**Javascript Executor in Selenium**

A JavaScript is a small chunk of a program that makes a website interactive. For example, a JavaScript could create a pop-up alert box, or provide a dynamic dropdown menu.

JavaScript code can listen to even on the web page, like reacting to a click on a button, reacting to when checking a checkbox, or when we enter any value into the text bar.

Javascript can change the content of the HTML page dynamically; it can also change the attributes of a web element like changing a button from the disabled state to enabled state.

JavascriptExecutor is something that is present in every selenium tool, and in all the languages, we can execute it on the browser.

We should go for Javascript Executor only when we are not able to perform a particular task with our selenium; like some, we may not be able to click an element, such cases we can go for Javascript Executor.

JavascriptExecutor is an interface present under org.openqa.selenium package in Selenium Webdriver. JavascriptExecutor Interface has two abstract methods which are:

1. executeScript()

2. executeAsyncScript()

How to use Javascript Executor in Selenium Webdriver:

We have to cast driver object into JavascriptExecutor type to use the methods present in the JavascriptExecutor interface.

**// cast the driver object to JavascriptExecutor**

**JavascriptExecutor js = (JavascriptExecutor) driver;**

**// access the methods**

**js.executeScript("javascript command");**

**js.executeAsyncScript("javascript command");**

executeScript():

The executeScript method executes given JavaScript in the current frame if the control is inside the frame; if the control is on page level (window level, i.e. outside frames), javascript will be executed in the current frame.

All the local variables created inside the javascript block will be destroyed once the block execution is over, but if the variable is created globally, it will be present for other javascript code.

JavascriptExecutor just runs the javascript in the browser, but it will not return any value by default, if we want to return any value from the javascript block, then we must use return statement along with our javascript.

We can manipulate the returned value based on our need like asserting or to check whether it contains something.

If Javascript code has a return statement, then it could return one of the following:

1. This method returns a WebElement if the javascript command results in an HTML element.

2. A Double value is returned if the javascript command results in Decimal value.

3. A Long value will be returned if the javascript command results in a non-decimal number value.

4. A Boolean is returned if the query results in a boolean value.

5. For all other primary values, String will be returned.

6. List<Object> will be returned if the query results in an array. It also supports nested lists. Lists must fall under above one of the category (WebElement, Decimal value, Long, Boolean, String).

7. Map<String, Object> will be returned if the query results in Map and values in the map must fall under the above category(WebElement, Decimal value, Long, Boolean, String).

* executeScript() the result is returned directly,
* executeAsyncScript() returns the result through a callback method, input to the script as the final input argument.

The main difference between two are that scripts executed with async must explicitly signal they are finished by invoking the provided callback. This callback is always injected into the executed function as the last argument.

<https://intellipaat.com/community/9619/webdriver-executeasyncscript-vs-executescript>

<http://makeseleniumeasy.com/2018/02/19/part-5-usages-of-javascripts-in-selenium-understanding-method-executeasyncscript-of-javascriptexecutor/>

window.scrollBy(0,300) is the javascript command for scrolling a page, scrollBy(horizontalDistance, verticalDistance) accepts two parameter.

1. horizontalDistance - distance to be scrolled horizontally

2. verticalDistance - distance to be scrolled vertically

We have to set 'hozontalDistance=0' for scrolling vertically, vice-versa