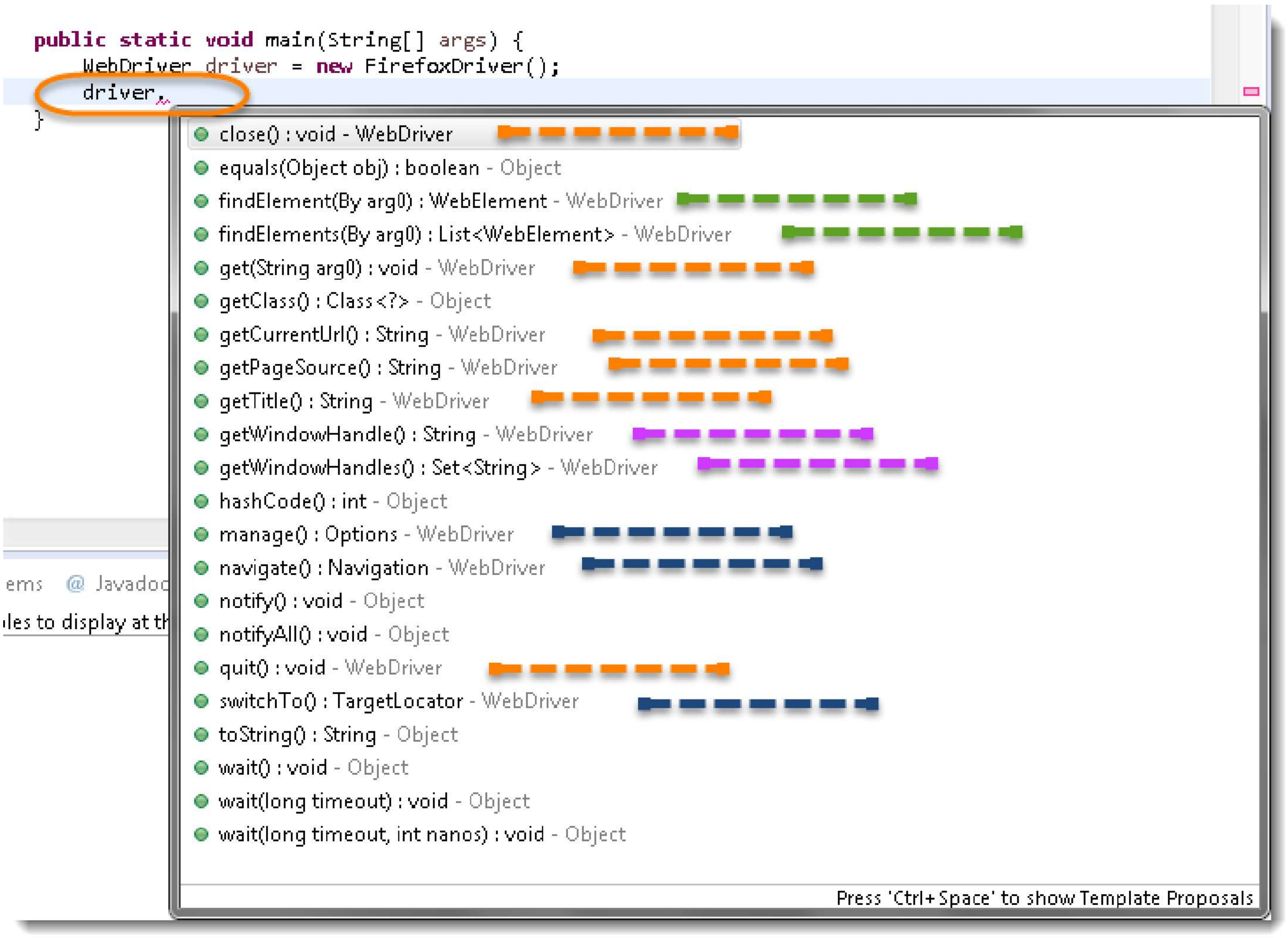
# Selenium Webdriver Browser Commands

Now the next question is, How to access the methods of WebDriver? To check what all we have in WebDriver, create a *driver object* from *WebDriver* and press ***dot key.*** This will list down all the methods of WebDriver.



