Contents

Selenium

Selenium is one of the most widely used open source Web UI automation testing tools. It supports automation testing of websites across different browsers such as Edge, Chrome, Safari and Firefox etc. It supports various platforms and programming languages. User can automate functional tests and easily integrates them with Maven, Jenkins and other build automation and continuous integration tools.

Selenium Suite Components

Selenium IDE

It is an add-on to Firefox browser that provides record and playback capability. It has limited features and test scripts generated are not robust and portable.

Selenium RC

Selenium Remote Control (RC) is officially deprecated by Selenium. It uses a dedicated server for running automation scripts.

Selenium WebDriver

It provides different drivers for different browsers and support different programming languages.

Selenium Grid

It helps in running selenium test scripts in parallel mode across multiple remote machines.

Why Selenium

1. Open Source
2. Scripting can be done in many programming languages like C#, Java, Python, Perl and Ruby etc.
3. Supports all popular browsers like Edge, Firefox, Chrome, IE, Opera and Safari etc.
4. Selenium grid helps in parallel and distributed test execution.

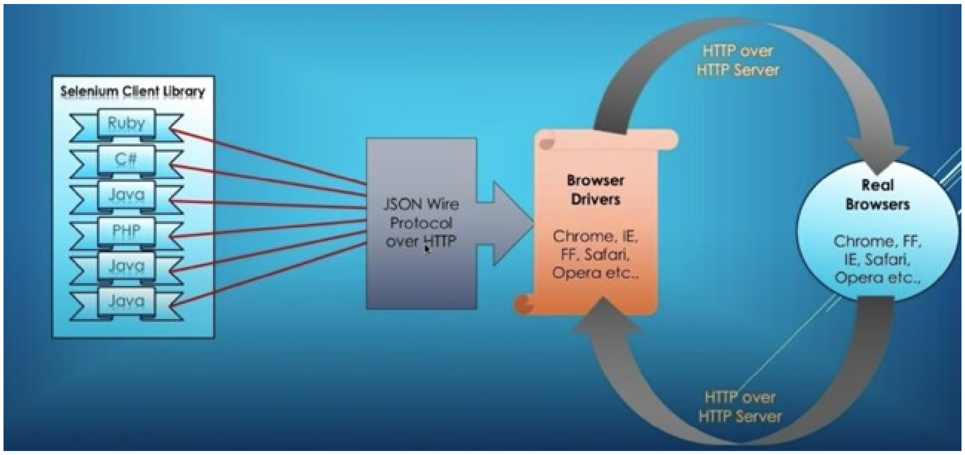
Limitations of Selenium

1. Selenium does not support desktop application test automation. User need to use Robot Class or AutoIT for performing any windows based actions as part of Web Automation testing.
2. REST or SOAP web services cannot be automated
3. For reporting / logging, reading-writing external files rely on external libraries (TestNG, Extent Report, Allure Reports and POI Jars for excel file reading etc.)

How Selenium WebDriver works

Selenium WebDriver (Selenium 2.0) automates the browsers actions by calling their native events (Click, MouseOver, KeyPress etc.) directly unlike Selenium RC which injects Javascript in browsers for test automation. WebDriver is much faster than Selenium RC and also handles scenarios like alerts, pop-ups, Keyboard and Mouse actions easily. Some of the widely used drivers in Selenium are ChromeDriver, FirefoxDriver, InternetExplorerDriver, SafariDriver and HtmlUnitDriver (headless driver) etc.

Selenium Architecture



About JSON Wire Protocol

https://www.selenium.dev/documentation/legacy/json_wire_protocol/

Selenium First Program

Required software for Selenium Program

1. Install Java
2. Download Eclipse (ensure Maven plugin is available)

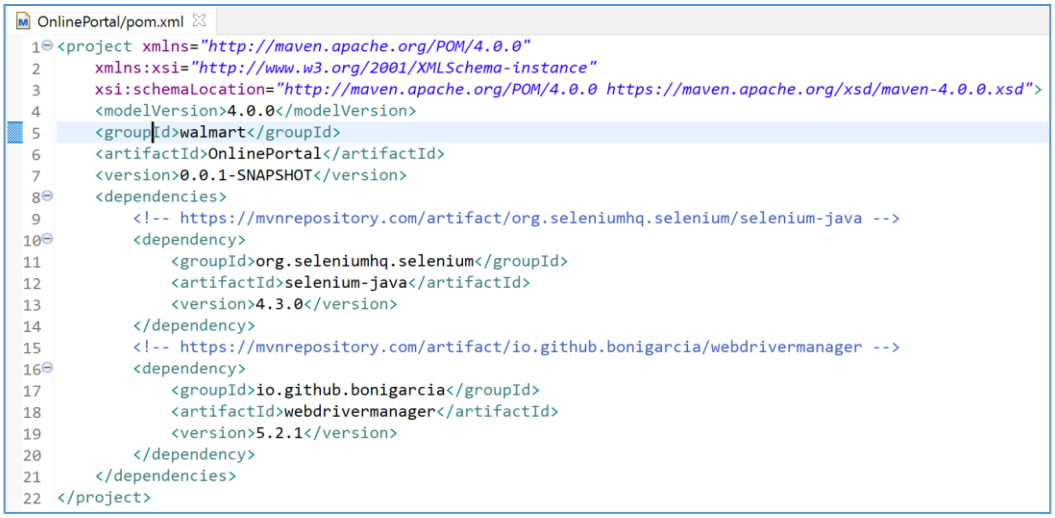
Additional details:

Selenium JAR from https://www.selenium.dev/downloads/

Browser Driver from https://chromedriver.chromium.org/

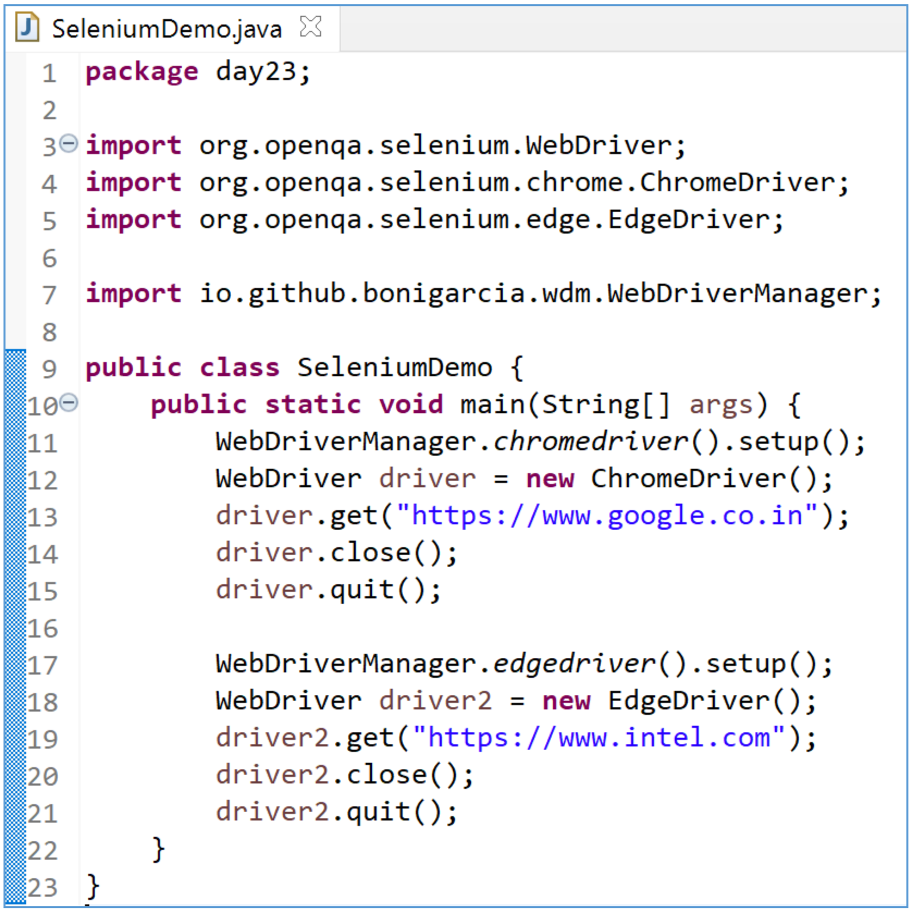
Steps to create Selenium Program

1. Open Eclipse and create a simple Maven Project
2. Update POM.xml with Selenium-Java and WebDriverManager dependency



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>walmart</groupId>
    <artifactId>OnlinePortal</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <dependencies>
        <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>4.3.0</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
        <dependency>
            <groupId>io.github.bonigarcia</groupId>
            <artifactId>webdrivermanager</artifactId>
            <version>5.2.1</version>
        </dependency>
    </dependencies>
</project>
```

First Program



```java
package day23;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.edge.EdgeDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

public class SeleniumDemo {
    public static void main(String[] args) {
        WebDriverManager.chromedriver().setup();
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.co.in");
        driver.close();
        driver.quit();

