**Operations on Web Elements**

**How to Type in text box using Selenium Web Driver**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class InputTxt{

public static void main(String[] args) {

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

"C:\\Users\\dasar\\Downloads\\chromedriver\_win32");

WebDriver driver = new ChromeDriver();

String url = "https://www.tutorialspoint.com/index.htm";

driver.get(url);

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

// identify element and input text inside it

WebElement l =driver.findElement(By.className("gsc-input"));

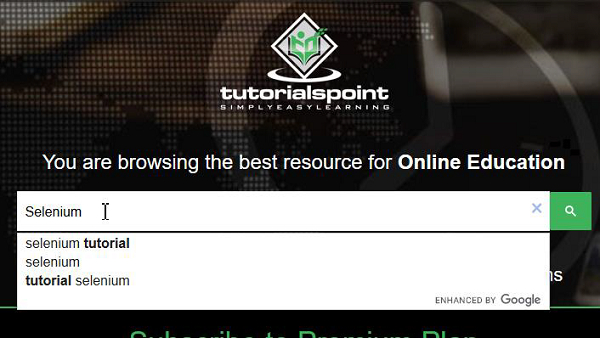
l.sendKeys("Selenium");

driver.quit();

}

}

Output



**How to get a Text Box value using Selenium Web Driver**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import java.util.concurrent.TimeUnit;

public class GetTextTyped{

public static void main(String[] args) {

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

"C:\\Users\\ghs6kor\\Desktop\\Java\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

//implicit wait

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

//URL launch

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

// identify element

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

t.sendKeys("Tutorialspoint");

// obtain value with getAttribute

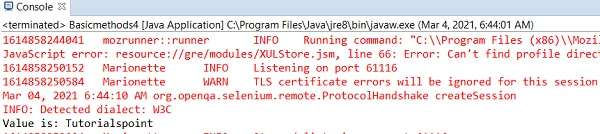
System.out.println("Value is: " + t.getAttribute("value"));

driver.quit();

}

}

Output



**How to read the placeholder value of a Text Box using getAttribute() method**

package attribute;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class AttributeTest {

WebDriver driver;

//Method to set up the browser and open the dummy website

@BeforeTest

public void setUp() {

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

driver = new ChromeDriver();

driver.get("https://phptravels.com/demo/");

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

}

//Test method to get the different attribute values

@Test

public void getAttrVals() {

//Fetching the image source

WebElement image = driver.findElement(By.xpath("//img[@class=\"well-sm icon-resources img100\"]"));

System.out.println("The image source is : "+image.getAttribute("src"));

//Fetching the placeholder value

WebElement emailField = driver.findElement(By.id("address"));

String placeholderVal = emailField.getAttribute("placeholder");

String expectedVal = "your@email.com";

if(placeholderVal.equalsIgnoreCase(expectedVal))

System.out.println("The placeholder value is expected, ie : "+placeholderVal);

else

System.out.println("The placeholder value is not correct --- " +placeholderVal);

//Getting attribute value of an attribute that is not present on the page

WebElement logInBtn = driver.findElement(By.xpath("//\*[@class=\"btn yx-nfn yx-njg\"]"));

System.out.println("The value of an attribute that doesn't exist is : " +logInBtn.getAttribute("type"));

}

//Method to close the webdriver session

@AfterTest

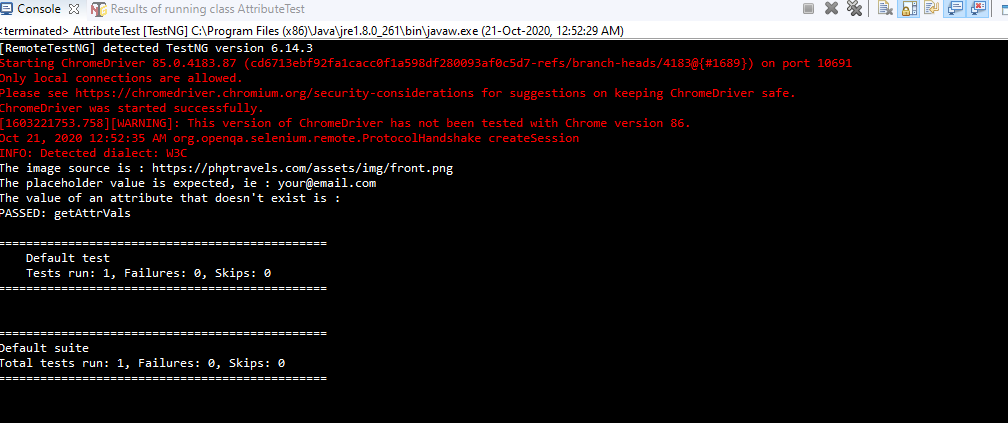
public void burnDown() {

driver.quit();

}

}

Output



**Deleting/clear text from the Text Box**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class DelDefaultVal{

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Users\\ghs6kor\\Desktop\\Java\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String url = "https://www.tutorialspoint.com/index.htm";

driver.get(url);

driver.manage().timeouts().implicitlyWait(4, TimeUnit.SECONDS);

// identify element

WebElement l = driver.findElement(By.id("gsc-i-id1"));

// input text

l.sendKeys("Selenium");

// delete default value with clear()

l.clear();

driver.quit()

}

}

**Selecting the Radio Button by Label Text / value**

package TestPackage;

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 Selenium {

public static void main(String[] args) {

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

WebDriver driver = new ChromeDriver();

driver.get("https://www.demoqa.com/radio-button");

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

/\*\*

\* Find radio button using ID, Validate isSelected and then click to select

\*/

WebElement radioEle = driver.findElement(By.id("yesRadio"));

boolean select = radioEle.isSelected();

System.out.print(select);

// performing click operation if element is not already selected

if (select == false) {

radioEle.click();

}

/\*\*

\* Find radio button using Xpath, Validate isDisplayed and click to select

\*/

WebElement radioElem = driver.findElement(By.xpath("//div/input[@id='impressiveRadio']"))

boolean sel = radioEle.isDisplayed();

// performing click operation if element is displayed

if (sel == true) {

radioElem.click();

}

/\*\*

\* Find radio button using CSS Selector, Validate isEnabled and click to select

\*/

WebElement radioNo = driver.findElement(By.cssSelector("input[id='noRadio']"));

boolean selectNo = radioEle.isDisplayed();

// performing click operation if element is enabled

if (selectNo == true) {

radioNo.click();

}

}

}

**Find out number of elements in a radio group**

package project1;

import java.io.IOException;

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 radio {

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

WebDriver driver = new FirefoxDriver();

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

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

driver.get("file:///E:/Selenium class/Programs/Sonali/bin/project1/form2selenium.html");

WebElement radio = driver.findElement(By.xpath("//input[@id='i-green']"));

radio.click();

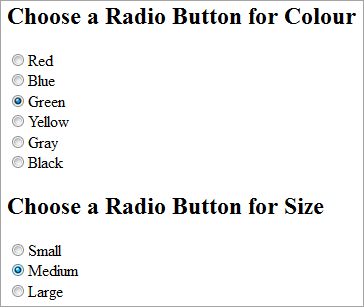
WebElement radio2 = driver.findElement(By.xpath("//input[@value='Medium']"));

radio2.click();

}

}

Output

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2020/04/Radio1_selenium.png)

**Print first selected option from a dropdown**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

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

public class SelecedItem{

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Users\\ghs6kor\\Desktop\\Java\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String u =" https://www.tutorialspoint.com/selenium/selenium\_automation\_practice.htm"driver.get(u);

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

// identify element

WebElement t=driver.findElement(By.xpath("//\*[@name='continents']"));

//Select class for dropdown

Select select = new Select(t);

// select an item with text visible

select.selectByVisibleText("Australia");

// get selected option with getFirstSelectedOption() with its text

WebElement o = select.getFirstSelectedOption();

String selectedoption = o.getText();

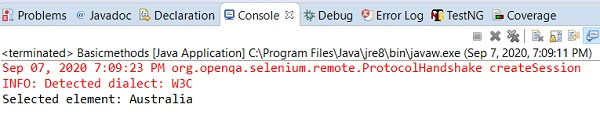
System.out.println("Selected element: " + selectedoption);

driver.close();

}

}

Output



**Select an option by value from a dropdown**

package newpackage;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

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

import org.openqa.selenium.By;

public class accessDropDown {

public static void main(String[] args) {

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

String baseURL = "http://demo.guru99.com/test/newtours/register.php";

WebDriver driver = new FirefoxDriver();

driver.get(baseURL);

Select drpCountry = new Select(driver.findElement(By.name("country")));

drpCountry.selectByVisibleText("ANTARCTICA");

//Selecting Items in a Multiple SELECT elements

driver.get("http://jsbin.com/osebed/2");

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

fruits.selectByVisibleText("Banana");

fruits.selectByIndex(1);

}

}

**Select an option by visible text from a dropdown**

package seleniumProject;

import java.util.ArrayList;

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 GetAllSelectedOptions

{

public static void main(String[] args)

{

WebDriver driver;

driver = new FirefoxDriver();

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

String URL = "https://selenium08.blogspot.com/2019/11/dropdown.html";

driver.get(URL);

// Wait for some time to load the whole web page.

