**Tip Cómo escribir XPath Efectivos**

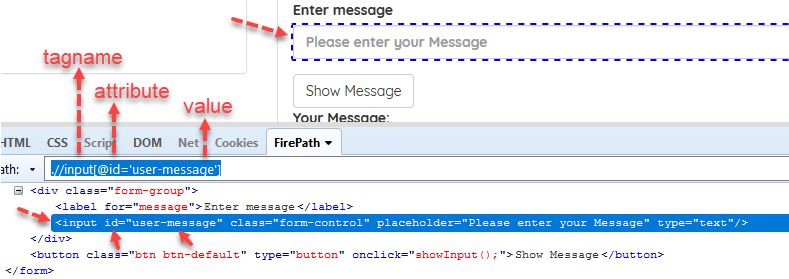
En este documento, se describiré **cómo escribir XPath inteligentes y no frágiles.**Cuando escribimos nuestros scripts de prueba, generalmente preferimos usar id, name, class, etc. estos tipos de localizadores. Sin embargo, a **veces no podemos encontrar ninguno en el DOM y también a veces necesitamos usar localizadores inteligentes para contenidos dinámicos**. Por lo tanto, en este tipo de situaciones, **debemos usar localizadores inteligentes y no frágiles**. Estos localizadores son capaces de encontrar elementos web complejos y dinámicamente cambiantes.

**Selectores de selenio XPath**

Podemos encontrar la ubicación de cualquier elemento en una página web usando expresiones de ruta XML. La sintaxis básica para XPath se muestra a continuación:

**Sintaxis =  //****tagname [@ attribute = ' Value ']**

**Ejemplo =  // input [@ id = ' user-message ']**

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c3f51c7a87f.png)

También se puede usar la **clase** , el **nombre** , el **texto del enlace** y los otros atributos para ubicar un elemento con XPath como se muestra arriba.

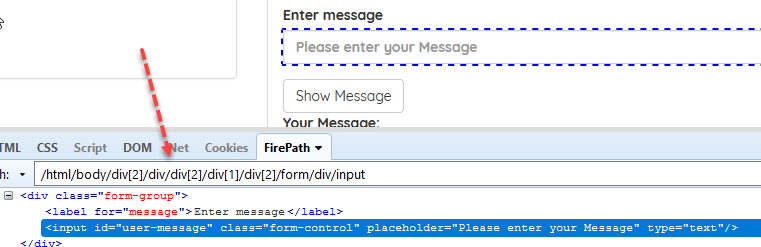
**XPath Absoluto y Relativo**

La diferencia entre XPath absoluta y relativa es muy fácil.

**Absoluto XPath**

* Es una forma directa de localizar un elemento.
* Es muy frágil.
* Comienza con una sola barra inclinada "/" que significa comenzar a buscar desde la raíz.

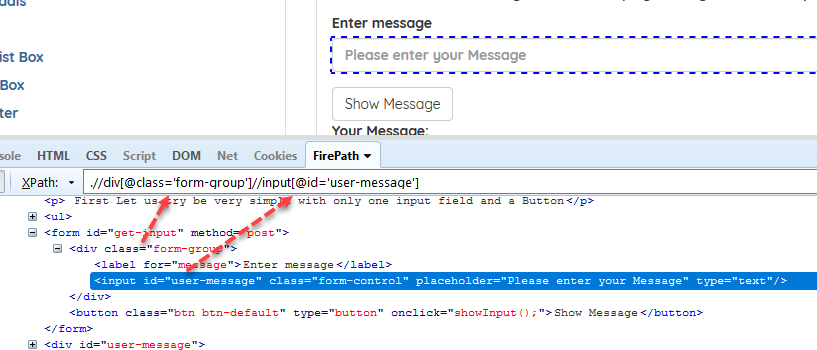
**Ejemplo:***/ html / body / div [2] / div / div [2] / div [1] / div [2] / form / div / input*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c3fdb665204.png)

