

DOM, definición

La palabra DOM (Document Object Model) o Modelo de Objetos de Documento hace referencia a la interfaz de programación de aplicaciones ([API](#)) para documentos HTML, XHTML y XML que define la estructura lógica de los documentos y la manera en la que se accede a ellos, así como su manipulación.

A través del DOM, los desarrolladores pueden crear nuevos documentos, navegar por su estructura, añadir, modificar o eliminar elementos y su contenido.

El responsable del DOM es el [W3C \(World Wide Web Consortium\)](#).

Estructura del DOM

En el DOM, los documentos se representan bajo una estructura del tipo árbol, a través del cual, de un elemento principal, dependen uno, ninguno o varios elementos secundarios. Los tipos de nodos que se encuentran en una página web se enumeran a continuación:

- **Document:** es el nodo raíz a partir de cual se crean los demás.
- **Element:** hace referencia a cada una de las etiquetas HTML siendo el único nodo que puede contener atributos y del que se pueden crear otros nodos.
- **Attr:** cada atributo de una etiqueta genera este nodo, el cual, tiene su valor como una propiedad.
- **Text:** hace referencia al texto de una etiqueta HTML.
- **Comment:** son los comentarios que se incluyen en el HTML.

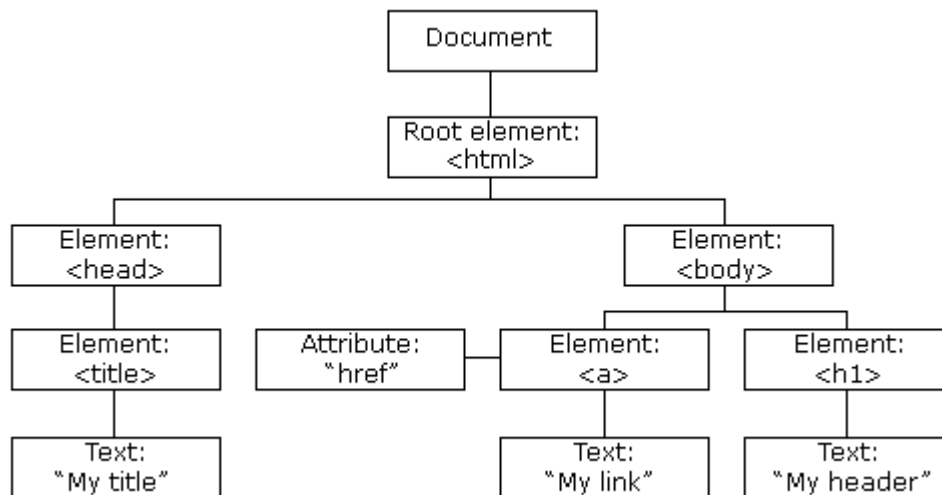
Estructura DOM para los documentos HTML

En un documento HTML, la etiqueta principal es **html**, conocida como el elemento root. A partir de ésta, se crean dos etiquetas llamadas **head** y **body**, ambas al mismo nivel.

En la etiqueta **head**, se definen los metadatos del documento, como el título de la página, las etiquetas meta que hacen referencia, por ejemplo a su codificación y se añaden scripts para el funcionamiento del documento.

En la etiqueta **body**, se define la estructura o maquetado del documento. Se pueden añadir tantas etiquetas como sean necesarias además, se pueden

manipular los estilos a través de CSS y ayudar a la interacción a través del lenguaje JavaScript. En esta etiqueta, también pueden ser añadidos scripts.



Lo que el DOM no es según el W3C

Según las especificaciones del W3C, **el DOM no es**:

- El Modelo de Objetos del Documento no es una especificación binaria.
- El Modelo de Objetos del Documento no es una manera de ofrecer objetos persistentes para XML o HTML, DOM especifica los documentos como XML y HTML son representados como objetos, de modo que pueden ser utilizados por programas orientados a objetos.
- El Modelo de Objetos del Documento no es un conjunto de estructuras de datos, es un modelo de objeto que especifica aplicaciones.
- El Modelo de Objetos del Documento no define qué información en un documento es relevante o que información en un documento es estructurado.

¿Cuál es la diferencia con el BOM?

El BOM (Browser Object Model) o Modelo de Objetos de Navegador hace referencia a **la convención de los elementos del navegador**. A diferencia con el DOM, el BOM **no requiere una estructura específica ni ninguna definición**, por lo que los proveedores de navegadores pueden desarrollar la estructura que más les convenga.

El concepto fue introducido por las versiones 3.0 de los navegadores Netscape e Internet Explorer el cual, permite acceder y modificar las propiedades de las ventanas de los navegadores. Algunas de los elementos que lo conforman son los siguientes:

- Se pueden crear, mover, redimensionar y cerrar ventanas de los navegadores.
- Se puede obtener información sobre el navegador.
- Se pueden gestionar las cookies.
- Se pueden obtener y modificar propiedades de la página actual y de la pantalla del usuario.
- Se pueden manipular objetos ActiveX en el navegador Internet Explorer.

Relación del DOM con el SEO

Dado que los tiempos de renderización de un sitio web influyen directamente en el posicionamiento de dicho sitio web, es necesario conocer cómo influye la estructura del DOM en el mismo.

Reducir y aplicar correctamente los elementos del DOM

Cuantos más elementos tenga el DOM, más tardará en cargar la web ya que **aumenta mucho el uso de memoria y hacen que los cálculos de los estilos de las hojas CSS duren más**, por lo que reducir el uso de los mismos a los justamente necesarios para el maquetado de la web, influirá de manera positiva en su renderización. Además, se deben utilizar las etiquetas de forma correcta. Por ejemplo, se deben utilizar los encabezados siguiendo los niveles, primero h1, luego h2, después h3 y así sucesivamente hasta h6.

Correcto uso de JavaScript

Otro factor importante a tener en cuenta es el correcto uso de JavaScript y AJAX (Asynchronous JavaScript and XML) ya que permite a los desarrolladores realizar **cambios en el contenido sin tener que cargar todo el HTML**, por lo que la web cargará más rápido y los cambios se apreciarán más rápido.

Uso de un CDN

Un CDN (Content Delivery Network) es una red de servidores que se encuentran conectados y distribuidos por todo el mundo en varios centros de datos, que tienen por objetivo la entrega flexible de contenido. Se debe considerar el uso de CDN, por ejemplo, de librerías como jquery dentro de los

scripts del DOM ya que reducen considerablemente el tiempo de carga de la web ya que, el usuario se traerá los contenidos desde el CDN más cercano geográficamente hablando. Esto **evita el tener que tener dicha librería almacenada dentro del servidor** dónde se encuentra alojado dicho sitio web, por lo que la cantidad de contenido a procesar será menor.

Minificar el contenido del HTML, CSS y JavaScript

La técnica de minificar los ficheros es ampliamente conocida dentro del desarrollo web. Minificar consiste en hacer que todo el contenido de un fichero se quede sin espacios, saltos de línea y comentarios, de forma que **el usuario no aprecia ningún cambio en el diseño, solamente afecta al rendimiento**. Desde el punto de vista del desarrollador, una vez minificado el contenido y si el desarrollador necesita realizar algún cambio, éste puede utilizar alguna herramienta para que el código vuelva a su estado original. Minificar tanto los HTML, los CSS y los JavaScript es una buena técnica.