



**We are on a mission to address the digital skills gap for 10 Million+ young professionals, train and empower them to forge a career path into future tech**

# Handling Browser - Navigation

22 APRIL, 2022



# Contents

- Browser Commands
- Navigation Commands
- Web Elements Commands



# Browser Commands



## Browser Commands

# Browser Commands

- Listed below are the basic and most commonly used Browser Commands in Selenium Webdriver:
  1. Get command
  2. Get Title command
  3. Get Current URL command
  4. Get Page Source Command
  5. Close command
  6. Quit command

## Browser Commands

# Browser Commands

### 1. Get command

- The `get()` method takes a string URL as a parameter and returns nothing.
- This method opens the specified URL in the current browser window.
- URL must be in the form of <https://www.google.com>.
- If the HTTPS is not included then it will throw a message “Cannot navigate to invalid URL”.
- The general syntax to use get command is as follows:

**get (String arg) : void**

- Example

```
driver.get("https://www.google.com"); OR String URL = "https://www.google.com";  
driver.get(URL);
```

## Browser Commands

# Browser Commands

## 2. Get Title command

- The getTitle() method takes nothing as a parameter and returns the page title of the currently loaded web page. If the web page has no title, it will return a null String.
- The general syntax to use getTitle command is as follows:

**driver.getTitle();**

**OR**

**String Title = driver.getTitle();**

## Browser Commands

# Browser Commands

### 3. Get Current URL command

- The `getCurrentURL()` method takes nothing as a parameter and returns URL of the web page currently loaded in the browser.
- This method is used to get string URL of the current web page loaded in the opened browser.
- The general syntax to use getTitle command is as follows:

```
driver.getCurrentUrl();
```

**OR**

```
String CurrentUrl = driver.getCurrentUrl();
```



## Browser Commands

# Browser Commands

### 4. Get Page Source command

- The `getPageSource()` method accepts nothing as a parameter and returns the source code of the current web page.
- This method is used to get the page source code of the currently loaded web page.
- The general syntax to use `getCurrentUrl` command is as follows:

```
driver.getPageSource();
```

OR

```
String pageSource = driver.getPageSource();
```

# Output of getPageSource()

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904})
on port 51392
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver
safe.
ChromeDriver was started successfully.
[1655948563.487][WARNING]: This version of ChromeDriver has not been tested with Chrome version 102.
Jun 23, 2022 7:12:43 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Jun 23, 2022 7:12:43 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 102, so returning the closest version found: a no-op
implementation
Jun 23, 2022 7:12:43 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Unable to find CDP implementation matching 102.
Jun 23, 2022 7:12:43 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific
version of the CDP using something similar to 'org.seleniumhq.selenium:selenium-devtools-v86:4.0.0' where
the version ("V86") matches the version of the chromium-based browser you're using and the version number of
the artifact is the same as Selenium's.
<html class="client-js" lang="en" dir="ltr"><head>
<meta charset="UTF-8">
<title>Wikipedia, the free encyclopedia</title>
<script>document.documentElement.className="client-
js";RLCONF={"wgBreakFrames":false,"wgSeparatorTransformTable":["",""],"wgDigitTransformTable":
["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":
["","January","February","March","April","May","June","July","August","September","October","November","Dece
mber"],"wgRequestId":"8d43591d-
d761-425d-8929-35a2baccbb6c","wgCSPNonce":false,"wgCanonicalNamespace":"","wgCanonicalSpecialPageName":false
,"wgNamespaceNumber":0,"wgPageName":"Main_Page","wgTitle":"Main
Page","wgCurRevisionId":1093586708,"wgRevisionId":1093586708,"wgArticleId":15580374,"wgIsArticle":true,"wgIs
Redirect":false,"wgAction":"view","wgUserName":null,"wgUserGroups":["*"],"wgCategories":
[],"wgPageContentLanguage":"en","wgPageContentModel":"wikitext","wgRelevantPageName":"Main_Page","wgRelevant
ArticleId":15580374,"wgIsProbablyEditable":false,"wgRelevantPageIsProbablyEditable":false,"wgRestrictionEdit
":["sysop"],"wgRestrictionMove":["sysop"],"wgIsMainPage":true,"wgFlaggedRevsParams":{"
tags":{"status":{"levels":1}}},"wgVisualEditor":
{"pageLanguageCode":"en","pageLanguageDir":"ltr","pageVariantFallbacks":"en"},"wgMFDDisplayWikibaseDescriptio
ns":
{"search":true,"nearby":true,"watchlist":true,"tagline":false},"wgWMESchemaEditAttemptStepOversample":false,
"wgWMEPageLength":3000,"wgNoticeProject":"wikipedia","wgMediaViewerOnClick":true,"wgMediaViewerEnabledByDefa
ult":true,"wgPopupsFlags":10,"wgULSCurrentAutonym":"English","wgEditSubmitButtonLabelPublish":true,"wgCentra
lAuthMobileDomain":false,"wgULSPosition":"interlanguage","wgULSisCompactLinksEnabled":true,"wgWikibaseItemId
":"Q5296","gEHomepageSuggestedEditsEnableTopics":true,"wgGETopicsMatchModeEnabled":false};RLSTATE={"ext.glob
alCssJs.user.styles":"ready","site.styles":"ready","user.styles":"ready","ext.globalCssJs.user":"ready","use
r":"ready","user.options":"loading","skins.vector.styles.legacy":"ready","ext.visualEditor.desktopArticleTar
get.noscript":"ready","ext.wikimediaBadges":"ready","ext.uls.interlanguage":"ready"
get.noscript":"ready","ext.wikimediaBadges":"ready","ext.uls.interlanguage":"ready"
```

## Browser Commands

# Browser Commands

### 5. Close command

- The close method takes nothing as a parameter and returns nothing.
- This method is used to close only the browser window that web driver is currently controlling.
- The general syntax to use close command is as follows:

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

## Browser Commands

# Browser Commands

### 6. Quit command

- The quit() method accepts nothing as a parameter and returns nothing.
- The closes all the browser windows that are handled by the selenium.
- The general syntax to use quit command is as follows:

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

# Navigation Commands





## Navigation Commands

# Navigation Commands

- Listed below are the most commonly used Navigation Commands in Selenium WebDriver:

1. Navigate to command
2. Back command
3. Forward command
4. Refresh commands





## Navigation Commands

# Navigate using URL

**1. GoToUrl** - This command is used to navigate to particular URL or website.

### Syntax:

```
driver.navigate().to("URL");
```

where URL is website url that the user need to navigate.

### Example:

```
driver.navigate().to("http://www.google.com");
```

## Navigation Commands

# Navigate using commands

### Command used for navigation with the help of command

- **`driver.navigate().forward()`** -Takes you forward by one page on the browser's history.
- **`driver.navigate().back()`** -Takes you back by one page on the browser's history.
- **`driver.navigate().refresh()`** -Takes you back the refreshed current URL in the browser.

Note : return type is void and these commands will not accept any parameter.

## Sample Java Selenium Program

```
package com.mysamp1;

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

import io.github.bonigarcia.wdm.WebDriverManager;

public class navigate {

    public static void main(String[] args) {
        WebDriverManager.chromedriver().setup();
        ChromeOptions co=new ChromeOptions();
        WebDriver driver=new ChromeDriver(co);
        driver.get("http://omayo.blogspot.com");
        driver.navigate().to("http://google.com");
        driver.navigate().back();
        driver.navigate().forward();
        driver.navigate().refresh();

    }

}
```

## Navigation Commands

# navigation Commands – Practice

### *Practice Exercise – 1*

- 1. Launch new Browser*
- 2. Open DemoQA.com website*
- 3. Click on Registration link using "**driver.findElement(By.xpath("./[@id='menu-item-374']/a')).click();***
- 4. Come back to Home page (Use 'Back' command)*
- 5. Again go back to Registration page (This time use 'Forward' command)*
- 6. Again come back to Home page (This time use 'To' command)*
- 7. Refresh the Browser (Use 'Refresh' command)*
- 8. Close the Browser*

# Web Element Commands



## Web Elements Commands

# Web Elements Commands

- Web element refers to an HTML element, which contains a start tag, an end tag and the content in between.
- Listed below are the basic and most commonly used Browser Commands in Selenium Webdriver:

1. Clear Command

2. Click Command

3. getText Command

4. isSelected Command

5. isEnabled Command

6. isDisplayed Command

7. getSize Command

8. getCssValue Command

9. getLocation Command

10. submit Command

11. getAttribute Command

12. getTagName Command

13. sendKeys Command



## Web Elements Commands

# Web Elements Commands

### 1. Clear command

- The Clear() method has no effect on other elements.
- If this element is a text entry element, this will clear the value.
- This method accepts nothing as a parameter and returns nothing.
- Text entry elements are INPUT and TEXTAREA elements.
- Example