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

// Locate dropdown element on web page by By.xpath.

WebElement dropdown = driver.findElement(By.xpath("//select[@name='Month']"));

// Create an object of Select class and pass the dropdown of type WebElement as an argument.

Select multiplechoicelist = new Select(dropdown);

// Verify that dropdown allows the multiple-choice list or not.

if(multiplechoicelist.isMultiple())

{

System.out.println( "Dropdown allows multiple-choice list");

}

else {

System.out.println("Dropdown does not allow multiple-choice");

}

// If multiple-choice allow, select the different choice by sending visible texts.

multiplechoicelist.selectByVisibleText("July");

multiplechoicelist.selectByVisibleText("May");

multiplechoicelist.selectByVisibleText("March");

// Verify that the number of choices in the list selected.

if(multiplechoicelist.getAllSelectedOptions().size() == 3)

{

System.out.println("3 options have been chosen:");

}

else {

System.out.println("Code not worked");

}

// Create a List. For this, create an object of ArrayList class by using the reference of List interface.

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

// Call add() method of List to add expected elements for selection.

expectedSelection.add("July");

expectedSelection.add("May");

expectedSelection.add("March");

// Iterating WebElement by using the advanced for loop to retrieve the actually selected elements and then add selected elements in the list using add() method.

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

for(WebElement element : multiplechoicelist.getAllSelectedOptions())

{

actualSelection.add(element.getText()); // Here, getText() method of WebElement class has been used to add the text label of all the options in the list.

}

// Now compare actualSelection with expectedSelection by using containsAll() method to check that correct options are selected in the list..

if(actualSelection.containsAll(expectedSelection))

{

System.out.println(actualSelection);

}

driver.close();

}

}

**Output:**

Dropdown allows multiple-choice list

3 options have been chosen:

[March, May, July]

**Select an option by index from a dropdown**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

import java.util.List;

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

public class SelectOptions{

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "C:\\Users\\ghs6kor\\Desktop\\Java\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String url = "https://www.tutorialspoint.com/tutor\_connect/index.php"; driver.get(url);

driver.manage().timeouts().implicitlyWait(12, TimeUnit.SECONDS);

Select s = new Select(driver.findElement(By.xpath("//select[@name=’selType’]")));

// select an option by value method

s.selectByValue("name");

Thread.sleep(1000);

// select an option by index method

s.selectByIndex(0);

Thread.sleep(1000);

// select an option by visible text method

s.selectByVisibleText("By Subject");

Thread.sleep(1000);

driver.quit();

}

}

**Clicking a link using partialLinkText**

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Partial\_Link {

public static void main(String[] args) {

// System Property for Chrome Driver

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

// Instantiate a ChromeDriver class.

WebDriver driver=new ChromeDriver();

// Launch Website

driver.navigate().to("https://www.testandquiz.com/selenium/testing.html");

}

}

**Clicking a link using link Text**

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

public class LinkText {

public static void main(String[] args) {

// TODO Auto-generated method stub

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

WebDriver driver=new ChromeDriver();

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

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

//Opening the air bnb Goa homestays page

driver.get("https://www.airbnb.co.in/s/Goa/all");

//locating an element via link text in Selenium now, and clicking on that stay

driver.findElement(By.linkText("Standard One Bedroom Suite with Pool &amp; Jacuzzi")).click();

driver.quit();

}

}

**Find out all the links in a web page**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.\*;

public class P1 {

public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/newtours/";

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

WebDriver driver = new ChromeDriver();

String underConsTitle = "Under Construction: Mercury Tours";

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

driver.get(baseUrl);

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

String[] linkTexts = new String[linkElements.size()];

Int i = 0;

//extract the link texts of each link element

for (WebElement e : linkElements) {

linkTexts[i] = e.getText();

i++;

}

//test each link

for (String t : linkTexts) {

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

if (driver.getTitle().equals(underConsTitle)) {

System.out.println("\"" + t + "\""

+ " is under construction.");

} else {

System.out.println("\"" + t + "\""

+ " is working.");

}

driver.navigate().back();

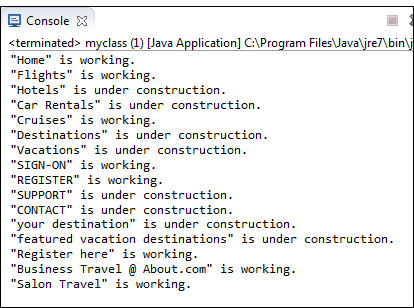
}

driver.quit();

}

}

**Output**

[](https://cdn.guru99.com/images/image013(1).png)

**Clicking on an image link**

package newproject;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class MyClass {

public static void main(String[] args) {

String baseUrl = "https://www.facebook.com/login/identify?ctx=recover";

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

WebDriver driver = new ChromeDriver();

driver.get(baseUrl);

//click on the "Facebook" logo on the upper left portion

driver.findElement(By.cssSelector("a[title=\"Go to Facebook home\"]")).click();

//verify that we are now back on Facebook's homepage

if (driver.getTitle().equals("Facebook - log in or sign up")) {

System.out.println("We are back at Facebook's homepage");

} else {

System.out.println("We are NOT in Facebook's homepage");

}

driver.close();

}

}

**How to find an element that contains a specific text using contains()**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class TextMatch{

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "C:\\Users\\ghs6kor\\Desktop\\Java\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String url = "https://www.tutorialspoint.com/index.htm";

driver.get(url);

driver.manage().timeouts().implicitlyWait(12, TimeUnit.SECONDS);

// identify element with text()

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

// identify element with contains()

WebElement m=driver.findElement(By.xpath("//\*[contains(text(),'GATE')]"));

System.out.println("Element with text(): " + l.getText());

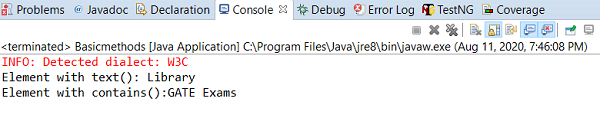
System.out.println("Element with contains(): " + m.getText());

driver.quit();

}

}

Output



**How to find an element using text()**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class Match{

public static void main(String[] args) {

System.setProperty("<Path of the ChromeDriver>");

WebDriver driver = new ChromeDriver();

String url = "https:/browserstack.com”;

driver.get(url);

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

// Locating element with text()

WebElement e = driver.findElement(By.xpath("//\*[text()='Get started free']"));

System.out.println("Element with text(): " + e.getText() );

driver.quit();

}

}

**Web Driver Methods**

**get()**Syntax: get(url)

Example: driver.get();

Purpose: It will load a new web page in the current browser window. This is done using an http get operation, and the method will block until the load is complete.

Parameters: URL - The URL to load and it should be a fully qualified URL

**getCurrentUrl()**

Syntax: getCurrentUrl()

Example: driver.getCurrentUrl();

Purpose: Gets a string representing the current URL that the browser is opened.

Returns: The URL of the page currently loaded in the browser

**getTitle()**

Syntax: getTitle()

Example: driver.getTitle();

Purpose: Gets the title of the current web page.

Returns: The title of the current page, with leading and trailing white space stripped, or null if one is not already set

**getPageSource()**

Syntax: getPageSource()

Example: driver.getPageSource();

Purpose: Get the source of the currently loaded page. If the page has been modified after loading (for example, by Javascript) there is no guarantee that the returned text is that of the modified page.

Returns: The source of the current page

**findElement()**

Syntax: WebElement findElement(By by)

Example: driver.findElements(By.xpath("//");

Purpose: Find the first WebElement using the given method.

Parameters: By - The locating mechanism

Returns: The first matching element on the current page

Throws: NoSuchElementException - it will return exception if no matching elements are found

**findElements()**

The findElements command returns an empty list if there are no elements found using the given locator strategy and locator value. Below is the syntax of findElements command.

List<WebElement> elementName = driver.findElements(By.LocatorStrategy("LocatorValue"));

Example:

List<WebElement> listOfElements = driver.findElements(By.xpath("//div"));

**getWindowHandles()**

Syntax: Set getWindowHandles()

Example: driver.getWindowHandles();

Purpose: Return a set of window handles which can be used to iterate over all the open windows of this Webdriver instance by passing them to switchTo().WebDriver.Options.window()

Returns: A set of window handles which can be used to iterate over all the open windows.

**getWindowHandle()**

Syntax: String getWindowHandle()

Example: driver.getWindowHandle();

Parameter: Return an opaque handle to this window that uniquely identifies it within this driver instance. This can be used to switch to this window at a later date

switchTo

WebDriver.TargetLocator switchTo()

The next future commands will be performed to a different frame or window.

