

$x ("//input [@id ='searchInput']")

$x -> is the function in xpath.

// -> represents anywhere in html DOM

/ -> direct child of html DOM.

input -> tag Name

$x ("//input [@id ='searchInput'] [@name ='search']")

starts-with (property, propertyValue)

$x ("//input [starts-with(@id, 'search')]")

contains (property, propertyValue)

$x ("//input [contains(@id, 'archInp')]")

substring ( property, start, end) : end = (end + start -1)

$x ("//input [substring (@id, 2) = 'earchInput' ]")

$x ("//input [substring (@id, 2, 5) = 'earch' ]")

In above: end = 2+5-1 = 6 – so search string needs to give from 2nd character to till 6th character.

\*\*Index always starts with 1 in Xpath.

***//driver.find\_elements\_by\_xpath("//\*[contains(text(), 'My Button')]")***

$x ("//title [text() = 'browser title' ]") – text() is to access inner text.

Or

$x ("//title [ . = 'browser title' ]")

$x ("//title [normalize-space(text()) = 'browser title']") – remove all spaces or convert spaces b/w two character in single space.

Parents and sibling concepts in xpath:

<Parent>

<child1>

<sub-child1>

<sub-sub-child1>

<sub-sub-child2>

< sub-child2>

< sub-child3>

< sub-child\_same\_tag>

< sub-child\_same\_tag>

<child2>

<child\_same\_tag>

<child\_same\_tag>

<child\_same\_tag>

<child\_same\_tag>

</Parent>

Sibling relationship (if object sibling can be identified, how to identify object itself with the help of his sibling):

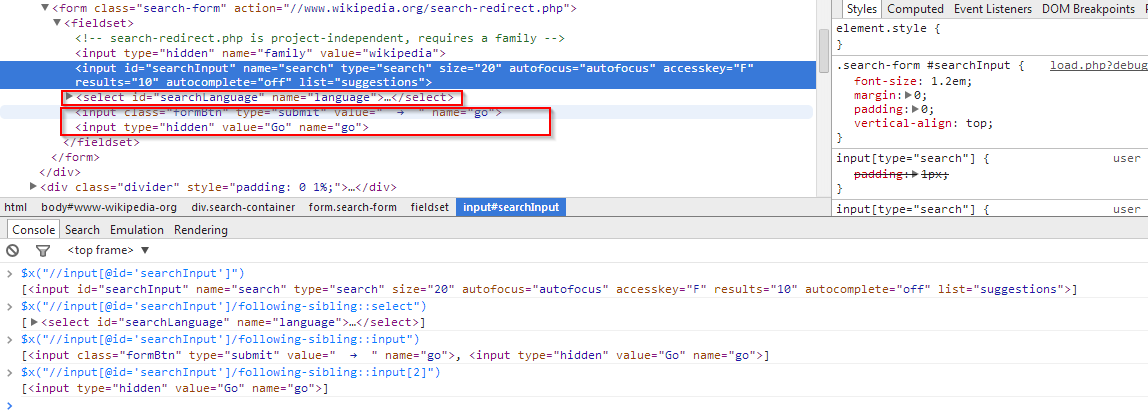
$x ("//child1-tag [@id ='child1-id']/following-sibling :: child2-tag")

// will move control over child2 (sibling of child1)

$x ("//child1-tag [@id ='child1-id']/following-sibling :: child\_same\_tag")

// will give all the childs with same tag (all sibling of child1 after child1)

$x ("//Parent-tag [@id ='parent-id']/following-sibling :: child\_same\_tag[2]")

// will give 2nd element child element with same tag (2nd sibling of child1 after child1)

Here in above example tag ‘input’ can be identified using its ‘id’, so we can identify tag ‘select’ (sibling of tag ‘input’), also we can identify tag ‘input’ (sibling of tag ‘input’).

Parent child relationship (if we know can identify the child then how we can identify parent):

$x ("//child [@id ='child-id']/ parent :: parent-tag or \* or /../..")

// will move control over parent

parent-tag –> will move control to any parent

\* -> will give the immediate parent

/..-> for immediate parent; /../.. ->for parent of parent

http://www.joecolantonio.com/2011/06/07/qtp-how-to-use-the-fireevent-method-when-a-click-does-not-work/