

8. Flexbox vs CSS Grid

Tabla de contenidos.

- **8. Flexbox vs CSS Grid**
 - [8.1. Maquetación en flexbox vs CSS Grid](#)
 - [8.2. Características de Flexbox](#)
 - [8.3. Características de CSS Grid](#)

8.1. Maquetación en flexbox vs CSS Grid

[Flexbox](#) y [Grid](#) son dos mecanismos de CSS que resultan especialmente útiles de cara a la maquetación de páginas web. Comparamos sus principales funcionalidades y, en un primer vistazo, tenemos que reconocer que **Grid puede hacer cosas que Flexbox no puede hacer y Flexbox puede hacer cosas que Grid no puede hacer**. Con la ventaja de que ambos pueden trabajar juntos: un elemento de Grid puede ser un contenedor Flexbox y un elemento de Flexbox puede ser un contenedor de Grid.

Independientemente de qué sistema utilicemos maquetar usando FLEX o usando GRID es mucho más fácil que maquetar usando sistemas tradicionales basados únicamente en float, position, tablas etc..

Así pues, no hay razón para utilizar sólo Grid CSS o sólo Flexbox. **La mejor recomendación en aprender ambos y usarlos juntos.**

8.2. Características de Flexbox

En qué destaca Flexbox respecto a Grid

- Flexbox es la mejor opción para alinear el contenido dentro de los elementos.
- Utiliza Flexbox para posicionar los detalles más pequeños de un diseño.
- Flexbox funciona mejor sólo en una dimensión (filas o columnas).

Desventajas:

- Para el mismo layout que GRID necesito una estructura HTML más compleja

8.3. Características de CSS Grid

Ventajas de Grid respecto a Flex:

- Para diseños 2D (filas y columnas).
- CSS Grid es perfecto para construir una imagen más grande. Hace que sea realmente fácil de manejar el diseño de la página e incluso puede manejar diseños poco ortodoxos y asimétricos.
- Requiere menos intervención de consulta, lo que hace que sea realmente potente en funcionalidades del estilo auto layout: *minmax()*, *repeat()*, y *auto-fill*.

Desventajas:

- La alineación dentro de las celdas