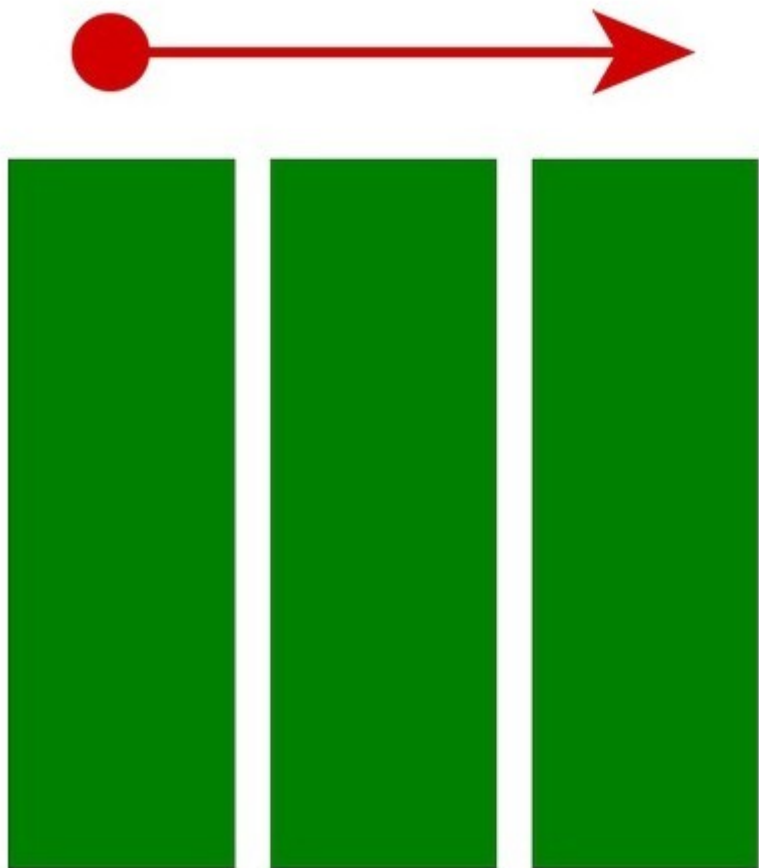
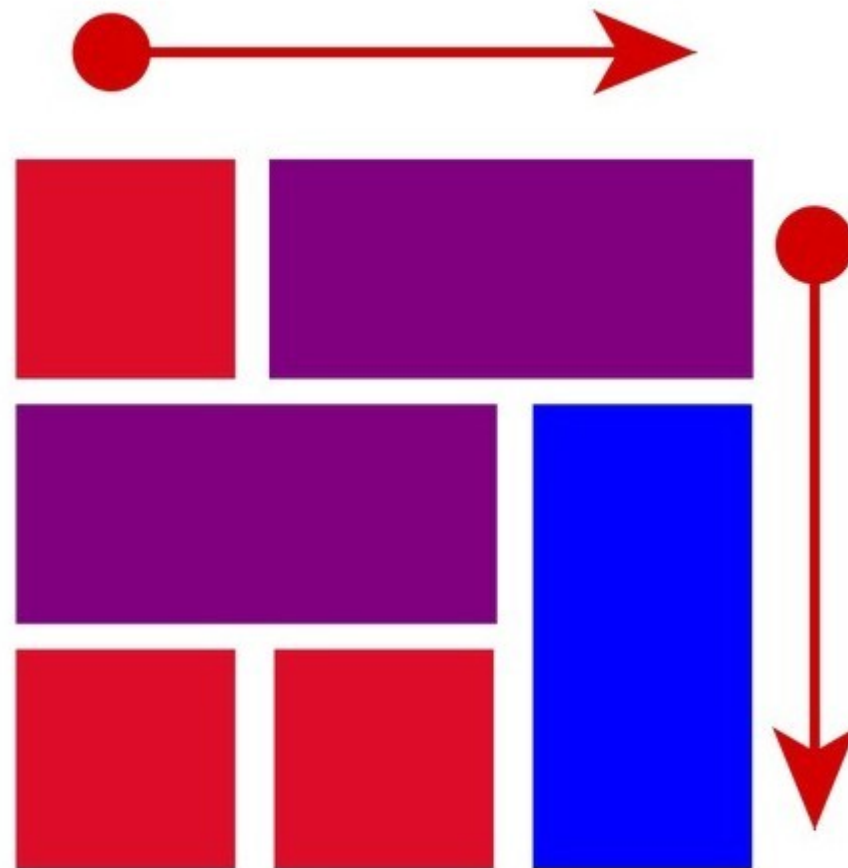


