Automation Testing : Selenium

:- Selenium IDE

:- WebDriver

:- Grid

Webdriver :

:- web driver is java interface

:- **Selenium Webdriver** is an open-source collection of APIs which is used for testing web applications.

:- not developed by any company , different developer is comes together and develop it

Interface =

Know what to do √ hence

But don't know how to do = hence empty method later implemented

```
Interface  webdriver
 {
 Empty method;
 }
      Class FirefoxDriver  implements web driver
      {      …define method ;  }
```

Now we use this classes eg. FirefoxDriver , ChromeDriver etc

FirefoxDriver :

System.setProperty("webdriver.gecko.driver", "D:\\geckodriver.exe");

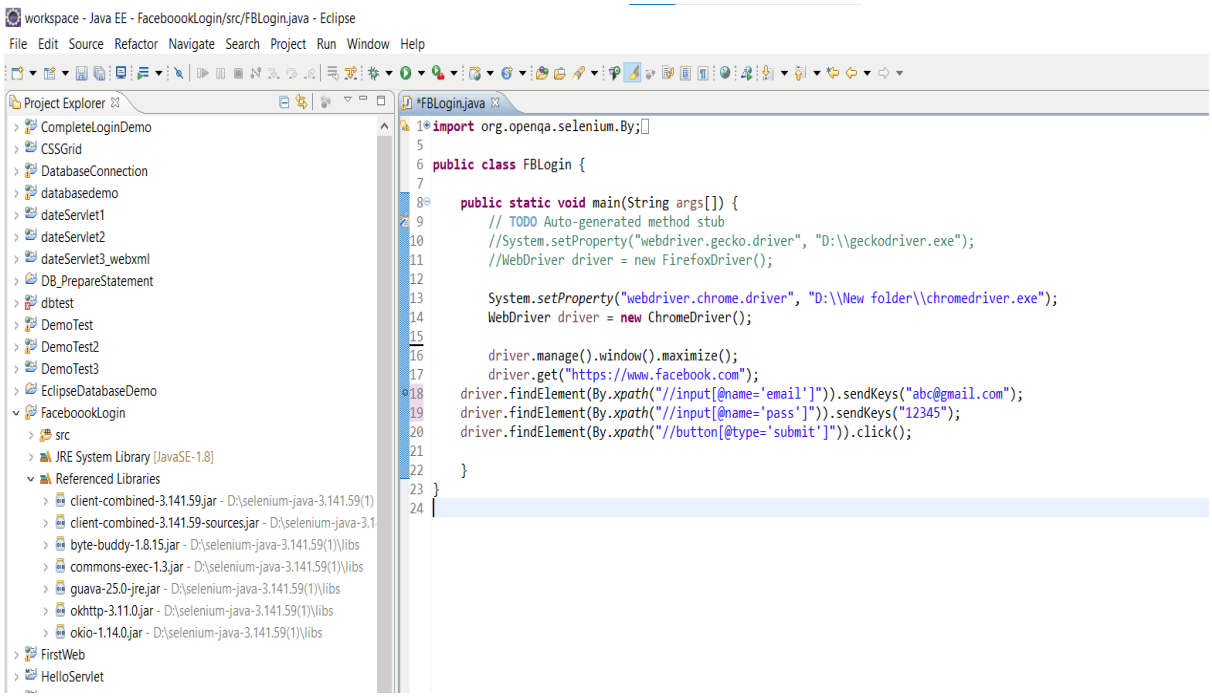
ChromeDriver

System.setProperty("webdriver.chrome.driver", "D:\\ChromeDriver\\chromedriver.exe");

IEDriver

System.setProperty("webdriver.ie.driver", "D:\\IE Driver Server\\IEDriverServer.exe");

Open eclipse >> add new java project >> add external Library



import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

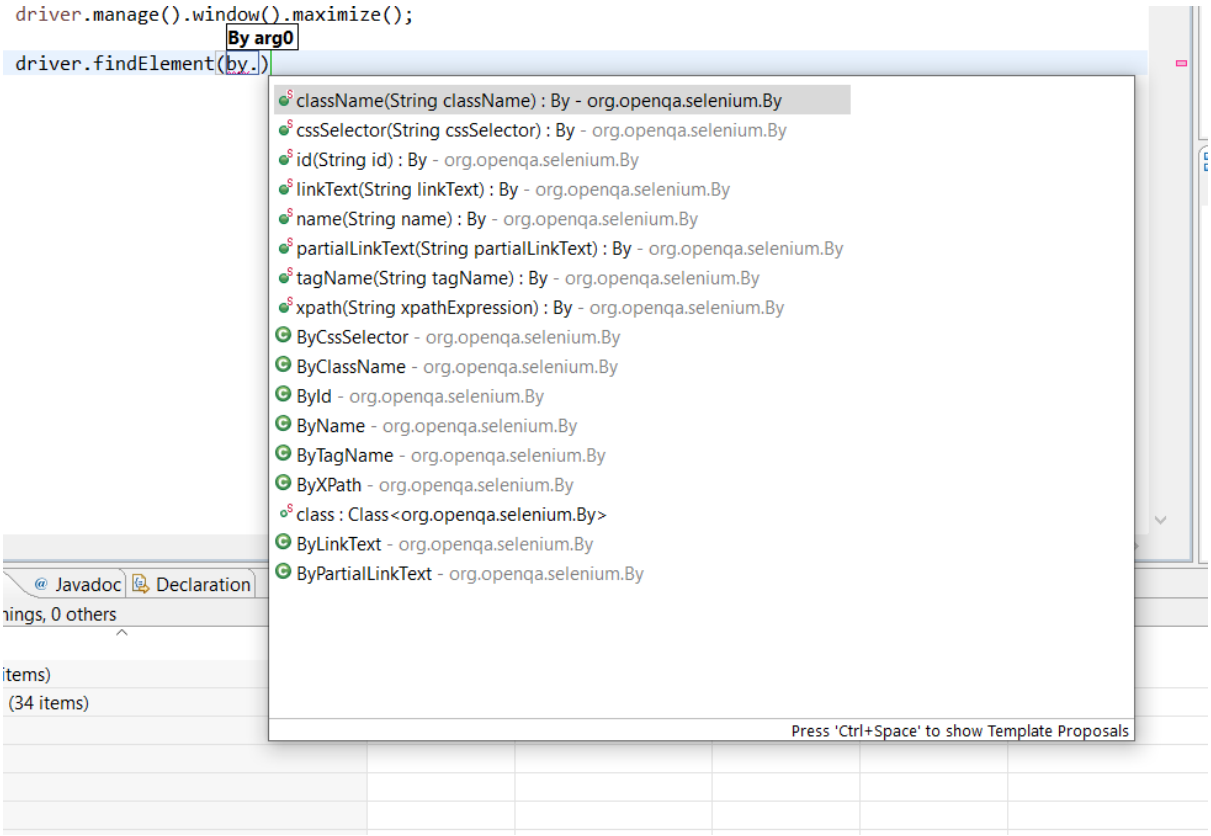
```
public class FBLogin {

	public static void main(String args[]) {

		//System.setProperty("webdriver.gecko.driver", "D:\\geckodriver.exe");
		//WebDriver driver = new FirefoxDriver();

		System.setProperty("webdriver.chrome.driver", "D:\\New folder\\chromedriver.exe");
		WebDriver driver = new ChromeDriver();
		driver.manage().window().maximize();
		driver.get("https://www.facebook.com");

	driver.findElement(By.xpath("//input[@name='email']")).sendKeys("abc@gmail.com");
	driver.findElement(By.xpath("//input[@name='pass']")).sendKeys("12345");
	driver.findElement(By.xpath("//button[@type='submit']")).click();

	}
}
```

Selector :

Locator is a command that tells Selenium IDE which GUI elements (say Text Box, Buttons, Check Boxes etc) its needs to operate on.

If name attribute same for multiple element then use id attribute or combination of Tag and name , tag and class etc **simply xpath is good**

Tag | Attribute | value



ID :

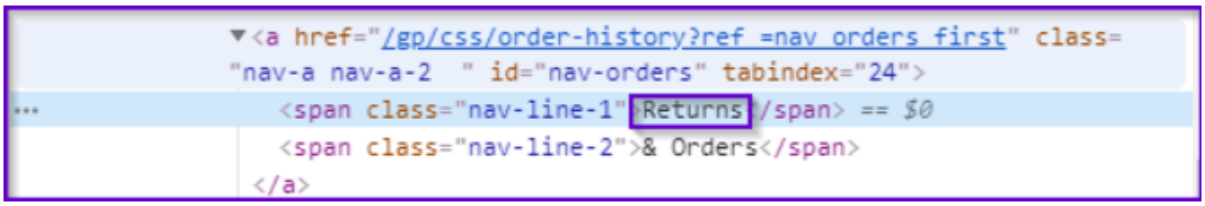
WebElement = driver.findElement(By.id("twotabsearchtextbox"))

will raise NoSuchElementException if not found

Name

elementName = driver.findElement(By.name("field-keywords"))

LinkText

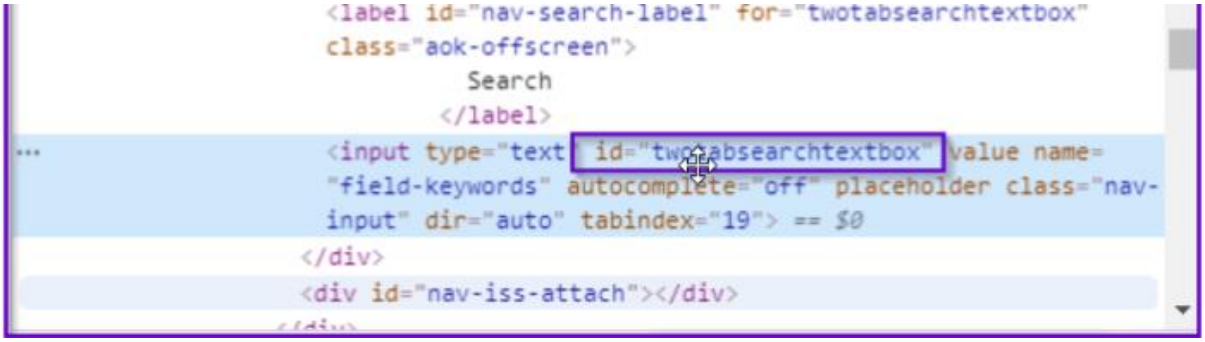


elementLinktext = driver.findElement(By.linkText("Returns"))

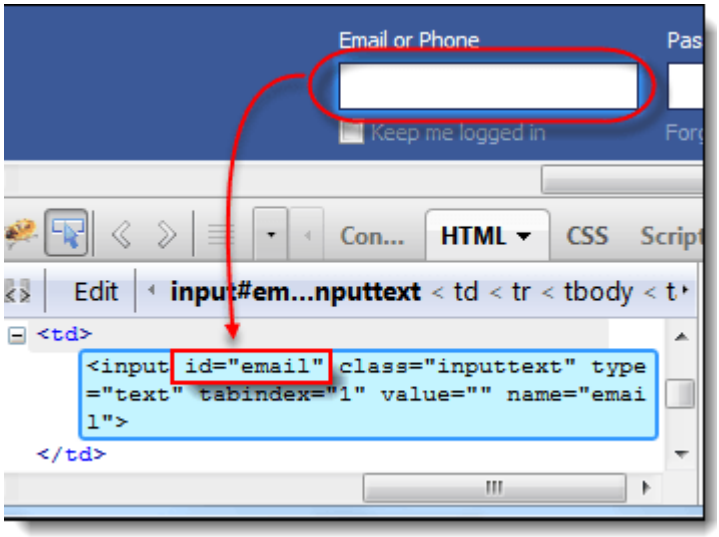
Css Selecter

elementcss= driver.findElement(By.cssSelector('div.nav-search-input'))

Xpath	//input[@id='twotabsearchtextbox']

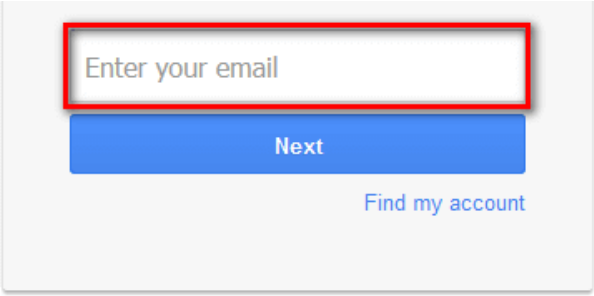


elementxpath = driver.findElement(By.xpath("//input[@id='twotabsearchtextbox']"))

