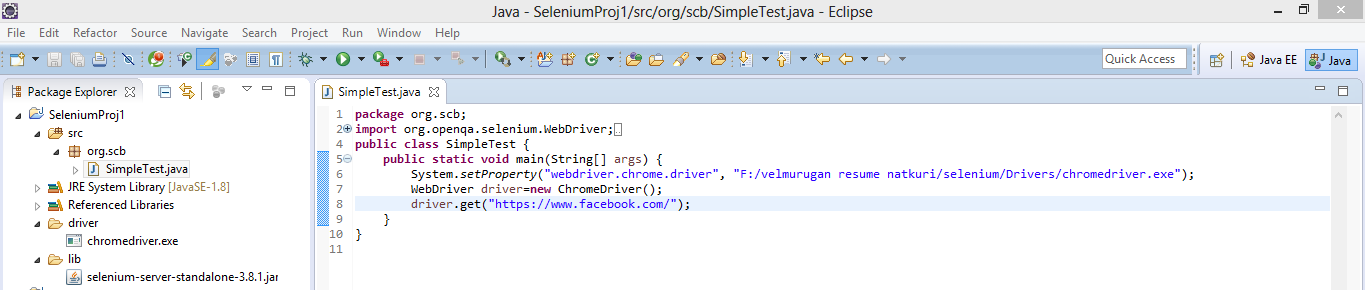
Selenium 3.5 (and greater)

**IE:**

Browser: Internet Explorer 11

Driver Version:

Write Simple basic selenium program?



**Mozilla Firefox:**

System.setProperty("webdriver.gecko.driver","path of geckodriver.exe");

WebDriver driver = new FirefoxDriver();

**IE:**

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

WebDriver driver = new InternetExplorerDriver();

**Simple Program?**

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

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

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

txtUserName.sendKeys("Java");

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

txtpassword.sendKeys("Java@123");

WebElement loginBtn=driver.findElement(By.*xpath*("//input[@value='Log In']"));

loginBtn.click();

driver.close();

**Action?**

Actions🡪class

Action🡪interface

webElement🡪class

**DragAndDrop?**

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

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

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

actions.dragAndDrop(source, target).perform();

**RightClick?**

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

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

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

actions.contextClick(target).perform();

**Mouse over action?**

WebElement web = driver.findElement(By.*xpath*(".//\*[text()='Product Category']"));

//mouse action declaration

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

a.moveToElement(web).perform();

driver.findElement(By.*xpath*(".//\*[text()='iPhones']")).click();

**Mouse RightClick and choose?**

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

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

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

actions.contextClick(target).perform();

Robot r=**new** Robot();

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

r.keyRelease(KeyEvent.***VK\_DOWN***);

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

r.keyRelease(KeyEvent.***VK\_DOWN***);

r.keyPress(KeyEvent.***VK\_ENTER***);

r.keyRelease(KeyEvent.***VK\_ENTER***);

//or

**for**(**int** i=1;i<=2;i++)

{

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

r.keyRelease(KeyEvent.***VK\_DOWN***);

}

r.keyPress(KeyEvent.***VK\_ENTER***);

r.keyRelease(KeyEvent.***VK\_ENTER***);

**DoubleClick inside textbox?**

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

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

actions.doubleClick(target).perform();

**Enter-Select-rightClick-past-into-anotherTxt?**

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

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

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

source.sendKeys("Java");

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

actions.doubleClick(source).perform();

actions.contextClick(source).perform();

Robot r=**new** Robot();

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

r.keyRelease(KeyEvent.***VK\_DOWN***);

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

r.keyRelease(KeyEvent.***VK\_DOWN***);

r.keyPress(KeyEvent.***VK\_ENTER***);

r.keyRelease(KeyEvent.***VK\_ENTER***);

actions.click(target).perform();

//Press Ctrl+V

r.keyPress(KeyEvent.***VK\_CONTROL***);

r.keyPress(KeyEvent.***VK\_V***);