        WebDriverManager.edgedriver().setup();
        WebDriver driver2 = new EdgeDriver();
        driver2.get("https://www.intel.com");
        driver2.close();
        driver2.quit();
    }
}
```

Notes:

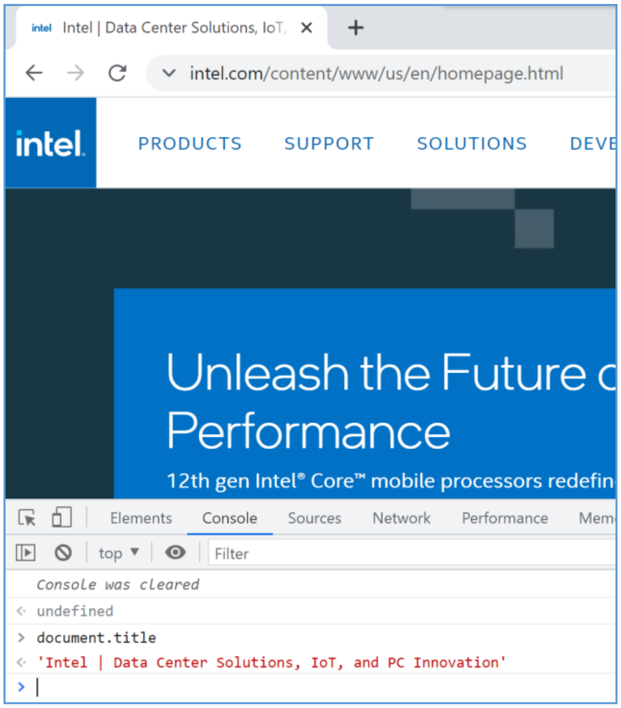
1. WebDriver is an interface. It has empty methods but not implemented. These methods can be implements anyone until method type and signatures are not violated.
2. ChromeDriver is a class that has been written specially for Chrome Browser. It has many methods and implemented and can be instantiated. ChromeDriver can perform all actions on the Chrome browser as defined in the interface WebDriver.

Browser Developer Tools

Use F12 function key to get a Browser Developer tools.

Browser Console

Click on Console Tab to give any browser (Document) related commands like below:


Intel | Data Center Solutions, IoT,
intel.com/content/www/us/en/homepage.html
intel.
PRODUCTS
SUPPORT
SOLUTIONS
DEVE
Unleash the Future c
Performance
12th gen Intel® Core™ mobile processors redefin
Elements
Console
Sources
Network
Performance
Mem
top
Filter

```
Console was cleared
< undefined
> document.title
< 'Intel | Data Center Solutions, IoT, and PC Innovation'
> |
```

Object Identification

Selenium By Class

Selenium By Class used to locate elements within a document using className, cssSelector, equals, hashCode, id, linkText, name, partialLinkTest, tagName and **xpath**.

Note: Ensure your xpath is able to identify your application WebElement uniquely.

XPath

XPath(XML Path Language) is based on a tree representation of the XML document and used for selecting nodes the document. It also provides ability to navigate around the tree.

1. Absolute XPath – A single "/" at the beginning instructs XPath engine start looking for an element from root node.
   /html/body/div[2]/div/div/footer/section[3]/div/ul/li[3]/a
2. Relative XPath – A double "//" instructs XPath engine look for an element anywhere in the XML document for given criteria.
   Syntax: //TagName[@AttributeName='Attribute Value']
   Examples:
   //*[@id='lst-ib']
   //button[@label='Continue']
3. Partial XPath – If locators properties changing frequently then try with "Contains" keyword.
   //div[contains(@class, 'footer-nav')]
   //h1[text()='Mobile Deposit Confirmation']

Partial XPath Keyword

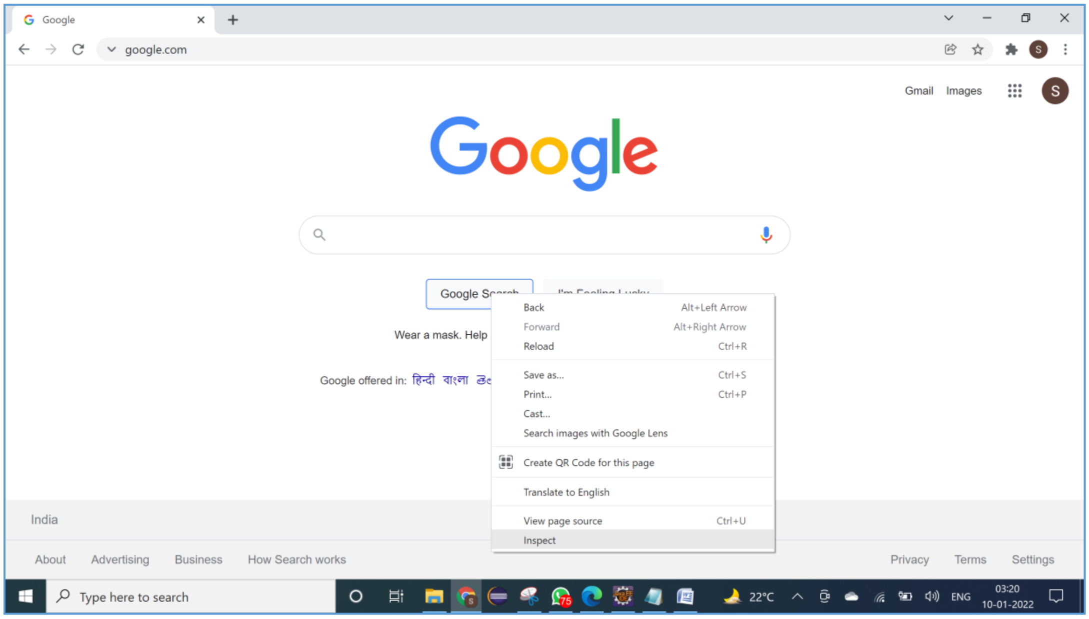
1. Contains
2. Text

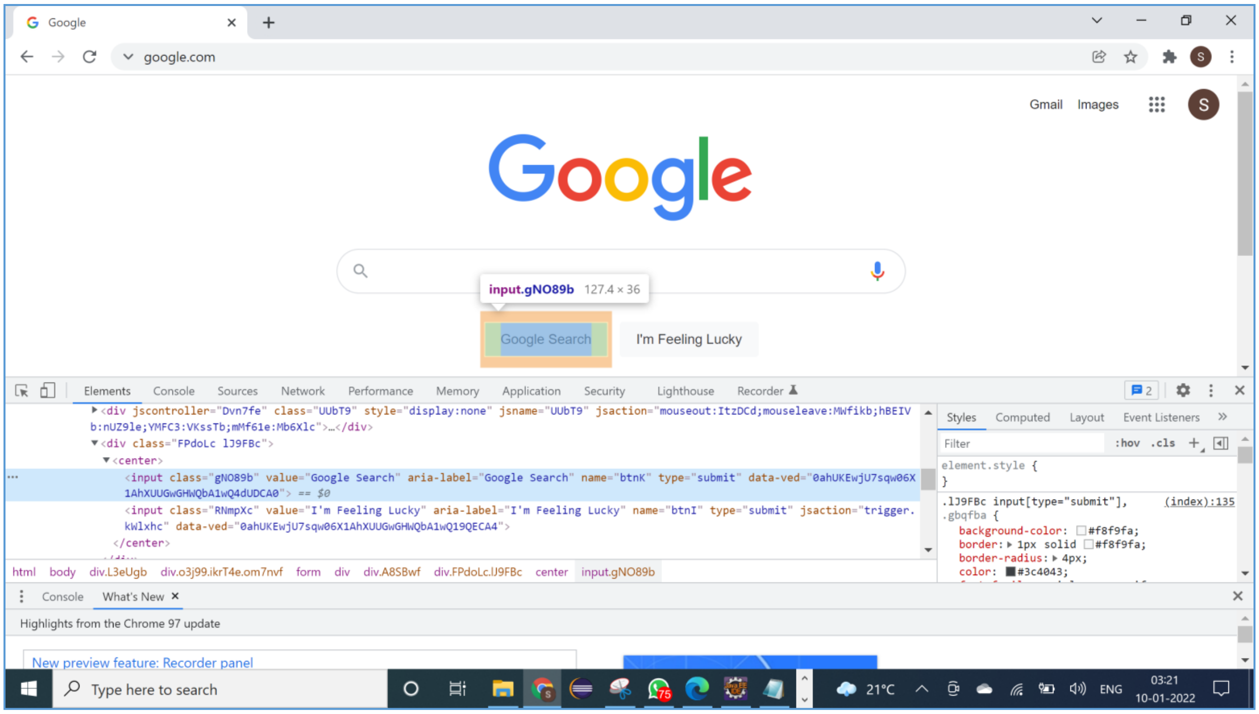
3. Start-with

DOM – Document Object Model

To write xpath of any WebElement in a Chrome browser follow below steps:

1. Open browser and launch your application URL
2. Mouse Hover on WebElement you want to get xpath
3. Mouse Right click on the WebElement and click on Inspect from shortcut menu
4. You will get DOM of the webpage

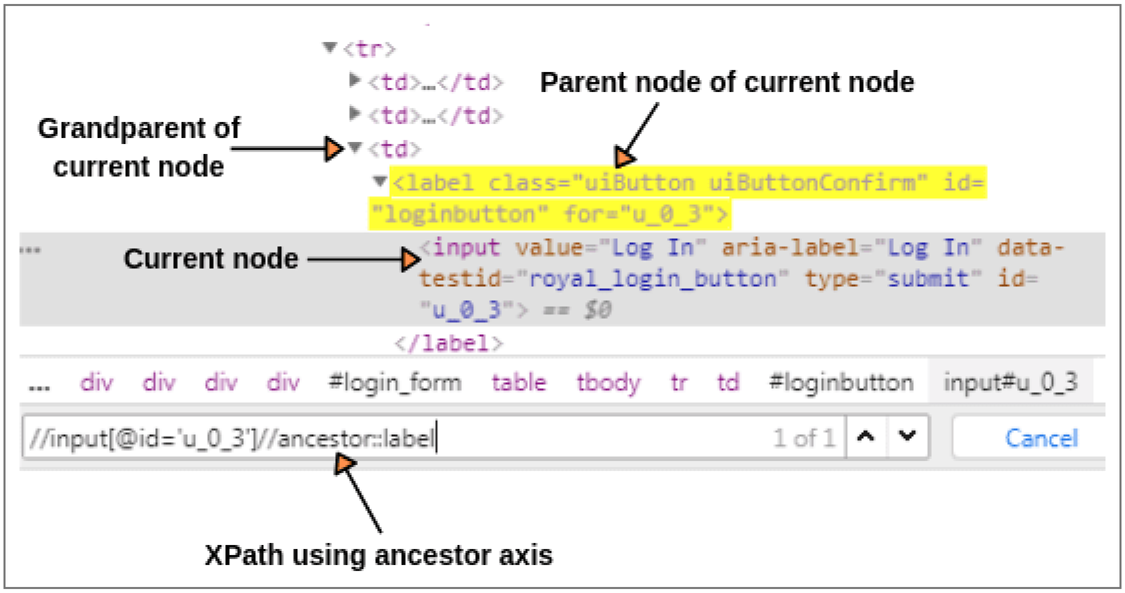




