In the above syntax:

- //: It is use to select the current node.
- tagname: It's the name of the tag of a particular node.
- @: It is used to select attributes.
- Attribute: It is the name of the attribute of the node.
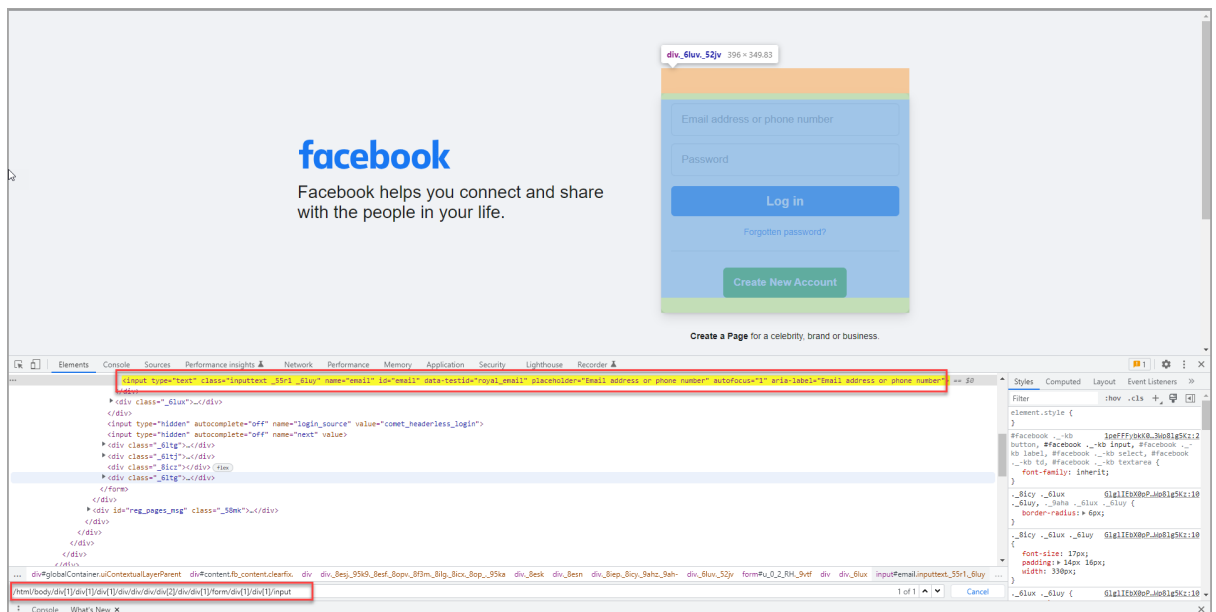- Value: It is the value of the attribute.

What is Xpath?

- Xpath is defined as an XML path.
- It is a syntax or language of finding any element on the web page using XML path expression.
- Xpath is used to find the location of any element on a webpage using HTML DOM structure.

Type of Xpath?

- **There are 2 types of Xpath**
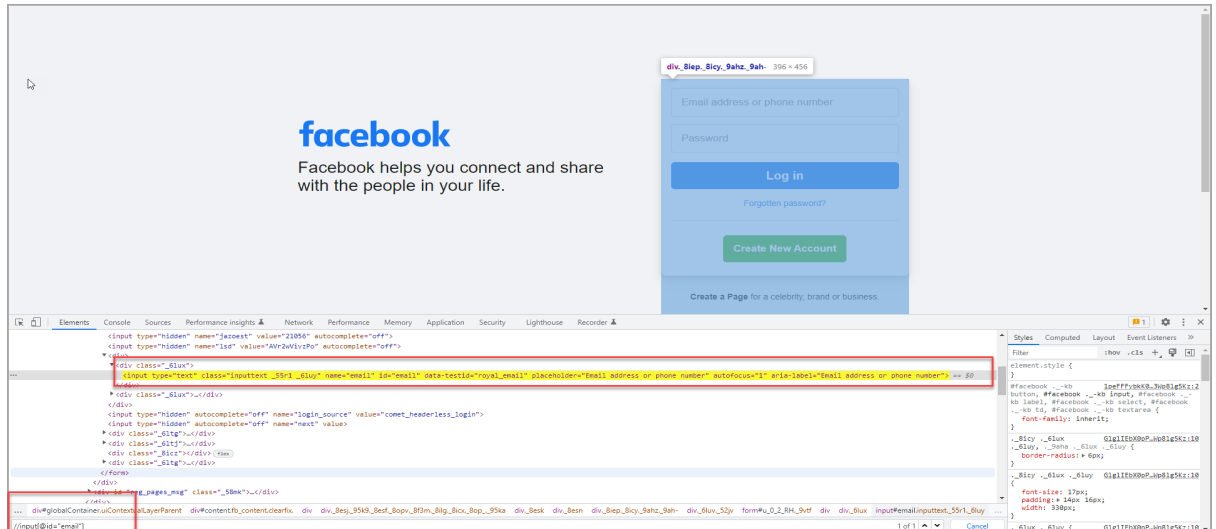 - Absolute Xpath
  - Relative Xpath