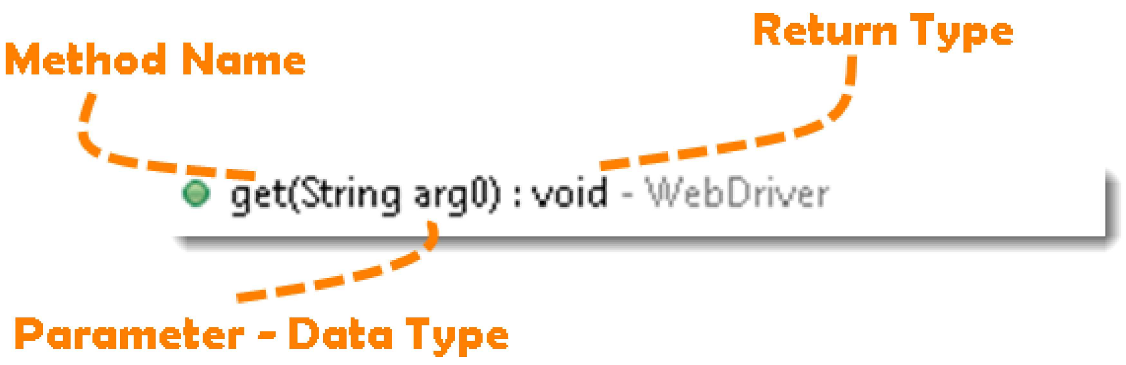
```
public static void main(String[] args) {
    WebDriver driver = new FirefoxDriver();
    driver.
}
```

```
close() : void - WebDriver
equals(Object obj) : boolean - Object
findElement(By arg0) : WebElement - WebDriver
findElements(By arg0) : List<WebElement> - WebDriver
get(String arg0) : void - WebDriver
getClass() : Class<?> - Object
getCurrentUrl() : String - WebDriver
getPageSource() : String - WebDriver
getTitle() : String - WebDriver
getWindowHandle() : String - WebDriver
getWindowHandles() : Set<String> - WebDriver
hashCode() : int - Object
manage() : Options - WebDriver
navigate() : Navigation - WebDriver
notify() : void - Object
notifyAll() : void - Object
quit() : void - WebDriver
switchTo() : TargetLocator - WebDriver
toString() : String - Object
wait() : void - Object
wait(long timeout) : void - Object
wait(long timeout, int nanos) : void - Object
Press 'Ctrl+Space' to show Template Proposals
```

***Note:*** *Methods followed by* ***Object*** *keyword are the generic methods gets from Object Class in Java. You will find these methods for every object of java language*.

*The suggestions marked in* ***Orange Color*** *are the methods directly under WebDriver and will be covering these in this chapter only*.
*The suggestions marked in* ***Blue Color*** *are Nested Classes under WebDriver and will be covered in detail separately in the following chapters*.
*The suggestions marked in* ***Green Color*** *are also Interfaces like WebDriver and will be covered in detail separately in the following chapters.*
*The suggestions marked in* ***Violet Color*** *are similar methods like* ***Orange*** *but will be covered in detail separately in the following chapters*.

Let's just start discussing the ***Orange colored*** methods of ***Selenium WebDriver*** but before that try to understand the syntax of the displayed suggestion by Eclipse for WebDriver.



**Method**: A Java method is a collection of statements that are grouped together to perform an operation.

> ***Method Name:*** *To access any method of any class, we need to create an object of a class and then all the public methods will appear for the object.*
> ***Parameter****: It is an argument that is passed to a method as a parameter to perform some operation. Every argument must be passed with the same data type. For e.g.* ***get(String arg0)*** *: void. This is asking for a* ***String type*** *argument.*
> ***Return Type****: Method can return a value or returning nothing (****void****). If the void is mentioned after the method, it means the method is returning no value. And if it is returning any value, then it must display the type of the value for e.g. getTitle() : String*.

Now it would be very easy to understand the WebDriver commands in the below chapter. The very first thing you like to do with Selenium is to ***Opening*** a new browser, ***Perform*** a few tasks and ***Closing*** the browser. Below are the numbers of commands you can apply on the Selenium opened browser.

## *Get-Command - How to Open a WebPage in Selenium?*

***get(String arg0)***: ***void*** - This method ***Load*** a new web page in the current browser window. Accepts String as a parameter and returns nothing.

***Command - driver.get(appUrl);***

Where ***appUrl*** is the website address to load. It is best to use a fully qualified URL.

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

//Or can be written as
```

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

## *Get Title Command - How to get the Title of the Webpage in Selenium?*

***getTitle(): String*** - This method fetches the ***Title*** of the current page. Accepts nothing as a parameter and returns a String value.

***Command - driver.getTitle();***

As the return type is a String value, the output must be stored in String object/variable.

```
driver.getTitle();

//Or can be used as

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

## *Get Current URL Command - How to read the URL of the Webpage in Selenium?*

***getCurrentUrl(): String*** - This method fetches the string representing the ***Current URL***which is opened in the browser. Accepts nothing as a parameter and returns a String value.