Returns: A Target Locator which can be used to switch or select a frame or window

**switchTo()**

package Demo;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class iFramesDemo {

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

System.setProperty("webdriver.gecko.driver","D:\\Data\_Personal\\Demo \\geckodriver-v0.23.0-win64\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

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

//Finding all iframe tags on a web page

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

int numberOfTags = elements.size();

System.out.println("No. of Iframes on this Web Page are: " +numberOfTags);

// Switch to the frame using the frame id

System.out.println("Switching to the frame");

driver.switchTo().frame("aswift\_0");

// Print the frame source code

System.out.println("Frame Source" +driver.getPageSource());

// Switch back to main web page

driver.switchTo().defaultContent();

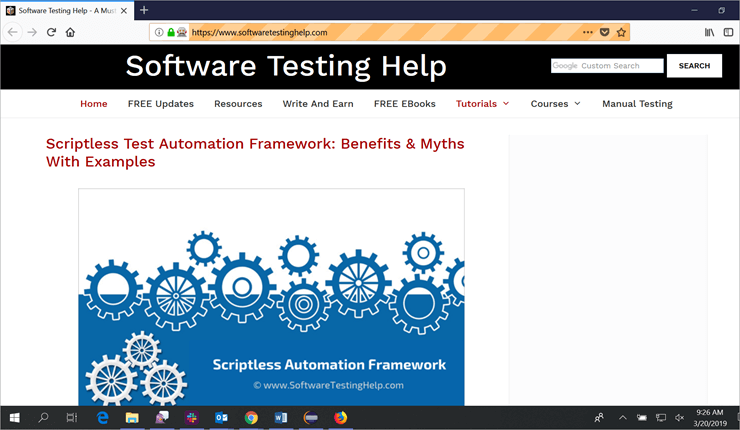
driver.quit();

}

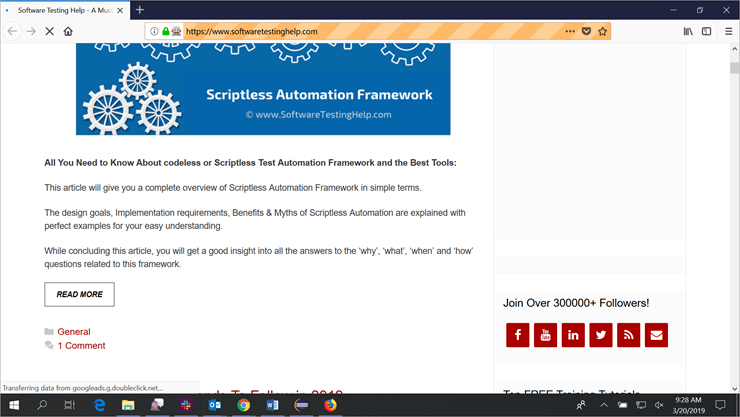
}

**Output:**

Open the website: https://www.softwaretestinghelp.com

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/04/CodeOutput1.png)

Switch to the frame named aswift\_0.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/04/CodeOutput2-1.png)

Print the number of iframes on the web page on the eclipse console window.

[Code Output 3](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/04/CodeOutput3.png)

Print the source code of frame on the eclipse console after switching to the frame.

https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/04/word-image-1.png

**manage()**

Syntax: WebDriver.Options manage()

Purpose: Gets the Option interface

Returns: An option interface

**driver.navigate().to()**

driver.navigate().to() is used to launch the particular website by passing the URL but we can use forward and backward button to navigate between the pages during test case writing.

**There are two ways to define this method:**

**This method Loads a new web page in the current browser window. It accepts a String parameter and returns void.**

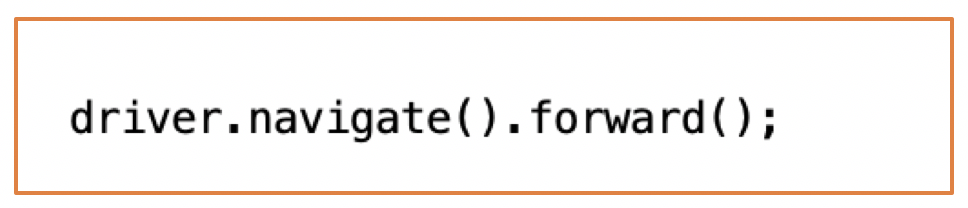


**This method accepts URL class instance by importing package import java.net.URL; and returns void.**



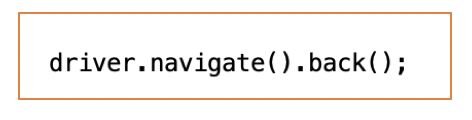
**2)     Forward Command – driver.navigate().forward();**

This method enables the web browser to click on the forward button in the existing browser window, it takes you forward by one page on the browser’s history.  It neither accepts nor returns anything.

* 

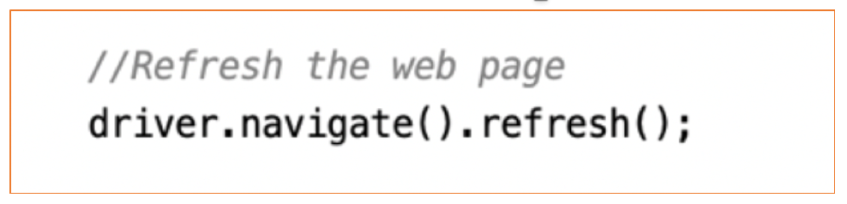
3) **Back Command*–*driver.navigate().back();**

This method enables the web browser to click on the **back** button in the existing browser window. it takes you backwards by one page on the browser’s history. It neither accepts nor returns anything.

* 

4) **Refresh Command- driver.navigate().refresh();**

This method refreshes the current page. It neither accepts nor returns anything.

* 

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

public class Navigation\_command {

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.testandquiz.com/selenium/testing.html");

//Click on the Link Text using click() command

driver.findElement(By.linkText("This is a Link")).click();

//Go back to Home Page

driver.navigate().back();

//Go forward to Registration page

driver.navigate().forward();

// Go back to Home page

driver.navigate().to("https://www.testandquiz.com/selenium/testing.html");

//Refresh browser

driver.navigate().refresh();

//Closing browser

driver.close();

}

}

**driver.navigate().back()**

**driver.navigate().forward()**

**driver.navigate().refresh()**

package automationFramework;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Navigationcommand {

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

// TODO Auto-generated method stub

String exePath = "chromedriver\_win32\\chromedriver.exe";

System.setProperty("webdriver.chrome.driver", exePath);

WebDriver driver = new ChromeDriver();

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

//navigateTo use

driver.navigate().to(URL);

System.out.println("The facebook is launches successfully");

//navigate refresh

driver.navigate().refresh();

System.out.println("The browser is refreshed");

driver.findElement(By.linkText("Forgotten account?")).click();

//navigate back

driver.navigate().back();

System.out.println("performed click on back browser button");

Thread.sleep(1000);

//navigate forward

driver.navigate().forward();

System.out.println("navigated again back to forgot password page");

//closing the browser

driver.close();

System.out.println("The browser is closed successfully and test is passed");

}

}

**Output**

Starting ChromeDriver 2.30.477700 (0057494ad8732195794a7b32078424f92a5fce41) on port 15811

Only local connections are allowed.

Aug 05, 2017 4:45:55 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: OSS

The facebook is launches successfully

The browser is refreshed

performed click on back browser button

navigated again back to forgot password page

The browser is closed successfully and test is passed

**click()**

Selenium offers a ‘.click()’ method to help you perform various mouse-based operations for your web-application. You can use the Selenium click button method for various purposes such as selecting the radio button and checkbox or simply clicking on any button or link, drag and drop, click and hold, etc.

package pack1;

import java.util.List;

import java.util.Random;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class ClickRandom {

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

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

WebDriver driver = new ChromeDriver();

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

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

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

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

int linkcount = link.size();

System.out.println(linkcount);

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

String linktext = link.get(i).getText();

System.out.println(linktext);

}

Random r = new Random();

link.get(r.nextInt(2)).click();

String title = driver.getTitle();

System.out.println("Page Title is " + title);

Thread.sleep(5000);

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

}

}

**sendKeys()**

sendkeys() is a method in Selenium that allows QAs to type content automatically into an editable field while executing any tests for forms. These fields are web elements that can be identified using locators like element id, name, class name, etc.

Consider a scenario in which a developer has created a website, and now the website needs to be tested for functionality. Here’s where QAs come into the picture. They start writing test scripts using Selenium WebDriver in order to replicate user actions on a browser.

