

TÉCNICAS



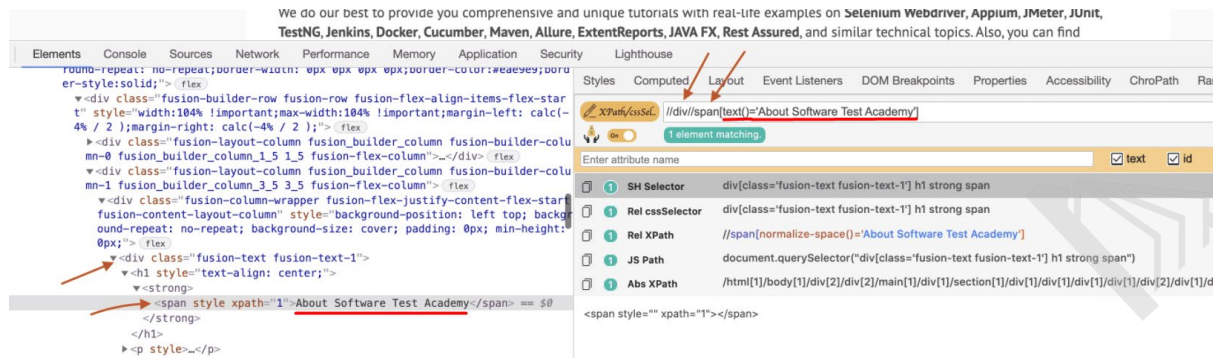
SOMOS
EVOLUCIÓN

TÉCNICAS

TEXT()

Como puede ver en la siguiente captura de pantalla, comencé con "//div", luego pasé por alto h1 y las etiquetas strong, y continué la búsqueda con el segundo "/" y luego encontré el elemento en la etiqueta span. Finalmente, la declaración se finaliza como se muestra a continuación.

Ejemplo: //div//span[text()='Acerca de Software Test Academy']

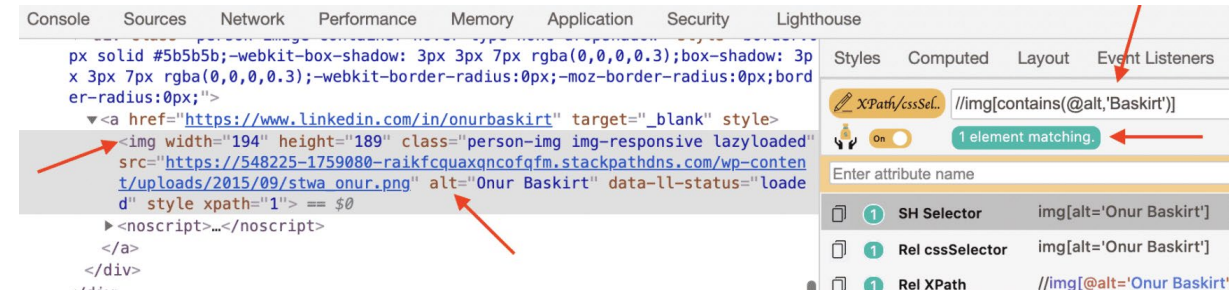


CONTAINS()

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: // etiqueta [contiene (@atributo , ' valor ')]

Ejemplo: //img[contains(@alt,'Baskirt')]

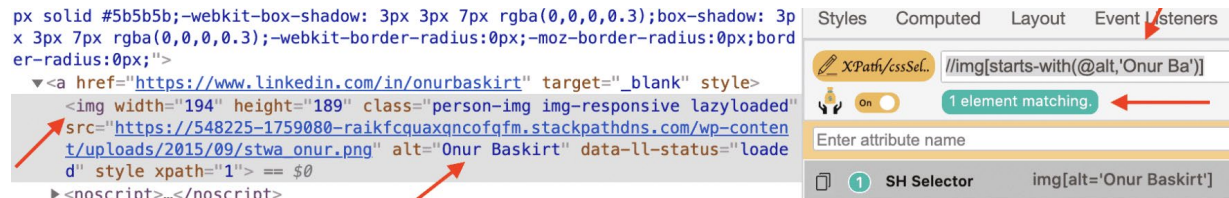


START-WITCH

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

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

Ejemplo: //img[starts-with(@alt,'Onur Ba')]

