# Selenium Tips and Tricks



## How to set language in profile

```
profile.setPreference( "intl.accept_languages", "no,en-us,en" );
```

## Using Drag and Drop

It may not be immediately obvious, but if you're using a browser that supports it , you can use Action classes and then it's easy to do drag and drop:

```
 Actions builder = new Actions(driver);
   Action dragAndDrop = builder.clickAndHold(someElement)
       .moveToElement(otherElement)
       .release(otherElement)
       .build();
   dragAndDrop.perform();
```

## How to execute JS

This code will inject a JS in the element which is not visible:

```
public static void displayElementWhichIsNotVisible(By findBy) {
 WebElement element = null;
 JavascriptExecutor js = (JavascriptExecutor) Page.webdriver;
 element = Page.webdriver.findElement(findBy);
 js.executeScript("arguments[0].setAttribute('style',
arguments[1]);",element,"display='block';color: red; border: 2px solid yellow;");
}
```

## Capture JS errors from Firefox

```
package tipsAndTricks;
import java.io.IOException;
import java.util.List;
import net.jsourcerer.webdriver.jserrorcollector.JavaScriptError;
 import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxProfile;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

public class CaptureJavascriptErrors {
private static WebDriver driver;

@BeforeClass public void setUp() throws IOException {
 FirefoxProfile ffProfile = new FirefoxProfile();
 JavaScriptError.addExtension(ffProfile);
 driver = new FirefoxDriver(ffProfile);
 driver.get("http://www.telegraaf.nl/"); }

@AfterClass public void tearDown() {
 List jsErrors = JavaScriptError.readErrors(driver);
 System.out.println("###start displaying errors");
 for(int i = 0; i < jsErrors.size(); i++) {
 System.out.println(jsErrors.get(i).getErrorMessage());
 System.out.println(jsErrors.get(i).getLineNumber());
 System.out.println(jsErrors.get(i).getSourceName()); }
 System.out.println("###start displaying errors");
 driver.close(); driver.quit(); }

@Test public void returnJavascriptErrors() throws InterruptedException {
 Thread.sleep(5000);
} }
```

## Highlight elements

If you want to highlight elementes during the execution of test you can look over the below method

```
public void highlightElement(WebElement element) {
 for (int i = 0; i < 2; i++) {
 JavascriptExecutor js = (JavascriptExecutor) driver;
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element,
"color: yellow; border: 2px solid yellow;");
js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, "");
} }
```

You can call the method before calling the action method for that element:

```
highlightElement(locator);
Element.click(locator);
```