//Release //Press Ctrl+V

r.keyRelease(KeyEvent.***VK\_CONTROL***);

r.keyRelease(KeyEvent.***VK\_V***);

**DropDown or Select Option?**

**1.Single value**

**2.Multiple value**

<select id="dropdown\_7" name="dropdown\_7" class=" piereg\_validate[required]">

<option value="Ice">Iceland</option>

<option value="Ind">India</option >

<option value="Pak">Pakistan</option

</select>

**Select : c**

**Methods:**

WebElement dDnCountry=driver.findElement(By.*id*("dropdown\_7"));

Select s=**new** Select(dDnCountry);

s.selectByValue("Ind");

s.selectByIndex(2);

s.selectByVisibleText("India");

s.deselectAll();

s.deselectByIndex(2);

s.deselectByValue("PK");

s.deselectByVisibleText("Pakistan");

List<WebElement> options = s.getOptions();

List<WebElement> allSelectedOptions = s.getAllSelectedOptions();

WebElement firstSelectedOption = s.getFirstSelectedOption();

**boolean** multiple = s.isMultiple();

**getAttribute(); and getText();**

It is a method, used to print the value whatever you gave in the text box

<input value="HiTx" id="name\_3\_lastname" name="last\_name" **class**="input\_fields piereg\_validate" type="text">Hi</input>

<label **class**="" **for**="dropdown\_7">Country</label>

Value,id,name,class,.. 🡪getAttribute()

Country 🡪 getText

Ex:

WebElement getAt=driver.findElement(By.*id*("name\_3\_lastname "));

String attribute = getAt.getAttribute("value");// HiTx

String text = getAt.getText();//Hi

**getOptions();**

WebElement dDnCountry=driver.findElement(By.*id*("dropdown\_7"));

Select s=**new** Select(dDnCountry);

List<WebElement> options = s.getOptions();

//Ehance for loop

**for** (WebElement webElement : options) {

System.***out***.println(webElement.getText());

}

//for Loop

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

String text = options.get(i).getText();

System.***out***.println(text);

s.selectByVisibleText(text);

}

**isEnabled();and isDisplayed();isSelected;**

isEnabled is used to check whether the particular webElement is enabled or not. Usually apply on only button.

isDisplayed is used to check whether the particular webElemet available or not in web Page. Some time at runtime the webElement not available. Apply for txt,radio,..

Ex:

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

**boolean** displayed = pass.isDisplayed();

**boolean** enabled = pass.isEnabled();

**boolean** selected = pass.isSelected();

**Verify RadioButton Checked or not?**

WebElement radio=driver.findElement(By.*id*("raido"));

radio.click();

**boolean** selected = radio.isSelected();

System.***out***.println(selected);

**Clear();**

This method is used to clear the content in textbox.

WebElement txt=driver.findElement(By.*id*("txt"));

txt.clear();

**getWindowHandle(); and getWindowHandles();**

Ex1:

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

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

String windowHandle = driver.getWindowHandle();

driver.switchTo().window("1");

Ex2:

Set<String> windowHandles = driver.getWindowHandles();

**for** (String x : windowHandles) {

**if**(x.equals("3"))

{

driver.switchTo().window(x);

}

}

**defaultContent();**

defaultContent is used to go home page or parent page.

WebDriver defaultContent = driver.switchTo().defaultContent();

**How to get particular windows?**

Set<String> windowHandles = driver.getWindowHandles();

List<String> ll=(ArrayList<String>)windowHandles;

driver.switchTo().window(ll.get(3));

**Write a program for Screen shot with base class Program?**

Program:

**package** org.scb;

**import** java.io.IOException;

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

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

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

**public** **class** SimpleTest **extends** base {

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

IOException {

WebDriver driver=*getDriver*("https://www.facebook.com/");

*takeScreenshot*(driver, "FbHome");

WebElement txtUserName = driver.findElement(By.*name*("email"));

*sendKey*(txtUserName, "Java");

*takeScreenshot*(driver, "UserName");

