



« [Tutorial 10: Mejores prácticas para probar la aplicación de Android usando TestComplete](#)

[Pruebas típicas: documentos, retraso y costo adicional](#)
[proyecto de sof](#)

Usar Contiene, Hermano, Ancestro para Encontrar Elemento en Selenium

En este capítulo, vamos a analizar nuevas funciones conocidas como Hermano, Contiene y Antepasado para encontrar elemento en Selenium para nuestro script de prueba. Estas funciones están presentes en la biblioteca XPath 1.0. XPath junto con la combinación de estas nuevas funciones, podemos ubicar cualquier elemento web complejo de la página web. Comprendamos estas funciones una por una en detalle.

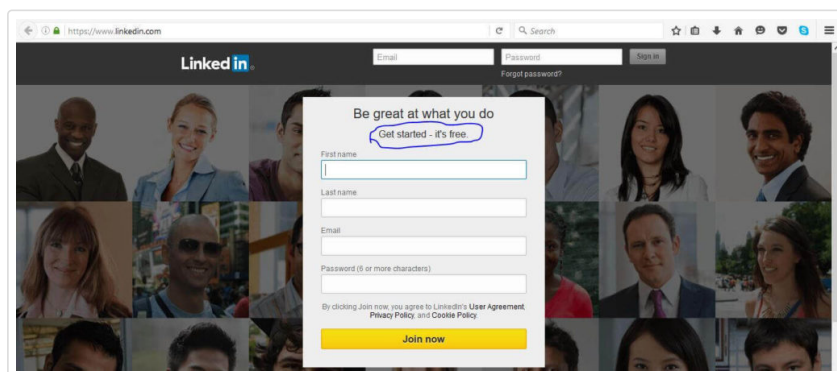


Encontrar elemento en el uso de selenium contiene

Al usar la función 'contiene', podemos extraer una lista de elementos web que contienen el texto correspondiente en toda la página web. A continuación se muestra la sintaxis para usar la función 'contains' dentro de XPath.

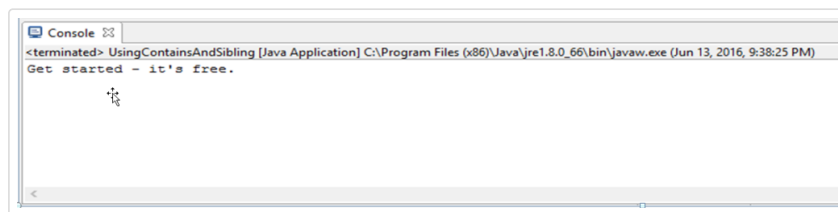
```
1 List<WebElement> elements =
  driver.findElements(By.xpath("//h1[contains(text(),'Be great
  at what you do')]/parent::form/h2"));
```

Comprendamos esto mejor con la ayuda del siguiente ejemplo. Aquí, estamos cargando una página de registro de un sitio web vinculado y la ayuda de XPath y la función 'contiene', vamos a imprimir el texto "Comenzar, es gratis" en la consola, como se muestra debajo, con un color azul.



A continuación se muestra el guión de prueba para lograr la tarea que acabamos de analizar.

```
01 package seleniumpackage;
02
03 import java.util.List;
04
05 import org.openqa.selenium.By;
06 import org.openqa.selenium.WebDriver;
07 import org.openqa.selenium.WebElement;
08 import org.openqa.selenium.firefox.FirefoxDriver;
09
10 public class UsingContainsAndSibling {
11
12     public static void main(String args[]){
13         WebDriver driver = new FirefoxDriver();
14         driver.get("https://www.linkedin.com/");
15         List<WebElement> elements = driver.findElements(By.xpath("//h1[contains(text(),'Be great at what you
16 do')]/parent::form/h2"));
17         for (WebElement webElement : elements) {
18             System.out.println(webElement.getText());
19         }
20         driver.quit();
21     }
22 }
```



Explicación del script de prueba

- En primer lugar, invocamos al controlador de Firefox y vamos a cargar la página de registro de LinkedIn.
- Al utilizar el método 'findElement' de la clase WebDriver, estamos ubicando una lista de elementos web donde estamos pasando los criterios de XPath como argumento para este método.
- Los criterios de XPath sugieren ubicar un elemento web con una etiqueta como 'h1' que contiene texto como 'Sé excelente en lo que haces'. Con la ayuda de esta etiqueta, estamos ubicando el elemento principal que es la etiqueta 'formulario'. Podemos usar este 'formulario' para ubicar el siguiente elemento web de etiqueta 'h2'.
- Después de ubicar el elemento de etiqueta 'h2', buscamos el texto e imprimimos ese texto en la consola.
- La consola muestra el texto como 'Comenzar, es gratis'.

