Practical No. 4. Browser command and navigation Commands in Selenium.

Date: ________________

Aim:

To learn Browser command and navigation Commands in Selenium.

Theory:

As we know, WebDriver talks to individual browsers natively. This way it has better control, not just on the web page, but on the browser itself. Navigate is one such feature of WebDriver that allows the test script developer to work with the browser's Back, Forward, and Refresh controls. Using the WebDriver's navigation feature, you can emulate those actions.

Navigation Commands

1. navigate()

The method that is used for this purpose is **navigate().**

The following is its API syntax:

WebDriver.Navigation navigate()

There is no input parameter for this method, but the return type is the WebDriver.Navigation interface, which contains all of the browser navigation options that help you navigate through your browser's history.

2. to();

the to() method is used to navigate to a web URL.

The API syntax is as follows:

void to(java.lang.String url)

The input parameter for this method is the url string that has to be loaded in the browser. This method will load the page in the browser by using the HTTP GET operation, and it will block everything else until the page is completely loaded. This method is the same as the driver.get(String url) method

3. back()

This method is used to emulate our browser's Back button.

The API syntax for this method is pretty straightforward, as follows:

void back()

This method doesn't take any input and doesn't return anything as well, but takes the browser one level back in its history.

4. **forward()**

This method is pretty much similar to the back() method, but takes the browser one level in the oppositedirection.

The API syntax for the method is as follows:

void forward()

This method doesn't take any input and doesn't return anything as well, but takes the browser one level forward in its history

5. **refresh()**

This method will reload the current URL to emulate the browser's refresh (F5 key) action.

The API syntax is as follows:

void refresh()

Browser Commands

The very basic browser operations of WebDriver include opening a browser; perform few tasks and then closing the browser.

Given are some of the most commonly used Browser commands for Selenium WebDriver.

1. **void get(String arg0)**

   In WebDriver, this method loads a new web page in the existing browser window. It accepts String as parameter and returns void.

   The respective command to load a new web page can be written as:

```
driver.get(URL);
      // Or can be written as
 String URL = "URL";
driver.get(URL);
```

2. **String getTitle();**

   In WebDriver, this method fetches the title of the current web page. It accepts no parameter and returns a String.

   The respective command to fetch the title of the current page can be written as:

```
driver.getTitle();
     // Or can be written as
String Title = driver.getTitle();
```

3. **getCurrentUrl(): String**

In WebDriver, this method fetches the string representing the Current URL of the current web page. It accepts nothing as parameter and returns a String value.

The respective command to fetch the string representing the current URL can be written as:

```
driver.getCurrentUrl();
   //Or can be written as
 String CurrentUrl = driver.getCurrentUrl();
```

4. **getPageSource(): String**

   In WebDriver, this method returns the source code of the current web page loaded on the current browser. It accepts nothing as parameter and returns a String value.

   The respective command to get the source code of the current web page can be written as:

```
driver.getPageSource();
   //Or can be written as
   String PageSource = driver.getPageSource();
```

5. **close(): void**

   This method terminates the current browser window operating by WebDriver at the current time. If the current window is the only window operating by WebDriver, it terminates the browser as well. This method accepts nothing as parameter and returns void.

   The respective command to terminate the browser window can be written as:

```
driver.close();
```

6. **quit(): void**

   This method terminates all windows operating by WebDriver. It terminates all tabs as well as the browser itself. It accepts nothing as parameter and returns void.

   The respective command to terminate all windows can be written as:

```
driver.quit();
```

