

# Propiedades de Grid

Ahora, veamos cuáles son las propiedades de **Grid**, sus posibles valores y lo que hace cada una de ellas:

**display:** Especifica el elemento como un contenedor de cuadrícula o subcuadrícula.

- **grid:** crea un contenedor de cuadrícula
- **inline-grid:** crea un contenedor de subcuadrícula en línea

**grid-template-columns:** Define el número y tamaño de columnas de la cuadrícula

- **<track-size>:** define el tamaño de cada columna, por ejemplo: 100px, 20%, auto, etc.
- **repeat():** define un patrón repetitivo de columnas, por ejemplo: `repeat(3, 1fr)` creará tres columnas de igual tamaño

**grid-template-rows:** define el número y tamaño de filas de la cuadrícula

- **<track-size>:** define el tamaño de cada fila, por ejemplo: 100px, 20%, auto, etc.
- **repeat():** define un patrón repetitivo de filas, por ejemplo: `repeat(3, 1fr)` creará tres filas de igual tamaño

**grid-template-areas:** define áreas nombradas en la cuadrícula

- **<grid-area-name>:** define un nombre para una celda o área
- **“.”:** indica una celda vacía

**grid-area:** Asigna un elemento a una ubicación en la cuadrícula definida por las propiedades `grid-row` y `grid-column`. También se puede usar para asignar un nombre a un área definida en la propiedad `grid-template-areas`.

**grid-template:** abreviatura de `grid-template-rows`, `grid-template-columns`, y `grid-template-areas`.

**grid-column-gap:** Define el espacio entre columnas de la cuadrícula.

- **<length>:** Define el tamaño en píxeles, por ejemplo: 10px, 1rem, etc.
- **normal:** Valor predeterminado, define un tamaño estándar.

**grid-row-gap:** define el espacio entre filas de la cuadrícula.

- **<length>**: Define el tamaño en píxeles, por ejemplo: 10px, 1rem, etc.
- **normal**: Valor predeterminado, define un tamaño estándar.

**grid-gap**: Abreviatura de grid-row-gap y grid-column-gap.

**grid-auto-columns**: Define el tamaño predeterminado de columnas que no se han especificado en grid-template-columns.

- **<track-size>**: Define el tamaño de cada columna, por ejemplo: 100px, 20%, auto, etc.
- **max-content**: Define el tamaño máximo posible basado en el contenido de la celda.
- **min-content**: Define el tamaño mínimo posible basado en el contenido de la celda.
- **auto**: Valor predeterminado, ajusta automáticamente el tamaño de la columna.

**grid-auto-rows**: Define el tamaño predeterminado de filas que no se han especificado en grid-template-rows.

- **<track-size>**: Define el tamaño de cada fila, por ejemplo: 100px, 20%, auto, etc.
- **max-content**: Define el tamaño máximo posible basado en el contenido de la celda.
- **min-content**: Define el tamaño mínimo posible basado en el contenido de la celda.
- **auto**: Valor predeterminado, ajusta automáticamente el tamaño de la fila.

**grid-auto-flow**: Define cómo se distribuyen los elementos en la cuadrícula cuando no hay suficiente espacio en las filas o columnas especificadas. Puede ser row, column, dense, o row dense o column dense.

**grid-column**: Especifica la ubicación de un elemento a través de su posición en las columnas de la cuadrícula. Puede ser una sola línea, un rango de líneas o un nombre de área.

**grid-column-start**: Define la ubicación inicial de un elemento a lo largo del eje horizontal de la cuadrícula.

**grid-column-end**: Define la ubicación final de un elemento a lo largo del eje horizontal de la cuadrícula.

**grid-row**: Especifica la ubicación de un elemento a través de su posición en las filas de la cuadrícula. Puede ser una sola línea, un rango de líneas o un nombre de área.

**grid-row-start**: Define la ubicación inicial de un elemento a lo largo del eje vertical de la cuadrícula.

**grid-row-end**: Define la ubicación final de un elemento a lo largo del eje vertical de la cuadrícula.

## Ejemplo:

### Crear una cuadrícula

Lo primero que queremos hacer es crear una rejilla contenedora. Podemos hacerlo declarando `display: grid` en el elemento contenedor. En este ejemplo estamos utilizando un `div` con la clase de contenedor.

### Definir filas y columnas

Hay varias formas de definir filas y columnas. Para nuestra primera grid, utilizaremos las propiedades `grid-template-columns` y `grid-template-rows`. Estas propiedades nos permiten definir el tamaño de las filas y columnas de nuestra rejilla. Para crear dos filas de altura fija de 150px y tres columnas de ancho fijo de 150px, simplemente escriba:

```
grid-template-columns: 150px 150px 150px;  
grid-template-rows: 150px 150px;
```

Para añadir una cuarta columna de 70px de ancho, escribe:

```
grid-template-columns: 150px 150px 150px 70px;
```

..y así sucesivamente para añadir más columnas.

### Añadir un medianil

Añadir un medianil a tu cuadrícula es increíblemente fácil con CSS Grid Layout. Simplemente añade:

```
grid-gap: 1rem;
```

Esa simple línea te da un medianil del mismo tamaño entre todas las filas y columnas. Para definir el tamaño del medianil para columnas y filas individualmente, puedes usar

las propiedades `grid-column-gap` y `grid-row-gap`.

Ahora vamos a ponerlo todo junto. Este es nuestro HTML:

```
<div class="container">
  <div class="item item1"></div>
  <div class="item item2"></div>
  <div class="item item3"></div>
  <div class="item item4"></div>
  <div class="item item5"></div>
  <div class="item item6"></div>
</div>
```

Con sólo unas pocas líneas de CSS, podemos crear una cuadrícula simple:

```
.container {
  display: grid;
  grid-template-columns: 150px 150px 150px;
  grid-template-rows: 150px 150px;
  grid-gap: 1rem;
}
```

Este es el resultado:

