



INTRODUCCIÓN

CSS GRID

Made by Valentino



Qué es CSS Grid?

CSS Grid es un sistema de diseño en CSS que permite crear layouts complejos en una cuadrícula bidimensional. A diferencia de otros métodos de diseño, como Flexbox, que funcionan mejor en una dimensión (fila o columna), CSS Grid facilita la organización de elementos tanto en filas como en columnas simultáneamente. Es ideal para crear diseños de página responsivos y flexibles, permitiendo definir áreas, distribuir espacios, y controlar el tamaño y el posicionamiento de los elementos en relación a la cuadrícula.



Diferencias con Flexbox

La principal diferencia entre CSS Grid y Flexbox es la dirección de diseño: CSS Grid es bidimensional, permitiendo organizar elementos en filas y columnas a la vez, mientras que Flexbox es unidimensional, diseñado para organizar elementos en una sola dirección (fila o columna). CSS Grid es ideal para estructuras de diseño complejas y de página completa, mientras que Flexbox se adapta mejor a la alineación y distribución de elementos en una sola línea, como menús o barras de navegación. Además, CSS Grid permite definir áreas explícitas de diseño, lo que facilita la creación de layouts específicos de manera más intuitiva.

```
CS
{
  flex;
  2px solid
  100px;
  300px;
```