In some cases, QAs need to provide data in specific data input fields to validate functionality. For example, to test a login page, the username and password fields require some data to be entered. Here’s where QAs use the sendkeys() method to enter the field values.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.\*;

public class Signup {

public static void main(String[] args)

{

//declaration and instantiation of objects/variables System.setProperty("webdriver.chrome.driver","G:\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String baseUrl = “https://www.browserstack.com/users/sign\_up";

driver.get(baseUrl);

//Get the Web Element corresponding to the field Business Email (Textfield)

WebElement email = driver.findElement(By.id(“user\_email\_login”));

//Get the Web Element corresponding to the Password Field

WebElement password = driver.findElement(By.id(“user\_password”));

//Find the Sign me in button

WebElement login = driver.findElement(By.id("user\_submit"));

email.sendKeys(“abc@xyz.com”);

password.sendKeys(“abcdef12345”)

login.click();

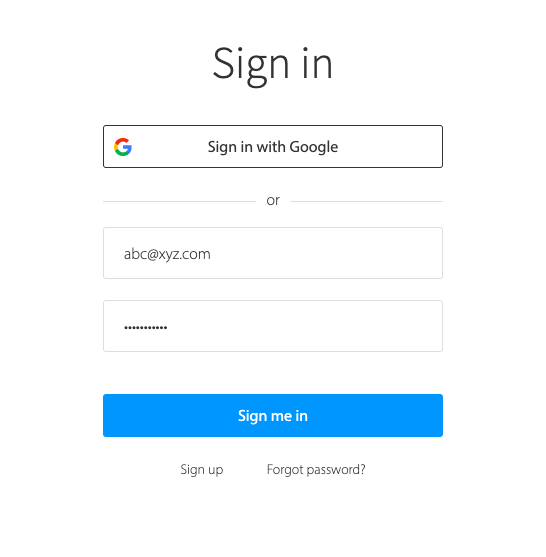
System.out.println("Signed in with Click");

driver.close();

}

}

Output



**getAttribute()**

The getAttribute() method is declared in the WebElement interface, and it returns the value of the web element’s attribute as a string. For attributes having boolean values, the getAttribute() method will return either true or null.

public class GetAttributeSample

{

public static void main(String[] args)

{

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

WebDriver driver= new ChromeDriver();

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

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

WebElement searchTextBox= driver.findElement(By.id("search\_form\_input\_homepage"));

// retrieving html attribute value using getAttribute() method

String typeValue=searchTextBox.getAttribute("type");

System.out.println("Value of type attribute: "+typeValue);

String autocompleteValue=searchTextBox.getAttribute("autocomplete");

System.out.println("Value of autocomplete attribute: "+autocompleteValue);

// Retrieving value of attribute which does not exist

String nonExistingAttributeValue=searchTextBox.getAttribute("nonExistingAttribute");

System.out.println("Value of nonExistingAttribute attribute: "+nonExistingAttributeValue);

}

}

**Output:**

Value of type attribute: text

Value of autocomplete attribute: off

Value of nonExistingAttribute attribute: null

When the above code is executed, it automatically fetches the attributes – type and autocomplete. For the attribute which is not available, it returns the null value.

**getTagName()**

It gets the tag name of the current element.

package com.selenium.practise;

import org.openqa.selenium.By;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.Point;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class ExampleWebElementMethods {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "E:\\Soft Wares\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String Url = "https://google.com";

driver.get(Url);

WebElement sigin = driver.findElement(By.id("gb\_70"));

//getLocation() WebElement method

Point location = sigin.getLocation();

System.out.println("Location of X :"+ location.getX()+ "," +"Location of Y :" +location.getY());

//getSize() WebElement method

Dimension size = sigin.getSize();

System.out.println("Width :"+size.getWidth()+ "," +"Height : "+size.getHeight());

//getText() WebElement method

String text = sigin.getText();

System.out.println("Text :" +text);

//getTagName() WebElement method

String tagName = sigin.getTagName();

System.out.println("Tagname :"+tagName);

//close the browser

driver.close();

}

}

**Output:**

Location of X :934,Location of Y :16

Width :69,Height : 30

Text :Sign in

Tagname :a

**getText()**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class GetTextMethd{

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "C:\\Users\\ghs6kor\\Desktop\\Java\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String url ="https://www.tutorialspoint.com/videotutorials/subscription.php";

driver.get(url);

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

// identify element

WebElement l = driver.findElements(By.cssSelector("h2"));

// extract text with getText()

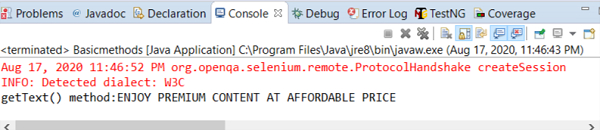
System.out.println("getText() method:" + l.getText());

driver.quit();

}

}

**Output**



**isDisplayed()**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class VisibleElement{

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Users\\ghs6kor\\Desktop\\Java\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

driver.get("https://www.tutorialspoint.com/about/about\_careers.htm");

// identify element

WebElement p=driver.findElement(By.xpath("//h1"));

//isDisplayed() to check if element visible

boolean s= p.isDisplayed();

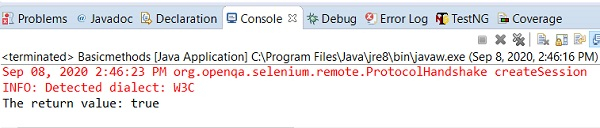
System.out.println("The return value: " + s);

driver.close();

}

}

**Output**



**isEnabled()**

package 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;

public class IsEnabledMethodEx

{

public static void main(String[] args)

{

// Create an object of FirefoxDriver class.

WebDriver driver = new FirefoxDriver();

// Launch the Firefox browser and open URL.

String URL = “https://www.google.com”;

driver.get(URL);

// Maximize browser window.

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

// Delay for specified amount of time to load page.

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

// Declare and initialize variable to store the expected title of the webpage.

String expectedTitle = “Google”;

// Fetch title of the web page and save it using a variable actualTitle with data type string.

String actualTitle = driver.getTitle();

// Compare expected title of the page with actual title of the page and print the result.

if(expectedTitle.equals(actualTitle))

{

System.out.println(“Verification Successful – Correct title is displayed on the home webpage.”);

}

else

{

System.out.println(“Verification Failed: Incorrect title is displayed on the home webpage.”);

}

// Check that “Google Search box” is enabled or not. If Search box is enabled, it will return true.

WebElement searchBox = driver.findElement(By.xpath(“//input[@name=’q’]”));

if(searchBox.isEnabled())

{

System.out.println(“Search box is enabled. Return: ” +searchBox.isEnabled());

}

else

{

System.out.println(“Search box is disabled. Return: ” +searchBox.isEnabled());

}

// Enter text in the “Google Search” box.

WebElement sendText = driver.findElement(By.xpath(“//input[@name=’q’]”));

sendText.sendKeys(“Selenium Tool”);

// Check that “Google Search button” is enabled or not.

WebElement searchButton = driver.findElement(By.xpath(“//input[@class=’gNO89b’]”));

if(searchButton.isEnabled()) {

System.out.println(“Search button is enabled. Return: ” +searchButton.isEnabled());

}

else {

System.out.println(“Search button is not enabled. Return: ” +searchButton.isEnabled());

}

driver.close();

}

}

**Output:**

Verification Successful – Correct title is displayed on the home webpage.

Search box is enabled. Return: true

Search button is enabled. Return: true

**isSelected()**

package 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;

public class IsSelectedMethodEx

{

public static void main(String[] args)

{

WebDriver driver = new FirefoxDriver();

String URL = "https://selenium08.blogspot.com/2019/07/check-box-and-radio-buttons.html";

driver.get(URL);

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

WebElement red = driver.findElement(By.xpath("//input[@name='color' and @value='red']"));

// Check if it is enabled before selecting it.

if(red.isEnabled()) {

System.out.println("Red checkbox is enabled. Return: " +red.isEnabled());

// Since check box is enabled. Therefore, click it.

red.click();

// Now check that it is selected or not.

if(red.isSelected()) {

System.out.println("Red checkbox is selected. Return: " +red.isSelected());

}

else {

System.out.println("Red checkbox is not selected. Return: " +red.isSelected());

}

}

else {

System.out.println("Red checkbox is not enabled. Return: " +red.isEnabled());

}

// Find XPath for radio button Opera.

WebElement opera=driver.findElement(By.xpath("//input[@name='browser' and @value='Opera']"));

// Check radio button opera is selected or not.

if(opera.isSelected()) {

System.out.println("Radio button opera is selected. Return: " +opera.isSelected());

}

else {

System.out.println("Radio button opera is not selected. Return: " +opera.isSelected());

}

driver.close();

}

}

**Output:**

Red checkbox is enabled. Return: true

Red checkbox is selected. Return: true

Radio button opera is not selected. Return: false

**driver.close()**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

import java.util.List;

import java.util.Set;

import java.util.Iterator;

public class CloseWindow {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "Path to Chrome Driver");

WebDriver driver = new ChromeDriver();

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

//implicit wait

driver.manage().timeouts().implicitlyWait(2, TimeUnit.SECONDS);

driver.findElement(By.name(“q")).sendkeys(Selenium);

driver.findElement(By.name(“btnk")).click();

// window handles

Set w = driver.getWindowHandles();

// window handles iterate

Iterator t = w.iterator();

String h = (String) t.next();

String p = (String) t.next();

// switching child window

driver.switchTo().window(h);

