Javascriptexecutor:

1.can be used where we have a complex HTML code and the id is very dynamic.

2.can be used to take screenshot with pointers or borders.

3.used to boost up/enhance the selenium performance

It is an interface used to execute javascript code thru selenium webdriver

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 any desired operation on a web element.

Some of the scenarios where javascript executor is used:

### 1.*Changing style attribute of the element*s

### 2. Finding Element

### 3.*****Getting Element Attributes*****

### 4.*****Total Frames in Browser*****

### *****5.Size of Window*****

### *****6.Navigating to different Page*****

### *****7.Generate Alert Pop Window*****

### *****8.Click Action*****

### *****9.Refresh Browser*****

### *****10.Get InnerText of a Webpage*****

### *****11.Get Title of a WebPage*****

### *****12.Scroll Page*****

It has two methods to execute any script

1.executescript()

2.executeAsynscript()

One such example is highlighting webelement using js executor:

Highlighting webelement using js executor:

**public** **static** **void** flash(WebElement element,WebDriver driver )

{

String bgcolor=element.getCssValue("backgroundColor");//gets the bgcolor of webelement

**for**(**int** i=0;i<100;i++)//flashes/highlights

{

*change*(driver,"rgb(255,255,0)",element);//changes to any color specified

*change*(driver,bgcolor,element);//restores original color

}

}

**public** **static** **void** change(WebDriver driver,String color,WebElement element)

{

JavascriptExecutor js=(JavascriptExecutor) driver;

js.executeScript("arguments[0].style.backgroundColor='"+ color+"'", element);

}

Javascript executor with selenium webdriver:

Note:

**public** **static** **void** change(WebDriver driver,String color,WebElement element)

{

JavascriptExecutor js=(JavascriptExecutor) driver;

js.executeScript("arguments[0].style.backgroundColor='"+ color+"'", element);

}

arguments[0]🡪we are passing arguments[0] since we are passing only one webelement

**public** **static** String gettitlebyjs(WebDriver driver)

{

JavascriptExecutor js=(JavascriptExecutor) driver;

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

**return** title;

}

return document.title🡪document is the object of document object

If you want to access any element in an HTML page, you always start with accessing the document object

Difference between isdisplayed(),isenabled(),and isselected() methods?

*driver*.get("https://ui.cogmento.com/");

//isdisplayed() method applicable for all webelements

**boolean** result=*driver*.findElement(By.*xpath*("//\*[@class='ui fluid large blue submit button']")).isDisplayed();

System.***out***.println(result);

//isenabled() -->to check whether an element is enabled or disabled

**boolean** result1=*driver*.findElement(By.*xpath*("//\*[@class='ui fluid large blue submit button']")).isEnabled();

System.***out***.println(result1);

Scenario for isenabled():

In some webpage,only if we check



I agree to terms and conditions

Submit button below will be enabled,otherwise it will be disabled

Primarily used with buttons

Isselected()

Used to chk whether web element is selected or not.

Predominantly used with radio buttons,checkboxes and dropdown