

parentNode

Por medio de parentNode podemos seleccionar el elemento padre de otro elemento. Por ejemplo, si tenemos el siguiente código:

```
<div>
  <p id="introduccion">Párrafo introductorio.</p>
</div>
```

la siguiente línea de *script*:

```
document.getElementById('introduccion').parentNode;
```

selecciona el elemento padre del elemento identificado como introduccion, en este caso el div.

firstChild

Con firstChild lo que seleccionamos es el primer hijo de un elemento. Por desgracia, hay discrepancias entre los diversos navegadores sobre qué debe considerarse o no hijo de un nodo, por lo que esta propiedad en ocasiones complica demasiado un *script*.

Supongamos el siguiente fragmento de código:

```
<div id="contenido">
  <p>Un párrafo.</p>
  <p>Otro párrafo.</p>
</div>
```

Parece, al menos intuitivamente, que document.getElementById('contenido').firstChild debería seleccionarnos el primer párrafo. Sin embargo, sólo ocurre así en Internet Explorer. El problema es que navegadores como Opera o Firefox interpretan también como hijos de un elemento los posibles espacios en blanco y saltos de línea que el elemento pueda contener. Estos, por tanto considerarían que el salto de línea entre el cierre del div y la apertura del primer p es el primer hijo.

Por ello, a menos que se quiera buscar alguna solución por medio de JavaScript, suelo aconsejar no emplear firstChild si se puede evitar, y limitarlo además a elementos cuyos contenidos sean en

línea, que por lo general no contarán con saltos de línea en su interior. Por ejemplo, si el fragmento fuera el siguiente:

`<p>Nota: La autoconservación está destinada al fracaso.</p>`

`firstChild` devolvería de forma consistente el elemento `strong` en todos los navegadores actuales.

`lastChild`

La propiedad `lastChild` funciona exactamente como `firstChild`, pero se refiere al último de los hijos de un elemento. Se aplican, por tanto, las mismas indicaciones anteriores.

`nextSibling`

Gracias a `nextSibling`, lo que podemos seleccionar es el siguiente hermano de un elemento.

Se aplican las mismas limitaciones que para las dos propiedades anteriores.

`previousSibling`

`previousSibling` funciona igual que `nextSibling`, pero selecciona el hermano anterior de un elemento.