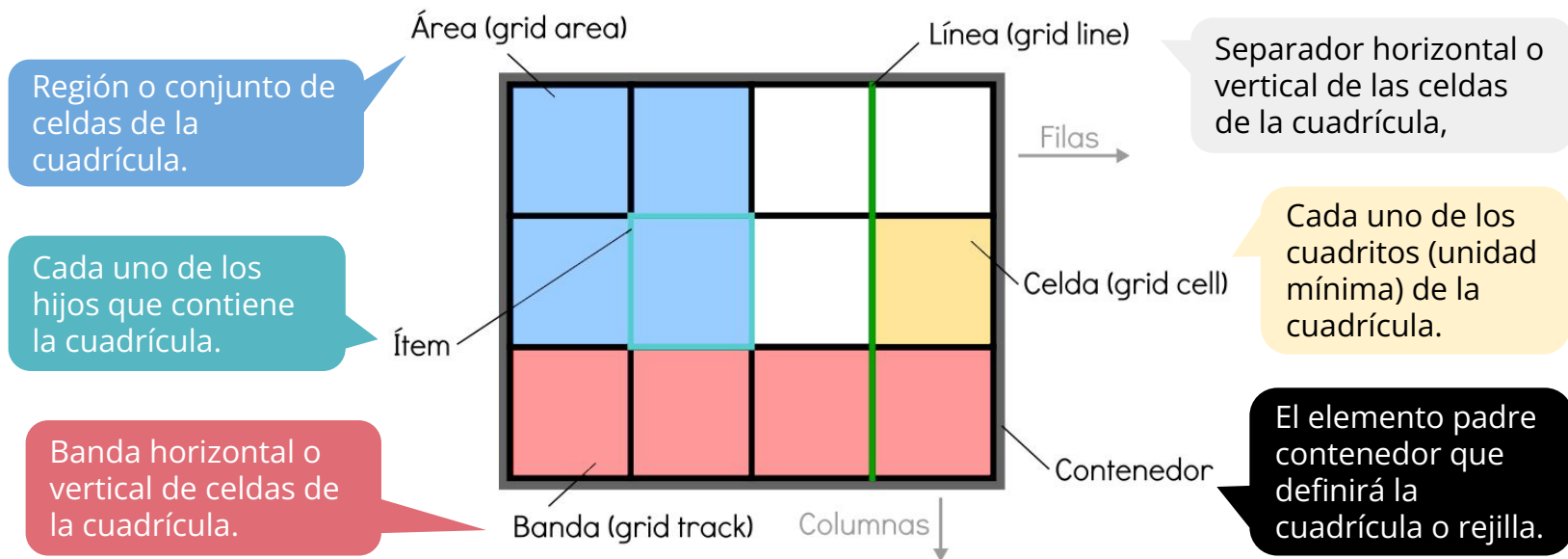


Grid

Elementos de CSS grid

Para utilizar **CSS grid** necesitaremos tener en cuenta una serie de conceptos que usaremos a partir de ahora y que definiremos a continuación:



display: grid | inline-grid

Para activar la cuadrícula **grid** hay que utilizar sobre el elemento **contenedor** la propiedad **display** y especificar el valor **grid** o **inline-grid**.

inline-grid: Establece una cuadrícula con **ítems** en línea, de forma equivalente a *inline-block*.

grid: Establece una cuadrícula con ítems en **bloque**, de forma equivalente a *block*.

