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

JavascriptExecutor executor = (JavascriptExecutor)driver;

executor.executeScript("arguments[0].click();", element);

@Test

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

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

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

"E:\\MyWorkspace\_Completed\\FreeCRMTest\\Software\\chromedriver.exe");

//WebDriver driver = new FirefoxDriver();

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

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

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

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

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

highLighterMethod(driver,ele);

//ele.sendKeys("SoftwareTestingMaterial.com");

//ALTERNATIVE METHOD TO ENTER TEXT OTHER THAN SENDKEYS

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("arguments[0].value='SoftwareTestingMaterial.com';", ele);

}

----------------------------------------------------------------------

//Creating a custom function

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

JavascriptExecutor js = (JavascriptExecutor) driver;

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

}

How to post facebook status message

https://softwareautomata.wordpress.com/2017/07/16/facebook-post-selenium-webdriver-with-maven-and-testng/

**Selenium Generic Methods (Framework Independent)**

We all know, It’s always good to have reusable methods in your framework. Creating methods for every action to be performed on a webpage is not so easy while automating using Selenium. In order to suppress, we can create a class where we could store all the methods which are useful for your project. So I have created a Class ‘Utils’ under ‘utilty’ package & in this class, i’m gonna create static methods (creating an object is not required) and i’ll call these methods wherever i need them.This would be so simple to call (Syntax: ClassName.MethodName) these reusable methods.I just created these methods to make you understand, method names can be changed though.**You can get the code by clicking**[**HERE.**](https://drive.google.com/file/d/0B9cSuPte-XMwZWtMOHhWdklHYkE/view?usp=sharing)

[](https://4.bp.blogspot.com/-2aNxJiDjPF4/V1LyfZZtTpI/AAAAAAAAAF8/lxLSZBqSspk6dwH-kdEB7jd9JWe6rfr7ACLcB/s1600/green-lets-gone-md.png)

package utility;  
  
import java.awt.image.BufferedImage;  
import java.io.BufferedInputStream;  
import java.io.BufferedOutputStream;  
import java.io.File;  
import java.io.FileOutputStream;  
import java.net.URL;  
import java.net.URLConnection;  
import java.util.Iterator;  
import java.util.List;  
import java.util.Set;  
import java.util.concurrent.TimeUnit;  
  
import javax.imageio.ImageIO;  
  
import org.apache.commons.io.FileUtils;  
import org.openqa.selenium.Alert;  
import org.openqa.selenium.By;  
import org.openqa.selenium.Dimension;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.Keys;  
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;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.firefox.FirefoxProfile;  
import org.openqa.selenium.ie.InternetExplorerDriver;  
import org.openqa.selenium.interactions.Action;  
import org.openqa.selenium.interactions.Actions;  
import org.openqa.selenium.interactions.internal.Coordinates;  
import org.openqa.selenium.internal.Locatable;  
import org.openqa.selenium.support.ui.ExpectedConditions;  
import org.openqa.selenium.support.ui.Select;  
import org.openqa.selenium.support.ui.WebDriverWait;  
  
public class Utils {  
public static WebDriver driver = null;  
public static String browserName = null;  
  
public static WebDriver openFirefoxBrowser() throws Exception {  
try {  
driver = new FirefoxDriver();  
Log.info(“New driver instantiated”);  
driver.manage().window().maximize();  
driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);  
Log.info(“Implicit wait applied on the driver for 10 seconds”);  
driver.get(Constant.URL);  
Log.info(“Web application launched successfully”);  
  
} catch (Exception e) {  
Log.error(“Class Utils | Method OpenBrowser | Exception desc : ”  
+ e.getMessage());  
}  
return driver;  
}  
  
public static String getBrowserName() {  
browserName = System.getProperty(“browser”);  
  
if (browserName == null)  
browserName = “ie”;  
return browserName;  
}  
  
public static WebDriver createWebDriver(String browser) {  
System.out.println(“Browser: ” + browser);  
  
switch (browser.toLowerCase()) {  
case “ff”:  
case “firefox”:  
driver = new FirefoxDriver();  
break;  
  
case “ch”:  
case “chrome”:  
System.setProperty(“webdriver.chrome.driver”,  
“E://chromedriver.exe”);  
driver = new ChromeDriver();  
break;  
  
case “ie”:  
case “internetexplorer”:  
driver = new InternetExplorerDriver();  
break;  
  