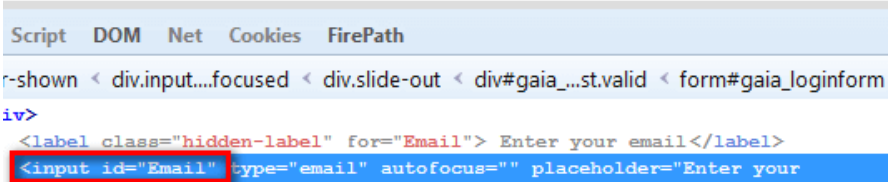


CSS selecter

Tag and Id



Create account

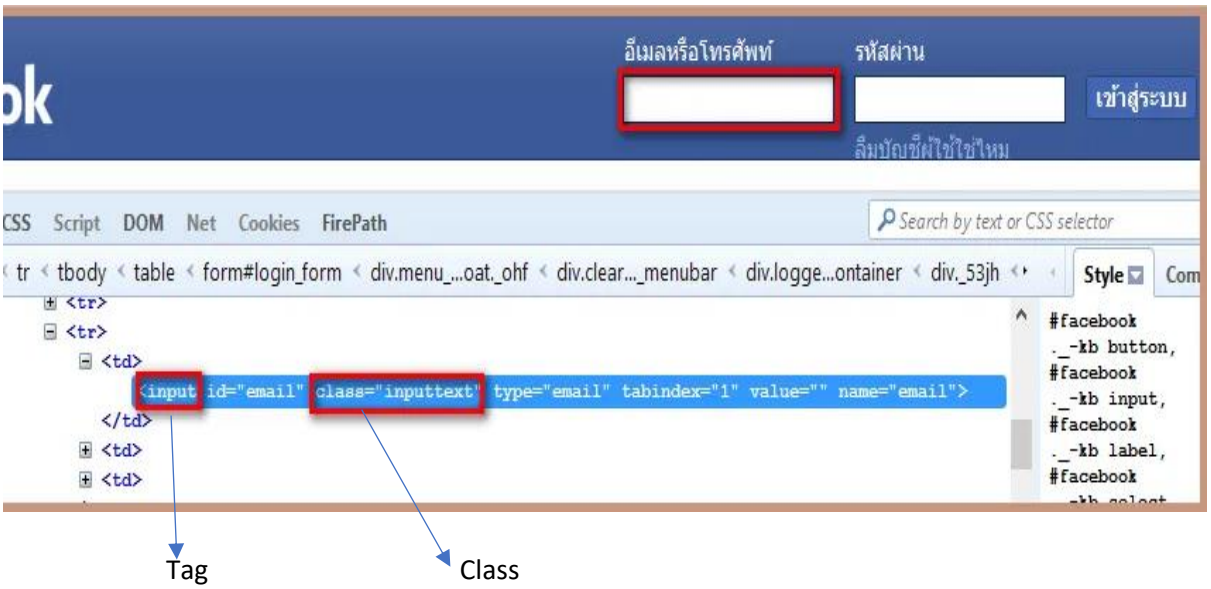


```
package seleniumTutorial;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Locators {

    public static void main (String [] args){

    WebDriver driver = new FirefoxDriver();

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

    // Here Tag = input and Id = Email

    driver.findElement(By.cssSelector("input#Email")).sendKeys("Software Testing Material");

    }

}
```

Tag

ID

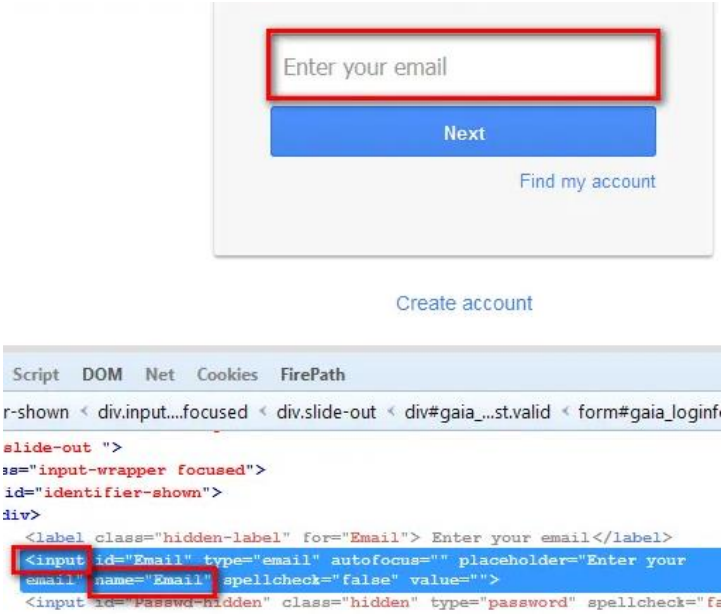
Tag and class



```
public class Locators {
    public static void main (String [] args){
    WebDriver driver = new FirefoxDriver();
    driver.get("https://www.facebook.com/");
    // Here Tag = input and Class = email
    driver.findElement(By.cssSelector("input.inputtext")).sendKeys("Software Testing Material");
    }
}
```

Tag class

Tag and Attribute



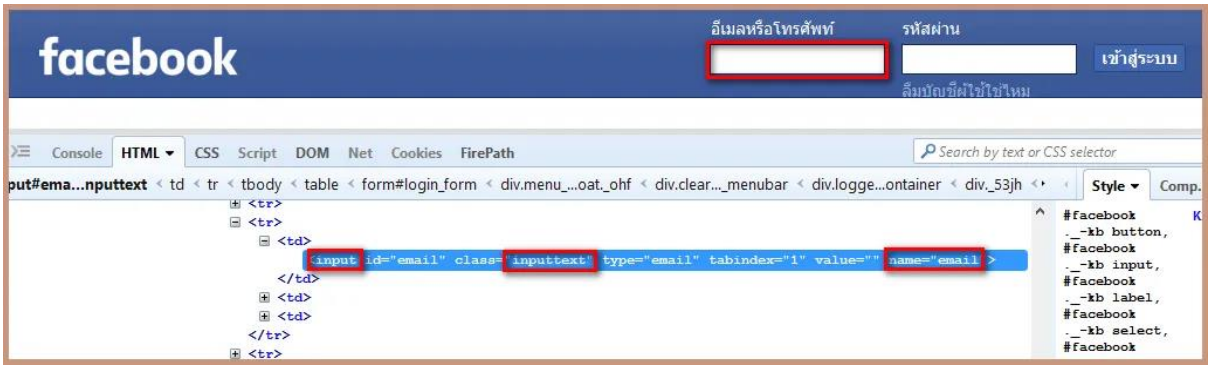

```
seleniumTutorial;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Locators {
    public static void main (String [] args){
```

```
    WebDriver driver = new FirefoxDriver();

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

    // Here Tag = input and Id = Email

    driver.findElement(By.cssSelector("input[name=Email]")).sendKeys("Software Testing
Material");

}
```

tag

attribute=value

Tag, Class And Attribute:



Tag = input

Class = inputtext

ID = name

```
package seleniumTutorial;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Locators {

    public static void main (String [] args){
```

```
    WebDriver driver = new FirefoxDriver();

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

    // Here Tag = input and Class = email

    driver.findElement(By.cssSelector("input.inputtext[name=email]")).sendKeys("Software Testing
Material");

    }

}
```

tag class id

- First try to use **Id**, **name**, **class**, etc.
- Then, try to interrogate by **CSS**
- Then, use **XPath** to find elements.

ID	**#idname**
Class	**.classname**

1) Using Absolute Path CSS Selector

Inspect element

Example

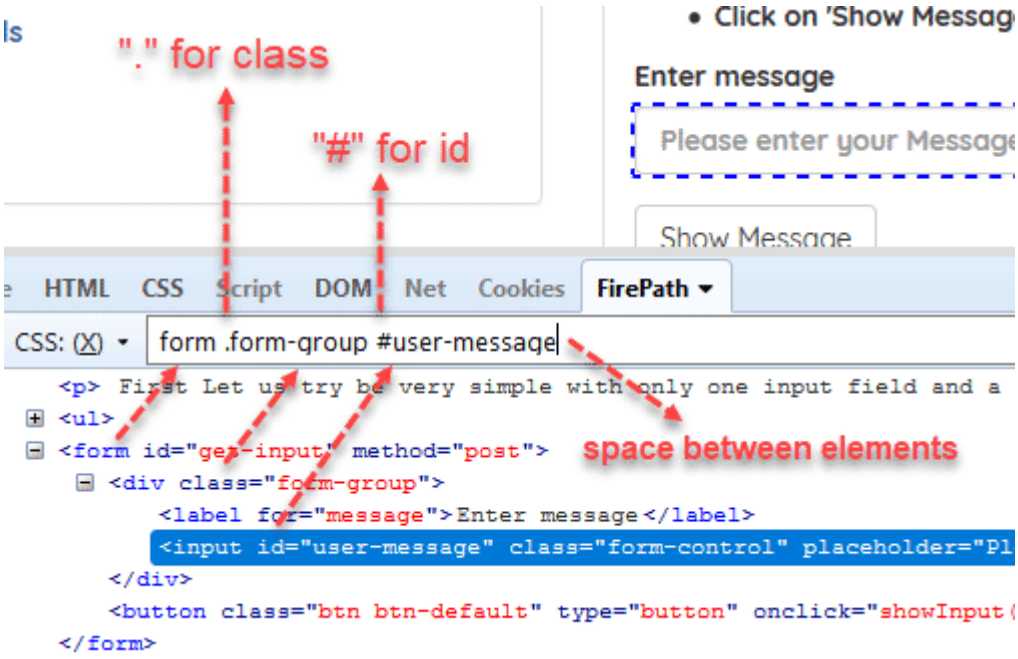
Syntax: **form>div>input**

2) Using Non-Absolute Path CSS Selector

Example

Syntax: **form .form-group #user-message**

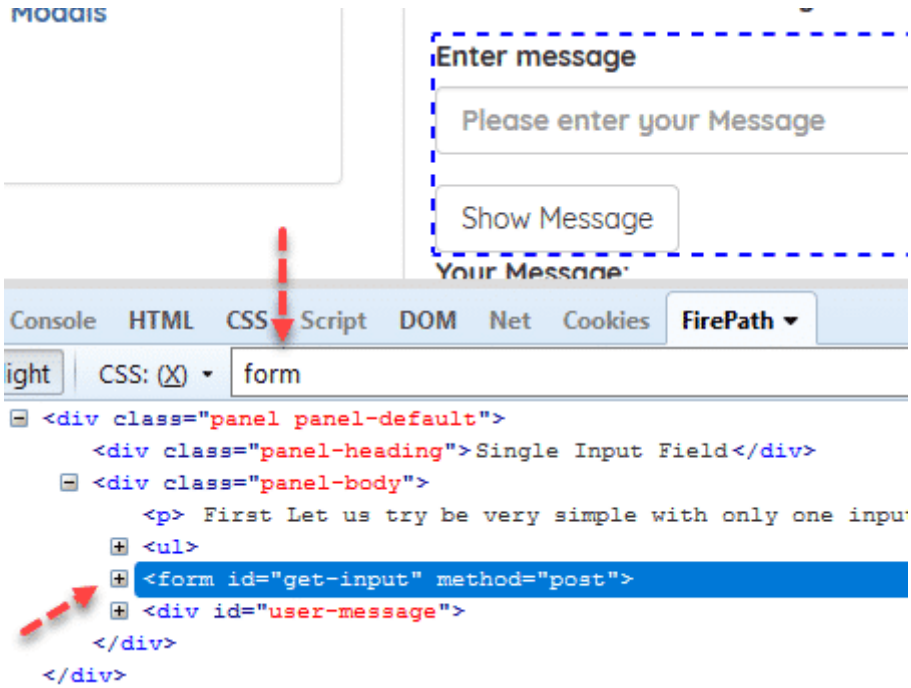
Find the element form and then find class which has name .form-group then find id which has name #user-message



3) Using Tag Name CSS Selector in Selenium

You can **write the tag name directly** like “form”, “div”, “img”,”body”, etc. As below figure, I wrote the “form” tag to locate the form element.
*(Note: For XPath we use **//tagname** for this.)*