CSS GRIDS (Cuadrículas)

	header	
main	empty	sidebar
footer		

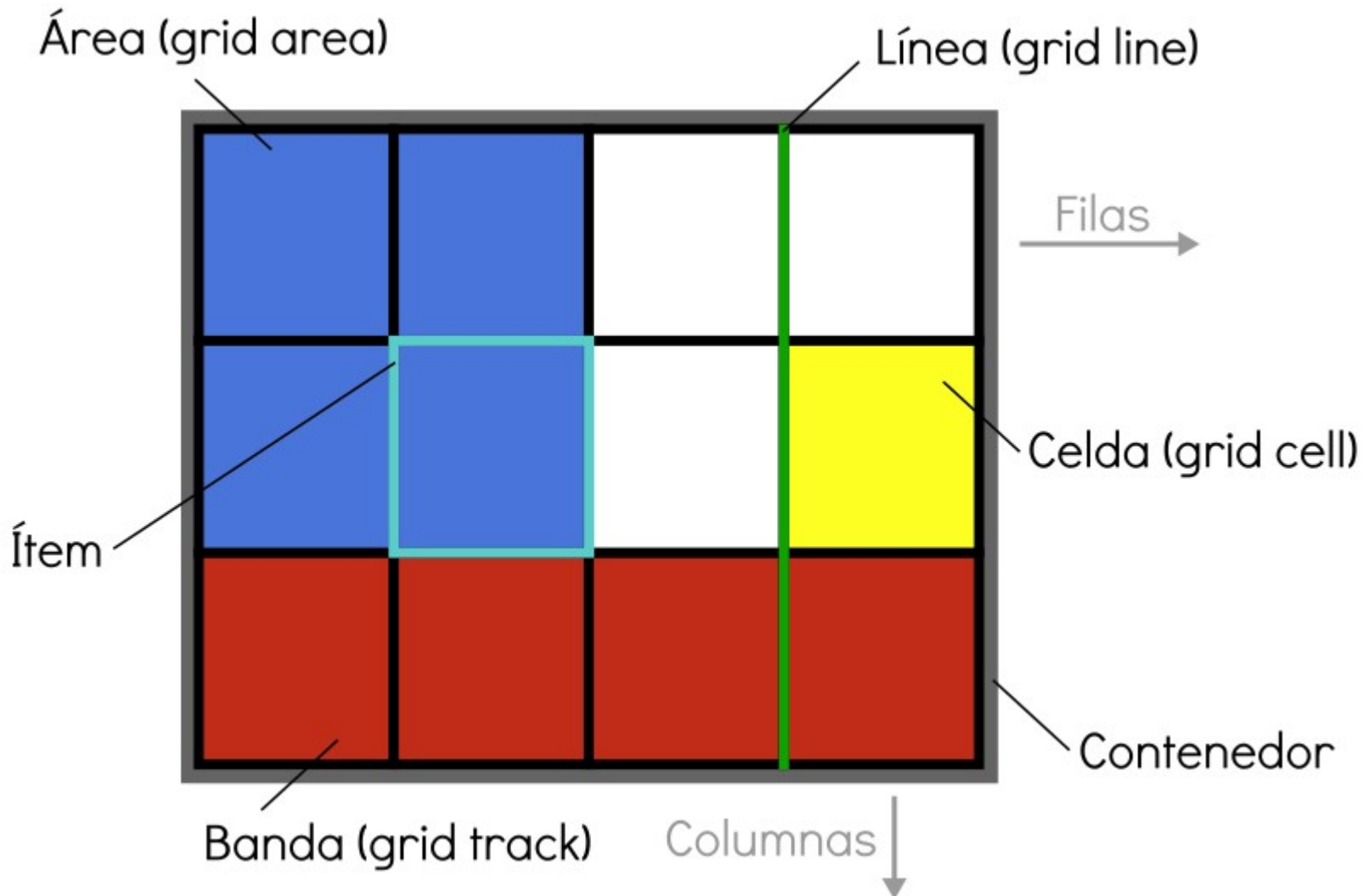
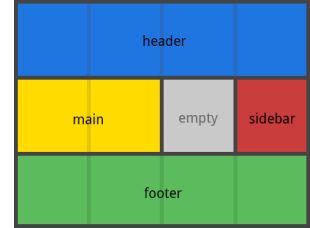