default:  
System.out.println(“Invalid browser name ” + browser);  
System.exit(0);  
break;  
}// switch  
  
driver.manage().deleteAllCookies();  
driver.manage().window().maximize();  
driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);  
driver.manage().timeouts().setScriptTimeout(45, TimeUnit.SECONDS);  
  
return driver;  
}  
  
public static void switchToNewWindow() {  
Set s = driver.getWindowHandles();  
Iterator itr = s.iterator();  
String w1 = (String) itr.next();  
String w2 = (String) itr.next();  
driver.switchTo().window(w2);  
}  
  
public static void switchToOldWindow() {  
Set s = driver.getWindowHandles();  
Iterator itr = s.iterator();  
String w1 = (String) itr.next();  
String w2 = (String) itr.next();  
driver.switchTo().window(w1);  
}  
  
public static void switchToParentWindow() {  
driver.switchTo().defaultContent();  
}  
  
public static WebDriver openIEBrowser() throws Exception {  
try {  
System.setProperty(“webdriver.ie.driver”, “D:\\IEDriverServer.exe”);  
driver = new InternetExplorerDriver();  
Log.info(“New driver instantiated”);  
driver.manage().window().maximize();  
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);  
Log.info(“Implicit wait applied on the driver for 10 seconds”);  
driver.get(Constant.URL);  
driver.navigate()  
.to(“javascript:document.getElementById(‘overridelink’).click()”);  
Log.info(“Web application launched successfully”);  
  
} catch (Exception e) {  
Log.error(“Class Utils | Method OpenBrowser | Exception desc : ”  
+ e.getMessage());  
}  
return driver;  
}  
  
/\*  
\* public static String getMethodName() {  
\*  
\* String methodName = Thread.currentThread().getStackTrace()[1]  
\* .getMethodName(); System.out.println(methodName);  
\*  
\* return methodName; }  
\*/  
  
public static void waitForElement(WebElement element) {  
  
WebDriverWait wait = new WebDriverWait(driver, 10);  
wait.until(ExpectedConditions.elementToBeClickable(element));  
}  
  
public static void waitTillElementFound(WebElement ElementTobeFound,  
int seconds) {  
WebDriverWait wait = new WebDriverWait(driver, seconds);  
wait.until(ExpectedConditions.visibilityOf(ElementTobeFound));  
}  
  
public static void takeScreenshotOfWebelement(WebDriver driver,  
WebElement element, String Destination) throws Exception {  
File v = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);  
BufferedImage bi = ImageIO.read(v);  
org.openqa.selenium.Point p = element.getLocation();  
int n = element.getSize().getWidth();  
int m = element.getSize().getHeight();  
BufferedImage d = bi.getSubimage(p.getX(), p.getY(), n, m);  
ImageIO.write(d, “png”, v);  
  
FileUtils.copyFile(v, new File(Destination));  
}  
  
public static void setWindowSize(int Dimension1, int dimension2) {  
driver.manage().window().setSize(new Dimension(Dimension1, dimension2));  
  
}  
  
public static void takeScreenshotMethod(String Destination)  
throws Exception {  
File f = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);  
FileUtils.copyFile(f, new File(Destination));  
}  
  
public static void pressKeyDown(WebElement element) {  
element.sendKeys(Keys.DOWN);  
}  
  
public void pressKeyEnter(WebElement element) {  
element.sendKeys(Keys.ENTER);  
}  
  
public static void pressKeyUp(WebElement element) {  
element.sendKeys(Keys.UP);  
}  
  
public static void moveToTab(WebElement element) {  
element.sendKeys(Keys.chord(Keys.ALT, Keys.TAB));  
}  
  
public static void handleHTTPS\_IEbrowser() {  
driver.navigate().to(  
“javascript:document.getElementById(‘overridelink’).click()”);  
}  
  
public static void handleHTTPS\_Firefox() {  
FirefoxProfile profile = new FirefoxProfile();  
profile.setAcceptUntrustedCertificates(false);  
driver = new FirefoxDriver(profile);  
}  
  
public static void waitTillPageLoad(int i) {  
  
driver.manage().timeouts().pageLoadTimeout(i, TimeUnit.SECONDS);  
  
}  
  
public static void clickAllLinksInPage(String destinationOfScreenshot)  
throws Exception {  
  