```
▶<div jscontroller="Dvn7fe" class="UUbT9" style="display:none" jsname="UUbT9" jsaction="mouseout:ItzDCd;mouseleave:MWfikb;hBEIV
b:nUZ9le;YMFC3:VKssTb;mMf61e:Mb6Xlc">…</div>
▼<div class="FPdoLc lJ9FBc">
  ▼<center>
      <input class="gNO89b" value="Google Search" aria-label="Google Search" name="btnK" type="submit" data-ved="0ahUKEwjU7sqw06X
      1AhXUUGwGHWQbA1wQ4dUDCA0"> == $0
      <input class="RNmpXc" value="I'm Feeling Lucky" aria-label="I'm Feeling Lucky" name="btnI" type="submit" jsaction="trigger.
      kWlxhc" data-ved="0ahUKEwjU7sqw06X1AhXUUGwGHWQbA1wQ19QECA4">
    </center>
```

```
element.style {
}
.lJ9FBc input[type="submit"],          (index):135
.gbqfba {
  background-color: □#f8f9fa;
  border: ▶ 1px solid □#f8f9fa;
  border-radius: ▶ 4px;
  color: ■#3c4043;
```

XPath Axes

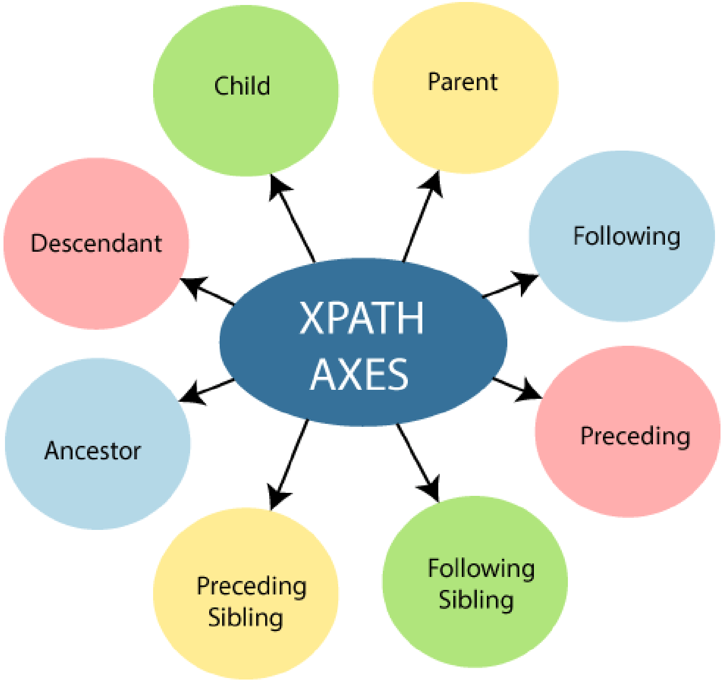


```
▼<tr>
  ►<td>…</td>
  ►<td>…</td>
  ▼<td>
    ▼<label class="uiButton uiButtonConfirm" id=
     "loginbutton" for="u_0_3">
        <input value="Log In" aria-label="Log In" data-
        testid="royal_login_button" type="submit" id=
        "u_0_3"> == $0
     </label>
```

... div div div div #login_form table tbody tr td #loginbutton input#u_0_3

//input[@id='u_0_3']//ancestor::label 1 of 1

Parent node of current node
Grandparent of current node
Current node
Cancel
XPath using ancestor axis



WebElement Class

Anything that is present on the web page is a WebElement such as text box, button, etc. WebElement represents an HTML element. Selenium WebDriver encapsulates a simple form element as an object of the WebElement. It basically represents a DOM element and all the HTML documents are made up by these HTML elements. Each HTML element consists of a start tag and an end tag. The content lies between the tags.

WebElement searchBox = driver.findElement(By.name(“q”));

In above example, searchBox is a WebElement and user can perform any below actions:

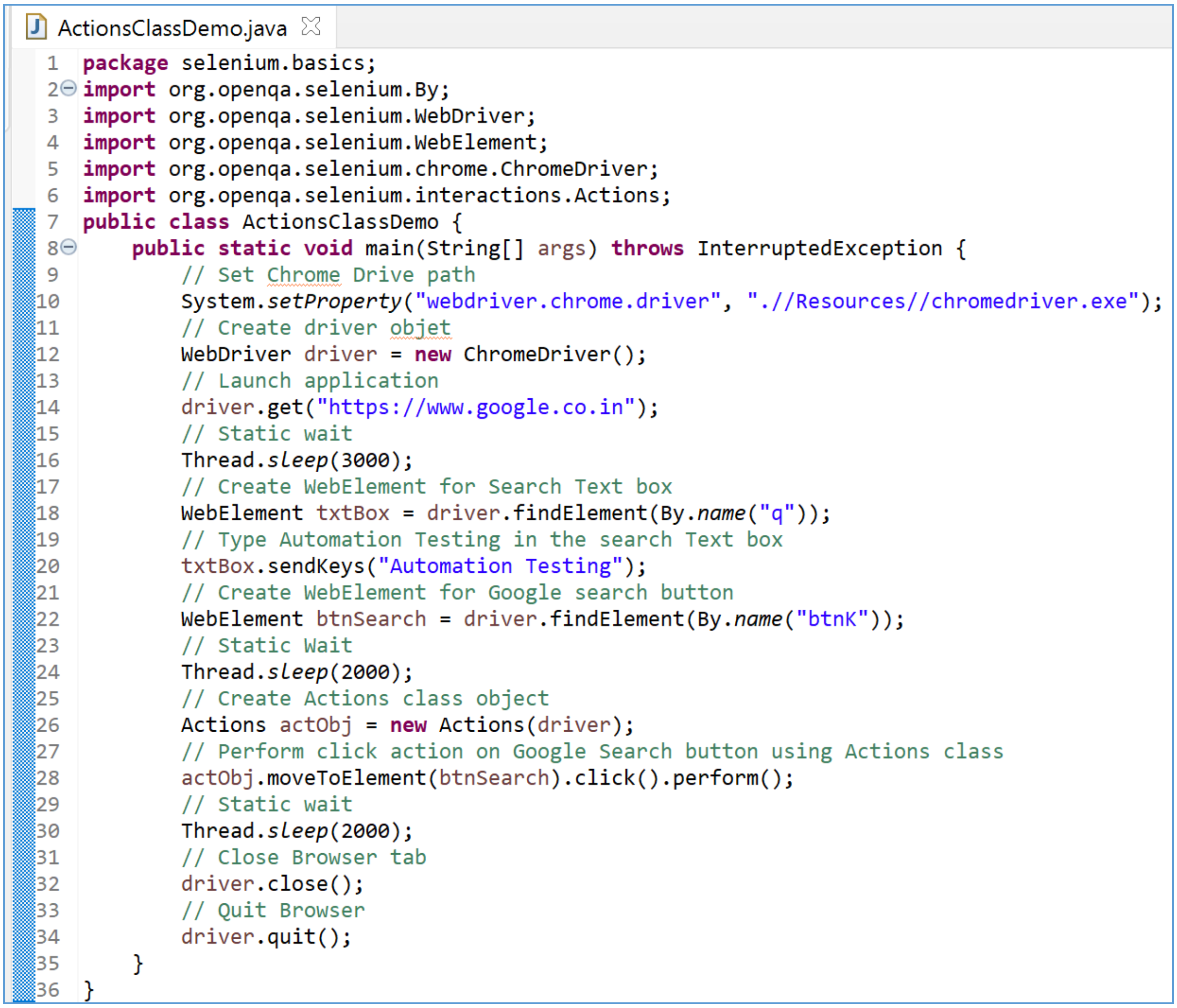
1. .sendKeys
2. .isDisplayed
3. .isSelected
4. .submit
5. .isEnabled
6. .clear
7. .click

Refer this link for more WebElement commands / actions.

Refer this link to understand various WebElement Exceptions.

Actions Class

Selenium Actions class is used to perform Mouse hover events, keyboard events. User can use this class for advanced actions such as drag and drop, multi selection etc.