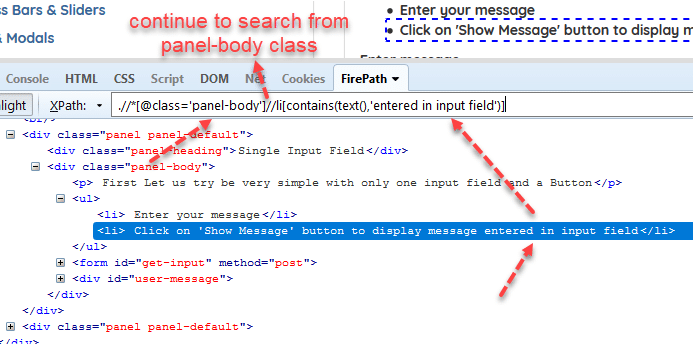
**XPath relativo**

* Comienza desde el medio del HTML DOM.
* Empieza con una barra doble "//" que significa que puede comenzar a buscar en cualquier lugar de la estructura DOM.
* Más corto que el XPath Absolute
* Menos frágil

**Ejemplo:***// div [@ class = 'form-group'] // input [@ id = 'user-message']*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c433764c4b0.png)

**Ejemplo:***// \* [@ class = 'panel-body'] // li [contains (text (), 'entered in input field ')]*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c402ff3651e.png)

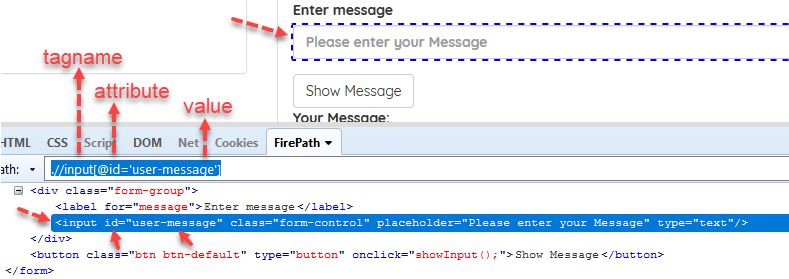
**Escribir XPaths Inteligentes para elementos complejos y dinámicos**

***Sitio de demostración de referencia:****http://www.seleniumeasy.com/test/basic-first-form-demo.html*

**1) Escribir XPath con Tag & Attribute & Value Trio**

**Sintaxis: // tag [ @attribute = ' value ']**

**Ejemplo:***// input [@id, 'user-message']*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c3f51c7a87f.png)

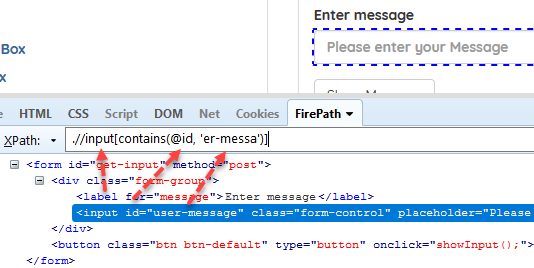
**Ejemplos:**Xpath = // input [@ type = 'enviar texto']  
Xpath = // label [@ id = 'clkBtn']  
Xpath = // input [@ value = 'SEND']  
Xpath = // \* [@ clase = 'swtestacademy']  
*->* "\*" significa, busca la clase "swtestacademy" para todas las etiquetas.   
Xpath = //a[@href='https://www.swtestacademy.com/ ']  
Xpath = //img[@src='cdn.medianova.com/images/img\_59c4334feaa6d.png']

**2) Escribir XPath con contains ()**

Es un localizador de selenio XPath muy útil y, a veces, salva la vida de un ingeniero de automatización de pruebas. Cuando un atributo de un elemento es dinámico, puede usar contains () para la parte constante del elemento web pero también puede usar contains () en cualquier condición cuando lo necesite.

**Sintaxis: // tag [****contains ( @attribute , ' value ')]**

**Ejemplo:***// input [contains (@id, 'er-messa')]*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c43a0603be8.png)

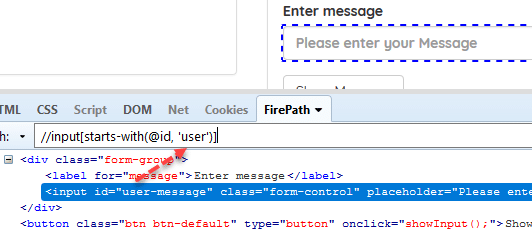
**Ejemplos:**Xpath = // \* [contains (@ name, 'btnClk')]   
-> Se interroga a "btnClk" para todos los atributos de nombre en el DOM.   
  
Xpath = // \* [contains (text (), 'here')]   
-> Interroga el texto "aquí" en el DOM.   
  