Example

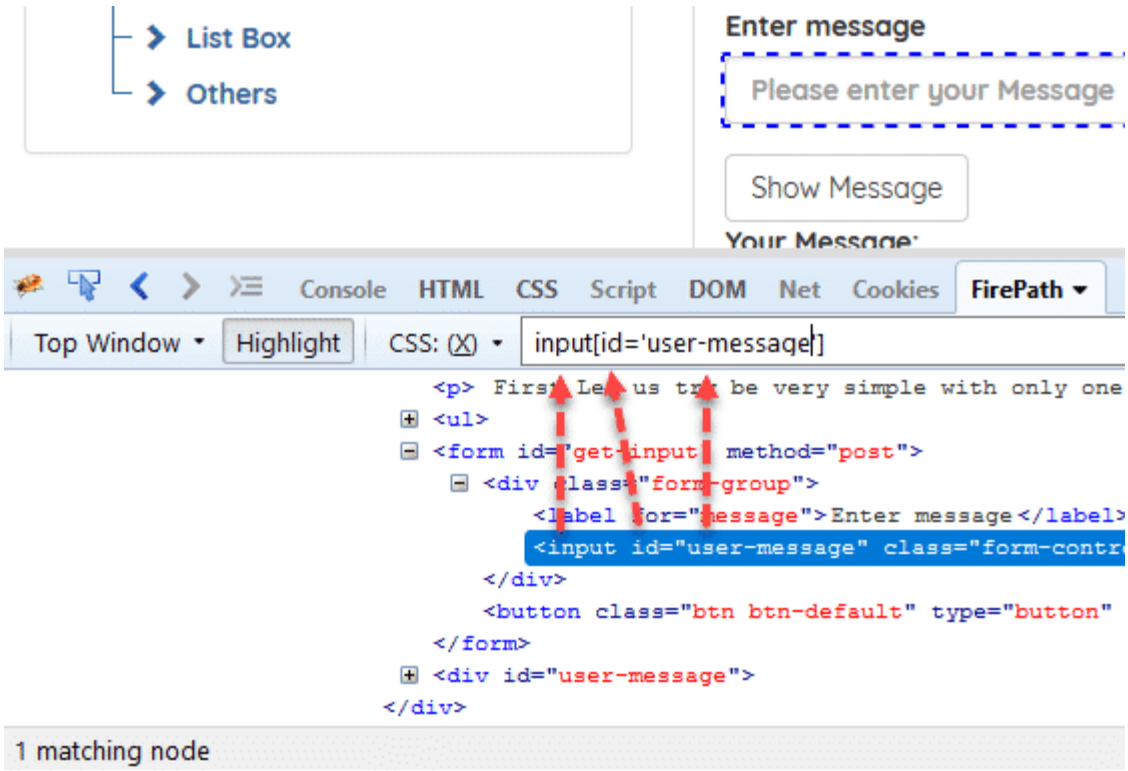


4) Using Tag & Attribute & Value Trio

You can use **tag[attribute=’value’]** syntax.
*(Note: For XPath we use **tag[@attribute=’value’]** for this.)*

Example

Syntax: **input[id=’user-message’]**

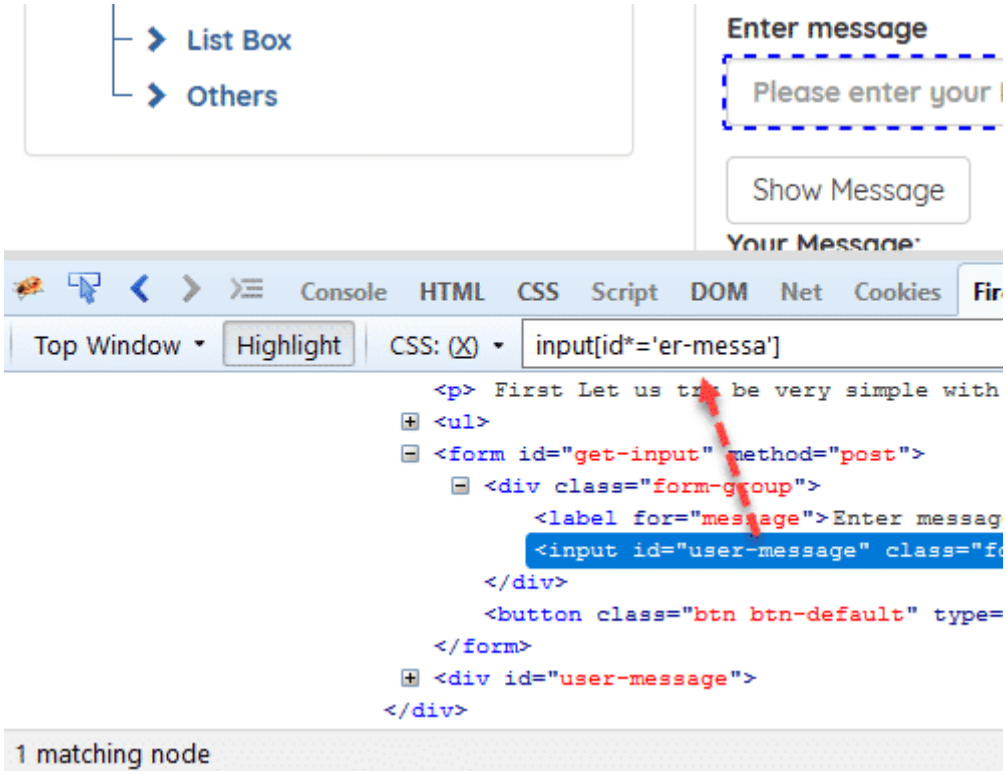


5) Using Containing Text of an Attribute

You can use **tag[attribute*='containing text']** syntax.
*(Note: For XPath we use **tag[contains((@attribute,'containing text')]** for this.)*

Example

Syntax: **input[id*='er-messa']**



6) Using Starting Text of an Attribute

You can use **tag[attribute^='starting text']** syntax.
(Note: For XPath we use ***tag[starts-with(@attribute, 'starting text')]*** *for this.)*

Example

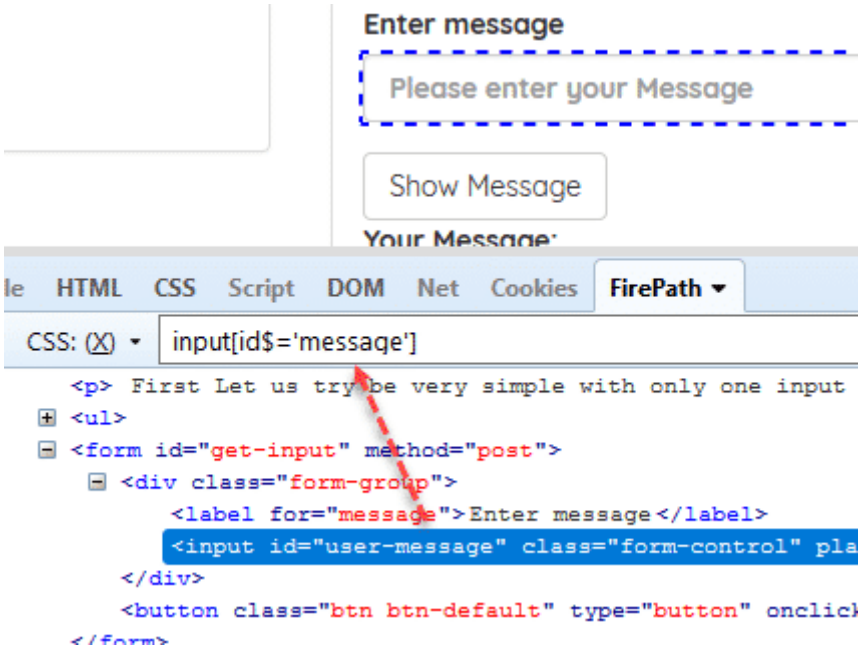
Syntax: **input[id^='user']**

7) Using Ending Text of an Attribute

You can use **tag[attribute$='ending text']** syntax.

Example

Syntax: **input[id$='message']**

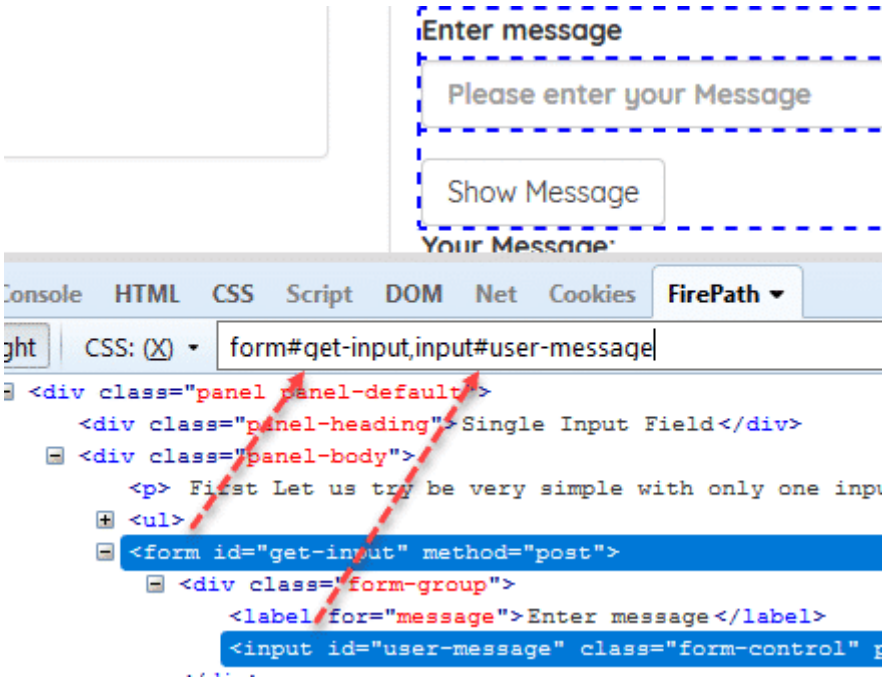


8) Using Comma Operator to Implement OR Operation

You can use **"," operator between two CSS locator statements**.