Flexbox
One Dimensions

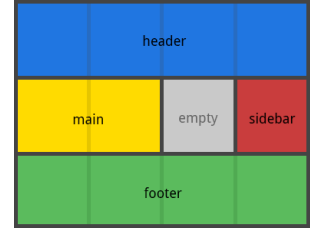


CSS Grids
Two Dimensions

CONCEPTOS CSS GRIDS



CONCEPTOS CSS GRIDS

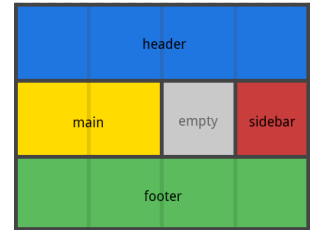


- × **Contenedor**: Existe un elemento padre que es el contenedor que definirá la cuadrícula o rejilla.
- × **Ítem**: Cada uno de los hijos que contiene la cuadrícula (elemento contenedor).
- × **Celda (grid cell)**: Cada uno de los cuadritos (unidad mínima) de la cuadrícula.
- × **Area (grid area)**: Región o conjunto de celdas de la cuadrícula.
- × **Banda (grid track)**: Banda horizontal o vertical de celdas de la cuadrícula.
- × **Línea (grid line)**: Separador horizontal o vertical de las celdas de la cuadrícula.