Xpath = //\*[contains(@href,'swtestacademy.com ')]   
-> Interroga el enlace "swtestacademy.com" en el DOM.

**3) Escribir XPath con starts-with**

Este método verifica el texto inicial de un atributo. Es muy útil usarlo cuando el valor del atributo cambia dinámicamente, pero también se puede usar este método para valores de atributo que no cambian.

**Sintaxis: // tag [ starts-with ( @attribute , ' value ')]**

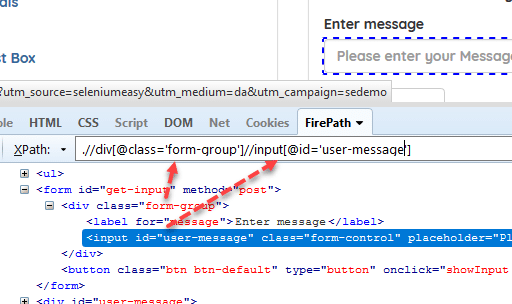
**Ejemplo:***// tag [starts-with (@id, 'user')]*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c589282806a.png)

**4) Escribir XPath con Declaración de XPaths Encadenados**

Podemos encadenar múltiples declaraciones relativas de XPath con **"//" doble barra** para encontrar la ubicación de un elemento como se muestra a continuación.

**Ejemplo:***// div [@ class = 'form-group'] // input [@ id = 'user-message']*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c43e265a7c9.png)

**5) Escribir XPath con Declaración " or "**

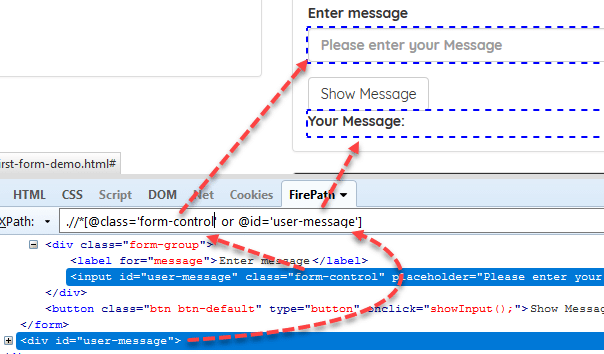
En este método, utilizamos dos condiciones de interrogación, como A y B, y devolvemos un conjunto de resultados como se muestra a continuación:

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Resultado** |
| Falso | Falso | Sin elemento |
| Cierto | Falso | Retorna A |
| Falso | Cierto | Retorna B |
| Cierto | Cierto | Retorna ambas |

**"or" distingue entre mayúsculas y minúsculas**, no debe usar mayúscula "OR".

**Sintaxis: // tag [ XPath Statement-1  or  XPath Statement-2 ]**

**Ejemplo:***// \* [@ id = 'user-message' or @ class = 'form-control']*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c59330543a4.png)

**6) Escribir XPath con la Declaración " and "**

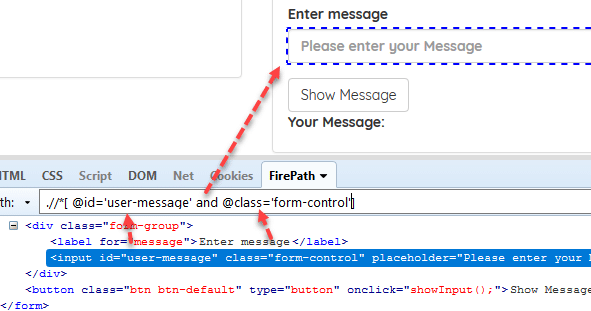
En este método, utilizamos dos condiciones de interrogación, como A y B, y devolvemos un conjunto de resultados como se muestra a continuación:

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Resultado** |
| Falso | Falso | Sin elemento |
| Cierto | Falso | Sin elemento |
| Falso | Cierto | Sin elemento |
| Cierto | Cierto | Devuelve ambos |

**"and" distingue entre mayúsculas y minúsculas**, no debe usar mayúscula "AND".

**Sintaxis: // tag [ XPath Statement-1 and XPath Statement-2 ]**

**Ejemplo:***// \* [@id = 'user-message' and @class = 'form-control']*

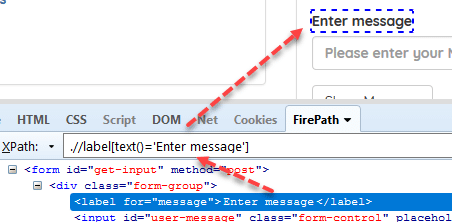
[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c59455274e0.png)

**7) Escribir XPath con text()**

Podemos encontrar un elemento con su texto exacto.

