**XPath session for Selenium**

XPath is designed to allow the navigation of XML documents,with the purpose of selecting individual elements, attributes, or some other part of an XML document for specific processing.

What is XML?  
The Extensible Markup Language (XML) is the context in which the XML Path Language, XPath, exists.

XML provides a standard syntax for the markup of data and documents.

XML documents contain one or more elements. If an element contains content,whether other elements or text, then it must have a start tag and an end tag. The text contained between the start tag and the end tag is the element’s content.

<Element> *//Start tag*

Element content goes here.*//Element Content*

</Element>*//End Tag*

An element may have one or more attributes, which will provide additional information  
about the element type or its content.  
  
Below is the sample XML:

<?xml version='1.0'?>

<**Catalog**>

<**Book**>

<**Title**>XML Session</**Title**>

<**Author**>Selenium Easy</**Author**>

</**Book**>

</**Catalog**>

It can also be written as:

<?xml version='1.0'?>

<**Catalog**>

<**Book** Title="XML Session" Author="Selenium Easy">

</**Book**>

</**Catalog**>

XPath can be viewed as a way to navigate round XML documents. Thus XPath has similarities to a set of street directions.

When you need to search for a address, you should know what is your starting point to reach your destination.

In XPath the starting point is called the context node.

**Absolute XPath**  
Absolute XPath starts with the root node or a forward slash (/).  
The advantage of using absolute is, it identifies the element very fast.  
Disadvantage here is, if any thing goes wrong or some other tag added in between, then this path will no longer works.

**Example:**  
If the Path we defined as  
1. html/head/body/table/tbody/tr/th

If there is a tag that has added between body and table as below  
2. html/head/body/form/table/tbody/tr/th

The first path will not work as 'form' tag added in between

**Relative Xpath**  
A relative xpath is one where the path starts from the node of your choise - it doesn't need to start from the root node.

It starts with Double forward slash(//)

**Syntax:**  
//table/tbody/tr/th

Advantage of using relative xpath is, you don't need to mention the long xpath, you can start from the middle or in between.

Disadvantage here is, it will take more time in identifying the element as we specify the partial path not (exact path).

If there are multiple elements for the same path, it will select the first element that is identified

**XPath Axes :**  
XPath has a total of 13 different axes, which we will look at in this section. An XPath axis tells the XPath processor which “direction” to head in as it navigates around the hierarchical tree of nodes.

|  |  |
| --- | --- |
| **Xpath axis Name** | **Description** |
| self | Which contains only the context node |
| ancestor | contains the ancestors of the context node, that is, the parent of the context node, its parent, etc., if it has one. |
| ancestor-or-self | contains the context node and its ancestors |
| attribute | contains all the attribute nodes, if any, of the context node |
| child | contains the children of the context node |
| descendant | contains the children of the context node, the children of those children, etc. |
| descendant-or-self | contains the context node and its descendants |
| following | contains all nodes which occur after the context node, in document order |
| following-sibling | Selects all siblings after the current node |
| namespace | contains all the namespace nodes, if any, of the context node |
| parent | Contains the parent of the context node if it has one |
| preceding | contains all nodes which occur before the context node, in document order |
| preceding-sibling | contains the preceding siblings of the context node |

The below are the Axes that are very useful  
**1. Child Axes**  
The child axis defines the children of the context node.  
Child::\*  
**Syntax:**