header		
main	empty	sidebar
footer		

PROPIEDADES CONTAINER

GRIDS CON FILAS Y COLUMNAS EXPLÍCITAS



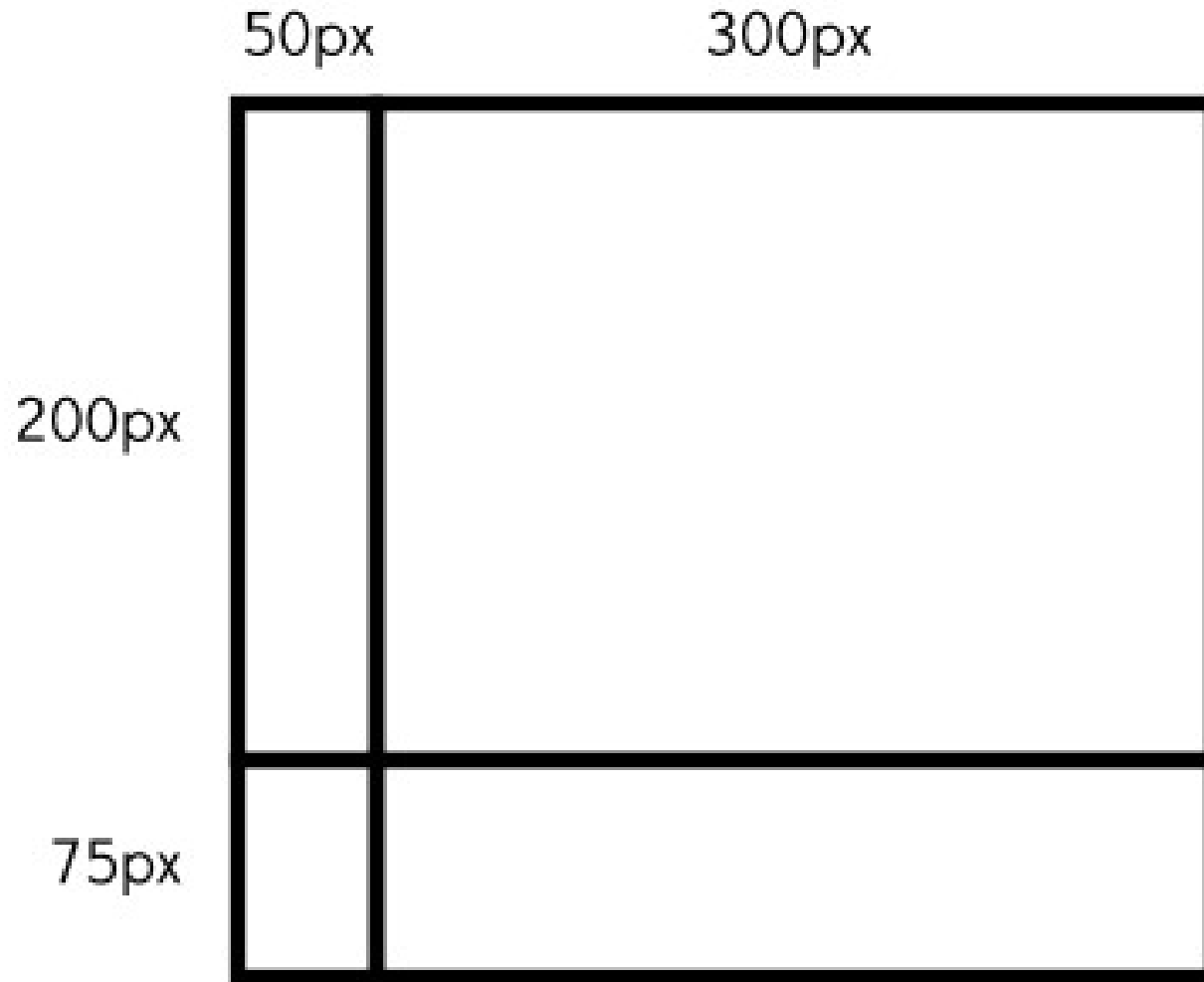
```
HTML <div class="grid"> <!-- contenedor -->
      <div class="a">Item 1</div> <!-- cada uno de los ítems del grid -->
      <div class="b">Item 2</div>
      <div class="c">Item 3</div>
      <div class="d">Item 4</div>
    </div>
```

- 1.- Container → display: grid | inline-grid;
- 2.- Crear cuadrículas con un tamaño explícito:

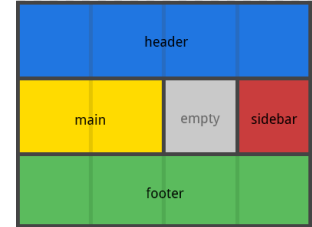
```
CSS .grid {
    display: grid;
    grid-template-columns: 50px 300px;
    grid-template-rows: 200px 75px;
}
```

GRIDS Y COLUMNAS EXPLÍCITAS

header		
main	empty	sidebar
footer		



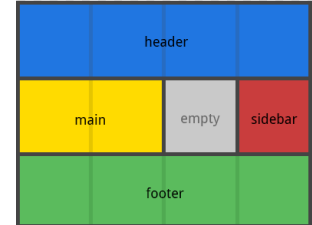
FILAS Y COLUMNAS REPETITIVAS



CSS

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

- × 4 columnas → 100px 50px 50px 200px
- × 2 filas → 50px 100px 50px 100px

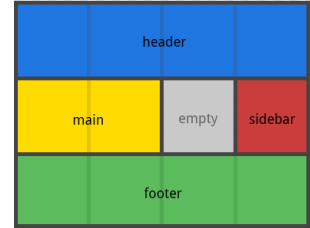


GRIDS POR AREAS

- × Es posible indicar el nombre y posición concreta de cada área de una cuadrícula.

