

XPath en Selenium: Cómo encontrar y escribir texto, contiene, OR, Y

Por Krishna Rungta Actualizado julio 16, 2022

En la automatización de Selenium, si los elementos no son encontrados por los localizadores generales como id, clase, nombre, etc., entonces XPath se utiliza para encontrar un elemento en la página web.

En este tutorial, aprenderemos sobre el Xpath y diferentes expresiones XPath para encontrar los elementos complejos o dinámicos, cuyos atributos cambian dinámicamente al actualizar o cualquier operación.

En este tutorial de XPath, aprenderá-

- ¿Qué es XPath?
- Tipos de ruta X
 - XPath absoluto
 - XPath relativo
- XPath básico
- Contiene()
- Uso de OR & AND
- Función Starts-with
- Texto()
- Métodos de ejes XPath
 - Siguiente
 - Antepasado
 - Niño
 - Pre-antepasado

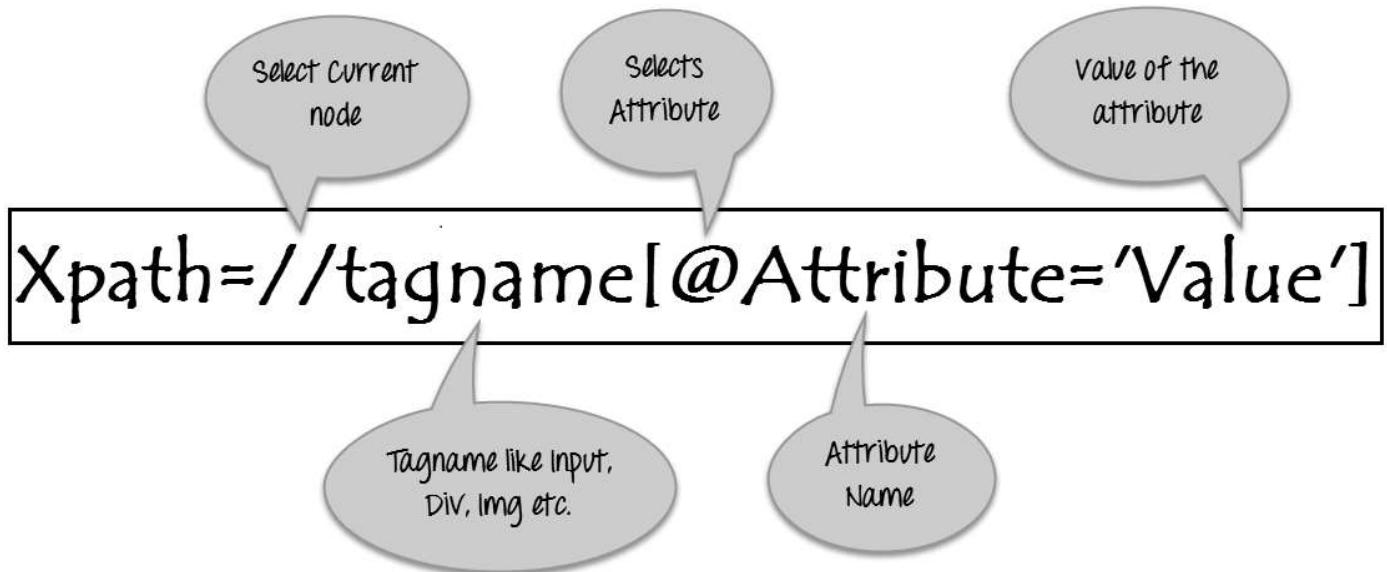


- Padre
- Propio
- Descendiente

¿Qué es XPath en Selenium?

XPath en Selenium es una ruta XML utilizada para la navegación a través de la estructura HTML de la página. Es una sintaxis o lenguaje para encontrar cualquier elemento en una página web utilizando la expresión de ruta XML. XPath se puede utilizar para documentos HTML y XML para encontrar la ubicación de cualquier elemento en una página web utilizando la estructura DOM HTML.

El formato básico de XPath en selenium se explica a continuación con captura de pantalla.