Implementation

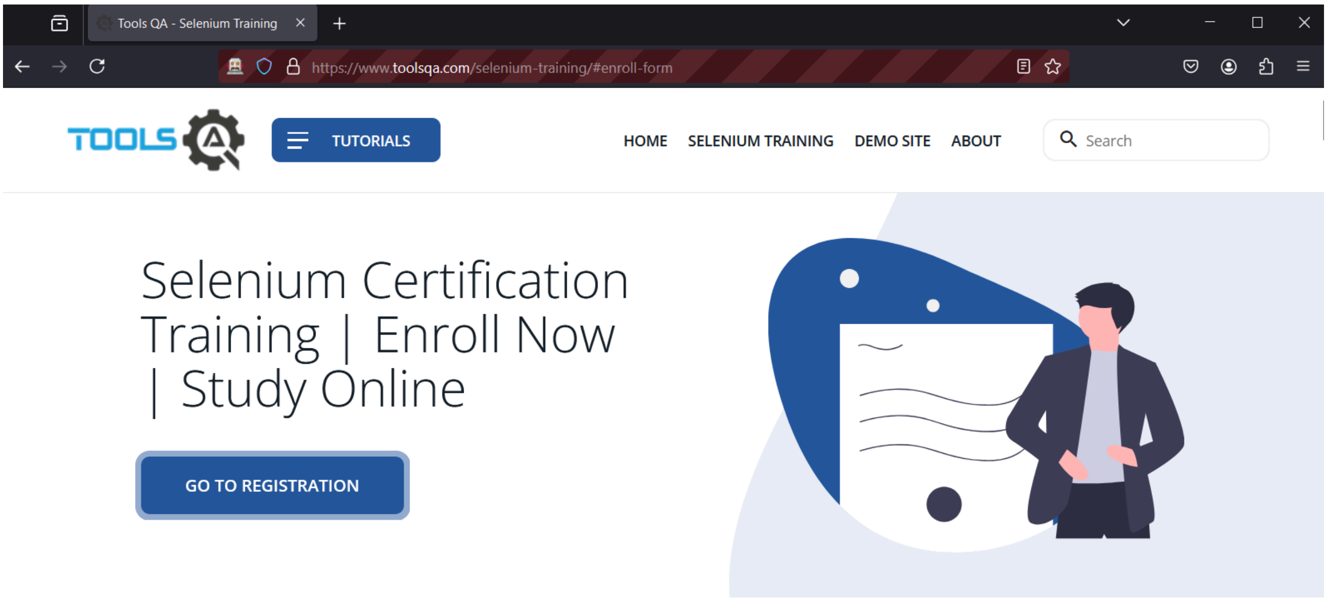
1. Write a selenium script to navigate to https://www.toolsqa.com/selenium-training/ website and perform following actions-1)Click on Registration button link,2) Come back to Home page (Use 'Back' command), 3)Again go back to Registration page (This time use 'Forward' command), 4) Again come back to Home page (This time use 'To' command), 5) Refresh the Browser (Use 'Refresh' command).

   Code:

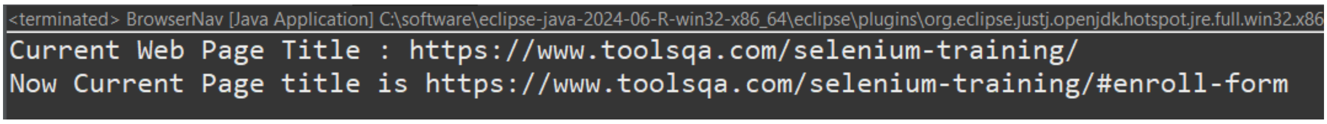
```
package pkg;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class BrowserNav {
    public static void main(String[] args) throws InterruptedException {
    // TODO Auto-generated method stub
    System.setProperty("webdriver.gecko.driver","G:\\Selenium_setup\\geckodriver.exe");

    // Web driver for FireFox
    WebDriver driver = new FirefoxDriver();

     // Navigate to the ToolsQA Selenium Training website
      driver.get("https://www.toolsqa.com/selenium-training/");

      // Display Title of WebPage
    System.out.println("Current Web Page Title : " + driver.getCurrentUrl());

      // Click on the Registration button link
      driver.findElement(By.className("btn-primary-shadow")).click();

    // navigate
    driver.navigate().to(driver.getCurrentUrl()+"");
    System.out.println("Now Current Page title is " + driver.getCurrentUrl());
```

```
        // Wait for a few seconds to see the registration page
        Thread.sleep(2000);

        // Come back to Home page using 'Back' command
        driver.navigate().back();

        // Wait for a few seconds to see the home page
        Thread.sleep(2000);

        // Again go back to Registration page using 'Forward' command
        driver.navigate().forward();

        // Wait for a few seconds to see the home page
        Thread.sleep(2000);

        // Refresh the Browser
        driver.navigate().refresh();
      driver.quit();
      }
}
```