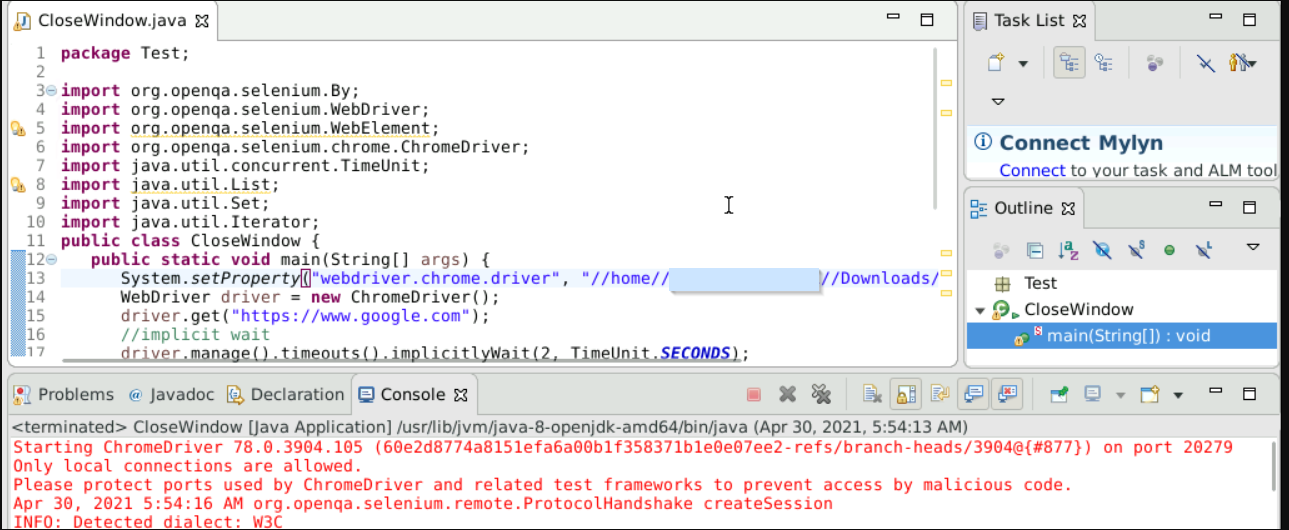
// close only the child browser window

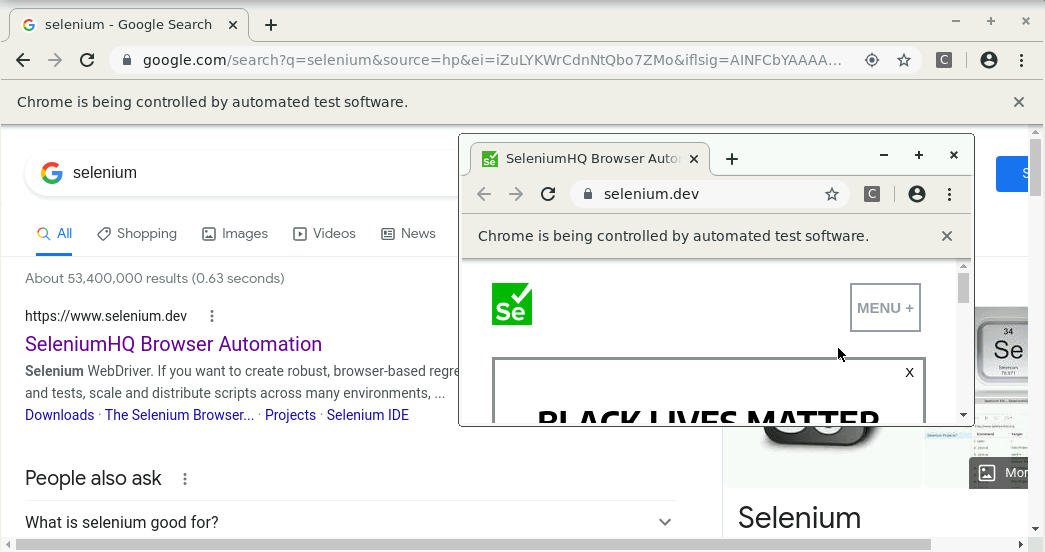
driver.close();

}

}’

**Output**



****

**getLocation()**

package com.selenium.practise;

import org.openqa.selenium.By;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.Point;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class ExampleWebElementMethods {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "E:\\Soft Wares\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String Url = "https://google.com";

driver.get(Url);

WebElement sigin = driver.findElement(By.id("gb\_70"));

//getLocation() WebElement method

Point location = sigin.getLocation();

System.out.println("Location of X :"+ location.getX()+ "," +"Location of Y :" +location.getY());

//getSize() WebElement method

Dimension size = sigin.getSize();

System.out.println("Width :"+size.getWidth()+ "," +"Height : "+size.getHeight());

//getText() WebElement method

String text = sigin.getText();

System.out.println("Text :" +text);

//getTagName() WebElement method

String tagName = sigin.getTagName();

System.out.println("Tagname :"+tagName);

//close the browser

driver.close();

}

}

**Output:**

Location of X :934,Location of Y :16

Width :69,Height : 30

Text :Sign in

Tagname :a

**4. Popups/Alerts and Windows**

**Capturing the alert message using getText()**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import java.util.concurrent.TimeUnit;

public class JsEnterText{

public static void main(String[] args) {

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

"C:\\Users\\ghs6kor\\Desktop\\Java\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

//implicit wait

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

//URL launch

driver.get("https://the-internet.herokuapp.com/javascript\_alerts");

// identify element

WebElement l = driver.

findElement(By.xpath("//\*[text()='Click for JS Alert']"));

l.click();

//switch focus to alert

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

//get alert text

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

System.out.println("Alert text is: " + s);

//accepting alert

a.accept();

driver.quit();

}

}

**Output**

https://www.tutorialspoint.com/assets/questions/media/50540/Alert1.jpg

**Prompt Alert with Text Box to enter the text**

public String prompt(String expectedDialogText, String inputText) {

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

String actualDialogText = alert.getText();

if (inputText != null)

alert.sendKeys(inputText);

alert.accept();

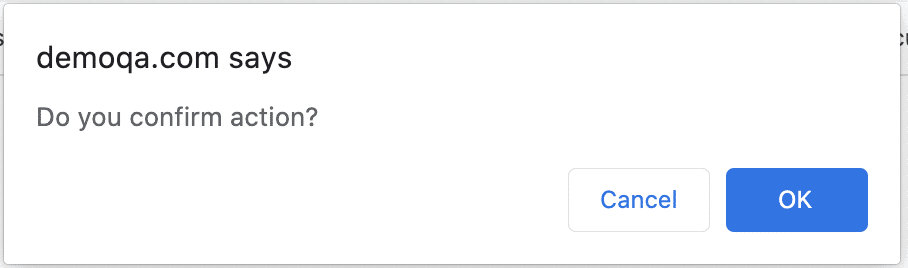
checkDialogText(driver, expectedDialogText, actualDialogText);

return actualDialogText;

}

**Confirmation Alert with Ok and Cancel buttons**

Confirmation Alert: These alerts get some confirmation from the user in the form of accepting or dismissing the message box. They are different from prompt alerts in a way that the user cannot enter anything as there is no text-box available. Users can only read the message and provide the inputs by pressing the OK/Cancel button.



**Clicking OK button of the alert using accept()**

void accept() // To click on the 'OK' button of the alert.

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

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.NoAlertPresentException;

import org.openqa.selenium.Alert;

public class AlertDemo {

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

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

WebDriver driver = new ChromeDriver();

// Alert Message handling

driver.get("http://demo.guru99.com/test/delete\_customer.php");

driver.findElement(By.name("cusid")).sendKeys("53920");

driver.findElement(By.name("submit")).submit();

// Switching to Alert

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

// Capturing alert message.

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

// Displaying alert message

System.out.println(alertMessage);

Thread.sleep(5000);

// Accepting alert

alert.accept();

}

}

**Output :**

When you execute the above code, it launches the site. Try to delete Customer ID by handling confirmation alert that displays on the screen, and thereby deleting customer id from the application.

**Clicking Cancel button of the alert using dismiss()**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.Alert;

public class alert\_test {

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.testandquiz.com/selenium/testing.html");

//Handling alert boxes

//Click on generate alert button

driver.findElement(By.linkText("Generate Alert Box")).click();

//Using Alert class to first switch to or focus to the alert box

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

//Using accept() method to accept the alert box

alert.accept();

//Handling confirm box

//Click on Generate Confirm Box

driver.findElement(By.linkText("Generate Confirm Box")).click();

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

//Using dismiss() command to dismiss the confirm box

//Similarly accept can be used to accept the confirm box

((Alert) confirmBox).dismiss();

}

}

**Handle single window using driver.getWindowHandle()**

package selenium;

import org.testng.annotations.Test;

import java.util.Iterator;

import java.util.Set;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class MultipleWindowsClass{

@Test

public void testMultipleWindows() throws InterruptedException{

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

// To open browser

WebDriver driver = new ChromeDriver();

// To maximize browser

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

// To open Naukri website with multiple windows

driver.get("<a href="http://www.naukri.com/">http://www.naukri.com/</a>");

// It will return the parent window name as a String

String mainWindow=driver.getWindowHandle();

// It returns no. of windows opened by WebDriver and will return Set of Strings

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

// Using Iterator to iterate with in windows

Iterator<String> itr= set.iterator();

while(itr.hasNext()){

String childWindow=itr.next();

// Compare whether the main windows is not equal to child window. If not equal, we will close.

if(!mainWindow.equals(childWindow)){

driver.switchTo().window(childWindow);

System.out.println(driver.switchTo().window(childWindow).getTitle());

driver.close();

}

}

// This is to switch to the main window

driver.switchTo().window(mainWindow);

}

}

**Switch to window using driver.switchTo().window()**

package selenium;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class demo5 {

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

{

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

WebDriver driver = new ChromeDriver();

driver.get("<a href="http://toolsqa.com/automation-practice-switch-windows/">http://toolsqa.com/automation-practice-switch-windows/</a>");

String parentWindowHandle = driver.getWindowHandle();

System.out.println("Parent window's handle -> " + parentWindowHandle);