**Sintaxis: // tag [ text () = ' text value ']**

**Ejemplo:***.//label[text()='Enter message ']*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c59684e73fb.png)

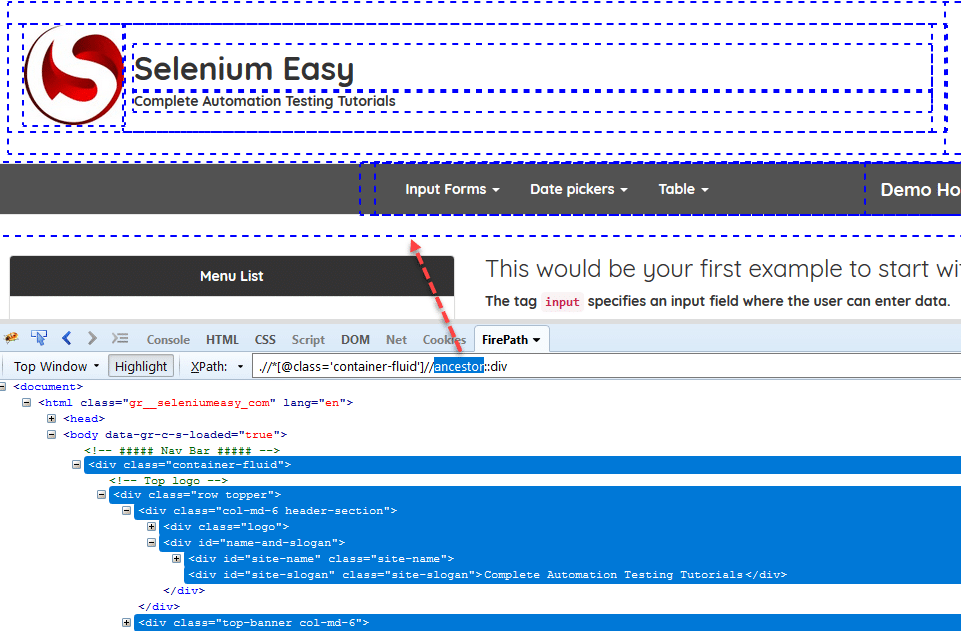
**8) Escribir XPath con ancestor**

Encuentra el elemento antes del enunciado ancestro y lo establece como un nodo superior y luego comienza a **buscar los elementos en ese nodo**. En el siguiente ejemplo,

1- Primero, encuentra la clase cuya identificación es “container-fluid”

2- Luego, comienza a encontrar elementos div en ese nodo.

**Ejemplo**: *// \* [@ class = 'container-fluid'] // ancestor :: div*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c5a079b1efd.png)

Puede seleccionar grupos div específicos cambiando las profundidades div como se muestra a continuación.

*.//\*[@class='container-fluid']//ancestor::div[1] - Devuelve 13 nodos*  
*.//\*[@class='container-fluid']//ancestor::div[2] - Devuelve 7 nodos*  
*.//\*[@class='container-fluid']//ancestor::div[3] - Devuelve 5 nodos*  
*.//\*[@class='container-fluid']//ancestor :: div [4] - Devuelve 3 nodos*  
*.//\*[@class='container-fluid']//ancestor::div[5] - Devuelve 1 nodo*

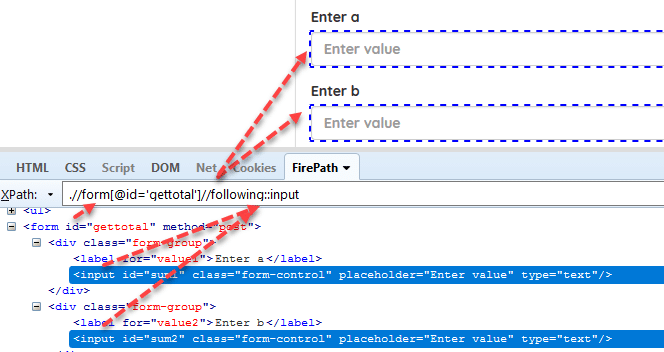
**9) Escribir XPath con los following**

Comienza a ubicar elementos **después del nodo primario dado** . Encuentra el elemento antes de la siguiente declaración y establece como el nodo superior y luego comienza a buscar **todos los elementos después de ese nodo** . En el siguiente ejemplo,

1- Primero, encuentra el form cuyo id es "gettotal"

2- Luego, comienza a buscar todos los elementos input después de ese nodo.

