Flexbox y Grid

Layout Moderno en CSS

📝 2026-01 — ⏱️ 30 min.

Flexbox

Flexbox - Introducción

Flexbox es un modelo de layout unidimensional (fila o columna).

Perfecto para **alinear elementos** y distribuir espacio.

```
.container {
    display: flex;
}
```

El contenedor se convierte en **flex container**. Los hijos directos se convierten en **flex items**.

Flexbox - Propiedades del Container

```css
.container {
    display: flex;

    /* Dirección */
    flex-direction: row | row-reverse | column | column-reverse;

    /* Justificación (eje principal) */
    justify-content: flex-start | center | flex-end | space-between | space-around;

    /* Alineación (eje cruzado) */
    align-items: flex-start | center | flex-end | stretch;

    /* Permitir salto de línea */
    flex-wrap: nowrap | wrap | wrap-reverse;

    /* Espacio entre líneas */
    align-content: flex-start | center | flex-end | space-between;
}
```

Flexbox - Propiedades de Items

```css
.item {
    /* Crecer para llenar espacio */
    flex-grow: 1;

    /* Encogerse si es necesario */
    flex-shrink: 1;

    /* Tamaño base */
    flex-basis: 200px;

    /* Forma corta */
    flex: 1 1 200px; /* grow shrink basis */

    /* Alineación individual */
    align-self: flex-start | center | flex-end;

    /* Orden */
    order: 2;
}
```

Flexbox - Ejemplos Comunes

Centrar elemento:

```
.container {
    display: flex;
    justify-content: center;
    align-items: center;
}
```

Distribución uniforme:

```
.container {
    display: flex;
    justify-content: space-between;
}
```

Items de igual tamaño:

```
.item {
    flex: 1;
}
```

CSS Grid

Grid - Introducción

Grid es un modelo de layout bidimensional (filas y columnas).

Perfecto para layouts complejos y estructuras de página.

```
.container {
    display: grid;
}
```

El contenedor se convierte en **grid container**. Los hijos directos se convierten en **grid items**.

Grid - Definir Filas y Columnas

```css
.container {
    display: grid;

    /* 3 columnas de 200px cada una */
    grid-template-columns: 200px 200px 200px;

    /* 2 filas de 100px cada una */
    grid-template-rows: 100px 100px;

    /* Usando repeat() */
    grid-template-columns: repeat(3, 200px);

    /* Fracciones (fr) */
    grid-template-columns: 1fr 2fr 1fr;

    /* Auto */
    grid-template-columns: auto 200px auto;
}
```

Grid - Gap (Espaciado)

```css
.container {
    display: grid;
    grid-template-columns: repeat(3, 1fr);

    /* Espacio entre columnas */
    column-gap: 20px;

    /* Espacio entre filas */
    row-gap: 20px;

    /* Forma corta (fila, columna) */
    gap: 20px 30px;

    /* Mismo gap para ambos */
    gap: 20px;
}
```

Grid - Colocar Items

```css
.item {
    /* Empezar en columna 1, terminar en columna 3 */
    grid-column: 1 / 3;

    /* Empezar en fila 1, terminar en fila 3 */
    grid-row: 1 / 3;

    /* Forma corta */
    grid-column: 1 / span 2; /* Ocupar 2 columnas */
    grid-row: 1 / span 2;    /* Ocupar 2 filas */
}

.full-width {
    grid-column: 1 / -1; /* Toda la fila */
}
```

Grid - Template Areas

```css
.container {
    display: grid;
    grid-template-areas:
        "header header header"
        "sidebar main main"
        "footer footer footer";
    grid-template-columns: 200px 1fr 1fr;
}

.header  { grid-area: header; }
.sidebar { grid-area: sidebar; }
.main    { grid-area: main; }
.footer  { grid-area: footer; }
```

Forma visual e intuitiva de definir layouts.

Grid - Alineación

```css
.container {
    /* Alinear items horizontalmente */
    justify-items: start | center | end | stretch;

    /* Alinear items verticalmente */
    align-items: start | center | end | stretch;

    /* Alinear todo el grid horizontalmente */
    justify-content: start | center | end | space-between;

    /* Alinear todo el grid verticalmente */
    align-content: start | center | end | space-between;
}
```

Flexbox vs Grid

¿Cuándo usar cada uno?

Flexbox (1D):

- ✅ Una dirección (fila o columna)
- ✅ Distribución de espacio
- ✅ Alineación de items
- ✅ Navegación, botones
- ✅ Componentes pequeños

Grid (2D):

- ✅ Filas y columnas simultáneas
- ✅ Layouts complejos
- ✅ Estructuras de página
- ✅ Galerías, dashboards
- ✅ Diseños precisos

Combinar Flexbox y Grid

Ambos se pueden combinar:

```
/* Grid para el layout general */
.page {
    display: grid;
    grid-template-areas:
        "header header"
        "sidebar main"
        "footer footer";
}

/* Flexbox dentro del header */
.header {
    display: flex;
    justify-content: space-between;
    align-items: center;
}
```

Grid para la estructura principal, Flexbox para componentes internos.

Recursos de Práctica

Práctica Interactiva:

- Flexbox Froggy: https://flexboxfroggy.com
- Grid Garden: https://cssgridgarden.com
- Flexbox Defense: http://www.flexboxdefense.com

Documentación:

- MDN Flexbox: https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout
- MDN Grid: https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout

Cheat Sheets:

- CSS Tricks Flexbox Guide: https://css-tricks.com/snippets/css/a-guide-to-flexbox
- CSS Tricks Grid Guide: https://css-tricks.com/snippets/css/complete-guide-grid