Formato básico de XPath



EXPLORE MORE

Learn Java Programming
with Beginners Tutorial

08:32

Linux Tutorial for
Beginners: Introduction



Sintaxis para XPath selenio:

XPath contiene la ruta del elemento situado en la página web. La sintaxis estándar de XPath para crear XPath es.

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

- **//**: Seleccione el nodo actual.
- **Nombre de la etiqueta**: Tagname del nodo en particular.
- **@**: Seleccione atributo.
- **Atributo**: Nombre del atributo del nodo.
- **Valor**: Valor del atributo.

Para encontrar el elemento en las páginas web con precisión hay diferentes tipos de localizadores:

Localizadores	Encuentra diferentes elementos en la página web
XPath	
IDENTIFICACIÓN	Para buscar el elemento por IDENTIFICADOR del elemento
Nombre de clase	Para buscar el elemento por Classname del elemento
Nombre	Para buscar el elemento por nombre del elemento
Texto del enlace	Para buscar el elemento por texto del enlace
XPath	XPath necesario para encontrar el elemento dinámico y recorrer entre varios elementos de la página web
Ruta CSS	La ruta CSS también localiza elementos que no tienen nombre, clase o ID.

Tíos de ruta X



Hay dos tipos de XPath:

1) XPath absoluto

2) XPath relativo

XPath absoluto:

Es la forma directa de encontrar el elemento, pero la desventaja del XPath absoluto es que si se realizan cambios en la ruta del elemento, entonces ese XPath falla.

La característica clave de XPath es que comienza con la barra diagonal única(/), lo que significa que puede seleccionar el elemento del nodo raíz.

A continuación se muestra el ejemplo de una expresión Xpath absoluta del elemento que se muestra en la pantalla de abajo.

NOTA: Puede practicar el siguiente ejercicio de XPath en este

<http://demo.guru99.com/test/selenium-xpath.html>

Identify XPath on FireFox



Haga clic aquí si el video no es accesible

XPath absoluto:



/html/body/div[2]/div[1]/div/h4[1]/b/html[1]/body[1]/div[2]/div[1]/div[1]/h4[1]/b[1]

The screenshot shows the FirePath extension in the Firefox developer toolbar. A speech bubble labeled "Element" points to the "Testing" link in the left sidebar. Another speech bubble labeled "Absolute Path" points to the expanded DOM tree on the right. The XPath expression in the toolbar is:

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

The DOM tree highlights the element **Testing** under the **h4** heading. The message "1 matching node" is displayed at the bottom.

Xpath absoluto

Xpath relativo:

Relativo Xpath comienza desde el centro de la estructura DOM HTML. Comienza con doble barra diagonal (//). Puede buscar elementos en cualquier lugar de la página web, lo que significa que no es necesario escribir una xpath larga y puede comenzar desde el medio de la estructura DOM de HTML. Siempre se prefiere relative Xpath, ya que no es una ruta completa desde el elemento raíz.

A continuación se muestra el ejemplo de una expresión XPath relativa del mismo elemento que se muestra en la pantalla siguiente. Este es el formato común utilizado para encontrar el elemento por XPath.



Xpath in Chrome



Haga clic aquí si el video no es accesible

Relative XPath: //div[@class='featured-box columnsizel']//h4[1]//b[1]

The screenshot shows the FirePath extension integrated into the Chrome developer tools. A speech bubble labeled "Element" points to the "TESTING" section of the page. Another speech bubble labeled "Relative Path" points to the XPath input field which contains the expression `//*[text()='Testing']`. The DOM tree on the right shows the path taken by the XPath query, starting from the root node and navigating down to the `Testing` element.

```

<div class="row featured-boxes">
  <div class="col-md-3">
    <div class="featured-box" style="height: 700px;">
      <h4>
        <b>Testing</b>
      </h4>
    </div>
  </div>
  <div class="col-md-9">
    <ul id="java_technologies" class="menu">
      <p style="line-height: 15px;"><b>Java</b></p>
      <li><a href="#">Java</a></li>
      <li><a href="#">Java EE</a></li>
      <li><a href="#">Java ME</a></li>
      <li><a href="#">Java SE</a></li>
    </ul>
  </div>
</div>

```

1 matching node

XPath relativo



Los ejes XPath buscan diferentes nodos en el documento XML desde el nodo de contexto actual. Los ejes XPath son los métodos utilizados para encontrar elementos dinámicos, lo que de otro modo no sería posible por el método XPath normal que no tiene ID, Classname, Name, etc.

