

# CSS GRID

**DEV.F**  
DESARROLLAMOS(PERSONAS);

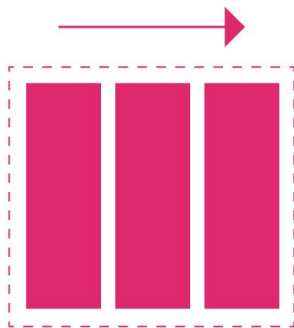
dev

# Flexbox vs CSS Grid

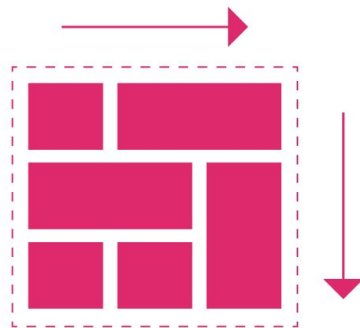
Uno de los procesos más problemáticos de CSS, es el proceso de colocar y distribuir los elementos a lo largo de una página.

El sistema **flexbox** está orientado a estructuras de una sola dimensión.

**Grid CSS** permite crear rápidamente cuadrículas sencillas y potentes.



**FLEXBOX**  
una dimensión



**CSS GRID**  
dos dimensiones

# Compatibilidad de CSS Grid

## CSS Grid Layout (level 1) - CR

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors. Includes support for all `grid-*` properties and the `fr` unit.

Current aligned Usage relative Date relative Filtered All 

Usage







% of all users  ?

Global

96.49% + 0.4% = 96.88%

unprefixed:

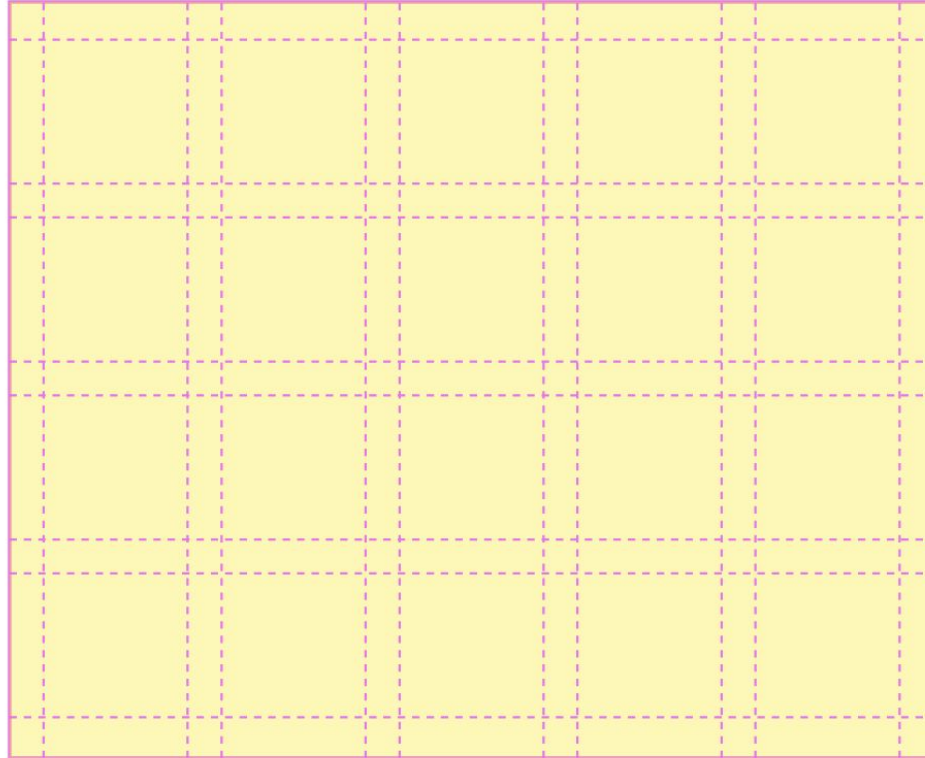
96.49%

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-28			2-39													
<sup>1</sup> 29-56 			<sup>3</sup> 40-51 	10-27												
<sup>4</sup> 57	<sup>2</sup> 12-15 	3.1-10	<sup>4</sup> 52-53	<sup>1</sup> 28-43 	6-9											
58-112	16-112	10.1-16.3	54-112	44-97	<sup>2</sup> 10 		3.2-10.2	4-5.4								
							10.3-16.3	6.2-19.0		12-12.1		2.1-4.4.4				2.5
113	113	16.4	113	98	<sup>2</sup> 11 	113	16.4	20	all	73	13.4	113	113	13.1	13.18	3.1
114-116		16.5-TP	114-115				16.5									

