**What is XPath**

XPath is defined as **XML path**. **It is a syntax or language for 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. The basic format of XPath is explained below with screen shot.

Xpath=//tagname[@attribute='value']

* **// :** Select current node.
* **Tagname:**Tagname of the particular node.
* **@:** Select attribute.
* **Attribute:** Attribute name of the node.
* **Value:** Value of the attribute.

## 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.

The key characteristic of XPath is that it begins with the single forward slash(/) ,which means you can select the element from the root node.

Below is the example of an absolute xpath expression of the element shown in the below screen.

**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 at the webpage.

You can start from the middle of the HTML DOM structure and no need to write long xpath.

## Using XPath Handling complex & Dynamic elements in Selenium

**1) Basic XPath:**

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

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

Ex: Xpath=//\*[contains(text(),'here')]

Xpath=//\*[contains(@href,'guru99.com')]

**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.

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

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

**4) Start-with function:**Start-with function finds the element whose attribute value changes on refresh or any operation on the webpage.In this expression, match the starting text of the attribute is used to find the element whose attribute changes dynamically. You can also find the element whose attribute value is static (not changes).

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

Id=" message12"

Id=" message345"

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

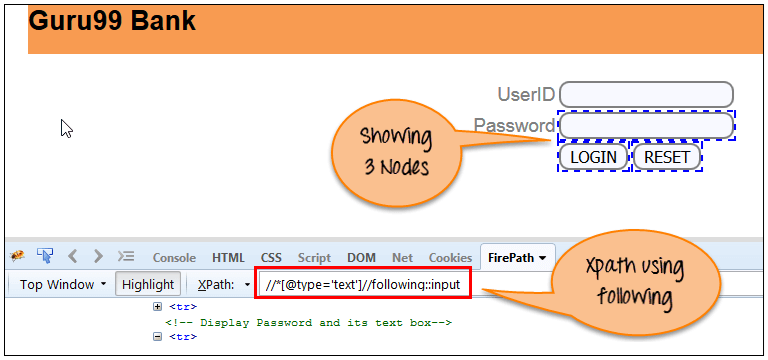
**5) Text():**In this expression, with text function, we find the element with exact text match as shown below. In our case, we find the element with text "UserID"

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

**6) XPath axes methods**: These XPath axes methods are used to find the complex or dynamic elements

**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]

**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

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

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

**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

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:

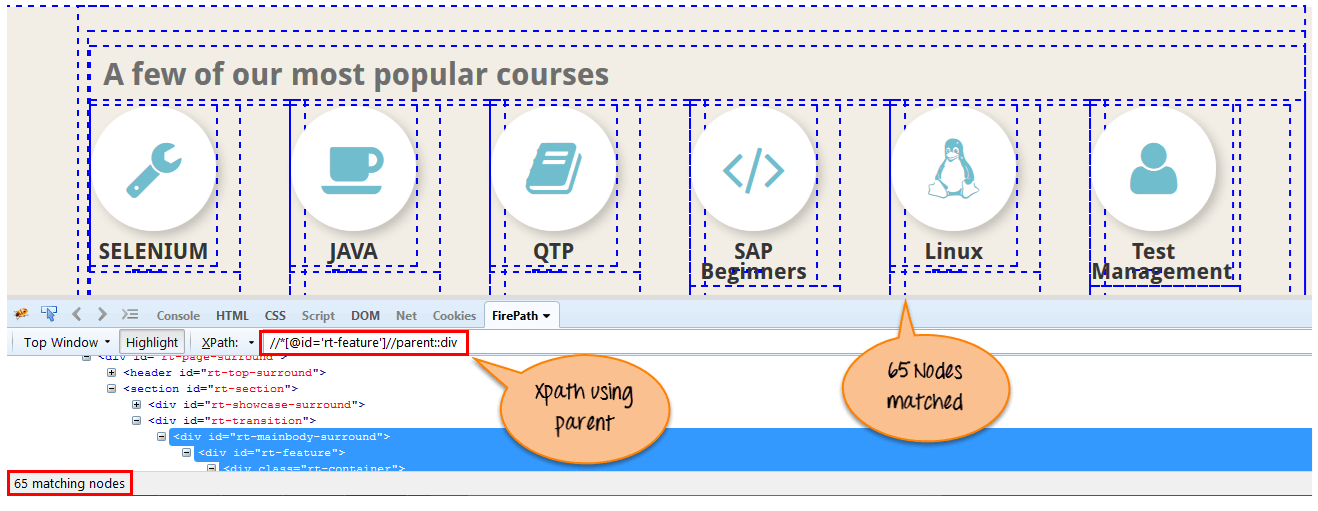
Xpath=//\*[@type='submit']//preceding::input[1]

**Following-sibling:** Select the following siblings of the context node. Siblings are at the same level of the current node

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

**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]