WebElement txtPassword = driver.findElement(By.*name*("pass"));

*sendKey*(txtPassword, "Java@123");

*takeScreenshot*(driver, "Password");

WebElement btnLogin = driver.findElement(By

.*xpath*("//input[@value='Log In']"));

*click*(btnLogin);

*takeScreenshot*(driver, "Login");

*quite*(driver);

}

}

Program:

**package** org.scb;

**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.WebElement;

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

**public** **class** base {

**public** **static** WebDriver getDriver(String url)

{

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

"C:/Users/Amaresh Gunasekar/Downloads/chromedriver\_win32 (1)/chromedriver.exe");

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

driver.get(url);

**return** driver;

}

**public** **static** **void** takeScreenshot(WebDriver driver,String name) **throws** IOException

{

TakesScreenshot tk = (TakesScreenshot) driver;

File scrPath = tk.getScreenshotAs(OutputType.***FILE***);

File desPath = **new** File(

"F:\\velmurugan resume natkuri\\selenium\\Document\\"+name+".png");

FileUtils.*copyFile*(scrPath, desPath);

}

**public** **static** **void** sendKey(WebElement element,String value)

{

element.sendKeys(value);

}

**public** **static** **void** click(WebElement element)

{

element.click();

}

**public** **static** **void** quite(WebDriver driver)

{

driver.quit();

}

}

**Alerts();**

Types:

Simple 🡪ok button

Confirm 🡪ok and cancel button

Promt 🡪 txt,ok and cancel button

Ex:

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

a.accept();//ok

a.dismiss();//cancel

a.sendKeys("Java");

a.accept();

**How to get txt from Alert?**

If alert has one txt box means we can use like below

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

**JavaScript?**

JavascriptExecutor is a Interface.

executeScript() is a method

argument[0].setAttribute('value','Inserted value')

**Ex:**

js.executeScript("argument[0].setAttribute('value','java')",txtemail);

**Ex:**

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

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

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

WebElement loginBtn=driver.findElement(By.*id*("Log In"));

JavascriptExecutor js=(JavascriptExecutor)driver;

js.executeScript("argument[0].setAttribute('value','java')",txtemail);

js.executeScript("argument[0].setAttribute('value','java@123')",txtpwd);

js.executeScript("argument[0].click",loginBtn);

WebElement txtFname=driver.findElement(By.*id*("f\_name"));

WebElement txtLname=driver.findElement(By.*id*("l\_name"));

js.executeScript("argument[1].setAttribute('value','java@123')",txtFname,txtLname);

js.executeScript("document.getElementById('email').setAttribute('value','Hello')");

// to print the value

Object s = js.executeScript("return document.getElementById('email'). getAttribute('value')");

System.***out***.println(s);

// to click login

js.executeScript("return document.getElementById('u\_0\_r').click()");

**ScrollUp and Down**

argument[0].ScrollIntoView(true) :scrollDown

argument[0].ScrollIntoView(false) :scrollUp

Ex:

JavascriptExecutor js=(JavascriptExecutor)driver;

WebElement btn1=driver.findElement(By.*xpath*("//a[text()=’val’]"));

WebElement btn2=driver.findElement(By.*xpath*("//a[text()=’sel’]"));

js.executeScript("argument[0].ScrollIntoView(true)",btn1);

js.executeScript("argument[0].ScrollIntoView(false)",btn2);

//OR

//here select which element we want to move

WebElement from = driver.findElement(By.*xpath*("//\*[text()='History']"));

//here select the destination we want to drop

WebElement to = driver.findElement(By.*xpath*("//\*[text()='Teens']")); Actions act=**new** Actions(driver);

//drog and drop perform act.clickAndHold(from).moveToElement(to).release(to).build().perform();

**PageDown?**

JavascriptExecutor j=(JavascriptExecutor) driver;

WebElement w = driver.findElement(By.*xpath*("//\*[text()='Live Demo']"));

w.sendKeys(Keys.***PAGE\_DOWN***);

**Implicit and explicitly wait() and Fluent Wait**

Implicit wait is wait for whenever call driver.findElement(By.id,xpath,..)

