

Esta página ha sido traducida del inglés por la comunidad. Aprende más y únete a la comunidad de MDN Web Docs.

## Cómo funciona CSS

Hemos aprendido los conceptos básicos de CSS, para qué sirve y cómo escribir hojas de estilo simples. En esta lección vamos a echar un vistazo a cómo un navegador crea una página web a partir de CSS y HTML.

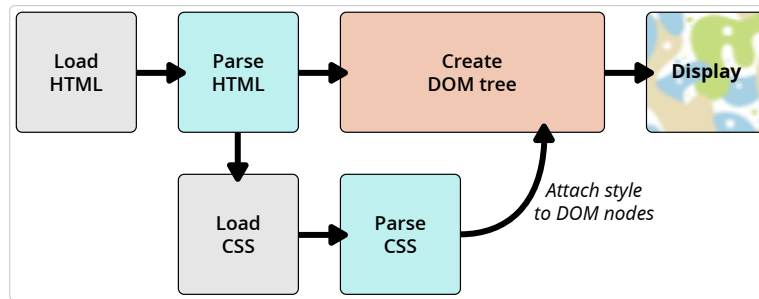
<b>Prerrequisitos:</b>	Conocimientos básicos de informática, tener el <a href="#">software básico</a> instalado, conocimientos básicos de cómo <a href="#">trabajar con archivos</a> y conceptos básicos de HTML (véase <a href="#">Introducción a HTML</a> ).
<b>Objetivo:</b>	Entender los conceptos básicos de cómo el navegador analiza el CSS y el HTML y lo que sucede cuando encuentra un CSS que no entiende.

### ¿Cómo funciona realmente el CSS?

Cuando un navegador muestra un documento, ha de combinar el contenido con la información de estilo del documento. Procesa el documento en una serie de etapas, que enumeraremos a continuación. Ten en cuenta que este es un modelo muy simplificado de lo que sucede cuando un navegador carga una página web y que cada navegador gestiona el proceso de manera diferente. Pero esto es más o menos lo que sucede.

1. El navegador carga el HTML (por ejemplo, lo recibe de la red).
2. Convierte el [HTML](#) en un [DOM](#) (*Modelo de objetos del documento*). El DOM representa el documento en la memoria del ordenador. Lo explicaremos más detalladamente en la sección siguiente.
3. Entonces, el navegador va a buscar la mayor parte de los recursos vinculados al documento HTML, como las imágenes y los videos incrustados... ¡y también el CSS vinculado! JavaScript aparece un poco más adelante en el proceso, pero no vamos a hablar de ello aún para evitar complicar las cosas.
4. El navegador analiza el CSS y ordena en diferentes «cubos» las diferentes reglas según el tipo de selector. Por ejemplo, elemento, clase, ID, y así sucesivamente. Para cada tipo de selector que encuentre, calcula qué reglas deben aplicarse y a qué nodos en el DOM se les aplica el estilo según corresponda (este paso intermedio se llama árbol de renderización).
5. El árbol de renderización presenta la estructura en que los nodos deben aparecer después de aplicarle las reglas.
6. En la pantalla se muestra el aspecto visual de la página (esta etapa se llama pintura).

El siguiente diagrama ofrece una visión sencilla de este proceso.



## Acerca del DOM

Un DOM tiene una estructura en forma de árbol. Cada elemento, atributo o bloque en el lenguaje de marcado se convierte en un [nodo DOM](#) con estructura de árbol. Los nodos se definen por su relación con otros nodos DOM. Algunos elementos son padres de nodos secundarios, y estos nodos hijos tienen hermanos.

Comprender el DOM te ayuda a diseñar, depurar y mantener tu CSS porque en el DOM es donde tu CSS se encuentra con el contenido del documento. Cuando comiences a trabajar con las herramientas DevTools (o herramientas del desarrollador) del navegador, te moverás por el DOM mientras seleccionas elementos con el fin de ver qué reglas se aplican.

## Una representación real de un DOM

En lugar de una explicación larga y aburrida, veamos un ejemplo para entender cómo un código HTML se convierte en DOM.

