



APRENDERAPROGRAMAR.COM

JAVASCRIPT CHILDNODES,  
PARENTNODE,  
FIRSTCHILD, LASTCHILD,  
NEXTSIBLING,  
CHILDREN.LENGTH,  
CHILDELEMENTCOUNT  
(CU01125E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº25 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

## ACCESO A NODOS DEL DOM

Los navegadores web representan la página web en forma de árbol de nodos, por tanto tendremos que saber cómo acceder a dichos nodos para crear efectos dinámicos (cambios en los nodos) con JavaScript. Para ello JavaScript usa términos como childNodes, nodeType, nodeName, nodeValue, firstChild, lastChild, parentNode, nextSibling, previousSibling, etc. que vamos a estudiar.



Algunas de las palabras clave de las que disponemos en JavaScript para acceder a los nodos del DOM son las que enumeramos a continuación (veremos un ejemplo de código después):

Palabra clave	Significado	Ejemplo aprenderaprogramar.com
<b>parentNode</b>	Nodo padre de un nodo	<code>...childNodes[1].childNodes[3].parentNode</code>
<b>childNodes</b>	Array conteniendo los hijos de un nodo	<code>document.childNodes[1].childNodes[1]</code>
<b>firstChild</b>	Primer hijo de un nodo (empezando por la izquierda)	<code>document.firstChild</code>
<b>lastChild</b>	Ultimo hijo de un nodo (el más a la derecha)	<code>document.childNodes[1].lastChild</code>
<b>nextSibling</b>	Próximo nodo hermano (situado a la derecha)	<code>document.childNodes[1].nextSibling</code>
<b>previousSibling</b>	Anterior nodo hermano (situado a la izquierda)	<code>...childNodes[2].childNodes[0].previousSibling</code>

Palabra clave	Significado	Ejemplo aprenderaprogramar.com
<b>nodoAccedido.children.length</b>	Permite conocer el número de nodos hijo de un nodo	<code>document.childNodes[1].children.length</code>
<b>nodoAccedido.atributo</b>	Permite acceder al atributo de un nodo	<code>...childNodes[1].childNodes[3].name</code>

Palabra clave	Significado	Ejemplo aprenderaprogramar.com
<b>nodeName</b>	Etiqueta del nodo, como texto en mayúsculas (por ejemplo H1, DIV, SPAN...)	...childNodes[2].childNodes[0].nodeName
<b>nodeType</b>	Número que identifica el tipo de nodo (9 para document, 1 para element, 3 para text, 8 para comment)	document.nodeType
<b>nodeValue</b>	Contenido en forma de texto de un nodo de tipo text o de un nodo de tipo comment	...childNodes[2].childNodes[0].nodeValue

Para los nodos de tipo elemento hay invocaciones equivalentes (mismo efecto que la invocación general para un nodo cualquiera, pero sólo aplicables a nodos tipo elemento):

Invocación general	Equivalente para nodos tipo element
<b>firstChild</b>	firstElementChild
<b>lastChild</b>	lastElementChild
<b>nextSibling</b>	nextElementSibling
<b>previousSibling</b>	previousElementSibling
<b>children.length</b>	childElementCount

La mayor parte de los navegadores reconocen todas estas palabras claves, pero pueden existir navegadores (en especial los más antiguos) que no reconozcan algunas de ellas.

Como veremos más adelante, podemos acceder a un nodo usando JavaScript para hacer cambios dinámicos en el nodo. También podemos acceder a un nodo simplemente para mostrar algo por pantalla o recuperar alguna información. Por ejemplo:

```
Alert ('El nombre del nodo es: ' + document.childNodes[1].nodeName + ' y el tipo del nodo es: '+ document.childNodes[1].nodeType + '\n\n');
```

Esta instrucción nos mostraría por pantalla el valor de nodeName y nodeType para el segundo nodo hijo de document (dado que el primero es document.childNodes[0]). En caso de que tratemos de invocar un nodo que no exista, se producirá un error y JavaScript no se ejecutará. No aparecerá ningún mensaje de error por pantalla, simplemente no se ejecuta JavaScript.

También podemos almacenar la referencia a un nodo en una variable. Por ejemplo:

```
var nodoBody = document.childNodes[1].childNodes[2];
```

Aquí estamos asignando el nombre de variable “nodoBody” a un nodo (suponemos que document.childNodes[1].childNodes[2] es una referencia válida a un nodo). Ahora podremos hacer uso de este nombre de variable como equivalente al nodo, por ejemplo:

```
alert ('Segundo hijo de nodo body es: ' + nodoBody.childNodes[1].nodeName + '\n\n');
```

¿Qué tipo de datos corresponde a la variable nodoBody? Cuando hablamos de tipos de datos en JavaScript dijimos que existían datos primitivos y objetos. En este caso, la variable es de tipo objeto. Los objetos podemos verlos como “algo” (estructura de datos) de naturaleza compleja. Ya hemos visto que los arrays podían ser considerados objetos especiales y lo mismo podemos decir de los nodos del DOM.

En los próximos epígrafes del curso veremos ejemplos de código para acceder a los nodos y modificarlos, de momento nos es suficiente con irnos familiarizando con la sintaxis y los conceptos básicos relacionados con el DOM.

## EJERCICIO

Crea el árbol de nodos DOM para el siguiente código:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo DOM - aprenderaprogramar.com</title><meta charset="utf-8">
</head>
<body>Texto en body
<div id="cabecera" class="brillante">
<h1>Portal web <span class="destacado">aprenderaprogramar.com</span>, para aprender</h1>

</div>
<!-- Final del código--></body></html>
```

Para comprobar si tu respuesta es correcta puedes consultar en los foros [aprenderaprogramar.com](http://aprenderaprogramar.com).

**Próxima entrega:** CU01126E

**Acceso al curso completo** en [aprenderaprogramar.com](http://aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:  
[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=78&Itemid=206](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206)