Default pooling time is 500ms. We can override this time Ex: WebDriverWait wait=**new** WebDriverWait(driver,20,600);Here we are override as 600.

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

WebDriverWait wait = new WebDriverWait(WebDriverRefrence,TimeOut);

EX:

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

WebElement myDynamicElement = driver.findElement(By.*id*("myDynamicElement"));

WebElement myElement = driver.findElement(By.*id*("myElement"));

**explicitly wait()**

WebElement myDynamicElement = driver.findElement(By.*id*("btn"));

WebElement myElement = driver.findElement(By.*id*("btn"));

WebDriverWait wait=**new** WebDriverWait(driver,20);

wait.until(ExpectedConditions.*elementToBeClickable*(myElement));

wait.until(ExpectedConditions.*elementToBeClickable*(myDynamicElement));

The following are the Expected Conditions that can be used in Explicit Wait

1. alertIsPresent()
2. elementSelectionStateToBe()
3. elementToBeClickable()
4. elementToBeSelected()
5. frameToBeAvaliableAndSwitchToIt()
6. invisibilityOfTheElementLocated()
7. invisibilityOfElementWithText()
8. presenceOfAllElementsLocatedBy()
9. presenceOfElementLocated()
10. textToBePresentInElement()
11. textToBePresentInElementLocated()
12. textToBePresentInElementValue()
13. titleIs()
14. titleContains()
15. visibilityOf()
16. visibilityOfAllElements()
17. visibilityOfAllElementsLocatedBy()
18. visibilityOfElementLocated()

**Fluent Wait()**

The fluent wait is used to tell the web driver to wait for a condition, as well as the **frequency** with which we want to check the condition before throwing an "ElementNotVisibleException" exception.

**Frequency:**Setting up a repeat cycle with the time frame to verify/check the condition at the regular interval of time

Let's consider a scenario where an element is loaded at different intervals of time. The element might load within 10 seconds, 20 seconds or even more then that if we declare an explicit wait of 20 seconds. It will wait till the specified time before throwing an exception. In such scenarios, the fluent wait is the ideal wait to use as this will try to find the element at different frequency until it finds it or the final timer runs out.

Wait wait = new FluentWait(WebDriver reference)

.withTimeout(timeout, SECONDS)

.pollingEvery(timeout, SECONDS)

.ignoring(Exception.class);

EX:

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

.withTimeout(30, TimeUnit.SECONDS)

.pollingEvery(5, TimeUnit.SECONDS)

.ignoring(NoSuchElementException.class);

WebElement clickseleniumlink = wait.until(new Function<Webdriver, WebElement>(){

public WebElement apply(WebDriver driver ) {

return driver.findElement(By.xpath("/html/body/div[1]/section/div[2]/div/div[1]/div/div[1]/div/div/div/div[2]/div[2]/div/div/div/div/div[1]/div/div/a/i"));

}

});

//click on the selenium link

clickseleniumlink.click();

**Idenfifier**

***Object Identification / locators:***

* **Id**
* **Class**
* **Name**
* **Xpath**
* **Link**
* **Partiallinktext**
* **Linktext**

***Firefox plug-ins to find locator:***

1. **Firebug**
2. **Firepath**

* **Add these 2 plug-ins in firefox- To find the locators**
* **In locators we can define 2 types (i.e.)**
* **/ 🡪 Absolute path (shows Full HTML DOM structure)**
* **// 🡪 Relative path(shows particular locators)**
* **In finding Xpath, matching node should be only one**

**Ex:**

driver.findElement(By.*xpath*("//\*[@id='pass']")).sendKeys("9047360818");

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

* **findElement()**🡪 is a method, it is used to find the locators
* sendKeys() 🡪 is a method, it is used to enter the value
* click() 🡪 is a method, it is used to click the login button or whatever
* Thread.sleep(3000)🡪 it is used wait for some time(here,3000ms) to execute