List<WebElement> Links = driver.findElements(By.tagName(“a”));  
System.out.println(“Total number of links :” + Links.size());  
  
for (int p = 0; p < Links.size(); p++) {  
System.out.println(“Elements present the body :”  
+ Links.get(p).getText());  
Links.get(p).click();  
Thread.sleep(3000);  
System.out.println(“Url of the page ” + p + “)”  
+ driver.getCurrentUrl());  
takeScreenshotMethod(destinationOfScreenshot + p);  
navigate\_back();  
Thread.sleep(2000);  
}  
  
}  
  
public static void keyboardEvents(WebElement webelement, Keys key,  
String alphabet) {  
webelement.sendKeys(Keys.chord(key, alphabet));  
  
}  
  
public static void navigate\_forward() {  
driver.navigate().forward();  
}  
  
public static void navigate\_back() {  
driver.navigate().back();  
}  
  
public static void refresh() {  
driver.navigate().refresh();  
}  
  
public static void waitMyTime(int i) {  
driver.manage().timeouts().implicitlyWait(i, TimeUnit.SECONDS);  
  
}  
  
public static void clearTextField(WebElement element) {  
element.clear();  
  
}  
  
public static void clickWebelement(WebElement element) {  
try {  
boolean elementIsClickable = element.isEnabled();  
while (elementIsClickable) {  
element.click();  
}  
  
} catch (Exception e) {  
System.out.println(“Element is not enabled”);  
e.printStackTrace();  
}  
}  
  
public static void clickMultipleElements(WebElement someElement,  
WebElement someOtherElement) {  
Actions builder = new Actions(driver);  
builder.keyDown(Keys.CONTROL).click(someElement)  
.click(someOtherElement).keyUp(Keys.CONTROL).build().perform();  
}  
  
public static void highlightelement(WebElement element) {  
for (int i = 0; i < 4; i++) {  
JavascriptExecutor js = (JavascriptExecutor) driver;  
js.executeScript(  
“arguments[0].setAttribute(‘style’, arguments[1]);”,  
element, “color: solid red; border: 6px solid yellow;”);  
js.executeScript(  
“arguments[0].setAttribute(‘style’, arguments[1]);”,  
element, “”);  
  
}  
  
}  
  
public static boolean checkAlert\_Accept() {  
try {  
Alert a = driver.switchTo().alert();  
String str = a.getText();  
System.out.println(str);  
  
a.accept();  
return true;  
  
} catch (Exception e) {  
  
System.out.println(“no alert “);  
return false;  
  
}  
}  
  
public static boolean checkAlert\_Dismiss() {  
try {  
Alert a = driver.switchTo().alert();  
String str = a.getText();  
System.out.println(str);  
  
a.dismiss();  
return true;  
  
} catch (Exception e) {  
  
System.out.println(“no alert “);  
return false;  
  
}  
}  
  
public static void scrolltoElement(WebElement ScrolltoThisElement) {  
Coordinates coordinate = ((Locatable) ScrolltoThisElement)  
.getCoordinates();  
coordinate.onPage();  
coordinate.inViewPort();  
}  
  
public static void checkbox\_Checking(WebElement checkbox) {  
boolean checkstatus;  
checkstatus = checkbox.isSelected();  
if (checkstatus == true) {  
System.out.println(“Checkbox is already checked”);  
} else {  
checkbox.click();  
System.out.println(“Checked the checkbox”);  
}  
}  
  
public static void radiobutton\_Select(WebElement Radio) {  
boolean checkstatus;  
checkstatus = Radio.isSelected();  
if (checkstatus == true) {  
System.out.println(“RadioButton is already checked”);  
} else {  
Radio.click();  
System.out.println(“Selected the Radiobutton”);  
}  
}  
  
// Unchecking  
public static void checkbox\_Unchecking(WebElement checkbox) {  
boolean checkstatus;  
checkstatus = checkbox.isSelected();  
if (checkstatus == true) {  
checkbox.click();  
System.out.println(“Checkbox is unchecked”);  
} else {  
System.out.println(“Checkbox is already unchecked”);  
}  
}  
  
public static void radioButton\_Deselect(WebElement Radio) {  
boolean checkstatus;  
checkstatus = Radio.isSelected();  
if (checkstatus == true) {  
Radio.click();  
System.out.println(“Radio Button is deselected”);  
} else {  
System.out.println(“Radio Button was already Deselected”);  
}  
}  
  
public static void dragAndDrop(WebElement fromWebElement,  
WebElement toWebElement) {  
Actions builder = new Actions(driver);  
builder.dragAndDrop(fromWebElement, toWebElement);  
}  
  
