**Selenium Class**

# Chapter#6 – Selenium WebDriver Commands

1. **Opening a URL**
2. **Using Get method-**
3. **Using Navigate method-**
4. **Clicking on WebElement**
5. **Writing in a Textbox**
6. **Clearing text in a Textbox**
7. **Fetching text written over any web element**
8. **Navigating backward in a browser**
9. **Navigating forward in a browser**
10. **Refreshing the browser**
11. **Closing browser**
12. **Sample Script**

## Opening a URL

### Using Get method

The driver.get() method is used to navigate to a web page by passing the string URL as parameter.

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

### Using Navigate method

The driver.navigate().to() method does the task of opening a web page like driver.get() method.

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

## Clicking on WebElement

The click() method in Selenium is used to perform the click operation on web elements. In our previous tutorial, Locators in Selenium WebDriver, we studied about locating the webElements in Selenium. The click() method is applied on the webElements identified, to perform the click operation.

**//Clicking an element directly**

driver.findElement(By.id("button1")).click();

**//Or by first creating a WebElement and then applying click() operation**

WebElement submitButton = driver.findElement(By.id("button2"));

submitButton.click();

## Writing in a Textbox

The sendKeys() method can be used for writing in a textbox or any element of text input type.

**//Creating a textbox webElement**

WebElement element = driver.findElement(By.name("q"));

**//Using sendKeys to write in the textbox**

element.sendKeys("ArtOfTesting!");

## Clearing text in a Textbox

The clear() method can be used to clear the text written in a textbox or any web element of text input type.

**//Clearing the text written in text fields**

driver.findElement(By.name("q")).clear();

## Fetching text written over any web element

In automation, many times we need to fetch the text written over a web element for performing some assertions or debugging. For this, we have getText() method in selenium webDriver.

**//Fetching the text written over web elements**

driver.findElement(By.id("element123")).getText();

## Navigating backward in a browser

Selenium provides navigate().back() command to move backward in the browser’s history.

**//Navigating backwards in browser**

driver.navigate().back();

## Navigating forward in a browser

Selenium provides navigate().forward() command to move forward in a browser.

**//Navigating forward in browser**

driver.navigate().forward();

## Refreshing the browser

There are multiple ways to refresh a page in Selenium WebDriver-

Using driver.navigate().refresh() command

Using sendKeys(Keys.F5) on any textbox on the webpage

## Closing browser

Selenium provides two commands to close browsers close() and quit(). The driver.close() command is used to close the browser having focus. Whereas, the driver.quite() command is used to close all the browser instances open.

**//To close the current browser instance**

driver.close();

**//To close all the open browser instances**

driver.quit();

## Sample Script

You can use the sample script below to automate the dummy webpage. The comments mentioned throughout the script will guide you through the whole automation process. Also, note that the Thread.sleep() used in the sample script is to pause the automation in between events. This Thread.sleep() is not required and is only included to provide you some time to see the automation events.

public class seleniumBasicCommands {

public static void main(String Args[]) throws InterruptedException{

//Set system property for driver executable's path

System.setProperty("webdriver.gecko.driver", "{path to the browser driver}");

//Create Firefox driver's instance

WebDriver driver = new FirefoxDriver();

//Set implicit wait of 10 seconds

//This is required for managing waits in selenium webdriver

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

//Launch sampleSiteForSelenium

driver.get("https://artoftesting.com/sampleSiteForSelenium");

//Fetch the text "This is sample text!" and print it on the console

//Use the id of the div to locate it and then fecth text using the getText() method

String sampleText = driver.findElement(By.id("idOfDiv")).getText();

System.out.println(sampleText);

//Waiting for 3 seconds just for the user to efficiently check automation

//It’s not mandatory though

Thread.sleep(3000);

//Using linkText locator to find the link and then using click() to click on it

driver.findElement(By.linkText("This is a link")).click();

Thread.sleep(3000);

//Finding textbox using id locator and then using send keys to write in it

driver.findElement(By.id("fname")).sendKeys("Kuldeep Rana");

Thread.sleep(3000);

//Clear the text written in the textbox

driver.findElement(By.id("fname")).clear();

Thread.sleep(3000);

//Clicking on button using click() command

driver.findElement(By.id("idOfButton")).click();

Thread.sleep(3000);

//Find radio button by name and check it using click() function

driver.findElement(By.name("male")).click();

Thread.sleep(3000);

//Find checkbox by cssSelector and check it using click() function

driver.findElement(By.cssSelector("input.Automation")).click();

Thread.sleep(3000);

//Using Select class for for selecting value from dropdown

Select dropdown = new Select(driver.findElement(By.id("testingDropdown")));

dropdown.selectByVisibleText("Database Testing");

Thread.sleep(50000);

//Close the browser

driver.close();

}