```java
package selenium.basics;
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.interactions.Actions;
public class ActionsClassDemo {
    public static void main(String[] args) throws InterruptedException {
        // Set Chrome Drive path
        System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
        // Create driver objet
        WebDriver driver = new ChromeDriver();
        // Launch application
        driver.get("https://www.google.co.in");
        // Static wait
        Thread.sleep(3000);
        // Create WebElement for Search Text box
        WebElement txtBox = driver.findElement(By.name("q"));
        // Type Automation Testing in the search Text box
        txtBox.sendKeys("Automation Testing");
        // Create WebElement for Google search button
        WebElement btnSearch = driver.findElement(By.name("btnK"));
        // Static Wait
        Thread.sleep(2000);
        // Create Actions class object
        Actions actObj = new Actions(driver);
        // Perform click action on Google Search button using Actions class
        actObj.moveToElement(btnSearch).click().perform();
        // Static wait
        Thread.sleep(2000);
        // Close Browser tab
        driver.close();
        // Quit Browser
        driver.quit();
    }
}
```

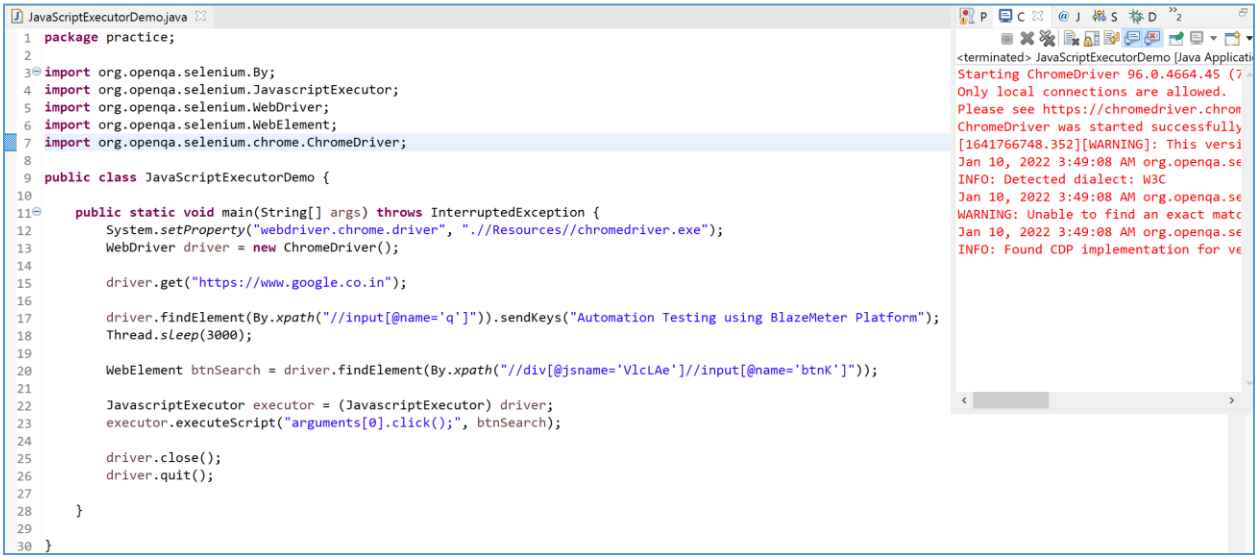
Another Example:

```
public class ActionsDemo {

	public static void main(String[] args) throws InterruptedException {
		System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
		WebDriver driver  = new ChromeDriver();
		driver.manage().window().maximize();
		Actions act = new Actions(driver);
		driver.get("https://www.online.citibank.co.in/");
		Thread.sleep(2000);
		WebElement lnkCreditCards = driver.findElement(By.xpath("//a[@title='Credit Cards' and @id='topMnucreditcards']"));
		act.moveToElement(lnkCreditCards).perform();
		Thread.sleep(10000);
		driver.close();
		driver.quit();
	}
}
```

JavascriptExecutor

Sometimes Selenium WebDriver gives ElementNotInteractableException then user might need JavascriptExecutor. Please look at below URL for more information.

https://www.browserstack.com/guide/javascriptexecutor-in-selenium



JavaScriptExecutorDemo.java

```
package practice;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class JavaScriptExecutorDemo {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
        WebDriver driver = new ChromeDriver();

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

        driver.findElement(By.xpath("//input[@name='q']")).sendKeys("Automation Testing using BlazeMeter Platform");
        Thread.sleep(3000);

        WebElement btnSearch = driver.findElement(By.xpath("//div[@jsname='VlcLAe']//input[@name='btnK']"));

        JavascriptExecutor executor = (JavascriptExecutor) driver;
        executor.executeScript("arguments[0].click();", btnSearch);

        driver.close();
        driver.quit();

    }

}
```

<terminated> JavaScriptExecutorDemo [Java Applicati

```
Starting ChromeDriver 96.0.4664.45 (7
Only local connections are allowed.
Please see https://chromedriver.chrom
ChromeDriver was started successfully
[1641766748.352][WARNING]: This versi
Jan 10, 2022 3:49:08 AM org.openqa.se
INFO: Detected dialect: W3C
Jan 10, 2022 3:49:08 AM org.openqa.se
WARNING: Unable to find an exact matc
Jan 10, 2022 3:49:08 AM org.openqa.se
INFO: Found CDP implementation for ve
```

Scroll Up and Scroll Down using JavascriptExecutor

```
//scroll down
js.executeScript("window.scrollBy(0,250)", "");

//Scroll up
js.executeScript("window.scrollBy(0,-350)", "");
```

Move to Element using JavascriptExecutor

```
JavascriptExecutor js = (JavascriptExecutor) driver;
WebElement lnkHelp = driver.findElement(By.xpath("//a[text()='Help']"));
js.executeScript("arguments[0].scrollIntoView();", lnkHelp);
```

Working with Edit / Text box

Use .sendKeys method to type or enter any text in an Edit or Text box of your webpage.

Approach #1:

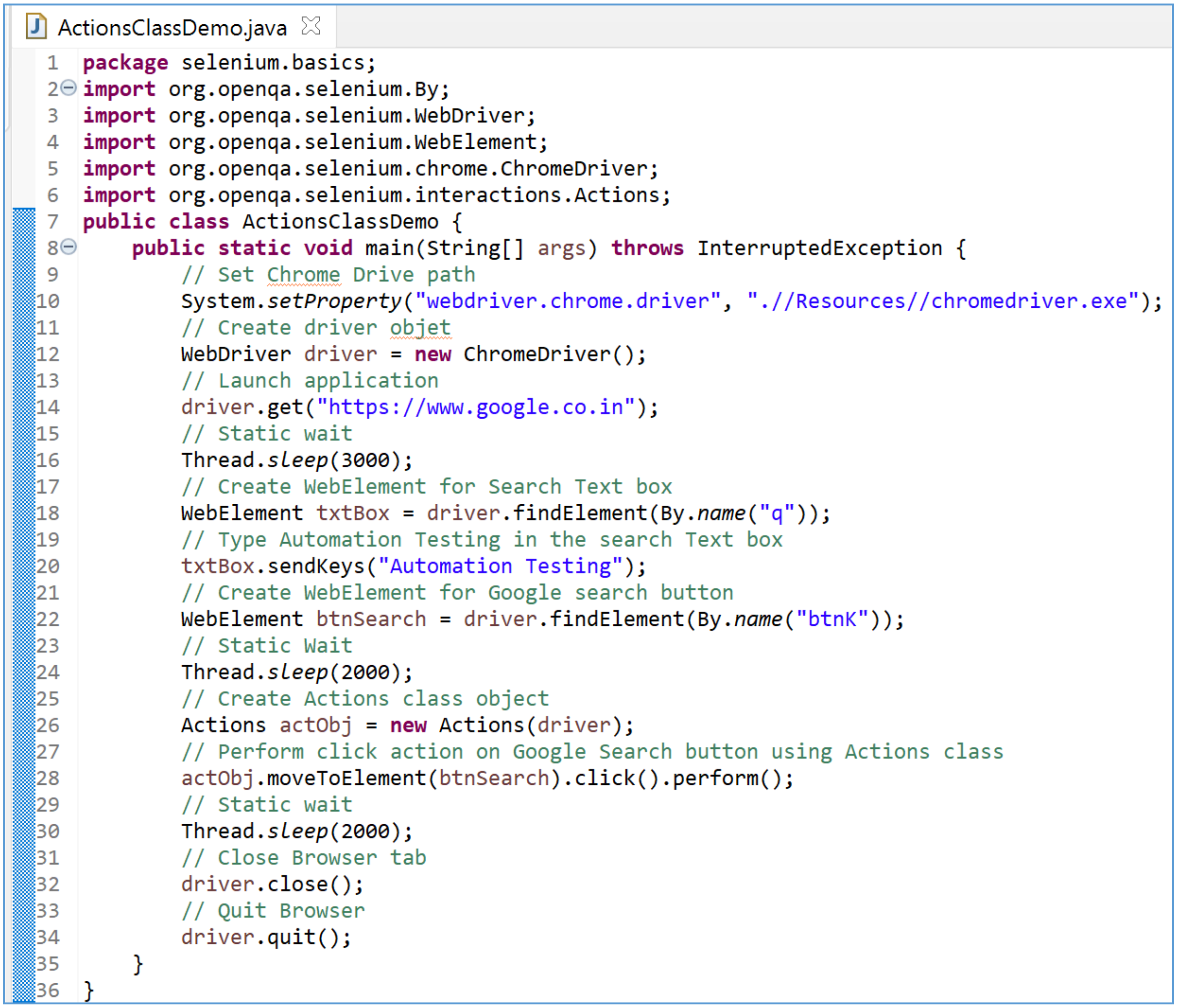
```
driver.findElement(By.xpath("//input[@name='q']")).sendKeys("Automation Testing using
BlazeMeter Platform");
```

Approach #2:

```
WebElement searchBox = driver.findElement(By.xpath("//input[@name='q']"));
searchBox.sendKeys("Automation Testing using BlazeMeter Platform");
```

Actions Class

Selenium Actions class is used to perform Mouse hover events, keyboard events. User can use this class for advanced actions such as drag and drop, multi selection etc.



```
package selenium.basics;
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.interactions.Actions;
public class ActionsClassDemo {
    public static void main(String[] args) throws InterruptedException {
        // Set Chrome Drive path
        System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
        // Create driver objet
        WebDriver driver = new ChromeDriver();
        // Launch application
        driver.get("https://www.google.co.in");
        // Static wait
        Thread.sleep(3000);
        // Create WebElement for Search Text box
        WebElement txtBox = driver.findElement(By.name("q"));
        // Type Automation Testing in the search Text box
        txtBox.sendKeys("Automation Testing");
        // Create WebElement for Google search button
        WebElement btnSearch = driver.findElement(By.name("btnK"));
        // Static Wait
        Thread.sleep(2000);
        // Create Actions class object
        Actions actObj = new Actions(driver);
        // Perform click action on Google Search button using Actions class
        actObj.moveToElement(btnSearch).click().perform();
        // Static wait
        Thread.sleep(2000);
        // Close Browser tab
        driver.close();
        // Quit Browser
        driver.quit();
    }
}
```

Another Example:

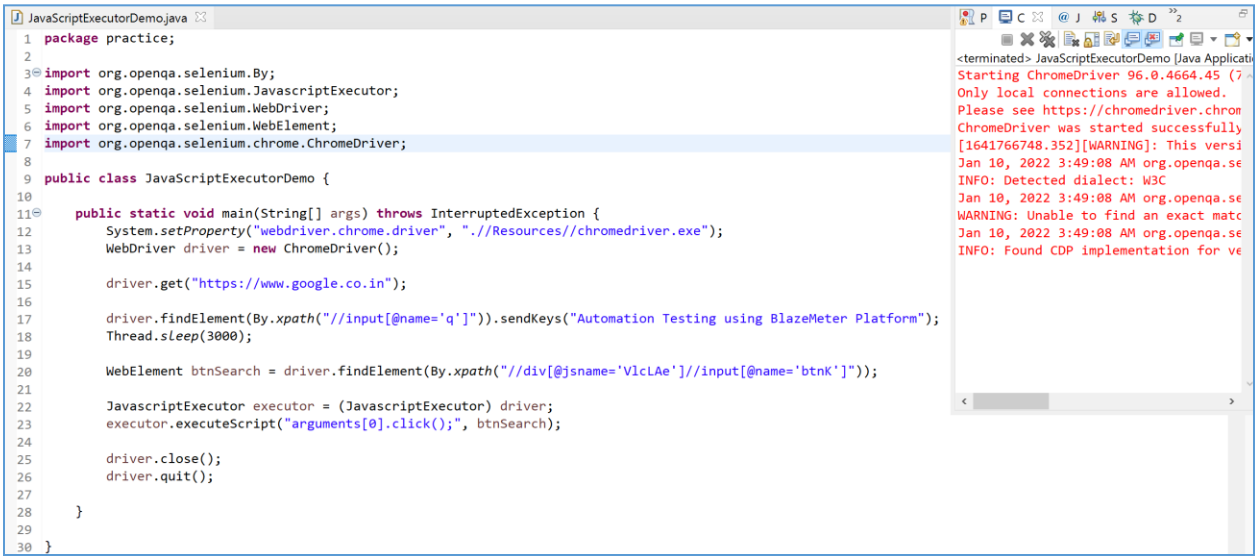
```
public class ActionsDemo {

	public static void main(String[] args) throws InterruptedException {
		System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
		WebDriver driver  = new ChromeDriver();
		driver.manage().window().maximize();
		Actions act = new Actions(driver);
```

```
                driver.get("https://www.online.citibank.co.in/");
                Thread.sleep(2000);
                WebElement lnkCreditCards = driver.findElement(By.xpath("//a[@title='Credit Cards' and @id='topMnucreditcards']"));
                act.moveToElement(lnkCreditCards).perform();
                Thread.sleep(10000);
                driver.close();
                driver.quit();
        }
}
```

JavascriptExecutor

Sometimes Selenium WebDriver gives ElementNotInteractableException then user might need JavascriptExecutor. Please look at below URL for more information.

https://www.browserstack.com/guide/javascriptexecutor-in-selenium



JavaScriptExecutorDemo.java

```
package practice;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class JavaScriptExecutorDemo {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
        WebDriver driver = new ChromeDriver();

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

        driver.findElement(By.xpath("//input[@name='q']")).sendKeys("Automation Testing using BlazeMeter Platform");
        Thread.sleep(3000);

        WebElement btnSearch = driver.findElement(By.xpath("//div[@jsname='VlcLAe']//input[@name='btnK']"));

        JavascriptExecutor executor = (JavascriptExecutor) driver;
        executor.executeScript("arguments[0].click();", btnSearch);

        driver.close();
        driver.quit();

    }

}
```

```
<terminated> JavaScriptExecutorDemo [Java Applicati
Starting ChromeDriver 96.0.4664.45 (7
Only local connections are allowed.
Please see https://chromedriver.chrom
ChromeDriver was started successfully
[1641766748.352][WARNING]: This versi
Jan 10, 2022 3:49:08 AM org.openqa.se
INFO: Detected dialect: W3C
Jan 10, 2022 3:49:08 AM org.openqa.se
WARNING: Unable to find an exact matc
Jan 10, 2022 3:49:08 AM org.openqa.se
INFO: Found CDP implementation for ve
```

Scroll Up and Scroll Down using JavascriptExecutor

```
//scroll down
js.executeScript("window.scrollBy(0,250)", "");

//Scroll up
js.executeScript("window.scrollBy(0,-350)", "");
```

Move to Element using JavascriptExecutor

```
JavascriptExecutor js = (JavascriptExecutor) driver;
WebElement lnkHelp = driver.findElement(By.xpath("//a[text()='Help']"));
js.executeScript("arguments[0].scrollIntoView();", lnkHelp);
```

Working with Edit / Text box

Use .sendKeys method to type or enter any text in an Edit or Text box of your webpage.

Approach #1:

```
driver.findElement(By.xpath("//input[@name='q']")).sendKeys("Automation Testing using BlazeMeter Platform");
```

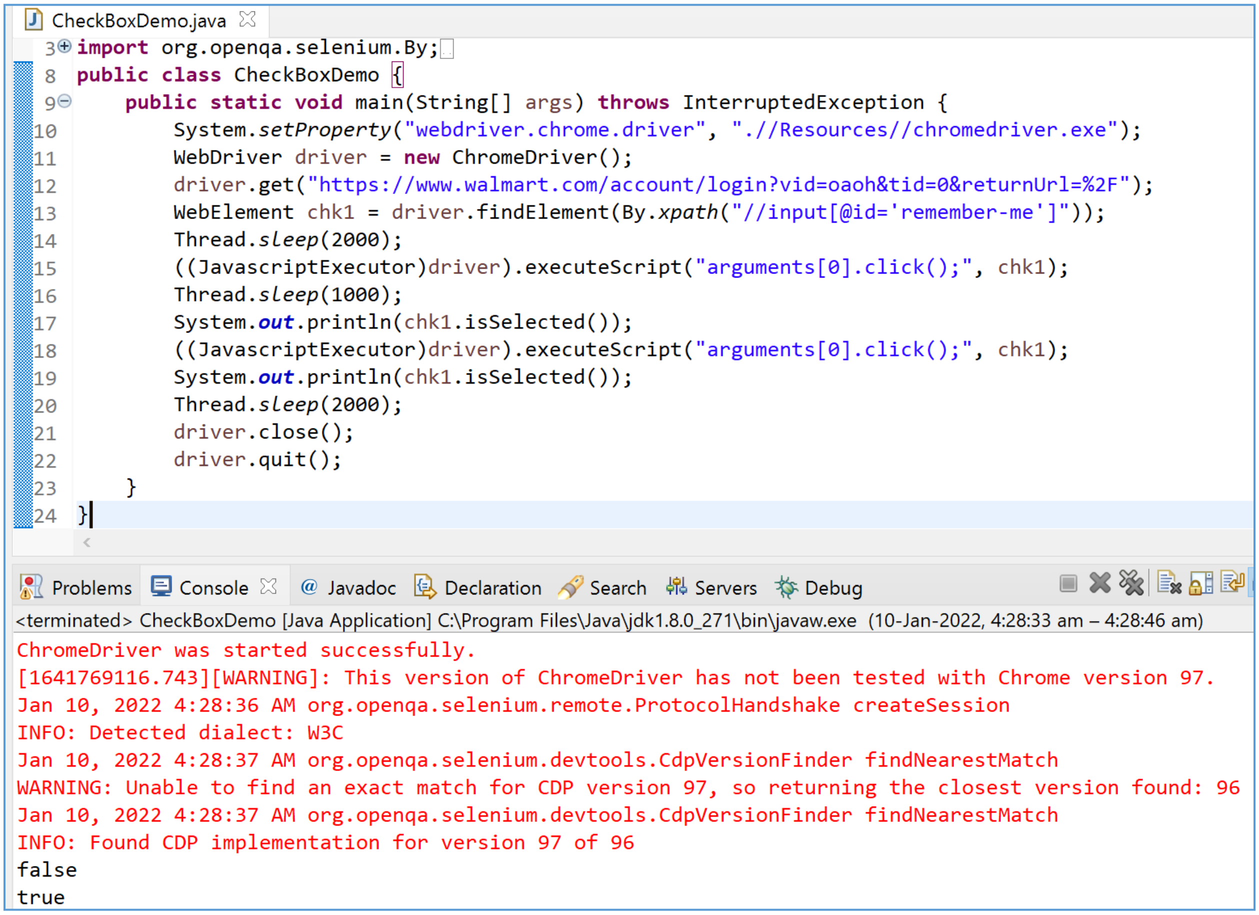
Approach #2:

```
WebElement searchBox = driver.findElement(By.xpath("//input[@name='q']"));
searchBox.sendKeys("Automation Testing using BlazeMeter Platform");
```

Working with check box

Use .click() method to check or uncheck a check box.

Use .isSelected() method to get the check box status that is selected or not.



```
import org.openqa.selenium.By;
public class CheckBoxDemo {
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.walmart.com/account/login?vid=oaoh&tid=0&returnUrl=%2F");
        WebElement chk1 = driver.findElement(By.xpath("//input[@id='remember-me']"));
        Thread.sleep(2000);
        ((JavascriptExecutor)driver).executeScript("arguments[0].click();", chk1);
        Thread.sleep(1000);
        System.out.println(chk1.isSelected());
        ((JavascriptExecutor)driver).executeScript("arguments[0].click();", chk1);
        System.out.println(chk1.isSelected());
        Thread.sleep(2000);
        driver.close();
        driver.quit();
    }
}
```

```
<terminated> CheckBoxDemo [Java Application] C:\Program Files\Java\jdk1.8.0_271\bin\javaw.exe  (10-Jan-2022, 4:28:33 am – 4:28:46 am)
ChromeDriver was started successfully.
[1641769116.743][WARNING]: This version of ChromeDriver has not been tested with Chrome version 97.
Jan 10, 2022 4:28:36 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Jan 10, 2022 4:28:37 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 97, so returning the closest version found: 96
Jan 10, 2022 4:28:37 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 97 of 96
false
true
```

Working with option button

Use .click() method to select any option button.

Working with Dropdown list box

```
WebElement dropDown = driver.findElement(By.id("Best Match"));
Select select1 = new Select(dropDown);
select1.selectByIndex(2);
select1.selectByValue("Price Low to High");
System.out.println(select1.getAllSelectedOptions());
System.out.println(select1.getOptions());
System.out.println(select1.getFirstSelectedOption());
```

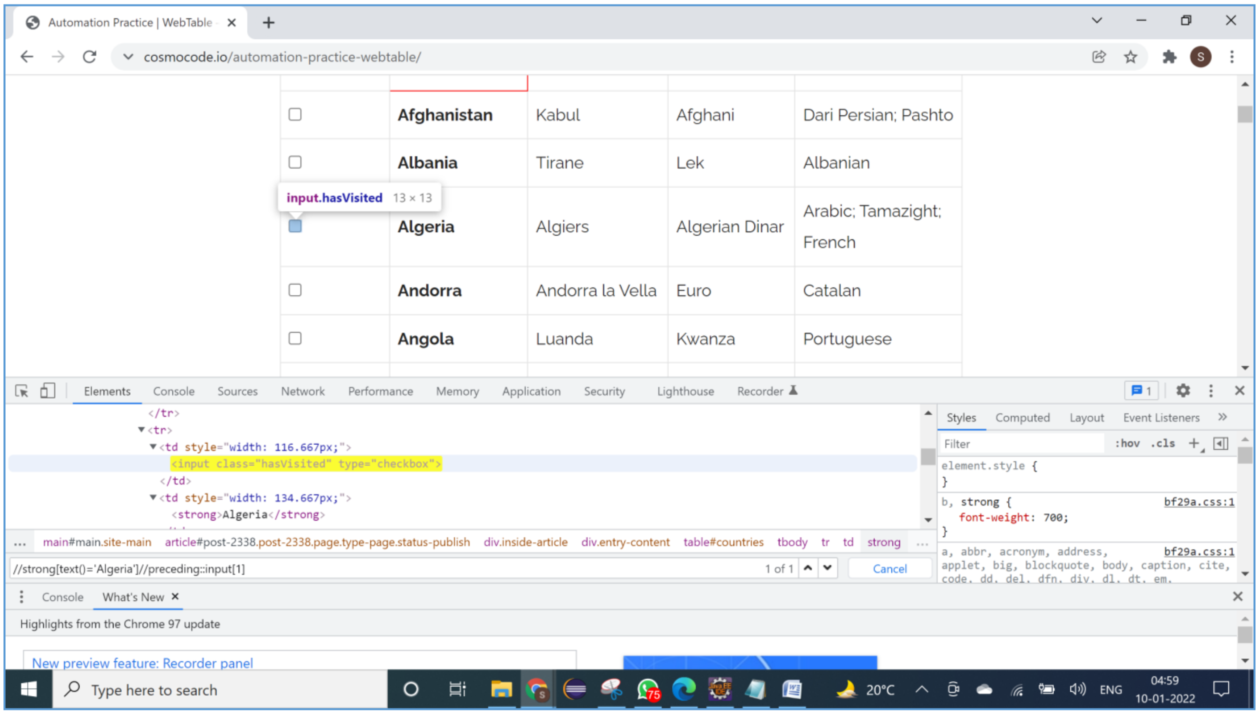
Refer this link for more information.

Working with Web Table

There are multiple ways to interact with a Web Table in a Webpage. User might need to use XPath Axes when there is no unique name. Here is an example:

https://cosmocode.io/automation-practice-webtable/

//strong[text()='Algeria']//preceding::input[1]



□	**Afghanistan**	Kabul	Afghani	Dari Persian; Pashto
□	**Albania**	Tirane	Lek	Albanian
□	**Algeria**	Algiers	Algerian Dinar	Arabic; Tamazight; French
□	**Andorra**	Andorra la Vella	Euro	Catalan
□	**Angola**	Luanda	Kwanza	Portuguese

