# **First Selenium Webdriver Script: JAVA Code Example**

Using the[Java](https://www.guru99.com/java-tutorial.html)class "myclass"  that we created in the previous tutorial, let us try to create a WebDriver script that would:

1. fetch Mercury Tours' homepage
2. verify its title
3. print out the result of the comparison
4. close it before ending the entire program.

## **WebDriver Code**

Below is the actual WebDriver code for the logic presented by the scenario above

Note: Starting Firefox 35, you need to use gecko driver created by Mozilla to use Web Driver. Selenium 3.0, gecko and firefox has compatibility issues and setting them correctly could become an uphill task. If the code does not work, downgrade to Firefox version 47 or below. Alternatively, you can run your scripts on Chrome. Selenium works out of the box for Chrome. You just need to change 3 lines of code to make your script work with Chrome or Firefox

package newproject;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

//comment the above line and uncomment below line to use Chrome

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

public class PG1 {

public static void main(String[] args) {

// declaration and instantiation of objects/variables

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

WebDriver driver = new FirefoxDriver();

//comment the above 2 lines and uncomment below 2 lines to use Chrome

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

//WebDriver driver = new ChromeDriver();

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

String expectedTitle = "Welcome: Mercury Tours";

String actualTitle = "";

// launch Fire fox and direct it to the Base URL

driver.get(baseUrl);

// get the actual value of the title

actualTitle = driver.getTitle();

/\*

\* compare the actual title of the page with the expected one and print

\* the result as "Passed" or "Failed"

\*/

if (actualTitle.contentEquals(expectedTitle)){

System.out.println("Test Passed!");

} else {

System.out.println("Test Failed");

}

//close Fire fox

driver.close();

}

}

## **Explaining the code**

## **Importing Packages**

To get started, you need to import following two packages:

1. ****org.openqa.selenium.\*****- contains the WebDriver class needed to instantiate a new browser loaded with a specific driver
2. ****org.openqa.selenium.firefox.FirefoxDriver****- contains the FirefoxDriver class needed to instantiate a Firefox-specific driver onto the browser instantiated by the WebDriver class

If your test needs more complicated actions such as accessing another class, taking browser screenshots, or manipulating external files, definitely you will need to import more packages.

## **Instantiating objects and variables**

Normally, this is how a driver object is instantiated.

[Creating your First Script in Webdriver](https://www.guru99.com/images/image004(2).png)

A FirefoxDriver class with no parameters means that the default Firefox profile will be launched by our Java program. The default Firefox profile is similar to launching Firefox in safe mode (no extensions are loaded).

For convenience, we saved the Base URL and the expected title as variables.

## **Launching a Browser Session**

WebDriver's ****get()**** method is used to launch a new browser session and directs it to the URL that you specify as its parameter.

[Creating your First Script in Webdriver](https://www.guru99.com/images/image005(2).png)

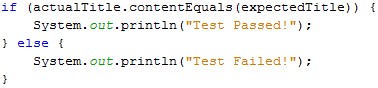
## **Get the Actual Page Title**

The WebDriver class has the ****getTitle()**** method that is always used to obtain the page title of the currently loaded page.

[Creating your First Script in Webdriver](https://www.guru99.com/images/image006(2).png)

## **Compare the Expected and Actual Values**

This portion of the code simply uses a basic Java if-else structure to compare the actual title with the expected one.

[](https://www.guru99.com/images/image007(2).png)

## **Terminating a Browser Session**

The "****close()****" method is used to close the browser window.

[Creating your First Script in Webdriver](https://www.guru99.com/images/image008(3).png)

## **Terminating the Entire Program**

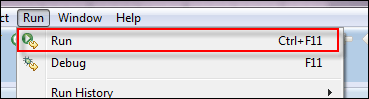
If you use this command without closing all browser windows first, your whole Java program will end while leaving the browser window open.

[Creating your First Script in Webdriver](https://www.guru99.com/images/image009(3).png)

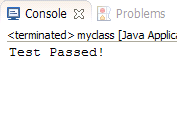
## **Running the Test**

There are two ways to execute code in Eclipse IDE.

1. On Eclipse's menu bar, click ****Run > Run.****
2. Press ****Ctrl+F11**** to run the entire code.

[](https://www.guru99.com/images/image010(2).png)

 If you did everything correctly, Eclipse would output "Test Passed!"

[](https://www.guru99.com/images/image011(2).png)