css

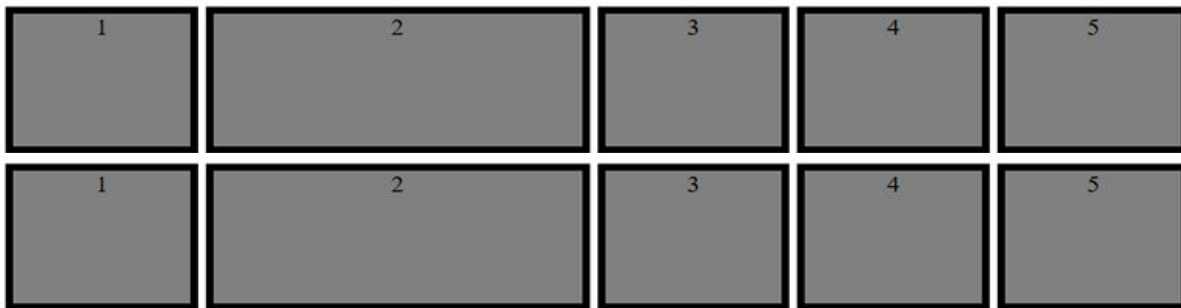
```
.grid { display: grid; }
```

css

```
.grid {  
    display: inline-grid;  
}
```

Armar la rejilla con columnas, filas y espacios

Es posible crear cuadrículas con un tamaño explícito usando las propiedades CSS `grid-template-columns` y `grid-template-rows`, que sirven para indicar las dimensiones de cada celda de la cuadrícula.



Con el siguiente código tendremos una cuadrícula como la anterior:

grid-template-rows

Establece el tamaño de cada fila.

grid-template-columns

Establece el tamaño de cada columna.

CSS

```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 2fr 1fr 1fr 1fr;  
  grid-template-rows: 100px 100px;  
  grid-gap: 5px;  
}
```



Trabajar con tamaños relativos, en lugar de especificaciones estáticas, permite que la cuadrícula pueda adaptarse automáticamente al tamaño de la pantalla. La segunda columna (en nuestro ejemplo) siempre será dos veces más grande que las otras.

1fr es una medida sobre el espacio disponible (free space).

grid-gap

Crea un espacio vacío

Se pueden usar unidades absolutas

CSS

```
grid-template-columns: 100px 50px 50px 200px;  
grid-template-rows: 50px 100px 50px 100px;
```

Uso de `repeat()`

La expresión resultante sería: `repeat([número de veces], [valor o valores])`

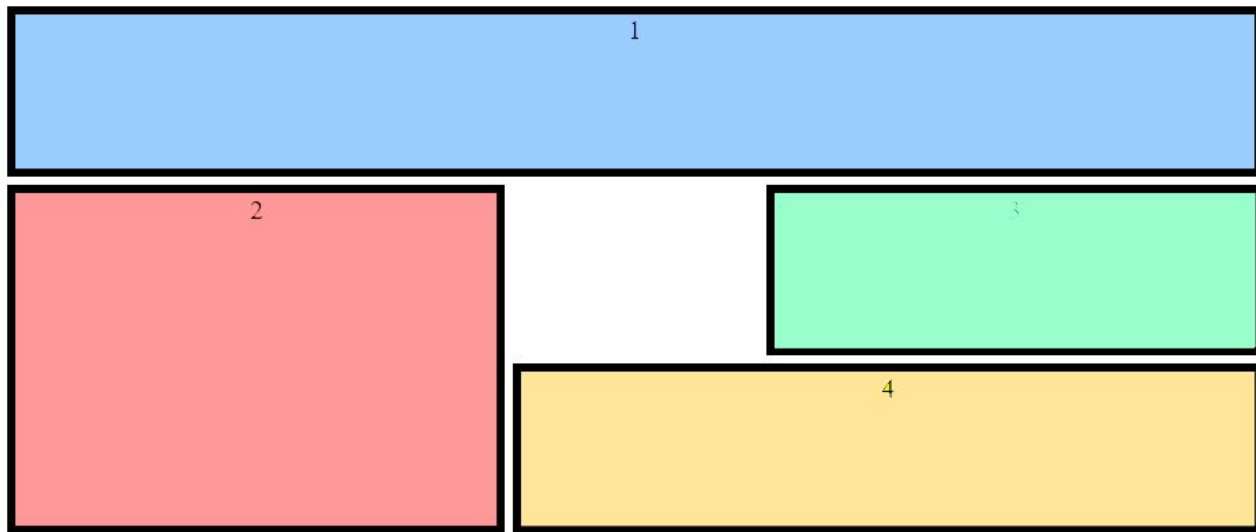
CSS

```
grid-template-columns: 100px repeat(2, 50px) 200px;  
grid-template-rows: repeat(2, 50px 100px);
```

Asignar áreas

CSS grid layout permite combinar **celdas** en **áreas** y nombrarlas. Esto facilita la tarea de dividir los elementos en la rejilla.