```
      </tr>
    ▼<tr>
      ▼<td style="width: 116.667px;">
          <input class="hasVisited" type="checkbox">
        </td>
      ▼<td style="width: 134.667px;">
          <strong>Algeria</strong>
```

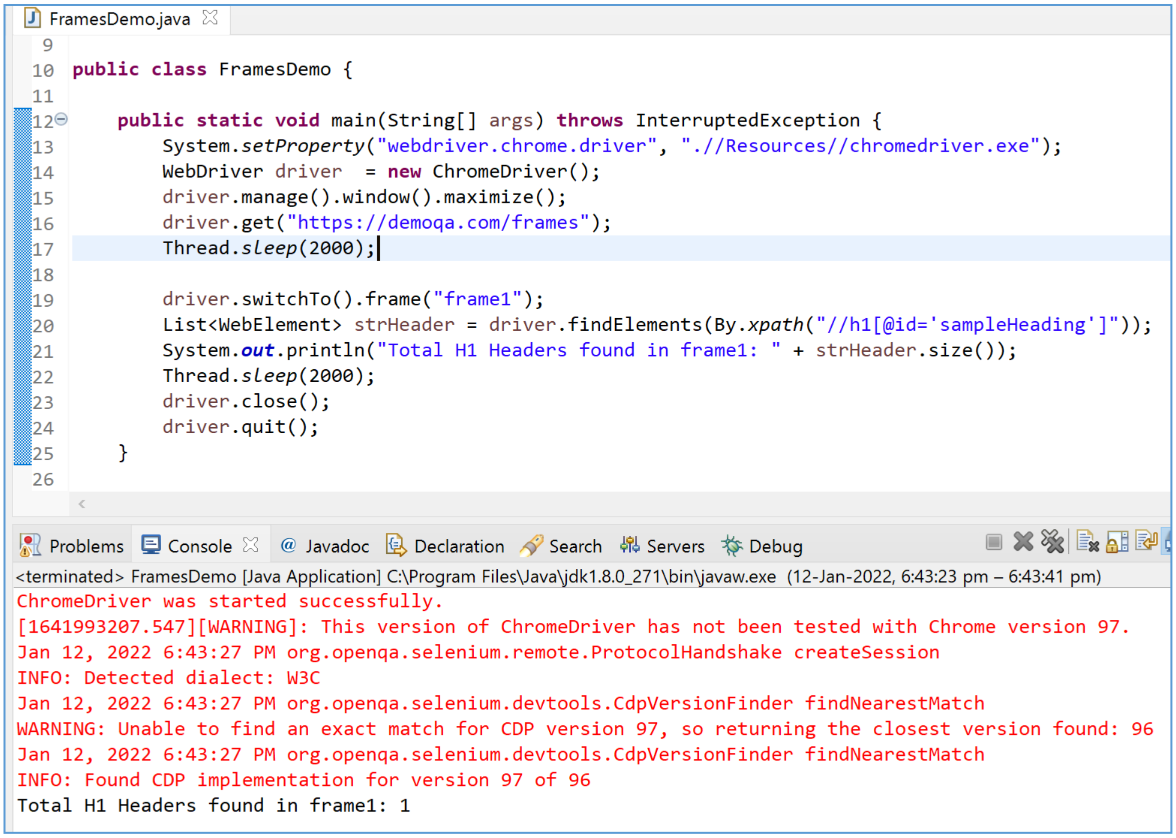
... main#main.site-main article#post-2338.post-2338.page.type-page.status-publish div.inside-article div.entry-content table#countries tbody tr td strong ...

//strong[text()='Algeria']//preceding::input[1] 1 of 1 Cancel

```
element.style {
}
b, strong {                         bf29a.css:1
   font-weight: 700;
}
a, abbr, acronym, address,          bf29a.css:1
applet, big, blockquote, body, caption, cite,
code, dd, del, dfn, div, dl, dt, em,
```

Working with Frames

iFrame (Inline Frame) is a web page embedded in a web page. Selenium WebDriver could not find any WebElements inside iFrame directly. User need to switch to iFrame to proceed further testing.

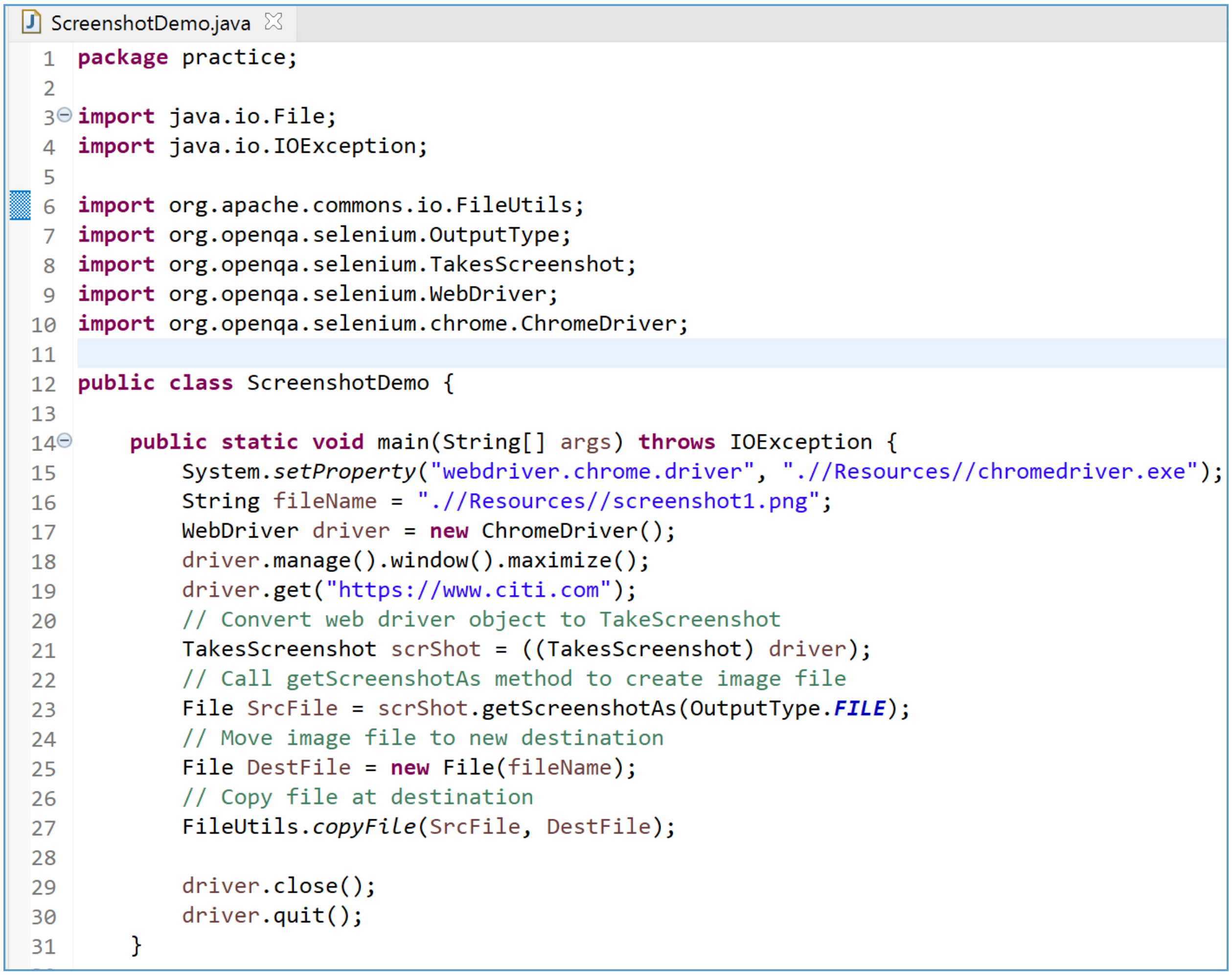


```java

public class FramesDemo {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
        WebDriver driver  = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://demoqa.com/frames");
        Thread.sleep(2000);

        driver.switchTo().frame("frame1");
        List<WebElement> strHeader = driver.findElements(By.xpath("//h1[@id='sampleHeading']"));
        System.out.println("Total H1 Headers found in frame1: " + strHeader.size());
        Thread.sleep(2000);
        driver.close();
        driver.quit();
    }

```

<terminated> FramesDemo [Java Application] C:\Program Files\Java\jdk1.8.0_271\bin\javaw.exe (12-Jan-2022, 6:43:23 pm – 6:43:41 pm)

```
ChromeDriver was started successfully.
[1641993207.547][WARNING]: This version of ChromeDriver has not been tested with Chrome version 97.
Jan 12, 2022 6:43:27 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Jan 12, 2022 6:43:27 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 97, so returning the closest version found: 96
Jan 12, 2022 6:43:27 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found CDP implementation for version 97 of 96
Total H1 Headers found in frame1: 1
```

Capture Screenshot



```java
package practice;

import java.io.File;
import java.io.IOException;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class ScreenshotDemo {

    public static void main(String[] args) throws IOException {
        System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");
        String fileName = ".//Resources//screenshot1.png";
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.citi.com");
        // Convert web driver object to TakeScreenshot
        TakesScreenshot scrShot = ((TakesScreenshot) driver);
        // Call getScreenshotAs method to create image file
        File SrcFile = scrShot.getScreenshotAs(OutputType.FILE);
        // Move image file to new destination
        File DestFile = new File(fileName);
        // Copy file at destination
        FileUtils.copyFile(SrcFile, DestFile);

        driver.close();
        driver.quit();
    }
```

Navigation Commands

```
driver.navigate().to("www.google.com");
driver.navigate().forward();
driver.navigate().back();
driver.navigate().refresh();
```

Alerts

Frequently used alerts methods in test automation.