*//child::table*

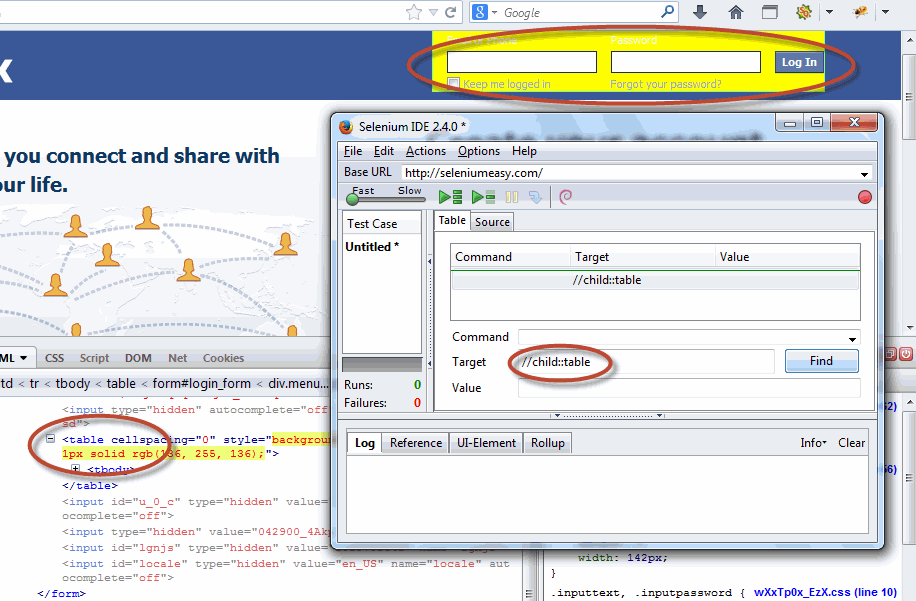
The first location step selects the child element node of the root node, which represents the element root

element in the source document.

The child axis is the default axis, so it need not be explicitly expressed in the abbreviated.

It can be simply re-written as:  
//table/tbody

//child::\*/child::td[position()>1]  
The position ( ) function, evaluates the context position of the context node within the context size. The position ( ) function is applied to the selected nodes in document order. It will select the second td in a table

It will select all the nodes that are Child nodes of table.  
Please find the below screen shot for example.  


**2. Parent Axes**  
The parent axis contains only a maximum of one node. The parent node may be either the root node or an element node.  
The root node has no parent; therefore, when the context node is the root node, the parent axis is empty. For all other element nodes the parent axis contains one node.

Syntax:

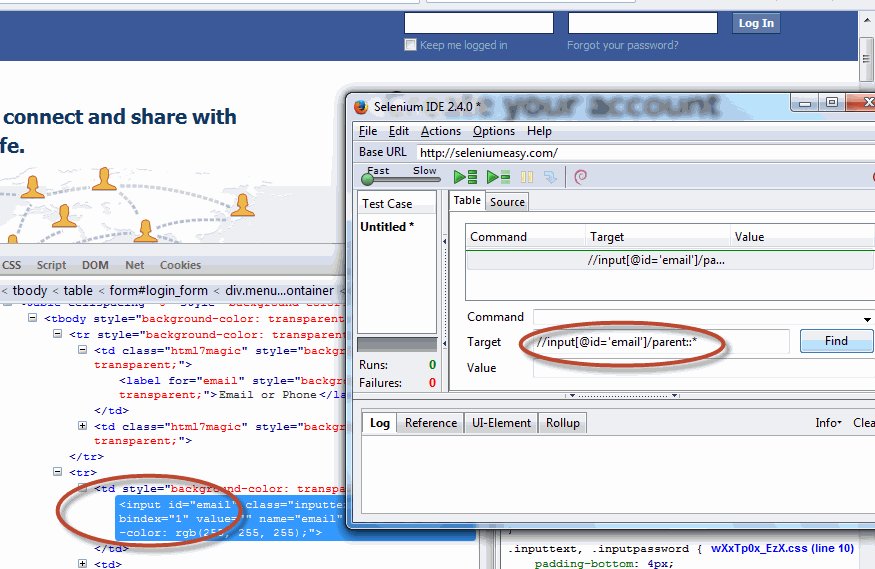
parent::node()

The below example will selects the parent node **of** the input tag **of** Id='email'.

Ex: //input[@id='email']/parent::\*

the above can also be re-written as

//input[@id='email']/..

Below is the image that shows you to identify using above example.  


