



Una guía de

# CSS Grid



CSS Grid

# Introducción

- Sistema de diseño bidimensional basado en cuadrículas
- Cambia por completo la forma en que se diseñan UIs basadas en cuadrículas
- Antes se usaban subsistemas de propiedades css a partir de **table, float, position, display: inline-block**

Eran esencialmente “hacks”.  
Dejaban de lado funcionalidades importantes  
(centrado vertical, por ejemplo)

- **Flexbox** ayudó mucho, pero...  
Está destinado a diseños unidimensionales más simples  
(Flexbox y Grid funcionan muy bien juntos)

Grid es el primer subsistema CSS creado específicamente para resolver los problemas de diseño que hemos estado “emparchando” desde que se comenzaron a diseñar sitios web



CSS Grid

# Conceptos básicos

- Se aplica **CSS grid** a un módulo contenedor que tenga elementos contenidos
- Como en **Flexbox**, es una relación entre “padres” e “hijos” (los “nietos” no participan de esta relación)
- Como en **Flexbox**, estas relaciones pueden anidarse (para incluir “nietos”, “bisnietos”, etc.)
- Como en **Flexbox**, elementos en la cuadrícula pueden colocarse en cualquier orden mediante CSS

- Se define el elemento contenedor como **CSS grid** con:

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

- Se establecen los tamaños de columna y fila con:

```
grid-template-columns: ...;  
grid-template-rows: ...;
```

- Y se emplazan los elementos contenidos con:

```
.elemento {  
  grid-column: ...;  
  grid-row: ...;  
}
```



CSS Grid

# Soporte de navegadores



# CSS Grid / Soporte de Navegadores / Escritorio

IE	Edge *	Firefox	Chrome	Safari
		2 - 39	4 - 28	
		3 40 - 51	1 29 - 56	
6 - 9	2 12 - 15	4 52 - 53	4 57	3.1 - 10
2 10	16 - 17	54 - 67	58 - 75	10.1 - 12
2 11	18	68	76	12.1
	76	69 - 70	77 - 79	13 - TP

# CSS Grid / Soporte de Navegadores / Mobile

[illegible]



CSS Grid

# Terminología

## Contenedor

- Elemento sobre el que se aplica **display: grid**.
- Es el “padre” directo de los elementos de la cuadrícula.
- En este ejemplo el contenedor de la cuadrícula es **.container**

```
<div class="container">  
  <div class="item"></div>  
  <div class="item"></div>  
  <div class="item"></div>  
</div>
```

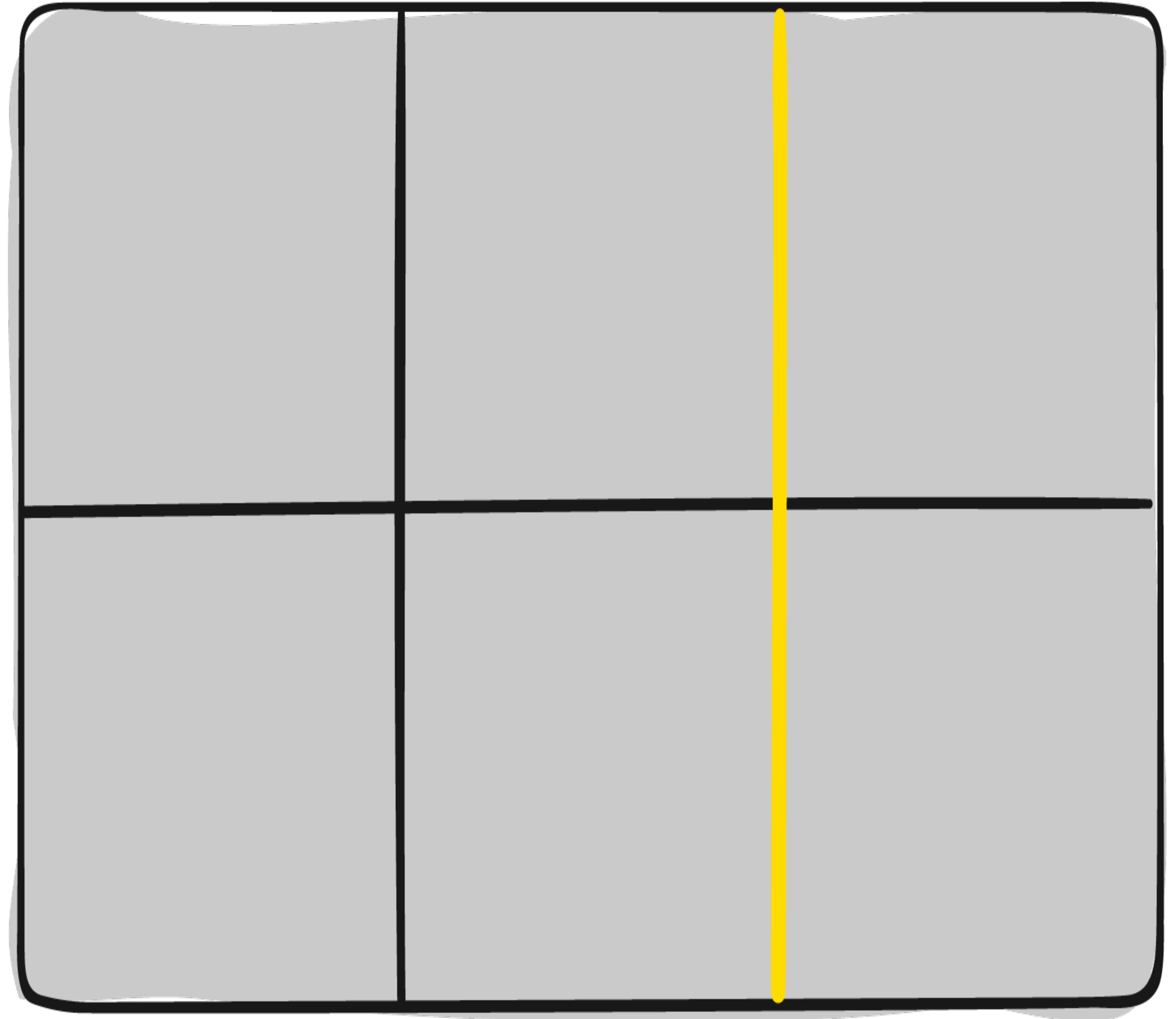
## Elemento (o item)

- Es “hijo” directo del contenedor al que se aplica **display: grid**.
- En este ejemplo **.item** son elementos de la cuadrícula.
- En este ejemplo **.sub-item**. NO es elemento de la cuadrícula

```
<div class="container">  
  <div class="item"></div>  
  <div class="item">  
    <p class="sub-item"></p>  
  </div>  
  <div class="item"></div>  
</div>
```