WebElement clickElement = driver.findElement(By.id("button1"));

for(int i = 0; i < 3; i++)

{

clickElement.click();

Thread.sleep(3000);

}

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

String lastWindowHandle = "";

for(String handle : allWindowHandles)

{

System.out.println("Switching to window - > " + handle);

System.out.println("Navigating to google.com");

driver.switchTo().window(handle); //Switch to the desired window first and then execute commands using driver

driver.get("<a href="http://google.com">http://google.com</a>");

lastWindowHandle = handle;

}

//Switch to the parent window

driver.switchTo().window(parentWindowHandle);

//close the parent window

driver.close();

//at this point there is no focused window, we have to explicitly switch back to some window.

driver.switchTo().window(lastWindowHandle);

driver.get("<a href="http://toolsqa.com">http://toolsqa.com</a>");

}

}

**Switch to frame using driver.switchTo().frame()**

public class SwitchToFrame\_ID {

public static void main(String[] args) {

WebDriver driver = new FirefoxDriver(); //navigates to the Browser driver.get("http://demo.guru99.com/test/guru99home/");

// navigates to the page consisting an iframe

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

driver.switchTo().frame("a077aa5e"); //switching the frame by ID

System.out.println("\*\*\*\*\*\*\*\*We are switch to the iframe\*\*\*\*\*\*\*");

driver.findElement(By.xpath("html/body/a/img")).click();

//Clicks the iframe

System.out.println("\*\*\*\*\*\*\*\*\*We are done\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

**Output:**

Browser navigates to the page consisting the above iframe and clicks on the iframe.

**Switch to popup using driver.switchTo.alert()**

import java.util.Iterator;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Popup\_Demo {

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

WebDriver driver=new FirefoxDriver(); driver.get("Webpage link");

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

Thread.sleep();

driver.findElement(By.id("PopUp")).click(); // Clicking on the popup button

Robot robot = new Robot();

robot.mouseMove(400.5); // Navigating through mouse hover. Note that the coordinates might differ, kindly check the coordinates of x and y axis and update it accordingly.

robot.mousePress(InputEvent.BUTTON1\_DOWN\_MASK);

Thread.sleep(2000);

robot.mouseRelease(InputEvent.BUTTON1\_DOWN\_MASK);

Thread.sleep(2000);

driver.quit();

}

}

**5. Selenium Miscellaneous Scenarios**

**Write a test case to capture the screenshots with WebDriver**

package captureScreenshot;

// Import all classes and interface

import java.io.File;

import library.Utility;

import org.openqa.selenium.io.FileHandler;

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.ITestResult;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.Test;

public class FacebookScreenshot {

// Create Webdriver reference

WebDriver driver;

@Test

public void captureScreenshot() throws Exception

{

// Initiate Firefox browser

driver=new FirefoxDriver();

// Maximize the browser

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

// Pass application url

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

// Here we are forcefully passing wrong id so that it will fail our testcase

driver.findElement(By.xpath(".//\*[@id='emailasdasdas']")).sendKeys("Learn Automation");

}

// It will execute after every test execution

@AfterMethod

public void tearDown(ITestResult result)

{

// Here will compare if test is failing then only it will enter into if condition

if(ITestResult.FAILURE==result.getStatus())

{

try

{

// Create refernce of TakesScreenshot

TakesScreenshot ts=(TakesScreenshot)driver;

// Call method to capture screenshot

File source=ts.getScreenshotAs(OutputType.FILE);

// Copy method specific location here it will save all screenshot in our project home directory and

// result.getName() will return name of test case so that screenshot name will be same

try{

FileHandler.copy(source, new File("./Screenshots/"+result.getName()+".png"));

System.out.println("Screenshot taken");

}

}

catch (Exception e)

{

System.out.println("Exception while taking screenshot "+e.getMessage());

}

}

// close application

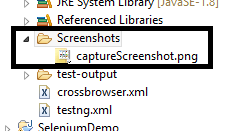
driver.quit();

}

}

**Output**

The above code will execute fine and if the test case will fail it will capture the screenshot. Check below screenshot

[](https://i1.wp.com/learn-automation.com/wp-content/uploads/2015/08/Takes-screenshot-on-failure-in-Selenium-WEbdriver.png)

**Find out the broken links on a web page**

package automationPractice;

import java.io.IOException;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.Iterator;

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 BrokenLinks {

private static WebDriver driver = null;

public static void main(String[] args) {

// TODO Auto-generated method stub

String homePage = "http://www.zlti.com";

String url = "";

HttpURLConnection huc = null;

int respCode = 200;

driver = new ChromeDriver();

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

driver.get(homePage);

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

Iterator<WebElement> it = links.iterator();

while(it.hasNext()){

url = it.next().getAttribute("href");

System.out.println(url);

if(url == null || url.isEmpty()){

System.out.println("URL is either not configured for anchor tag or it is empty");

continue;

}

if(!url.startsWith(homePage)){

System.out.println("URL belongs to another domain, skipping it.");

continue;

}

try {

huc = (HttpURLConnection)(new URL(url).openConnection());

huc.setRequestMethod("HEAD");

huc.connect();

respCode = huc.getResponseCode();

if(respCode >= 400){

System.out.println(url+" is a broken link");

}

else{

System.out.println(url+" is a valid link");

}

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

driver.quit();

}

}

**Implicit and Explicit wait commands**

**Implicit**

Package waitExample;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.\*;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

public class WaitTest {

private WebDriver driver;

private String baseUrl;

private WebElement element;

@BeforeMethod

public void setUp() throws Exception {

driver = new FirefoxDriver();

baseUrl = "http://www.google.com";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testUntitled() throws Exception {

driver.get(baseUrl);

element = driver.findElement(By.id("lst-ib"));

element.sendKeys("Selenium WebDriver Interview questions");

element.sendKeys(Keys.RETURN);

List<WebElement> list = driver.findElements(By.className("\_Rm"));

System.out.println(list.size());

}

@AfterMethod

public void tearDown() throws Exception {

driver.quit();

}

}

**Explicit**

package waitExample;

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.support.ui.ExpectedConditions;

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

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

public class ExpectedConditionExample {

// created reference variable for WebDriver

WebDriver driver;

@BeforeMethod

public void setup() throws InterruptedException {

// initializing driver variable using FirefoxDriver

driver=new FirefoxDriver();

// launching gmail.com on the browser

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

// maximized the browser window

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

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

}

@Test

public void test() throws InterruptedException {

// saving the GUI element reference into a "element" variable of WebElement type

WebElement element = driver.findElement(By.id("Email"));

// entering username

element.sendKeys("dummy@gmail.com");

element.sendKeys(Keys.RETURN);

// entering password

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

// clicking signin button

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

// explicit wait - to wait for the compose button to be click-able

WebDriverWait wait = new WebDriverWait(driver,30);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(text(),'COMPOSE')]")));

// click on the compose button as soon as the "compose" button is visible

driver.findElement(By.xpath("//div[contains(text(),'COMPOSE')]")).click();

}

@AfterMethod

public void teardown() {

// closes all the browser windows opened by web driver

driver.quit();

}

}

**Action class with the following operations**

**Keyboard key press event**

**Pressing enter button on the key board**

**ClickAndHold event, Drag and Drop**

**MoveToElement, Mouse Hover Event**

**Double Click event**

package automationFramework;

import static org.junit.Assert.assertEquals;

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;

public class KeyboardEvents {

public static void main(String[] args) {

// Initialize ChromeDriver

// Here we assume that the ChromeDriver path has been set in the System Global variables

WebDriver driver=new ChromeDriver();

//Navigate to the demo site

driver.get("https://demoqa.com/text-box");

// Enter the Full Name

WebElement fullName = driver.findElement(By.id("userName"));

fullName.sendKeys("Mr.Peter Haynes");

//Enter the Email

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

email.sendKeys("PeterHaynes@toolsqa.com");

// Enter the Current Address

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

currentAddress.sendKeys("43 School Lane London EC71 9GO");

// Copy the Current Address

currentAddress.sendKeys(Keys.CONTROL);

currentAddress.sendKeys("A");

currentAddress.sendKeys(Keys.CONTROL);

currentAddress.sendKeys("C");

//Press the TAB Key to Switch Focus to Permanent Address

currentAddress.sendKeys(Keys.TAB);

//Paste the Address in the Permanent Address field

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

permanentAddress.sendKeys(Keys.CONTROL);

permanentAddress.sendKeys("V");

//Compare Text of current Address and Permanent Address

assertEquals(currentAddress.getAttribute("value"),permanentAddress.getAttribute("value"));

driver.close();

}

}

**Pressing enter button on the key board**

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);

**Drag and Drop**

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.Actions;

public class DragAndDropExample {

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

//Set system properties for geckodriver System.setProperty("webdriver.gecko.driver", "Path\_of\_the\_driver");

WebDriver driver = new FirefoxDriver();

String URL = "https://the-internet.herokuapp.com/drag\_and\_drop";

driver.get(URL);

// It is always advisable to Maximize the window before performing DragNDrop action driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(10000, TimeUnit.MILLISECONDS);

//Actions class method to drag and drop

Actions builder = new Actions(driver);

WebElement from = driver.findElement(By.id("column-a "));

WebElement to = driver.findElement(By.id("column-b"));

//Perform drag and drop

builder.dragAndDrop(from, to).perform();

//verify text changed in to 'Drop here' box

String textTo = to.getText();

if(textTo.equals("Dropped!")) {

System.out.println("PASS: File is dropped to target as expected");

}else {

System.out.println("FAIL: File couldn't be dropped to target as expected");

}

driver.close();

}

}

**MoveToElement**

public Actions moveToElement​(WebElement target)

Moves the mouse to the middle of the element. The element is scrolled into view and its location is calculated using getClientRects.