Los métodos de ejes se utilizan para encontrar esos elementos, que cambian dinámicamente en la actualización o en cualquier otra operación. Hay pocos métodos de ejes comúnmente utilizados en Selenium Webdriver como hijo, parent, antepasado, hermano, precedente, yo, etc.

Cómo escribir XPath dinámico en Selenium WebDriver

1) XPath básico:

La expresión XPath selecciona nodos o lista de nodos sobre la base de atributos como **ID**, **Name**, **Classname**, etc. del documento XML como se ilustra a continuación.

Xpath=//input[@name='uid']

Aquí hay un enlace para acceder a la página <http://demo.guru99.com/test/selenium-xpath.html>

The screenshot illustrates the use of the FirePath extension in a Firefox browser to inspect an HTML form. The 'UserID' input field is highlighted with a dashed border. A tooltip 'Element' points to the input field. A larger callout bubble labeled 'Basic xpath' points to the FirePath toolbar where the XPath expression `//input[@name='uid']` is entered. The FirePath interface shows the DOM tree with the selected element highlighted.

XPath básico

Algunas expresiones xpath más básicas:

```
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='//guru99.com/images/home/java.png']
```

2) Contiene():

Contains() es un método utilizado en la expresión XPath. Se utiliza cuando el valor de cualquier atributo cambia dinámicamente, por ejemplo, la información de inicio de sesión.

La función de contención tiene la capacidad de encontrar el elemento con texto parcial como se muestra en el ejemplo de XPath a continuación.

En este ejemplo, intentamos identificar el elemento simplemente usando el valor de texto parcial del atributo. En la siguiente expresión XPath se utiliza el valor parcial 'sub' en lugar del botón enviar. Se puede observar que el elemento se encuentra con éxito.

El valor completo de 'Type' es 'submit' pero usando solo el valor parcial 'sub'.

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



```
Xpath=//*[contains(@name, 'btn')]
```

En la expresión anterior, hemos tomado el 'nombre' como atributo y 'btn' como un valor parcial como se muestra en la siguiente captura de pantalla. Esto encontrará 2 elementos (LOGIN & RESET) ya que su atributo 'name' comienza con 'btn'.

UserID
Password
LOGIN RESET

↓

FirePath

Top Window • Highlight XPath: //*[contains(@name, 'btn')]

+ <tr>
+ <tr>!-- Display Password and its text box-->
+ <tr>
- <tr>
 <td>
 <input type="submit" value="LOGIN" name="btnLogin"/>
 <input type="reset" value="RESET" name="btnReset"/>
 </td>
</tr>
</tbody>
</table>

2 Nodes Matched

2 matching nodes

Del mismo modo, en la siguiente expresión, hemos tomado el 'id' como atributo y el 'message' como un valor parcial. Esto encontrará 2 elementos ('User-ID no debe estar en blanco' y 'La contraseña no debe estar en blanco') ya que su atributo 'name' comienza con 'message'.

```
Xpath=//*[contains(@id, 'message')]
```



The screenshot shows a login form with two fields: 'UserID' and 'Password'. Both fields have validation messages: 'User-ID must not be blank!' and 'Password must not be blank!' respectively. Below the form is a FirePath tool window. The 'XPath' tab is selected with the query `//*[contains(@id,'message')]`. The tool highlights two nodes in the DOM tree corresponding to the validation messages. A callout box in the FirePath window says '2 Nodes Matched'. At the bottom left of the FirePath window, it says '2 matching nodes'.

En la siguiente expresión, hemos tomado el "texto" del enlace como un atributo y "aquí" como un valor parcial como se muestra en la siguiente captura de pantalla. Esto encontrará el enlace ('aquí') ya que muestra el texto 'aquí'.