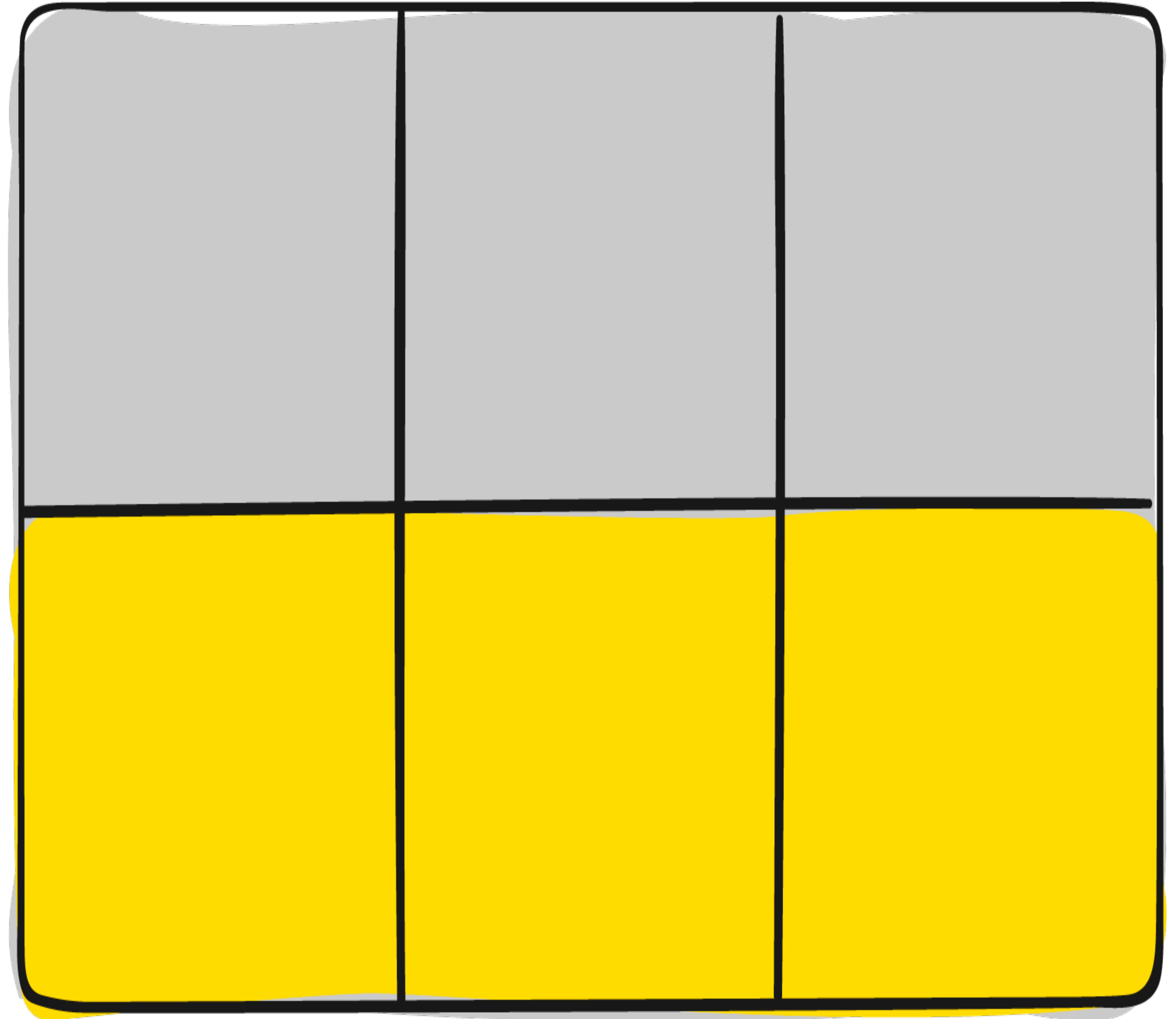
### Línea (Grid Line)

- Líneas divisorias que componen la estructura de la cuadrícula.
- Pueden ser verticales (“column grid lines”) u horizontales (“row grid lines”) y se sitúan a ambos lados de una fila o columna.
- La línea amarilla es ejemplo de línea de columna de cuadrícula



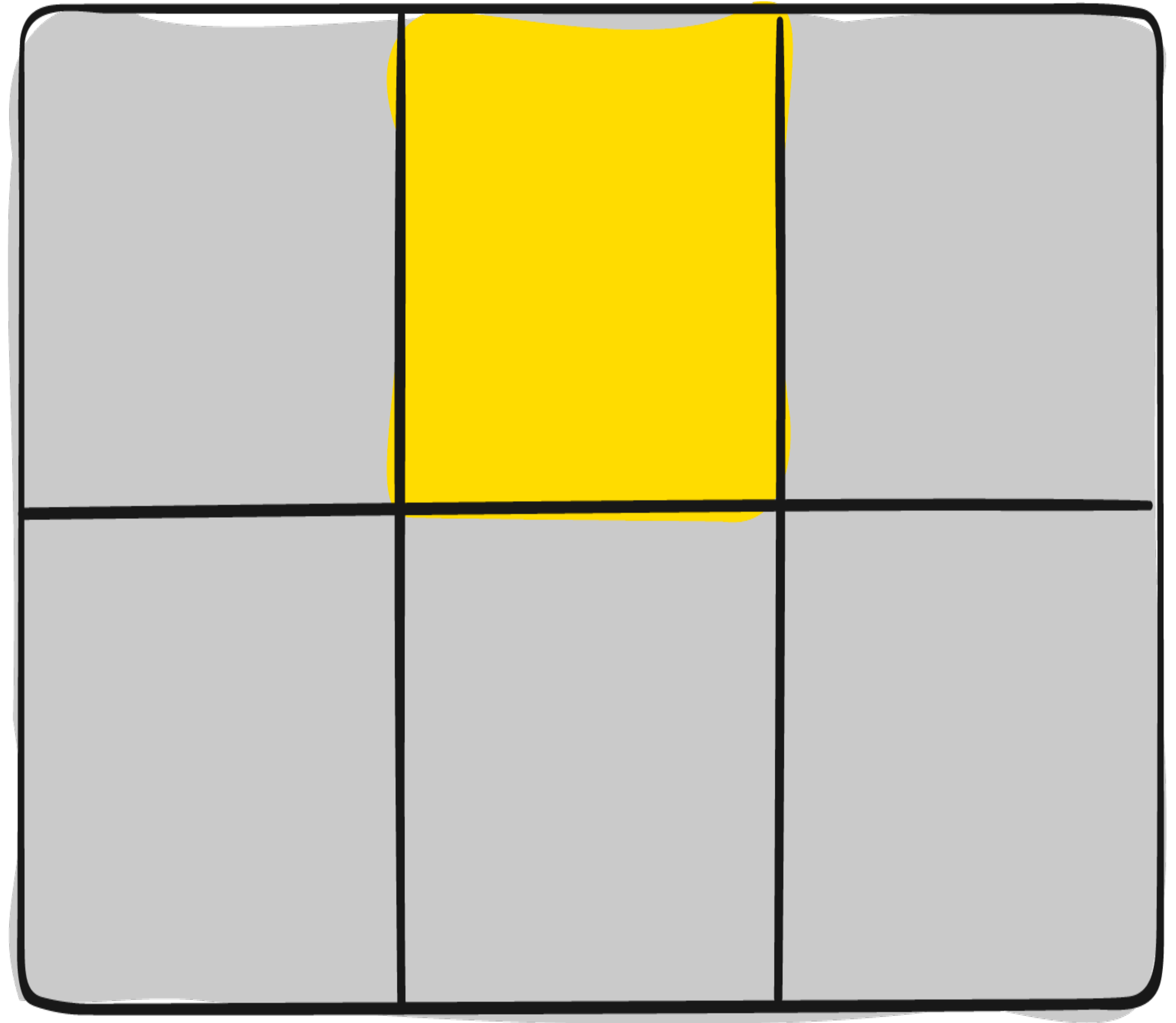
### Carril o Pista (Grid Track)

