**What is JavascriptExecutor interface in selenium?**

- JavascriptExecutor it an interface in selenium WebDriver.

- it is used to perform operation web page object and browser based operations.

- JavascriptExecutor it helps to execute javascript language through selenium WebDriver.

- JavascriptExecutor it define 2 methods

1) executeScript() method

2) executeAsyncScript() method

- we use executeScript() method to perform operation web page

- by using JavascriptExecutor interface we can perform different operation as like

i) click on element

ii) send text to the text box

iii) handle the drop down

iv) open a url

v) we can navigate back

vi) we can navigate forward

vii) we can refresh the browser

viii) capture the title

ix) capture URL

x) change the element background colour

xi) scroll up and scroll down

xii) generate the alert pop up

etc..

--------

**i) click on element**

there are 2 ways we can click on element by using JavascriptExecutor interface

**i) by using arguments[0].click() method**

**ii) by using getElementById('id value').click() method**

**-------------------------------------------------------**

**i) by using arguments[0].click() method**

- find the web element in HTML DOM structure.

WebElement wb = driver.findElement(By.name("name value"));

- Convert WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing arguments[0].click() method and WebElement instance from JavascriptExecutor interface

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

Scenario 1:

open a url https://demoqa.com/automation-practice-form

find and click on female radio button using JavascriptExecutor interface

Step 1: connect to actual browser

Step 2: up casting

WebDriver driver =new ChromeDriver();

step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

step 4: find the female radio button

WebElement wb = driver.findElement(By.id("gender-radio-2"));

step 5: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 6: use executeScript() method by passing arguments[0].click() and WebElement instance

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

package JavascriptExecutorTutorial1;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

public static void main(String[] args) throws InterruptedException {

// Step 1: connect to actual browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

// step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

Thread.sleep(5000);

// step 4: find the female radio button

WebElement wb = driver.findElement(By.id("gender-radio-2"));

// step 5: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 6: use executeScript() method by passing arguments[0].click() and

// WebElement instance

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

}

}

-------------------------------------------------------

**ii) by using getElementById('id value').click() method**

- document.getElementById().click() method it will find th element in web page and it will click on it.

- if u want to use getElementById() method then inside HTML DOM structure id should be present.

syntax

- convert the WebDriver Object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing document.getElementById('id value').click() method.

js.executeScript("document.getElementById('id value').click();");

Scenario:

open a url https://demoqa.com/automation-practice-form

find the element and click on it using JavascriptExecutor

Step 1: connect to actual browser

Step 2: up casting

WebDriver driver =new ChromeDriver();

Step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 5: use executeScript() method by passing document.getElementById('id value').click() method

js.executeScript("document.getElementById('gender-radio-3').click();");

package JavascriptExecutorTutorial1;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

public static void main(String[] args) throws InterruptedException {

// Step 1: connect to actual browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

// step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

Thread.sleep(5000);

// step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 5: use executeScript() by passing document.getElementById('id

// value').click();

js.executeScript("document.getElementById('gender-radio-3').click();");

}

}

=====

**2) Enter value in text box using JavascriptExecutor interface**

there are 2 ways we can enter value in text box using JavascriptExecutor interface

i) arguments[0].value='expected text' attribute

ii) document.getElementById('id value').value='expected value' attribute

i) arguments[0].value='expected text' attribute

- find the element in web page using findElement() method

WebElement wb = driver.findElement(By.id(""));

- convert the WebDriver Object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing arguments[0].value='Anjali' and WebElement instance

js.executeScript("arguments[0].value='Anjali';", wb);

Scenario 1:

open a https://demoqa.com/automation-practice-form

find first name text box and enter Anjali value using JavascriptExecutor interface.