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



Steps To Generate Access

1. Visit - [here](#)
 2. Enter your email id

↓

Top Window ▾ Highlight XPath: //*[contains(text(),'here')]

<col>

Visit - [here](http://demo.guru99.com/)

Enter your email id

Login credentials is allocated to you and mailed at your id

1 matching node

3) Uso de OR & Y:

En la expresión OR, se utilizan dos condiciones, si la 1^a condición O la 2^a condición debe ser verdadera. También es aplicable si alguna condición es verdadera o tal vez ambas. Significa que cualquier condición debe ser verdadera para encontrar el elemento.

En la siguiente expresión XPath, identifica los elementos cuyas condiciones individuales o ambas son verdaderas.

```
Xpath=//*[@type='submit' or @name='btnReset']
```

Resaltando ambos elementos como el elemento "LOGIN" que tiene el atributo 'type' y el elemento "RESET" que tiene el atributo 'name'.



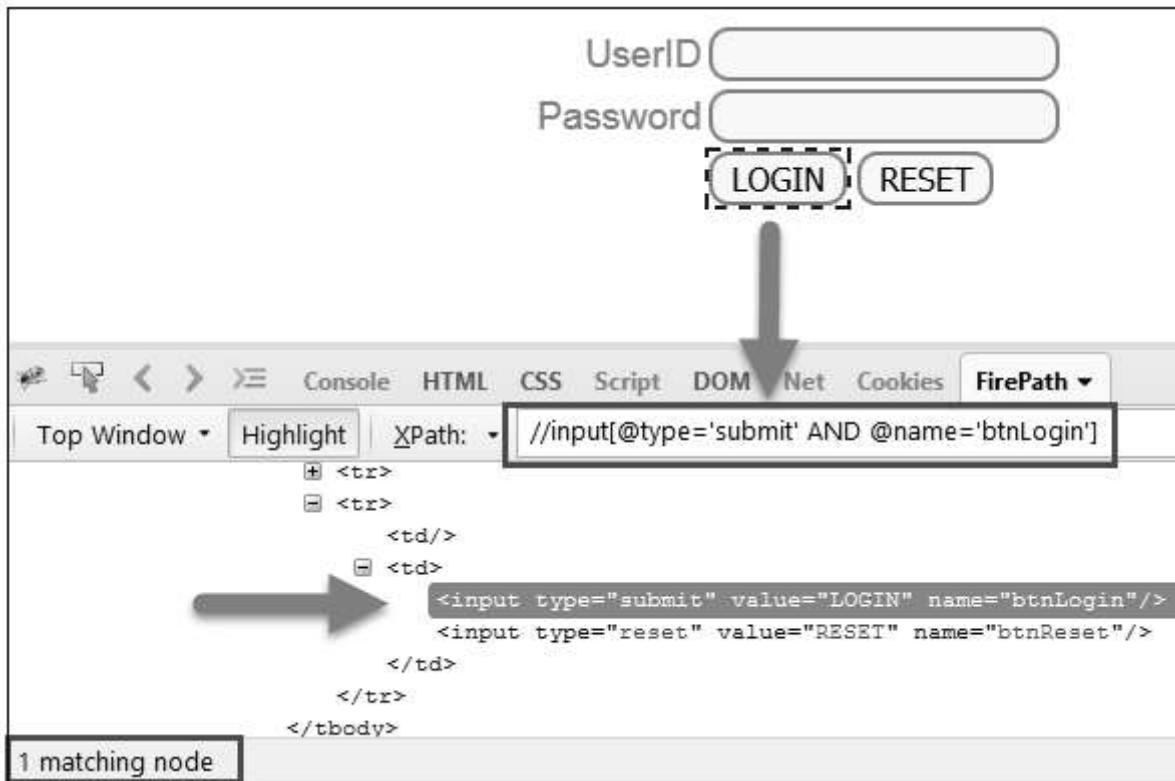
The screenshot shows a web browser window for 'Gtpl Bank'. At the top, there is a header bar with the bank's name. Below it is a login form with fields for 'UserID(*)' and 'Password(*)', both marked with an asterisk indicating they are mandatory. A note '* mandatory field' is displayed below the password field. To the right of the password field are two buttons: 'LOGIN' and 'RESET'. The 'Application' tab in the DevTools toolbar is selected. In the bottom-left corner of the DevTools interface, there is an XPath expression: `//*[@@type='submit' or @@name='btnReset']`. This expression uses double '@' symbols before type and name attributes, which is a common mistake in XPath syntax. A dashed arrow points from this incorrect expression to the 'LOGIN' button on the page, highlighting the error.

