**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 ver.

Algunas de las palabras clave de las que disponemos en JavaScript para acceder a los nodos del DOM son las que enumeramos a continuación

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

|  |  |  |
| --- | --- | --- |
| **Palabra clave** | **Significado** | **Ejemplo aprenderaprogramar.com** |
| **nodoAccedido.children.length** | Permite conocer el número de nodos hijo de un nodo | document.childNodes[1].children.length |
| **nodoAccedido.atributo** | Permite acceder al atributo de un nodo | ...childNodes[1].childNodes[3].name |

|  |  |  |
| --- | --- | --- |
| **Palabra clave** | **Significado** | **Ejemplo aprenderaprogramar.com** |
| **nodeName** | Etiqueta del nodo, como texto en mayúsculas (por ejemplo H1, DIV, SPAN…) | ...childNodes[2].childNodes[0].nodeName |
| **nodeType** | Número que identifica el tipo de nodo (9 para document, 1 para element, 3 para text, 8 para comment) | document.nodeType |
| **nodeValue** | 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** |
| **firstChild** | firstElementChild |
| **lastChild** | lastElementChild |
| **nextSibling** | nextElementSibling |
| **previousSibling** | previousElementSibling |
| **children.length** | 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.

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');