- Espacio entre dos líneas de cuadrícula adyacentes
- Se pueden pensar como columnas o filas de la cuadrícula
- En amarillo, pista de cuadrícula entre la segunda y la tercera línea de fila de cuadrícula



### Celda (Grid Cell)

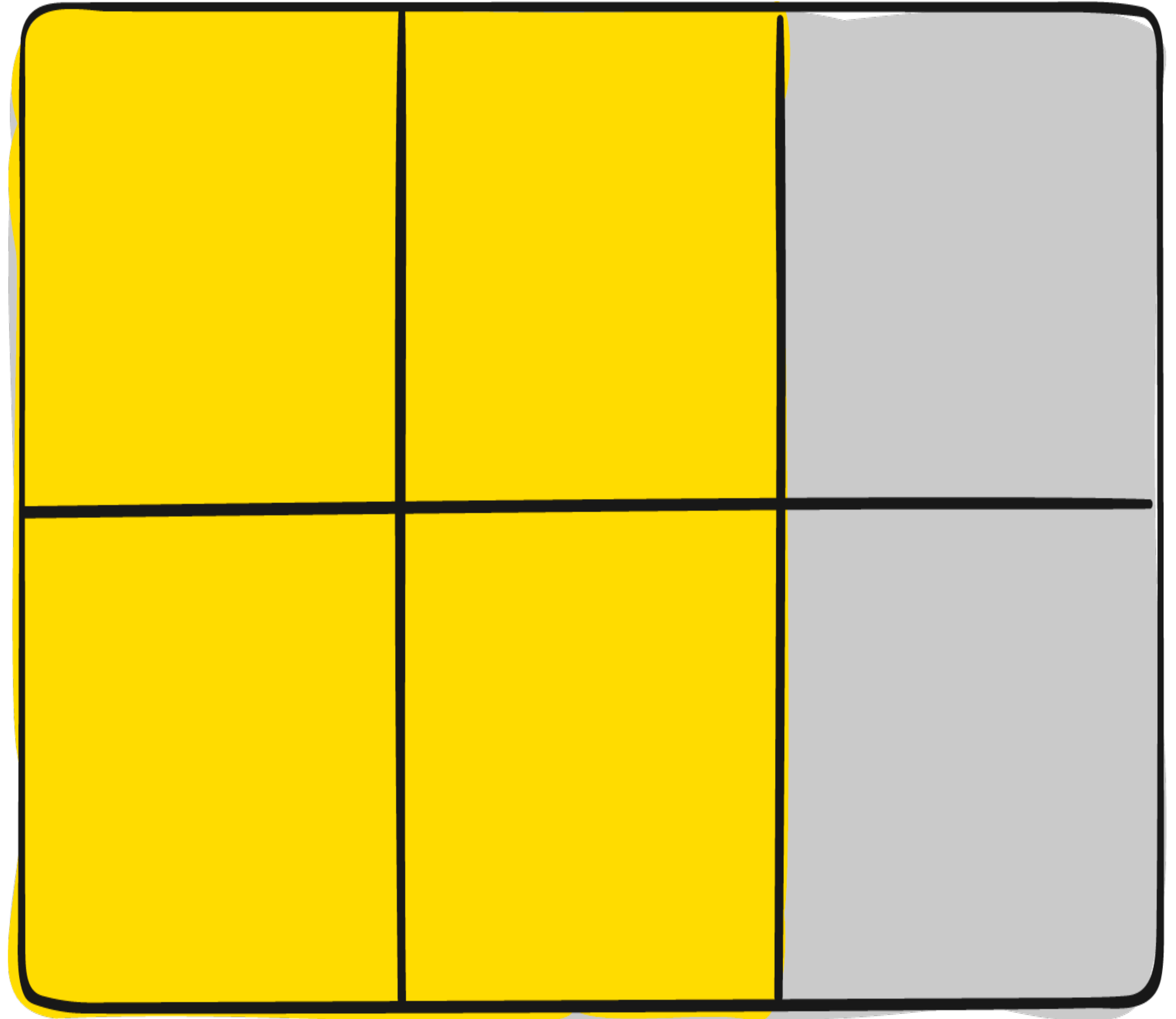
- Espacio entre dos líneas de fila y dos líneas de columna adyacentes
- Es la “unidad” de la cuadrícula
- En amarillo, celda de cuadrícula entre las líneas de fila 1 y 2, y líneas de columna 2 y 3





### Área (Grid Area)

- Espacio total rodeado por cuatro líneas de cuadrícula
- Puede estar compuesta por cualquier número de celdas
- En amarillo, área de cuadrícula entre las líneas de fila 1 y 3, y líneas de columna 1 y 3





CSS Grid

# Propiedades CSS para el contenedor

# display

Define el elemento como un contenedor de cuadrícula y establece un nuevo contexto de formato de cuadrícula para sus elementos contenidos

Valores:

<b>grid</b>	genera una cuadrícula “bloque”
<b>inline-grid</b>	genera una cuadrícula “en línea”

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

**grid-template-columns**

**grid-template-rows**

Define columnas y filas de la cuadrícula con una lista de valores separados por espacios. Los valores representan el tamaño de la pista y el espacio entre ellos representa la línea de la cuadrícula

