**import** org.openqa.selenium.Dimension;

**import** org.openqa.selenium.Point;

Point p = driver.manage().window().getPosition();

Dimension d = driver.manage().window().getSize();

driver.manage().window().setPosition(**new** Point((d.getHeight()-p.getX()), (d.getWidth()-p.getY())));

**try** {

Thread.*sleep*(3000);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

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

**Web element locators:**

**Id:**

Demo url 🡪 https://demoqa.com/text-box

driver.get("https://demoqa.com/text-box");

driver.findElement(By.id("userName")).sendKeys("Selenium");

**Classname:**

Demo url 🡪[**https://demoqa.com/radio-button**](https://demoqa.com/radio-button)

driver.get("https://demoqa.com/radio-button");

driver.findElement(By.*className*("custom-control-label")).click();

**Name:**

Demo url 🡪 <https://www.saucedemo.com/>

Use the field using user name.

**Link text:**

Demo url 🡪 <https://demoqa.com/links>

Use the field home link

**Partial Link text:**

Demo url 🡪 <https://demoqa.com/links>

Use the field bad Request link with text as “bad Req” instead of just “Bad”

**Tag name:**

Demo url 🡪 <https://www.google.com/>

driver.findElement(By.*tagName*("input")).sendKeys("Selenium");

**CSS Selector:**

**Using tagname with id:**

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

driver.findElement(By.*cssSelector*("input#user-name")).sendKeys("Selenium");

**using tagname with class:**

driver.get("https://demoqa.com/radio-button");

driver.findElement(By.*cssSelector*("label.custom-control-label")).click();

**https://demoqa.com/text-box**

By.cssSelector("input[id= ‘userName’]")

**XPath in Selenium:**

XML Path used for navigation through the HTML structure of the page. It is a syntax or language for finding any element on a web page using XML path expression.

Diagram

Description automatically generated

**2 types of XPath**

1. Absolute XPath
2. Relative XPath

Demo site: <http://demo.guru99.com/test/selenium-xpath.html>



Find the XPath for the field “Testing” in the above link as shown below

**Absolute Path** 🡪 “/html/body/div[4]/div[1]/div/h4[1]”

**Relative Path** 🡪 //b[contains(text(),'Testing')]

**XPath axes:**

XPath axes search different nodes in XML document from current context node. XPath Axes are the methods used to find dynamic elements, which otherwise not possible by normal XPath method having no ID, Classname, Name, etc.

1. **Basic XPath:**

For the field “UserID” in the above link

XPath=//input[@name='uid']

XPath=//input[@type='text']

1. **Contains:**

Contains() is a method used in XPath expression. It is used when the value of any attribute changes dynamically, for example, login button

The **contain** feature has an ability to find the element with partial text

XPath 🡪 //\*[contains(@type,'sub')] or //input[contains(@type,’sub’] or //\*[contains(@value,'LOG')]

With Full text

Xpath=//\*[contains(@name,'btn')] or //\*[contains(@name,'btnLogin')]

1. **Using OR & AND:**

For this example let us take the “Login” and “Reset” button

XPath=//\*[@type='submit' or @name='btnReset'] 🡪 both login and reset button will get highlighted

XPath = //input[@type='reset' and @name='btnReset'] 🡪 only reset button will get highlighted

1. **XPath using Starts-with:**

**XPath starts-with()** is a function used for finding the web element whose attribute value gets changed on refresh or by other dynamic operations on the webpage. For example, if you have the locator id has value as shown below, then XPath will be

Id=” message12″

Id=” message345″

XPath=//label[starts-with(@id,'message')]

1. **XPath using Text():**

The field label “UserID” can be found using the following xpath

Xpath=//td[text()='UserID'] used when we want to get the text of the field name

1. **XPath axes methods:**

These XPath axes methods are used to find the complex or dynamic elements.

1. **Following:**

Selects all elements in the document of the current node( ) [ UserID input box is the current node] as shown in the below screen.

Xpath=//\*[@type='text']//following::input

Note: When we get multiple nodes for the same, then use the node number as in

Xpath=//\*[@type='text']//following::input[1]

1. **Ancestor:**

The ancestor axis selects all ancestor’s element (grandparent, parent, etc.) of the current node as shown in the below screen.

In the below expression, we are finding ancestors element of the current node(“ENTERPRISE TESTING” node)

Xpath=//\*[text()='Enterprise Testing']//ancestor::div

1. **Child:**

Selects all children elements of the current node (Java) as shown in the below screen.

Xpath=//\*[@id='java\_technologies']//child::li

1. **Preceding:**

Select all nodes that come before the current node as shown in the below screen

In the below expression, it identifies all the input elements before “LOGIN” button that is **Userid** and **password** input element.

Xpath=//\*[@type='submit']//preceding::input

1. **Following-Sibling:**

Select the following siblings of the context node. Siblings are at the same level of the current node as shown in the below screen. It will find the element after the current node.

xpath=//\*[@type='submit']//following-sibling::input

1. **Parent:**

Selects the parent of the current node as shown in the below screen.

Xpath=//\*[@id='rt-feature']//parent::div

Rest of the stuff 🡪 please go through <https://www.guru99.com/xpath-selenium.html#basic-xpath>

Web Element Commands

getAccessibleName 🡪 will give you the name of the field

getAriaRole 🡪 will give you the type of the field like textbox, button etc

getAttribute 🡪 will give the value of the given attribute

getCssValue 🡪 will give the value for the attributes mentioned under the styles tab

getDomAttribute & getDomProperty 🡪 will give the value for the attributes mentioned under the respective tag

getRect() 🡪 gives the X and Y co-ordinates as well as height and width of the element. It is useful when we want to perform a click operation based on the co-ordinates using actions class

actions.moveByOffset(getX+1, getY+1).click();

actions.build().perform();