```
WebElement element = driver.findElement(By.id("UserName"));  
  
element.clear();
```

**OR**

```
driver.findElement(By.id("UserName")).clear();
```

## Web Elements Commands

# Web Elements Commands

## 2. Click command

- This command helps to interact with web elements like text elements, links, radio boxes and many more.
- This method accepts nothing as a parameter and returns nothing.
- Example

```
WebElement element = driver.findElement(By.linkText("ToolsQA"));  
element.click();
```

**OR**

```
driver.findElement(By.linkText("ToolsQA")).click();
```

## Web Elements Commands

### 3. getText command

- This command retrieves the visible innerText of the element. This accepts nothing as a parameter but returns a String value.
- Command : `Element.getText()`
- This returns an innerText of the element, including sub-elements, without any leading or trailing whitespace.
- Example

```
WebElement element = driver.findElement(By.xpath("anyLink"));
```

```
String linkText = element.getText();
```

## Web Elements Commands

# Web Elements Commands

### 4. isEnabled command

- isEnabled() method is used to check if the web element is enabled or disabled within the web page.
- This method returns “true” value if the specified web element is enabled on the web page otherwise returns “false” value if the web element is disabled on the web page.
- Example

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

```
boolean status = element.isEnabled();
```

//Or can be written as

```
boolean status = driver.findElement(By.id("UserName")).isEnabled();
```

## Web Elements Commands

# Web Elements Commands

### 4. isEnabled command

- Example : //Or can be used as

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

```
boolean status = element.isEnabled();
```

```
// Check that if the Text field is enabled, if yes enter value
```

```
if(status)
```

```
{
```

```
element.sendKeys("ToolsQA");
```

```
}
```

## Web Elements Commands

### 5. isDisplayed command

- The isDisplayed() method is used to check whether an element is displayed on a web page or not.
- It returns a boolean value (true) if the target element is displayed otherwise returns false.
- Example

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

```
boolean status = element.isDisplayed();
```

//Or can be written as

```
boolean staus = driver.findElement(By.id("UserName")).isDisplayed();
```



## Web Elements Commands

### 6. isSelected command

- The isSelected() method checks that if an element is selected on the web page or not.
- It returns a boolean value (true) if selected, else false for deselected.
- It can be executed only on a radio button, checkbox, and select options.
- Example

```
WebElement element = driver.findElement(By.id("Sex-Male"));
```

```
boolean status = element.isSelected();
```

//Or can be written as

```
boolean staus = driver.findElement(By.id("Sex-Male")).isSelected();
```

## Web Elements Commands

# Web Elements Commands

### 7. submit command

- This method works well/better than the click() if the current element is a form, or an element within a form.
- This accepts nothing as a parameter and returns nothing.
- Example

```
WebElement element = driver.findElement(By.id("SubmitButton")); element.submit();
```

//Or can be written as

```
driver.findElement(By.id("SubmitButton")).submit();
```

## Web Elements Commands

### 8. `getAttribute` command

- This method gets the value of the given attribute of the element.
- This accepts the String as a parameter and returns a String value.
- Attributes are Ids, Name, Class extra and using this method you can get the value of the attributes of any given element.
- Example

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

```
String attValue = element.getAttribute("id"); //This will return value of the passed attribute id "SubmitButton"
```

## Web Elements Commands

### 9. getTagName command

- This method gets the tag name of this element.
- This accepts nothing as a parameter and returns a String value.
- This does not return the value of the name attribute but return the tag for e.g. "input" for the element `<input name="foo"/>`.
- Example

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

```
String tagName = element.getTagName();
```

//Or can be written as

```
String tagName = driver.findElement(By.id("SubmitButton")).getTagName();
```

## Web Elements Commands

### 10. getSize command

- This method fetch the width and height of the rendered element.
- This accepts nothing as a parameter but returns the Dimension object.
- This method returns the size of the element on the page.
- Example

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

```
Dimension dimensions = element.getSize();
```

```
System.out.println("Height :" + dimensions.height + "Width : "+ dimensions.width);
```

## Web Elements Commands

### 11. `getCssValue` command

- This method Fetch CSS property value of the give element.
- This accepts nothing as a parameter and returns a String value.
- Color values should be returned as rgba strings, so, for example if the "**background-color**" property is set as "**green**" in the HTML source, the returned value will be "**rgba(0, 255, 0, 1)**".
- Example

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

```
String tagName = element.getCssValue();
```

//Or can be written as

```
String tagName = driver.findElement(By.id("SubmitButton")).getCssValue();
```



## Web Elements Commands

### 12. getLocation command

- This method locate the location of the element on the page.
- This accepts nothing as a parameter but returns the Point object.
- This returns the **Point object**, from which we can get **X and Y coordinates of specific element**.
- Example

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

