

## Modelo en Objetos para la Representación de Documentos (DOM)

**Document Object Model** o **DOM** ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos') es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos [HTML](#), [XHTML](#) y [XML](#). El ente encargado de normar, documentar y difundir todos estos modelos y lenguajes de marcado es *World Wide Web Consortium* (W3C).

W3c ( <https://www.w3.org/Icons/SVG/svg-w3c-h.svg> )

### Análisis sintáctico del HTML

Cuando el navegador web comienza a recibir los datos HTML de una página web a través de la red, inmediatamente el [analizador sintáctico](#) comienza a trabajar para convertir el HTML en un Modelo de Objeto de Documento (DOM).

## [\[ SVG: Esquema DOM \]](#)

### Esquema DOM

El primer paso de este proceso de análisis sintáctico es descomponer el HTML en fichas que representen las etiquetas de inicio, las etiquetas de finalización y sus contenidos. A partir de eso puede construir el DOM.

## Obtención de los recursos externos

Cuando el analizador sintáctico se encuentra con un recurso externo como un archivo CSS o JavaScript, va y solicita esos archivos. El analizador continuará mientras se carga un archivo CSS, aunque bloqueará la *renderización* (dibujado en pantalla) hasta que se haya cargado y analizado (más sobre eso en un momento).

Los archivos JavaScript son un poco diferentes ya que por defecto bloquean el análisis del HTML mientras se carga el archivo JavaScript y luego se analiza. Hay dos atributos que pueden ser añadidos a las etiquetas de los guiones (*scripts*) para mitigar esto: **defer** (diferir) y **async** (asíncrono). Ambos permiten que el analizador continúe mientras el archivo JavaScript se carga en segundo plano, pero funcionan de manera diferente en la forma en que se ejecutan. Más sobre eso en un poco también; en resumen:

- **defer**: significa que la ejecución del archivo se retrasará hasta que se complete el análisis sintáctico del documento. Si varios archivos tienen el atributo `defer`, se ejecutarán en el orden en que fueron descubiertos en el HTML.
- **async**: significa que el archivo se ejecutará tan pronto como se cargue, lo que podría ocurrir durante o después del proceso de análisis sintáctico y, por lo tanto, no se puede garantizar el orden en que se ejecutan los guiones de **async**.

Ejemplos de ambas instrucciones:

## Precarga de recursos

Aparte de esto, los navegadores web modernos continuarán explorando el código HTML mientras esté bloqueado y "mirando hacia adelante" a ver qué recursos externos están llegando y luego los descargan especulativamente. La manera en que lo hacen varía entre los diferentes navegadores, por lo que no se puede confiar en que se comporten de cierta manera en específico. Para marcar un recurso como importante y, por lo tanto, más probable que se descargue al principio del proceso de *renderizado*, se puede utilizar una etiqueta de enlace con **preload** de esta manera:

## Análisis sintáctico del CSS y el CSSOM

El [Modelo de Objetos CSS \(CSSOM\)](#) es un mapa de todos los selectores CSS y las propiedades relevantes para cada selector en forma de árbol, con un nodo raíz, hermano, descendiente, hijo y otra relación. El CSSOM es muy similar al DOM. Ambos forman parte de la ruta de *renderización* crítica, que es una serie de pasos que deben darse para *renderizar* correctamente un sitio web. El CSSOM, junto con el DOM, para construir el árbol de *renderización*, que a su vez es utilizado por el navegador para diseñar y pintar la página web.

Similar a los archivos HTML y el DOM, cuando se cargan los archivos CSS deben ser analizados y convertidos en un árbol - esta vez el CSSOM. Describe todos los selectores CSS de la página, su jerarquía y sus propiedades.

Donde el CSSOM difiere del DOM es que no puede ser construido de manera incremental, ya que las reglas de CSS pueden sobreescribirse en varios puntos diferentes debido a su especificidad. Es por eso que los bloques CSS se *renderizan*, ya que hasta que todo el CSS es analizado y el CSSOM es construido, el navegador no puede saber dónde y cómo posicionar cada elemento en la pantalla.

## Ejecución de los guiones JavaScript

Dependiendo de cómo y cuándo se cargan los recursos de JavaScript serán determinantes para ejecutar JavaScript, pero eventualmente en algún momento se analizarán, compilarán y ejecutarán. Diferentes navegadores tienen diferentes motores de JavaScript para realizar esta tarea. El análisis sintáctico de JavaScript puede ser un proceso costoso en términos de recursos de un ordenador, mucho más que otros tipos de recursos, por lo que optimizarlo es tan importante para lograr un buen rendimiento.