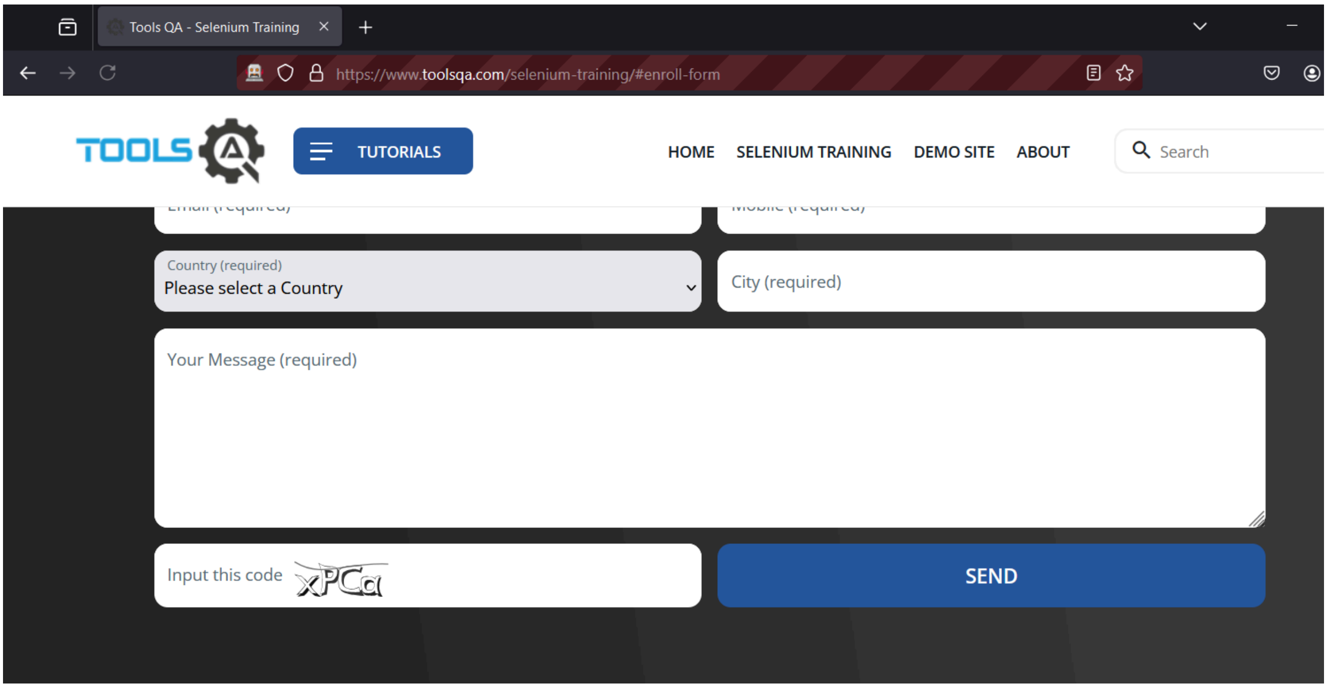
Output:



Tools QA - Selenium Training
https://www.toolsqa.com/selenium-training/#enroll-form
TOOLS QA
TUTORIALS
HOME
SELENIUM TRAINING
DEMO SITE
ABOUT
Search
Selenium Certification Training | Enroll Now | Study Online
GO TO REGISTRATION



```
<terminated> BrowserNav [Java Application] C:\software\eclipse-java-2024-06-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86
Current Web Page Title : https://www.toolsqa.com/selenium-training/
Now Current Page title is https://www.toolsqa.com/selenium-training/#enroll-form
```