**Following, preceding, Following -sibling, preceding-sibling, Ancestor, contains?**

**Following is select all below tags at from webElement.**

**Preceding is select all above tags from WebElement**

**Ancestor is select all below and above tags the WebElement**

**Following- sibling is select all below child tags from webElement.**

**Preceding- sibling is select all above child tags from WebElement**

**Contains is select by give string**

**1.Contains**

**2.Sibling**

**3.Ancestor**

|  |  |  |  |
| --- | --- | --- | --- |
| short | long | Means | try it! |
| [none] | following::item | all the items that come after me | find all the title elements that occur afterthe titleStmt |
| [none] | following-sibling::l | the l children of my parent that come after me (i.e., the rest of the stanza) | find all the children of titleStmt that follow author |
| [none] | preceding::pb | the pbs that come before me | find all the ab elements that occur before To be, or not to be: |
| [none] | preceding-sibling::head | the head children of my parent that come before me | do any elements ever occur before the speaker within a speech? |
| @rend | attribute::rend | my rend attribute | list all the who attributes |
| @\* | attribute::\* | all of my attributes | do any of the stage elements have attributes at all? |

**Ex:**

Xpath: "//ul/li[contains(text(),'doprep')]/preceding-sibling::li"

Xpath: "//ul/li[contains(text(),'doprep')]/following-sibling::li"

Xpath : "//li[preceding-sibling::li='doprep']"

Xpath: "//li[following-sibling::li='doprep']"

Xpath : "//ul/li[preceding-sibling::li[.='doprep'] and following-sibling::li[.='savior']]"

Xpath: "//ul/li[preceding-sibling::li[contains(text(),'doprep')] and following-sibling::li[contains(text(),'Savior')]]"

Xpath: "//h4/a[contains(text(),'SAP M')]"

Xpath: "//div[.//a[text()='SELENIUM']]/ancestor::div[@class='rt-grid-2 rt-omega']/following-sibling::div"

Xpath: "//ul/li[contains(text(),'doprep')]/preceding::li"

Xpath: "//ul/li[contains(text(),'doprep')]/following::li"

**Close();**

It is used to close the application.

it close the current browser.

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

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

drive.close("https://www.facebook.com/");

**Quit():**

Destroy the object.

Close all browser opened in webdriver

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

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

drive.quite("https://www.facebook.com/");

***Browser Commands:***

Some basic methods:

getTitle()

It is a method, used to print the title of the page

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

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

driver.findElement(By.*xpath*("//\*[@id='email']")).sendKeys("vengat161193");driver.findElement(By.*xpath*("//\*[@id='pass']")).sendKeys("9047360818");

String s = driver.getTitle();

System.***out***.println(s);

*getCurrenturl():*

* It is a method, used to check whether particular page is open or not

**Ex:**

String s = driver.getCurrentUrl();

**if**(s.equals("http://www.adactin.com/HotelApp/index.php")){

System.***out***.println("u r in adactin website");

}

**else**{

System.***out***.println("u r not in adactin website");

}

**What are the Exception you have handled?**

**NoSuchElementException:**

It is throws when particular id/xpath/class or whatever is not available in DOM structure

**Navigate commands:**

**1.navigate().to():**

It is a method, used to navigate one website to another website.

**Ex:**

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

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

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

**2.refresh():**

It is a method, used to refresh the navigation website.

***Ex:***

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

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

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

driver.navigate().refresh();

**3.back():**

It is a method, used to move back to the first page.

***Ex:***

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

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

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

driver.navigate().back();

**4.forward();**

It is a method, used to move forward to next page.