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.
- It begins with the single forward slash (/), which means you can select the element from the root node.
- Below it is the example of an absolute XPath expression of the element shown in the below screen.

- **/html/body/div[1]/div[1]/div[1]/div/div/div/div[2]/div/div[1]/form/div[1]/div[1]/input**

Relative Xpath?

- For Relative Xpath the path starts from the middle of the HTML DOM structure.
- It starts with the double forward slash (//) which means it can search the element anywhere on the webpage.
- **Ex: //input[@id="email"]**

Type of Xpath Functions

- There are so many methods defined in selenium to locate elements on a webpage but we are going to discuss only a few of them which are widely used in writing the automation script:

  ○ Contains() : //tag[contains(@attribute, 'value')]

  ○ starts-with() : //tag[starts-with(@attribute, 'value')]

  ○ text(): //tag[text()= 'value')]   //tag[contains(text(), 'value')]

  ○ Position() : //tag[(@attribute= 'value')] [position()=3]

  ○ last(): //tag[(@attribute= 'value')] [last()]

  ○ concat(): concat(string1, string2 [,stringn]*)

  ○ OR AND : //input[@type='email' and/ or @name='email']

  ○ index: //div[@class='google']/input[index]

- **Xpath axes**

  ○ Child : Xpath= //div[@class=' col-sm-12 google-sign-form']/child::input

  ○ descendant : Xpath= //div[@class=' col-LG-3 col-MD-4 col-sm-6 sign-form']//descendant:: *

  ○ descendant-or-self : Xpath= //div[@class=' col-lg-3 col-md-4 col-sm-6 sign-form']//descendant-or-self::*

  ○ Ancestor : Xpath= //input[@name='email]//ancestor::*

○ Parent : Xpath=//input[@name='password']//preceding::div/parent::*

○ Ancestor-or-self : Xpath= //input[@name='email]//ancestor-or-self::*

○ Preceding

○ Preceding-sibling: Xpath=//input[@name='password']//preceding-sibling::*

○ following : Xpath= //input[@name=' organization_name']//following::*

○ following-sibling: Xpath= //li[@class='sign-in']//following-sibling::*

○ **https://www.softwaretestingo.com/xpath-in-selenium-webdriver/**

CSS Selector Rule – #id

For the Sample HTML below:

<button id="submitButton1" type="button" class="btn">Submit</button>

CSS Locator – #submitButton1

Description – '#submitButton1' will select the element with id 'submitButton1'.

CSS Selector Rule – .class:

For the Sample HTML below:

<button id="submitButton1" type="button" class="btn">Submit</button>
CSS Locator – .btn

Description – '.btn' will select all the elements with class 'btn'.

CSS Selector Rule – tagName

For the Sample HTML below:

<input id="frame" type="text" name="firstName" class="textbox">
CSS Locator – input

Description – 'input' will select all the input type elements.

Using attributes and their value
CSS Selector Rule - [attributeName='attributeValue']

For the Sample HTML below:

<input id="frame" type="text" name="firstName" class="textbox">
CSS Locator – [name='firstName']

Description – [name='firstName'] will select the elements with name attribute having value 'firstName'.

CSS Selector Rule – tag#id

For the Sample HTML below:

<input id="frame" type="text" name="firstName" class="textbox">

CSS Locator – input#fname

Description – input#fname will select the 'input' element with id 'frame'.

CSS Selector Rule – tag.class

For the Sample HTML below-

\<input id="frame" type="text" name="firstName" class="textbox"\>
CSS Locator – input.textbox

Description – input.textbox will select the 'input' element with id 'textbox'.

CSS Selector Rule – tag[attributeName='attributeValue']

For the Sample HTML below:

\<input id="frame" type="text" name="firstName" class="textbox"\>
CSS Locator – input[name='firstName']

Description – input[name='firstName'] will select the 'input' element with 'name' attribute having value 'firstName'.

Locating Child Elements (direct child only)

CSS Selector Rule – parentLocator>childLocator

For the Sample HTML below:

\<div id="buttonDiv" class=" small"\>
\<button id="submitButton1" type="button" class="btn"\>Submit\</button\>
\</div\>
CSS Locator – div#buttonDiv>button

Description – 'div#buttonDiv>button' will first go to div element with id 'buttonDiv' and then select its child element – 'button'.Locating elements inside other elements (child or sub child)

CSS Selector Rule – locator1 locator2

For the Sample HTML below:

<div id="buttonDiv" class="small">
<button id="submitButton1" type="button" class="btn">Submit</button>
</div>
CSS Locator – div#buttonDiv button

Description – 'div#buttonDiv button' will first go to div element with id 'buttonDiv' and then select 'button' element inside it (which may be its child or sub child).

CSS Selector Rule – :nth-child(n)

For the Sample HTML below:

<ul id="testingTypes">
   Automation Testing
   Performance Testing
   Manual Testing

CSS Locator – #testingTypes li:nth-child(2)

Description – '#testingTypes li:nth-child(2)' will select the element with id 'testingType' and then locate the 2nd child of type li i.e. 'Performance Testing' list item.

Locating Siblings

CSS Selector Rule – locator1+locator2

For the Sample HTML below:

<ul id="testingTypes">
   <li id="automation">Automation Testing
   Performance Testing
   Manual Testing

CSS Locator – li#automation + li

Description – 'li#automation + li' will first go to li element with id 'automation' and then select its adjacent li i.e. 'Performance Testing' list item.

For handling dynamic elements having ids and other locators dynamically generated(not known beforehand). We can make use of the above locators by using different parent-sibling relationships of

the dynamic elements. Apart from this, we can also use some special CSS locators using which we can match partial values of the attributes.

^ – Starts with

CSS Selector Rule – [attribute^=attributeValue]

For the Sample HTML below:

<button id="user1_btn_263" type="button" class="btn">Submit</button>
CSS Locator – id^="user1″

Description – 'id^="user1″' will select the element whose id starts with "user1" value.

$ – Ends with

CSS Selector Rule – [attribute$=attributeValue]

For the Sample HTML below-

<button id="user1_btn_263" type="button" class="btn">Submit</button>
CSS Locator – id$="btn_263″

Description – 'id$="btn_263″' will select the element whose id ends with "btn_263" value

* – Contains

CSS Selector Rule – [attribute*=attributeValue]

For the Sample HTML below:

<button id="user1_btn_263" type="button" class="btn">Submit</button>
CSS Locator – id*="btn"

Description – 'id*="btn"' will select the element whose id contains with "btn" value.

1. [Xpath Helpler Chrome Extension](#)
2. [Xpath Helpler Wizard Chrome Extension](#)
3. [Xpather Chrome Extension](#)
4. [ChroPath](#)
5. [Ruto](#)

Go through ESS Profile to locate no smaller than 100 locators.

No	Category	Subcategory	Locator	Xpath	CSS
1	My information - Personal	Personal data	Form of Address		
2			First Name		
3			Last Name		
4		Address			
5		Education			
6					
….	My information - employment				
98					
99	My information - Account				
100					