Parameters:

target - element to move to.

Returns:

A self reference.

**Mouse Hover Event**

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

public class MouseHover1 {

public static WebDriver driver;

public static void main(String[] args) {

//Set system properties for geckodriver This is required since Selenium 3.0

System.setProperty("webdriver.gecko.driver", "C:\\Selenium\\Toolsqa\\lib\\geckodriver.exe");

// Launch the URL

driver.get("https://demoqa.com/menu/");

System.out.println("demoqa webpage Displayed");

//Maximise browser window

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

//Adding wait

driver.manage().timeouts().implicitlyWait(10000, TimeUnit.MILLISECONDS);

//Instantiate Action Class

Actions actions = new Actions(driver);

//Retrieve WebElement 'Music' to perform mouse hover

WebElement menuOption = driver.findElement(By.xpath(".//div[contains(text(),'Music')]"));

//Mouse hover menuOption 'Music'

actions.moveToElement(menuOption).perform();

System.out.println("Done Mouse hover on 'Music' from Menu");

//Now Select 'Rock' from sub menu which has got displayed on mouse hover of 'Music'

WebElement subMenuOption = driver.findElement(By.xpath(".//div[contains(text(),'Rock')]"));

//Mouse hover menuOption 'Rock'

actions.moveToElement(subMenuOption).perform();

System.out.println("Done Mouse hover on 'Rock' from Menu");

//Now , finally, it displays the desired menu list from which required option needs to be selected

//Now Select 'Alternative' from sub menu which has got displayed on mouse hover of 'Rock'

WebElement selectMenuOption = driver.findElement(By.xpath(".//div[contains(text(),'Alternative')]"));

selectMenuOption.click();

System.out.println("Selected 'Alternative' from Menu");

// Close the main window

driver.close();

}

}

**Double Click event**

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.openqa.selenium.interactions.Action;

import org.openqa.selenium.interactions.Actions;

import java.util.concurrent.TimeUnit;

public class DoubleClick{

public static void main(String[] args) {

System.setProperty("<path of the chrome driver>");

WebDriver driver = new ChromeDriver();

driver.get("URL of target website / webpage") Define the URL of the target website.

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

//Instantiating Actions class

Actions act = new Actions(driver);

//Locate WebElement to perform double click

btnElement = driver.findElement(<locator of the element>);

//Double Click the button

act.doubleClick(btnElement).perform();

System.out.println("Double click operation performed");

driver.quit();

}

}

**Web Table operations Get row count Get data from a specific cell Dynamic web table handling**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.List;

public class webtable {

public static void main (String[] args)

{

System.setProperty("webdriver.chrome.driver","./src/main/resources/chromedriver");

WebDriver driver = new ChromeDriver();

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

driver.get("https://demoqa.com/webtables");

//Store the table size

WebElement webtable = driver.findElement(By.xpath("//div[contains(@class,'ReactTable')]"));

//Get the rows which has data

List<WebElement> rowsWithData = webtable.findElements(By.xpath("//div[contains(@class,'rt-td') and text()]/ancestor::div[contains(@class,'rt-tr-group')]"));

//Print the text of 2nd row

System.out.println("Data of 2nd row is: \n" + rowsWithData.get(1).getText());

driver.quit();

}

}

**Dynamic web table handling**

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 IterateWebTableInSelenium {

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

//Setting the path of Chrome Browser Driver

String BrowserDriverPath= "C:\\SeleniumBrowserDrivers\\chromedriver.exe";

//Setting System property for Chrome browser Driver.

System.setProperty("webdriver.chrome.driver",BrowserDriverPath);

//Create a new Instance for Chrome Browser

WebDriver driver = new ChromeDriver();

driver.get("https://myskillpoint.com/handle-dynamic-web-tables-in-selenium-webdriver/");

List<WebElement> rowElements = driver.findElements(By.xpath("//table[@id='tablepress-7']//tbody/tr"));

int iRowCount=rowElements.size();

List<WebElement> headerElements = driver.findElements(By.xpath("//table[@id='tablepress-7']//thead/tr/th"));

int icolCount = headerElements.size();

for(int i=1;i<=iRowCount;i++)

{

System.out.println("Printing values for Row Number: "+ i);

for(int j=1;j<=icolCount;j++)

{

System.out.println("Table Cell Values for row column ("+i+","+j+"): "+ driver.findElement(By.xpath("//table[@id='tablepress-7']//tbody/tr["+i+"]/td["+j+"]")).getText());

}

}

}

}

**Output**

Printing values for Row Number: 1

Table Cell Values for row column (1,1): Shoe

Table Cell Values for row column (1,2): 3000.00

Table Cell Values for row column (1,3): 50

Printing values for Row Number: 2

Table Cell Values for row column (2,1): Jacket

Table Cell Values for row column (2,2): 5000.00

Table Cell Values for row column (2,3): 20

**Handling Ajax Auto suggestion**

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://demo.guru99.com/test/ajax.html";

WebDriver driver;

WebDriverWait wait;

@BeforeClass

public void setUp() {

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

//create chrome instance

driver = new ChromeDriver();

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

driver.navigate().to(URL);

}

@Test

public void test\_AjaxExample() {

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

wait = new WebDriverWait(driver, 5);

wait.until(ExpectedConditions.presenceOfElementLocated(container));

//Get the text before performing an ajax call

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

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

//Click on the radio button

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

//Click on Check Button

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

/\*Get the text after ajax call\*/

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

wait.until(ExpectedConditions.visibilityOf(TextElement));

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

/\*Verify both texts before ajax call and after ajax call text.\*/

Assert.assertNotEquals(textBefore, textAfter);

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

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

/\*Verify expected text with text updated after ajax call\*/

Assert.assertEquals(textAfter, expectedText);

driver.close();

}}

**Select a specific date from a calendar**

import java.util.List;

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 {

@Test

public void dateTimePicker(){

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

WebDriver driver = new ChromeDriver();

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

driver.get("http://demo.guru99.com/test/");

//Find the date time picker control

WebElement dateBox = driver.findElement(By.xpath("//form//input[@name='bdaytime']"));

//Fill date as mm/dd/yyyy as 09/25/2013

dateBox.sendKeys("09252013");

//Press tab to shift focus to time field

dateBox.sendKeys(Keys.TAB);

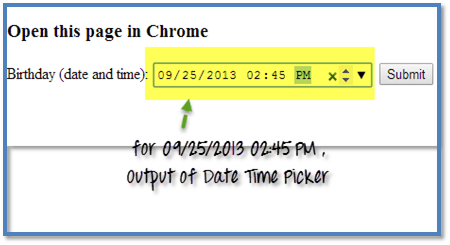
//Fill time as 02:45 PM

dateBox.sendKeys("0245PM");

}

}

**Output**

[](https://cdn.guru99.com/images/AdvanceSelenium/071514_0744_HandlingDat3.png)

**Cookies operations-Adding and Deleting cookies, Profile creation for browsers**

**Add a set of cookies**

//This method adds set of cookies for a domain

public void addCookiesToBrowser(Set<Cookie> cookies, String domain) {

for (Cookie c : cookies) {

if (c != null) {

if (c.getDomain().contains(domain)){

driver.manage().addCookie(

new Cookie(name, value, domain, path, expiry));

}

}

}

driver.navigate().refresh();

}

**Delete a specific Cookie**

//This method deletes a specific cookie

public void deleteCookieNamed(String name) {

driver.manage().deleteCookieNamed(name);

}

**Delete all Cookies**

//This method deletes all cookies

public void deleteAllCookies() {

driver.manage().deleteAllCookies();

}

**Profile creation for browsers**

// 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("xyzProfile");

// 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();

}

}

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.

**6. TestNG**

**Know the order of execution of TestNG annotations**

package com;

import org.testng.annotations.AfterClass;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterSuite;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class TestClass1 {

@BeforeSuite

public void beforeSuite() {

System.out.println(“Inside Before Suite Method.”);

}

@BeforeClass

public void beforeClass() {

System.out.println(“Inside Before Class of TestClass1.”);

}

@BeforeTest

public void beforeTest() {

System.out.println(“Inside Before Test Method of TestClass1.”);

}

@BeforeMethod

public void beforeMethod1() {

System.out.println(“Inside Before Method 1 of TestClass1.”);

}

@BeforeMethod

public void beforeMethod2() {

System.out.println(“Inside Before Method 2 of TestClass1.”);

}

@Test

public void testMethod1() {

System.out.println(“Inside Test Method 1 of TestClass1.”);

}

@Test

public void testMethod2() {

System.out.println(“Inside Test Method 2 of TestClass1.”);

}

@AfterTest

public void afterTest() {

System.out.println(“Inside After Test Method of TestClass1.”);

}

@AfterMethod

public void afterMethod1() {

System.out.println(“Inside After Method 1 of TestClass1.”);

}

@AfterMethod

public void afterMethod2() {

System.out.println(“Inside After Method 2 of TestClass1.”);

}

@AfterClass

public void afterClass() {

System.out.println(“Inside After Class Method of TestClass1.”);

}

@AfterSuite

public void afterSuite() {

System.out.println(“Inside After Suite Method.”);

}

}

Now, execute the TestNG class as TestNG Test, and you will see the output as below.