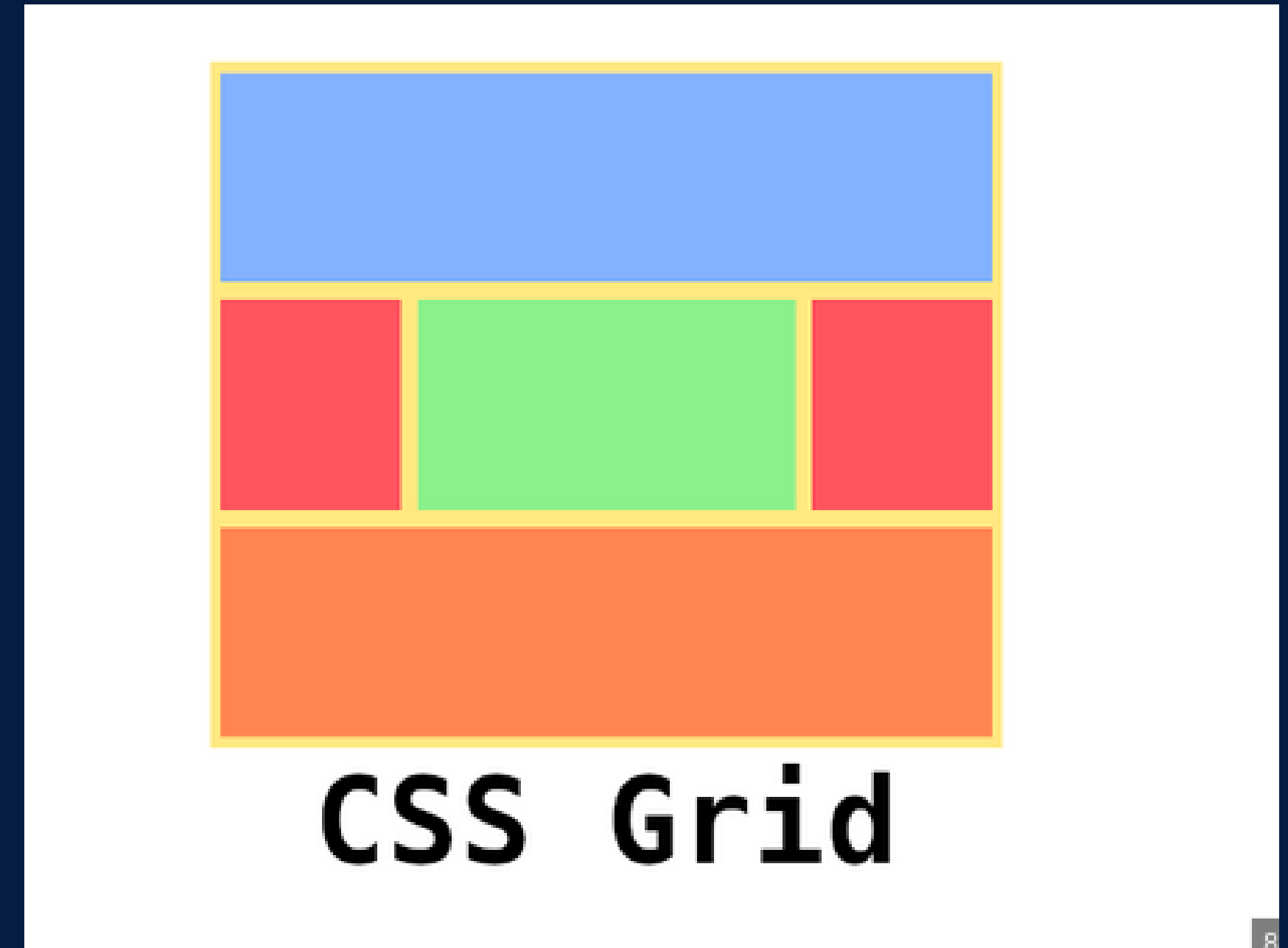
Ventajas y casos de uso

Ventajas de CSS Grid:

- Permite diseños complejos en dos dimensiones (filas y columnas).
- Facilita crear layouts responsivos mediante ajustes automáticos y personalizados en cada área de la cuadrícula.
- Proporciona un control preciso sobre la posición y el tamaño de los elementos.
- Define áreas de diseño claras, lo que mejora la organización visual y la legibilidad del código.

Casos de uso:


- Layouts de página complejos, como galerías de imágenes, blogs y páginas de inicio.
- Diseños responsivos que se adaptan a diferentes tamaños de pantalla.
- Distribución de contenido en secciones, encabezados, barras laterales y pies de página en aplicaciones web.



Grid Container y Grid Items

- Grid Container: Es el elemento que activa la cuadrícula cuando se le asigna la propiedad `display: grid`. Dentro del contenedor se define la estructura de la cuadrícula (filas, columnas, áreas) y el espacio entre ellas.
- Grid Items: Son los elementos directos dentro del Grid Container. Estos elementos se colocan en la cuadrícula y pueden posicionarse y dimensionarse de acuerdo con las reglas del contenedor, ocupando uno o más espacios en filas y columnas.

Content 1

The background of the slide features a light gray grid pattern on a white background, which is partially obscured by a dark blue diagonal shape. In the bottom-left corner, there is a blue geometric shape consisting of a square and a triangle. The text is white and positioned on the dark blue background.

Grid Lines: Son las líneas divisorias que separan las filas y las columnas en la cuadrícula. Se pueden referenciar para posicionar los elementos utilizando índices, como `grid-row` y `grid-column`.

Tracks: Son las filas y columnas del Grid, representando el espacio entre dos líneas consecutivas. Una "track" puede ser una fila (`grid-template-rows`) o una columna (`grid-template-columns`).

Areas: Son las regiones definidas dentro del Grid, que agrupan varias filas y columnas. Los elementos pueden ocupar estas áreas usando la propiedad `grid-area`, lo que facilita el diseño de layouts específicos y bien organizados.

- Grid Gaps: Son los espacios entre las filas y columnas en la cuadrícula. Se definen con `grid-gap`, `grid-row-gap` o `grid-column-gap` para ajustar la separación entre los elementos sin afectar su tamaño o disposición.
- Unidades FR: La unidad `fr` (fracción) distribuye el espacio disponible de manera proporcional entre las columnas o filas. Por ejemplo, `1fr` ocupa una fracción del espacio disponible, y `2fr` ocupa el doble de espacio que `1fr`.
- `auto`: La unidad `auto` permite que un elemento ajuste su tamaño automáticamente según su contenido. En Grid, se usa comúnmente para las filas o columnas cuando no se define un tamaño fijo, permitiendo que el espacio se ajuste de acuerdo con el contenido del elemento.

PROPIEDADES PRINCIPALES

Las propiedades `grid-template-columns` y `grid-template-rows` en CSS Grid definen el tamaño y la estructura de las columnas y filas de un Grid Container.