Encontrar elemento en selenio usando hermanos

Podemos usar la función 'hermano' para buscar un elemento web que sea un hermano del elemento web principal o, en otras palabras, si conocemos el elemento padre, podemos ubicar un elemento web que puede usar el atributo hermano del XPath. A continuación se muestra sintaxis para usar la función 'hermano' dentro de XPath.

```
1 //SIBLING Demonstration
2 List<WebElement> sibElements = driver.findElements(By.xpath("//a[contains(text(),'Inside div block 2.')]//parent::div//following-sibling::div[@class='c']//a"));
```

Encuentre un elemento en el selenio usando un antepasado

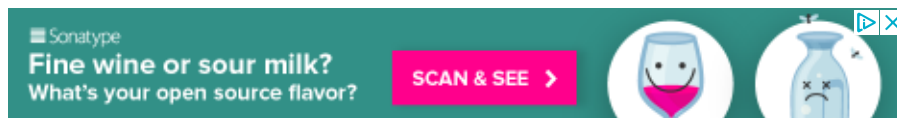
Podemos usar la función 'antecesor' para buscar un elemento web que sea un antecesor del elemento web actual o, en otras palabras, si conocemos el elemento padre, podemos ubicar un elemento web que puede usar el atributo antecesor del XPath. A continuación se muestra sintaxis para usar la función 'ancestro' dentro de XPath.

```
1 //ANCESTOR Demonstration
2 List<WebElement> ancElements = driver.findElements(By.xpath("//a[contains(text(),'Inside div block 2.')]//ancestor::div//a"));
```

En el siguiente ejemplo, estamos demostrando el uso de una función padre, hermano y ancestro que podría usarse en colaboración con X para encontrar una lista de elemento web. A continuación se muestra el código HTML para demostrar el uso de estas funciones.

```
01 <html>
02 <head>
03 <title>Form elements Demo by Selenium WebDriver</title>
04 </head>
05 <body>
06 <form>
07 <header>
08 This is header.
09 </header>
10 <div class="a">
11 <a href="#">Inside div block 1.</a>
12 <div class="b">
13 <a href="#">Inside div block 2.</a>
14 <div class="c">
15 <a href="#">Inside div block 3.</a>
16 </div>
17 </div>
18 </div>
19 <footer>
20 This is Footer.
21 </footer>
22 </body>
23 </html>
```

Explicación del código HTML



En el código HTML anterior, hemos definido tres bloques div anidados con la clase como a, byc, respectivamente. Cada uno de los bloque encierra un hipervínculo y se define como 'Dentro del bloque div 1.', 'Dentro del bloque div 2' y 'Dentro del bloque div 3' dependiendo del á bloque div en la que esté encerrado. Coloque este código HTML en el C tal que se puede acceder con URL ([archivo: /// C: /selenium_demo/Contain-Sibling-Ancestor.html](file:///C:/selenium_demo/Contain-Sibling-Ancestor.html)).

A continuación se muestra el script de prueba en Java para demostrar las funciones contener, padre, hermano y ancestro en colaboración localizador XPath.

Explicación del script de prueba

- Para empezar, invocamos al controlador de Firefox después de crear una instancia de la clase API de WebDriver 'FirefoxDriver'.
- A continuación, usamos el objeto WebDriver para invocar y cargar la URL bajo prueba (aquí [file: /// C: /selenium_demo/Contain-Sibling-Ancestor.html](file:///C:/selenium_demo/Contain-Sibling-Ancestor.html)).
- Ahora, estamos utilizando el método 'findElement' de la clase WebDriver en una lista de elementos web. Aquí, estamos pasando los criterios de XPath como un argumento para este método.
- En primer lugar, estamos demostrando los criterios de XPath donde hemos utilizado las funciones 'contiene', 'padre' y 'hermano' cor muestra a continuación. A partir de este criterio, estamos obteniendo un elemento web de hipervínculo que contiene texto como 'Inside block 2'. cuando se encuentra, estamos ubicando el elemento padre que no es más que la ^{2ª} etiqueta de bloque div. Sabemos que el siguiente hermano del ^{2º} bloque div es ^{3º} bloque div. Por lo tanto, estamos usando la función 'siguiente-hermano' que tiene la clase 'c' y, por lo tanto, selecciona el hipervínculo con una etiqueta como 'a'. Este elemento web particular se registra en la lista que tiene referencia como 'sibElements'.

```
1 //SIBLING Demonstration
2 List<WebElement> sibElements = driver.findElements(By.xpath("//a[contains(text(), \"
3 + \"'Inside div block 2.')]/parent::div/following-sibling::div[@class='c']/a\"));
```