Valores:

<b>&lt;tamaño de pista&gt;</b>	longitud, porcentaje o fracción del espacio libre en la cuadrícula
<b>&lt;line-name&gt;</b>	nombre arbitrario a elección

**grid-template-columns**

**grid-template-rows**

Valores:

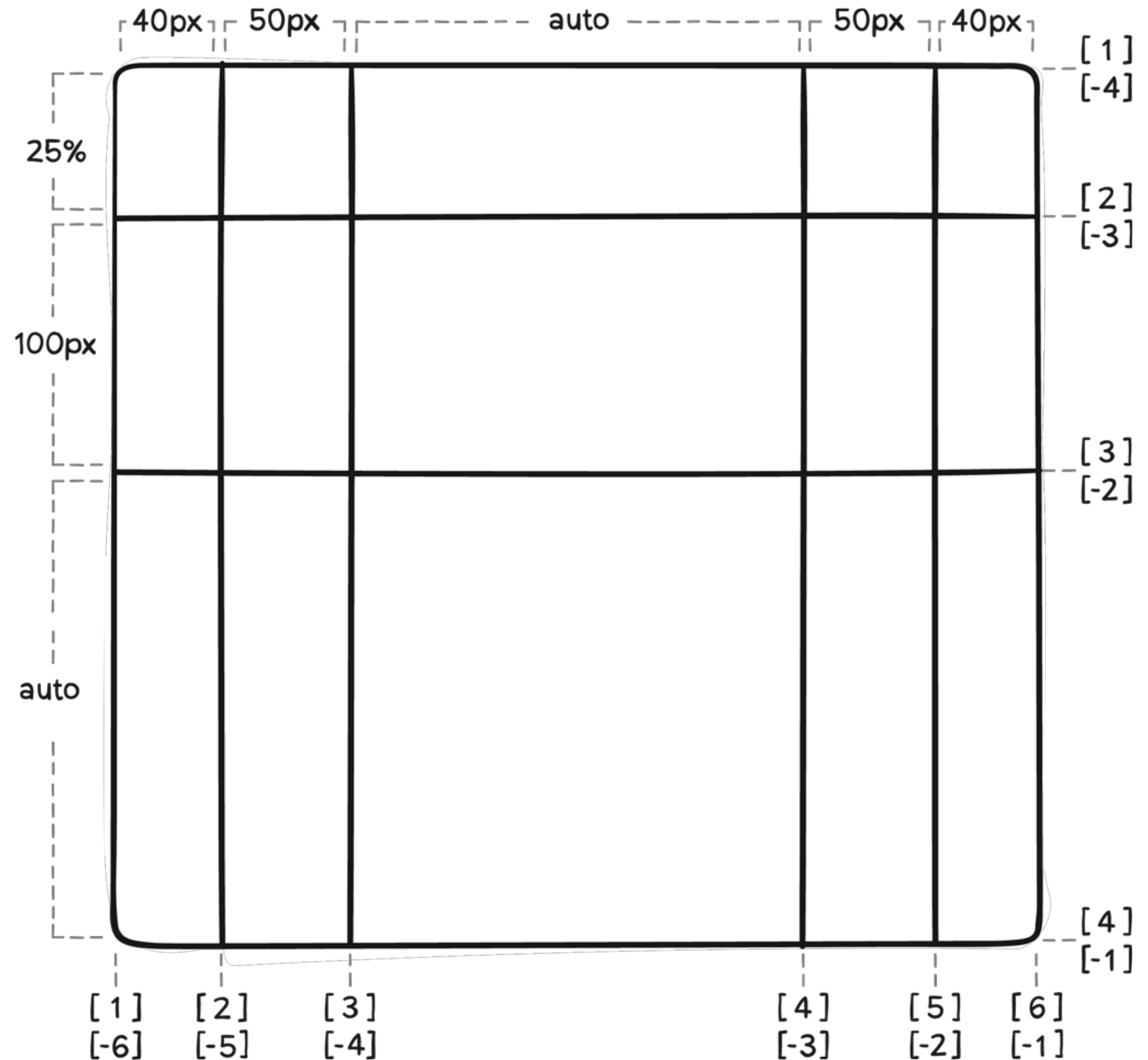
**<tamaño de pista>** longitud, porcentaje o fracción del espacio libre en la cuadrícula  
**<line-name>** nombre arbitrario a elección

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

## CSS Grid / Propiedades / Contenedor

Cuando se deja un espacio vacío entre los valores de la pistas, a las líneas de la cuadrícula se les asignan automáticamente números positivos y negativos

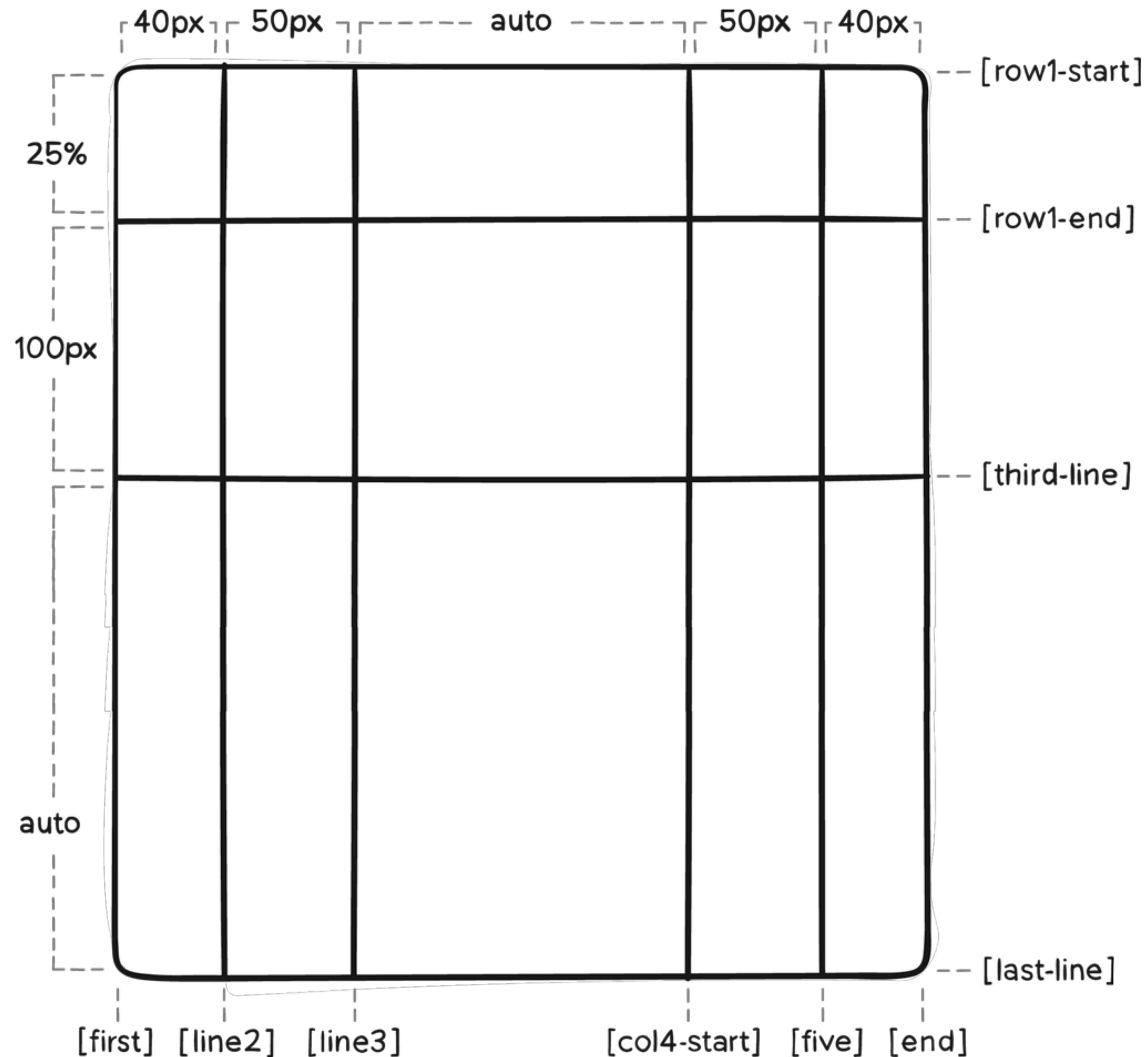
```
.container {  
  grid-template-columns:  
    40px 50px auto 50px 40px;  
  grid-template-rows: 25%  
    100px auto;  
}
```