Step 1:connect to actual browser

step 2: up casting

WebDriver driver =new ChromeDriver();

Step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

step 4: find the first name text box

WebElement wb = driver.findElement(By.id('firstName'));

step 5: convert the WebDriver object into JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 6: use executeScript() method by passing arguments[0].value='expected value' and WebElement instance

js.executeScript("arguments[0].value='Anjali';", wb);

package JavascriptExecutorTutorial1;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

public static void main(String[] args) throws InterruptedException {

// Step 1: connect to actual browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

// step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

Thread.sleep(5000);

// step 4: find first name text box

WebElement wb = driver.findElement(By.id("firstName"));

// step 5: convert WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 6: use executeScript() method by passing arguments[0].value='Anjali' and

// WebElement instance

js.executeScript("arguments[0].value='Anjali';", wb);

}

}

---------------------------------------------------------------------------------------------------------------------

**ii) document.getElementById('id value').value='expected value' attribute**

- document.getElementById().value attribute it will find the element in web page and it will enter the text

- if u want to use getElementById() method then inside HTML DOM structure id should be present.

- convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing document.getElementById('id value').value='Expected value';

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

Scenario 2:

open a https://demoqa.com/automation-practice-form

find and enter value using JavascriptExecutor

Step 1: connect the actual browser

Step 2: up casting

WebDriver driver =new ChromeDriver();

step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 5: use executeScript() method by passing document.getElementById('id value').value='Expected value';

js.executeScript("document.getElementById('firstName').value='Pooja';");

package JavascriptExecutorTutorial1;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo4 {

public static void main(String[] args) throws InterruptedException {

// Step 1: connect to actual browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

// step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

Thread.sleep(5000);

//step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

//step 5: use executeScript() method by passing document.getElementById('id value').value='Expected value'

js.executeScript("document.getElementById('firstName').value='Pooja';");

}

}

==========================================================================================

**3) open a url using JavascriptExecutor interface.**

-if u want to open a url by using JavascriptExecutor interface then use window.location='expected url' attribute.

- first we have to convert WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- then use executeScript() method by passing window.location='expected value' attribute

js.executeScript("window.location='URL';");

Scenario 1;

open facebook url using JavascriptExecutor

step 1: connect browser

step 2: up casting

WebDriver driver = new ChromeDriver();

step 3: convert WebDriver Object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 4: use executeScript() method by passing window.location='expected url' attribute

js.executeScript("window.location='https://www.facebook.com';");

package JavascriptExecutorTutorial1;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo5 {

public static void main(String[] args) {

// Step 1: connect to actual browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

// step 3: convert WebDriver object into JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 4: use executeScript() method by passing window.location='url';

js.executeScript("window.location='https://www.facebook.com';");

}

}

-----------------------------------------------------------------------------------------

**4) open a url in new tab using JavascriptExecutor interface.**

- if u want to open a new tab with url by using JavascriptExecutor then we use window.open('expected url') method

- convert the WebDriver object into JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing window.open('expected url')

js.executeScript("window.open('url');");

Scenario:

open a google url

and open new tab with fb url using JavascriptExecutor

Step 1: connect browser

Step 2: up casting

WebDriver driver =new ChromeDriver();

step 3: open url

driver.get("https://www.google.com");

step 4: convert the WebDriver object into JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 5: use executeScript()method by passing window.open('expected url') method

js.executeScript("window.open('https://www.facebook.com');");

package JavascriptExecutorTutorial1;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo6 {

public static void main(String[] args) throws InterruptedException {

// Step 1: connect to actual browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

// open a url

driver.get("https://www.google.com");

Thread.sleep(5000);

// convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// use executeScript() method by passing window.open('expected url')

js.executeScript("window.open('https://www.facebook.com');");

}

}

========================================================================

incognito mode

open a https://www.saucedemo.com/ using JavascriptExecutor

maximize

implicitly wait

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

explicit wait

take screenshot

enter firstname and last name using JavascriptExecutor

click on login using JavascriptExecutor

click 3 add to cart button using JavascriptExecutor

click on container using JavascriptExecutor

click on checkout button JavascriptExecutor

enter first name, last name and pin code using JavascriptExecutor

click on continue button JavascriptExecutor

click on finish using JavascriptExecutor

click on back using JavascriptExecutor

click on menu JavascriptExecutor

take full screenshot using AShot maven dependency

click on logout

========================================================================

**click on element**

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

js.executeScript("document.getElementById('id value').click();");

enter text in text box

js.executeScript("arguments[0].value='Expected value';", wb);