***Ex:***

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

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

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

driver.navigate().back();

driver.navigate().forward();

**Browser Manage command:**

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

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

**Radio button:**

We can able to select only one at a time

***Ex:***

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

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

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

.click();

***Xpath:***

Structure or combination of absolute path and relative path

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

**for** (**int** i = 1; i <= 2; i++) {

Thread.*sleep*(3000);

String s = driver.findElement(

By.*xpath*("//\*[@id='u\_0\_k']/span[" + i + "]/label")).getText();

System.***out***.println(s);

***Finding Xpath using different ways:***

1. **//**label[contains(text(), “female”)]
2. **//**label[@class=’-58mg’][1]

* If we get 2 matching points , we give 1 or 2

1. //input[@id=’email’]
2. //\*[@id=’email’]

* If we use \*, consider any tag

**CSS Select:**

<a class=”value” id=”abc”,…>

**Ex:**

driver.findElement(By.*class(“a.class”)*);

driver.findElement(By.*class(“a#abc”)*);

driver.findElement(By.*class(“a#..”)*);

***Debug:***

* It is used run a program step by step in eclipse.
* Mark some lines from left corner of the program
* Then right click🡪debug as
* Then click to check step over/step down/step info in the eclipse top

**FrameWork**

1.Cucumber 🡪BDD(Behaviour DarivenDevelopmenet)

2.Data Driven is Excel sheet

3.POM (Page Object Model)

4.JUnit 🡪 TDD(Test DarivenDevelopmenet)

5.TestNg-It is advance version of JUnit 🡪 TDD(Test DarivenDevelopmenet)

**WebTable?**

<table>

<tr>

<td></td><td></td><td></td>

</tr>

<tr>

<td></td><td></td><td></td>

</tr>

<table>

<table>

<tr>

<td></td><td></td><td></td>

</tr>

<tr>

<td></td><td></td><td></td>

</tr>

<table>

..

**To print particular values(data) in the table:**

String s = driver.findElement(By.*xpath*(".//\*[@id='content']/table/tbody/tr[2]/td[2]")).getText();

System.***out***.println(s);

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

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

List<WebElement> tRows = driver.findElements(By.*tagName*("tr"));

**for**(WebElement rows:tRows){

List<WebElement> tData = rows.findElements(By.*tagName*("td"));

**for**(WebElement data:tData){

System.***out***.println(data.getText());

**Using Normal for loop**

List<WebElement> tRows = driver.findElements(By.*tagName*("tr"));

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

List<WebElement> tData = tRows.findElements(By.*tagName*("td"));

**for**(**int** j=0;j<tData.size();j++){

System.***out***.println(tData.get(j).getText()); }}

**To print particular data only:**

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

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

List<WebElement> tRows = driver.findElements(By.*tagName*("tr"));

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

List<WebElement> tData = tRows.findElements(By.*tagName*("td"));

**for**(**int** j=0;j<tData.size();j++){

**if**(tData.get(j).getText().equals("Mecca")){

System.***out***.println(tData.get(j).getText());

}

}

**To print relevant data(Dynamic table**

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

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

List<WebElement> tRows = driver.findElements(By.*tagName*("tr"));

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

List<WebElement> tData = tRows.findElements(By.*tagName*("td"));

**for**(**int** j=0;j<tData.size();j++){

**if**(tData.get(j).getText().equals("Dubai")){

driver.findElement(By.*xpath*(".//\*[@id='content']/table/tbody/tr["+i+"]/td[6]/a"));

System.***out***.println(tData.get(j).getText());

}

}

**To print relevant data Multiple table**

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

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

WebElement w= driver.findElements(By.class*Name*("tss-table-s13"));

List<WebElement> tRows = w.findElements(By.*tagName*("tr"));

**for**(WebElement rows:tRows){

List<WebElement> tData = rows.findElements(By.*tagName*("td"));

**for**(WebElement data:tData){

System.***out***.println(data.getText());

**CHECK BOX:**

* In check box, we can able to select more than one value at a time.