En la expresión AND, se usan dos condiciones, ambas condiciones deben ser verdaderas para encontrar el elemento. No encuentra elemento si alguna condición es falsa.

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

En la siguiente expresión, resalta el elemento 'LOGIN' ya que tiene el atributo 'type' y 'name'.





4) Xpath comienza con

XPath starts-with() es una función utilizada para encontrar el elemento web cuyo valor de atributo se cambia al actualizar o mediante otras operaciones dinámicas en la página web. En este método, el texto inicial del atributo coincide para encontrar el elemento cuyo valor de atributo cambia dinámicamente. También puede encontrar elementos cuyo valor de atributo es estático (no cambia).

Por ejemplo :- Supongamos que el ID de un elemento particular cambia dinámicamente como:

Id="mensaje12"

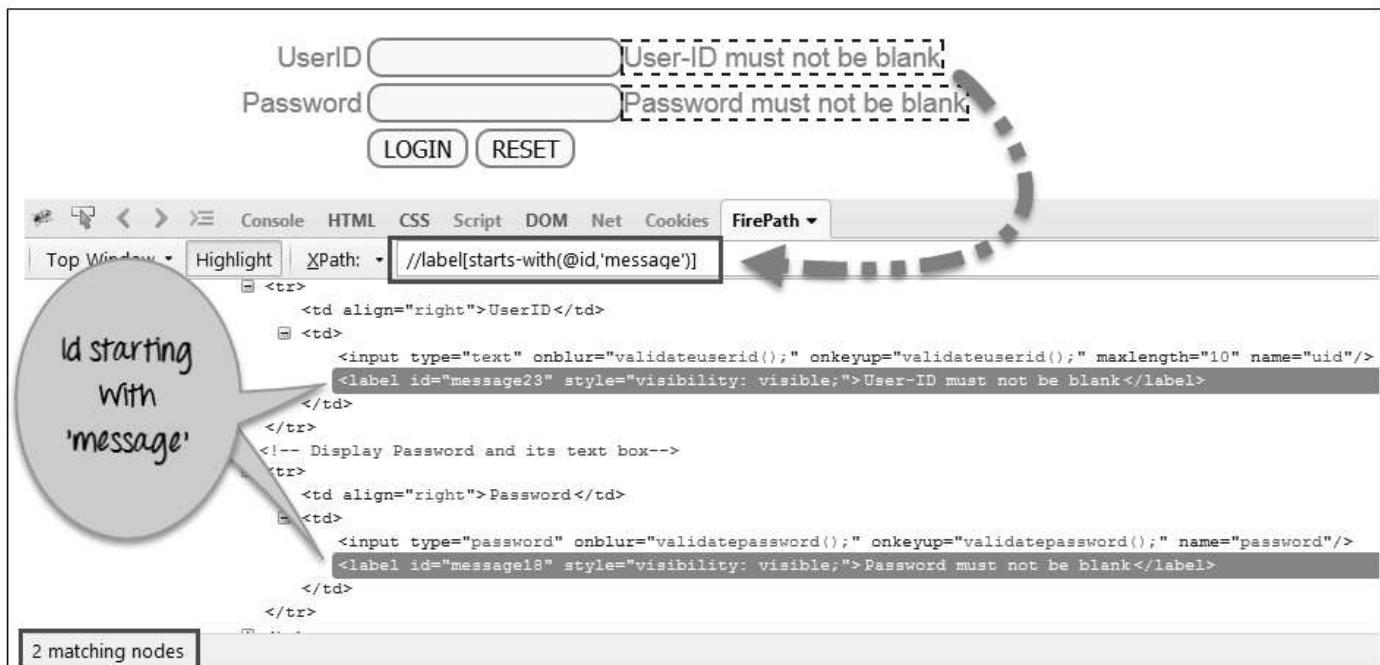


Id=" mensaje8769"

y así sucesivamente.. pero el texto inicial es el mismo. En este caso, usamos la expresión Start-with.

