

Flexbox

¿Qué es Flexbox?

Flexbox (o modelo de caja flexible) es un módulo de diseño en CSS3. Permite una distribución eficiente y alineación de elementos dentro de un contenedor, incluso cuando el tamaño de los elementos es desconocido o dinámico. Flexbox se centra en distribuir espacio a lo largo de un eje (horizontal o vertical) y en organizar elementos en relación entre sí dentro de un contenedor.

¿Para qué se usa?

Flexbox se usa para crear diseños de páginas web más flexibles y adaptables. Facilita la alineación de elementos y la distribución del espacio, especialmente cuando se trata de diferentes tamaños de pantalla y dispositivos. Es ideal para diseños unidimensionales, es decir, cuando necesitas controlar la disposición de los elementos en una sola dirección (fila o columna).

Propiedades principales de Flexbox

1. Propiedades del contenedor:

- display: flex; - Define un contenedor flex.
- flex-direction - Define la dirección en la que los elementos flexibles se colocan en el contenedor. Valores: row, row-reverse, column, column-reverse.
- flex-wrap - Controla si los elementos flexibles deben o no envolver al llegar al final del contenedor. Valores: nowrap, wrap, wrap-reverse.
- justify-content - Alinea los elementos flexibles a lo largo del eje principal. Valores: flex-start, flex-end, center, space-between, space-around, space-evenly.
- align-items - Alinea los elementos flexibles a lo largo del eje secundario. Valores: flex-start, flex-end, center, baseline, stretch.
- align-content - Alinea las líneas flexibles cuando hay espacio extra en el eje transversal. Valores: flex-start, flex-end, center, space-between, space-around, stretch.

2. Propiedades de los elementos flexibles:

- order - Define el orden de los elementos flexibles dentro del contenedor.
- flex-grow - Define la capacidad de un elemento para crecer en relación a los otros elementos flexibles.
- flex-shrink - Define la capacidad de un elemento para encogerse en relación a los otros elementos flexibles.
- flex-basis - Define el tamaño base del elemento flexible antes de que el espacio restante sea distribuido.
- align-self - Permite la alineación de un elemento en el eje secundario independientemente de otros elementos.

Casos de uso comunes

1. Distribución de Navegación: Flexbox es ideal para crear menús de navegación donde los elementos deben estar igualmente espaciados o alineados a un borde.

2. **Centrado de Elementos:** Flexbox facilita centrar elementos vertical y horizontalmente en su contenedor, algo que puede ser complicado con otros métodos de CSS.
3. **Diseños Responsivos:** Permite que los elementos se reorganicen y cambien de tamaño de manera flexible para adaptarse a diferentes tamaños de pantalla.
4. **Cards Layouts:** Crear layouts de tarjetas donde las tarjetas deben tener el mismo ancho y distribuirse uniformemente es simple con Flexbox.
5. **Footers Pegajosos:** Permite que un footer se quede pegado al fondo de la página sin importar la cantidad de contenido en el cuerpo.

CSS Grid

¿Qué es CSS Grid?

CSS Grid es un sistema de diseño en CSS que permite crear layouts complejos de manera sencilla y eficiente. A diferencia de Flexbox, que está diseñado para manejar una sola dimensión (horizontal o vertical), CSS Grid maneja ambas dimensiones simultáneamente (filas y columnas), lo que lo hace ideal para diseños bidimensionales.

Propiedades Principales de CSS Grid

1. **Propiedades del contenedor:**
 - `display: grid;` - Define un contenedor grid.
 - `grid-template-columns` - Define el número y el tamaño de las columnas en una cuadrícula.
 - `grid-template-rows` - Define el número y el tamaño de las filas en una cuadrícula.
 - `grid-gap` o `gap` - Define el espacio entre las filas y columnas de la cuadrícula.
 - `justify-items` - Alinea los elementos dentro de sus áreas de celda a lo largo del eje horizontal. Valores: `start`, `end`, `center`, `stretch`.
 - `align-items` - Alinea los elementos dentro de sus áreas de celda a lo largo del eje vertical. Valores: `start`, `end`, `center`, `stretch`.
 - `justify-content` - Alinea la cuadrícula completa dentro del contenedor en el eje horizontal.
 - `align-content` - Alinea la cuadrícula completa dentro del contenedor en el eje vertical.
2. **Propiedades de los elementos grid:**
 - `grid-column` - Define en qué columna comienza un elemento y cuántas columnas ocupa.
 - `grid-row` - Define en qué fila comienza un elemento y cuántas filas ocupa.
 - `grid-area` - Define el área que ocupa un elemento en términos de líneas de inicio y fin para filas y columnas.

Casos de Uso Comunes

1. **Layouts complejos:** Crear layouts de página complejos con múltiples filas y columnas.
2. **Tablas de Precios:** Diseñar tablas de precios o de comparación.
3. **Dashboards:** Crear dashboards con widgets que tienen tamaños variables.

4. Galerías de Imágenes: Organizar imágenes en una galería de manera eficiente.

¿Cómo funciona grid-template-columns?

La propiedad grid-template-columns define el número y el tamaño de las columnas en una cuadrícula. Puede aceptar unidades de longitud (como px, %, em, rem) y funciones CSS específicas como repeat, minmax, auto, etc.

```
css
.container {
  display: grid;
  grid-template-columns: 200px 1fr 2fr;
}
```

En este ejemplo, la cuadrícula tendrá tres columnas: la primera de 200px, la segunda de tamaño flexible ocupando el espacio restante, y la tercera el doble de tamaño que la segunda.

¿Por qué usar CSS Grid en lugar de Flexbox?

Flexbox es ideal para:

- Diseños unidimensionales.
- Alinear elementos en una sola dirección (horizontal o vertical).
- Componentes pequeños y simples como menús de navegación o botones alineados.

CSS Grid es ideal para:

- Diseños bidimensionales.
- Crear layouts de página completos con filas y columnas.
- Organizar elementos que necesitan tanto filas como columnas.

Similitudes entre CSS Grid y Flexbox

- Ambos son sistemas de diseño en CSS.
- Ambos permiten la alineación y distribución de espacio entre elementos.
- Ambos pueden anidarse (usar Flexbox dentro de una cuadrícula y viceversa).

Diferencias entre CSS Grid y Flexbox

- Dimensión: Flexbox es unidimensional (maneja filas o columnas), mientras que Grid es bidimensional (maneja filas y columnas simultáneamente).
- Complejidad del Layout: Flexbox es mejor para componentes individuales o simples, Grid es mejor para layouts complejos de página.
- Funcionalidad: Grid tiene más propiedades para controlar tanto filas como columnas, mientras que Flexbox se enfoca en una sola dirección.

En pocas palabras CSS Grid es una herramienta poderosa para crear layouts complejos de manera eficiente. Sus propiedades permiten controlar tanto filas como columnas, lo que lo hace ideal para diseños bidimensionales. Flexbox, por otro lado, es excelente para alinear y distribuir elementos en una sola dirección. Ambos sistemas pueden trabajar juntos, aprovechando lo mejor de cada uno para diferentes partes de un diseño web.