# CSS Grid / Propiedades / Contenedor

Las líneas se pueden nombrar.  
Los nombres van entre corchetes:

```
.container {  
  grid-template-columns:  
    [first] 40px [line2] 50px  
    [line3] auto [col4-start]  
    50px [five] 40px [end];  
  grid-template-rows:  
    [row1-start] 25% [row1-  
end] 100px [third-line]  
    auto [last-line];  
}
```



## CSS Grid / Propiedades / Contenedor

Una línea puede tener más de un nombre.

En el ejemplo, la segunda línea tendrá dos nombres: row1-end y row2-start

```
.container {  
  grid-template-rows: [row1-start] 25% [row1-end row2-start] 25%  
  [row2-end];  
}
```



Definiciones con partes repetidas pueden usar la notación **repeat()** :

```
.container {  
  grid-template-columns:  
  repeat(3, 1.5em [col-start]);  
}
```

Equivale a:

```
.container {  
  grid-template-columns:  
  1.5em [col-start] 1.5em [col-  
start] 1.5em [col-start];  
}
```

La unidad **fr** permite establecer el tamaño de una pista como fracción del espacio libre del contenedor de cuadrícula.

En el ejemplo, cada elemento tendrá un tercio del ancho del contenedor:

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

El espacio libre se calcula después de cualquier elemento no flexible. En el ejemplo, la cantidad total de espacio libre disponible para las unidades **fr** no incluye los **4em**:

```
.container {  
  grid-template-columns: 1fr 4em 1fr 1fr;  
}
```

### **grid-template-areas**

- Define una plantilla haciendo referencia a los nombres de las áreas que se especifican con la propiedad **grid-area**
- Repetir el nombre de un área hace que el contenido abarque esas celdas
- Un punto significa una celda vacía
- La sintaxis en sí proporciona una visualización de la estructura de la cuadrícula

Valores:

<b>&lt;grid-area-name&gt;</b>	nombre de un área especificada con <b>grid-area</b>
<b>.</b> (punto)	una celda de cuadrícula vacía
<b>none</b>	no se definen áreas de cuadrícula