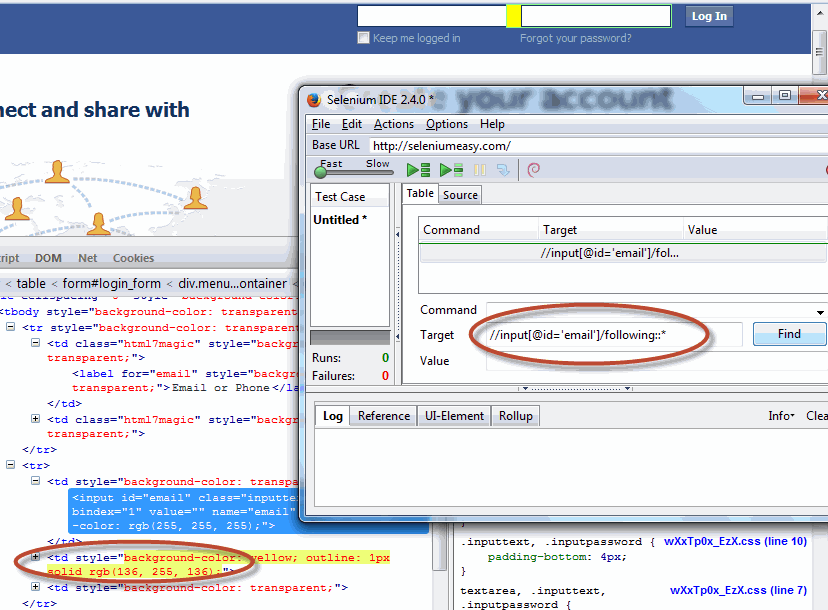
**3. Following Axes**  
“Following axis contains all nodes in the same document as the context node that are after the context node in document order.

**Syntax:**

The below syntax selects the immediate node following the specified node input[@id='email']

//input[@id='email']/following::\*

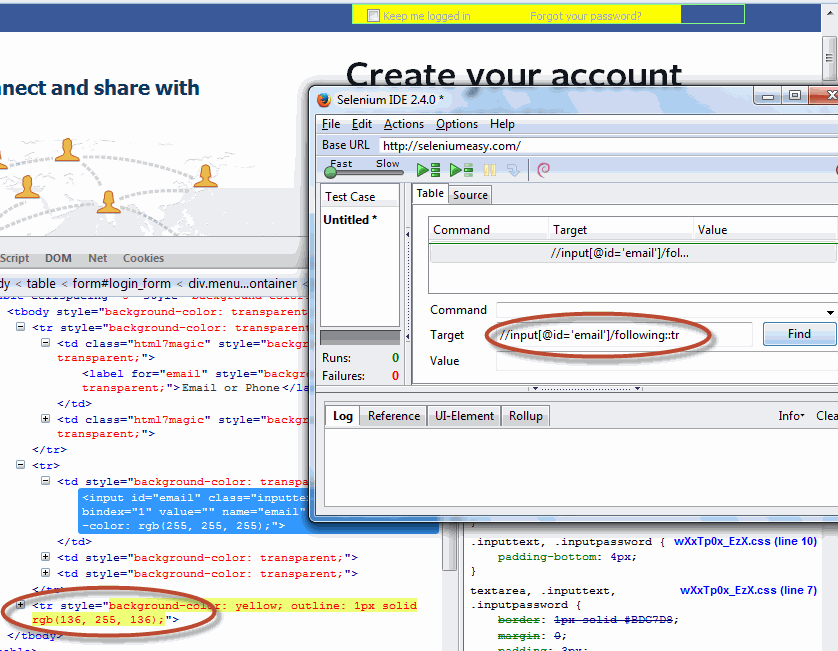
Below is the image that shows you to identify using above example.  
It will identify the immediate node which start after the current node.



The below syntax selects the immediate node of tag 'tr' with the specified node input[@id='email']

//input[@id='email']/following::**tr**

Below is the image that shows you to identify using above example.  
It will identify the immediate node which start after the current node.

