**Prerequisites:**

Learn HTML from W3C

https://www.w3schools.com/htmL/

**How to Write Effective XPath Selenium Selectors**

Refer Link: <https://www.swtestacademy.com/xpath-selenium/>

We can find the location of any element on a web page using XML path expressions. The basic syntax for XPath is shown below:

**Syntax = //tagname[@attribute=’Value‘]**

Example = //input[@id=’user-message‘]

You can also use class, name, link text, and the other attributes to locate an element with XPath as shown above.

**Absolute and Relative XPath**

**Absolute XPath**

* It is a direct way to locate an element.
* It is very brittle.
* Starts with single slash “/” that means starting to search from the root.

Navigate to <https://login.yahoo.com/>

Below is the Absolute XPath for the email text

Example: /html/body/div[2]/div/div/form/div/input

**Relative XPath**

* Starts from the middle of the HTML DOM.
* Starts with a double slash “//” that means it can start to search anywhere in the DOM structure.
* Shorter than Absolute XPath.
* Less fragile.

Examples:

Navigate to <https://login.yahoo.com/>

Below is the Relative XPath for the email text

//\*[@id=’login-username’]

**Handling Dynamic objects/ids on the page**

**1) Writing XPath with Tag & Attribute & Value Trio**

Syntax: //tag[@attribute=’value‘]

Example: //input[@id, ‘user-message’]

Examples:

Navigate to <https://login.yahoo.com/>

Below is the Relative XPath for some of the elements

//\*[@id="login-signin"]

//\*[@id="mbr-forgot-link"]

//a[@id="createacc"]

**2) Writing XPath with contains()**

When an attribute of an element is dynamic, then you can use contains() for the constant part of the web element but also you can use contains() in any condition when you need.

Syntax: //tag[contains(@attribute, ‘value‘)]

Navigate to link <https://login.yahoo.com/> in chrome and search for below xpath

Example: //input[contains(@id,'login-username')]

//input[contains(@name,'username')]

–> It interrogates “'username'” for all name attributes in the DOM.

//\*[contains(text(),'sign')]

–> It interrogates the text “sign” in the DOM.

**3) Writing XPath with starts-with**

This method checks the starting text of an attribute. It is very handy to use when the attribute value changes dynamically but also you can use this method for non-changing attribute values.

Syntax: //tag[starts-with(@attribute, ‘value‘)]

Navigate to link <https://login.yahoo.com/> in chrome and search for below xpath

Example: //input[starts-with(@id, 'login-username')]

**4) Writing XPath with Chained XPaths Declarations**

We can chain multiple relative XPath declarations with “//” double slash to find an element location as shown below.

Navigate to link <https://login.yahoo.com/> in chrome and search for below xpath

Example: //div[@id='username-country-code-field']//div[@class='country-code-dropdown country-dropdown-container hide']

**5) Writing XPath with “or” Statement**

In this method, we use two interrogation conditions such as A and B and return a result-set as shown below:

A and B are treated as two independent XPath

A B Result

False False No Element

True False Returns A

False True Returns B

True True Returns Both

“or” is case-sensitive, you should not use capital “OR”.

Syntax: //tag[XPath Statement-1 or XPath Statement-2]

Navigate to link <https://login.yahoo.com/> in chrome and search for below xpath

Example: //\*[@id="login-signin" or @id='mbr-forgot-link']

**6) Writing XPath with “and“ Statement**

In this method, we use two interrogation conditions such as A and B and return a result-set as shown below:

A and B are attributes of an element

A B Result

False False No Element

True False No Element

False True No Element

True True Returns Both

“and” is case-sensitive, you should not use capital “AND”.

Syntax: //tag[XPath Statement-1 and XPath Statement-2]

Navigate to link <https://login.yahoo.com/> in chrome and search for below xpath

Example: //\*[@id=’user-message’ and @class=’form-control’]

**7) Writing XPath with text()**

We can find an element with its exact text.

Syntax: //tag[text()=’text value‘]

Navigate to link <https://login.yahoo.com/> in chrome and search for below xpath

Example:

//\*[text()='Terms']

//\*[text()='(Updated)']



**Locator Strategy**

Selenium gives user options to locate elements in 9 different ways.

* Id
* Name
* Linktext
* Partial Linktext
* Tag Name
* Class Name
* CSS (Cascaded Spread Sheets)
* XPath (XML path)

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** Locator\_Strategy {

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

WebDriver driver = **new** ChromeDriver();

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

WebElement textbox1 = driver.findElement(By.*id*("login-username"));

textbox1.sendKeys("Searching element by ID");

WebElement textbox2 = driver.findElement(By.*name*("username"));

textbox2.sendKeys("Searching element by name");

Thread.*sleep*(3000);

WebElement linktext1 = driver.findElement(By.*linkText*("Trouble signing in?"));

linktext1.click();

Thread.*sleep*(3000);

driver.navigate().back();

WebElement linktext2 = driver.findElement(By.*partialLinkText*("Trouble"));

linktext2.click();

List<WebElement> tagname = driver.findElements(By.*tagName*("a"));

**for** (**int** i = 0; i < tagname.size(); i++) {

System.***out***.println(i + " " + tagname.get(i).getText());

}

Thread.*sleep*(3000);

driver.navigate().back();

WebElement classname = driver.findElement(By.*className*("phone-no "));

classname.sendKeys("using classname");

;

WebElement xpath = driver.findElement(By.*xpath*("//input[@name='username']"));

xpath.sendKeys("using xpath");

}

}