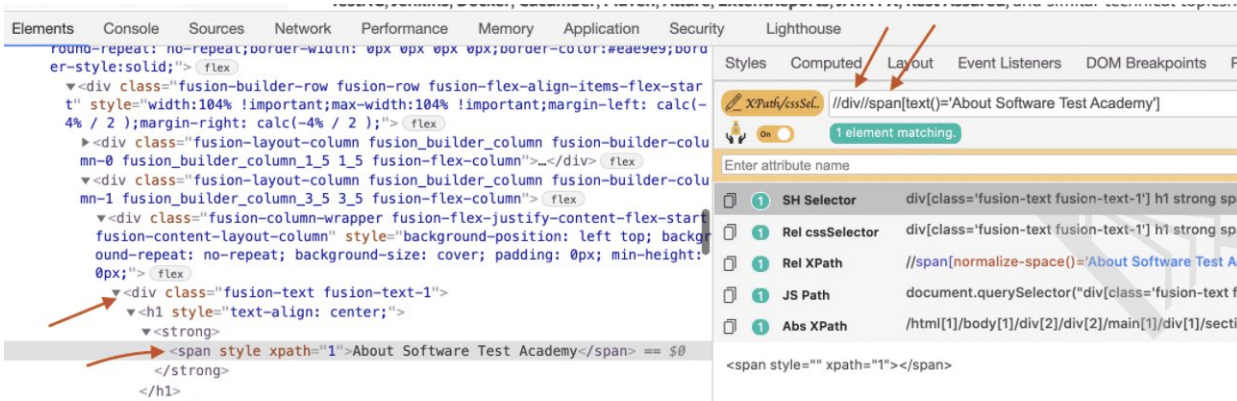


TÉCNICAS

DECLARACION ENCADENADA

Podemos encadenar varias declaraciones XPath relativas con la barra inclinada doble `"/"` para encontrar la ubicación de un elemento, como se muestra a continuación. Encadenamos `"//div"` con `"//span[text()='About Software Test Academy']"` en el siguiente ejemplo.

Ejemplo: `//div//span[text()='Acerca de Software Test Academy']`



OPERADOR " OR "

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

A	B	Resultado
Falso	Falso	Ningún elemento
Cierto	Falso	Devuelve A
Falso	Cierto	Devoluciones B
Cierto	Cierto	Devuelve ambos

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

Sintaxis: `// etiqueta [Declaración XPath-1 o Declaración XPath-2]`

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



TÉCNICAS

OPERADOR “ AND ”

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

A	B	Resultado
Falso	Falso	Ningún elemento
Cierto	Falso	Ningún elemento
Falso	Cierto	Ningún elemento
Cierto	Cierto	Devuelve ambos

“and” distingue entre mayúsculas y minúsculas , no debe usar “AND” mayúscula.

Sintaxis: // etiqueta [Declaración XPath-1 y Declaración XPath-2]

Ejemplo: //*[@id='mensaje-usuario' and @class='control-formulario']

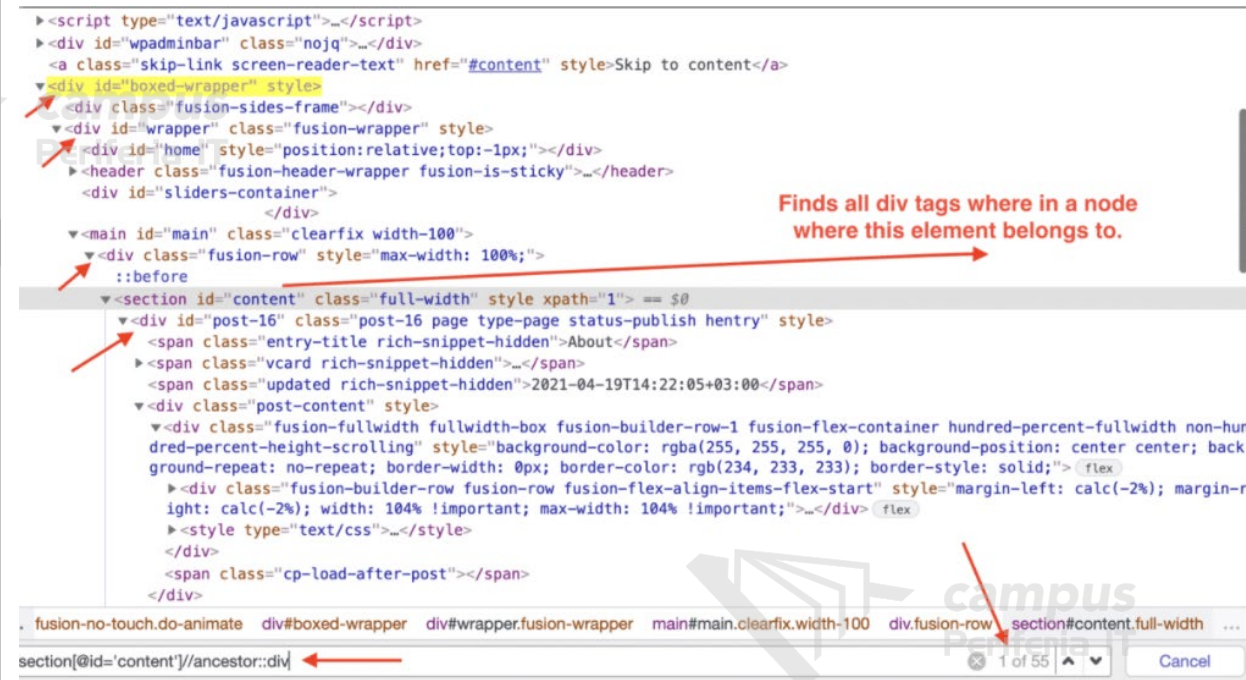


ANCESTROS

Encuentra el elemento antes de la declaración del antepasado y lo establece como un nodo superior y luego comienza a buscar los elementos en ese nodo . En el siguiente ejemplo:

1. Primero, encuentra el elemento cuyo XPath es //section[@id='content']
2. Luego, comienza a encontrar todos los elementos div en el nodo al que pertenece el XPath anterior.

Ejemplo : //sección[@id='contenido']/ancestro::div



TÉCNICAS

ANCESTROS

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

```
//section[@id='content']//ancestor::div[1]      - Devuelve 53 nodos
//section[@id='content']//ancestor::div[2]- Devuelve 33 nodos
//      section[@id='content']//ancestor::div[3]      - Devuelve 24 nodos
//section[@id='content']//ancestor::div[4]- Devuelve 19 nodos
```

FOLLOWING

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

1. Primero, encuentra este XPath con atributo de clase: `//div[@class='fusion-sliding-bar-wrapper']`
2. Luego, comienza a buscar todos los elementos de la sección después de ese nodo.

Ejemplo : `//div[@class='fusion-sliding-bar-wrapper']//following::sección`



CHILD

Selecciona todos los elementos secundarios del nodo actual.

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



También puede seleccionar el elemento de "sección" requerido usando la sintaxis de sección[1], sección[2], sección[3], etc.,




TÉCNICAS

DESENDANT

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

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

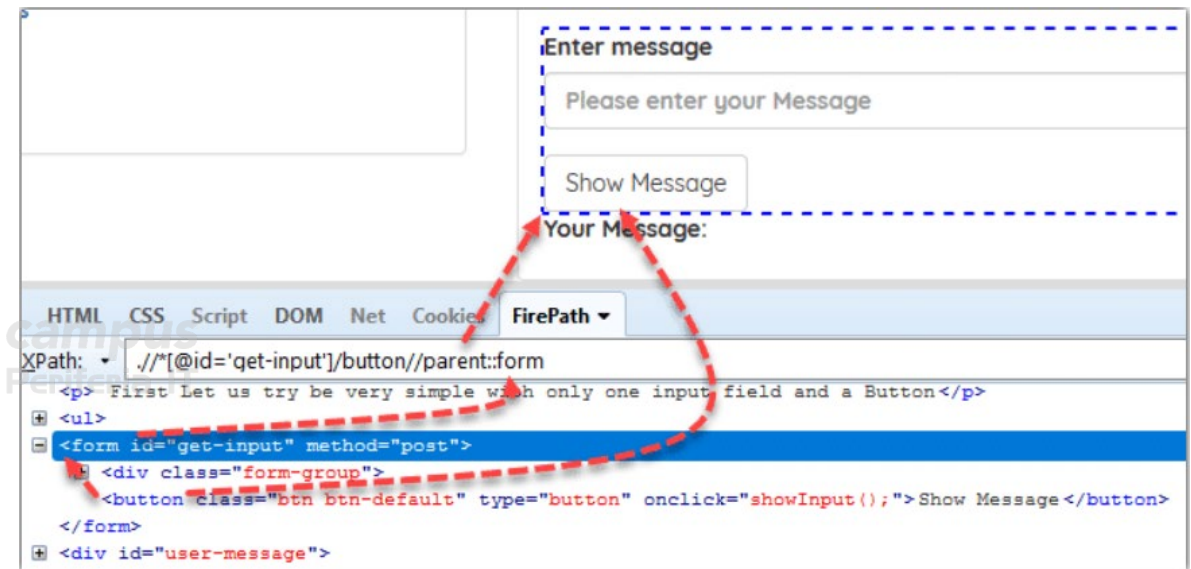


The screenshot shows the FirePath tool interface. The XPath query entered is `//nav[@class='fusion-main-menu']//*[@id='menu-main']//descendant::li`. The resulting HTML structure is displayed below, with the `<ul id='menu-main' class='fusion-menu'>` block highlighted in blue. This block contains several `` elements, each with a unique ID and class, representing the menu items.

PARENT

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

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



The screenshot shows the FirePath tool interface. The XPath query entered is `.///*[@id='get-input']/button//parent::form`. The resulting HTML structure is displayed below, with the `<form id='get-input' method='post'>` block highlighted in blue. This block contains a `<div class='form-group'>` block, which in turn contains a `<button class='btn btn-default' type='button'>` element. A red dashed arrow points from the button element to the form element, illustrating the parent relationship.