Selecting Nodes Using XPATH

XPath uses path expressions to select nodes in an XML document.

The node is selected by following a path or steps.

The most useful path expressions are listed below:

Expression	Description
/	Selects from the root node (HTML)
//	Selects nodes in the document from the current node that match the selection no matter where they are
@	Selects attributes/properties

XPath Axes

An axis defines a node-set relative to the current node.

AxisName	Result
ancestor	Selects all ancestors (parent, grandparent, etc.) of the current node
ancestor-or-self	Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself
Attribute	Selects all attributes of the current node
Child	Selects all children of the current node
Descendant	Selects all descendants (children, grandchildren, etc.) of the current node
descendant-or-self	Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself
Following	Selects everything in the document after the closing tag of the current node
following-sibling	Selects all siblings after the current node

namespace	Selects all namespace nodes of the current node
Parent	Selects the parent of the current node
Preceding	Selects all nodes that appear before the current node in the document, except ancestors,attribute nodes and namespace nodes
preceding-sibling	Selects all siblings before the current node
Self	Selects the current node

Let's discuss examples for below XPath functions –

- Relative xpath examples for HRM
- XPath Using AND (Conjunction)
- XPath Using OR (disjunction)
- XPath Using Text()
- XPath Using Contains
- XPath Using descendant
- XPath Using Sibling
- XPath Using Ancestor

Let's study them in detail -

XPath Using AND (Conjunction)

```
<input name="txtUserName"  type="text"  class="loginText"  tabindex="1">
```

```
//input[@name='txtUserName']
//input[@name='txtUserName'][@type='text']
//input[@name='txtUserName'][@class =' loginText']

//input[@name='txtUserName' and @class='loginText']
//input[@name='txtUserName' and @type='text']
//input[@type='text' and @class='loginText']
```

XPath Using OR (disjunction)

```
//input[@name='txtUserName' or @class='loginText']
//input[@name='txtUserName' or  @type='text']
//input[@type='text' or @class='loginText']
```

XPath Using Text()

```
<a href="./index.php?ACT=logout" target="rightMenu">Logout</a>
```

```
//a[text()='Logout']
```

Contains: By using 'contains' function in XPath, we can extract all the elements which matches a particular text value.

Ex. Here we are searching an anchor .contains text as Logout.

```
<a href="./index.php?ACT=logout" target="rightMenu">Logout</a>

//a[contains(text(),'Logout')]
```

descendant	Selects all descendants (children, grandchildren, etc.) of the current node

Relative xpath for user name

//table[@id="Table_01"]/tbody/tr[1]/td[2]/table/tbody/tr[2]/td[2]/input

Relative xpath for user name using descendant

//table[@id="Table_01"]/descendant::input

//table[@id="Table_01"]/descendant::input[@type='text']

Way 3: (optional)

//table[@id='Table_01']//input

Sibling: Using sibling keyword, we can fetch a web element on the which is related to some other element.

--EX1--For -Following-sibling-----Get c++ faculty name ?

WAY1: //table[@id='idCourse']//td[contains(text(),'C++')]/following-sibling::td[1]

WAY2: //table[@id='idCourse']/descendant::td[contains(text(),'C++')]/following-sibling::td[1]

--EX1--For -Following-sibling-----Get c++ start date ?

//table[@id='idCourse']/descendant::td[contains(text(),'C++')]/following-sibling::td[2]

--Ex3--For preceding sibling ---Get course name of John?

//table[@id='idCourse']/descendant::td[contains(text(),'John')]/preceding-sibling::td[1]

Ancestor: To find an element on the basis of the parent element we can use ancestor attribute of XPath.

//td[contains(text(),'C++')]/ancestor::table

----------------------XPATH For Username- Line 4---

Absolute XPath:

/html/body/form/input

Relative xpath:

//form[@id=”loginForm”]/input[1]

//form[1]/input[1]

Same level XPath:

//input[@name=&quot;username&quot;]

//input[@type=&quot;text&quot;]

//input[@name=&quot;username&quot;][@type=&quot;text&quot;]