Example

Syntax: **form#get-input,input#user-message**

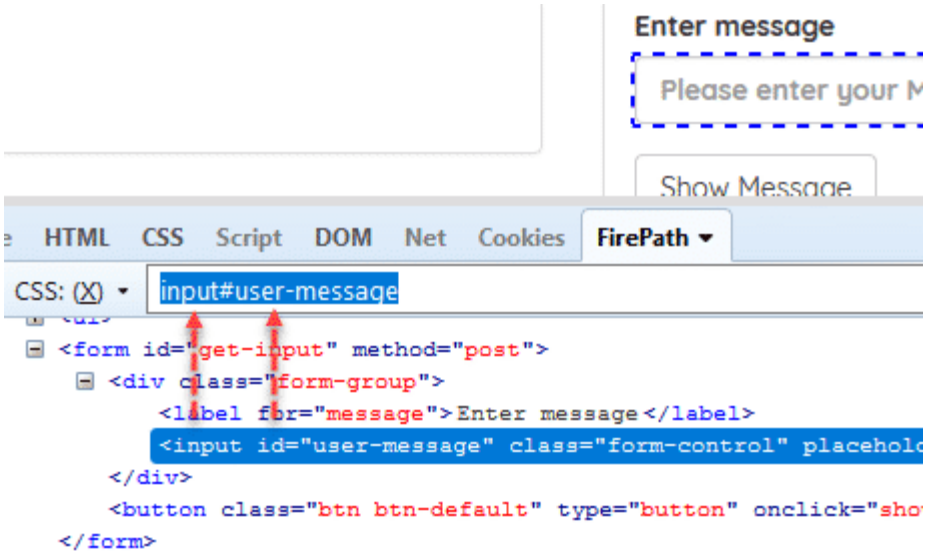


9) Using Tag and ID CSS Selector in Selenium

You can use **"Tag#Id"**

Example

Syntax: **input#user-message**



10) Using Tag and Class CSS Selector in Selenium

You can use **"Tag.Class"**

Example

Syntax: **input.form-control**

11) Using first-of-type CSS Selector in Selenium

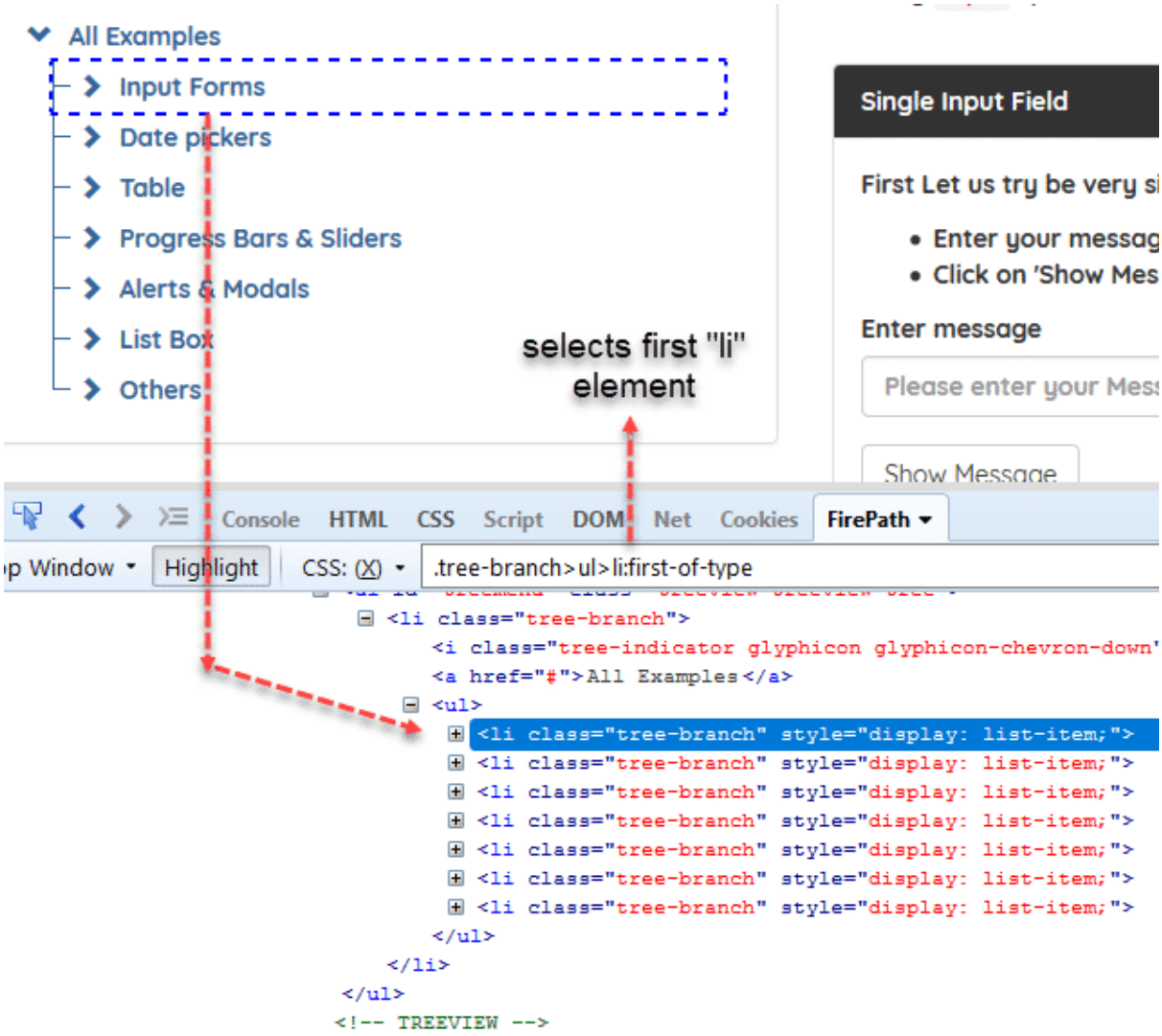
You can use **"Tag:first-of-type"**. It will select the first tag element.

Example

Syntax: **.tree-branch>ul>li:first-of-type**

Find the class = .tree-branch

Find the ul tag >> then find li tag >> but first li tag

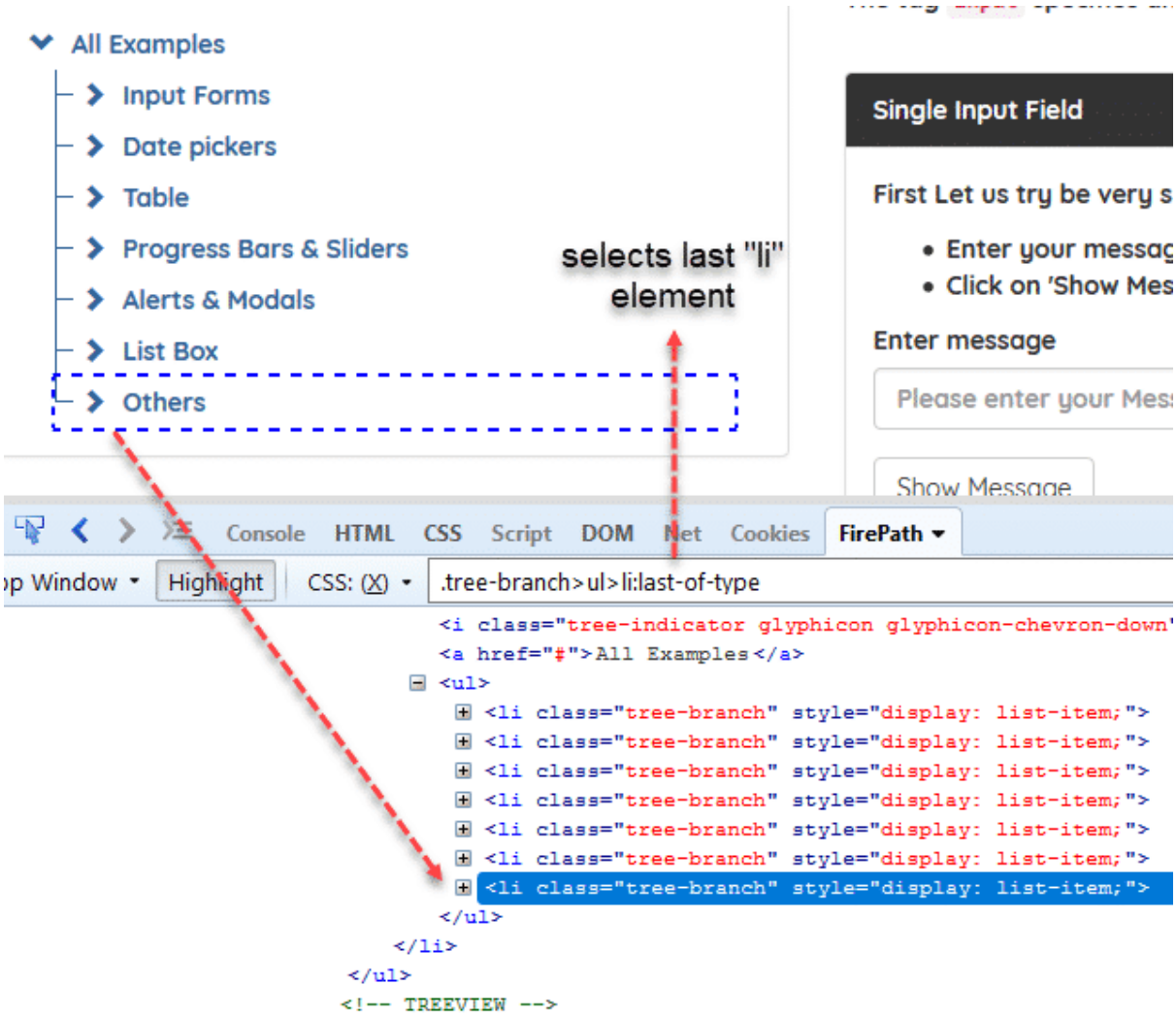


12) Using last-of-type CSS Selector in Selenium

You can use **"Tag:last-of-type"**. It will select the last tag element.

Example

Syntax: **.tree-branch>ul>li:first-of-type**



Note: If you want to find the last element or child you can use the below locators.

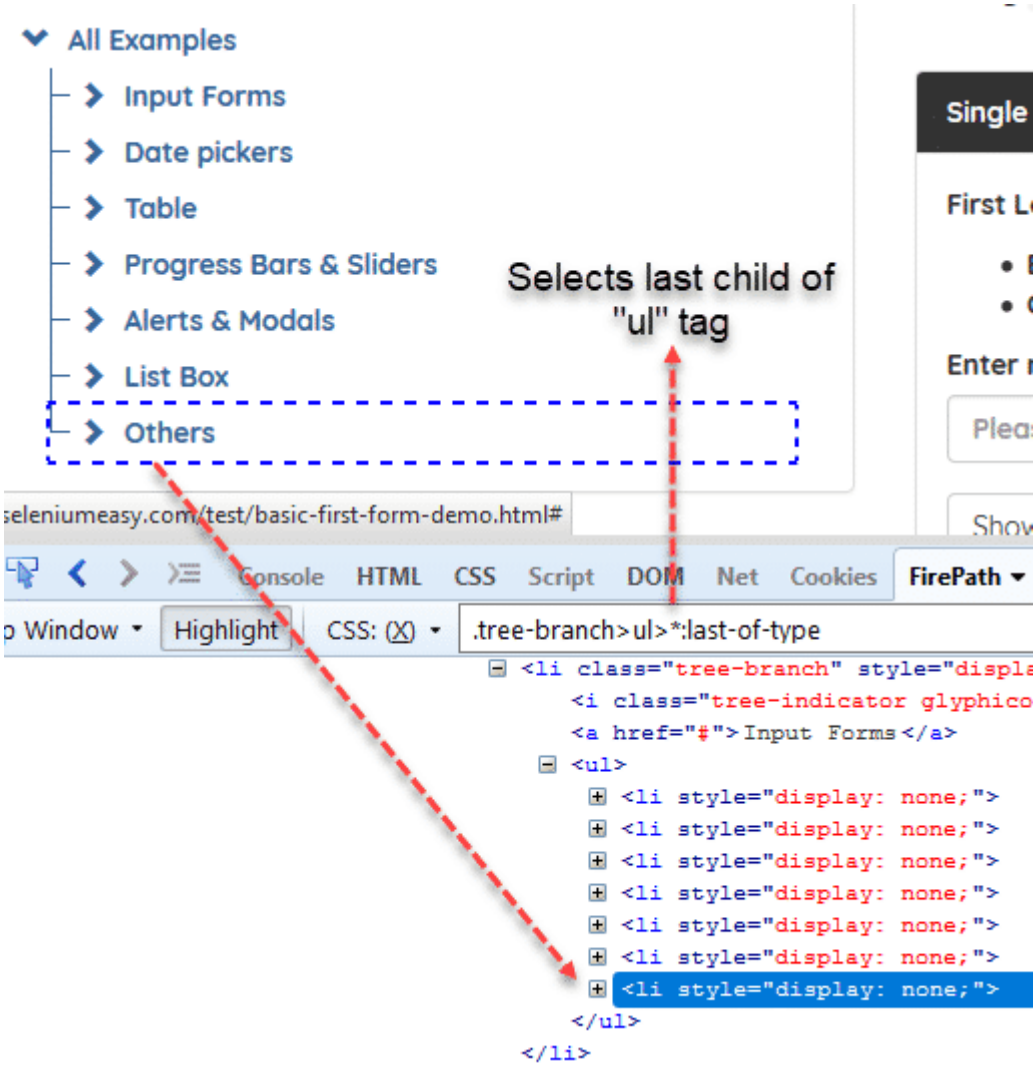
- **Tag:nth-last-of-type(*n*)**
- **Tag:nth-last-child(*n*)**

13) Using *:last-of-type CSS Selector in Selenium

You can use **"*last-of-type"**. It will select the last child of the parent tag.