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

open a url

js.executeScript("window.location='Expected URL';");

open url in new tab

js.executeScript("window.open('Expected url');");

1) how to click on element using JavascriptExecutor VVVVVIMMPPPPPPPPPP

2) how to enter text in text box using JavascriptExecutor VVVVVIMMPPPPPPPPPP

3) how to perform scroll up and scroll down using JavascriptExecutor VVVVVIMMPPPPPPPPPP

4) how to create border for element using JavascriptExecutor VVVVVIMMPPPPPPPPPP

5) how to handle drop down using JavascriptExecutor VVVVVIMMPPPPPPPPPP

**JavascriptExecutor interface:**

**1) How to navigate back in browser history by using JavascriptExecutor**

- convert the WebDriver object into the JavascriptExecutor

- use executeScript() method by passing history.go(negative int value);

- if u want to navigate one step back in browser history then we use -1 value inside the history.go() method.

- if u want to navigate two step back in browser history then we use -2 value inside the history.go() method.

- if u want to navigate three step back in browser history then we use -3 value inside the history.go() method.

syntax:

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("history.go(-1);"); //it will navigate one step back in browser history

js.executeScript("history.go(-2);"); //it will navigate two step back in browser history

js.executeScript("history.go(-3);"); //it will navigate three step back in browser history

**2) How to navigate forward in browser history by using JavascriptExecutor**

- convert the WebDriver object into the JavascriptExecutor

- use executeScript() method by passing history.go(positive values int value);

- if u want to navigate one step forward in browser history then we use +1 value inside the history.go() method.

- if u want to navigate two step forward in browser history then we use +2 value inside the history.go() method.

- if u want to navigate three step forward in browser history then we use +3 value inside the history.go() method.

syntax:

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("history.go(+1);"); //it will navigate one step forward in browser history

js.executeScript("history.go(2);"); //it will navigate two step forward in browser history

js.executeScript("history.go(3);"); //it will navigate three step forward in browser history

**3) How to navigate refresh the browser by using JavascriptExecutor**

- convert the WebDriver object into the JavascriptExecutor

- use executeScript() method by passing history.go(0);

syntax:

JavascriptExecutor js = (JavascriptExecutor)driver;

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

Scenario:

- Open a google application

- navigate to facebook application

- navigate to red bus application

- navigate to amazon application

- navigate one step back in browser history

- navigate one step forward in browser history

- refresh the browser using JavascriptExecutor

- navigate 2 step back in browser history

- navigate 2 step forward in browser history

- navigate 3 step back browser history

- navigate 3 step forward in browser history

- refresh the browser

Step 1: connect to actual browser

step 2: up casting concept

WebDriver driver =new Chromedriver();

step 3: open a google url

driver.get("https://www.google.com");

step 4: navigate to facebook application

driver.navigate().to("https://www.facebook.com");

step 5: navigate to redbus application

driver.navigate().to("https://www.redbus.com");

step 6: navigate to amazon application

driver.navigate().to("https://www.amazon.in");

step 7: convert the WebDriver Object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 8: navigate one step back in browser history

js.executeScript("history.go(-1);");

step 9: navigate one step forward in browser history

js.executeScript("history.go(1);");

step 10: refresh the browser or reload the browser

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

step 11: navigate 2 step back in browser history

js.executeScript("history.go(-2);");

step 12: navigate 2 step forward in browser history

js.executeScript("history.go(+2);");

step 13: navigate 3 step back in browser history

js.executeScript("history.go(-3);");

step 14: navigate 3 step forward in browser history

js.executeScript("history.go(+3);");

step 15: refresh the browser or reload the browser

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

package JavascriptExecutorTutorial2;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

public static void main(String[] args) throws InterruptedException {

// step 1: connect browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

// step 3: open a url

driver.get("https://www.google.com");

Thread.sleep(3000);

// step 4: navigate to facebook url

driver.navigate().to("https://facebook.com");

Thread.sleep(3000);

// step 5: navigate to redbus url

driver.navigate().to("https://redbus.com");

Thread.sleep(3000);

// step 6: navigate to facebook url

driver.navigate().to("https://amazon.in");

// step 7: convert the WebDriver Object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 8: navigate one step back browser history by using executeScript()

// method and by passing history.go(-1) keyword.

