



Estrategias de CSS - Atenea Conocimientos

Estrategias para encontrar elementos utilizando CSS

Para que Cypress.io pueda hacer clic en un elemento, escribir en él o colocar el mouse dentro o fuera, la herramienta primero debe encontrar el elemento. Estos elementos pueden ser encontrados con xpath o CSS. Sin embargo, Cypress.io nos permite solamente utilizar CSS para poder identificar objetos para utilizar en nuestro marco de automatización.

Como la mayoría de las cosas poderosas, CSS tiene una pequeña curva de aprendizaje. Sin duda, es mucho más desafiante que cortar y pegar desde una herramienta. Sin embargo, si invertimos tiempo en aprender los selectores de CSS, puede tener enlaces más potentes que son más fáciles de leer, menos frágiles y un poco más integrados en la plataforma del navegador.

Estas son algunas reglas, consejos y pseudoclasas de CSS para que podamos aprender a encontrar elementos de manera efectiva.

Introducción a los selectores de CSS

Un selector de CSS es una combinación de un selector de elementos y un valor que identifica el elemento web dentro de una página web. Son representaciones de cadena de etiquetas HTML, atributos, Id y Class. Como tales, son patrones que coinciden con los elementos de un árbol y son una de varias tecnologías que se pueden utilizar para seleccionar nodos en un documento XML.

I: Simple

Id

La identificación de un elemento en XPATH se define usando: "[@ id = 'ejemplo']" y en CSS usando: "#" - Las identificaciones deben ser únicas dentro del DOM.

Ejemplos:

XPath: `//div[@id='ejemplo']`

CSS: `#ejemplo`

Tipo de Elemento

En el ejemplo anterior vimos `//div` en el xpath. Ese es el tipo de elemento, que se puede ingresar para un cuadro de texto o botón, `img` para una imagen o `a` para un enlace.

Xpath: `//input` o

Css: `=input`

Hijo directo

Las páginas HTML están estructuradas como XML, con los hijos anidados dentro de los padres. Si puede localizar, por ejemplo, el primer enlace dentro de un `div`, puede construir una cadena para llegar a él. Un hijo directo en XPATH se define mediante el uso de un `/`, mientras que en CSS, se define mediante `>`.

Ejemplos:

XPath: `//div/a`

CSS: `div > a`

Hijo or Sub-Hijo

Escribir `divs` anidados puede resultar agotador y generar un código quebradizo. A veces, espera que el código cambie o desea omitir capas. Si un elemento puede estar dentro de otro o de uno de sus hijos, se define en XPATH usando `/**` y en CSS solo con un espacio en blanco.

Ejemplos:

XPath: `//div//a`

CSS: `div a`



Estrategias de CSS - Atenea Conocimientos

Class

Para las clases, las cosas son bastante similares en XPATH: "[@class = 'ejemplo']" mientras que en CSS es simplemente "."

Ejemplos:

XPath: //div[@class='ejemplo']

CSS: .ejemplo

II: Avanzado

Siguiente Hermano

Esto es útil para navegar por listas de elementos, como formularios o elementos ul. El próximo hermano le dirá a cypress que busque el siguiente elemento adyacente en la página que está dentro del mismo padre. Mostremos un ejemplo usando un formulario para seleccionar el campo después del nombre de usuario.

```
<form class = "form-signin" role = "form" action = "/index.php" method = "post">
<h4 class = "form-signin-heading"></h4>
<input type = "text" class = "form-control" id = "username" name = "username" placeholder = "username"
required autofocus></br>
<input type = "password" class = "form-control" id = "password" name = "password" placeholder = "password"
required>
<p>
<button class = "btn btn-lg btn-primary btn-block radius" type = "submit" name = "login">Login</button>
</form>
```

Escribamos un XPath y selector css que elegirá el campo de entrada después de "username". Esto seleccionará el "alias" input, o seleccionará un elemento diferente si se reordena el formulario.

XPATH: //input[@id='username']/following-sibling:input[1]

CSS: #username + input

Valores de Atributos

Si el orden de los elementos secundarios no son importantes, puedes usar un selector de atributos para elegir elementos en función de cualquier valor de atributo. Un buen ejemplo sería elegir el elemento "nombre de usuario" del formulario anterior sin agregar una clase.

Podemos seleccionar fácilmente el elemento de nombre de usuario sin agregar una clase o una identificación al elemento.

XPATH: `//input[@name='username']`
CSS: `input[name='username']`

Incluso podemos encadenar filtros para ser más específicos con nuestros selectores.

XPATH: `//input[@name='login'and @type='submit']`
CSS: `input[name='login'][type='submit']`

Eligiendo una coincidencia específica

Los selectores CSS nos permiten navegar por las listas con más delicadeza que los métodos anteriores. Si tenemos un `` y queremos seleccionar su cuarto elemento `` sin tener en cuenta ningún otro elemento, deberíamos usar `nth-child` o `nth-of-type`. `Nth-child` es una pseudo-clase. En CSS simple, eso le permite anular el comportamiento de ciertos elementos; también podemos usarlo para seleccionar estos elementos.

```
<ul id = "recordlist">
<li>Cat</li>
<li>Dog</li>
<li>Car</li>
<li>Goat</li>
</ul>
```

Si queremos seleccionar el cuarto elemento `li` (Goat) en esta lista, podemos usar el `nth-of-type`, que encontrará el cuarto `li` en la lista. Observe los dos dos puntos, un cambio reciente en la forma en que CSS identifica las pseudoclases.

CSS: `#recordlist li::nth-of-type(4)`