Example

Syntax: **.tree-branch>ul>*:last-of-type** *(Selects the last child of parent tag "ul".)*

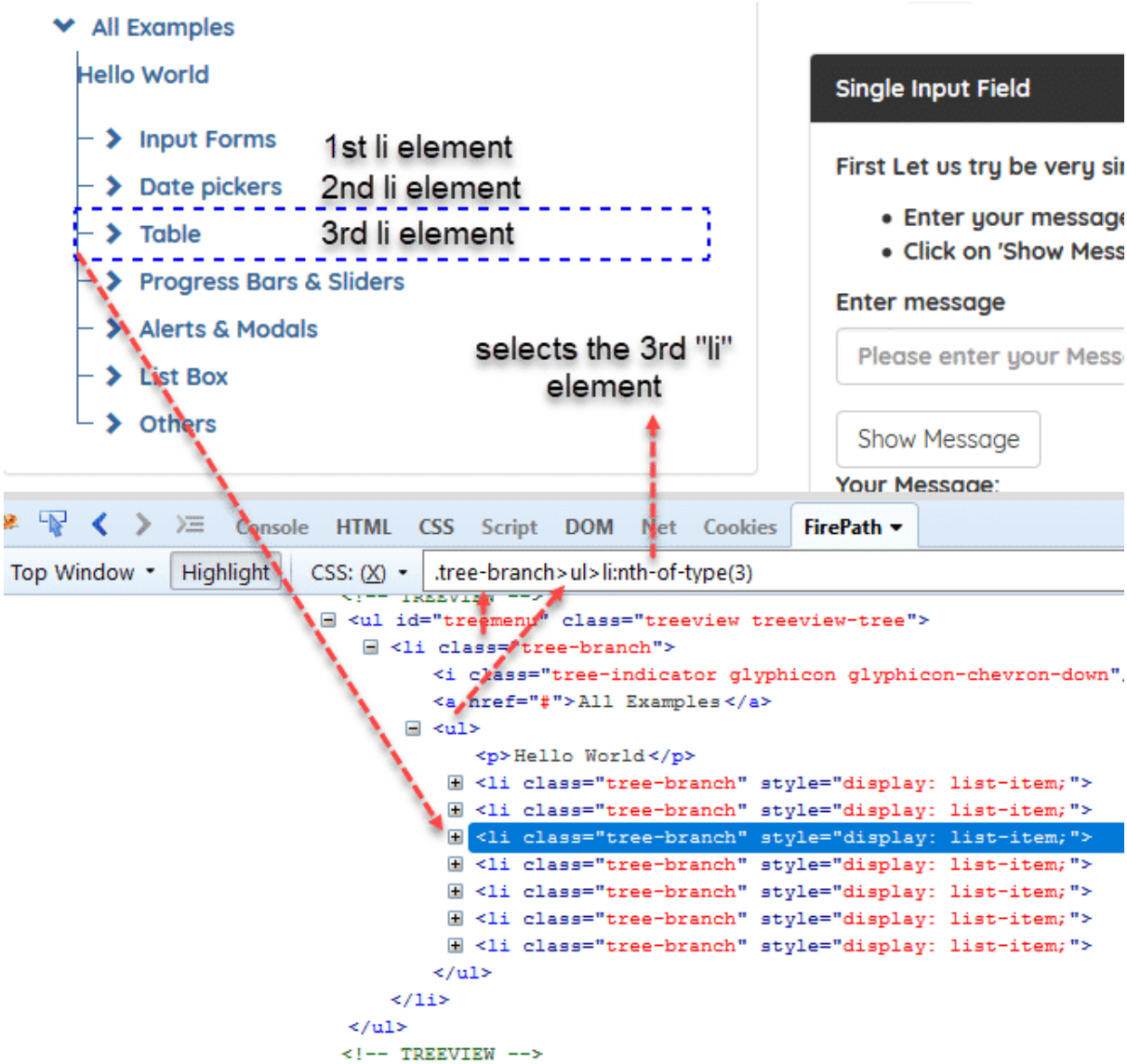


14) Using tag:nth-of-type(*n*) CSS Selector in Selenium

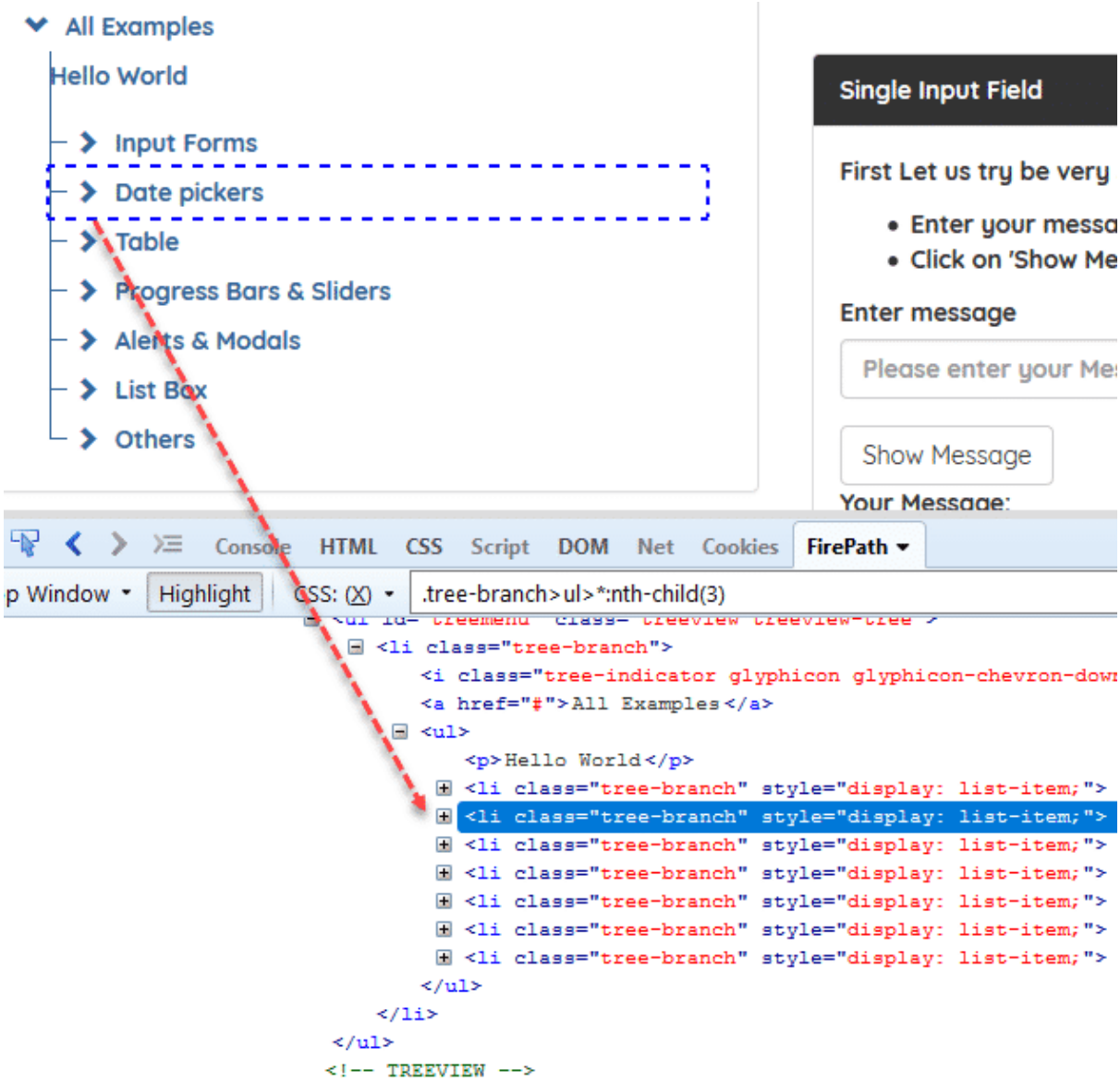
You can use **"tag:nth-of-type(*n*)"**. It will select the nth tag element of the list.

Example

Syntax: **.tree-branch>ul>li:nth-of-type(3)** *(Selects 3rd li element.)*



Note: If you don't specify a tag as ***:nth-of-type(*3*)** it will allow you to **select the third child**.

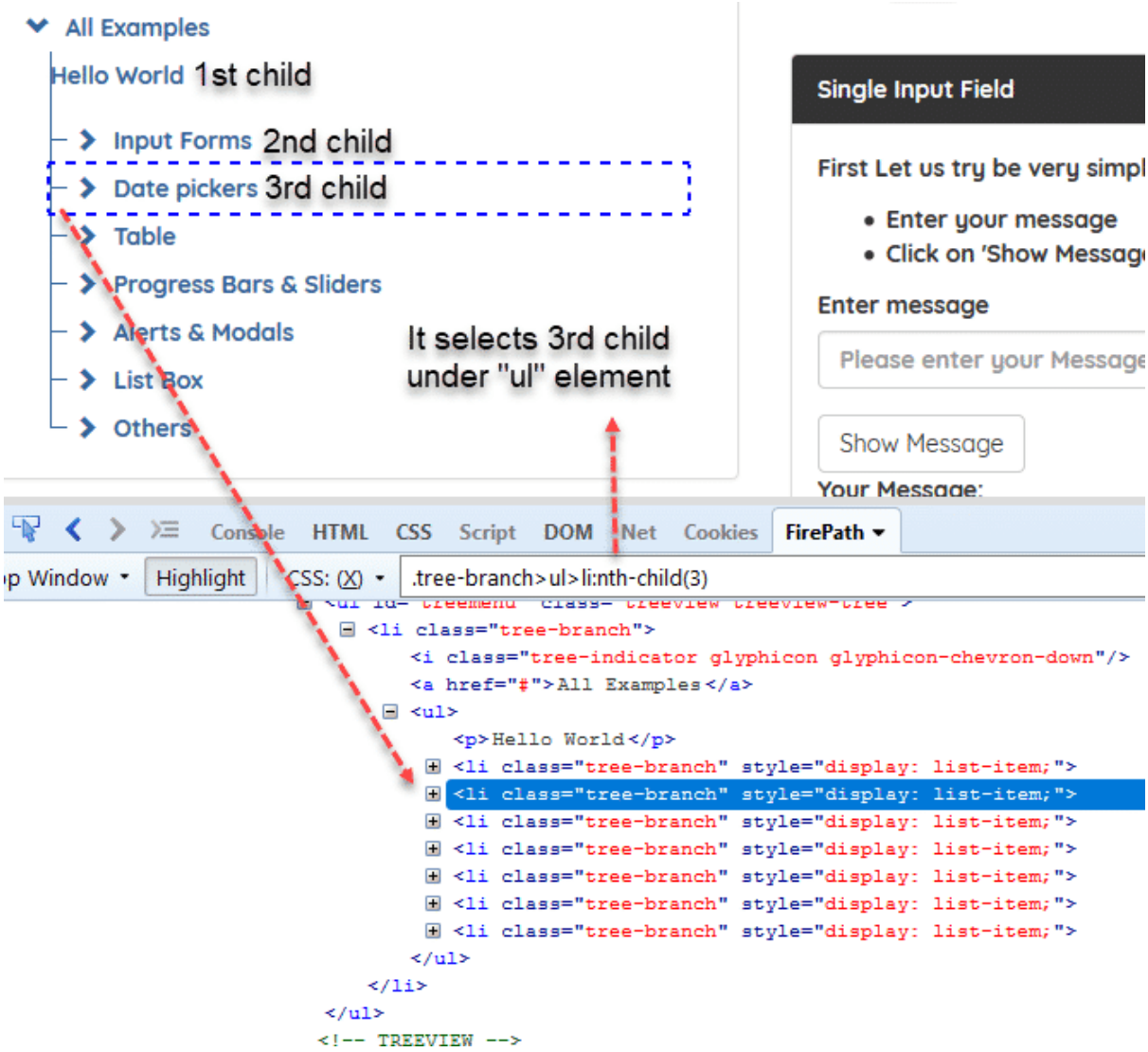


15) Using tag:nth-child(*n*) CSS Selector in Selenium

You can use **"tag:nth-child(*n*)"**. It will select the nth child.

Example

Syntax: **.tree-branch>ul>li:nth-child(3)** (*It will select the nth child.*)



16) Using Sibling "+" Operator CSS Selector in Selenium

You can use **"E1+ E2"**. First, it finds E1 then selects E2.

