# JavaScriptExecutor in Selenium WebDriver

JavaScriptExecutor is an Interface that helps to execute JavaScript through Selenium Webdriver. JavaScriptExecutor provides two methods "executescript" & "executeAsyncScript" to run javascript on the selected window or current page.

## Why do we need JavaScriptExecutor?

In Selenium Webdriver, locators like XPath, CSS, etc. are used to identify and perform operations on a web page.

In case, these locators do not work you can use JavaScriptExecutor. You can use JavaScriptExecutor to perform an desired operation on a web element.

Selenium supports javaScriptExecutor. There is no need for an extra plugin or add-on. You just need to import (**org.openqa.selenium.JavascriptExecutor**) in the script as to use JavaScriptExecutor.

## JavaScriptExecutor Methods

1. **executeAsyncScript**

With Asynchronous script, your page renders more quickly. Instead of forcing users to wait for a script to download before the page renders. This function will execute an asynchronous piece of JavaScript in the context of the currently selected frame or window in Selenium. The JS so executed is single-threaded with a various callback function which runs synchronously.

1. **executeScript**

This method executes JavaScript in the context of the currently selected frame or window in Selenium. The script used in this method runs in the body of an anonymous function (a function without a name). We can also pass complicated arguments to it.

The script can return values. Data types returned are

* Boolean
* Long
* String
* List
* WebElement.

The basic syntax for JavascriptExecutor is given below:

**Syntax:**

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

* Script – This is the JavaScript that needs to execute.
* Arguments – It is the arguments to the script. It's optional.
* ***Returns –***One of Boolean, Long, String, List, WebElement, or null.

Commands

* To type Text in Selenium WebDriver without using sendKeys() method

js.executeScript("document.getElementById('some id').value='someValue';");

js.executeScript("document.getElementById('Email').value='SoftwareTestingMaterial.com';");

* To click a Button in Selenium WebDriver using JavaScript

js.executeScript("document.getElementById('enter your element id').click();");

//or

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

* To handle Checkbox

js.executeScript("document.getElementById('enter element id').checked=false;");

* To generate Alert Pop window in selenium

js.executeScript("alert('Welcome To SoftwareTestingMaterial');");

* To refresh browser window using Javascript

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

* To get innertext of the entire webpage in Selenium

String sText =  js.executeScript("return document.documentElement.innerText;").toString();

System.out.println(sText);

* To get the Title of our webpage

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

System.out.println(sText);

* To get the domain

String sText =  js.executeScript("return document.domain;").toString();

System.out.println(sText);

* To get the URL of a webpage

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

System.out.println(sText);

* To perform Scroll on application using  Selenium

//Vertical scroll - down by 500  pixels

js.executeScript("window.scrollBy(0,500)");

// for scrolling till the bottom of the page we can use the code like

//js.executeScript("window.scrollBy(0,document.body.scrollHeight)");