

Hojas de Estilo en Cascada CSS,

Frameworks CSS

Css

CSS viene de las iniciales de "Cascading Style Sheets", que traducimos por "Hojas de estilo en cascada". Es el segundo lenguaje más esencial para crear páginas web. El primero sería HTML, con el que se define el contenido de la página. El segundo CSS, con el que se define la parte de la presentación, es decir, cómo deben mostrarse los elementos de la página, su posición, forma, espaciados, colores y en resumen, toda la parte estética.

CSS no tiene ningún sentido sin HTML, igual que a día de hoy HTML sin CSS también resultaría imposible, ya que la web ha evolucionado de tal forma que su aspecto es fundamental y el CSS es la única vía para conseguir personalizarlo.



Partes principales del CSS

Para desarrollar con CSS tenemos que trabajar con una serie de elementos, mediante los cuales se declaran los **estilos**, básicamente éstos son los más importantes:

- **Selectores**, mediante los cuales podemos especificar qué elementos de la página nos estamos refiriendo
- **Atributos** de estilo para definir qué cosas queremos estilizar sobre los selectores indicados
- Una serie de **valores**, que indican qué estilo se debe aplicar a cada atributo sobre cada selector. Los valores se expresan con **unidades CSS**, que sirven para cuantificar los valores (píxeles, puntos...)

Aprender CSS no es difícil. Básicamente se trata de entender esos componentes principales y conocer las posibles variantes que existen para conseguir los estilos que necesitamos. Por supuesto, también tiene una buena parte de creatividad, pero la mayoría de las veces el diseño lo entregan previamente antes de desarrollar una web, por lo que en realidad lo que más requiere es un conocimiento técnico.

Cuando se usa profesionalmente CSS se deben tener en cuenta muchos detalles y buenas prácticas, como la organización del código, la reutilización, la optimización, etc. que básicamente están tratados en los manuales y artículos que encontrarás en Desarrollo Web.

Evolución del lenguaje CSS

Las hojas de estilo en cascada llegaron al mundo de la web bastante más tarde que el lenguaje HTML. Aunque fue propuesto en 1994 el primer estándar no llegaría hasta prácticamente entrado 1997. Sería CSS 1.

Obviamente, CSS 1 supuso un gran avance para el mundo del desarrollo de páginas web, pero lo cierto es que para entonces los diseñadores se habían acostumbrado a mezclar el contenido con la presentación, por medio del uso de etiquetas HTML que estaban presentes y seguían funcionando en los navegadores. Además CSS 1 había sido presentado con bastantes carencias, lo que hizo que se tuviera que presentar un nuevo estándar rápidamente. Así fue como sería liberado como recomendación CSS 2 pasado apenas un año, en 1998.

A partir de este punto el mundo de los estándares abiertos para la web tuvo un parón generalizado. No solo CSS, sino HTML y otros lenguajes como Javascript. CSS 2 tuvo una revisión publicada como CSS 2.1 en la que añadieron algunos selectores nuevos, pero pasaron años antes de la presentación de una nueva versión. Con todo entramos en una etapa oscura en la que CSS no llegó a cubrir totalmente sus objetivos.

- La compatibilidad de CSS no era la misma en todos los navegadores
- Los navegadores se dedicaban a incorporar funcionalidades por medio de prefijos, que hacían más difícil y tedioso escribir el CSS.
- El estándar no cubría todas las necesidades de los diseñadores

Con todo, los diseñadores y desarrolladores web tenían que usar los famosos "hacks de CSS", que no eran más que artimañas para conseguir los resultados deseados de maneras poco ortodoxas, que acababan en un código complejo y muy orientado a atender necesidades de navegadores determinados. Por poner un par de ejemplos, se tenía que usar capas anidadas a varios niveles para conseguir cosas como los bordes redondeados, Javascript para conseguir tamaños de fuentes adaptables, imágenes con transparencia en canal alfa (PNG) para conseguir sombras, etc. Al final conseguir un diseño medianamente atractivo y homogéneo a lo largo de los navegadores solo con CSS era muy complicado y se tenía que continuar variando el HTML o el Javascript para solucionar sus carencias

Hoy podemos decir que CSS cubre las necesidades de los diseñadores, más aún después de la aparición de CSS Flexbox, que permite agregar una facilidad enorme a la hora de colocar los elementos en la página. Junto por supuesto del último aporte a las especificaciones de CSS Grid System, que por fin ha aportado un completo y versátil sistema de rejilla a la web, mediante el cual los diseñadores web disponen de herramientas verdaderamente potentes para posicionar los elementos en la página, de manera independiente a como estos aparezcan en el código HTML.

Separar el código CSS del código HTML

El enfoque de CSS es servir para definir la capa de presentación, es decir, la parte relacionada con el aspecto. Es algo que cualquier estudiante suele tener claro cuando está aprendiendo CSS, ya que al enseñar HTML probablemente se haya insistido, pero que siempre conviene reforzar.

En el código HTML colocamos el contenido, es decir, qué debe visualizarse, mientras que con CSS definimos la presentación, es decir, cómo debe visualizarse. Esto nos lleva a una serie de usos de CSS que debemos de respetar como buenas prácticas.

- Lo adecuado cuando trabajamos con CSS es escribir el código en ficheros independientes, que tendrán extensión .css. Cada cosa en su lugar!
- No conviene colocar código CSS por tanto dentro de archivos HTML. Debemos evitar colocar estilos en etiquetas <style> dentro del propio código HTML.
- Por supuesto, mucho menos aconsejable es colocar estilos en los atributos "style" de las etiquetas HTML.