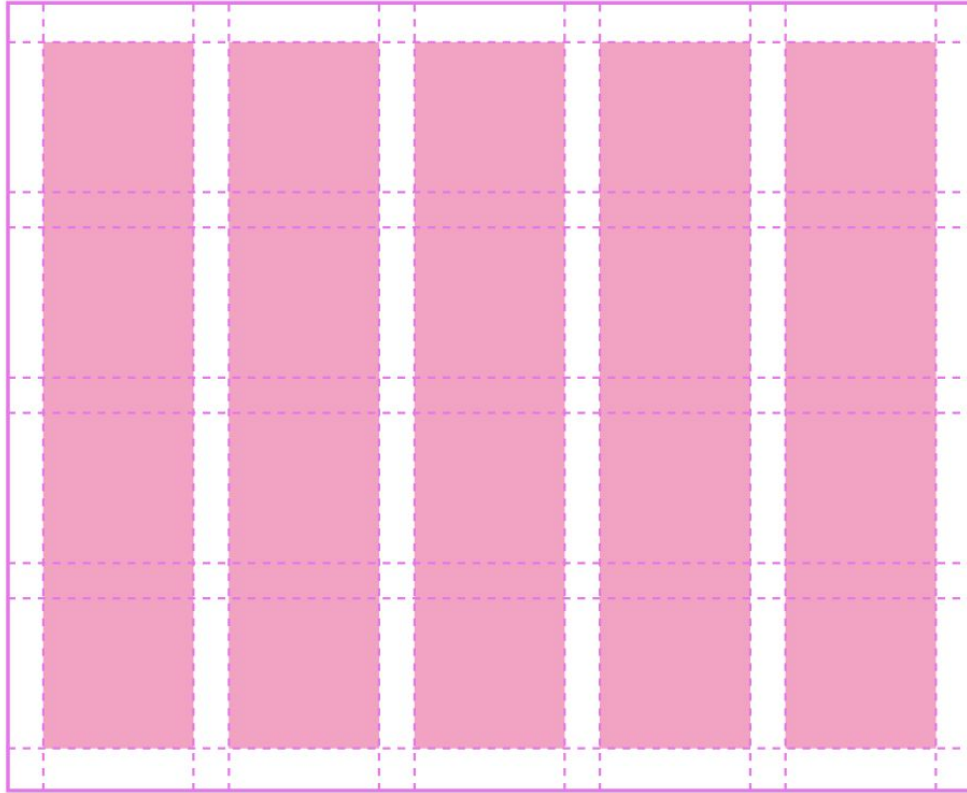
<https://caniuse.com/>

# Contenedor Grid | Grid Container

Contenedor



Columnas



A diagram illustrating a 5x5 grid structure. The grid is composed of 25 pink rectangular cells arranged in 5 rows and 5 columns. The cells are separated by dashed purple lines. The entire grid is enclosed within a solid purple border. The word "Columnas" is written above the grid.


A large rectangular area divided into a 4x5 grid of 20 identical pink rectangles. Each rectangle has a dashed border, indicating it is a template for a coloring or cutting activity. The rectangles are arranged in four rows and five columns, separated by thin white gaps. The entire grid is contained within a larger white rectangular frame.

# Conceptos básicos

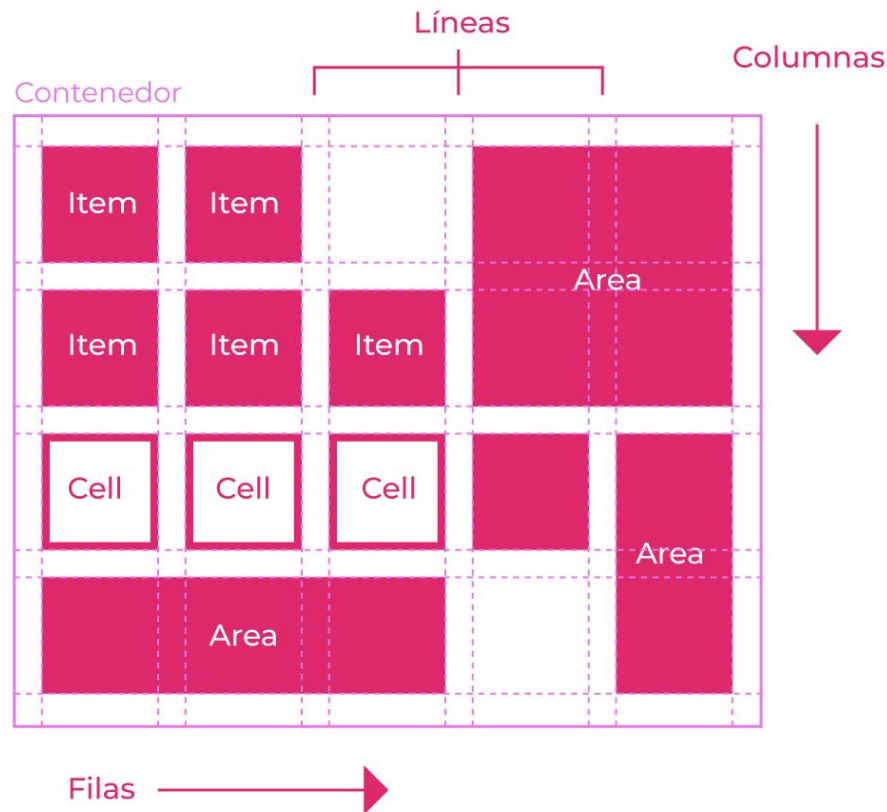
**Contenedor:** Elemento padre que es el contenedor que definirá la cuadrícula o rejilla.