Thread.sleep(3000);

js.executeScript("history.go(-1);");

Thread.sleep(3000);

// step 9: navigate one step forward browser history by using executeScript()

// method and by passing history.go(1) keyword

js.executeScript("history.go(1);");

Thread.sleep(3000);

// step 10: refresh the browser by using executeScript() method and by passing

// history.go(0) keyword

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

// step 11: navigate 2 step back in browser history by using

// executeScript()method and by passing history.go(-2)

js.executeScript("history.go(-2);");

Thread.sleep(3000);

// step 12: navigate 2 step forward in browser history by using executeScript()

// method and

// by passing history.go(2) keyword.

js.executeScript("history.go(+2);");

Thread.sleep(3000);

// step 13: navigate 3 step back in browser history by using

// executeScript()method and by passing history.go(-3)

js.executeScript("history.go(-3);");

Thread.sleep(3000);

// step 14: navigate 3 step forward in browser history by using executeScript()

// method and

// by passing history.go(3) keyword.

js.executeScript("history.go(+3);");

Thread.sleep(3000);

// step 15: refresh the browser by using executeScript() method and by passing

// history.go(0) keyword

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

}

}

==============================================================================================

How to navigate 10 step back in browser history?

How to navigate 10 step forward in browser history?

==================================================================================**============**

**4) how to capture browser title using JavascriptExecutor**

- convert the WebDriver Object into JavascriptExecutor

- use executeScript() by passing return document.title keyword and then use toString() method to convert to captured values to String.

- return type of toString() method is String.

syntax:

JavascriptExecutor js = (JavascriptExecutor)driver;

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

**5) how to capture browser URL using JavascriptExecutor**

- convert the WebDriver Object into the JavascriptExecutor

- use executeScript() method by passing return document.URL keyword and then use toString() method to convert to captured values to string.

- return type of toString() method is String.

syntax:

JavascriptExecutor js = (JavascriptExecutor)driver;

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

Scenario:

open a facebook application

capture title and url using JavascriptExecutor

Step 1: connect browser

step 2: up casting

step 3: open facebook url

driver.get("https://www.facebook.com");

step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 5: capture title using JavascriptExecutor

we use executeScript() method by passing return document.title keyword and then we use toString() method.

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

step 6: capture url using JavascriptExecutor

we use executeScript() method by passing return document.URL keyword and then we use toString() method

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

package JavascriptExecutorTutorial2;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

public static void main(String[] args) {

// step 1: connect browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

// step 3: open a url

driver.get("https://www.facebook.com");

// step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 5: use executeScript() method by passing return document.title keyword

// and then use toString() method

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

System.out.println(abc);

// step 6: use executeScript() method by passing return document.URL keyword and

// then to String() method

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

System.out.println(xyz);

}

}

=========

**How to perform scroll up to element by using JavascriptExecutor**  **VVVVVIMMMPPPPPPPPPPPP**

- first find the element in web page using findElement() method and given locator

WebElement wb = driver.findElement(By.xpath("xpath value"));

- convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing arguments[0].scrollIntoView(true) method and WebElement instance.

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

or

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

Scenario:

Open a amazon.in url

scroll up to Shop activity trackers and smartwatches text

step 1: connect to actual browser

step 2: up casting

WebDriver driver = new ChromeDriver();

step 3: open a url

driver.get("https://www.amazon.in");

step 4: find the element

WebElement wb = driver.findElement(By.xpath("//h2[text()='Shop activity trackers and smartwatches']"));

step 5: convert the WebDriver Object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 6; use executeScript() method by passing arguments[0].scrollIntoView() method and WebElement instance

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

package JavascriptExecutorTutorial2;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

public static void main(String[] args) throws InterruptedException {

// step 1: connect browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

// step 3: open a url

driver.get("https://www.redbus.in/");

Thread.sleep(5000);

// step 4: find the element in web page

WebElement wb = driver.findElement(By.xpath("//h2[text()='Global Presence']"));