Propiedad	Descripción
grid-template-areas	Indica la disposición de las áreas en el grid. Cada texto entre comillas simboliza una fila.
grid-area	Indica el nombre del área. Se usa sobre ítems hijos del grid.

GRIDS POR AREAS

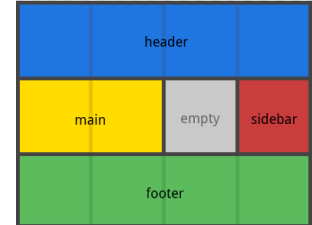


× Ejemplo:

```
CSS
.grid {
  display: grid;
  grid-template-areas: "head head"
                      "menu main"
                      "foot foot";
}

.a { grid-area: head; background: blue }
.b { grid-area: menu; background: red }
.c { grid-area: main; background: green }
.d { grid-area: foot; background: orange }
```



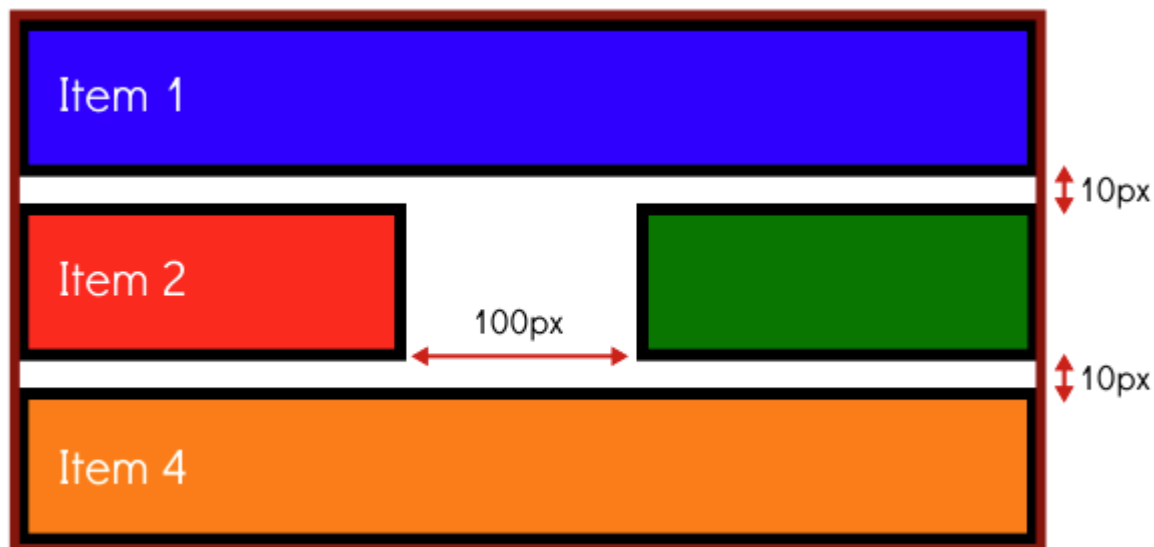


GRIDS CON HUECOS

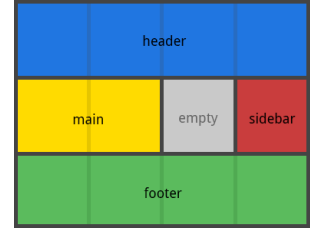
Propiedad	Descripción
grid-column-gap	Establece el tamaño de los huecos entre columnas (<i>líneas verticales</i>).
grid-row-gap	Establece el tamaño de los huecos entre filas (<i>líneas horizontales</i>).

