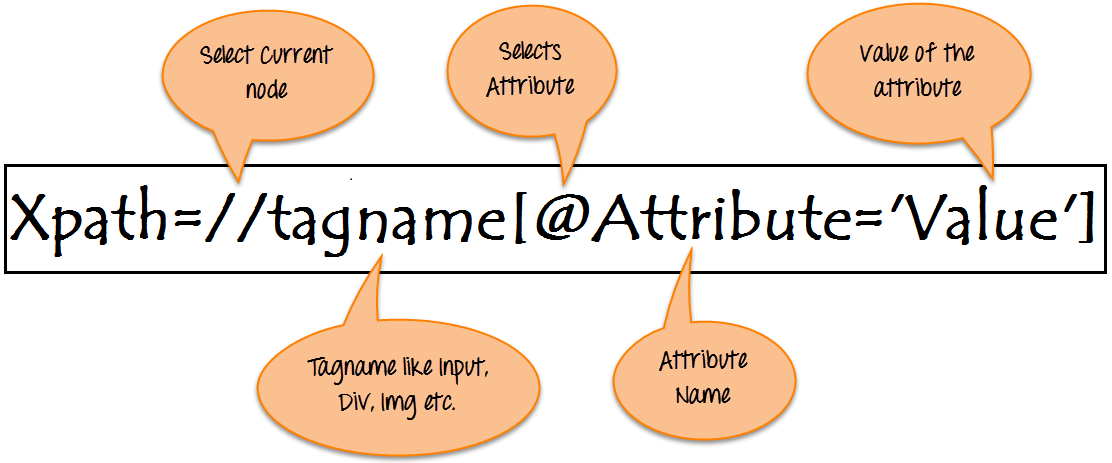
XPATH



## 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.

**Absolute xpath:**

html/body/div[1]/section/div[1]/div/div/div/div[1]/div/div/div/div/div[3]/div[1]/div/h4[1]/b

**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 starts from the middle of the HTML DOM structure and no need to write long xpath.

Below is the example of a relative XPath expression of the same element shown in the below screen. This is the common format used to find element through a relative XPath.

Relative xpath: //\*[@class='featured-box']//\*[text()='Testing']

# Finding By Attribute

*//TagName[@AttrId=’AttrVal’]*

//Img[@AttrId=’AttrVal’]

//Input[@AttrId=’AttrVal][@AttrId=’AttrVal]

//\*[@AttrId=’AttrVal’ and @AttrId=’AttrVal’]

//\*[@AttrId=’AttrVal’ or @AttrId=’AttrVal’]

# Finding By Contains

*//tagName[contains(@id,’message’)]*

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

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

# Finding By StartsWith

*//tagName[starts-with(@id,’message’)]*

Xpath = //\*[start-with(@id,’sometext’)]

Xpath = //Input[start-with(@id,sometext’)]

# Finding By text() Finds exact text

//tagname[text()=’message’]

Xpath = //TD[text()=’SomeText’]

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

1. **Following**
2. **Ancestor**
3. **Preceding**
4. **Following-sibling**
5. **Parent**
6. **Self**
7. **Descendant**

**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

**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

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

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

**d) 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

**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

**f) Parent:** Selects the parent of the current node as shown in the below screen.

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

## 