Tomemos el siguiente código HTML:

HTML	Play
<pre> &lt;p&gt;   Usaremos:   &lt;span&gt;Hojas&lt;/span&gt;   &lt;span&gt;de estilo&lt;/span&gt;   &lt;span&gt;en cascada&lt;/span&gt; &lt;/p&gt; </pre>	

En el DOM, el nodo que se corresponde con nuestro elemento `<p>` es un padre. Sus hijos son un nodo de texto y los tres nodos correspondientes a nuestros elementos `<span>`. Los nodos `SPAN` son también los padres, y los nodos de texto sus hijos:

```

P
├─ "Usaremos:"
├─ SPAN
│  └─ "Hojas"
├─ SPAN
│  └─ "de estilo"
└─ SPAN
   └─ "en cascada"

```

Así es como un navegador interpreta el código HTML anterior, interpreta el árbol DOM y luego lo muestra en el navegador, así:

Usaremos: Hojas de estilo en cascada

## La aplicación de CSS al DOM

Pongamos que hemos añadido un poco de CSS a nuestro documento, para darle estilo. Una vez más, el HTML es el siguiente:

HTML

Play

```
<p>
  Usaremos:
  <span>Hojas</span>
  <span>de estilo</span>
  <span>en cascada</span>
</p>
```

Supongamos que le aplicamos el CSS siguiente:

CSS

Play

```
span {
  border: 1px solid black;
  background-color: lime;
}
```

El navegador analizará el código HTML y creará un DOM a partir de este. A continuación, analizará el CSS. Dado que la única regla disponible en el CSS tiene un selector `span`, el navegador ordenará el CSS muy rápidamente! Aplicará la regla a cada uno de los tres `<span>`, que mostrarán en pantalla la representación visual final.

La salida actualizada es la siguiente:

Play

Usaremos: Hojas de estilo en cascada

En nuestro artículo [Depurar el CSS](#) que encontrarás en el siguiente módulo, vamos a utilizar las herramientas DevTools del navegador para depurar posibles problemas en el CSS. También aprenderemos más sobre cómo el navegador interpreta el CSS.

## ¿Qué ocurre si un navegador encuentra CSS que no entiende?

En una [lección anterior](#) mencionamos que no todos los navegadores implementan las novedades de CSS en el mismo momento. Además, no todo el mundo utiliza la última versión de un navegador. Dado que el CSS está en desarrollo constante y, por lo tanto, por delante de lo que los navegadores pueden reconocer, puede que te preguntes qué sucede si un navegador encuentra un selector o una declaración CSS que no reconoce.

La respuesta es que no hace nada y simplemente pasa a la siguiente parte del CSS.

Si un navegador analiza tus reglas y encuentra una propiedad o un valor que no entiende, lo ignora y avanza hasta la declaración siguiente. Esto sucederá si has cometido un error y has escrito mal una propiedad o un valor, o si la propiedad o el valor son demasiado nuevos y el navegador aún no los admite.

Del mismo modo, si un navegador encuentra un selector que no entiende, lo ignorará y pasará al siguiente.

En el siguiente ejemplo hemos utilizado la ortografía británica para la propiedad `color`, que invalida la propiedad porque no la reconoce. Así que el párrafo no se muestra en azul. Sin embargo, se han aplicado todos los demás estilos del CSS; solo se ha ignorado la línea que no es válida.

HTML

Play

```
<p>Quiero este texto en grande, en negrita y en color azul.</p>
```

CSS

Play

```
p {  
  font-weight: bold;  
  colour: blue; /* Ortografía incorrecta de la propiedad color */  
  font-size: 200%;  
}
```

Play

**Quiero este texto en grande, en negrita y en color azul.**

Este comportamiento es muy útil. Significa que puedes utilizar el CSS nuevo como una mejora, a sabiendas de que no se producirá ningún error si no se entiende: o bien el navegador entiende la característica nueva o no lo hace. Combinado con el funcionamiento del modo en cascada con el hecho de que los navegadores utilizarán la última CSS que encuentren en la hoja de estilo, cuando haya dos reglas con el mismo nivel de especificidad, también puedes ofrecer alternativas para los navegadores que no admiten el CSS nuevo.

Esto funciona especialmente bien cuando quieres utilizar un valor que es bastante nuevo que no admiten todos los navegadores. Por ejemplo, algunos navegadores antiguos no entienden `calc()` como valor. Podríamos dar un valor de sustitución para el ancho de una caja en píxeles, y a continuación dar un ancho con un valor `calc()` de `100% - 50px`. Los navegadores antiguos usarán la versión en píxeles y harán caso omiso de la indicación `calc()`, porque no la entienden. Los navegadores nuevos interpretarán la línea del ancho en píxeles, pero la anularán al llegar a la línea de `calc()` porque aparece después en la cascada.

CSS

Play

```
.box {  
  width: 500px;  
  width: calc(100% - 50px);  
}
```

En lecciones posteriores veremos muchas más formas de cómo admitir navegadores diferentes.

## Y finalmente

Casi has terminado este módulo; solo nos queda una cosa más por hacer. En el próximo artículo, [pondrás en práctica tu conocimiento nuevo](#) para cambiar el estilo de un ejemplo y probarte con un poco de CSS en el proceso.

### Help improve MDN

Was this page helpful to you?

[Learn how to contribute.](#)

This page was last modified on 21 dic 2024 by [MDN contributors](#).