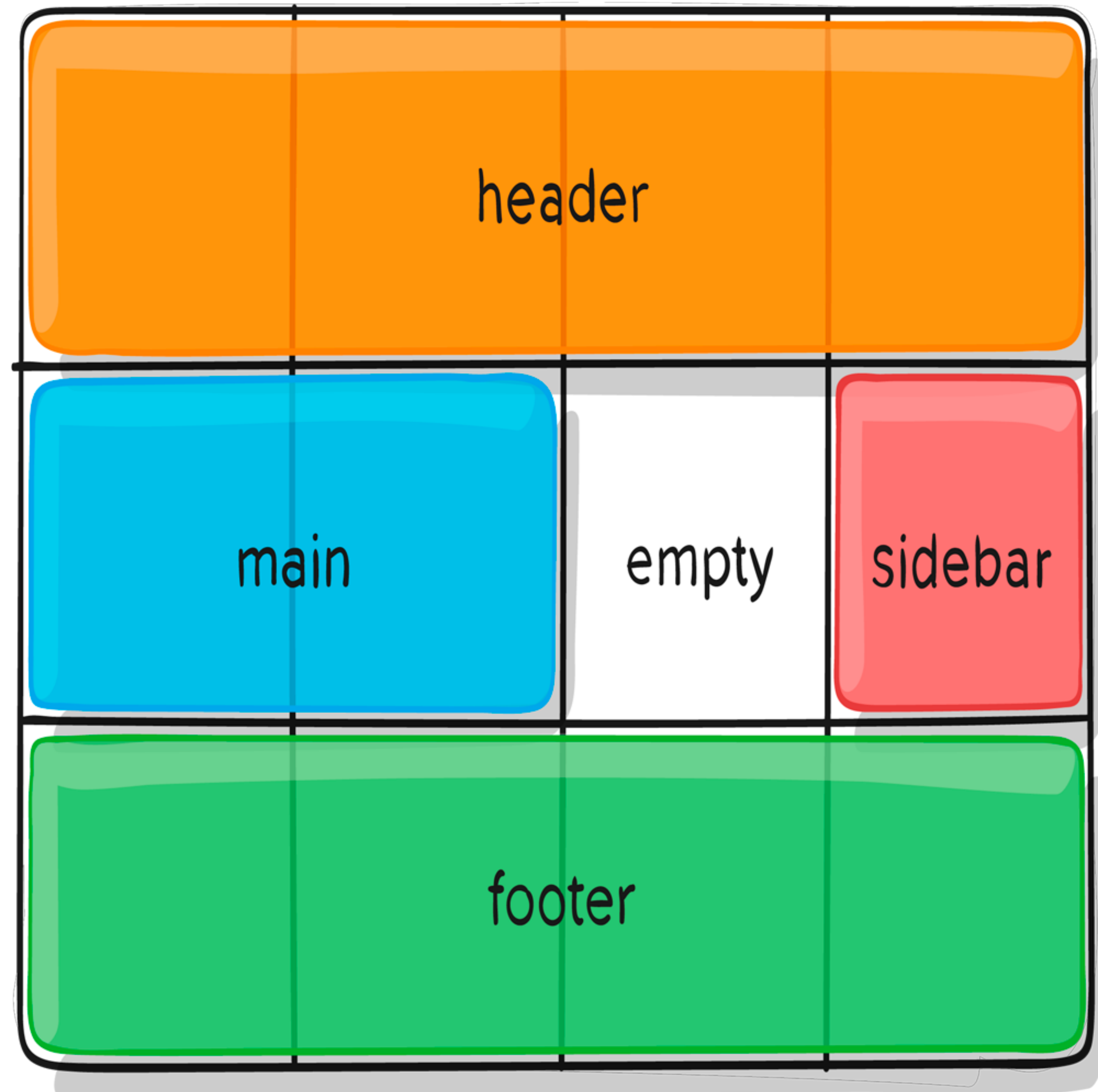
## grid-template-areas

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

```
.container {  
  display: grid;  
  grid-template-columns: 8em  
8em 8em 8em;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main    main    .    sidebar"  
    "footer footer footer footer";  
}
```

### grid-template-areas

- El código anterior creará una cuadrícula de cuatro columnas de ancho por tres filas de alto
- La fila superior completa será el área del encabezado
- La fila central estará compuesta por dos áreas principales, una celda vacía y un área de barra lateral
- La última fila es todo pie de página



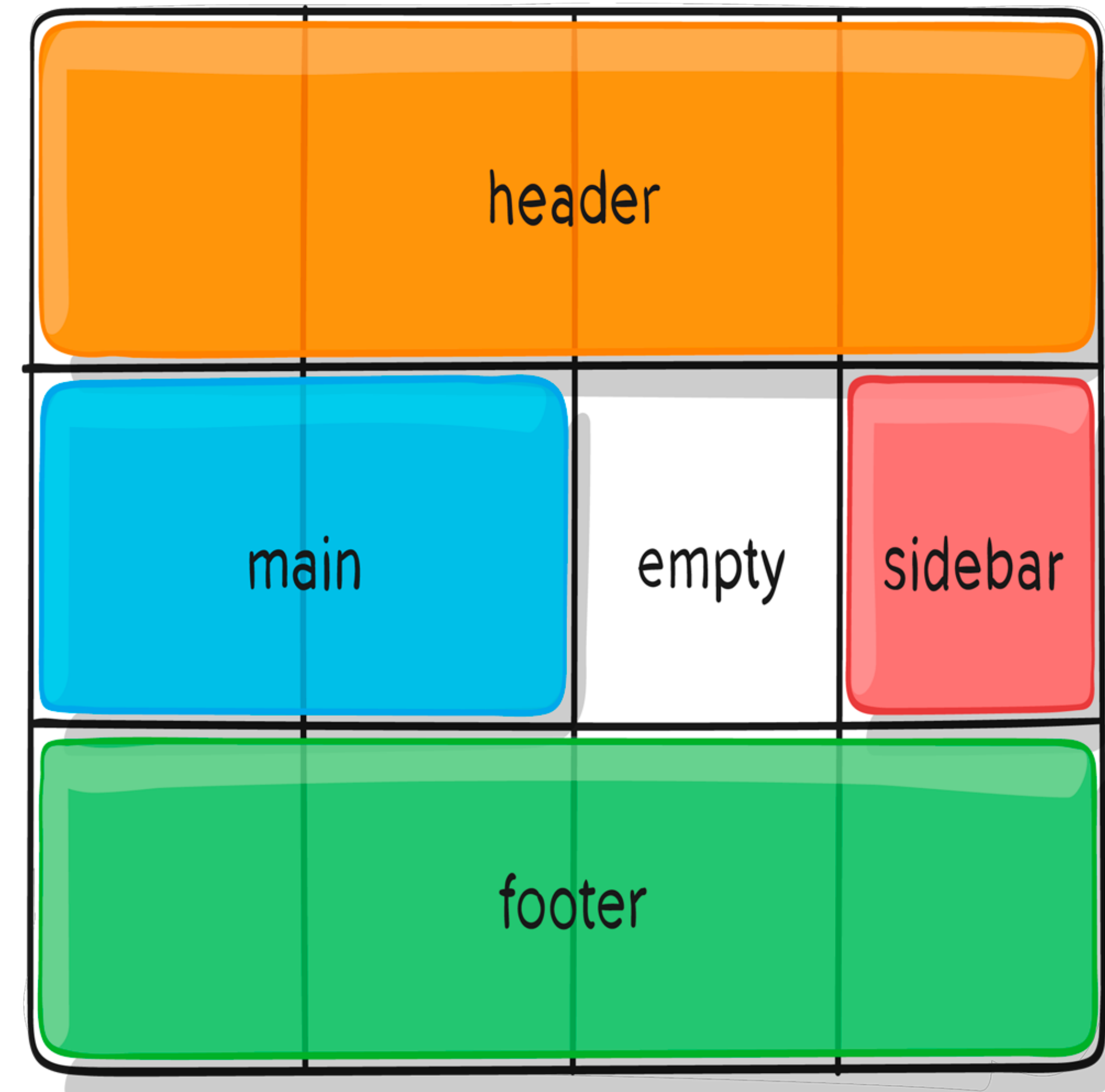
### grid-template-areas

- Cada fila de la declaración debe tener el mismo número de celdas
- Con esta sintaxis no se están nombrando líneas, solo áreas.

Las líneas en cada extremo de las áreas se nombran automáticamente.

Si el nombre de un área de cuadrícula es **equis**, el nombre de la línea de fila inicial y la línea de columna inicial del área será **equis-start**, y el nombre de su última línea de fila y última línea de columna será **equis-end**.

Esto significa que algunas líneas



pueden tener varios nombres, como la línea del extremo izquierdo en el ejemplo, que tendrá tres nombres: **header-start**, **main-start** y **footer-start**.



**grid-column-gap**

**grid-row-gap**

Especifica el tamaño de las líneas de la cuadrícula.  
Pueden pensarse como márgenes entre columnas / filas.

```
.container {  
  grid-template-columns: 10em 1fr 10em;  
  grid-template-rows: 15em auto 15em;  
  grid-column-gap: 1rem;  
  grid-row-gap: 1.5rem;  
}
```



### grid-gap

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

Si no se especifica un valor, se establece en el mismo valor para las dos propiedades

```
.container {  
  grid-template-columns: 10em 1fr 10em;  
  grid-template-rows: 15em auto 15em;  
  grid-gap: 1.5rem 1rem;  
}
```

El prefijo **grid-** se eliminará y la propiedad cambiará a **gap**.