```
Point point = element.getLocation();
```

```
System.out.println("X coordinate : " + point.x + "Y coordinate: " + point.y);
```

## Web Elements Commands

# Web Elements Commands

### 13.sendKeys command

- This command simulates typing into an element, which may set its value.
- This method accepts CharSequence as a parameter and returns nothing.
- This method works fine with text entry elements like **INPUT** and **TEXTAREA** elements.
- Example

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

```
element.sendKeys("ToolsQA");
```

//Or can be written as

```
driver.findElement(By.id("UserName")).sendKeys("ToolsQA");
```

# Web Elements Commands – Practice

### *Practice Exercise – 1*

1. Launch the web browser and open the application under test – <https://www.google.com>
2. Verify the web page title.
3. Verify that “Google Search Box” is enabled or not.
4. Enter the text in the “Google Search” box.
5. Verify that the “Google Search button” is enabled or not.

# Web Elements Commands – Practice

### *Practice Exercise – 2*

1. Launch the web browser and open the application under test – <https://www.google.com>
2. Verify the web page title.
3. Verify that “Google Search Box” is displayed or not.
4. Enter the text in the “Google Search” box.
5. Verify that the “Google Search button” is displayed or not.

# Web Elements Commands – Practice

### *Practice Exercise – 3*

1. Launch web browser and open the application under test –  
<https://selenium08.blogspot.com/2019/07/check-box-and-radio-buttons.html>
2. Verify that “Red checkbox” is enabled or not.
3. Check that “Red checkbox” is selected or not.
4. Check that “Opera radio button” is selected or not.

# THANK YOU

# Handling Browser Window





## Window Handling Commands

### 1) SwitchTo Window :

- This command is used to switch the focus to a new browser window (or tab) by supplying the Window Name or Window Handle as an argument to the command.

- **Syntax :**

**`driver.SwitchTo().Window(WindowHandle);`**

**`driver.SwitchTo().Window(WindowName);`**

## Window Handling Commands

### 2) CurrentWindowHandle :

- This command returns the window handle (as a String) of the current window.
- **Syntax :**

**String currentWindowHandle = driver.CurrentWindowHandle;**

### 3) WindowHandles :

- The WindowHandles command gets the handles of all the open instances of the browser under test. The return type is a String ReadOnlyCollection.
- **Syntax :**

**ReadOnlyCollection <string> IWebDriver.WindowHandles { get;};**

- **Example usage of WindowHandles:**

**List <string> totalWindowHandles = driver.WindowHandles.ToList();**

## Handling Browser Window

# Program 1

```
namespace HandlingMultipleWindows
{
    internal class MultipleWindows
    {
        static void Main(string[] args)
        {
            IWebDriver driver = new ChromeDriver();
            driver.Navigate().GoToUrl("https://demoqa.com/browser-windows");
            String parentWindowHandle = driver.CurrentWindowHandle;
            Console.WriteLine("Parent window's handle -> " + parentWindowHandle);
            IWebElement clickElement1 = driver.FindElement(By.Id("tabButton"));
            for (var i = 0; i < 3; i++)
            {
                clickElement1.Click();
                Thread.Sleep(3000);
            }
            List<string> FirstWindow = driver.WindowHandles.ToList();
            foreach (var handle in FirstWindow)
            {
                Console.WriteLine(handle);
            }
        }
    }
}
```

## Handling Browser Window

### Program 2

```
namespace HandlingMultipleWindows
{
    internal class MultipleWindows
    {
        static void Main(string[] args)
        {
            IWebDriver driver = new ChromeDriver();
            driver.Navigate().GoToUrl("https://demoqa.com/browser-windows");
            String parentWindowHandle = driver.CurrentWindowHandle;
            Console.WriteLine("Parent window's handle -> " + parentWindowHandle);
            IWebElement clickElement1 = driver.FindElement(By.Id("tabButton"));
            for (var i = 0; i < 3; i++)
            {
                clickElement1.Click();
                Thread.Sleep(3000);
            }
            List<string> FirstWindow = driver.WindowHandles.ToList();
            foreach (var handle in FirstWindow)
            {
                Console.WriteLine("Switching to window - > " + handle);
                Console.WriteLine("Navigating to google.com");
                driver.SwitchTo().Window(handle);
                driver.Navigate().GoToUrl("https://google.com");
            }
        }
    }
}
```