**Ítem:** Los hijos que contiene la cuadrícula.

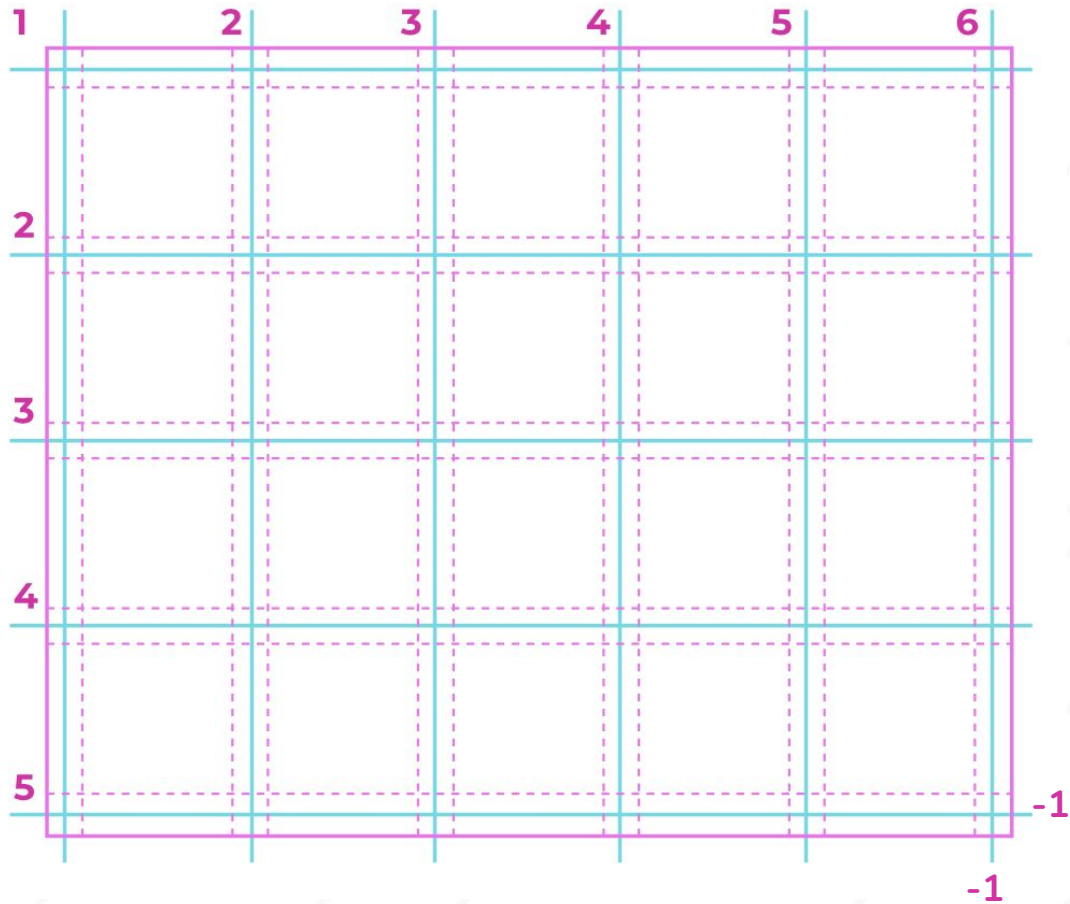
**Celda (grid cell):** Cada uno de los cuadritos (unidad mínima) de la cuadrícula.