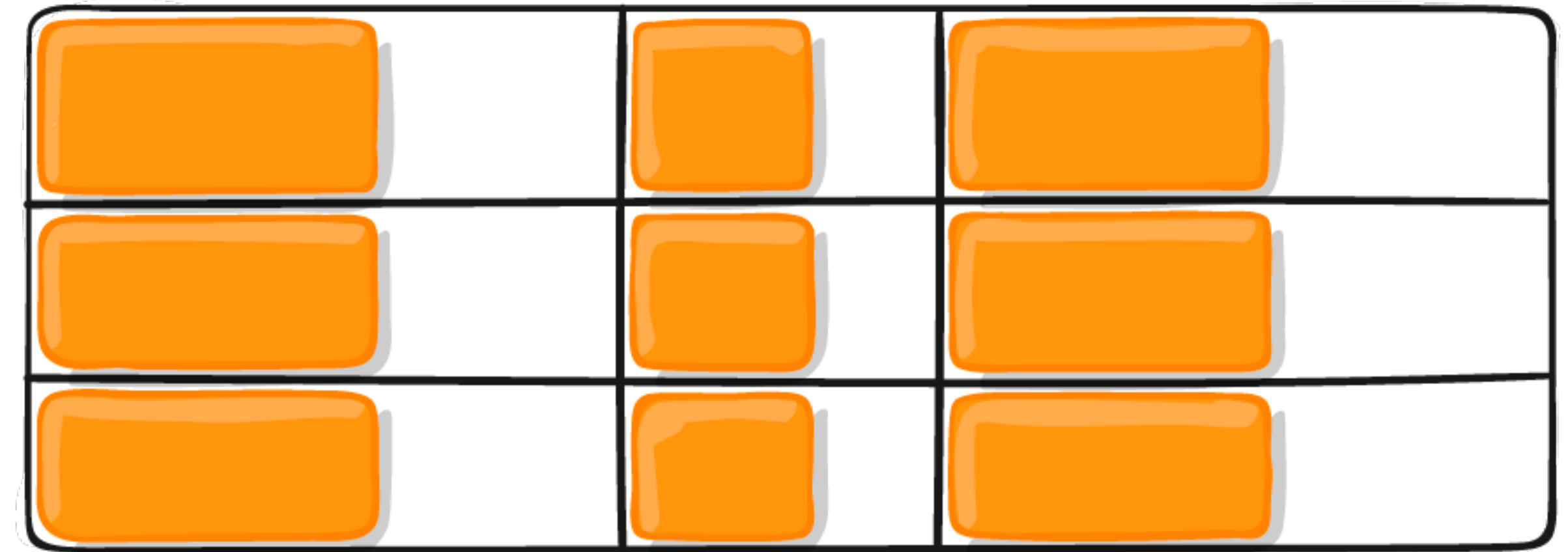
Sin prefijo ya es compatible con Chrome 68+, Safari 11.2 Release 50+ y Opera 54+

### justify-items

Alinea los elementos de la cuadrícula a lo largo del eje horizontal. Este valor aplica a todos los elementos de la cuadrícula.

Valores:

- start** los elementos son alineados al borde inicial de su celda
- end** los elementos son alineados al borde final de su celda
- center** alinea los elementos al centro de su celda
- stretch** llena todo el ancho de la celda (valor predeterminado)



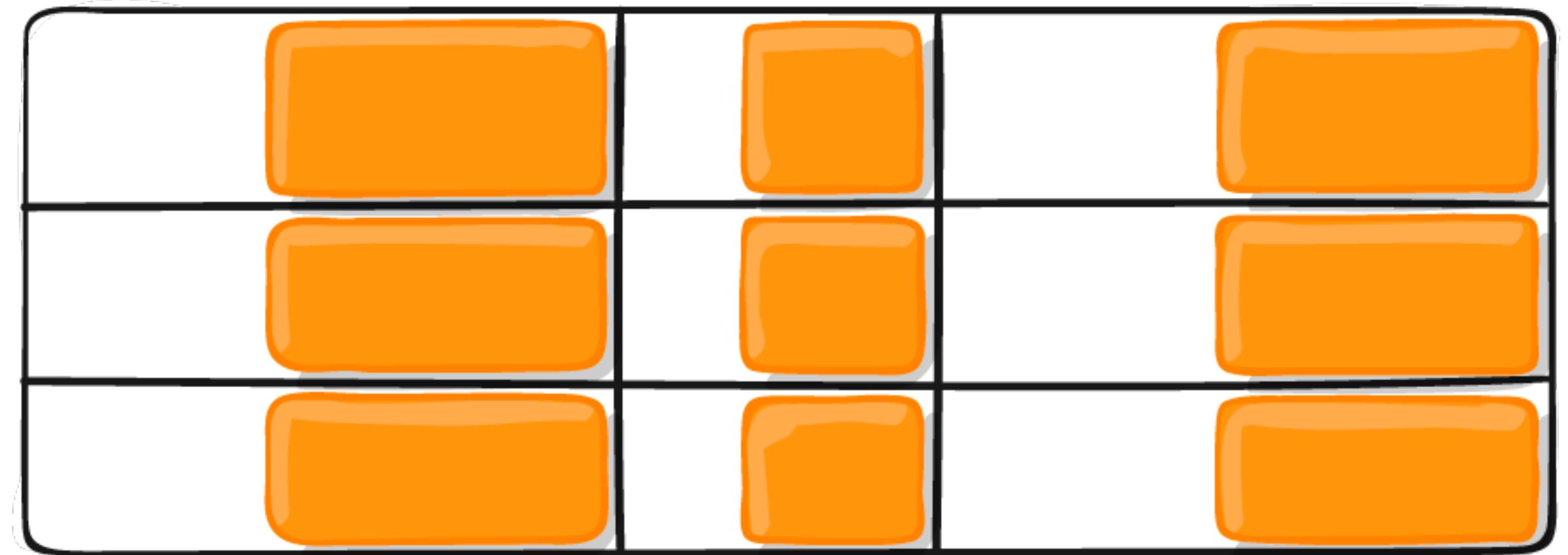
```
.container {  
  justify-items: start;  
}
```

### justify-items

Alinea los elementos de la cuadrícula a lo largo del eje horizontal. Este valor aplica a todos los elementos de la cuadrícula.