**Output:**

Inside Before Suite Method.

Inside Before Test Method of TestClass1.

Inside Before Class of TestClass1.

Inside Before Method 1 of TestClass1.

Inside Before Method 2 of TestClass1.

Inside Test Method 1 of TestClass1.

Inside After Method 1 of TestClass1.

Inside After Method 2 of TestClass1.

Inside Before Method 1 of TestClass1.

Inside Before Method 2 of TestClass1.

Inside Test Method 2 of TestClass1.

Inside After Method 1 of TestClass1.

Inside After Method 2 of TestClass1.

Inside After Class Method of TestClass1.

Inside After Test Method of TestClass1.

PASSED: testMethod1

PASSED: testMethod2

===============================================

Default test

Tests run: 2, Failures: 0, Skips: 0

===============================================

Inside After Suite Method.

===============================================

Default suite

Total tests run: 2, Failures: 0, Skips: 0

**Create testing.xml file to run the test cases in a class file**

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

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="Practice Suite">

<test name="Test Basics 1">

<parameter name="emailid" value="tester456@gmail.com"/>

<parameter name="password" value="test@123"/>

<classes>

<class name="practiceTests.testParameters"/>

</classes>

</test> <!-- Test -->

<test name="Test Basics 2">

<parameter name="emailid" value="tester789@gmail.com"/>

<classes>

<class name="practiceTests.testOptional"/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

**Create a test suite and test groups**

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >

<suite name="Test-Suite" >

<test name="ToolsQA" >

<classes>

<class name="TestNG" />

</classes>

</test>

</suite>

package com.howtodoinjava.groupExamples;

import org.testng.annotations.Test;

public class TestGroupExample

{

@Test(groups = { "test-group" })

public void testMethodOne() {

System.out.println("Test method one belonging to group.");

}

@Test

public void testMethodTwo() {

System.out.println("Test method two not belonging to group.");

}

@Test(groups = { "test-group" })

public void testMethodThree() {

System.out.println("Test method three belonging to group.");

}

}

**List down the assertions and use them in a test case**

package com.tests;

import static org.testng.Assert.assertFalse;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class BrowserStackTutorials {

@Test

public void testAssertFunctions() {

System.setProperty("webdriver.chrome.driver", "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable App & Cross Browser Testing Platform");

assertFalse(verifyTitle);

}

}

**Disable or ignore a test from running**

**1. How to disable test**

Disabling a test in TestNG can be achieved by setting the enabled attribute of the @Test annotation to false.

@Test( enabled=false )

This will disable the said test method from being executed as part of the test suite. If this attribute is set for the @Test annotation at the class level, all the tests inside the class will be disabled.

**2. Example of disabling tests**

In below test, we have three test methods i.e. testMethodOne(), testMethodTwo() and testMethodThree(). Out of these testMethodTwo() needs to be disabled.

public class DisableTestDemo

{

@Test(enabled = true)

public void testMethodOne() {

System.out.println("Test method one.");

}

@Test(enabled = false)

public void testMethodTwo() {

System.out.println("Test method two.");

}

@Test

public void testMethodThree() {

System.out.println("Test method three.");

}

}

**Output of above test run is given below:**

[TestNG] Running: C:\Users\somepath\testng-customsuite.xml

Test method one.

Test method three.

PASSED: testMethodOne

PASSED: testMethodThree

===============================================

Default test

Tests run: 2, Failures: 0, Skips: 0

**Make one test script dependent on the other and run both of them**

public class DependentTestExamples

{

@Test(dependsOnMethods = { "testTwo" })

public void testOne() {

System.out.println("Test method one");

}

@Test

public void testTwo() {

System.out.println("Test method two");

}

}

The preceding test class contains two test methods which print a message name onto the console when executed. Here, test method testOne depends on test method testTwo.

This is configured by using the attribute dependsOnMethods while using the Test annotation.

**Let’s run the tests now.**

Test method two

Test method one

PASSED: testTwo

PASSED: testOne

In the above test result you can see the message Test method two printed before the Test method one message. This shows that the testOne method got executed after testTwo as it depends on testTwo.

**Set priority to all the tests, execute and observe the order of execution**

package testNGPriority;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;

public class FirefoxTest

{

WebDriver driver; // Creating reference of WebDriver.

@Test(priority = 1)

public void driverSetup()

{

System.out.println("Running Firefox ");

driver = new FirefoxDriver(); // Create an object of ChromeDriver class.

}

@Test(priority = 2)

public void getURL()

{

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

}

@Test(priority = 3)

public void getTitle()

{

String title = driver.getTitle();

System.out.println(title);

}

@Test(priority = 4)

public void closeBrowser()

{

driver.close();

System.out.println("Test successfully passed");

}

}

**Output:**

Running Firefox

Google

Test successfully passed

**How to run the test multiple times using invocationCount**

@Test

public void PaymentDetails(){

System.out.println("Payment details validation is successful”);

}

@Test(invocationCount=5)

public void LoginAdmin(){

System.out.println("Login in admin is successful”);

}

@Test

public void LeaseDetails(){

System.out.println("Lease details verification is successful”);

}

**Pass parameters to test script**

package parameters;

import org.testng.annotations.Test;

import org.testng.AssertJUnit;

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 NoParameterWithTestNGXML {

String driverPath = "C:\\geckodriver.exe";

WebDriver driver;

@Test

public void testNoParameter() throws InterruptedException{

String author = "guru99";

String searchKey = "india";

System.setProperty("webdriver.gecko.driver", driverPath);

driver= new FirefoxDriver();

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

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

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

//Searching text in google text box

searchText.sendKeys(searchKey);

System.out.println("Welcome ->"+author+" Your search key is->"+searchKey);

System.out.println("Thread will sleep now");

Thread.sleep(3000);

System.out.println("Value in Google Search Box = "+searchText.getAttribute("value") +" ::: Value given by input = "+searchKey);

//verifying the value in google search box

AssertJUnit.assertTrue(searchText.getAttribute("value").equalsIgnoreCase(searchKey));

}

}

**Create data driven framework using DataProvider**

import org.openqa.selenium.By;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

public class ExcelExample{

@Test(dataProvider="testdata")

public void demoClass(String username, String password) throws InterruptedException {

System.setProperty("webdriver.chrome.driver", "Path of Chrome Driver");

Webdriver driver = new ChromeDriver();

driver.get("<a href="https://www.browserstack.com/users/sign\_in</a>");

driver.findElement(By.name("user[login]")).sendKeys(username);

driver.findElement(By.name("user[password]")).sendKeys(password);

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

Thread.sleep(5000);

Assert.assertTrue(driver.getTitle().matches("BrowserStack Login | Sign Into The Best Mobile & Browser Testing Tool"), "Invalid credentials");

System.out.println("Login successful");

}

@AfterMethod

void ProgramTermination() {

driver.quit();

}

@DataProvider(name="testdata")

public Object[][] testDataExample(){

ReadExcelFile configuration = new ReadExcelFile("Path\_of\_Your\_Excel\_File");

int rows = configuration.getRowCount(0);

Object[][]signin\_credentials = new Object[rows][2];

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

{

signin\_credentials[i][0] = config.getData(0, i, 0);

signin\_credentials[i][1] = config.getData(0, i, 1);

}

return signin\_credentials;

}

}

**How to group test cases**

public class Test1

{

@Test(groups = { "group1", "group2" })

public void test\_method1()

{

//Test implementation

}

@Test(groups = {"group2"} )

public void test\_method2()

{

//Test implementation

}

@Test(groups = {"group1"})

public void test\_method3()

{

//Test implementation

}

}

**Running test cases in parallel**

import org.testng.annotations.AfterClass;

import org.testng.annotations.Test;

import org.testng.annotations.Test;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.testng.annotations.Test;

public class ParallelTestWithMultiThread {

WebDriver driver;

@Test()

public void testOnChromeWithBrowserStackUrl()

{

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

driver=new ChromeDriver();

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

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

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

System.out.println("this is the test related to chrome browserstack homepage"+ " " +Thread.currentThread().getId());

}

@Test()

public void testOnChromeWithBrowserStackSignUp()

{

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

driver=new FirefoxDriver();

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

driver.get("https://www.browserstack.com/users/sign\_up");

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

driver.findElement(By.id("user\_full\_name")).sendKeys("Sadhvi Singh");

driver.findElement(By.id("user\_email\_login")).sendKeys("sadhvisingh24@gmail.com");

driver.findElement(By.id("user\_password")).sendKeys("browserstack");

System.out.println("this is the test related to chrome browserstack login"+ " " +Thread.currentThread().getId());

}

@AfterClass

public void close()

{

driver.quit();

}

}