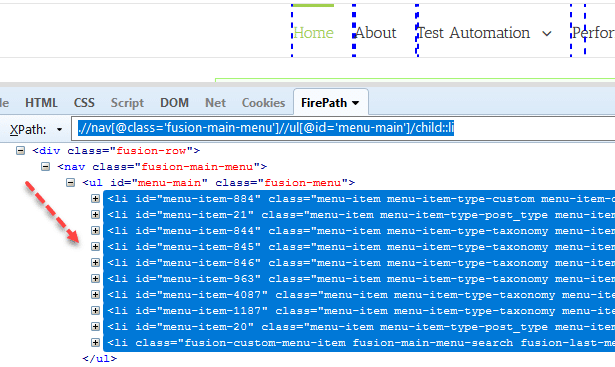
**Ejemplo** : *.//form[@id='gettotal']//* *following*::  *input*



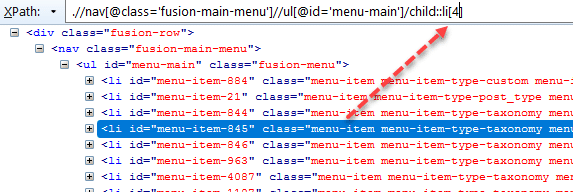
**10) Escribir XPath con un child**

Selecciona todos los elementos secundarios del nodo actual.

**Ejemplo** : *// nav [@ class = 'fusion-main-menu'] // ul [@ id = 'menu-main'] / child :: li*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c6c37765d52.png)

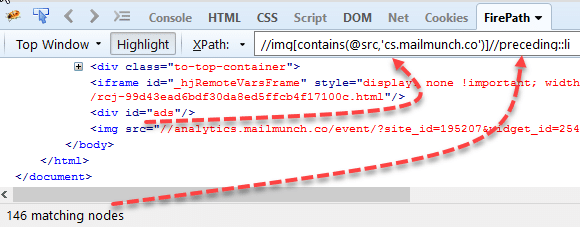
También puede **seleccionar el elemento "li" requerido utilizando la** sintaxis **li [1], li [2], li [3]** , etc., como se muestra a continuación.

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c6c5d7aa4bb.png)

**11) Escribir XPath con preceding**

Seleccione todos los nodos que vienen antes del nodo actual. Doy un ejemplo en swtestacademy. Encontraremos todos los elementos "li" en la página de inicio. Primero, ubicaremos el elemento inferior, luego usaremos preceding con "li" para encontrar todos los elementos "li" como se muestra a continuación.

**Ejemplo** :  *//img[contains(@src,'cs.mailmunch.co')]//preceding::li*

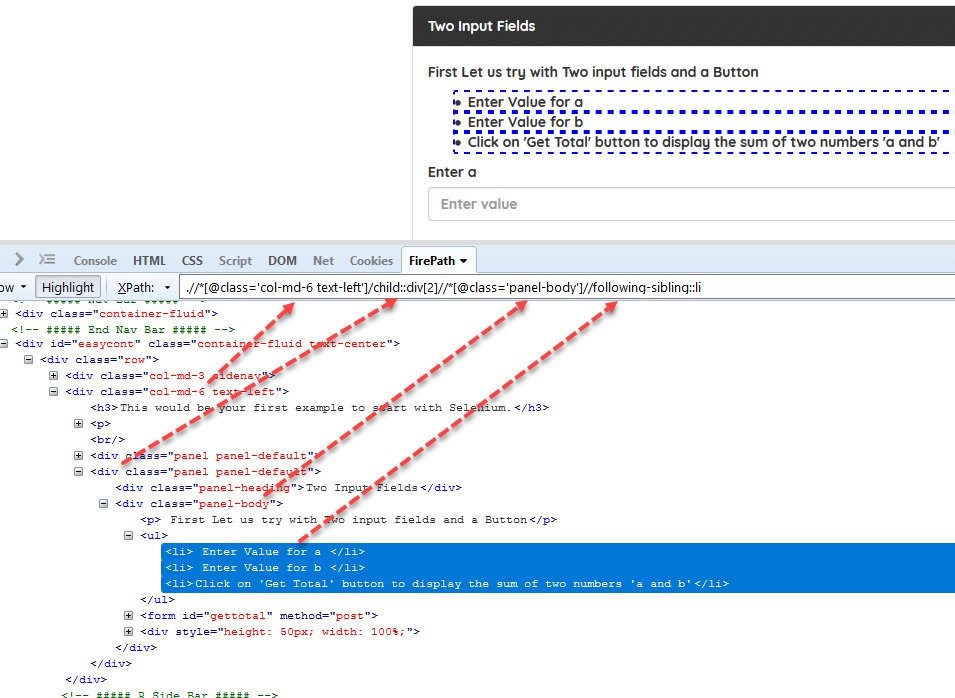
[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c6c9a364c70.png)

