

# Here is usages of WebDriver commands

## 1. Creating New Instance Of Firefox Driver

```
WebDriver driver = new FirefoxDriver();
```

## 2. Command To Open URL In Browser

```
driver.get("http://www.habeshaschool.com");
```

## 3. Locating elements : By ID, Name, CssSelector, LinkText, PartialLinkText, Xpath, TagName

Clicking on any element or button of webpage by Id is

```
driver.findElement(By.id("submitButton")).click();
```

## 4. Store text of targeted element in variable

```
String dropdown = driver.findElement(By.tagName("select")).getText();
```

## 5. Typing text in text box or text area.

```
driver.findElement(By.name("fname")).sendKeys("My First Name");
```

## 6. Applying Implicit wait in webdriver

```
driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);
```

## 7. Applying Explicit wait in webdriver with WebDriver canned conditions.

```
WebDriverWait wait = new WebDriverWait(driver, 15);
```

```
wait.until(ExpectedConditions.textToBePresentInElementLocated(By.xpath("//div[@id='timeLeft']"), "Time left: 7 seconds"));
```

## 8. Get page title in selenium webdriver

```
driver.getTitle();
```

## 9. Get Current Page URL In Selenium WebDriver

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

## 10. Get domain name using java script executor

```
JavascriptExecutor javascript = (JavascriptExecutor) driver;
```

```
String CurrentURLUsingJS=(String)javascript.executeScript("return document.domain");
```

## 11. Generate alert using webdriver's java script executor interface

```
JavascriptExecutor javascript = (JavascriptExecutor) driver;
```

```
javascript.executeScript("alert('Test Case Execution Is started Now..');");
```

## 12. Selecting or Deselecting value from drop down in selenium webdriver.

- Select By Visible Text

```
Select mydrpdwn = new Select(driver.findElement(By.id("Carlist")));
```

```
mydrpdwn.selectByVisibleText("Audi");
```

- Select By Value

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']"));
```

```
listbox.selectByValue("Italy");
```

- Select By Index

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']"));
```

```
listbox.selectByIndex(0);
```

- Deselect by Visible Text

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));
listbox.deselectByVisibleText("Russia");
```

- Deselect by Value

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));
listbox.deselectByValue("Mexico");
```

- Deselect by Index

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));
listbox.deselectByIndex(5);
```

- Deselect All

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));
listbox.deselectAll();
```

It will remove all selections from list box of software application's page.

- isMultiple()

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));
boolean value = listbox.isMultiple();
```

### 13. Navigate to URL or Back or Forward in Selenium Webdriver

```
driver.navigate().to("http://only-testing-blog.blogspot.com/2014/01/textbox.html");
driver.navigate().back();
driver.navigate().forward();
```

### 14. Verify Element Present in Selenium WebDriver

```
Boolean iselementpresent = driver.findElements(By.xpath("//input[@id='text2']")).size() != 0;
```

### 15. Capturing entire page screenshot in Selenium WebDriver

```
File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(screenshot, new File("D:\\screenshot.jpg"));
```

### 16. Action classes : build() and perform() methods. To mouse move/hover and key events In WebDriver

```
Actions action = new Actions(driver);
WebElement moveonmenu = driver.findElement(By.xpath("//div[@id='menu1']/div"));
action.moveToElement(moveonmenu).moveByOffset(50, 100).build().perform();
action.dragAndDrop(SoruceElement, DestiantionElement).build().perform();
action.contextClick(element).build().perform();
action.keyDown(targetElement, Keys.CONTROL.ARROW_DOWN);
action.keyUp(targetElement, Keys.CONTROL);
```

### 17. Robot class is used to take mouse and key interactions with the operating system like window popup for download confirmation box.

## 18. Handling Multiple Windows In Selenium WebDriver.

1. Get All Window Handles.
2. `Set<String> AllWindowHandles = driver.getWindowHandles();`
3. Extract parent and child window handle from all window handles.
4. `String window1 = (String) AllWindowHandles.toArray()[0];`  
`String window2 = (String) AllWindowHandles.toArray()[1];`
5. Use window handle to switch from one window to other window.
6. `driver.switchTo().window(window2);`

## 18. Check Whether Element is Enabled Or Disabled In Selenium Web driver.

```
boolean fname = driver.findElement(By.xpath("//input[@name='fname']")).isEnabled();  
System.out.print(fname);
```

## 19. Enable/Disable Textbox During Selenium Webdriver Test Case Execution.

```
JavascriptExecutor javascript = (JavascriptExecutor) driver;  
String todisable = "document.getElementsByName('fname')[0].setAttribute('disabled', '');";  
javascript.executeScript(todisable);  
String toenable = "document.getElementsByName('lname')[0].removeAttribute('disabled');";  
javascript.executeScript(toenable);
```

## 20. Selenium WebDriver Assertions With TestNG Framework

- assertEquals

```
Assert.assertEquals(actual, expected);
```

- assertNotEquals

```
Assert.assertNotEquals(actual, expected);
```

- assertTrue

```
Assert.assertTrue(condition);
```

- assertFalse

```
Assert.assertFalse(condition);
```

## 21. Submit() method to submit form

```
driver.findElement(By.xpath("//input[@name='Company']")).submit();
```

## 22. Handling Alert, Confirmation and Prompts Popups

```
String myalert = driver.switchTo().alert().getText();  
driver.switchTo().alert().accept();  
driver.switchTo().alert().dismiss();  
driver.switchTo().alert().sendKeys("This Is John");
```

## 23. Robots