// step 5: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 6: use executeScript() by passing arguments[0].scrollIntoView() method

// and WebElement instance

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

}

}

**how to create border for element using JavascriptExecutor VVVVVIMMMPPPPPPPPPP**

- find the element in web page

WebElement wb = driver.findElement(By.id("id value"));

- convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing arguments[0].style.border='3px solid red' and WebElement instance

js.executeScript("arguments[0].style.border='3px solid red';", wb);

Scenario:

open a facebook register page https://www.facebook.com/reg

find the first name text box

create border for first name text box

take the screenshot

Step 1: connect browser

step 2: up casting

WebDriver driver =new ChromeDriver();

step 3: open a url

driver.get("https://www.facebook.com/reg");

step 4: find the first name text box

WebElement wb = driver.findElement(By.name("firstname"));

step 5: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 6: use executeScript() method by passing arguments[0].style.border='3px solid red' and WebElement instance

js.executeScript("arguments[0].style.border='3px solid red';", wb);

step 7: convert WebDriver object into the TakesScreenshot

TakesScreenshot ts = (TakesScreenshot)driver;

step 8: use getScreenshotAs() by passing OutputType.FILE keyword

File src = ts.getScreenshotAs(OutputType.FILE);

step 9: store the screenshot in destination location

File dest = new File(System.getProperty("user.dir")+"\\ScreenshotFolder\\abc.png");

step 10: copy the screenshot from source to destination location using FileUtils class dot copyFile() method and by passing source and destination location

FileUtils.copyFile(src, dest);

package JavascriptExecutorTutorial2;

import java.io.File;

import java.io.IOException;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo4 {

public static void main(String[] args) throws InterruptedException, IOException {

// step 1: connect browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

// step 3: open a url

driver.get("https://www.facebook.com/reg");

Thread.sleep(5000);

// step 4: find the first name text box

WebElement wb = driver.findElement(By.name("firstname"));

// step 5: convert the WebDriver Object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 6: use executeScript() method by passing arguments[0].style.border='3px

// solid red' and WebElement instance

js.executeScript("arguments[0].style.border='3px solid red';", wb);

// step 7: convert the WebDriver Object into the TakesScreenshot

TakesScreenshot ts = (TakesScreenshot) driver;

// step 8: use getScreenshotAs() method and by passing OutputType.FILE

File src = ts.getScreenshotAs(OutputType.FILE);

// step 9: store screenshot in destination location

File dest = new File(System.getProperty("user.dir") + "\\PassScreenshotFolder\\abc.png");

// step 10: copy the file from source location to destination location

FileUtils.copyFile(src, dest);

}

}

1) Scroll up and scroll down up to element VVVVIMMPPPPPPP

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

2) border for element VVVVIMMPPPPPPP

js.executeScript("arguments[0].style.border='3px solid red';", wb);

3) navigate back in browser history

js.executeScript("history.go(-1);");

4) navigate forward in browser history

js.executeScript("history.go(+1);");

5) refresh the browser

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

6) capture title

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

7) capture url

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

package JavascriptExecutorTutorial2;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

public static void main(String[] args) throws InterruptedException {

// step 1: connect browser

// step 2: up casting

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

// step 3: open a url

driver.get("https://www.redbus.in/");

Thread.sleep(5000);

// step 4: find the element in web page

WebElement wb = driver.findElement(By.xpath("//h2[text()='Global Presence']"));

// step 5: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// step 6: use executeScript() by passing arguments[0].scrollIntoView() method

// and WebElement instance

// scroll down

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

Thread.sleep(7000);

// step 4: find the element in web page and scroll up

WebElement wb1 = driver.findElement(By.xpath("//div[text()='TRENDING OFFERS']"));

js.executeScript("arguments[0].scrollIntoView();", wb1);

}

}

**How to generate Alert Pop or Confirm Pop up or prompt Pop up in selenium WebDriver using JavascriptExecutor**

if we generate Alert Pop then it provide the physical text in alert pop up as well as ok button

if we generate Confirm Pop up then it provide the physical text in alert pop up, ok button and cancel button

if we generate the prompt Pop up then provide the physical text, text box, ok button and cancel button

we have convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

then we use executeScript() method by passing alert('message');

js.executeScript("alert('User is on Register Page');");