Además, puede usar [1], [2], etc. para seleccionar un elemento específico en la lista de elementos anterior.

**12) Escribiendo XPath con el following-sibling**

Seleccione los siguientes hermanos del context node.

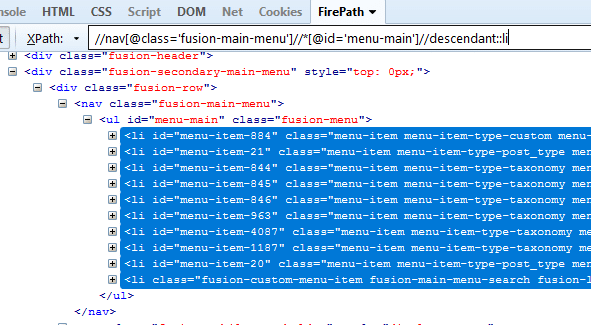
**Ejemplo** : *//\*[@class=’col-md-6 text-left’]/child::div[2]//\*[@class=’panel-body’]//following-sibling::li*

[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c6d20d3c51d.png)

**13) Escribir XPath con Descendant**

Identifica y devuelve todos los descendientes de los elementos al elemento actual, lo que significa que se desplaza hacia abajo debajo del nodo del elemento actual. A continuación, XPath devuelve todos los elementos "li" en el " menu-main".

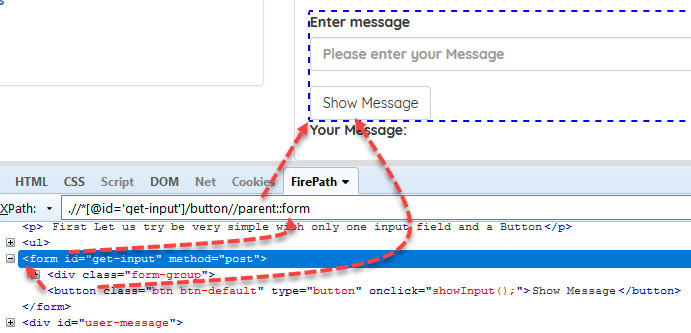
**Ejemplo** :  *// nav [@ class = 'fusion-main-menu'] // \* [@ id = 'menu-main'] // descendant:: li*



**14) Escribir XPath con el Parent**

Devuelve el padre del nodo actual como se muestra en el ejemplo siguiente.

