APRENDERAPROGRAMAR.COM

DOM O DOCUMENT OBJECT MODEL JAVASCRIPT. ¿QUÉ ES? ¿PARA QUÉ SIRVE? EL W3C. ARBOL DE NODOS. PARENT Y CHILD. EJEMPLOS (CU01123E)

Sección: Cursos
Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº23 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

DOM. DOCUMENT OBJECT MODEL.

Una de las mayores potencialidades de JavaScript es que puede manipular cualquier elemento de una página web y modificar su contenido, su tamaño, su color, su posición... e incluso hacer aparecer y desaparacer elementos. Para hacer esto posible necesitamos identificar con precisión cada elemento de una página web para poder indicarle a JavaScript sobre qué elemento debe operar.

Para permitir que los lenguajes de programación pudieran extraer información o manipular cualquier elemento de una página web era necesario definir de alguna manera qué tipos de elementos conforman una página web, cómo nombrarlos y cómo se relacionan entre sí. Inicialmente esto resultaba bastante problemático, ya que no existía un estándar o especificación oficial sobre cómo debía realizarse esto.

Para evitar los problemas de falta de estandarización, un organismo internacional (el W3C) definió un estándar denominado DOM ó Document Object Model (Modelo de objetos para representar documentos) que define qué elementos se considera que conforman una página web, cómo se nombran, cómo se relacionan entre sí, cómo se puede acceder a ellos, etc.

EL WORLD WIDE WEB CONSORTIUM O W3C

DOM es un estándar cuya definición ha sido creada por el World Wide Web Consortium, cuyas siglas son W3C. Se trata de un consorcio internacional que produce especificaciones, recomendaciones, manuales y herramientas en relación al desarrollo de internet. El W3C es un organismo que cuenta entre sus principales promotores al MIT (Massachussets Institute of Technology, USA), el ERCIM (European Research Consortium for Informatics and Mathematics, participado por numerosos países), la Keio University (Japón) y la Beihang University (China).

Uno de los objetivos principales del W3C es generar estándares: documentos donde se definen las sintaxis de lenguajes y protocolos que intervienen en el desarrollo de internet. El objetivo es promover que las empresas, instituciones y personas que participan o trabajan en desarrollos web utilicen un mismo lenguaje y se pongan de acuerdo a la hora de generar software y productos relacionados con internet. ¿Por qué decimos que “la raíz” de una página web es document y webPage ó internetDocument? Porque esta forma de nombrar y organizar las páginas web ha sido definida por el W3C de esta manera y todas (o casi todas) las empresas, instituciones y personas lo han aceptado.

No siempre lo que propone el W3C es aceptado. El W3C emite propuestas, no leyes de obligado cumplimiento porque no tiene capacidad legal para ello. Hay otras instituciones o grandes empresas que también hacen propuestas o tienen criterios propios, y en ocasiones esas propuestas o criterios alternativos hacen que haya distintos grupos de trabajo y distintos estándares o forma de funcionamiento del software.

Por último, hay que tener en cuenta que el W3C está formado por un equipo de personas que también cometen errores y que "lo que dice el W3C" no siempre tiene por qué ser lo mejor ni lo más usado. No obstante, hoy día el W3C es la principal institución de referencia a la hora de crear y difundir estándares relacionados con los desarrollos web, entre ellos el estándar CSS y el estándar DOM.

LAS VERSIONES DE DOM

El W3C trabaja continuamente para mejorar el DOM, corrigiendo errores e incorporando nuevas funcionalidades. Antes de llegar a una especificación o recomendación oficial se trabajan numerosos borradores que son sometidos a revisión y corrección. Cuando se alcanza un relativo grado de acuerdo entre los miembros del W3C se libera lo que se denomina una recomendación oficial de DOM ó versión a modo de propuesta para su uso y aplicación por todas las empresas, instituciones y personas.

Las versiones de DOM a lo largo de la historia han sido*:

- Dom Level 1: publicada en 1998.
- Dom Level 2: publicada en 2000.
- Dom Level 3: publicada en 2004.
- Dom Level 4: se esperaba como recomendación oficial en 2016.

*Las fechas indicadas son sólo orientativas, la realidad es que una version no aparece un día, sino que tiene un largo proceso de desarrollo que a veces dura años.

Nosotros nos atendremos a la versión de DOM de más amplia difusión en cada momento. No nos interesa tanto conocer la especificación oficial al completo, sino entender cómo está concebido y para qué sirve el DOM.