Selectores CSS

Los selectores en CSS nos permiten acceder a cualquier elemento o grupo de elementos, para aplicar estilos sobre el/ellos en una única declaración. Como su nombre indica, permiten seleccionar aquellos elementos sobre los que se van a aplicar las reglas de estilo.

Dentro del código CSS podemos usar selectores y aplicarles un conjunto de estilos determinado. Para ello escribimos el selector y colocamos los atributos de estilos encerrados entre llaves:

```
selector {  
  color: red;  
  width: auto;  
}
```

Existen selectores de lo más variado, que permiten ajustar de una manera muy precisa qué elementos se desea seleccionar. Los más importantes son los siguientes:

- **Etiqueta:** sirven para seleccionar todos los elementos de una misma etiqueta o tag HTML.

```
h1 {  
  font-size: 26px;  
}
```

- **Clase:** selecciona todos los elementos de una clase determinada. (Class de CSS).

```
.destacado {  
  font-weight: bold;  
  color: orange;  
}
```

- **Identificador:** permiten seleccionar etiquetas individuales por el atributo Id de la etiqueta.

```
#formularioAlta {  
  border: 1px solid #99c;  
}
```

- **Atributo:** permiten seleccionar todas las etiquetas que tengan un atributo dado, o bien un atributo con un valor determinado.

```
[title] {  
  text-decoration: none;  
}  
  
[align="center"] {  
  border: 1px solid red;  
}
```

Asociar un CSS a una página web

Para que una página de contenido gane un aspecto determinado, necesitaremos un medio para cargar o asociar el CSS dentro de un HTML. Esta operación se puede realizar a varios niveles, pero lo más habitual es enlazar el CSS por medio de la etiqueta <link>

```
<link rel="stylesheet" href="/css/app.css">
```

Esa etiqueta generalmente se pone en el <head>, para que el CSS se cargue desde el inicio y permite que la página HTML se presente con el estilo definido en el archivo CSS enlazado.

Generalmente esa misma etiqueta se colocará en todas las páginas del sitio web, permitiendo que todo el contenido de la web tenga el mismo aspecto definido en el CSS. Además, como el CSS se encuentra en un archivo externo, en todas las páginas se cargará la misma declaración de estilos, que generalmente se cacheará en el navegador, por lo que la segunda o siguientes páginas del sitio el CSS no se necesite descargar de nuevo.

Preprocesadores CSS

Los preprocesadores son herramientas indispensables para el desarrollo frontend actual. Permiten desarrollar CSS de una manera más ágil pero sobretodo crear un código más mantenible.

Para ello los preprocesadores extienden las características del lenguaje CSS, permitiendo uso de variables, funciones (mixins) diversas otras formas de acortar la escritura del código.

Los preprocesadores no se ejecutan en el entorno del navegador. En lugar de ello se procesan, compilando su código en código CSS estándar que si es capaz de entenderlo cualquier navegador. En ese proceso de compilación a menudo se incluye además una serie de mejoras en el código CSS como la minimización.

los preprocesadores más comunes son Less y [Sass](#). Siendo Sass el más extendido en la comunidad frontend con diferencia.

Frameworks CSS

Los frameworks CSS, a veces también llamados frameworks de diseño son herramientas que nos permiten implementar un marco de trabajo sólido sobre el que desarrollar la parte estética de los proyectos web. Ofrecen diversas ventajas, entre las que destacan dos principales:

- La agilidad para la generación de los estilos CSS y el aspecto estético de los sitios y aplicaciones web.
- La creación de un tema gráfico, que se puede mantener a lo largo de todo un sitio o aplicación, generando una mayor consistencia en el diseño.

Los frameworks CSS permiten por tanto un desarrollo optimizado, que además es muy útil en proyectos empresariales, donde sea necesario seguir una misma línea de diseño por varios desarrolladores.

Dentro del mundo de los frameworks CSS encontramos dos tipos principales:

Frameworks basados en componentes: que ofrecen una base de código CSS que permite crear fácilmente, por medio de clases, multitud de componentes como botones, listas, pestañas, menús desplegables, formularios estilizados, etc. Dentro de esta clasificación el framework más relevante es Bootstrap.

Frameworks basados en clases de utilidad: también llamados "utility first frameworks", que ofrecen una cantidad enorme de clases de utilidad atómicas, pensadas para aplicar un valor concreto a una propiedad determinada del CSS. Una clase de utilidad por si sola hace poco (podría ser simplemente un margen de 1 rem en la parte de abajo), pero mediante la combinación de diversas clases de utilidad se pueden genera todo tipo de interfaces de usuario totalmente personalizadas. Dentro de esta clasificación encontramos [Tailwind CSS](#).

La ventaja de los frameworks basados en componentes es que te dan mucho trabajo de estilos ya prefabricados, con temas de diseño muy definidos. Sin embargo, los frameworks basados en componentes en ocasiones también representan algunas desventajas, como mayor rigidez a la hora de desarrollar y dificultad para conseguir que el diseño de varios proyectos que usan el mismo framework se distingan unos de otros.

La ventaja de los frameworks basados en clases de utilidad es que el diseño que se consigue con ellos es totalmente único para cada sitio web, pero sin embargo la desventaja es que usan muchas clases para poder aplicar estilo a los elementos de la página.

