JavascriptExecutor Interface:

JavascriptExecutor is an interface in Selenium which helps to execute Javascript script through Selenium WebDriver.It has two abstract methods:

1. executeScript()
2. executeAsyncScript()
3. executeScript():

This method executes Javascript script in the context of currently selected window/frame. The script provided will be executed as the body of anonymous function.

1. executeAsyncScript():

This method executes asynchronous piece of Javascript code in the context of current selected window/frame.

Steps to execute Javascript code through Selenium WebDriver:

Step 1: Typecast WebDriver reference to JavascriptExecutor interface.

JavascriptExecutor js = (JavascriptExecutor)driver;

Step 2: Call one of the JavascriptExecutor methods using JavascriptExecutor reference

js.executeScript(script,args);

Different actions performed using Javascript:

1. Scroll action

* Using hardcoded coordinates
* Using location of web element in the web page
* Using element reference

1. Navigation to the application
2. Passing the data to text field
3. Click action
4. Fetching the title of current the web page
5. Fetching the URL of current the web page
6. Refreshing the current web page
7. Handling disabled elements

Code snippets for different actions:

1. Scroll action:

* Using hardcoded coordinates

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("window.scrollBy(0,5000)");

* Using location of the web element in the web page

WebElement element =driver.findElement(locator);

Point p = element.getLocation();

**int** x = p.getX();

**int** y = p.getY();

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("window.scrollBy("+x+","+y+")");

* Using element reference

WebElement element =driver.findElement(locator));

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("arguments[0].scrollIntoView(true)",

element);

1. Navigation to the application:

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("window.location=arguments[0]", url);

1. Passing the data to the text field:

JavascriptExecutor js = (JavascriptExecutor) driver;

WebElement element = driver.findElement(locator);

js.executeScript("arguments[0].value=arguments[1]",

element, text);

1. Click action:

JavascriptExecutor js = (JavascriptExecutor) driver;

WebElement element = driver.findElement(locator);

js.executeScript("arguments[0].click()", element);

1. Fetching the title of the current web page:

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.location=arguments[0]", url);

String title = js.executeScript("return document.title");

1. Fetching the URL of the current web page:

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.location=arguments[0]", url);

String url = js.executeScript("return document.URL");

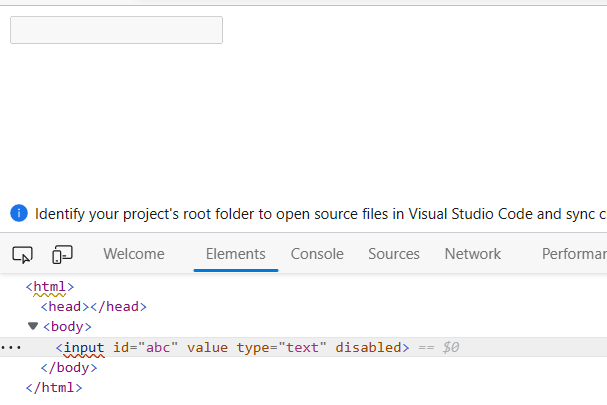
1. Refreshing the current web page:

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.location=arguments[0]", url);

js.executeScript("history.go(0)");

1. Handling disabled elements:

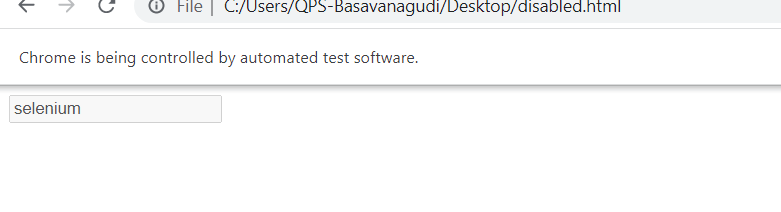


WebElement element =driver.findElement(By.*id*("abc"));

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("document.getElementById('abc').value

='selenium'");



To clear the text field:

js.executeScript("document.getElementById('abc').value

=''");