------------------------------------------------------------------------

Scenario:

open a fb application

generate alert pop up

handle alert pop up

Step 1:: connect browser

step 2: up casting

WebDriver driver =new ChromeDriver();

step 3: open a url

driver.get("https://www.facebook.com/reg");

step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 5: use executeScript() method by passing alert('custom message')

js.executeScript("alert('User is on Facebook Register page');");

step 6: switch focus to alert pop by using switchTo().alert() method

Alert alt = driver.switchTo().alert();

step 7: capture the alert pop physical text

String a = alt.getText();

System.out.println(a);

step 8: click on ok button

alt.accept();

package JavascriptExecutorTutorial3;

import org.openqa.selenium.Alert;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

public static void main(String[] args) throws InterruptedException {

// Connect browser

// up Casting

WebDriver driver = new ChromeDriver();

// open a url

driver.get("https://www.facebook.com/reg");

Thread.sleep(5000);

// convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// use executeScript() method by passing alert('custom message');

js.executeScript("alert('User is on facebook Register page');");

// switch focus to alert pop up by using switchTo().alert() method

Alert alt = driver.switchTo().alert();

// capture the physical text in alert pop uo

String a = alt.getText();

System.out.println(a);

Thread.sleep(5000);

// click on ok button

alt.accept();

Thread.sleep(5000);

driver.quit();

}

}

----------------------------------------------------------------------------------------------------------

**if we generate Confirm Pop up then it provide the physical text in alert pop up, ok button and cancel button**

- convert the WebDriver object into JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing confirm('custom message');

js.executeScript("confirm('custom message');");

Scenario:

open a google application

generate confirm pop up

handle confirm pop up

Step 1: connect browser

step 2: up casting

step 3: open a url

step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 5: use executeScript() method by passing confirm('custom message')

js.executeScript("confirm('user is on Google Home Page');");

step 6: switch focus to confirm up pop up/alert pop

Alert alt = driver.switchTo().alert();

step 7: capture the alert pop up physical text using getText() method

String a = alt.getText();

System.out.println(a);

step 8: click on cancel button using dismiss()method

alt.dismiss();

package JavascriptExecutorTutorial3;

import org.openqa.selenium.Alert;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

public static void main(String[] args) throws InterruptedException {

// Connect browser

// up Casting

WebDriver driver = new ChromeDriver();

// open a url

driver.get("https://www.google.com");

Thread.sleep(5000);

// convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// use executeScript() method by passing confirm('custom message');

js.executeScript("confirm('User is on Google home Page');");

// switch focus to alert pop or confirm pop up using driver.switchTo().alert()

// method

Alert alt = driver.switchTo().alert();

// capture the alert/confirm pop physical text

String a = alt.getText();

System.out.println(a);

Thread.sleep(5000);

// click on cancel button

alt.dismiss();

Thread.sleep(5000);

driver.quit();

}

}

--------------------------------------------------------------------------------------------------------------------

**if we generate the prompt Pop up then provide the physical text, text box, ok button and cancel button**

- - convert the WebDriver object into JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing confirm('custom message');

js.executeScript("prompt('custom message');");

Scenario:

open a google app

generate the prompt pop up

switch focus to prompt pop up or alert pop up

click on ok button

step 1: connect browser

step 2: up casting

WebDriver driver =new ChromeDriver();

step 3: open a url

driver.get("https://www.google.com");

step 4: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 5: use executeScript() by passing prompt('custom message');

js.executeScript("prompt('user is on google Home page');");

step 6: switch focus to prompt pop up using driver.switchTo().alert()

Alert alt = driver.switchTo().alert();

step 7: capture the alert pop up physical text

String a = alt.getText();

step 8: enter text in alert pop up text box

alt.sendKeys("selenium testing");

step 9: click on ok button

alt.accept();

package JavascriptExecutorTutorial3;

import org.openqa.selenium.Alert;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

public static void main(String[] args) throws InterruptedException {

// Connect browser

// up Casting

WebDriver driver = new ChromeDriver();