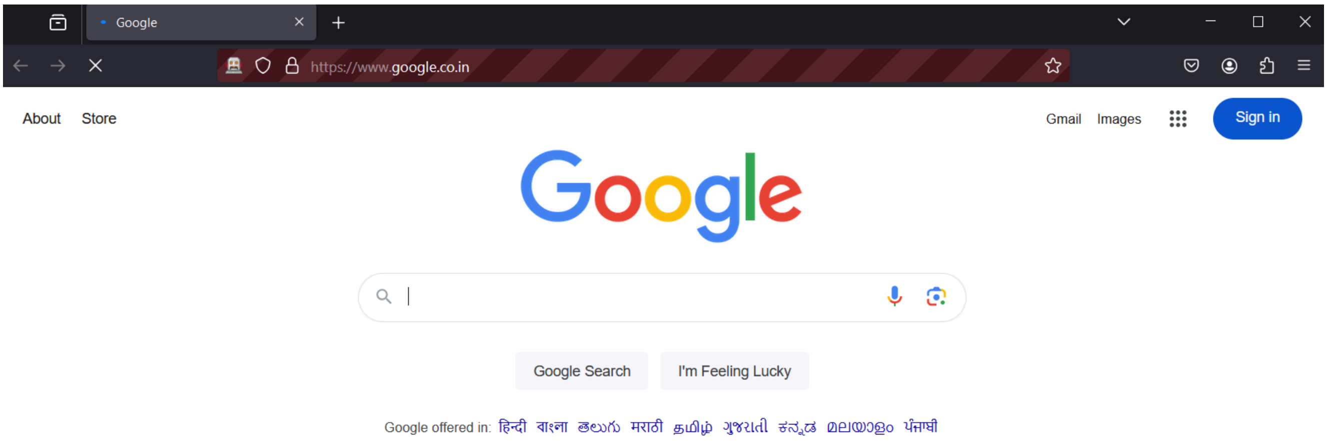
2. Automate the following test scenarios:
   a) Invoke firefox Browser
   b) Open URL: https://www.google.co.in/
   c) Get Page Title name and Title length
   d) Print Page Title and Title length on the Eclipse Console
   e) Get page URL and verify whether it is the desired page or not
   f) Get page Source and Page Source length
   g) Print page Length on Eclipse Console.
   h) Close the Browser

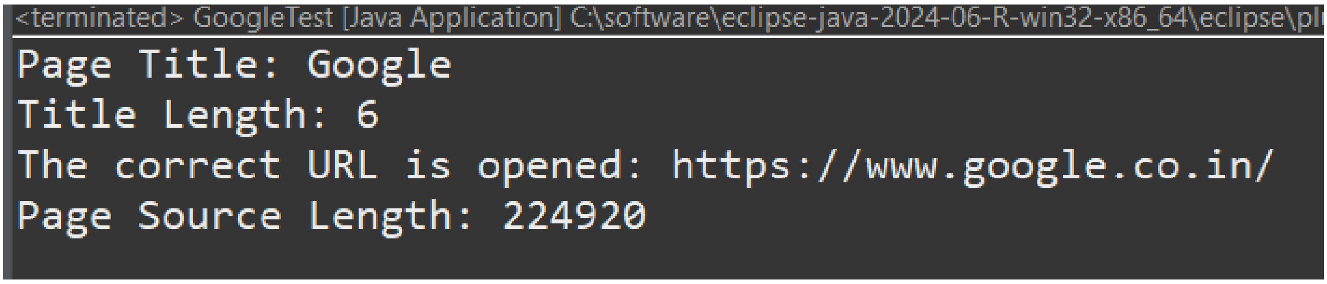
Code:

```
package pkg;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class GoogleTest {
  public static void main(String[] args) {
    // Set the path of the geckodriver (change the path as per your system)
      System.setProperty("webdriver.gecko.driver","G:\\Selenium_setup\\geckodriver.exe");
```

```
        // Create a new instance of the Firefox driver
        WebDriver driver = new FirefoxDriver();

        // Launch the Google website
        driver.get("https://www.google.co.in/");

        // Get and print the Page Title
        String title = driver.getTitle();
        System.out.println("Page Title: " + title);

        // Get and print the Page Title length
        int titleLength = title.length();
        System.out.println("Title Length: " + titleLength);

        // Get and verify the current URL
        String currentUrl = driver.getCurrentUrl();
        if (currentUrl.equals("https://www.google.co.in/")) {
            System.out.println("The correct URL is opened: " + currentUrl);
        } else {
            System.out.println("An incorrect URL is opened: " + currentUrl);
        }
        // Get and print the Page Source length
        String pageSource = driver.getPageSource();
        int pageSourceLength = pageSource.length();
        System.out.println("Page Source Length: " + pageSourceLength);
        // Close the browser
        driver.close();
    }
}
```

Output:





```
<terminated> GoogleTest [Java Application] C:\software\eclipse-java-2024-06-R-win32-x86_64\eclipse\pl
Page Title: Google
Title Length: 6
The correct URL is opened: https://www.google.co.in/
Page Source Length: 224920
```

Conclusion: Learnt to use Browser command and navigation Commands in Selenium

After performing this Practical/lab, students are expected to answer following questions

Q.1 What are different browser commands?

Q.2 What are different browser Navigation Commands?