**Select one value:**

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

driver.get("http://demoqa.com/registration/");

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

**Select more than one value:**

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

driver.get("http://demoqa.com/registration/");

List<WebElement> w = driver.findElements(By.*xpath*("//input[@type='checkbox']"));

**for**(WebElement x:w){

x.click();

}

Here,

* //input[@type='checkbox'] 🡪 if xpath we give like, we get 3 matching nodes, so using for loop we can able to select 3 checkbox at a time
* **findElements**🡪 **is a method, used to select more than one value**
* **WebElement**🡪 **is a interface**
* **By**🡪 **is a class name**

**To select two values:**

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

driver.get("http://demoqa.com/registration/");

List<WebElement> w = driver.findElements(By.*xpath*("//input[@type='checkbox']"));

**for**(WebElement x:w){

**if**(x.getAttribute("value").equals("dance")||x.getAttribute("value").equals("cricket ")){

x.click();

}}

**Using normal for loop to select all checkbox:**

List<WebElement> w = driver.findElements(By.*xpath*("//input[@type='checkbox']"));

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

w.get(i).click();

}}

}

**Using normal for loop to select two checkbox:**

List<WebElement> w = driver.findElements(By.*xpath*("//input[@type='checkbox']"));

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

**if**(w.get(i).getAttribute("value").equals("dance")||w.get(i).getAttribute("value").equals("cricket ")){

w.get(i).click();

}}

**To print the selected value**

List<WebElement> w = driver.findElements(By

.*xpath*("//input[@type='checkbox']"));

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

**if** (w.get(i).getAttribute("value").equals("dance")

|| w.get(i).getAttribute("value").equals("cricket")) {

w.get(i).click();

}

**if** (w.get(i).isSelected()) {

System.***out***.println(w.get(i).getAttribute("value"));

}

}

}

**To select the unselected value:**

List<WebElement> w = driver.findElements(By

.*xpath*("//input[@type='checkbox']"));

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

**if** (w.get(i).getAttribute("value").equals("dance")

|| w.get(i).getAttribute("value").equals("cricket")) {

w.get(i).click();

}

**if** (!w.get(i).isSelected()) {

w.get(i).click();

}

**iFrame**

iFrame is best example for Overloading

<iframe id=”1231 >

</iframe>

**How to find frame?**

\*DOM Stractor

\*Right click->frame specs

**Iframe can use below to get?**

driver.switchTo().frame("a0ud");// String id

driver.switchTo().frame("g\_gd");// String name

driver.switchTo().frame(driver.findElement(By.*id*("jj\_i")));// WebElement

driver.switchTo().frame(1);// int index

**Ex:**

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

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

driver.switchTo().frame("a0ud");// String id

WebElement txtname = driver.findElement(By.*id*("jj\_i"));// WebElement

txtname.click();

**How to comeout from the Iframe?**

driver.switchTo().defaultContent();

**How to count of iframe?**

List<WebElement> count = driver.findElements(By.*tagName*("iframe"));//iframe is a tag name

System.***out***.println(count.size());

driver.switchTo().frame(1);

**Write a base class method for page nation?**

**How to upload file method:**

We can’t upload a file using selenium.

So we have one class is there for uploading file (i.e.) Robot class or AutoIT

**Ex:**

**public** **static** **void** uploadFiles(File path) {

**try** {

Robot robot = **new** Robot();

robot.setAutoDelay(3000);

StringSelection selection = **new** StringSelection(

path.getAbsolutePath());

Toolkit.*getDefaultToolkit*().getSystemClipboard()

.setContents(selection, **null**);

**// press ctrl+v**

robot.keyPress(KeyEvent.***VK\_CONTROL***);

robot.keyPress(KeyEvent.***VK\_V***);

robot.setAutoDelay(3000);

**// release ctrl+v**

robot.keyRelease(KeyEvent.***VK\_CONTROL***);

robot.keyRelease(KeyEvent.***VK\_V***);

**// press enter**

robot.setAutoDelay(3000);

robot.keyPress(KeyEvent.***VK\_ENTER***);

robot.keyRelease(KeyEvent.***VK\_ENTER***);

} **catch** (AWTException e) {

e.printStackTrace();

}

}

Here ,

* Robot 🡪 class
* StringSelection 🡪 class
* Using this method we can upload any file.