public static void dragAndDrop\_Method2(WebElement fromWebElement,  
WebElement toWebElement) {  
Actions builder = new Actions(driver);  
Action dragAndDrop = builder.clickAndHold(fromWebElement)  
.moveToElement(toWebElement).release(toWebElement).build();  
dragAndDrop.perform();  
}  
  
public static void dragAndDrop\_Method3(WebElement fromWebElement,  
WebElement toWebElement) throws InterruptedException {  
Actions builder = new Actions(driver);  
builder.clickAndHold(fromWebElement).moveToElement(toWebElement)  
.perform();  
Thread.sleep(2000);  
builder.release(toWebElement).build().perform();  
}  
  
public static void hoverWebelement(WebElement HovertoWebElement)  
throws InterruptedException {  
Actions builder = new Actions(driver);  
builder.moveToElement(HovertoWebElement).perform();  
Thread.sleep(2000);  
  
}  
  
public static void doubleClickWebelement(WebElement doubleclickonWebElement)  
throws InterruptedException {  
Actions builder = new Actions(driver);  
builder.doubleClick(doubleclickonWebElement).perform();  
Thread.sleep(2000);  
  
}  
  
public static String getToolTip(WebElement toolTipofWebElement)  
throws InterruptedException {  
String tooltip = toolTipofWebElement.getAttribute(“title”);  
System.out.println(“Tool text : ” + tooltip);  
return tooltip;  
}  
  
public static void selectElementByNameMethod(WebElement element, String Name) {  
Select selectitem = new Select(element);  
selectitem.selectByVisibleText(Name);  
}  
  
public static void selectElementByValueMethod(WebElement element,  
String value) {  
Select selectitem = new Select(element);  
selectitem.selectByValue(value);  
}  
  
public static void selectElementByIndexMethod(WebElement element, int index) {  
Select selectitem = new Select(element);  
selectitem.selectByIndex(index);  
}  
  
public static void clickCheckboxFromList(String xpathOfElement,  
String valueToSelect) {  
  
List<WebElement> lst = driver.findElements(By.xpath(xpathOfElement));  
for (int i = 0; i < lst.size(); i++) {  
List<WebElement> dr = lst.get(i).findElements(By.tagName(“label”));  
for (WebElement f : dr) {  
System.out.println(“value in the list : ” + f.getText());  
if (valueToSelect.equals(f.getText())) {  
f.click();  
break;  
}  
}  
}  
}  
  
public static void downloadFile(String href, String fileName)  
throws Exception {  
URL url = null;  
URLConnection con = null;  
int i;  
url = new URL(href);  
con = url.openConnection();  
File file = new File(“.//OutputData//” + fileName);  
BufferedInputStream bis = new BufferedInputStream(con.getInputStream());  
BufferedOutputStream bos = new BufferedOutputStream(  
new FileOutputStream(file));  
while ((i = bis.read()) != -1) {  
bos.write(i);  
}  
bos.flush();  
bis.close();  
}  
  
public static void navigateToEveryLinkInPage() throws InterruptedException {  
  
List<WebElement> linksize = driver.findElements(By.tagName(“a”));  
int linksCount = linksize.size();  
System.out.println(“Total no of links Available: ” + linksCount);  
String[] links = new String[linksCount];  
System.out.println(“List of links Available: “);  
// print all the links from webpage  
for (int i = 0; i < linksCount; i++) {  
links[i] = linksize.get(i).getAttribute(“href”);  
System.out.println(linksize.get(i).getAttribute(“href”));  
}  
// navigate to each Link on the webpage  
for (int i = 0; i < linksCount; i++) {  
driver.navigate().to(links[i]);  
Thread.sleep(3000);  
System.out.println(driver.getTitle());  
}  
}  
}  
  
Now you can call these methods wherever and whenever you need, you can also modify as per your need.