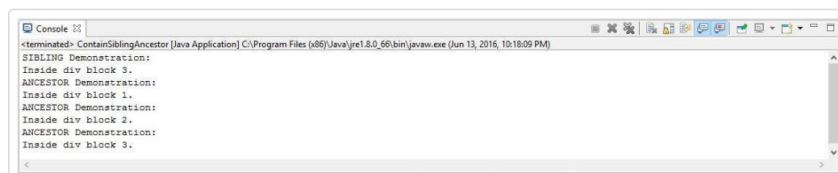
- En el siguiente paso, estamos atravesando los elementos de lista para 'sibElements' en un bucle 'for' para imprimir el valor de texto todo el elemento. En este caso, dado que había un único elemento web que cumple los criterios y tiene un valor de texto como 'Der bloque Div 3.' Por lo tanto, se imprime en la consola.
- En segundo lugar, estamos demostrando los criterios de XPath donde hemos utilizado las funciones 'contiene' y 'ancestro' como se muestra a continuación. A partir de este criterio, estamos obteniendo un elemento web de hipervínculo que contiene texto como 'Inside block 2'. cuando se encuentra, estamos ubicando el elemento web con una etiqueta como 'a'. Obviamente, podemos observar que l tres etiquetas de este tipo en cada uno de los tres bloques div anidados. Todos estos tres elementos web como etiqueta 'a' se devo a la lista de elementos web que tiene la referencia como 'ancElements'.

```
1 //ANCESTOR Demonstration<div>;
2
3 List<WebElement> ancElements = driver.findElements(By.xpath("//a[contains(text(), 'Inside div block
2.')] /ancestor::div//a\"));
```

- En el siguiente paso, estamos atravesando los elementos de lista para 'ancElements' en un bucle 'for' para imprimir el valor de texto todo el elemento. En este caso, dado que había tres elementos web que cumplen los criterios y tienen valores de texto como 'Inside block 1.', 'Inside div block 2.' y 'Dentro del bloque Div 3.' Por lo tanto, todos estos tres valores se imprimirán en la consola como se muestra a continuación.
- Por último, cerraremos la sesión actual del controlador de Firefox después de usar el método 'quit' de la clase WebDriver.

Salida

A continuación se muestra el resultado que se pudo observar una vez que ejecutamos el script de prueba anterior.



Descargar código: [Usar contiene, hermano, antepasado para encontrar elemento en selenium](#)

C onclusion:

En este capítulo, hemos aprendido acerca de las funciones Contains, Sibling, Ancestor y parent que están presentes en la biblioteca XPat que ayuda a localizar los elementos web complejos presentes en una página web.

Curso gratuito de entrenamiento de selenio

Si usted no es un lector habitual de este sitio web, le recomendamos encarecidamente que se registre para recibir nuestro boletín electrónico gratuito. Regístrese simplemente proporcionando su dirección de correo electrónico a continuación:

Ingrese su dirección de correo electrónico:

Revise el correo electrónico en su bandeja de entrada para obtener confirmación de las últimas actualizaciones de Software Testing de i gratuita.

Pruebas felices!

GUI Test Automation Tool - Easy & Comprehensive

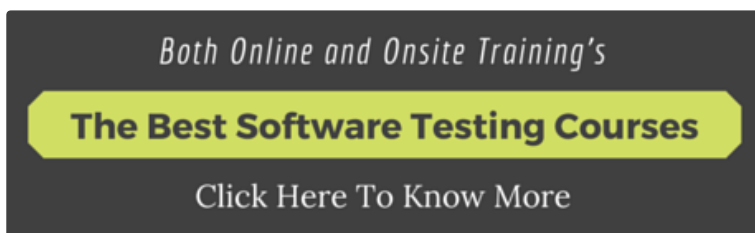
Robust UI object recognition with the Ranorex test automation tools. Try now! ranorex.com/GUI-Testing/Free-Trial



Artículos Relacionados:

1. [Selenium Training Series - Introducción al Selenium IDE](#)
2. [Sesiones, Ejecución paralela y Dependencia en el selenio](#)
3. [Prueba cruzada del navegador usando selenio](#)
4. [Manejo del selector de fecha y hora usando selenio](#)
5. [¡Seis razones principales por las que me gusta el selenio!](#)
6. [Guía completa sobre XPath en selenio](#)
7. [Comenzar con la instalación de Selenium WebDriver - Learn Selenium](#)
8. [Breve introducción de TestNG: ¿Cuál es la secuencia de ejecución de TestNG Annotations?](#)
9. [Cómo usar las anotaciones de JUnit en Selenium WebDriver Automation Script](#)
10. [Cómo crear proyectos en Eclipse IDE?](#)

Los mejores cursos de capacitación en pruebas de software



27 de febrero de 2018 | Etiquetas: [Ancestro](#) , [Encontrar elemento en selenio](#) , [Hermano](#) , [Uso contiene](#) | Categoría: [Pruebas de automatización](#) , [Selenio](#)