- `grid-template-columns`: Establece el número y el tamaño de las columnas en el contenedor. Se pueden usar valores fijos, porcentajes, unidades `fr`, o `auto` para ajustar el tamaño. Ejemplo: `grid-template-columns: 1fr 2fr 1fr;`
- `grid-template-rows`: Establece el número y el tamaño de las filas en el contenedor, utilizando las mismas unidades que en las columnas. Ejemplo: `grid-template-rows: 100px auto 200px;`

Ambas propiedades permiten crear una cuadrícula flexible y responsiva.

GRID-AUTO-COLUMNS/ROWS

Las propiedades `grid-auto-columns` y `grid-auto-rows` en CSS Grid definen el tamaño automático de las columnas y filas que se crean en el Grid Container cuando los elementos exceden la cantidad definida por `grid-template-columns` o `grid-template-rows`.

- `grid-auto-columns`: Establece el tamaño de las columnas que se generarán automáticamente si hay más elementos de los que caben en las columnas definidas. Ejemplo: `grid-auto-columns: 200px;`
- `grid-auto-rows`: Establece el tamaño de las filas que se generarán automáticamente si hay más elementos de los que caben en las filas definidas. Ejemplo: `grid-auto-rows: 100px;`

Ambas propiedades permiten manejar el crecimiento dinámico del layout cuando el contenido se expande más allá de las áreas definidas.

3

5

4

6

PROPIEDADES PRINCIPALES

`grid-column-start` / `grid-column-end`

Definen en qué línea de la cuadrícula comienza y termina un elemento en la dirección horizontal (columnas). Por ejemplo, `grid-column-start: 1`; `grid-column-end: 3`; hace que el elemento ocupe desde la línea 1 hasta la línea 3 en el eje horizontal.

`grid-template-areas`

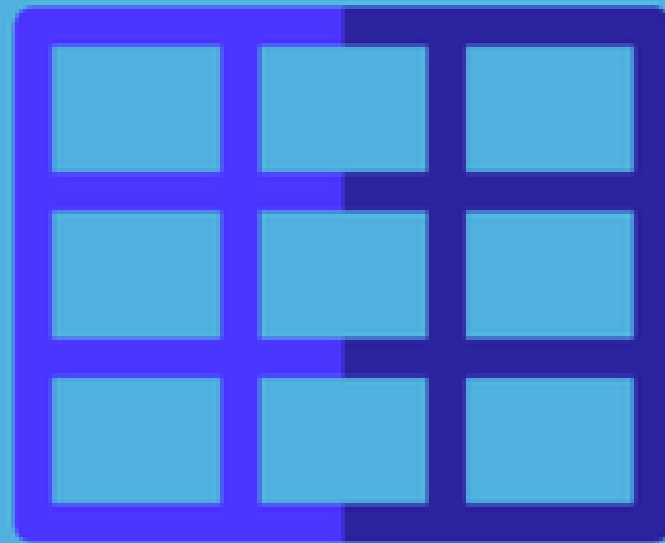
Permite crear áreas de diseño mediante nombres. Define un mapa visual para colocar los elementos en áreas específicas de la cuadrícula.

`grid-area`

Asigna un nombre de área a un Grid Item o especifica su posición usando `grid-row-start`, `grid-column-start`, `grid-row-end`, y `grid-column-end` en una sola propiedad.

Auto-fill y auto-fit

- auto-fill: Llena el espacio disponible con tantas columnas o filas como sea posible, incluso si hay espacio vacío extra.
- auto-fit: Ajusta automáticamente el número de columnas o filas para llenar el espacio disponible, eliminando las columnas o filas vacías.



9. Grid Auto-fit y Auto-fill

GRID ANIDADOS

Los Grid anidados ocurren cuando un elemento dentro de un Grid Item actúa como un Grid Container. Esto permite crear estructuras de cuadrícula dentro de otra cuadrícula, lo que facilita diseños más complejos y flexibles.

- Un Grid Container puede contener Grid Items que, a su vez, son Grid Containers.
- Cada Grid Container anidado puede tener su propia definición de filas y columnas, y los elementos dentro de él se colocan según esa cuadrícula interna.