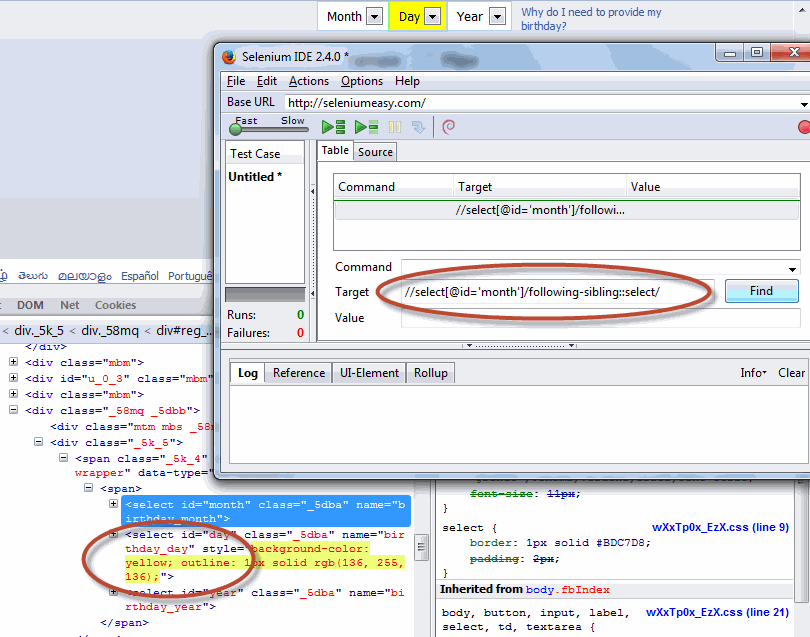


**4. Following Sibling Axes**

The following-sibling axis selects those nodes that are siblings of the context node (that  
is, the context node and its sibling nodes share a parent node) and which occur later in  
document order than the context node.

**Syntax:**  
//select[@id='month']/following-sibling::\*  
//select[@id='month']/following-sibling::select/

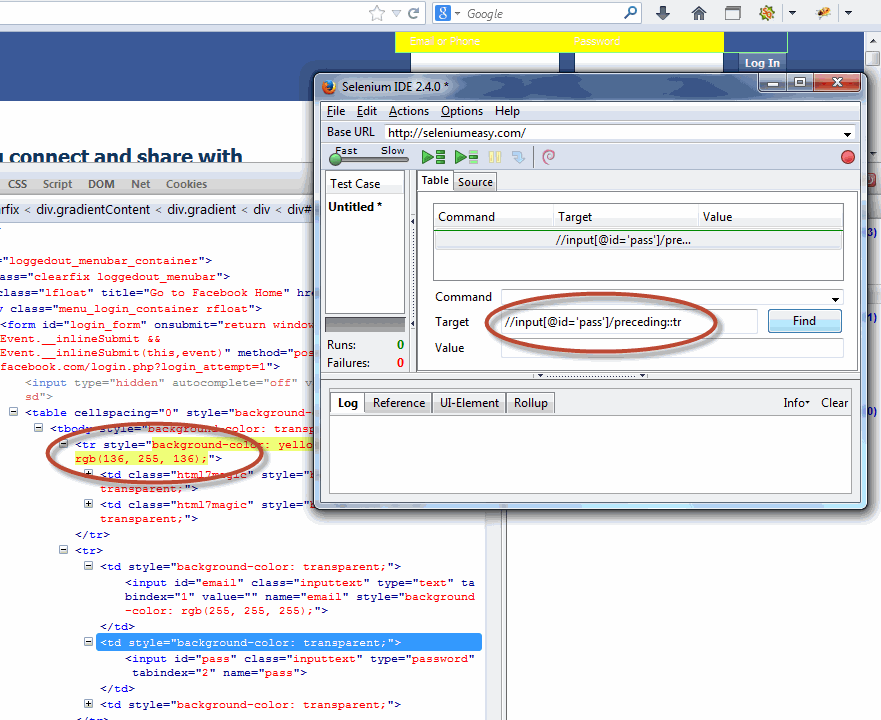
Please check the below image for the above syntax executed



**5. Preceding Axes**  
The preceding axis contains all nodes in the same document as the context node that are before the context node in document order.

**Syntax:**  
//input[@id='pass']/preceding::tr

Below screen shot shows how the preceding axes selects nodes that appear before the current node in the document, except ancestors, attribute nodes and namespace .



**6. Preceding Sibling Axes**

The preceding-sibling axis selects those nodes which are siblings of the context node (that is, the context node and its sibling nodes share a parent node) and which occur earlier in document order than the context node.

**Syntax:**

//**select**[@id='day']/preceding-sibling::**select**/

//select[@id='day']/preceding-sibling::\*

The below image shows how the preceding sibling axes selects siblings before the current node