En la siguiente expresión, hay dos elementos con un id que comienza con "mensaje" (es decir, 'User-ID no debe estar en blanco' y 'Password no debe estar en blanco'). En el siguiente ejemplo, XPath encuentra aquellos elementos cuyo 'ID' comienza con 'message'.

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



5) Función XPath Text()

La función **XPath text()** es una función integrada de selenium webdriver que se utiliza para localizar elementos basados en el texto de un elemento web. Ayuda a encontrar los elementos de texto exactos y localiza los elementos dentro del conjunto de nodos de texto. Los elementos que se van a localizar deben estar en forma de cadena.

En esta expresión, con la función de texto, encontramos el elemento con coincidencia de texto exacta como se muestra a continuación. En nuestro caso, encontramos el elemento con texto "UserID"



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

The screenshot shows the FirePath extension integrated into the browser's developer tools. The toolbar includes tabs for Console, HTML, CSS, Script, DOM, Net, Cookies, and FirePath. The FirePath tab is active. In the main pane, the 'Highlight' tab is selected. The 'XPath' field contains the expression `//td[text()='UserID']`. The DOM tree below shows an HTML table structure with several rows and columns. One specific `<td>` element under a `<tr>` is highlighted in grey, corresponding to the 'UserID' label in the UI. The status bar at the bottom indicates '1 matching node'.

6) Métodos de ejes XPath:

Estos métodos de ejes XPath se utilizan para encontrar los elementos complejos o dinámicos. A continuación veremos algunos de estos métodos.

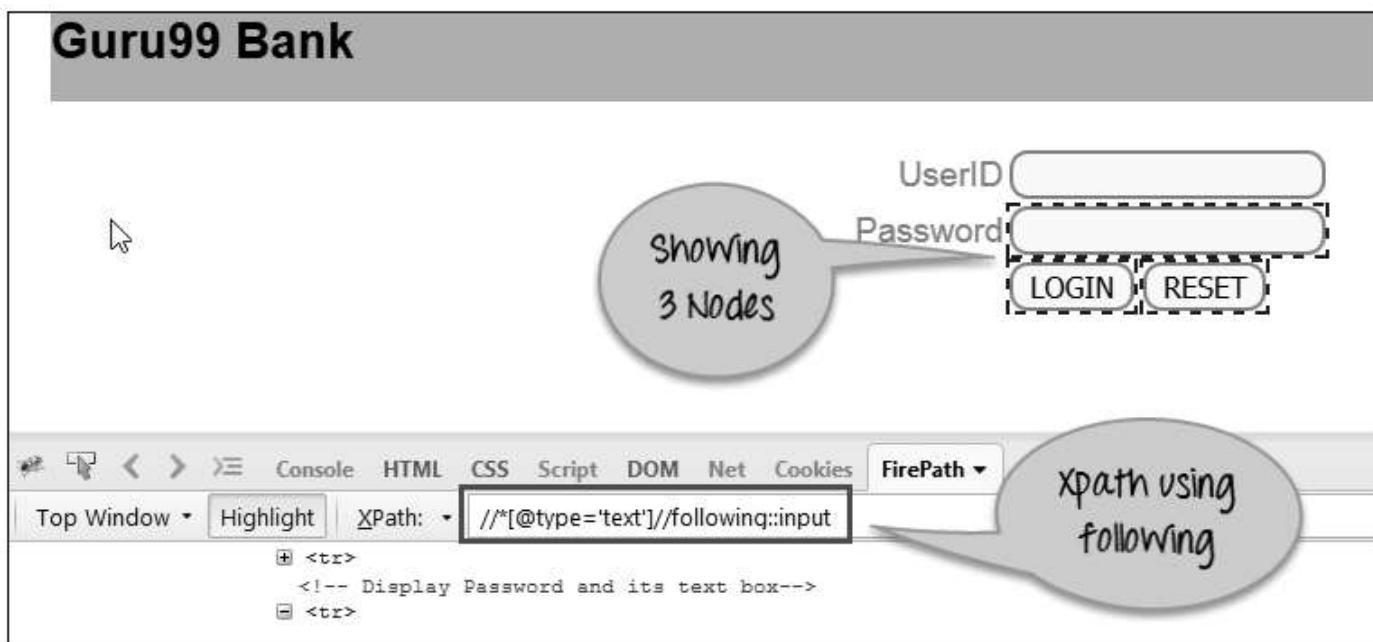
Para ilustrar estos ejes XPath, utilizaremos el sitio de demostración del banco Guru99.

a) Lo siguiente:

Selecciona todos los elementos del documento del nodo actual() [El cuadro de entrada UserID es el nodo actual] como se muestra en la pantalla siguiente.