I need to work with generic "page transition" methods like the click()  
method that takes the button name and returns the appropriate page? In my  
mind, this is what I would like to do:  
  
// page object  
public class MainPage() extends PageBase() {  
@FindBy( id="button1" ) WebElement button1;  
@FindBy( id="button2" ) WebElement button2;  
@FindBy( id="button3" ) WebElement button3;  
...  
public WhatTypeOfPageShouldThisBe click( String buttonName ) {  
If(buttonName.equals( "button1" )) {  
button1.click();  
Page1 page = new Page1( driver );  
}  
If(buttonName.equals( "button2" )) {  
button2.click();  
Page2 page = new Page2( driver );  
}  
If(buttonName.equals( "button3" )) {  
button3.click();  
Page3 page = new Page3( driver );  
}  
return page;  
}  
  
// This would fail if the page returned is not the expected one.  
@Test  
public class Test extends TestBase() {  
Page1 page1;  
Page2 page2;  
Page3 page3;  
  
page1 = Main.click( "button1" );  
...  
page2 = Main.click( "button2" );  
...  
page3 = Main.click( "button3" );  
}  
  
I would've rather used individual methods like clickButton1(),  
clickbutton2(), etc, to avoid this problem and others but the team feels  
that the page object will be cluttered with so many individual methods.  
(Personally, I think that individual methods take longer to create at first  
but makes debugging and maintenance much easier and targeted, but that's a  
story for another day. I have to pick and choose my battles. I gave in on  
this one.)  
  
How can I make the WhatTypeOfPageShouldThisBe class morph into different  
types of page based on the button clicked? Am I going the wrong way  
altogher?

# **Useful Functions for Automation Framework using Selenium Web Driver**

* [AUTOMATION TESTING](https://www.tothenew.com/blog/category/technology/automation-testing-testing-2/), [TESTING](https://www.tothenew.com/blog/category/technology/testing-2/)

28 / MAY / 2015 BY [SUMIT GAMBHIR](https://www.tothenew.com/blog/author/sumit-gambhir/) [2 COMMENTS](https://www.tothenew.com/blog/useful-functions-for-automation-framework-using-selenium-web-driver/#comments)

Share this blog

EmailTwitterFacebookLinkedIn

While doing automation using selenium, there are certain repetetive tasks that we need to perform in order to handle page elements. So, the aim of this blog is to highlight some common operations that we need to handle every now and then.

**Some operations are:**  
1. Open URL in different browsers (Chrome, Firefox, IE)  
2. Do mouse-hovers  
3. Taking snapshots for analysing  the failures  
4. Getting current time-stamp  
5. Handling a drop-down

**1. Open URL in different browsers (Chrome, Firefox, IE)**

Do the following steps first, to use this method:  
1. Download ‘chromedriver.exe’ and ‘IEDriverServer.exe’ from  
“http://www.seleniumhq.org/download/”  
2. Then save the downloaded file in ‘Drivers’ folder at path where ‘src’ folder exists

call ‘OpenApp’ function with passing two parameter: ‘BrowserName(ie. CH/FF/IE) & URL’  
e.g. OpenApp(“CH”,”http://www.seleniumhq.org/”) it will open this url in ChromeBrowser

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | public static WebDriver OpenApp(String BrowserName, String url){  fn\_LaunchBrowser(BrowserName);  fn\_OpenURL(url);  return driver;  }  public static void fn\_OpenURL(String url){  driver.get(url);  driver.manage().window().maximize();  }    public static WebDriver fn\_LaunchBrowser(String browsername){  if(browsername=="CH"){  System.setProperty("webdriver.chrome.driver", "Drivers\\chromedriver.exe");  driver= new ChromeDriver();  }else if(browsername=="FF"){  driver= new FirefoxDriver();  }else if(browsername=="IE"){  System.setProperty("webdriver.ie.driver", "Drivers\\IEDriverServer.exe");  driver= new InternetExplorerDriver();  }  driver.manage().timeouts().implicitlyWait(80, TimeUnit.SECONDS);  return driver;  } |

**2. Do mouse-hovers**

Incase, we need to click on a sub-menu that is visible only when users do mouse-hover on the main-menu, then we can do it using this function. Just pass web element position to this function.  
e.g MouseOver(driver.findElement(By.name(“Main-Menu”)))

|  |  |
| --- | --- |
| 1  2  3  4 | public static void MouseOver(WebElement we){  Actions actObj=new Actions(driver);  actObj.moveToElement(we).build().perform();  } |

**3. Take Snapshot for analysing the failures**

Scenario – As an automation tester, we must need to take snapshots of error & exceptions while running the script as a proof, so we can use this method.  
e.g. fn\_TakeSnapshot(driver, “where you want to save the SnapShot”)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | public static String fn\_TakeSnapshot(WebDriver driver, String DestFilePath) throws IOException{  String TS=fn\_GetTimeStamp();  TakesScreenshot tss=(TakesScreenshot) driver;  File srcfileObj= tss.getScreenshotAs(OutputType.FILE);  DestFilePath=DestFilePath+TS+".png";  File DestFileObj=new File(DestFilePath);  FileUtils.copyFile(srcfileObj, DestFileObj);  return DestFilePath;  } |