**Ejemplo** : *.//\*[@id='get-input']/button//parent::form*

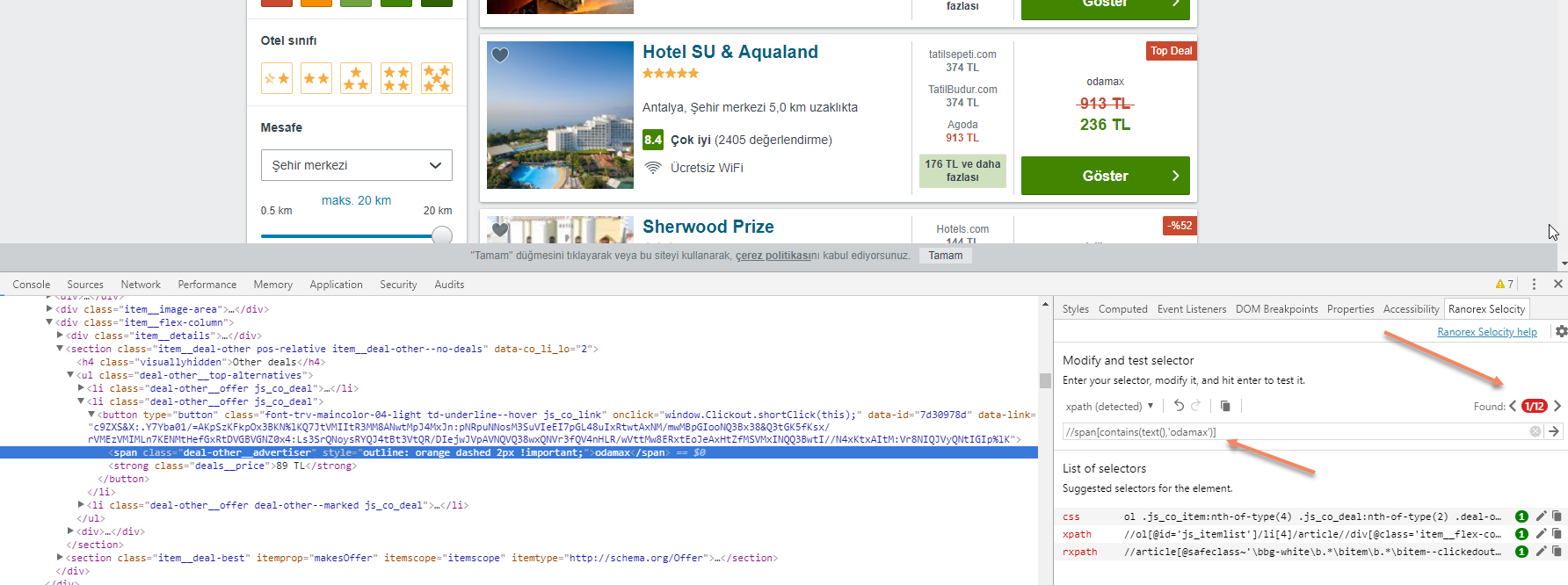
[](https://www.swtestacademy.com/wp-content/uploads/2017/09/img_59c6e1c0de69c.png)

**15) Obtenga solo un elemento específico de varios elementos con XPath Selenium**

En el sitio web de Trivago, busquemos la palabra clave "Antalya". Luego, encuentre el primer hotel de Odamax con XPath.

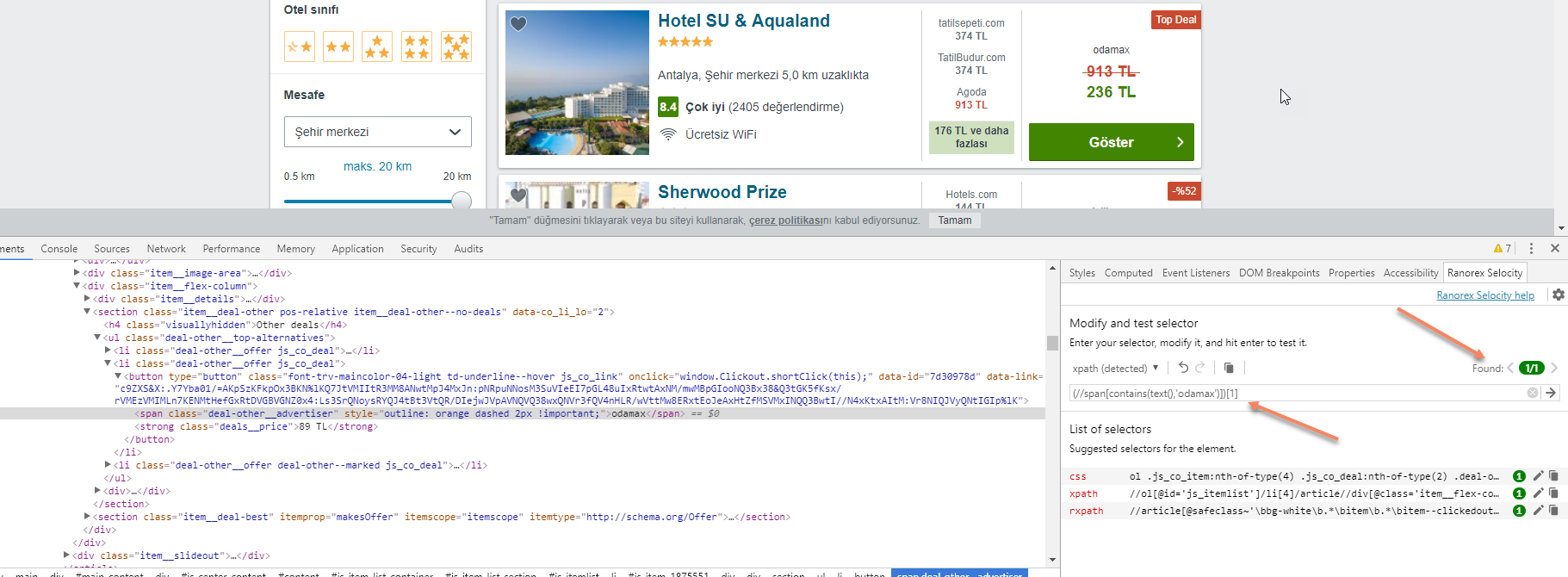
Primero, podemos encontrar todos los hoteles de Odamax usando su texto con el siguiente XPath:

**//span [contains (text (), 'odamax')]**

[](https://www.swtestacademy.com/wp-content/uploads/2018/03/img_5aacd8a995af9.png)

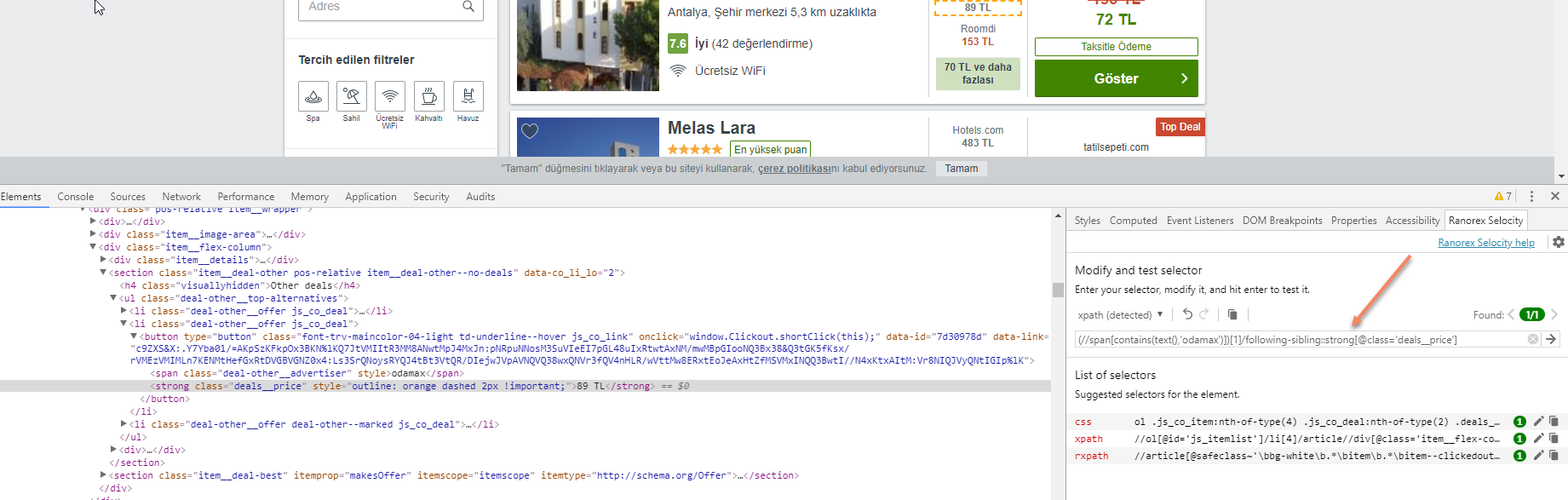
Por encima de XPath devuelve muchos hoteles Odamax, podemos seleccionar el primero con la siguiente expresión XPath:

**(// span [contains (text (), 'odamax')]) [1]**

[](https://www.swtestacademy.com/wp-content/uploads/2018/03/img_5aacd8d5f271a.png)

También puede seguir buscando y encontrar el elemento de precio del hotel relacionado con XPath a continuación:

**(// span [contains (text (), 'odamax')]) [1] / following-sibling :: strong [@ class = 'deals\_\_price']**

[](https://www.swtestacademy.com/wp-content/uploads/2018/03/img_5aacd94d651e0.png)