Xpath=//*[@type='text']//following::input





Hay 3 nodos de "entrada" que coinciden mediante el uso del eje "siguiente": contraseña, inicio de sesión y botón de reinicio. Si desea centrarse en cualquier elemento en particular, puede usar el siguiente método XPath:

```
Xpath=//*[@type='text']//following::input[1]
```

Puede cambiar el XPath de acuerdo con el requisito poniendo [1],[2]..... y así sucesivamente.

Con la entrada como '1', la siguiente captura de pantalla encuentra el nodo particular que es el elemento del cuadro de entrada 'Contraseña'!



The screenshot shows the Guru99 Bank login page. A callout bubble says "Showing particular Node". On the right, there are fields for "UserID" and "Password" with "LOGIN" and "RESET" buttons. Below the page, the FirePath developer tool is open. The "XPath" dropdown shows the expression `//*[@type='text']//following::input[1]`. The DOM tree highlights the password input field under the node `<td align="right">Password</td>`.

b) Ancestro:

El eje de ancestros selecciona todos los elementos de ancestros (abuelo, padre, etc.) del nodo actual como se muestra en la pantalla de abajo.

En la siguiente expresión, estamos encontrando el elemento ancestors del nodo actual (nodo "ENTERPRISE TESTING").

```
Xpath=/* [text ()='Enterprise Testing']//ancestor::div
```

The screenshot shows the Tutorials Library page with sections for TESTING, SAP, LIVE PROJECTS, and MUST LEARN!. A callout bubble in the SAP section says "13 Nodes matched". Below the page, the FirePath developer tool is open. The "XPath" dropdown shows the expression `//*[text()='Enterprise Testing']//ancestor::div`. A callout bubble at the bottom says "Xpath using ancestor".

Hay 13 nodos "div" que coinciden mediante el uso del eje "antepasado". Si desea centrarse en cualquier elemento en particular, puede usar el siguiente XPath, donde cambia el número 1, 2 según sus requisitos:

```
Xpath=//*[text()='Enterprise Testing']//ancestor::div[1]
```

Puede cambiar el XPath de acuerdo con el requisito poniendo [1], [2]..... y así sucesivamente.

c) Niño:

Selecciona todos los elementos secundarios del nodo actual (Java) como se muestra en la siguiente pantalla.

```
Xpath=//*[@id='java_technologies']//child::li
```

The screenshot shows the FirePath extension in the Firefox developer toolbar. The DOM tree is displayed under the element `<ul id="java_technologies" class="menu">`. A speech bubble points to the node `` under the path `//*[@id='java_technologies']//child::li`. The status bar at the bottom shows "71 matching nodes".

Hay 71 nodos "li" que coinciden mediante el uso del eje "hijo". Si desea centrarse en cualquier elemento en particular, puede usar la siguiente xpath:

```
Xpath=//*[@id='java_technologies']//child::li[1]
```



Puede cambiar el xpath de acuerdo con el requisito poniendo [1],[2]..... y así sucesivamente.

d) Anteriores:

Seleccione todos los nodos que vienen antes del nodo actual como se muestra en la siguiente pantalla.

En la siguiente expresión, identifica todos los elementos de entrada antes del botón "LOGIN" que es **Userid** y el elemento de entrada **de contraseña**.