CSS

```
.grid {  
  grid-column-gap: 100px;  
  grid-row-gap: 10px;  
}
```



POSICIONAR ELEMENTOS GRIDS

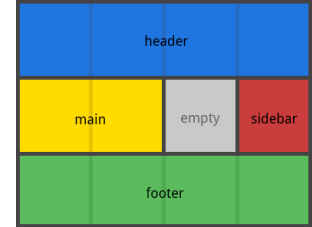


× Sobre container Grid:

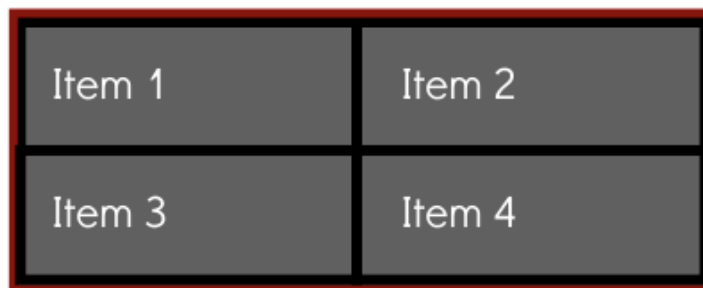
Propiedad	Valores	Descripción
justify-items	start end center stretch	Distribuye los elementos en el eje horizontal.
align-items	start end center stretch	Distribuye los elementos en el eje vertical.

- × Si queremos que alguno de los elementos Grid tenga un comportamiento distinto, aplicamos las siguientes propiedades sobre el propio elemento:
 - × justify-self
 - × align-self

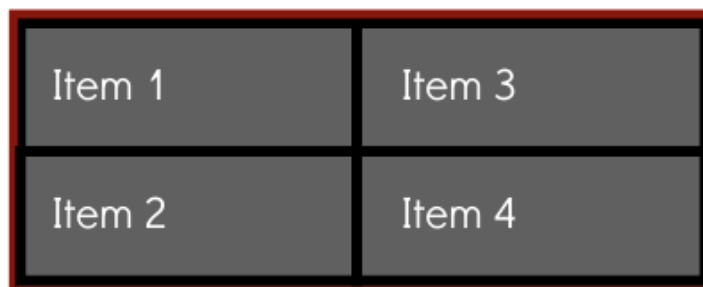
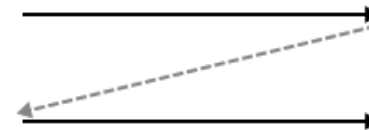
AJUSTE AUTOMÁTICO DE CELDAS



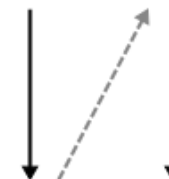
Propiedad	Valores	Descripción
grid-auto-columns	<i>[tamaño]</i>	Indica el tamaño automático de ancho que tendrán las columnas.
grid-auto-rows	<i>[tamaño]</i>	Indica el tamaño automático de alto que tendrán las filas.
grid-auto-flow	row column dense	Utiliza un algoritmo de autocolocación (intenta rellenar huecos).



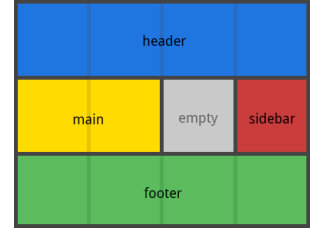
grid-auto-flow: row



grid-auto-flow: column



SHORTHAND GRID



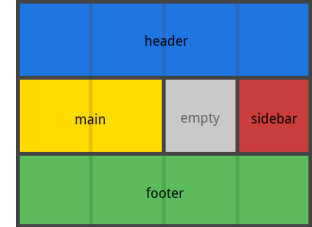
CSS

```
.grid {  
  /* grid: <grid-template> <grid-auto-flow> <grid-auto-rows> / <grid-auto-columns> */  
  
  grid: 100px 20px;  
  grid: 200px repeat(2, 100px) 300px;  
  grid: row;  
  grid: column dense;  
  grid: row 200px;  
  grid: row 400px / 150px;  
}
```

header		
main	empty	sidebar
footer		

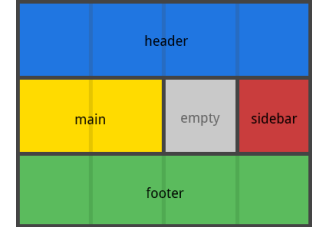
PROPIEDADES ITEMS

POSICIONAR ELEMENTOS GRIDS

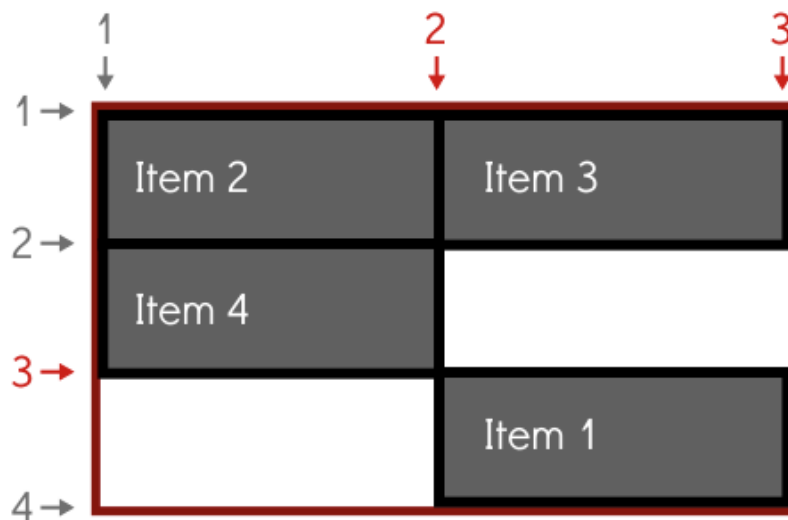


Propiedad	Descripción
grid-column-start	Indica en que columna empezará el ítem de la cuadrícula.
grid-column-end	Indica en que columna terminará el ítem de la cuadrícula.
grid-row-start	Indica en que fila empezará el ítem de la cuadrícula.
grid-row-end	Indica en que fila terminará el ítem de la cuadrícula.

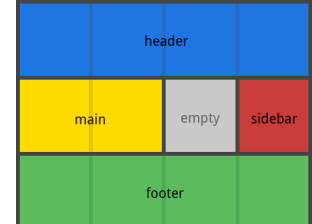
POSICIONAR ELEMENTOS GRIDS



← `grid-column-start: 1;`
`grid-column-end: 3;`



← `grid-column-start: 2;`
`grid-column-end: 3;`
`grid-row-start: 3;`



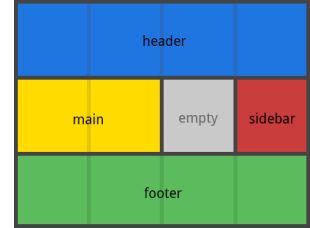
SHORTHAND

- x grid-column
- x grid-row

CSS

```
.grid {  
  display: grid;  
}  
  
.a {  
  /* grid-column: <grid-column-start> <grid-column-end> */  
  /* grid-row: <grid-row-start> <grid-row-end> */  
  grid-column: auto;  
  grid-column: 4 / 6;  
  grid-column: span 3;  
  grid-column: span 3 / 6;  
}
```

ENLACES



- x [CSS Tricks](#)
- x [Flexbox vs CSS Grids. Un ejemplo práctico I](#)
- x [Maquetación con CSS Grid. Un ejemplo práctico II](#)