// open a url

driver.get("https://www.google.com");

Thread.sleep(5000);

// convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// use executeScript() method by passing prompt('custom message');

js.executeScript("prompt('User is on Google home Page');");

// switch focus to prompt pop up

Alert alt = driver.switchTo().alert();

// capture pop up message

String a = alt.getText();

System.out.println(a);

// enter text in pop up

alt.sendKeys("selenium testing");

Thread.sleep(5000);

// click on ok button

alt.accept();

Thread.sleep(5000);

driver.quit();

}

}

===================================================================================

**Handle drop down using JavascriptExecutor**

- convert the WebDriver Object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

- use executeScript() method by passing document.getElementsByName() method and inside the method pass the HTML DOM value attribute values name.

js.executeScript("document.getElementsByName('html drop down values');");

Scenario:

open a facebook application

select the Nov month values

step1: connect to actual browser

step 2: up casting

step 3: open a url

driver.get("https://www.facebook.com/reg");

step 4: convert the WebDriver object into JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 5: use executeScript() method by passing document.getElementsByName() method and name locator value and value attribute value

js.executeScript("document.getElementsByName('birthday\_month'[0].value='11';)';");

package JavascriptExecutorTutorial3;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo4 {

public static void main(String[] args) throws InterruptedException {

// Connect browser

// up Casting

WebDriver driver = new ChromeDriver();

// open a url

driver.get("https://www.facebook.com/reg");

Thread.sleep(5000);

// convert WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

// use executeScript() method by passing getElementsByName() and inside the

// method pass name locator value and value attribute value

js.executeScript("document.getElementsByName('birthday\_month')[0].value='2';");

}

}

package JavascriptExecutorTutorial3;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo5 {

public static void main(String[] args) throws InterruptedException {

// Connect browser

// up Casting

WebDriver driver = new ChromeDriver();

// open a url

driver.get("https://demo.guru99.com/test/newtours/register.php");

Thread.sleep(5000);

// convert WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("document.getElementsByName('country')[0].value='ISRAEL';");

}

}

---------------------------------------------------------------------------------------

package JavascriptExecutorTutorial3;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo6 {

public static void main(String[] args) throws InterruptedException {

// Connect browser

// up Casting

WebDriver driver = new ChromeDriver();

// open a url

driver.get(

"https://portal2.passportindia.gov.in/AppOnlineProject/user/RegistrationBaseAction?request\_locale=en");

Thread.sleep(5000);

// convert WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("document.getElementsByClassName('dropdown-box')[0].value='31';");

}

}

===================================================================================

Scenario:

open a https://demoqa.com/automation-practice-form

select state Haryana

Step 1: connect browser

step 2: up casting

step 3: open a url

driver.get("https://demoqa.com/automation-practice-form");

step 4:click on state drop down

driver.findElement(By.id("state")).click();

step 5: convert the WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor)driver;

step 6: use executeScript() method by passing drop down value location and then use click() method

js.executeScript("document.getElementById('react-select-3-option-2').click();");

package JavascriptExecutorTutorial3;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo7 {

public static void main(String[] args) throws InterruptedException {

// Connect browser

// up Casting

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

// open a url

driver.get("https://demoqa.com/automation-practice-form");

Thread.sleep(5000);

driver.findElement(By.xpath("//div[text()='Select State']")).click();

// convert WebDriver object into the JavascriptExecutor

JavascriptExecutor js = (JavascriptExecutor) driver;

Thread.sleep(5000);

js.executeScript("document.getElementById('react-select-3-option-2').click();");

}

}

============================================================================

1) click on button using JavascriptExecutor VVVVIMMMPPPPPPP

2) enter text in text box using JavascriptExecutor VVVVIMMMPPPPPPP

3) scroll up and scroll down using JavascriptExecutor VVVVIMMMPPPPPPP

4) border for element using JavascriptExecutor VVVVIMMMPPPPPPP

5) handle drop down using JavascriptExecutor VVVVIMMMPPPPPPP

============================================================================