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

The screenshot shows a browser window with the FirePath extension active. The FirePath interface includes a toolbar with various icons, a dropdown menu, and a main pane displaying the DOM tree. The XPath expression `//*[@type='submit']//preceding::input` is entered in the search bar. The DOM tree highlights two input elements: one for 'UserID' and one for 'Password'. A callout bubble on the right says 'Showing 2 Nodes'. Another bubble points to the highlighted inputs with the text 'xpath using preceding'.

```

<!-- Display User ID label and its text box-->
<tr>
    <td align="right">UserID</td>
    <td>
        <input type="text" onblur="validateuserid(); onkeyup="validateuserid(); maxlengt
        <label id="message23"/>
    </td>
</tr>
<!-- Display Password and its text box-->
<tr>
    <td align="right">Password</td>
    <td>
        <input type="password" onblur="validatepassword(); onkeyup="validatepassword(); name="password"/>
        <label id="message18"/>
    </td>
</tr>

```

2 matching nodes

Hay 2 nodos de "entrada" que coinciden mediante el uso del eje "precedente". Si desea centrarse en cualquier elemento en particular, puede usar el siguiente XPath:

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



Puede cambiar el xpath de acuerdo con el requisito poniendo [1],[2]..... y así sucesivamente.

e) Hermanos siguientes:

Seleccione los siguientes hermanos del nodo de contexto. Los hermanos están en el mismo nivel del nodo actual como se muestra en la siguiente pantalla. Encontrará el elemento después del nodo actual.

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

Xpath using following-sibling

1 Nodes matched

UserID
Password
LOGIN RESET

Console HTML CSS Script DOM Net Cookies FirePath ▾

Top Window ▾ Highlight XPath: //*[@type='submit']//following-sibling::input

```

+ <tr>
  - <tr>
    - <td>
      - <td>
        <input type="submit" value="LOGIN" name="btnLogin"/>
        <input type="reset" value="RESET" name="btnReset"/>
      </td>
    </tr>
  </tbody>
</table>
</form>
```

1 matching node

Un nodo de entrada que coincide mediante el uso del eje "hermano siguiente".

f) Padre:

Selecciona el elemento primario del nodo actual como se muestra en la pantalla siguiente.



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

Hay 65 nodos "div" que coinciden mediante el uso del eje "padre". Si desea centrarse en cualquier elemento en particular, puede usar el siguiente XPath:

```
Xpath=//*[@id='rt-feature']/parent::div[1]
```

You can change the XPath according to the requirement by putting [1],[2].....and so on.

g) Self:

Selects the current node or 'self' means it indicates the node itself as shown in the below screen.



One node matching by using “self ” axis. It always finds only one node as it represents self-element.

```
Xpath =//*[@type='password']//self::input
```

h) Descendant:

Selects the descendants of the current node as shown in the below screen.

In the below expression, it identifies all the element descendants to current element (‘Main body surround’ frame element) which means down under the node (child node , grandchild node, etc.).

```
Xpath=//*[@id='rt-feature']//descendant::a
```



There are 12 “link” nodes matching by using “descendant” axis. If you want to focus on any particular element then you can use the below XPath:

```
Xpath=//*[@id='rt-feature']//descendant::a[1]
```

You can change the XPath according to the requirement by putting [1],[2].....and so on.

Summary:

XPath is required to find an element on the web page as to do an operation on that particular element.

- There are two types of selenium XPath:
 - **Absolute XPath**
 - **Relative XPath**
- XPath Axes are the methods used to find dynamic elements, which otherwise not possible to find by normal XPath method
- XPath expression select nodes or list of nodes on the basis of attributes like ID , Name, Classname, etc. from the XML document .

Also Check:- Selenium Tutorial for Beginners: Learn WebDriver in 7 Days



- How to Download and Install Selenium IDE for Firefox & Chrome
- Locate Elements by Link Text & Partial Link Text in Selenium Webdriver
- Github Integration with Selenium: Complete Tutorial
- How to Select Value from DropDown using Selenium Webdriver
- Selenium C# Webdriver Tutorial: NUnit Example

[Prev](#)[Report a Bug](#)[Next](#)

About

[About Us](#)[Advertise with Us](#)[Write For Us](#)[Contact Us](#)

Career Suggestion

[SAP Career Suggestion Tool](#)[Software Testing as a Career](#)

Interesting

[eBook](#)[Blog](#)[Quiz](#)[SAP eBook](#)

Execute online

[Execute Java Online](#)[Execute Javascript](#)