**Area (grid area):** Conjunto de celdas.

**Línea (grid line):** Separador horizontal o vertical de las celdas de la cuadrícula.



Líneas





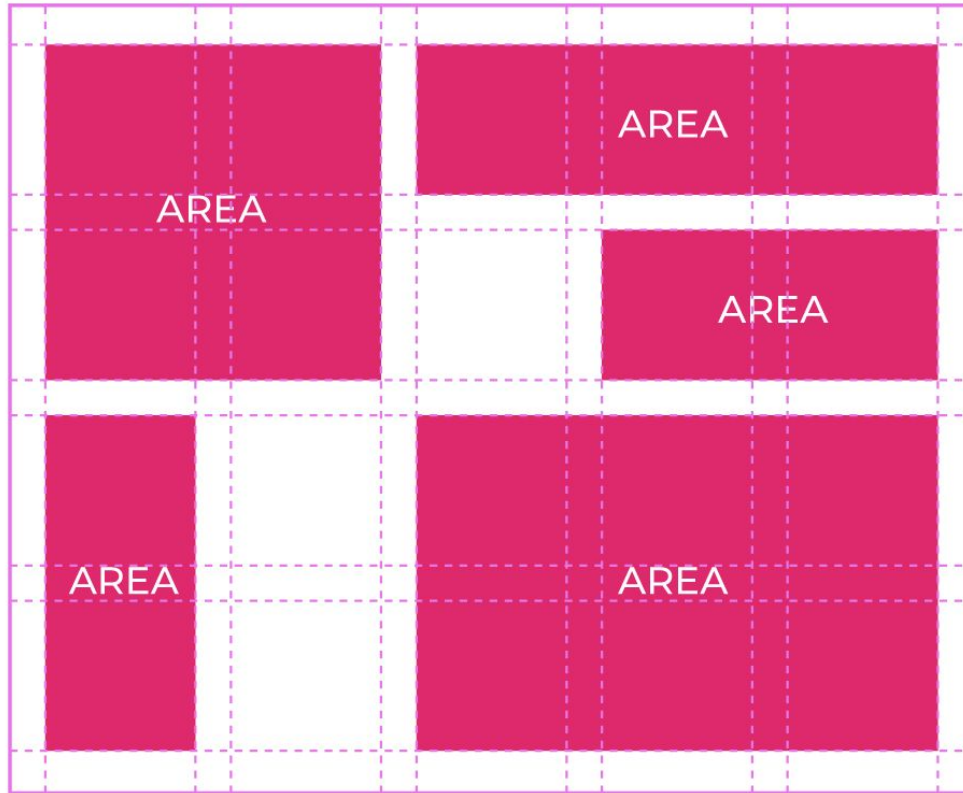
## Celdas

Cell	Cell		Cell	
Cell	Cell	Cell		
		Cell	Cell	Cell
Cell	Cell			

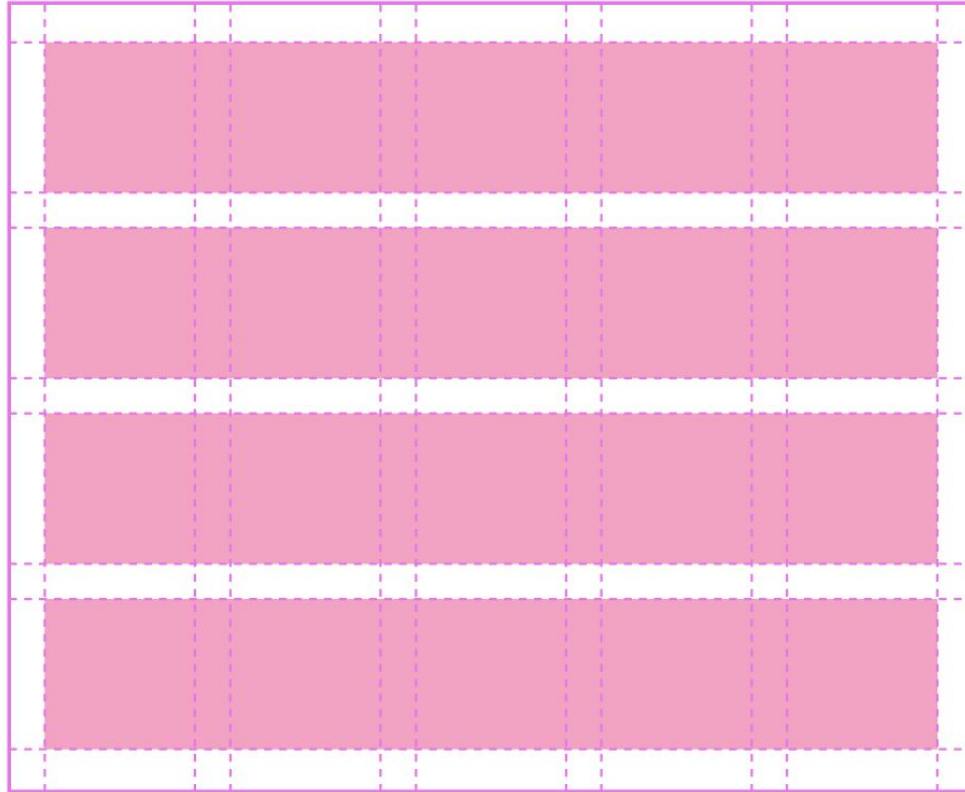
Items

Item	Item		Item	
Item	Item	Item		
		Item	Item	Item
Item	Item			

Area



# Grid Track



# Propiedades para el padre (Grid Container)

```
<div class="container">
  <div class="item1"></div>
  <div class="item2"></div>
  <div class="item3"></div>
  <div class="item4"></div>
  <div class="item5"></div>
  <div class="item6"></div>
  <div class="item7"></div>
</div>
```

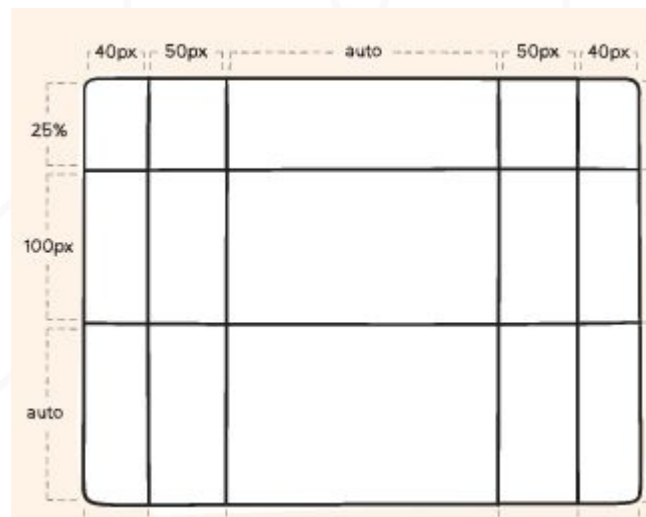
```
.container {
  display: grid | inline-grid;
}
```

- **grid** – Genera un cuadrícula a nivel de bloque
- **inline-grid** – Genera una cuadrícula a nivel de línea.

```

.container {
  grid-template-columns: ... ...;|
  /* e.g.
    1fr 1fr
    minmax(100px, 1fr) 3fr
    repeat(5, 1fr)
    50px auto 100px 1fr
  */
  grid-template-rows: ... ...;
  /* e.g.
    min-content 1fr min-content
    100px 1fr max-content
  */
}

```



```

.container {
  grid-template-columns: 40px 50px auto 50px 40px;
  grid-template-rows: 25% 100px auto;
}

```

# La fracción

- La unidad **fr** te permite establecer el tamaño como una fracción del espacio libre de la cuadrícula.
- Este espacio se calcula después de quitar todos los espacios establecidos con unidades estáticas.
- Todas las fracciones se suman para llenar el espacio.

```
.container {  
  grid-template-columns: 1fr 1fr 1fr;  
}
```

```
.container {  
  grid-template-columns: 1fr 50px 1fr 1fr;  
}
```

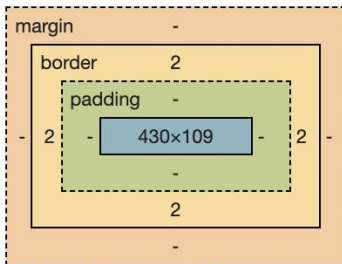
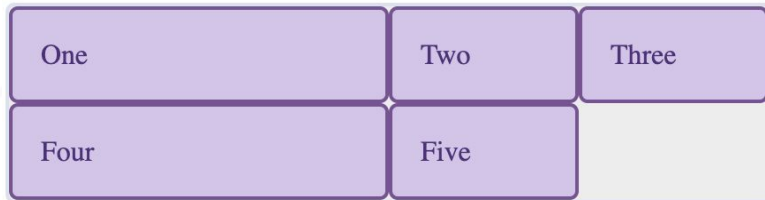
```
.container {  
  grid-template-columns: 1fr 3fr;  
}
```

```
grid-template-columns: repeat(3, 1fr);
```



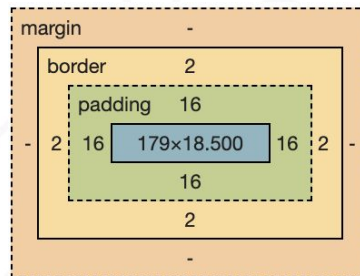
# La fracción

```
1  
2 .wrapper {  
3   display: grid;  
4   grid-template-columns: 2fr 1fr 1fr;  
5 }
```