## Eventos de carga

Una vez que el JavaScript (cargado de manera asíncrona) y el DOM estén completamente analizados y listos, se emitirá el evento especial [document.DOMContentLoaded](#). Para cualquier guión que requiera acceso al DOM (por ejemplo para manipularlo de alguna manera o escuchar los eventos de interacción del usuario) es una buena práctica esperar primero a este evento antes de ejecutar los guiones.

```
document.addEventListener('DOMContentLoaded', (event) => {  
    // Aquí comienza a funcionar nuestra aplicación.  
});
```

Después de que todo lo demás, como el JavaScript asíncrono, las imágenes, etc. hayan terminado de cargarse, entonces el evento [window.load](#) se dispara.

```
window.addEventListener('load', (event) => {  
    // Nuestra página web está totalmente cargada y analizada.  
});
```

## Árbol de *renderizado*

El [árbol de renderización](#) es una combinación del DOM y el CSSOM, y representa todo lo que se *renderizará* en la página. Esto no significa necesariamente que todos los nodos del árbol de *renderizado* estén presentes visualmente, por ejemplo, los nodos con estilos de **opacity: 0** o **visibility: hidden** estarán incluidos, y pueden ser leídos por un lector de pantalla, etc., mientras que los que están configurados con **display: none** se incluirán ninguno de ellos. Además, las etiquetas como **que no contienen ninguna información visual siempre serán omitidas. Al igual que con los motores de JavaScript, los [diferentes navegadores tienen diferentes motores de representación](#).**

## Presupuesto de pintura

Tal como cuando vamos a pintar una casa cada diciembre, debemos calcular cuánto es el área total para que (sabiendo cuántos metros cuadrados rinde cada litro de pintura) estimar el coste



- « ».
- « ».
- « ».

## En idioma francés

- « ».
- « ».
- « ».

## En idioma inglés



Beethoven era un buen compositor porque utilizaba ideas nuevas en combinación con ideas antiguas. Nadie, ni siquiera Beethoven podría inventar la música desde cero. Es igual con la informática

(Richard Stallman)

[akifrases.com](http://akifrases.com)

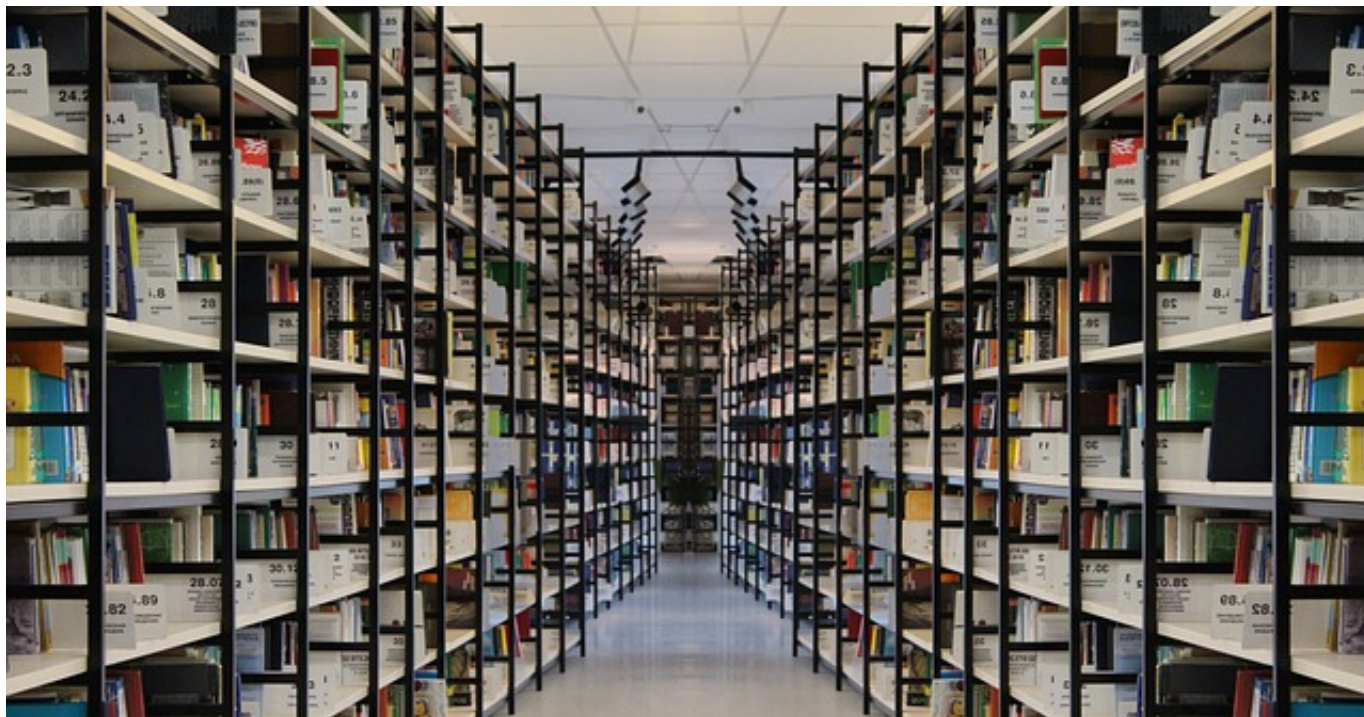
## KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

---

- «How the browser renders a web page» by James Starkie ( <https://dev.to/jstarmx/how-the-browser-renders-a-web-page-1ahc>)
- « ».
- « ».



Crédito de la imagen [Gerd Altmann](#), [trabajo](#), licencia de uso: [Pixabay](#)

---