**4. Get current Time-stamp**

Scenario – While reporting your script status, you need to pass time & date that when your test-script has finished. Another scenario: If you need unique username every time, you can append the timestamp with any constant string (e.g SumitMay28201513\_15\_10 PM), so here you are:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public static String fn\_GetTimeStamp(){  DateFormat DF=DateFormat.getDateTimeInstance();  Date dte=new Date();  String DateValue=DF.format(dte);  DateValue=DateValue.replaceAll(":", "\_");  DateValue=DateValue.replaceAll(",", "");  return DateValue;  } |

**5. Handle drop-down (select an option from list)**

Scenario: If you need to select an option from any drop-down, you can use following methods  
NOTE: ‘Select’ class in selenium works only those drop-downs, which lies in <select > tag, like in the following html.

|  |  |
| --- | --- |
| 1  2  3  4  5 | <div><select id="SelectID\_Three"><option value="selectValue">Select</option>  <option value="firstcolour">Yellow</option>  <option value="secondcolour">Lime</option>  <option value="thirdcolour">Red</option>  </select></div> |

The following code will help you solve the problem:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | //select the dropdown using "select by visible text", so pass VisibleText as 'Yellow' to funtion  public static void fn\_Select(WebElement WE, String VisibleText){  Select selObj=new Select(WE);  selObj.selectByVisibleText(VisibleText);  }    //select the dropdown using "select by index", so pass IndexValue as '2'  public static void fn\_Select(WebElement WE, int IndexValue){  Select selObj=new Select(WE);  selObj.selectByIndex(IndexValue);  }    //select the dropdown using "select by value", so pass Value as 'thirdcolor'  public static void fn\_Select(WebElement WE, String Value){  Select selObj=new Select(WE);  selObj.selectByValue(Value);  } |

String name = "Charles";

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

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

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

new Actions(driver).dragAndDrop(source, target).build().perform();

# [**Selenium WebDriver – Count Total number of Web Link and All Element on Webpage using size()**](http://seleniumyourself.blogspot.com/2015/05/selenium-webdriver-count-total-number.html)

By [risi](https://www.blogger.com/profile/18431351794481673008" \o "author profile)   on [14:12](http://seleniumyourself.blogspot.com/2015/05/selenium-webdriver-count-total-number.html)

[2](http://seleniumyourself.blogspot.com/2015/05/selenium-webdriver-count-total-number.html#comment-form)

[](http://4.bp.blogspot.com/-yJp-ZCj9xjo/VWF9eaUtacI/AAAAAAAAAn4/cK-lxzx7fBY/s1600/Counting-blessings-JBC.png)

So How to get any text is available in HTML Page Source Code using Selenium WebDriver.  
  
  
Steps :

1. Define Firefox Browser and open the Firefox Browser
2. Open the URL (Website)
3. Identify the number of Links on webpage and assign into Webelement List
4. Count the total Link list on Web Page
5. Print the total count of links on webpage
6. Identify all the elements on web page
7. Count the total all element on web page
8. Print the total count of all element on webpage
9. Print all the Tag Name and Text Name on webpage.

Practice Yourself :

**package** dayOne;

**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** Count\_Total\_Weblink\_and\_AllElement\_on\_Webpage {

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

              //Define the Webdriver for Browser i.e. Firefox

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

              //Open the URL (Website)

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

        //Identify the number of Link on webpage and assign into Webelement List

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

        // Count the total Link list on Web Page

**int** linkListCount = allLinkElements.size();

        //Print the total count of links on webpage

        System.*out*.println("Total Number of link count on webpage = "  + linkListCount);

       //Identify all the elements on web page

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

       //Count the total all element on web page

       linkListCount = allElements.size();

       //Print the total count of all element on webpage

       System.*out*.println("Total Number of All Element on webpage = "  + linkListCount);

       //Print all the Tag Name and Text Name on webpage  
       int i = 0;  
        for (WebElement Element : allElements) {  
            i = i +1;  
            System.out.println(Element.getTagName());  
            System.out.println(Element.getText());  
        }

        //Close the Broswer

       driver.close();

       // Quit the selenium

       driver.quit();

       }

}