Sample HTML:

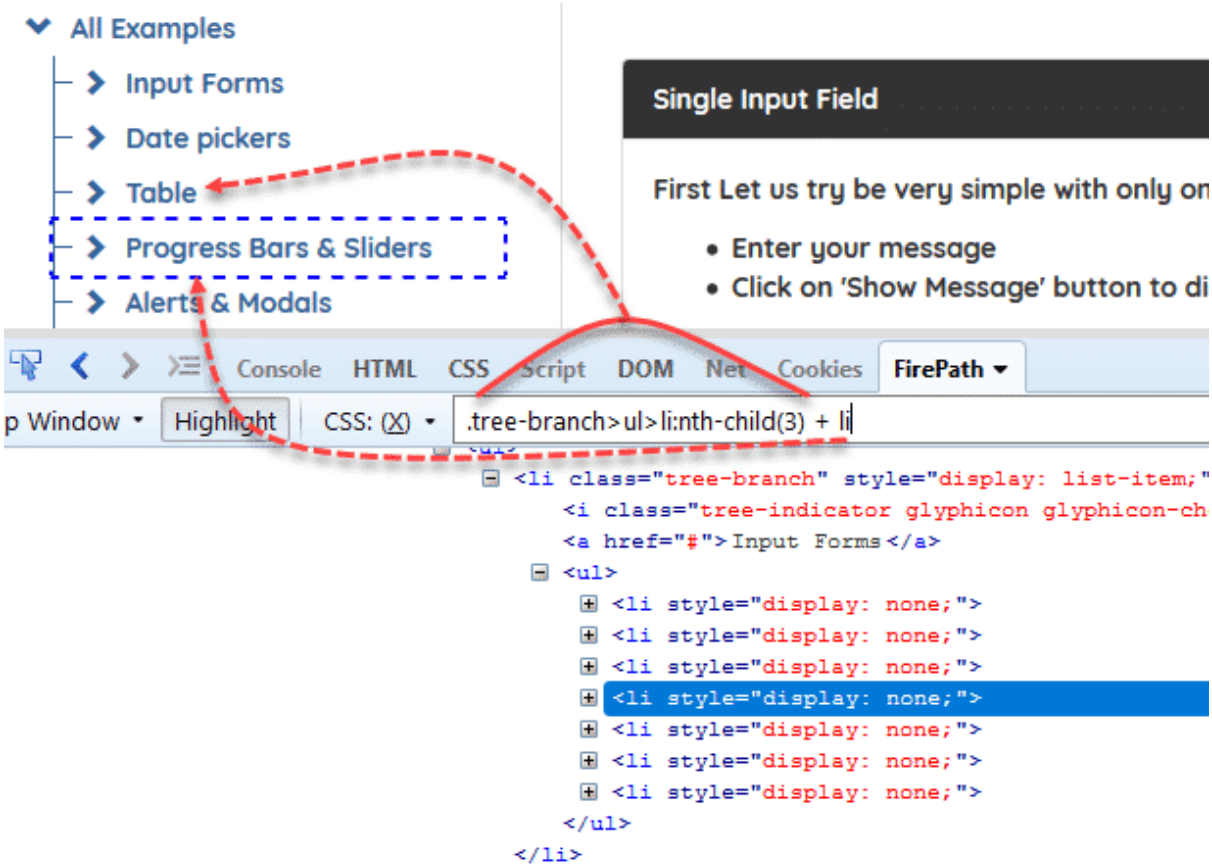
```
<ul id="Cars">
<li id="mercedes">Mercedes made in Germany!</li>
<li>BMW</li>
<li>Porsche</li>
</ul>
```

Syntax: li#mercedes + li

'**li#automation + li**' will first go to li element with id 'mercedes' and **then select its adjacent li** which is the 'BMW' list item.

Example

Syntax: **.tree-branch>ul>li:nth-child(3) + li** (*It will select the next element.*)



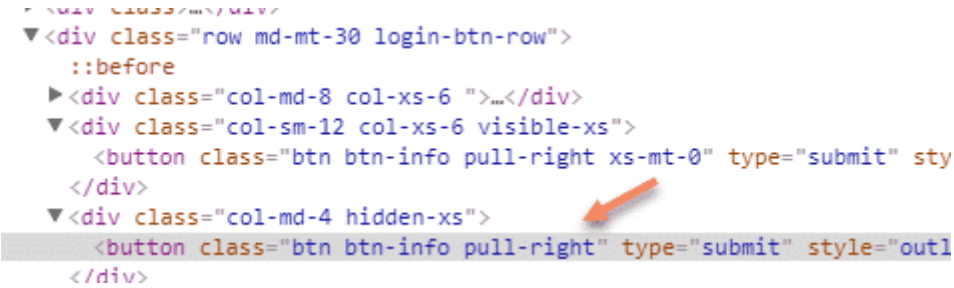
17) Exclude a CSS Class Name in Selenium CSS Locators

You can exclude any of the class names with **:not(.class-name)** syntax.

Example:

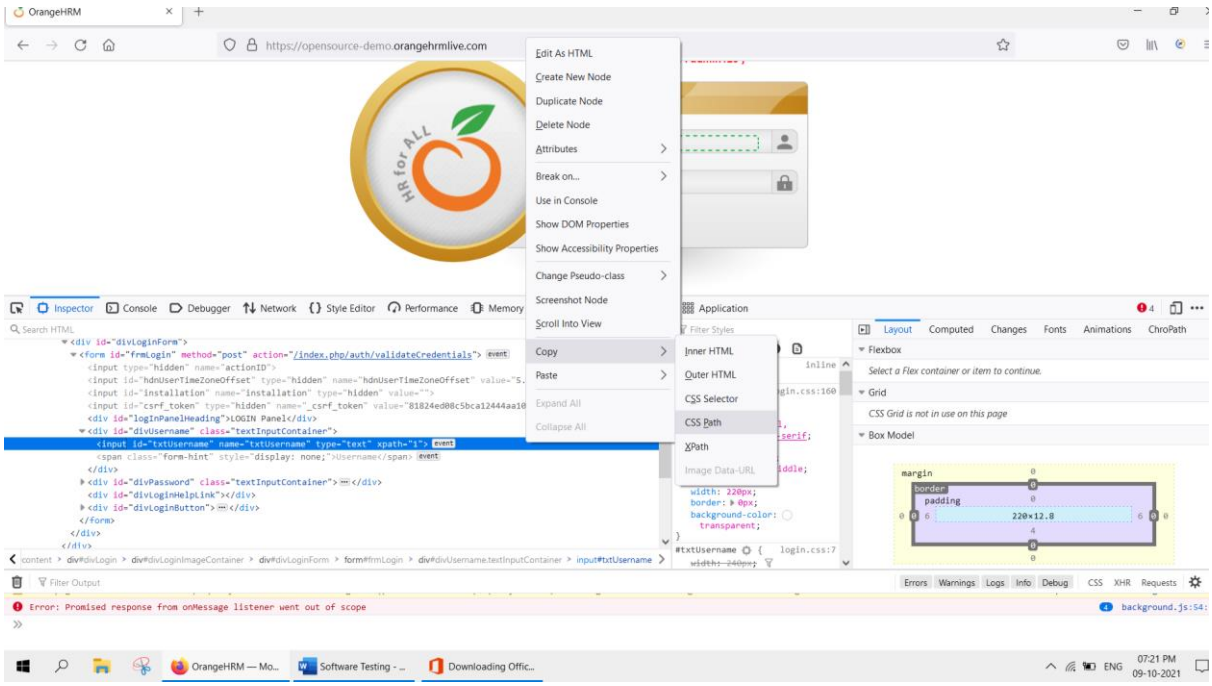
.btn.btn-info.pull-right:not(.xs-mt-0)

The above selector excludes "**xs-mt-o**" class and selects the below the line as shown below figure.



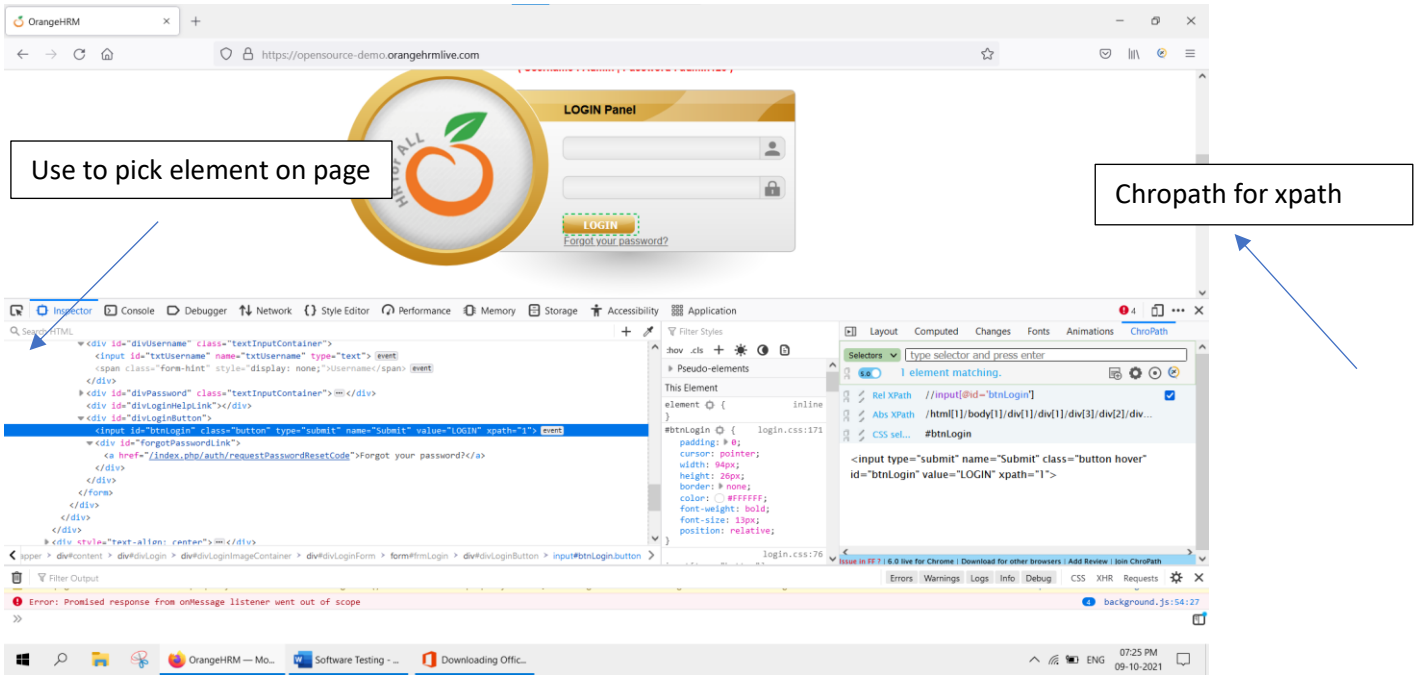

https://opensource-demo.orangehrmlive.com
OrangeHRM
OPEN SOURCE HR MANAGEMENT
(Username : Admin | Password : admin123)
LOGIN Panel
HR for ALL
LOGIN
Forgot your pa
Manage Logins
Undo
Redo
Cut
Copy
Paste
Delete
Select All
Check Spelling
Inspect Accessibility Properties
Inspect (Q)
OrangeHRM 4.
© 2005 - 2021 OrangeHRM, Inc. All rights reserved.

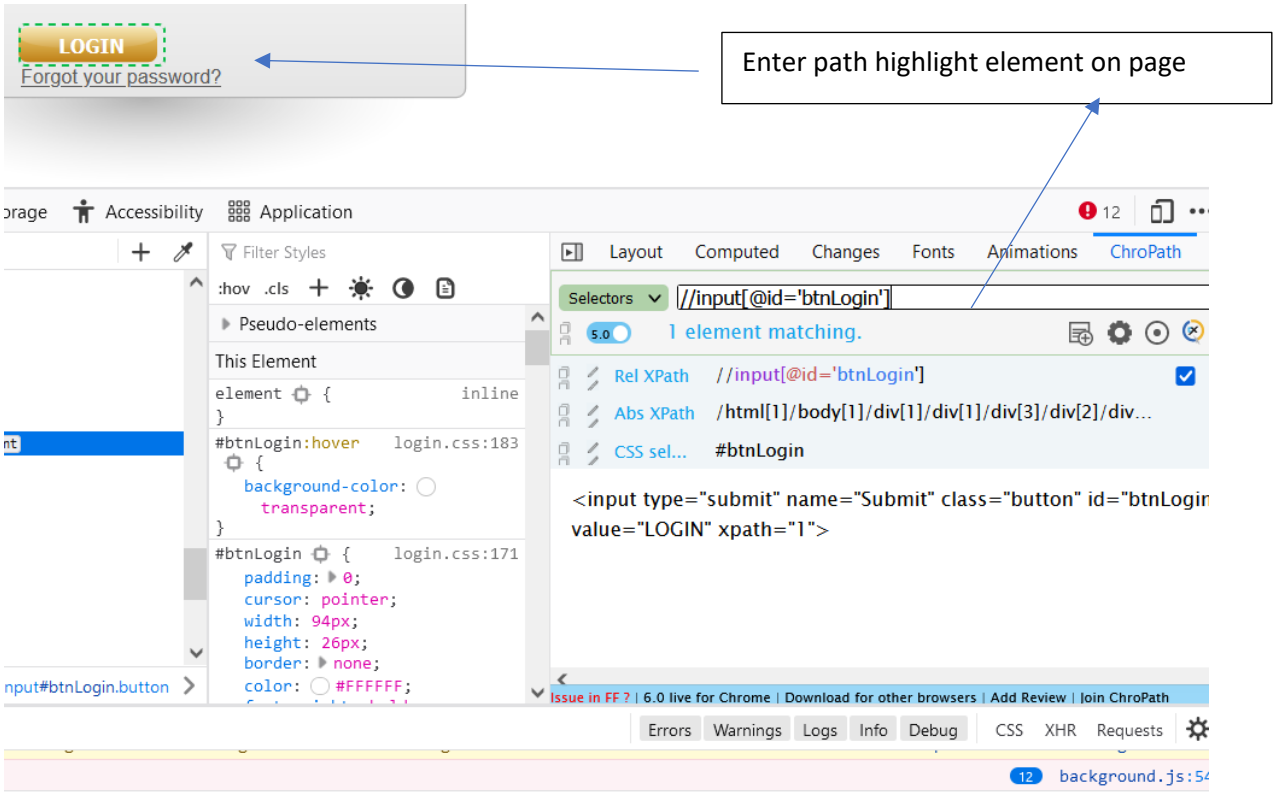
Right click on selected blue line >> copy >> xpath



//*[@id="txtUsername"]

In below screen





Xpath

Absolute XPath:

It is the direct way to find the element, but the disadvantage of the absolute XPath is that if there are any changes made in the path of the element then that XPath gets failed.