Los ajustes para esto se hacen en el contenedor con **grid-template-areas**.

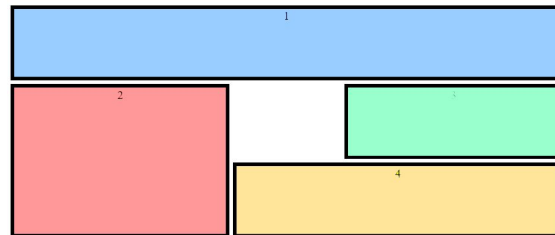


Código de gráfico anterior:

En este ejemplo, usamos el comando **grid-template-areas** y escribimos los nombres de las áreas deseadas en las **celdas** *línea por línea*.

Si **no queremos** asignar una celda y dejarla en blanco, insertamos un **punto (.)** en esta ubicación.

Cada fila queda entrecomillada.



CSS

```
.grid-container {  
  display: grid;  
  grid-template-rows: 100px 100px 100px;  
  grid-template-columns: 1fr 1fr 1fr 1fr 1fr;  
  grid-gap: 5px;  
  grid-template-areas:  
    "area1 area1 area1 area1 area1"  
    "area2 area2 . area3 area3"  
    "area2 area2 area4 area4 area4";  
}
```


Posiciones:

Veremos a continuación como utilizar distintas propiedades para alinear los **ítems** dentro de la **cuadrícula** tanto **individual** como **grupalmente**

align-self

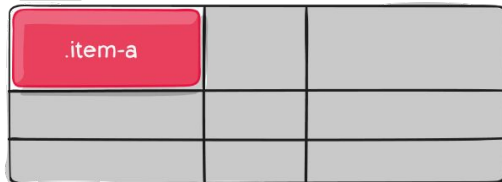
Con **align-self** podemos alinear un **elemento** de la **cuadrícula** dentro de una **celda** a lo largo del **eje del bloque** (columna).

Este valor se aplica al contenido dentro de **un solo elemento** de **cuadrícula**.

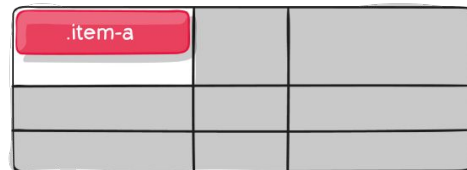
Valores:

- **start**
- **stretch**
- **end**
- **center**

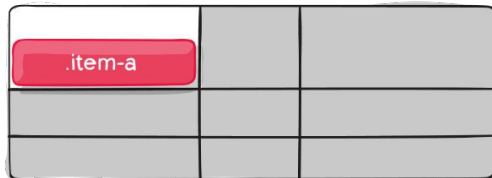
align-self: stretch;



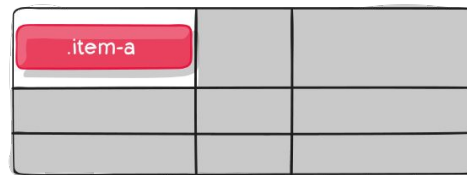
align-self: start;



align-self: end;



align-self: center;



justify-self

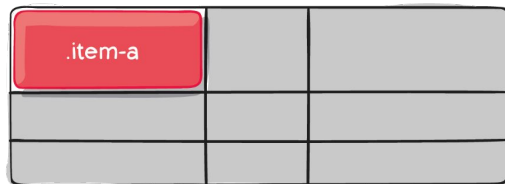
Con **justify-self** alineamos un **elemento** de la **cuadrícula** dentro de una **celda** a lo largo del **eje en línea** (fila).