```java
Alert alert = driver.switchTo().alert();
alert.dismiss();
alert.getText();
alert.accept();
alert.sendKeys("Hello World.");
```

Selenium Wait

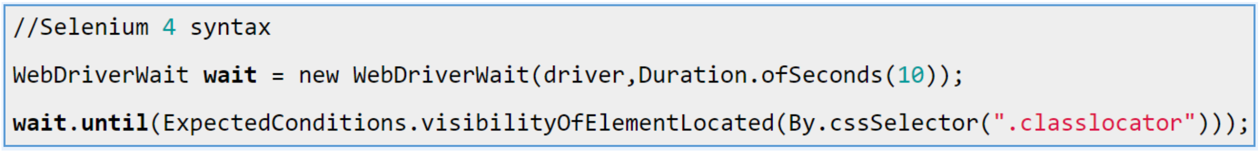
Selenium WebDriver provides three commands to implement waits in tests.

1. Implicit Wait

```java
driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);   //Sel <= 4
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10)); //Sel >4.x
```

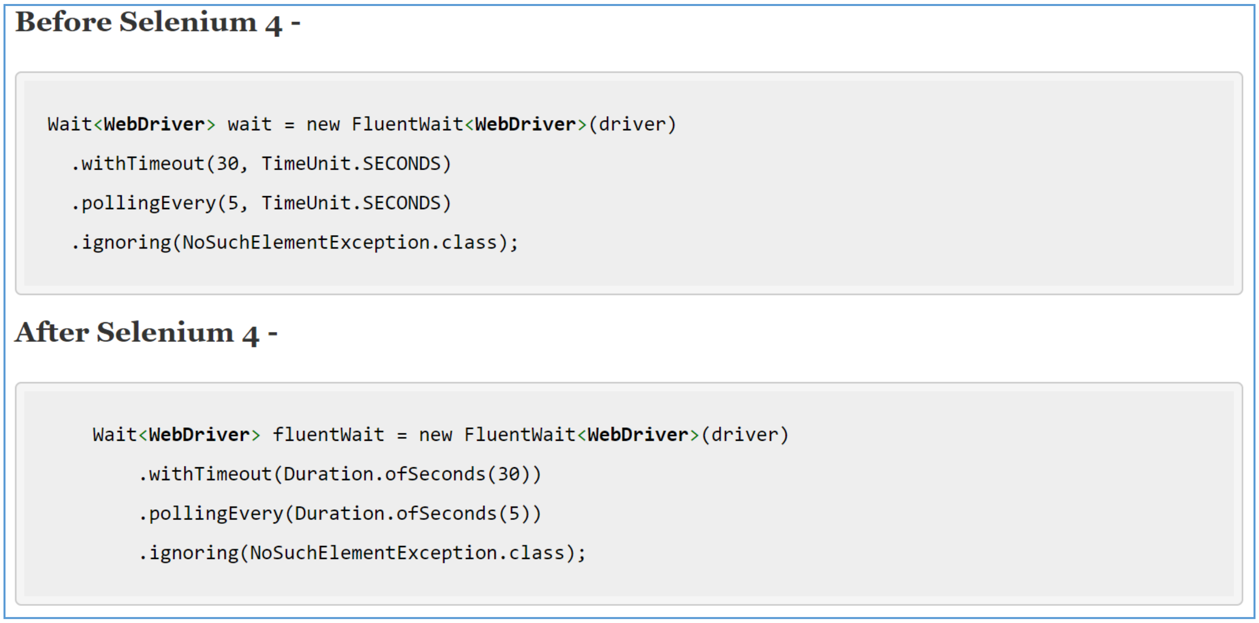
2. Explicit Wait

```
WebDriverWait explictWait = new ~~WebDriverWait~~(driver, 20);
```



```
//Selenium 4 syntax
WebDriverWait wait = new WebDriverWait(driver,Duration.ofSeconds(10));
wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector(".classlocator")));
```

3. Fluent Wait



Before Selenium 4 -

```
Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
  .withTimeout(30, TimeUnit.SECONDS)
  .pollingEvery(5, TimeUnit.SECONDS)
  .ignoring(NoSuchElementException.class);
```

After Selenium 4 -

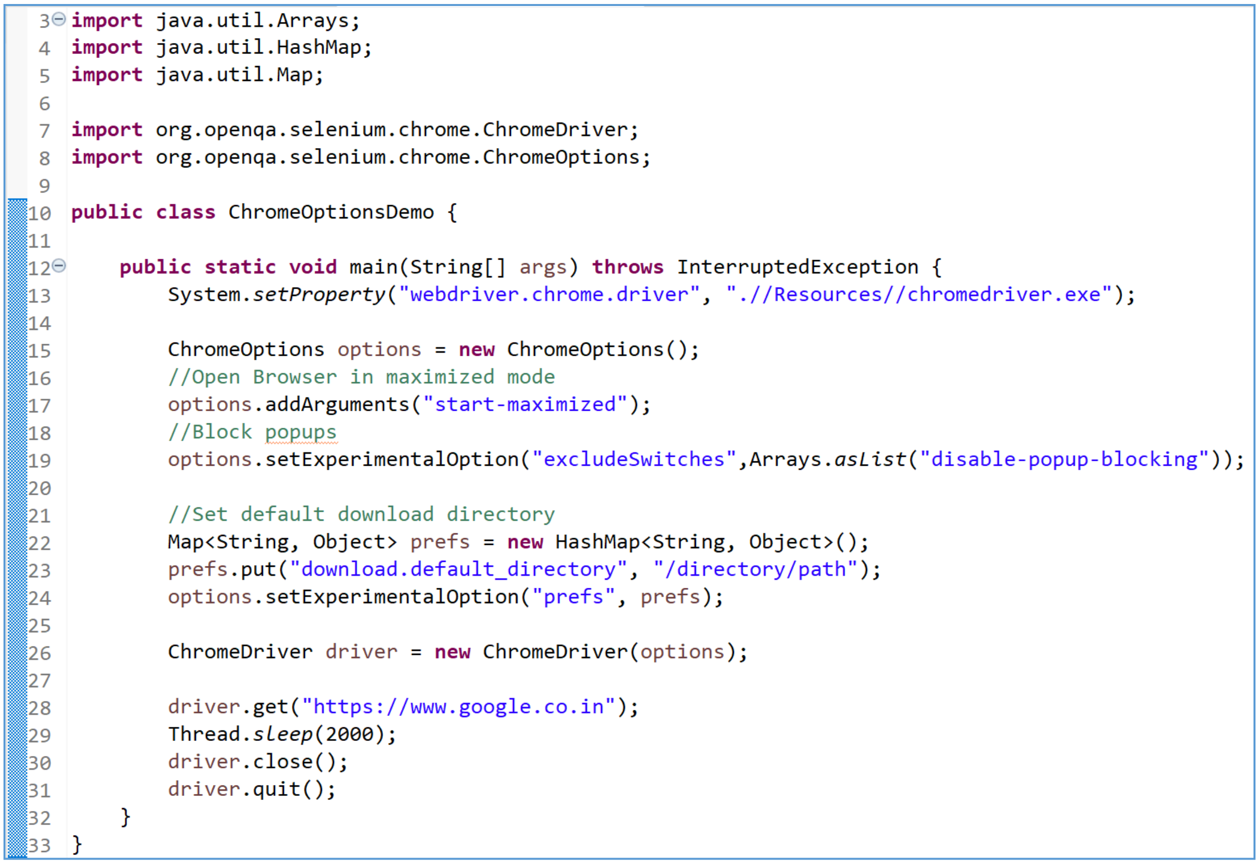
```
    Wait<WebDriver> fluentWait = new FluentWait<WebDriver>(driver)
        .withTimeout(Duration.ofSeconds(30))
        .pollingEvery(Duration.ofSeconds(5))
        .ignoring(NoSuchElementException.class);
```

https://devqa.io/webdriver-explicit-implicit-fluent-wait/

Browser Driver Options

Before Selenium 4 there was DesiredCapabilities class exists and we used these capabilities to define test environment such as OS, Browser name, version etc. In Selenium 4.x replaced with options object.

- Chrome – ChromeOptions
- Microsoft Edge – EdgeOptions
- Firefox – FirefoxOptions
- Safari – SafariOptions
- Internet Explorer (IE) – InternetExplorerOptions



```java
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;

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

public class ChromeOptionsDemo {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", ".//Resources//chromedriver.exe");

        ChromeOptions options = new ChromeOptions();
        //Open Browser in maximized mode
        options.addArguments("start-maximized");
        //Block popups
        options.setExperimentalOption("excludeSwitches",Arrays.asList("disable-popup-blocking"));

        //Set default download directory
        Map<String, Object> prefs = new HashMap<String, Object>();
        prefs.put("download.default_directory", "/directory/path");
        options.setExperimentalOption("prefs", prefs);

        ChromeDriver driver = new ChromeDriver(options);

        driver.get("https://www.google.co.in");
        Thread.sleep(2000);
        driver.close();
        driver.quit();
    }
}
```

Refer this link for more browser options.