Por otro lado, hay que tener en cuenta que "seguir con exactitud" una especificación oficial no significa que nuestra web vaya a funcionar perfectamente, debido a que no todos los navegadores reconocen todas las propiedades o sintaxis que se definen como recomendación oficial. También puede suceder que un navegador sí reconozca la sintaxis pero no ofrezca el mismo resultado que otro, lo cual da lugar a problemas en la visualización de páginas web.

Conseguir buenos resultados con JavaScript pasa por estar al día de la especificación del W3C pero también por seguir las novedades de la web, de los navegadores y siendo prácticos, por hacer muchas pruebas y comprobaciones con distintos navegadores o herramientas específicas para este fin.

REPRESENTACIÓN DOM DE UNA PÁGINA WEB

Decimos que una página web es un documento HTML. Este documento puede ser representado de diferentes maneras:

a) Representación web: como una página web en un navegador donde vemos imágenes, texto, colores, etc.

b) Representación texto: como un texto plano (código HTML) que podemos visualizar en cualquier editor de textos como el bloc de notas de Windows ó Notepad++ ó cualquier otro.

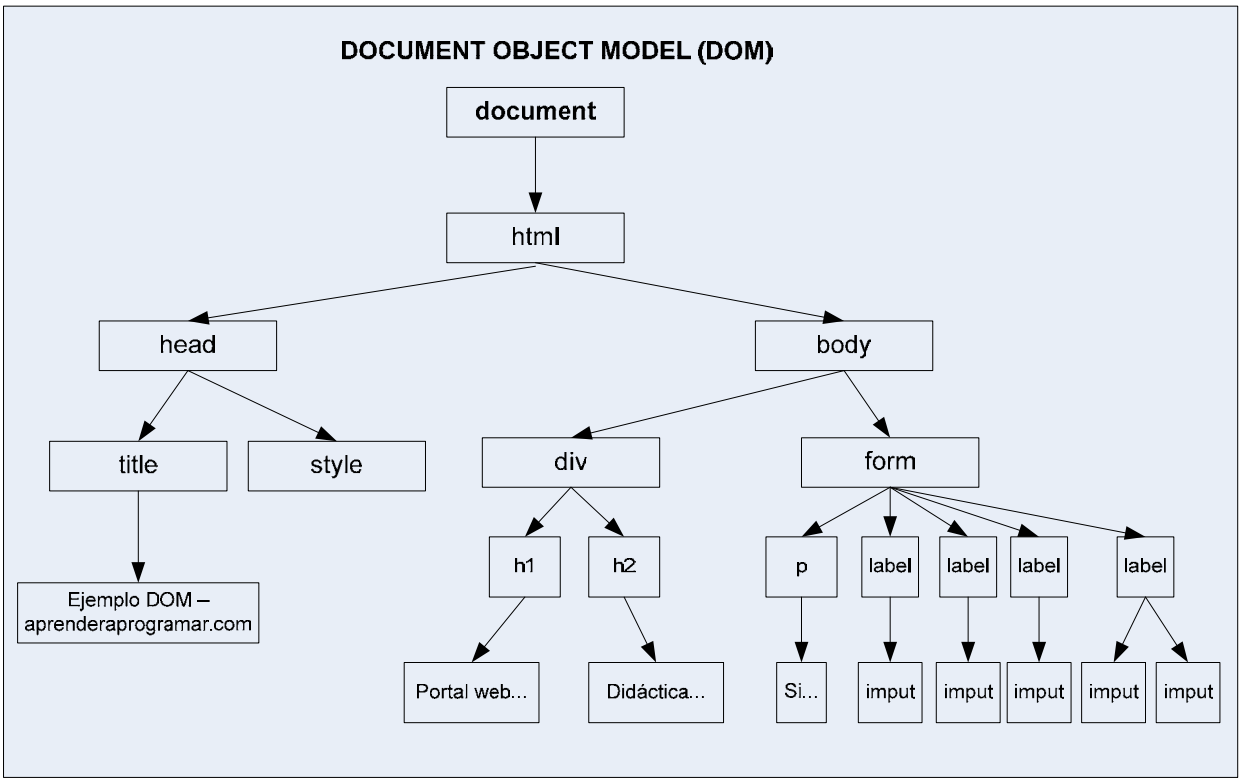
c) Representación DOM: como un árbol donde los elementos de la página web están organizados jerárquicamente, con nodos superiores (nodos padre o parent) y nodos que derivan de los nodos padre (nodos hijo o child).

