

CSS Flexbox y Grid - Guía Teórica

Flexbox y Grid son dos sistemas de diseño en CSS que permiten organizar y distribuir elementos en el espacio disponible de forma flexible y eficiente. Cada uno tiene su enfoque y ventajas según el tipo de diseño que se desea lograr. Esta guía teórica explica en profundidad sus conceptos clave, comportamientos y casos de uso.

Parte 1: Flexbox

Qué es Flexbox

Flexbox (Flexible Box) es un modelo de diseño unidimensional, pensado para distribuir elementos en una sola dirección: fila o columna. Facilita la alineación, el espaciado y el reordenamiento de los elementos dentro de un contenedor.

Ejes en Flexbox

- Eje principal (main axis): dirección en la que los elementos se distribuyen (horizontal por defecto).
- Eje transversal (cross axis): perpendicular al eje principal.

Roles

- Contenedor Flex: el elemento que activa el modelo flex al declarar `display: flex` o `display: inline-flex`.
- Items Flex: los hijos directos del contenedor flex.

Propiedades del contenedor

- `flex-direction`: define la dirección del eje principal (`row`, `row-reverse`, `column`, `column-reverse`).
- `flex-wrap`: permite que los items pasen a otra línea si no hay espacio suficiente (`nowrap`, `wrap`, `wrap-reverse`).
- `justify-content`: alinea los items a lo largo del eje principal (`start`, `center`, `end`, `space-between`, `space-around`, `space-evenly`).
- `align-items`: alinea los items en el eje transversal (`stretch`, `start`, `end`, `center`, `baseline`).
- `align-content`: distribuye las líneas del contenido en el eje transversal cuando hay múltiples líneas.

Propiedades de los items

- `order`: cambia el orden visual de los items.
- `flex-grow`: determina cuánto puede crecer un item si hay espacio extra.
- `flex-shrink`: define cuánto puede reducirse un item si no hay suficiente espacio.
- `flex-basis`: valor inicial del tamaño del item antes de que se reparta el espacio.
- `align-self`: sobreescribe `align-items` para un item específico.

Casos de uso típicos

- Alinear elementos horizontal o verticalmente.
- Crear barras de navegación.
- Reordenar elementos sin cambiar el HTML.

- Espaciar items proporcionalmente.

Parte 2: Grid

Qué es Grid

Grid (CSS Grid Layout) es un modelo bidimensional, que permite diseñar tanto en filas como en columnas. Es ideal para diseños complejos de interfaz, como grillas, layouts de revistas, tableros y dashboards.

Conceptos básicos

- Grid container: se activa con `display: grid` o `inline-grid`.
- Grid items: hijos directos del contenedor grid.
- Grid lines: líneas que dividen las filas y columnas.
- Grid tracks: segmentos entre líneas (columnas y filas).
- Grid cells: celdas individuales formadas por una columna y una fila.
- Grid areas: regiones que pueden ocupar uno o varios items.

Definición de columnas y filas

- `grid-template-columns` / `grid-template-rows`: definen el número y tamaño de columnas/filas.
- Las unidades pueden ser absolutas (px), relativas (%), automáticas (auto) o flexibles (fr).

Unidad fr

- `fr` significa fracción del espacio disponible.
- Permite repartir espacio sobrante entre columnas/filas.
- Ej: `1fr 2fr` reparte el espacio en 3 partes: 1 para la primera columna, 2 para la segunda.

Otras funciones clave

- `repeat()`: repite un valor varias veces.
- `minmax(min, max)`: define un rango para una pista (track).
- `auto-fill` / `auto-fit`: rellena el espacio con el máximo de columnas posibles.

Posicionamiento

- `grid-column` / `grid-row`: definen el inicio y fin de una celda.
- `grid-area`: define un área por nombre o por líneas.
- `place-items` / `place-content` / `place-self`: combinan alineación vertical y horizontal.

Casos de uso típicos

- Galerías de imágenes.
- Dashboards.
- Distribución de contenido en filas y columnas.
- Diseños responsivos complejos.

Comparativa Flexbox vs Grid

Comparativa Flexbox vs Grid		
Característica	Flexbox	Grid
Dimensión	Unidimensional	Bidimensional
Flujo	Eje principal (fila o columna)	Filas y columnas
Control de alineación	Muy bueno en un eje	Excelente en ambos ejes
Uso recomendado	Componentes pequeños, alineación	Layouts completos, rejillas
Espaciado automático	Flexible con <code>justify-content</code>	Preciso con tracks y áreas
Complejidad	Más simple	Más potente pero más complejo

Conclusión

Flexbox y Grid no compiten entre sí: se complementan. Flexbox es excelente para estructuras lineales y alineaciones dentro de componentes, mientras que Grid brilla en el diseño general de la página y la organización de contenido complejo.

Dominar ambos sistemas te permite crear interfaces modernas, responsivas y mantenibles con facilidad.