

CSS Grid

CSS Grid es un modelo de maquetación CSS en base a una rejilla, algo que podría no parecer tan novedoso, si tenemos en cuenta que diversas librerías de CSS lo habían intentado ya. Sistemas como 960 Grid System fueron pioneros en crear una base de código CSS para que los diseñadores pudieran posicionar los elementos en una distribución de filas y columnas

Como las últimas versiones de los navegadores ya ofrecen soporte para CSS Grid, sólo es necesario familiarizarse con su forma de trabajar de rejilla y aplicar el nuevo *display: grid* con el que comenzaremos a usar todas sus características para posicionar los elementos en la página:

- Ofrece un control total de las dos dimensiones.
- La ubicación de los ítems es libre; no es una tabla.
- Es posible colocar los ítem donde necesitemos, en cualquiera de las celdas que el Grid describe, incluso en celdas que no has descrito.
- Puede haber ítems que se coloquen solos gracias al su característica de *auto-placement*.
- También podemos conseguir comportamientos similares con una cantidad de atributos novedosas, aunque con distintos matices que las diferencian.

¿Qué hay de nuevo en CSS Grid?

El propio **Bootstrap** incluye entre otras cosas un sistema de rejilla. Al final, estos sistemas funcionaron, pero tenían varios problemas:

- Agregaban peso al CSS, con código de cientos de clases que muchas veces ni siquiera llegabas a usar.
- Había que aplicar constantemente clases y nombre que ensuciaban mucho el HTML.
- Creaba código de muy difícil mantenimiento, haciendo que el proyecto estuviera ligado de manera muy directa con un sistema propietario.

Sin embargo, ahora Grid es un estándar y no es necesario recargar ni ensuciar el código para aplicarlo. Todo se aplica directamente desde

el CSS, por medio de nuevos atributos, valores y unidades con una variada sintaxis. Además, un sitio con Grid resulta sencillo de mantener y de aplicar otras herramientas de CSS3 como las mediaqueries, y, por supuesto, es totalmente adaptable a los dispositivos (responsive).

Al ser una especificación estable, estándar y soportada por prácticamente todos los navegadores modernos (y si no, siempre podemos utilizar los polyfills) se considera realmente CSS, por lo que se puede utilizar en todos los sitios donde ya se utilice código CSS: sitios estáticos, programación con un lenguaje determinado, un framework o para cualquier tipo de dispositivos (escritorio, móviles, tablets) o webapps.

Grid vs Flexbox

Como veíamos en **¿Grid o Flexbox? ¿Qué resulta más útil en diseño y maquetación web?**, **Grid va un paso más allá de Flexbox y utiliza las dos dimensiones mientras que Flexbox aplica sólo una.** Con Flexbox sólo podías definir qué es lo que tenían que hacer los items en el eje horizontal o en el eje vertical. Cuando en el eje vertical u horizontal se terminaba el espacio, entonces los items se iban colocando dependiendo de la configuración de los atributos Flexbox. Por ejemplo, en el caso de usar *flex-wrap*, cuando llegaba un ítem que no cabía en la horizontal, se iba a la siguiente fila. Pero tú no controlas esas dos filas, los propios ítems son los que se van acomodando. **Te recomendamos consultar los tutoriales de Flexbox que hemos publicado recientemente.**

Sin embargo, CSS Grid nos permite trabajar con dos dimensiones y es posible decidir dónde ubicar las casillas, tanto en la horizontal como en la vertical, coordinando perfectamente las dimensiones de cada fila o columna, de manera global o perfectamente independiente. No necesita trucos, ni forzar que se queden sin sitio, **Grid realmente ofrece a los diseñadores web el control detallado de la posición y dimensiones de cada elemento en cada eje.**