Veamos un ejemplo de representación DOM. Escribe el siguiente código y guárdalo en un archivo de extensión html:

```
<!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">
<style type="text/css">
body {background-color:yellow; font-family: sans-serif;}
label {color: maroon; display:block; padding:5px;}
</style>
</head>

<body>
<div id="cabecera">
<h1>Portal web aprenderaprogramar.com</h1>
<h2>Didáctica y divulgación de la programación</h2>
</div>

<!-- Formulario de contacto -->
<form name ="formularioContacto" class="formularioTipo1" method="get" action="accion.html">
<p>Si quieres contactar con nosotros envíanos este formulario relleno:</p>
<label for="nombre"><span>Nombre:</span> <input id="nombre" type="text" name="nombre" /></label>
<label for="apellidos"><span>Apellidos:</span> <input id="apellidos" type="text" name="apellidos" /></label>
<label for="email"><span>Correo electrónico:</span> <input id="email" type="text" name="email" /></label>
<label>
<input type="submit" value="Enviar">
<input type="reset" value="Cancelar">
</label>
</form>

</body>

</html>
```

La anterior representación se corresponde con la representación del documento como texto. Visualiza el resultado. La imagen que vemos en el navegador se corresponde con la representación del documento como página web en un navegador.

La representación del documento conforme al estándar del DOM sería (de forma aproximada) esta:



La representación anterior es solo aproximada: no nos va a interesar representar una página web conforme al DOM, simplemente queremos conocer cómo se estructura una página web conforme al DOM para saber cómo podemos acceder a sus elementos y manipularlos usando JavaScript (u otro lenguaje).

El DOM no es parte de JavaScript, de hecho puede ser utilizado por otros lenguajes de programación. No obstante, el DOM está íntimamente ligado a JavaScript ya que JavaScript lo utilizará con profusión para acceder y modificar las páginas web dinámicamente.

Decimos que conforme al DOM la página web se representa como un árbol de nodos, interconectados y relacionados de acuerdo con una jerarquía.

Como nodo principal, inicial o padre de todos los nodos en el árbol de nodos de una página web se encuentra un nodo especial cuyo nombre es document.

El nodo raíz tiene como nodo hijo al nodo html, que representa a todo lo contenido dentro de las etiquetas <html> … <html>.

El nodo html tiene dos hijos: head y body. Sucesivamente cada nodo tiene uno o varios hijos hasta que se llega a un punto terminal donde no existen más hijos.

La construcción del árbol de nodos es realizada automáticamente por los navegadores de acuerdo con las reglas internas que tienen definidas. Por ejemplo una regla interna dice que una etiqueta define un nodo, por ejemplo una etiqueta <p> define un nodo, y a su vez el contenido de texto de esa etiqueta define otro nodo.

Una página web avanzada supone la construcción de un árbol de nodos que puede constar de miles de nodos y cambiar continuamente a medida que el usuario realiza la navegación a través de diferentes páginas web. Todo este proceso es realizado “en segundo plano” por el navegador y dada la potencia de los computadores actuales, es un proceso muy rápido.

No todos los navegadores construyen el árbol de nodos de igual manera, ni siquiera todos construyen el mismo árbol de nodos (lo cual puede resultar problemático al dar lugar a distintos resultados cuando ejecutemos JavaScript). La mayor parte de los navegadores siguen el estándar W3C para construir el árbol de nodos, pero algunos navegadores no lo siguen con exactitud o introducen algunas extensiones adicionales específicas de ese navegador.

Un nodo constituye un elemento complejo que suele llevar información asociada como veremos más adelante.

A través de JavaScript podremos acceder a nodos con instrucciones del tipo document.getElementById("menu"); (acceder a un elemento por su id), pero también podremos crear nodos (por ejemplo document.createElement("h1"); ó document.body.appendChild(heading);). Todo esto es posible gracias a la existencia del DOM.

EJERCICIO

Crea una página web html que conste de las etiquetas html, head, body. Dentro de body incorpora dos div: uno que contenga una etiqueta h1 con el texto “Curso JavaScript aprenderaprogramar.com” y otro que contenga tres párrafos que contengan: Párrafo 1, Párrafo 2 y Párrafo 3. Crea la representación del árbol de nodos conforme al DOM para este documento. Para comprobar si tu respuesta es correcta puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01124E

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