***Command - driver.getCurrentUrl();***

As the return type is String value, the output must be stored in String object/variable.

```
driver.getCurrentUrl();

//Or can be written as

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

## *Get Page Source Command - How to read the page source of the WebPage in Selenium?*

***getPageSource(): String*** - This method returns the ***Source Code*** of the page. Accepts nothing as a parameter and returns a String value.

***Command - driver.getPageSource();***

As the return type is String value, the output must be stored in String object/variable.

```
driver.getPageSource();

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

## *Close Command - How to close the Browser in Selenium?*

***close(): void*** - This method Close only the current window the WebDriver is currently controlling. Accepts nothing as a parameter and returns nothing.

***Command - driver.close();***

Quit the browser if it's the last window currently open.

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

## *Quit Command - How to close all the Browser's Window in Selenium?*

***quit(): void*** - This method ***Closes*** all windows opened by the *WebDriver*. Accepts nothing as a parameter and returns nothing.

***Command - driver.quit();***

Close every associated window.

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

***Note****: Important to note that this command will only close the browser's window opened by the selenium in the same session. If any browser is opened manually, this will have no impact on the* *same. Also, there is no impact on the browsers opened in another run or session even by the Selenium. The same is with close() method as well.*

## Selenium WebDriver Browser Command - Practice Exercises

### *Practice Exercise - 1*

- *Launch a new Chrome browser.*
- *Open Shop.DemoQA.com*
- *Get Page Title name and Title length*
- *Print Page Title and Title length on the Eclipse Console.*
- *Get Page URL and verify if it is a correct page opened*
- *Get Page Source (HTML Source code) and Page Source length*
- *Print Page Length on Eclipse Console.*
- *Close the Browser.*

#### *Solution*

```
package automationFramework;

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

public class WebDriverCommands {

        public static void main(String[] args) {

                String driverExecutablePath = "D:\\Drivers\\chromedriver.exe";
                System.setProperty("webdriver.chrome.driver", driverExecutablePath);
                // Create a new instance of the FireFox driver
                WebDriver driver = new ChromeDriver();

                // Storing the Application Url in the String variable
                String url = "https://www.shop.demoqa.com";

                //Launch the ToolsQA WebSite
                driver.get(url);

                // Storing Title name in the String variable
                String title = driver.getTitle();

                // Storing Title length in the Int variable
                int titleLength = driver.getTitle().length();

                // Printing Title & Title length in the Console window
                System.out.println("Title of the page is : " + title);
                System.out.println("Length of the title is : "+ titleLength);
                // Storing URL in String variable
                String actualUrl = driver.getCurrentUrl();

                if (actualUrl.equals(url)){
                        System.out.println("Verification Successful - The correct Url is
                }
                else {
                        System.out.println("Verification Failed - An incorrect Url is ope

                        //In case of Fail, you like to print the actual and expected URL
                        System.out.println("Actual URL is : " + actualUrl);
                        System.out.println("Expected URL is : " + url);
                }

                // Storing Page Source in String variable
                String pageSource = driver.getPageSource();

                // Storing Page Source length in Int variable
                int pageSourceLength = pageSource.length();

                // Printing length of the Page Source on console
                System.out.println("Total length of the Pgae Source is : " + pageSourceLe

                //Closing browser
                driver.close();
                }

        }
```

***Output***

*Title of the page is : ToolsQA Demo Site – ToolsQA – Demo E-Commerce Site Length of the title is : 50 Verification Failed - An incorrect Url is opened. Actual URL is : http://shop.demoqa.com/ Expected URL is : http://www.shop.demoqa.com Total length of the Pgae Source is : 88952*

## *Practice Exercise - 2*

. *Launch a new Chrome browser.*
. *Open ToolsQA Practice Automation Page for Switch Windows:* ***https://www.toolsqa.com/automation-practice-switch-windows/***
. *Use this statement to click on a New Browser Window button "**driver.findElement(By.id("New Browser Window")).click();**"*
. *Close the browser using close() command*

You will notice that only one window will close. Next time use quit() command instead of close(). At that time selenium will close both the windows.

```java
package automationFramework;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class WebDriverCommands_2 {

        public static void main(String[] args) {


                String driverExecutablePath = "D:\\Drivers\\chromedriver.exe";
                System.setProperty("webdriver.chrome.driver", driverExecutablePath);
                // Create a new instance of the FireFox driver
                WebDriver driver = new ChromeDriver();

                // Storing the Application Url in the String variable
                String url = "https://www.toolsqa.com/automation-practice-switch-windows/

                //Launch the ToolsQA WebSite
                driver.get(url);


        }

}
```