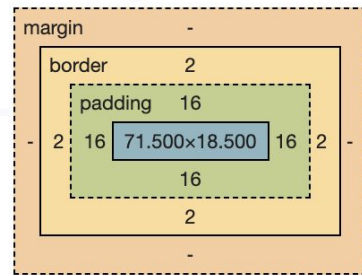
Estilos computados:

2fr



Width: 215 px

1fr



Width: 107.5 px

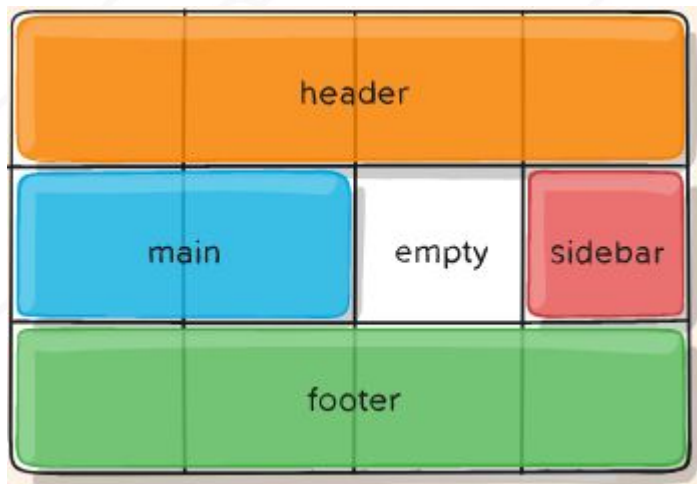
$$(107.5 \text{ px}) * 2 = 215 \text{ px}$$

$$(215 \text{ px}) + (107.5 \text{ px}) + (107.5 \text{ px}) = 430 \text{ px}$$



# Manejando áreas

```
.container {  
  grid-template-areas:  
    "<grid-area-name> | . | none | ..."  
    "...";  
}
```



```
.container {  
  display: grid;  
  grid-template-columns: 50px 50px 50px 50px;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}
```

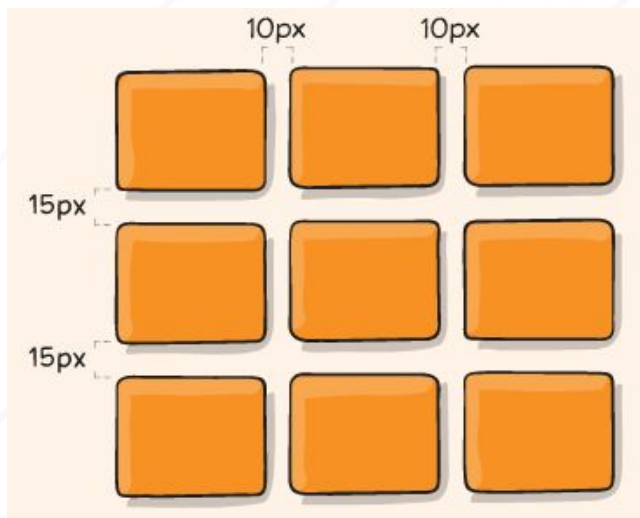
# Espaciado

```
.container {  
  /* standard */  
  gap: <grid-row-gap> <grid-column-gap>;  
  
  /* old */  
  grid-gap: <grid-row-gap> <grid-column-gap>;  
}
```

```
.container {  
  /* standard */  
  column-gap: <line-size>;  
  row-gap: <line-size>;  
  
  /* old */  
  grid-column-gap: <line-size>;  
  grid-row-gap: <line-size>;  
}
```

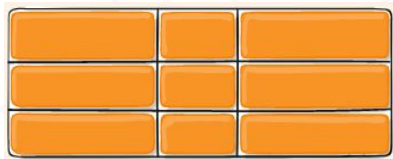
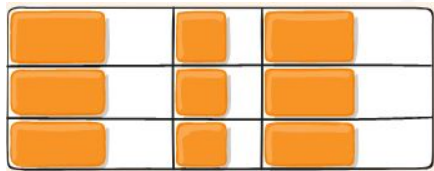
```
.container {  
  grid-template-columns: 100px 50px 100px;  
  grid-template-rows: 80px auto 80px;  
  column-gap: 10px;  
  row-gap: 15px;  
}
```

gap: 15px 10px;

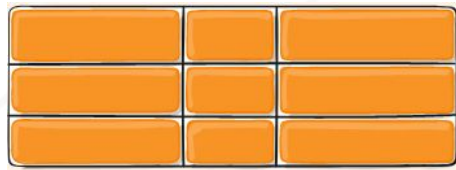
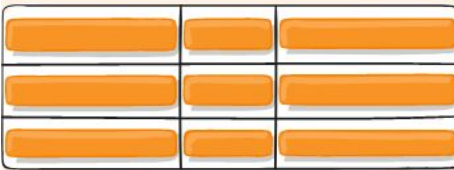
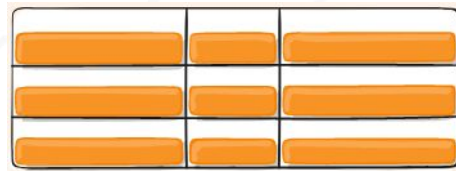
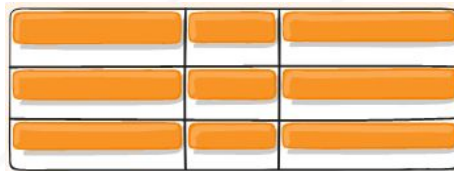


# Justificar y Alinear Elementos

```
.container {  
  justify-items: start | end | center | stretch;  
}
```



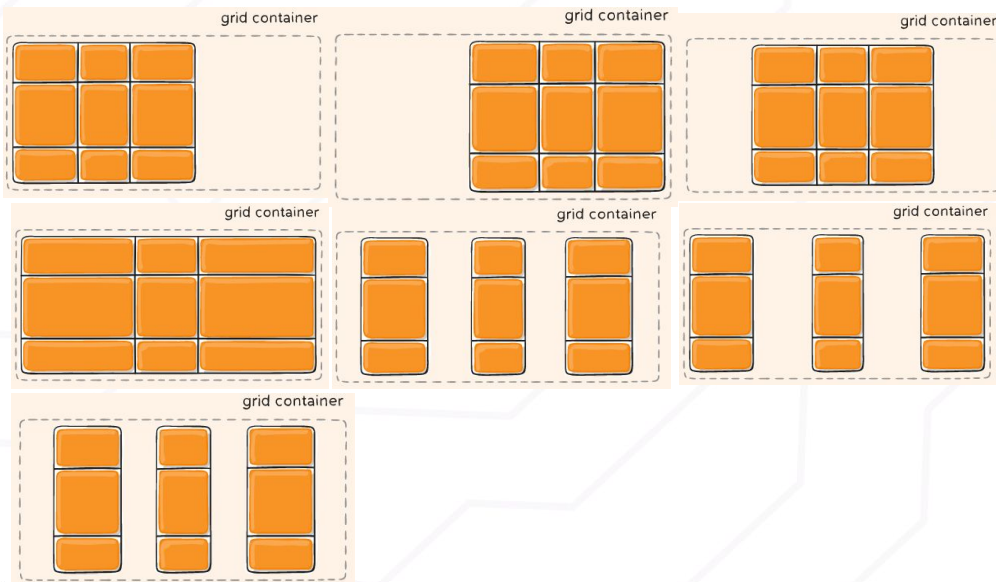
```
.container {  
  align-items: start | end | center | stretch;  
}
```



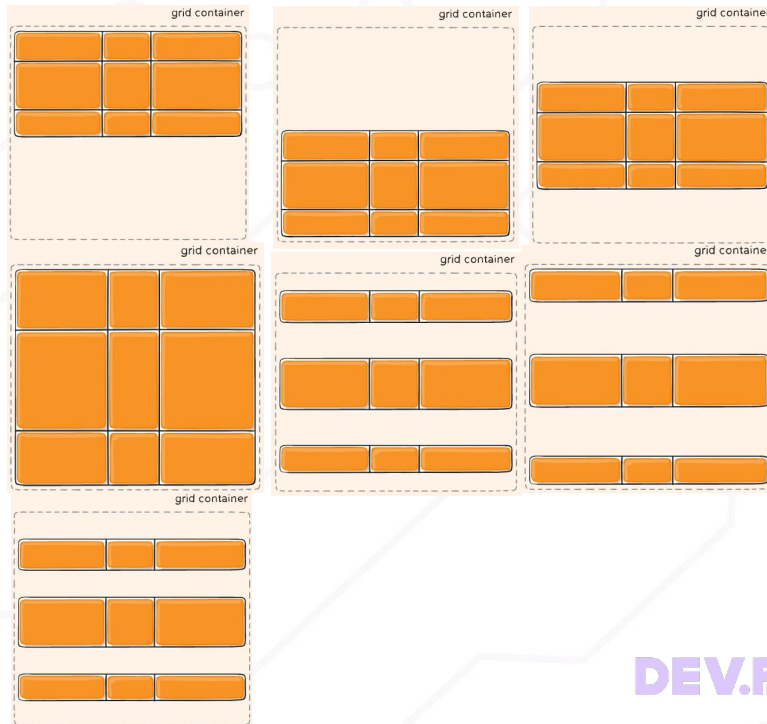
```
1 .container {  
2   display: grid;  
3   place-items: <align-items> / <justify-items> ;  
4 }
```

# Justificar y Alinear Contenido

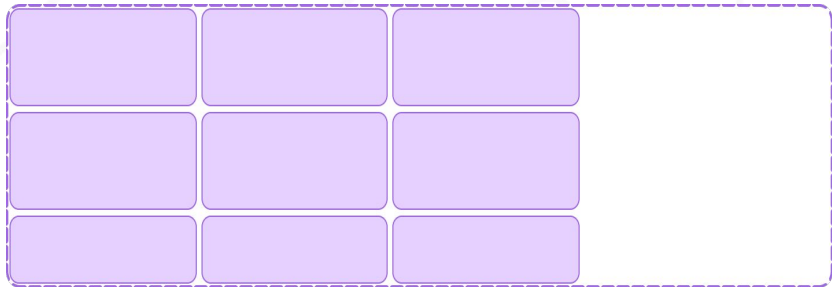
```
.container {  
  justify-content / align-content : start | end | center | stretch | space-around | space-between | space-evenly;  
}
```



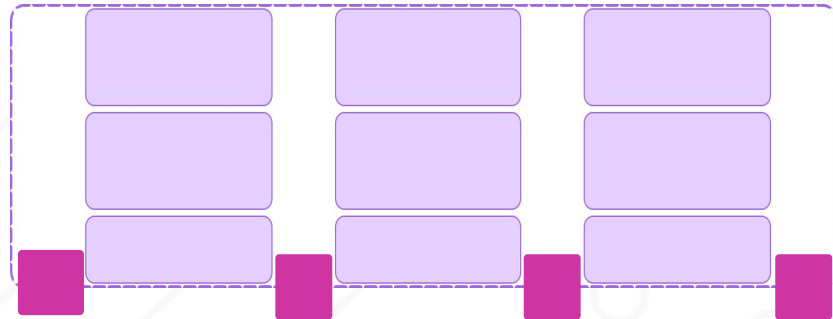
```
1 .container {  
2   display: grid;  
3   place-content: <align-content> / <justify-content> ;  
4 }
```



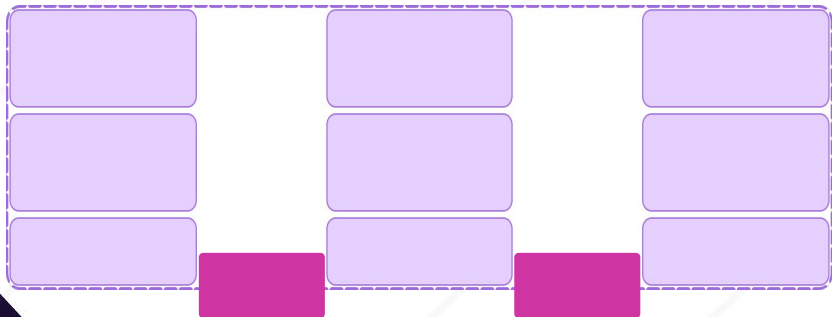
# space evenly | space between | space around



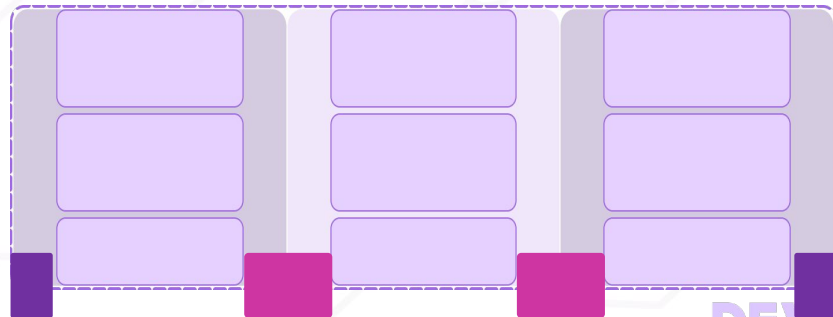
**space-evenly**



**space-between**



**space-around**

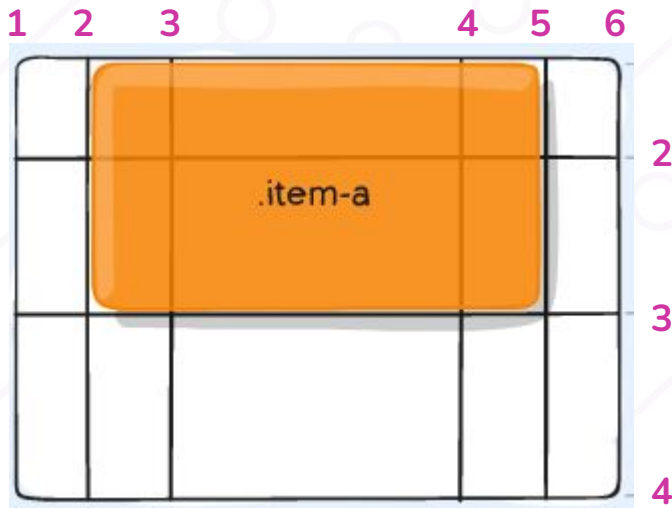


# Propiedades para los hijos (Grid Items)

```
.item-a {  
  grid-column-start: 2;  
  grid-column-end: 5;  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```

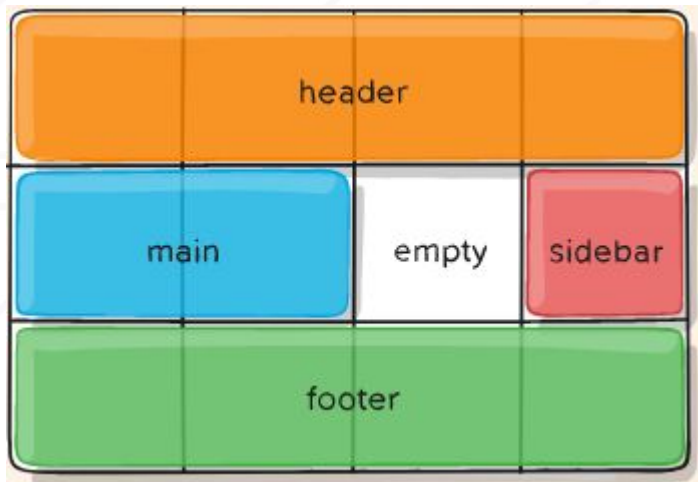
```
.item-a {  
  grid-column-start: 2;  
  grid-column-end: span 3;  
  grid-row-start: 1;  
  grid-row-end: span 2;  
}
```

```
.item-a {  
  grid-column: 2 / 5;  
  grid-row: 1 / 3;  
}
```





# Ubicando item en areas

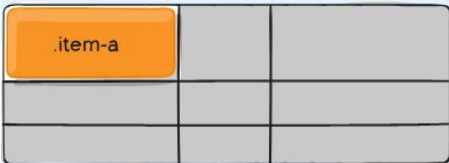
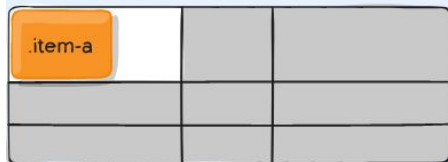


```
.container {  
  display: grid;  
  grid-template-columns: 50px 50px 50px 50px;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}
```

```
.item-a {  
  grid-area: header;  
}  
.item-b {  
  grid-area: main;  
}  
.item-c {  
  grid-area: sidebar;  
}  
.item-d {  
  grid-area: footer;  
}
```

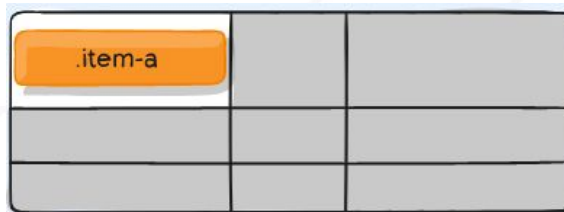
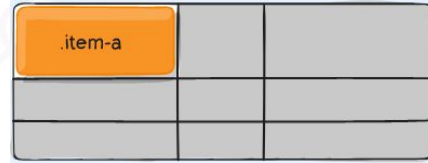
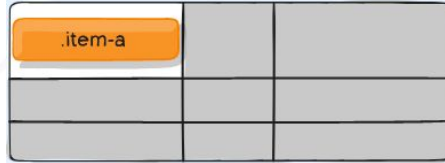
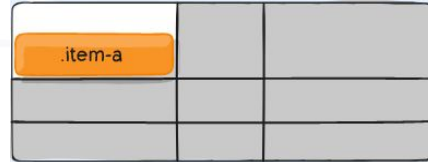
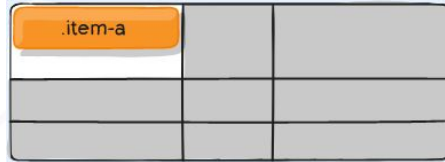
# Justificación y alineación individual

```
.item {  
  justify-self: start | end | center | stretch;  
}
```



```
.item-a {  
  place-self: center stretch;  
}
```

```
.item {  
  align-self: start | end | center | stretch;  
}
```

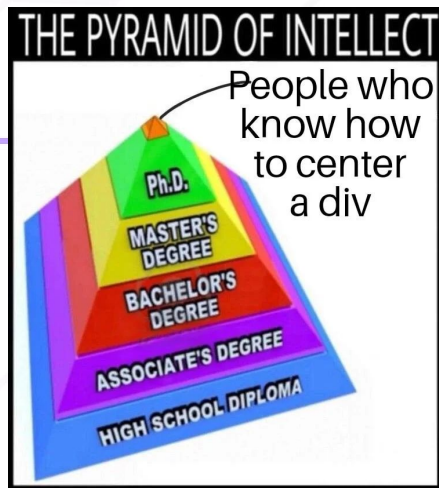
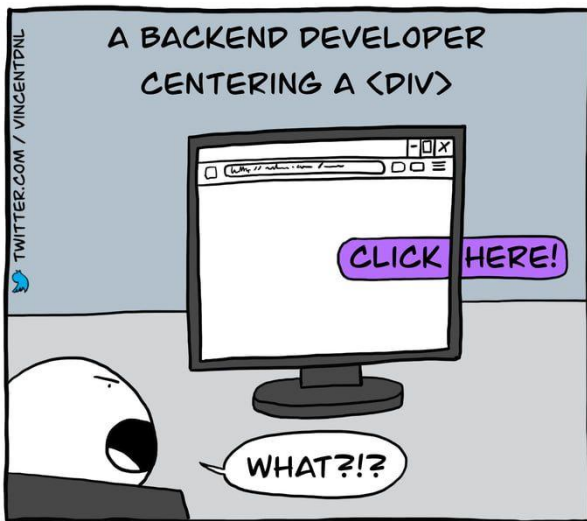




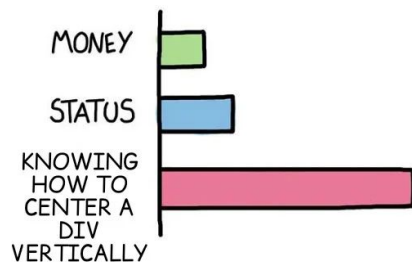
# BONUS

**DEV.F**  
DESARROLLAMOS(PERSONAS);

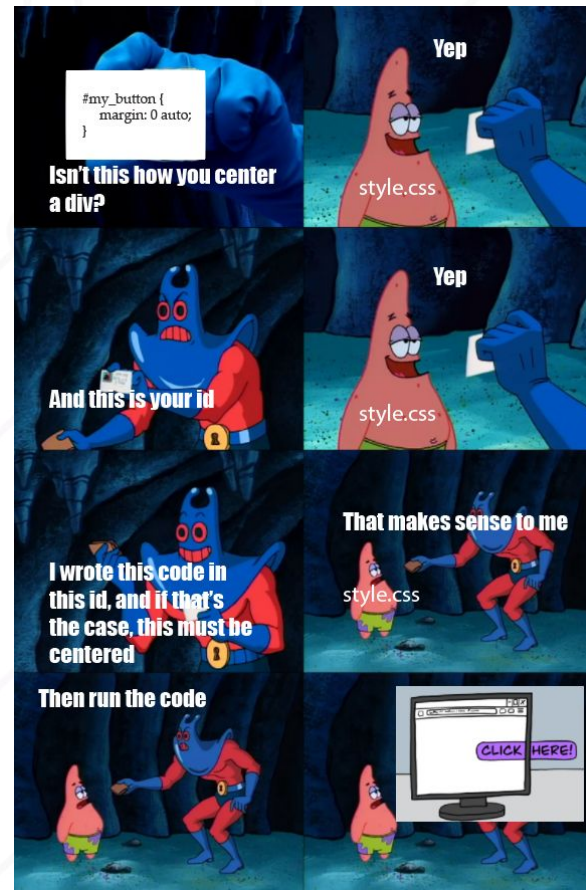
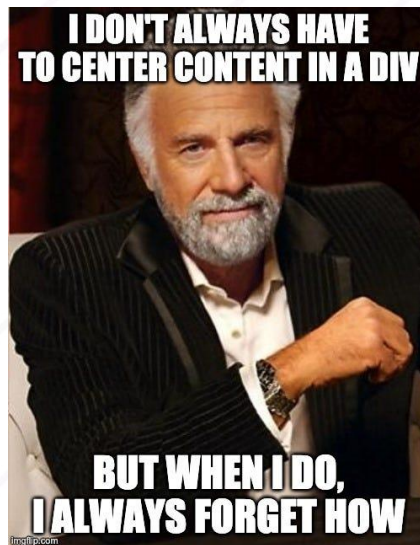
dev



## WHAT GIVES PEOPLE FEELINGS OF POWER



@iamnotanerd





```
1 .grid-container {  
2     display: grid;  
3     place-content: center center;  
4 }
```