Starts with == single forward slash(/)

Absolute XPath:

```
/html/body/div[2]/div[1]/div/h4[1]/b/html[1]/body[1]/div[2]/div[1]/div[1]/h4[1]/b[1]
```

Relative Xpath:

Relative Xpath starts from the middle of HTML DOM structure. It starts with double forward slash (//). It can search elements anywhere on the webpage, means no need to write a long xpath and you can start from the middle of HTML DOM structure. Relative Xpath is always preferred as it is not a complete path from the root element.

- Starts with //
- No need to form start , start form middle

- Some more basic xpath expressions:
- `Xpath=//input[@type='text']`
- `Xpath=  //label[@id='message23']`
- `Xpath=  //input[@value='RESET']`
- `Xpath=//*[@class='barone']`
- `Xpath=//a[@href='http://demo.guru99.com/']`
- `Xpath= //img[@src='//cdn.guru99.com/images/home/java.png']`

@ use for attiribute and note that attribute and its values are in [ ] square bracket

driver.findElement(By.xpath("//input[@value='Banana']")).click();

here find the element here input tag but which has attribute value and its value is banana

get commands

---

get() --> opens the URL on the browsers

getTitle() --> returns the page tile

getCurrentURL() -->returns the URL of the page

getPageSource() --> returns HTML code for page

---

1) Find how many options present in drop down

dropdown.getOptions().size()

2) Extract all the options and print them

dropdown.getOptions()

3) Select option from the dropdown

dropdown.selectByvisibleText(option)

dropdown.selectByIndex(index)

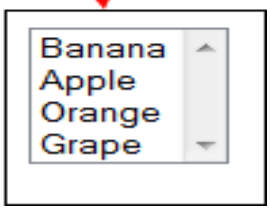
dropdown.selectByValue(value)

Dropdown/combobox/list box

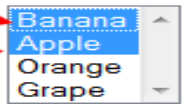
page source

```
<select id="fruits" multiple="">
    <option value="banana">Banana</option>
    <option value="apple">Apple</option>
    <option value="orange">Orange</option>
    <option value="grape">Grape</option>
</select>
```

HTML page




```
public static void main(String[] args) {

    WebDriver driver = new FirefoxDriver();

    driver.get("http://jsbin.com/osebed/2");
    Select fruits = new Select(driver.findElement(By.id("fruits")));
    fruits.selectByVisibleText("Banana");
    fruits.selectByIndex(1);
}
```



isMultiple()

- Returns TRUE if the drop-down element allows multiple selections at a time; FALSE if otherwise.
- **No parameters needed**

```
if (drpCountry.isMultiple()) {
    //do something here
}
```

Here is the complete code

```
package newpackage;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.By;

public class accessDropDown {
 public static void main(String[] args) {
                System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");
            String baseURL = "http://demo.guru99.com/test/newtours/register.php";
            WebDriver driver = new FirefoxDriver();
                driver.get(baseURL);

                Select drpCountry = new Select(driver.findElement(By.name("country")));
                drpCountry.selectByVisibleText("ANTARCTICA");

                //Selecting Items in a Multiple SELECT elements
                driver.get("http://jsbin.com/osebed/2");
                Select fruits = new Select(driver.findElement(By.id("fruits")));
                fruits.selectByVisibleText("Banana");
                fruits.selectByIndex(1);
 }
}
```

Select is class pre defined in class for combo box

Radio buttons/Checkboxs

1) Check the status selected or not

2) Select option - click()

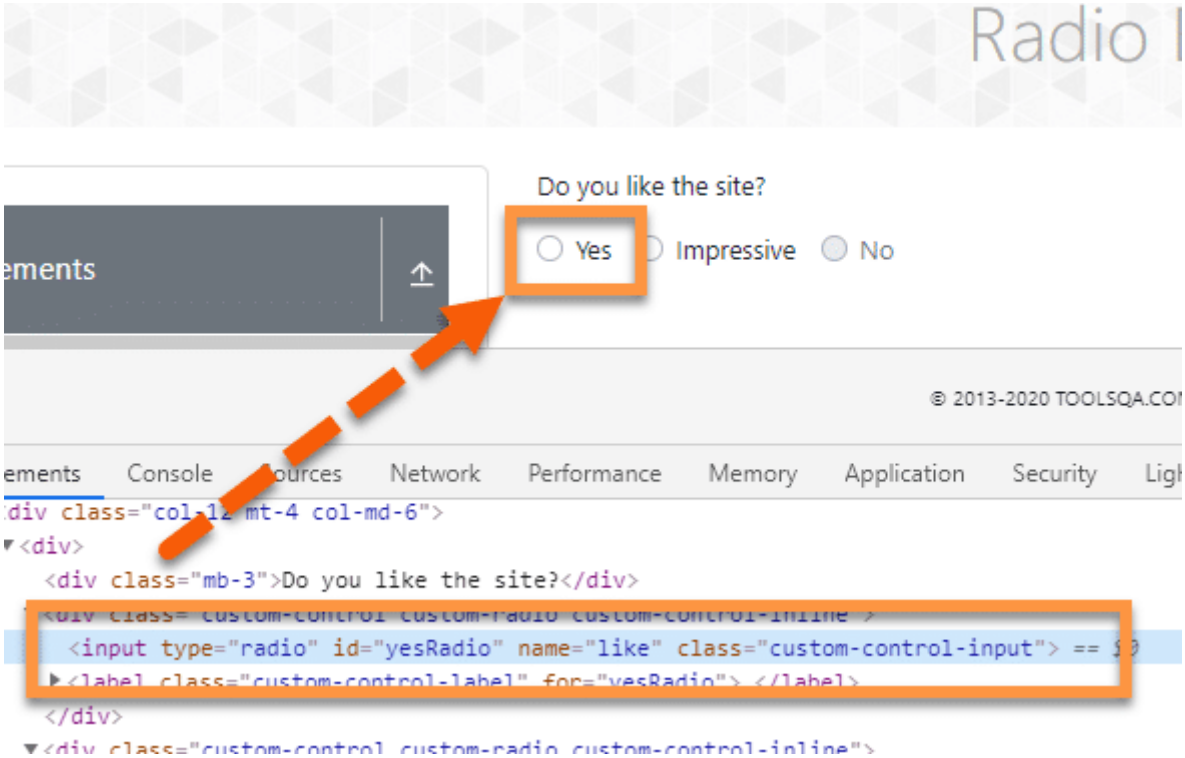
Conditional Commands (Retirns a boolean value- true/false)

isDisplayed() on page or not

isEnabled()

isSelected() // for radio buttons and check boxes

```
WebElement radmale=driver.findElement(By.id("RESULT_RadioButton-8_0"));

System.out.println(radmale.isDisplayed()); // check displayed or not - true

System.out.println(radmale.isEnabled()); //checks enable or not - true

System.out.println("Before selecting radio button , the status is:"+radmale.isSelected()); // false

radmale.click(); // select radio button

System.out.println("After selecting radio button , the status is:"+radmale.isSelected()); //true
```



Now, if we use the ***ID locator*** to recognize the element and perform the ***click*** operation, we will need to use the following *Selenium code:*

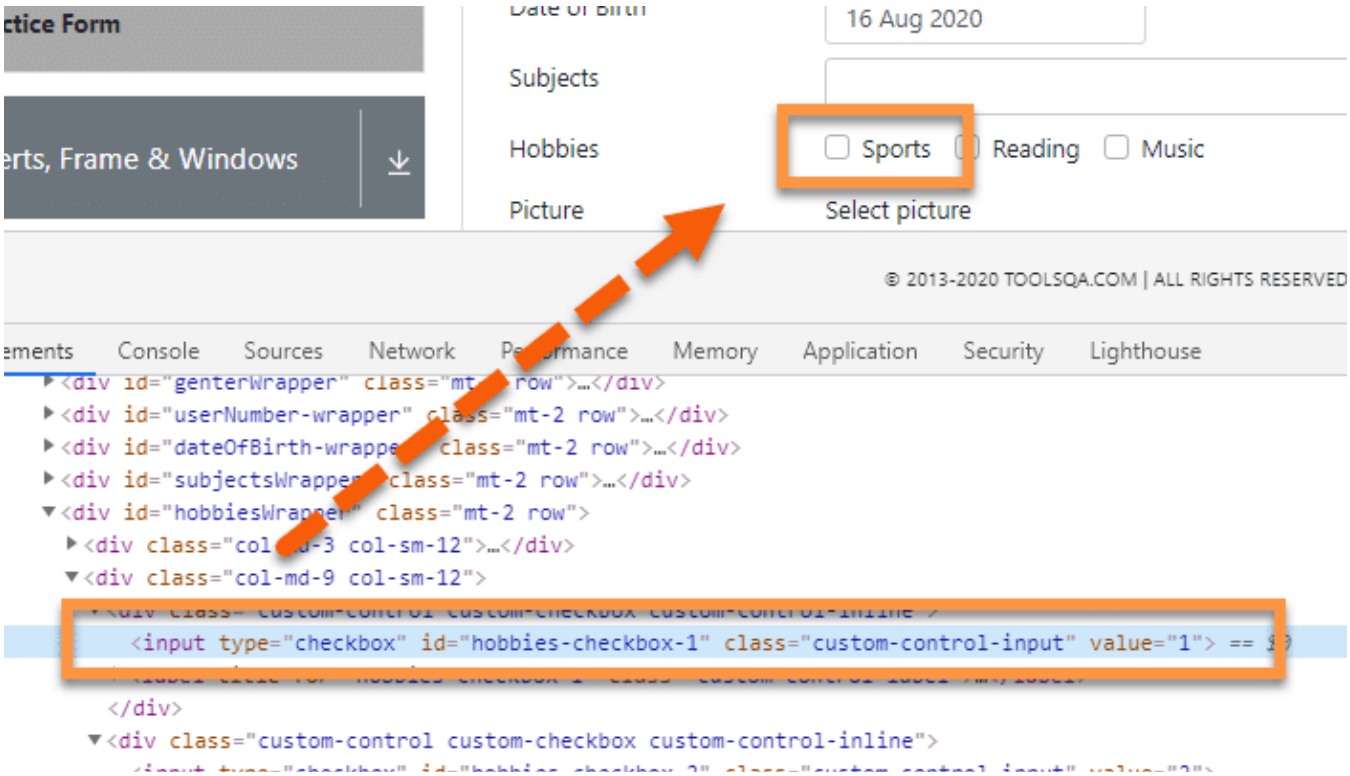
```
driver.findElement(By.id("yesRadio")).click();          //BY ID

OR
driver.findElement(By.name("like")).click();        // BY NAME

OR

driver.findElement(By.xpath("//div/input[@id='yesRadio']")).click();
```

OR

driver.findElement(By.cssSelector("input[id='yesRadio']")).click();

```
/**
* Validate Radio button using isSelected() methiod
*/

WebElement radioElement = driver.findElement(By.id("impressiveRadio"));
boolean selectState = radioElement.isSelected();

//performing click operation only if element is not selected
if(selectState == false) {
        radioElement.click();
}
```

Check box



driver.findElement(By.id("hobbies-checkbox-1")).click();

Links

driver.findElement(By.linkText("Software Testing Tutorials")).click();

Thread.sleep(3000);

Every time you click action then use Sleep () because speed is slow so wait

this will click on the link >> software testing tutorial

now again back to previous page as

:- driver.navigate().back()

driver.findElement(By.partialLinkText("Tools Training")).click();

Navigational commands

navigate().to(URL) --> same as driver.get()

navigate.back()

navigate.forward()

navigate.refresh()

eg driver.navigate().refresh();

fetch the links *(tags)* code

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.Iterator;
import java.util.List;

public class GetAllURLs {
   public static void main(String[] args) {

      //Create WebDriver instance and open the website.
      System.setProperty("webdriver.chrome.driver","./src/main/resources/chromedriver");
      WebDriver driver = new ChromeDriver();
      driver.manage().window().maximize();
      driver.get("https://demoqa.com/links");

      String url="";
      List<WebElement> allURLs = driver.findElements(By.tagName("a"));
      System.out.println("Total links on the Wb Page: " + allURLs.size());

      //We will iterate through the list and will check the elements in the list.
```

```
        Iterator<WebElement> iterator = allURLs.iterator();
        while (iterator.hasNext()) {
            url = iterator.next().getText();
            System.out.println(url);
        }

       //Close the browser session
        driver.quit();
      }
}
```

How to Handle IFrame / IFrames with Selenium WebDriver

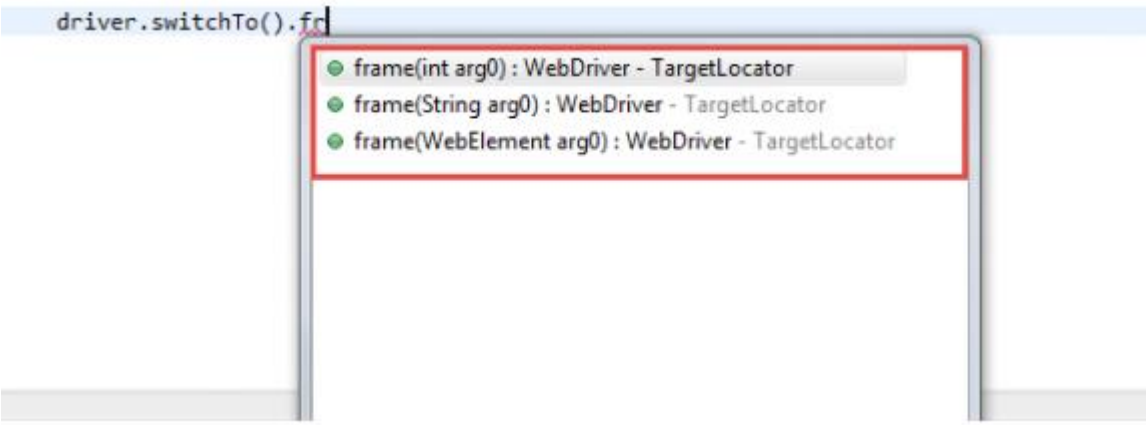
Eg . Form is inserted in frame

Before starting we have to understand that to work with different iFrames on a page we have to switch between these iFrames. To Switch between iFrames we have to use the driver's ***switchTo().frame*** command

This can be done in the three way as following

- ***switchTo.frame*** (*int frameNumber*) : Pass the frame index .

  switchTo.frame (*string frameNameOrId*): Pass the frame element Name or ID
- ***switchTo.frame***(*WebElement frameElement*): Pass the frame web element and



(method overloaded same method with different parameter , overloading = compile time polymorphism)

Now

- What is a frame index?
- How to get total number of frames on a webpage?

There are two ways

First by executing a JavaScript

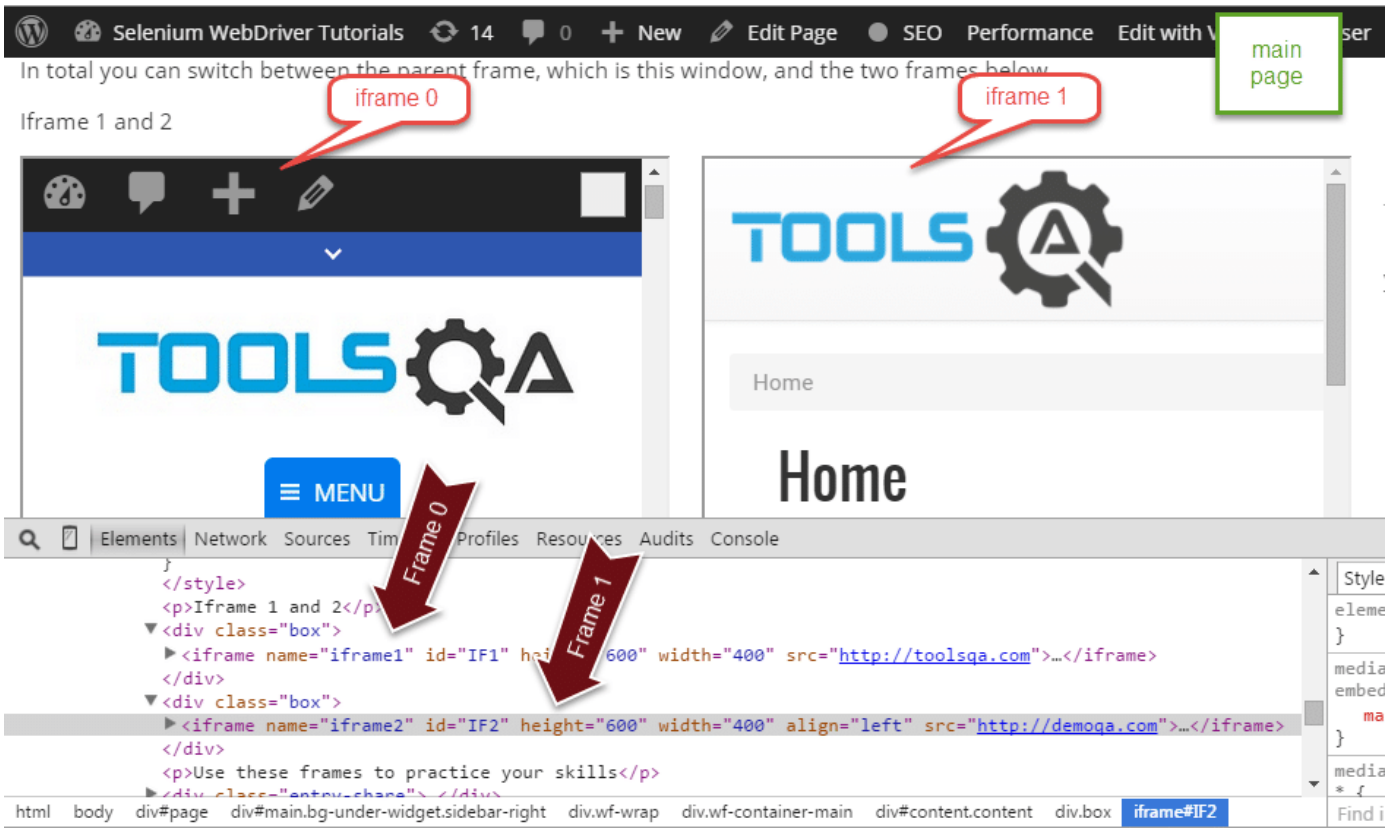
second is by finding the total number of web elements with a tag name of iFrame

```
//By finding all the web elements using iframe tag
List<WebElement> iframeElements = driver.findElements(By.tagName("iframe"));

System.out.println("The total number of iframes are " + iframeElements.size());
```

Switch to Frames by Index

Index of an iFrame is the position at which it occurs in the HTML page. In the above example, we have found a total number of iFrames. In the sample page, we have two IFrames, index of iFrame starts from 0. So there are two iFrames on the page with index 0 and 1.

Index 0 will be the iFrame which exists earlier in the HTML page. Refer to the image below:



to switch to *0th iframe*

```
public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new FirefoxDriver();
        driver.get("https://toolsqa.com/iframe-practice-page/");

        //Switch by Index
        driver.switchTo().frame(0);
        driver.quit();
}
```

Switch to 1st frame in HTML page

```
WebDriver driver = new FirefoxDriver();
driver.get("https://toolsqa.com/iframe-practice-page/");

//Switch by frame name
driver.switchTo().frame("iframe1");
driver.quit();
```

Switch by name

```
//Switch by frame ID
driver.switchTo().frame("IF1");
driver.quit();
```

Switch by id

```
//First find the element using any of locator stratedgy
WebElement iframeElement = driver.findElement(By.id("IF1"));

//now use the switch command
driver.switchTo().frame(iframeElement);
driver.quit();
```

Switch by WebElement

Switching back to the Main page from Frame

```
//Do all the required tasks in the frame 0
//Switch back to the main window
driver.switchTo().defaultContent();
driver.quit();
```

Driver.findElements() = find multiple elements

findElement() = single element

Wait Command

If network is slow then all page web element are not loaded fully it requires time to load

Hence we need to wait or stop execatuion of selenium script

Thread.sleep(milli seconds) :-

Slow performance because if element is avibalbe it will wait for times specified

implicitWait

driver.manage().timeouts().implicitlyWait(5,TimeUnit.SECONDS);

if element is ready it will execute program no wait for its time

explicitwait

based upon condition

Explicit waits are used to halt the execution until the time a particular condition is met or the maximum time has elapsed

WebDriverWait is class

```
WebDriverWait mywait=new WebDriverWait(driver,10000);

mywait.until(ExpectedConditions.visibilityOfElementLocated(By.name("userName")));
```

case1: locator matches multiple elements, findElement() ----> this can be located single element

case2: locator mathces multiple elements, findElements() --> this can be located multiple elements

case3: locator matches single elemetn, findElements() ---> this can be located singel element

Alerts/popus (switch between alerts)

```
driver.switchTo().alert().accept();          //closes popup by using OK button
driver.switchTo().alert().dismiss();         //closes popup by using Cancel button
```

```
driver.switchTo().frame(name)
driver.switchTo().frame(index)
driver.switchTo().frame(WebElement)
driver.switchTo().defaultContent();
```

Handle browser windows(switch between browser windows)

handle --> an ID for a browser window

```
driver.getwindowsHandle()
driver.getwindowsHandles()    // for multipe window
```

:- each tab or new browser is unique id

Action Class

Action Class in Selenium is a built-in feature provided by the selenium for handling keyboard and mouse events. It includes various operations such as multiple events clicking by control key, drag and drop events and many more. These operations from the action class are performed using the advanced user interaction API in Selenium Webdriver.

```
Actions builder = new Actions(driver);
```

Action act = new action(driver)

act.moveToElement(admin).moveToElement(usermgnt).moveToElement(users).click().build().perform();

Day-17

Mouse Actions

Mouse Hover

Right Click

Double Click

Drag and drop

MouseHover

----- act.moveToElement(webelement)

Right click

act.contextClick(webelement)

Double click

act.doubleClick(Webelement)

Drag and drop

act.dragAndDrop(source_element, target_element)

Resizing/Slider

---

act.moveToElement(element).dragAndDropBy(element,400, 0)

Keyboard Actions

---

act.sendKeys(Keys.CONTROL+"a")

Scrolling page

approach1

```
js.executeScript("window.scrollBy(0,2000)","");
```

approach2

```
WebElement
flag=driver.findElement(By.xpath("//*[@id=\"content\"]/div[2]/div[2]/table[1]/tbody/tr[86]/td[1]/img"));
js.executeScript("arguments[0].scrollIntoView();",flag);
```

approach3

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

Day-18

1) How to upload files

2) How to download files

Using :

1) AutoIT

2) Sikuli

How to upload files using WebDriver+Sikuli

---

1) Download Sikuli jar file

https://launchpad.net/sikuli/sikulix/1.1.2/+download/sikulixsetup-1.1.2.jar

2) Add jar file to project build path

Right Click on the project -> Build Path -> Configure Build Path

3) write code for upload file scenario in webdriver.

How to download files

Mime types for firefox browser

https://www.sitepoint.com/mime-types-complete-list/

Day-19

Data Driven testing using Excel

Download Apache poi for Microsoft Excel

http://www-us.apache.org/dist/poi/release/bin/poi-bin-3.17-20170915.zip

Excel File-->Workbook-->Sheet-->Row-->Cell

FileInputStream

FileOutputStream

1) How to read data from excel file.

2) How to write data into excel

1) count rows

2) count cells

3) read data from cels

4) write data into cells

XLUtils.java

---

countingrows()

countingcells()

readcellvalue()

writecellvalue()

ChromeDriver is a standalone server executable that implements WebDriver's JSON-wire protocol and works as a glue between the test script and Google Chrome, as shown in the following diagram: