XPATH

It is a language used when locating an XML(Extensible markup Language) nodes. Since HTML can be thought of as an implementation of XML

Xpath-Group of Nodes is called xpath

XML-

Each **node** of the tree is an **XML** element and is written with an opening and closing tag. An element can have one or more **XML** attributes . ... The standard for accessing and processing **XML** documents is the **XML** Document Object Model or DOM . The DOM represents elements, attributes and text within elements as **nodes** in a tree.

To find the element on web pages accurately there are different types of locators:

XPath Locators Find different elements on web page

ID-To find the element by ID of the element

Classname-To find the element by Classname of the element

Name-To find the element by name of the element

Link text-To find the element by text of the link

XPath- XPath required for finding the dynamic element and traverse between various elements of the web page

CSS path- CSS path also locates elements having no name, class or ID.

DemoLink- [**http://demo.guru99.com/test/selenium-xpath.html**](http://demo.guru99.com/test/selenium-xpath.html)

**Types of X-path**

There are two types of XPath:

**1) Absolute XPath**

**2) 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.

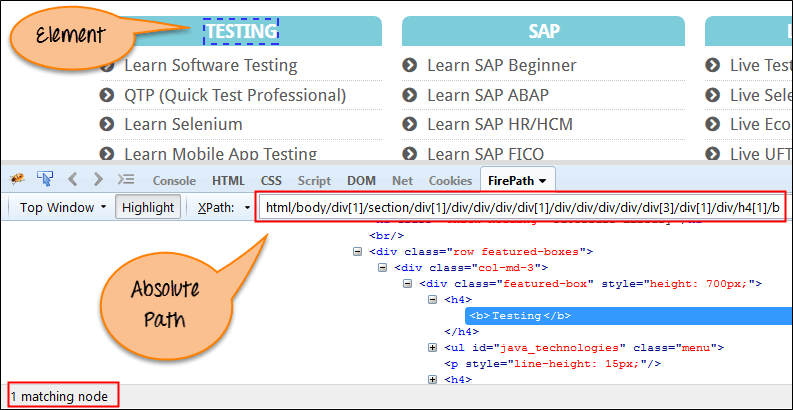
1.It Starts with single forward Slash(/)

2.'/' instructs the Xpath engine to search for the element with reference to the root node

3.Element identification is faster compared to Relative Xpath

4.Even with the slightest Change to the HTML DOM structure absolute xpath would Fail

5.Eg: html[1]/body[1]/div[2]/div[1]/div[1]/h4[1]/b[1]

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele2.png)

**Relative XPath** :

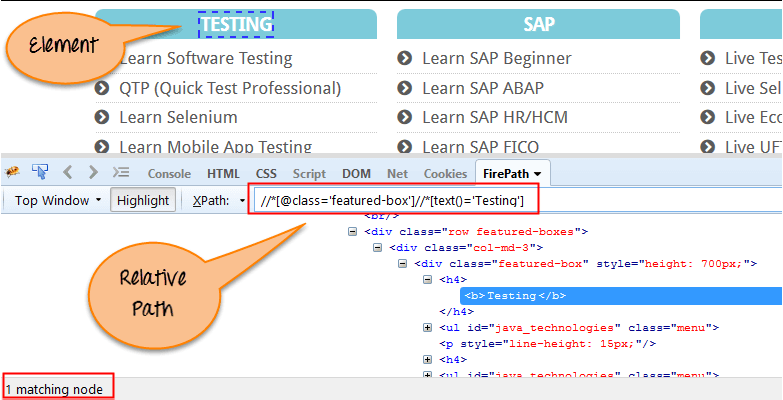
1.It Starts with double forward Slash(//)

2.'//' instructs the Xpath engine to search for the matching element anywhere in the DOM structure

3.Takes more Time to Identify the element as only a partial path is Specified

4. Relative Xpath is always preferred as it is not a complete path from the root element.

5.Eg: //div[@class='featured-box cloumnsize1']//h4[1]//b[1]

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele3.png)

Different ways of Relative Xpath in Selenium:

Relative Xpaths:

It begins from current location and is prefixed with'//'

synt://htmltagname[@attributeType='attributevalue']

1.xpath with single Attribute:

//htmltagname[@attributeType='attributevalue']

2.Multiple attribute:

//htmltagname[@attributeType='attributevalue'][@attributeType='attributevalue']

3.Text attribute:

//htmltagname[text()='textdetails']

4.Contains:

//htmltagname[contains(@attributeType,'attributevalue')]

5.Starts with:

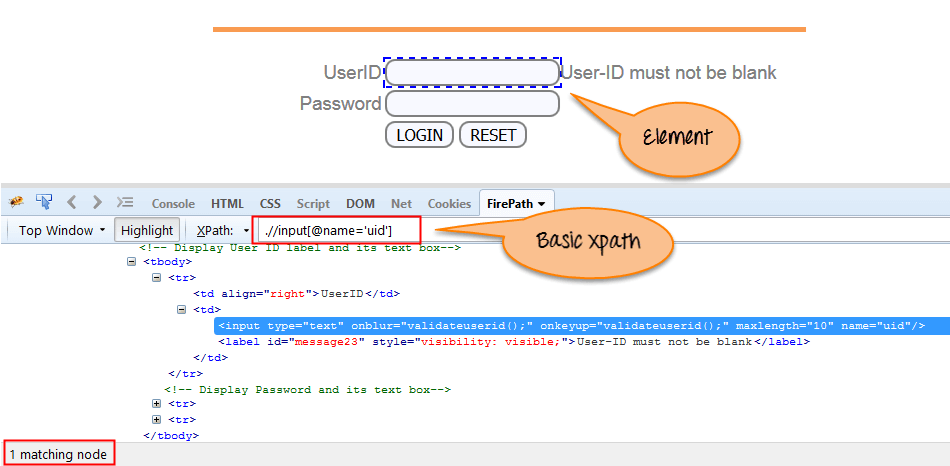
//htmltagname[starts-with(@attributeType,'attributevalue')]

**Using XPath Handling complex & Dynamic elements in Selenium**

**1) Basic XPath:**

XPath expression select nodes or list of nodes on the basis of attributes like **ID , Name, Classname**, etc. from the XML document as illustrated below.

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

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele4.png)

Basic XPath

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']

**2) Contains():**

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

The contain feature has an ability to find the element with partial text as shown in below XPath example.

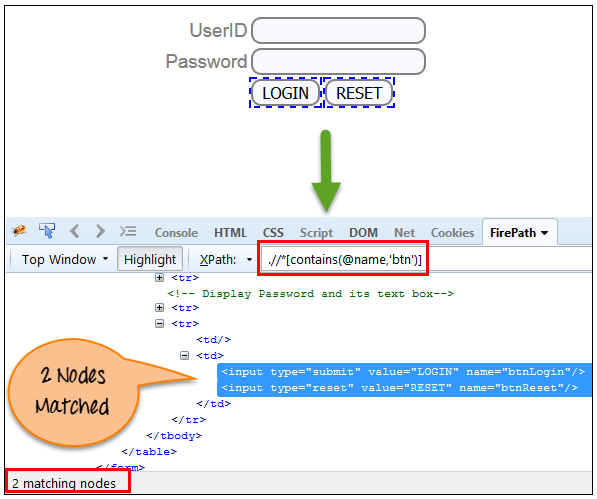
Complete value of 'Type' is 'submit' but using only partial value 'sub'.

Xpath=//\*[contains(@type,'sub')]

Complete value of 'name' is 'btnLogin' but using only partial value 'btn'.

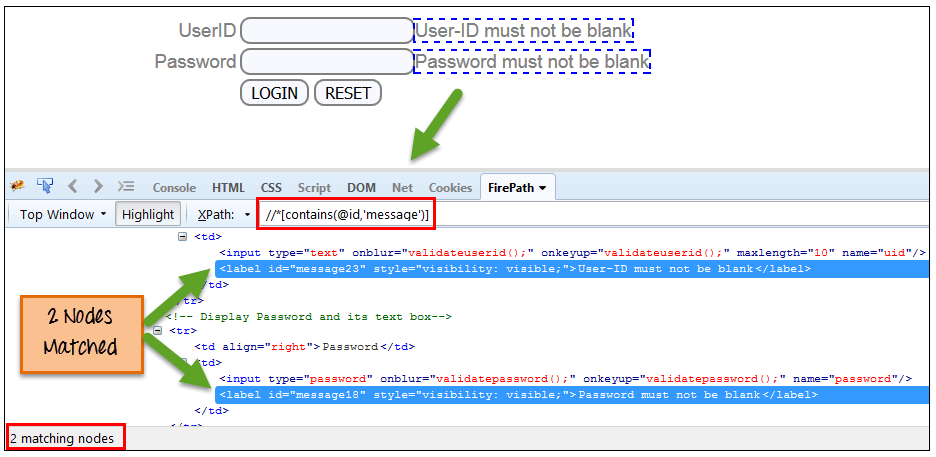
Xpath=//\*[contains(@name,'btn')]

In the above expression, we have taken the 'name' as an attribute and 'btn' as an partial value as shown in the below screenshot. This will find 2 elements (LOGIN & RESET) as their 'name' attribute begins with 'btn'.

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele5.png)

Similarly, in the below expression, we have taken the 'id' as an attribute and 'message' as a partial value. This will find 2 elements ('User-ID must not be blank' & 'Password must not be blank') as its 'name' attribute begins with 'message'.

Xpath=//\*[contains(@id,'message')]

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele6.png)

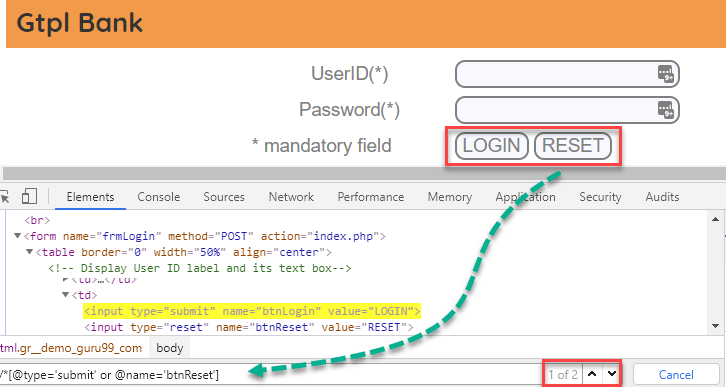
**3) Using OR & AND:**

In OR expression, two conditions are used, whether 1st condition OR 2nd condition should be true. It is also applicable if any one condition is true or maybe both. Means any one condition should be true to find the element.

In the below XPath expression, it identifies the elements whose single or both conditions are true.

Xpath=//\*[@type='submit' or @name='btnReset']

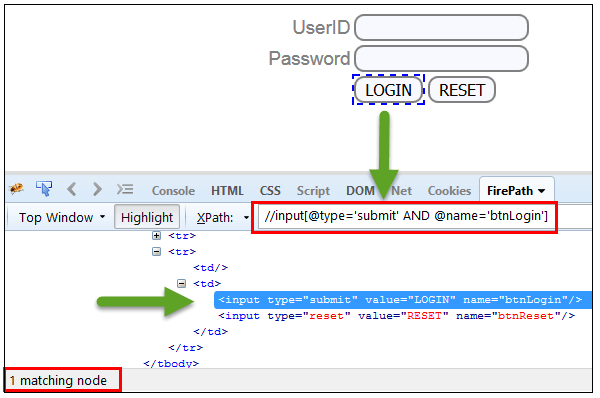
Highlighting both elements as "LOGIN " element having attribute 'type' and "RESET" element having attribute 'name'.

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele8.png)

In AND expression, two conditions are used, both conditions should be true to find the element. It fails to find element if any one condition is false.

Xpath=//input[@type='submit' and @name='btnLogin']

**In below expression, highlighting 'LOGIN' element as it having both attribute 'type' and 'name'.**

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele9.png)

**4) Xpath 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. In this method, the starting text of the attribute is matched to find the element whose attribute value changes dynamically. You can also find elements whose attribute value is static (not changes).

For example -: Suppose the ID of particular element changes dynamically like:

Id=" message12"

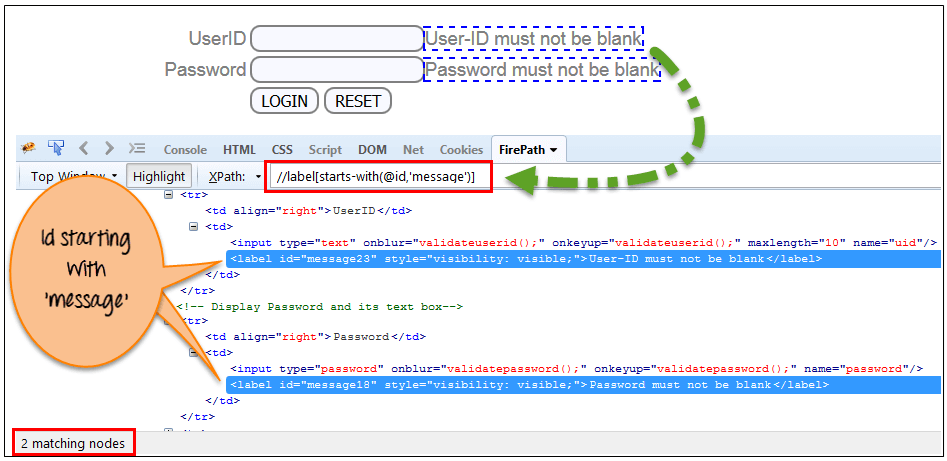
Id=" message345"

Id=" message8769"

and so on.. but the initial text is same. In this case, we use Start-with expression.

In the below expression, there are two elements with an id starting "message"(i.e., 'User-ID must not be blank' & 'Password must not be blank'). In below example, XPath finds those element whose 'ID' starting with 'message'.

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

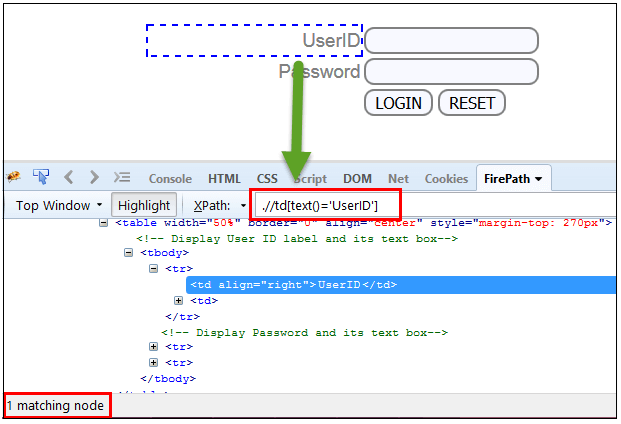
[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele10.png)

**5) XPath Text() Function**

The **XPath text() function** is a built-in function of selenium webdriver which is used to locate elements based on text of a web element. It helps to find the exact text elements and it locates the elements within the set of text nodes. The elements to be located should be in string form.

In this expression, with text function, we find the element with exact text match as shown below. In our case, we find the element with text "UserID".

Xpath=//td[text()='UserID']

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele11.png)

**6) XPath axes methods:**

These XPath axes methods are used to find the complex or dynamic elements. Below we will see some of these methods.

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.

Axes methods are used to find those elements, which dynamically change on refresh or any other operations. There are few axes methods commonly used in [Selenium Webdriver](https://www.guru99.com/introduction-webdriver-comparison-selenium-rc.html) like child, parent, ancestor, sibling, preceding, self, etc.

Diagram

Description automatically generated

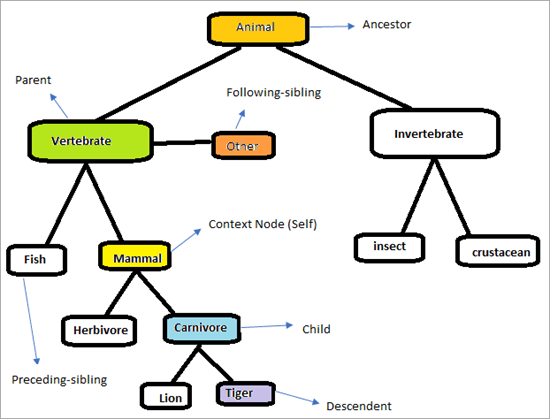
**Different XPath Axes Used In Selenium Testing**

**There are thirteen different axes that are listed below. However, we’re not going to use all of them during Selenium testing.**

1. **ancestor**: These axes indicate all the ancestors relative to the context node, also reaching up to the root node.
2. **ancestor-or-self:**This one indicates the context node and all the ancestors relative to the context node, and includes the root node.
3. **attribute:**This indicates the attributes of the context node. It can be represented with the “@” symbol.
4. **child:**This indicates the children of the context node.
5. **descendent:**This indicates the children, grandchildren, and their children (if any) of the context node. This does NOT indicate the Attribute and Namespace.
6. **descendent-or-self:** This indicates the context node and the children, and grandchildren and their children (if any) of the context node. This does NOT indicate the attribute and namespace.
7. **following:**This indicates all the nodes that appear **after** the context node in the HTML DOM structure. This does NOT indicate descendent, attribute, and namespace.
8. **following-sibling:**This one indicates all the sibling nodes (same parent as the context node) that **appear** after the context node in the HTML DOM structure. This does NOT indicate descendent, attribute, and namespace.
9. **namespace:**This indicates all the namespace nodes of the context node.
10. **parent:**This indicates the parent of the context node.
11. **preceding:**This indicates all the nodes that appear **before** the context node in the HTML DOM structure. This does NOT indicate descendent, attribute, and namespace.
12. **preceding-sibling:**This one indicates all the sibling nodes (same parent as context node) that appear **before** the context node in the HTML DOM structure. This does NOT indicate descendent, attribute, and namespace.
13. **self:**This one indicates the context node.

**Structure Of XPath Axes**

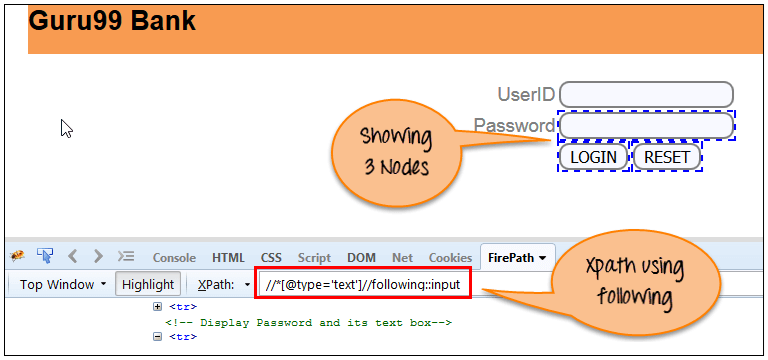
**Consider the below hierarchy for understanding how the XPath Axes work.**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2020/03/Image-16.png)

**a) 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

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele12.png)

There are 3 "input" nodes matching by using "following" axis- password, login and reset button. If you want to focus on any particular element then you can use the below XPath method:

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

You can change the XPath according to the requirement by putting [1],[2]…………and so on.

With the input as '1', the below screen shot finds the particular node that is 'Password' input box element.

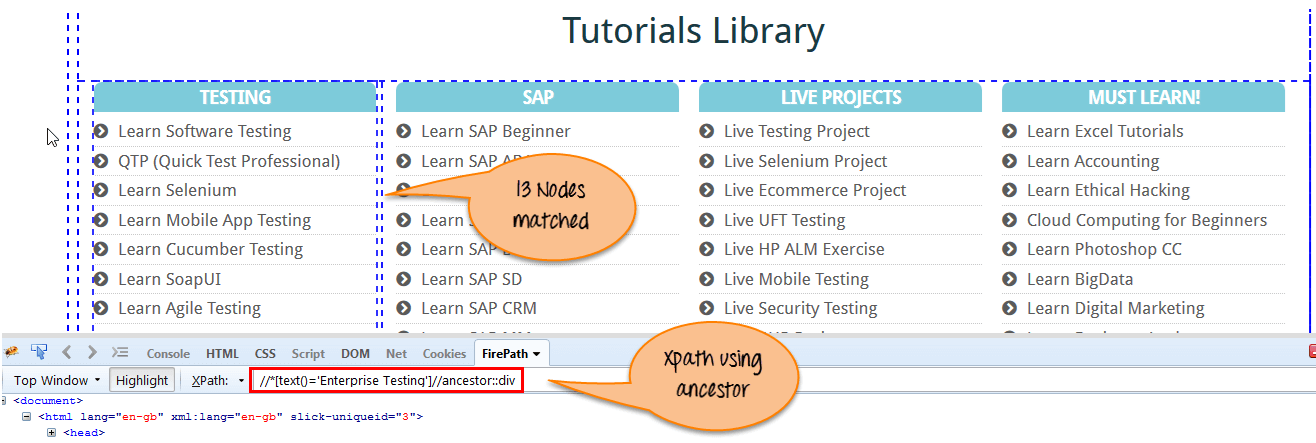
[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele13.png)

**b) Ancestor:**

The ancestor axis selects all ancestors 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

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele14.png)

There are 13 "div" nodes matching by using "ancestor" axis. If you want to focus on any particular element then you can use the below XPath, where you change the number 1, 2 as per your requirement:

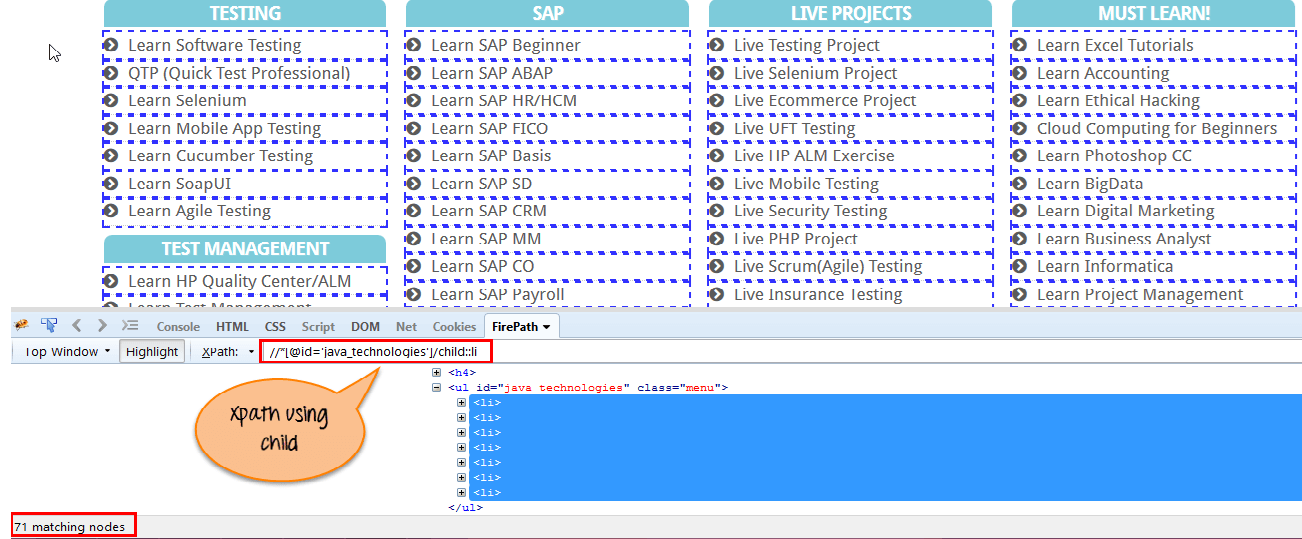
Xpath=//\*[text()='Enterprise Testing']//ancestor::div[1]

You can change the XPath according to the requirement by putting [1], [2]…………and so on.

**c) Child:**

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

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

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele15.png)

There are 71 "li" nodes matching by using "child" axis. If you want to focus on any particular element then you can use the below xpath:

Xpath=//\*[@id='java\_technologies']//child::li[1]

You can change the xpath according to the requirement by putting [1],[2]…………and so on.

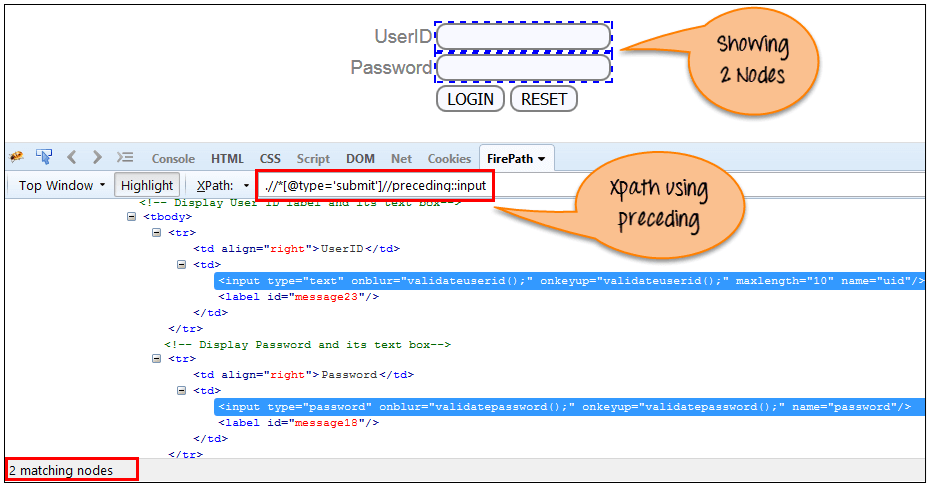
**d) Preceding:**

The preceding axis selects all nodes that come before the current node in the document, except ancestor, attribute nodes, and namespace nodes.

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

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele16.png)

There are 2 "input" nodes matching by using "preceding" axis. If you want to focus on any particular element then you can use the below XPath:

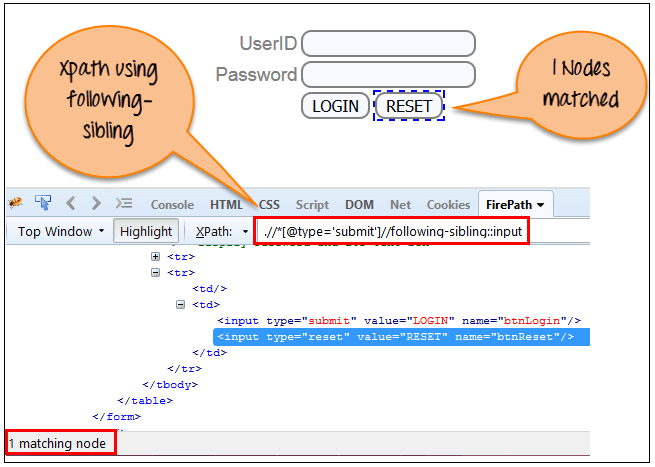
Xpath=//\*[@type='submit']//preceding::input[1]

You can change the xpath according to the requirement by putting [1],[2]…………and so on.

**e) 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

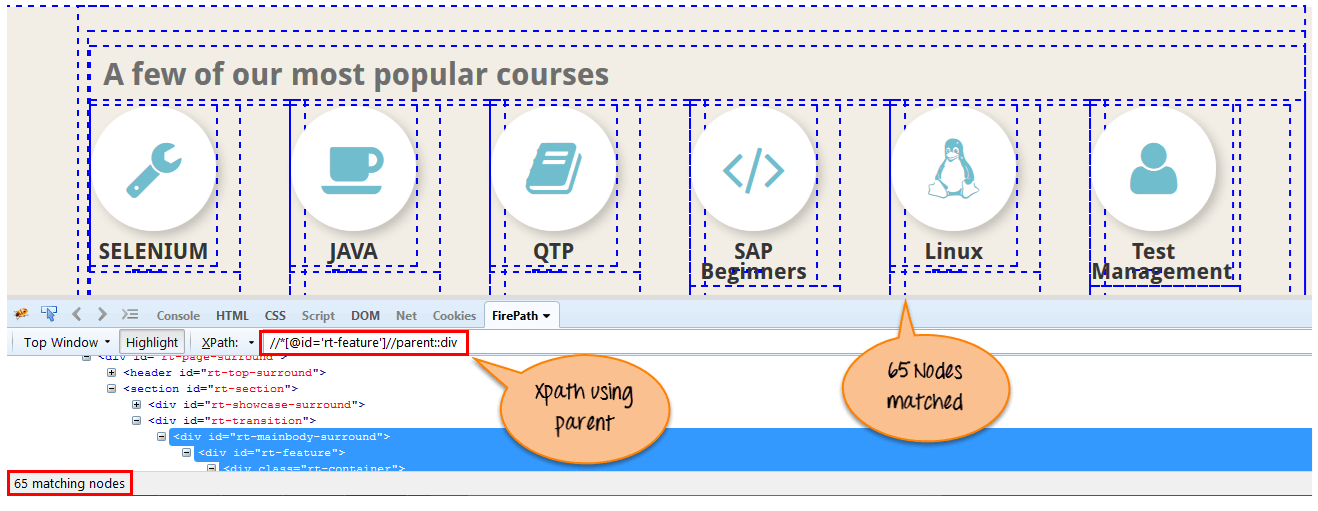
[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele17.png)

One input nodes matching by using "following-sibling" axis.

**f) Parent:**

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

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

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele18.png)

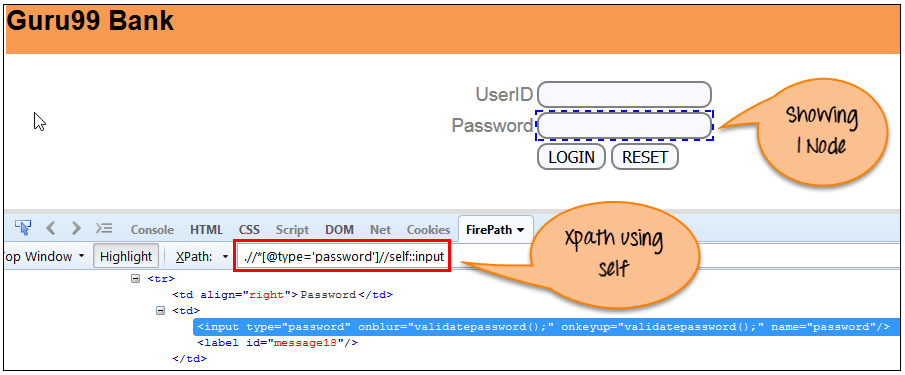
There are 65 "div" nodes matching by using "parent" axis. If you want to focus on any particular element then you can use the below XPath:

Xpath=//\*[@id='rt-feature']//parent::div[1]

You can change the XPath according to the requirement by putting [1],[2]…………and so on.

**g) Self:**

Selects the current node or 'self' means it indicates the node itself as shown in the below screen.

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele19.png)

One node matching by using "self " axis. It always finds only one node as it represents self-element.

Xpath =//\*[@type='password']//self::input

**h) Descendant:**

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

In the below expression, it identifies all the element descendants to current element ( 'Main body surround' frame element) which means down under the node (child node , grandchild node, etc.).

Xpath=//\*[@id='rt-feature']//descendant::a

[](https://www.guru99.com/images/3-2016/032816_0758_XPathinSele20.png)

There are 12 "link" nodes matching by using "descendant" axis. If you want to focus on any particular element then you can use the below XPath:

Xpath=//\*[@id='rt-feature']//descendant::a[1]

You can change the XPath according to the requirement by putting [1],[2]…………and so on.

//\*[@id='java\_technologies']//descendant::li//a[@title='Software Testing']

**Attribute Axis:**

This axis selects the element node on the basis of the attribute identifier (@) of the current node. If the current node is not an element node, this axis is empty. The expressions attribute::type and @type both are equivalent.

For example:

Open the webpage www.pixabay.com, right-click on the search input box, and go to inspect. We can write the XPath of search input box (current node) using the attribute axis.

XPath(Search box): //input[attribute::name = 'q']

**Namespace Axis:**

The namespace axis is one of 13 XPath axes that selects all namespace nodes associated with current node. If the current node is not an element node then this axis will be empty.

Example:

//namespace:element/namespace:child[@namespace:attribute='value']

You'll need to replace "namespace" with the actual namespace prefix used in your XML document. Similarly, replace "element", "child", "attribute", and "value" with the appropriate element, child, attribute, and value names you are targeting.

**Summary:**

XPath is required to find an element on the web page as to do an operation on that particular element.

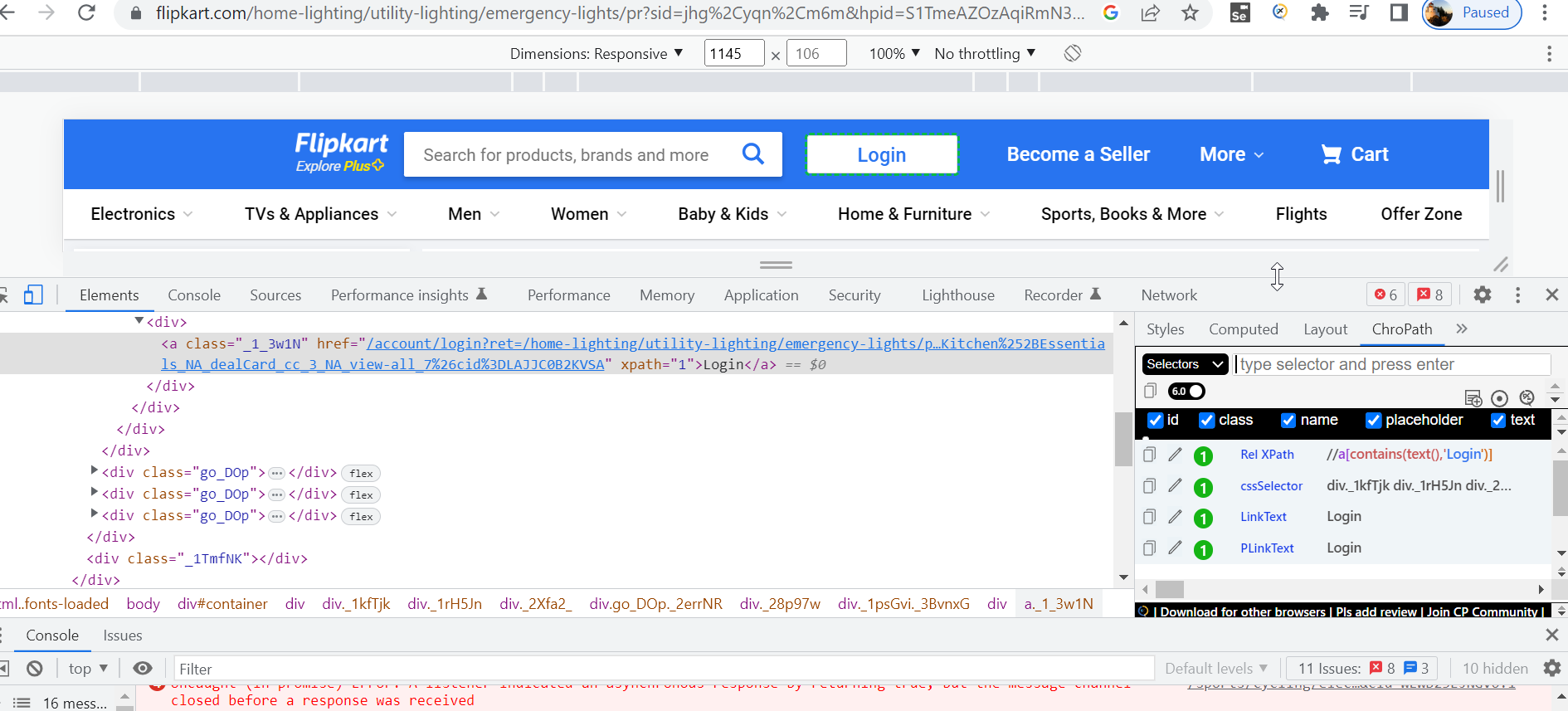
* There are two types of selenium XPath:
  + **Absolute XPath**
  + **Relative XPath**
* XPath Axes are the methods used to find dynamic elements, which otherwise not possible to find by normal XPath method
* XPath expression select nodes or list of nodes on the basis of attributes like ID , Name, Classname, etc. from the XML document .

Chropath Plugin

## ****What is ChroPath?****

ChroPath is a development tool to edit, inspect, and generate XPath and CSS selectors. ChroPath makes it easy to write, edit, extract and evaluate XPath and CSS queries on any webpage and saves at least 40–50% manual effort in automation script writing. ChroPath is the highest rated (4.6+) xpath tool.

✅ Feature of Chropath :  
✅ Single Click Locators Generation  
✅ Verify and Modify Locators  
✅ iframe Support & SVG element support  
✅ Multiple XPath and Automation command  
✅ Dynamic attribute Support  
✅ Save config in local  
✅ChroPath Studio  
✅Smart Maintenance: Verify all XPath in the single-shot!!

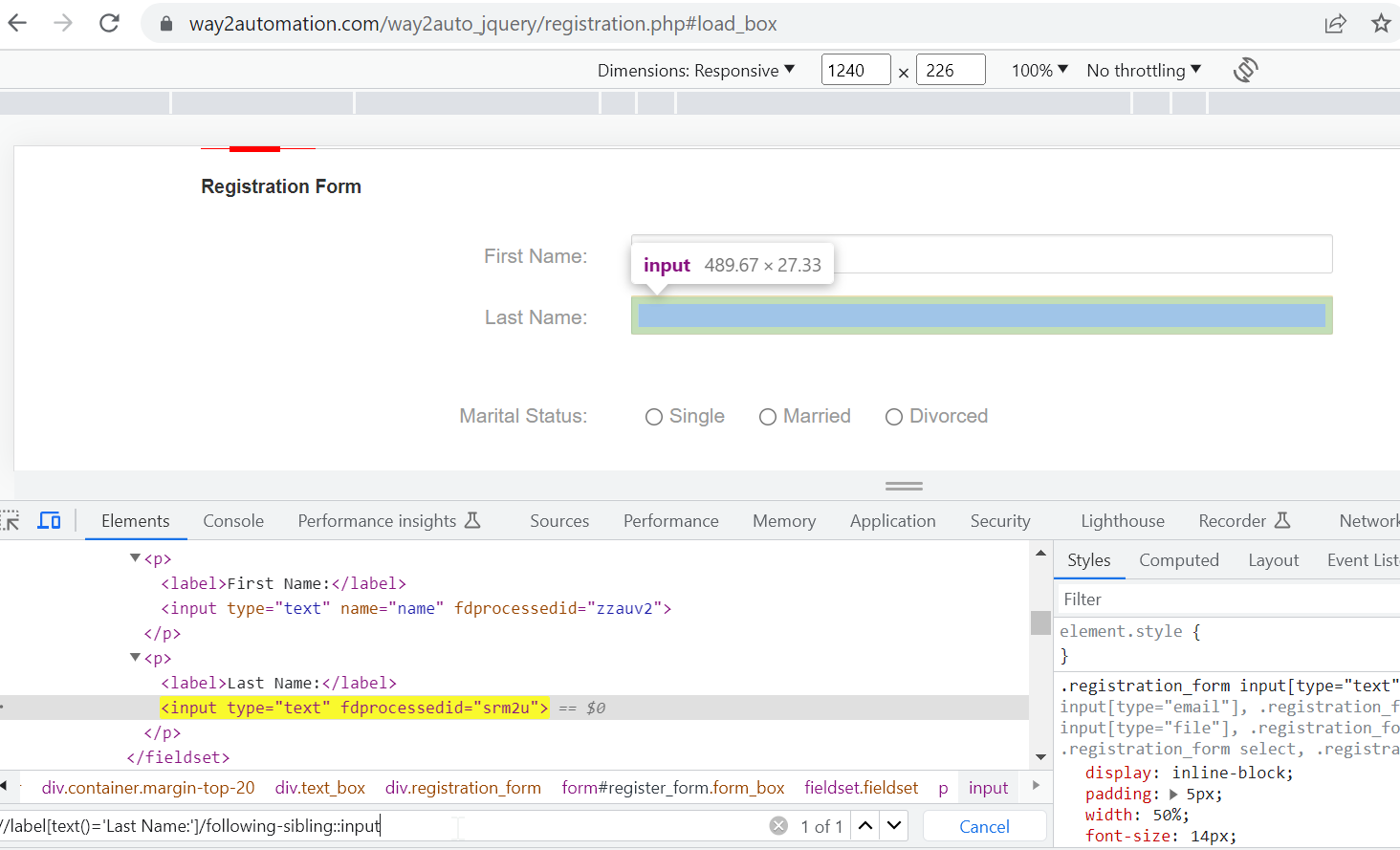


Examples

https://www.way2automation.com/way2auto\_jquery/registration.php#load\_box

A screenshot of a computer

Description automatically generated with medium confidence



References

Guru99

<https://www.scientecheasy.com/2019/08/xpath-axes.html/>

<https://www.toolsqa.com/selenium-webdriver/write-effective-xpaths/>

<https://www.softwaretestinghelp.com/xpath-axes-tutorial/>