Este valor se aplica a un elemento de la cuadrícula dentro de una sola celda.

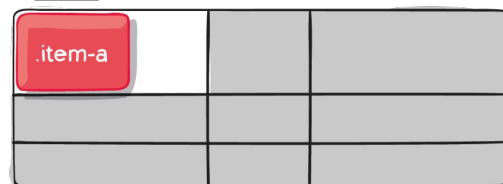
Valores:

- **start**
- **stretch**
- **end**
- **center**

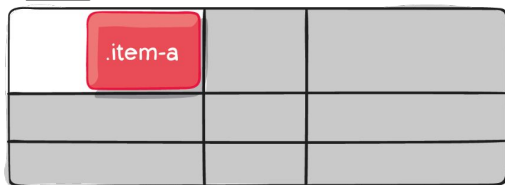
justify-self: stretch;



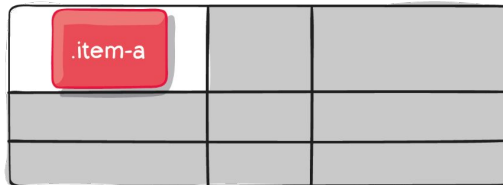
justify-self: start;



justify-self: end;



justify-self: center;



justify-content

A veces, el **tamaño total** de su **cuadrícula** puede ser menor que el tamaño de su **contenedor** de **cuadrícula**.

Esto podría suceder si todos los elementos de su cuadrícula tienen el tamaño de unidades **no flexibles** como px.

En este caso, puede establecer la **alineación** de la cuadrícula dentro del **contenedor** de la **cuadrícula**.

Esta propiedad **aline**a la **cuadrícula** a lo largo del **eje en línea** (fila).

Valores:

- **start**
- **end**
- **center**
- **stretch**
- **space-around**
- **space-between**
- **space-evenly**

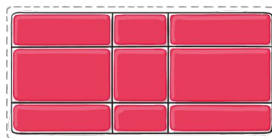
justify-content

Valores:

- start
- end
- center
- stretch
- space-around
- space-between
- space-evenly

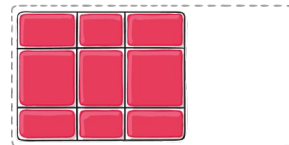
justify-content: stretch;

Contenedor Grid



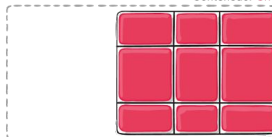
justify-content: start;

Contenedor Grid



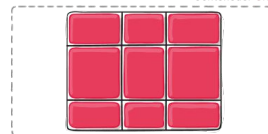
justify-content: end;

Contenedor Grid



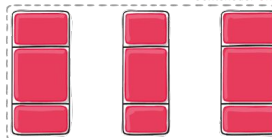
justify-content: center;

Contenedor Grid



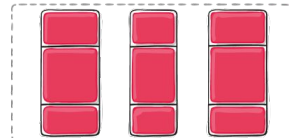
justify-content: space-between;

Contenedor Grid



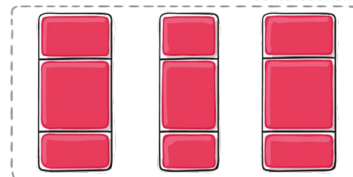
justify-content: space-around;

Contenedor Grid



justify-content: space-around;

Contenedor Grid



align-content

La propiedad **align-content** se encarga de **alinear** la **cuadrícula** a lo largo del **eje en bloque** (columna), ajustándola al contenedor de la **cuadrícula**.

Valores:

- **start**
- **end**
- **center**
- **stretch**
- **space-around**
- **space-between**
- **space-evenly**