Valores:

- start** los elementos son alineados al borde inicial de su celda
- end** los elementos son alineados al borde final de su celda
- center** alinea los elementos al centro de su celda
- stretch** llena todo el ancho de la celda (valor predeterminado)



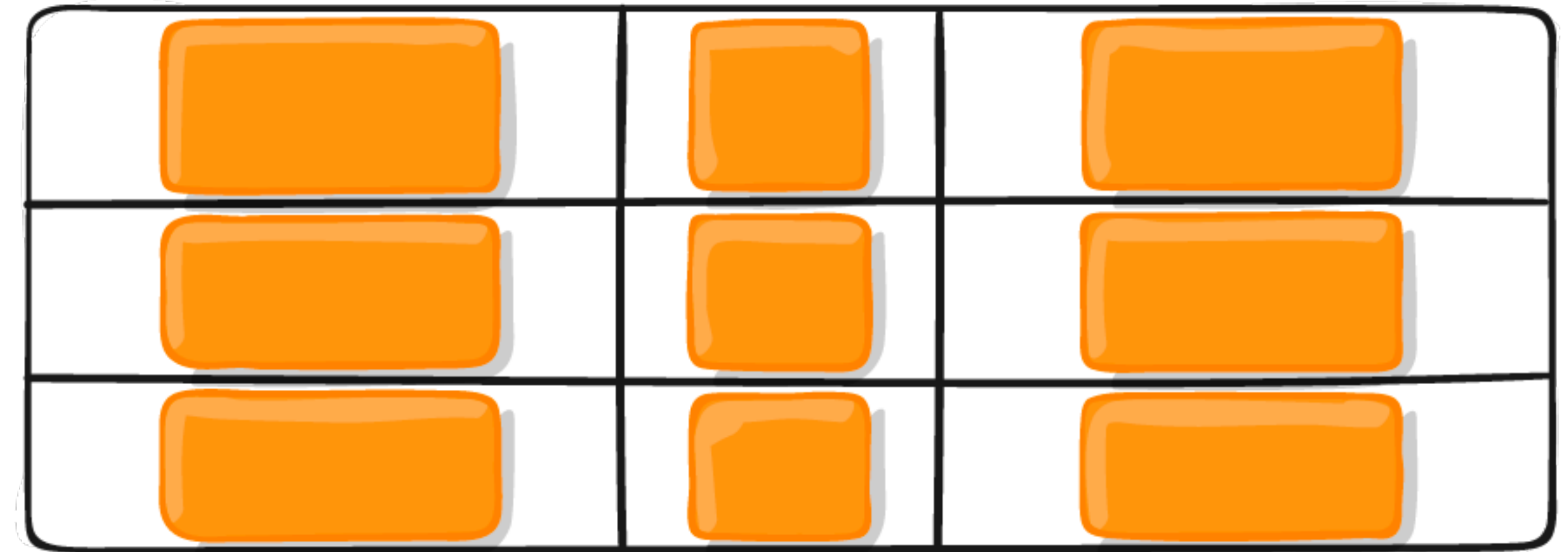
```
.container {  
  justify-items: end;  
}
```

### justify-items

Alinea los elementos de la cuadrícula a lo largo del eje horizontal. Este valor aplica a todos los elementos de la cuadrícula.

Valores:

- start** los elementos son alineados al borde inicial de su celda
- end** los elementos son alineados al borde final de su celda
- center** alinea los elementos al centro de su celda
- stretch** llena todo el ancho de la celda (valor predeterminado)



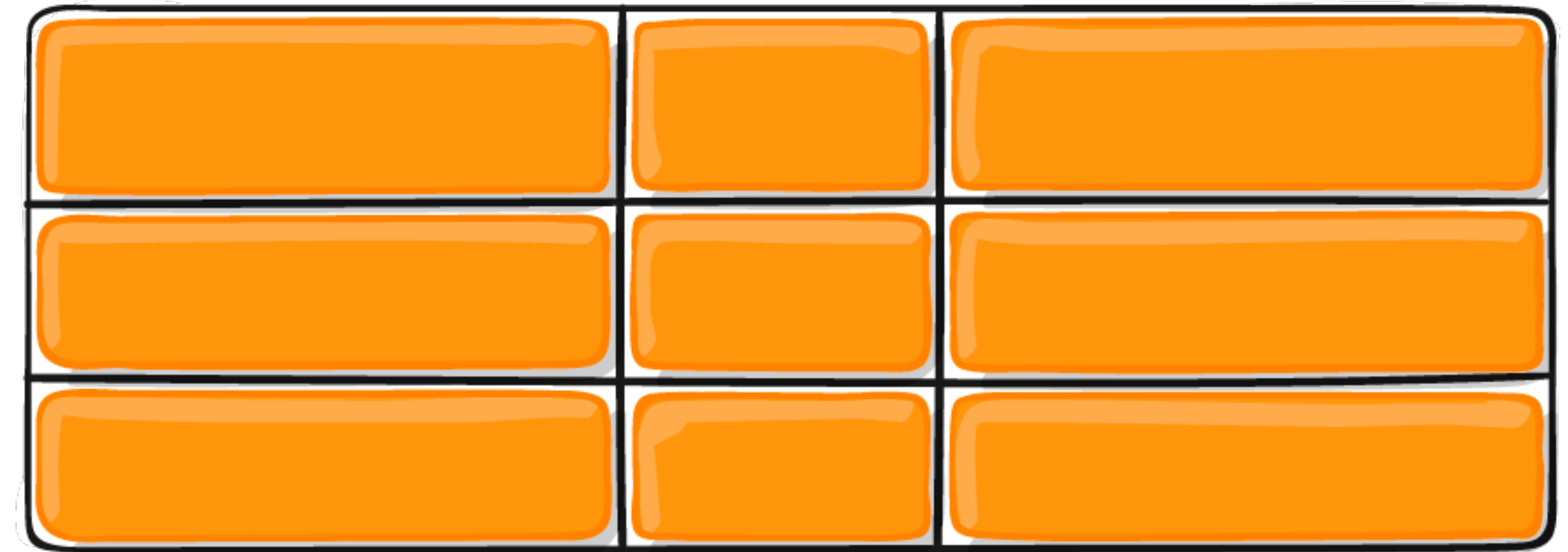
```
.container {  
  justify-items: center;  
}
```

### justify-items

Alinea los elementos de la cuadrícula a lo largo del eje horizontal. Este valor aplica a todos los elementos de la cuadrícula.

Valores:

- start** los elementos son alineados al borde inicial de su celda
- end** los elementos son alineados al borde final de su celda
- center** alinea los elementos al centro de su celda
- stretch** llena todo el ancho de la celda (valor predeterminado)



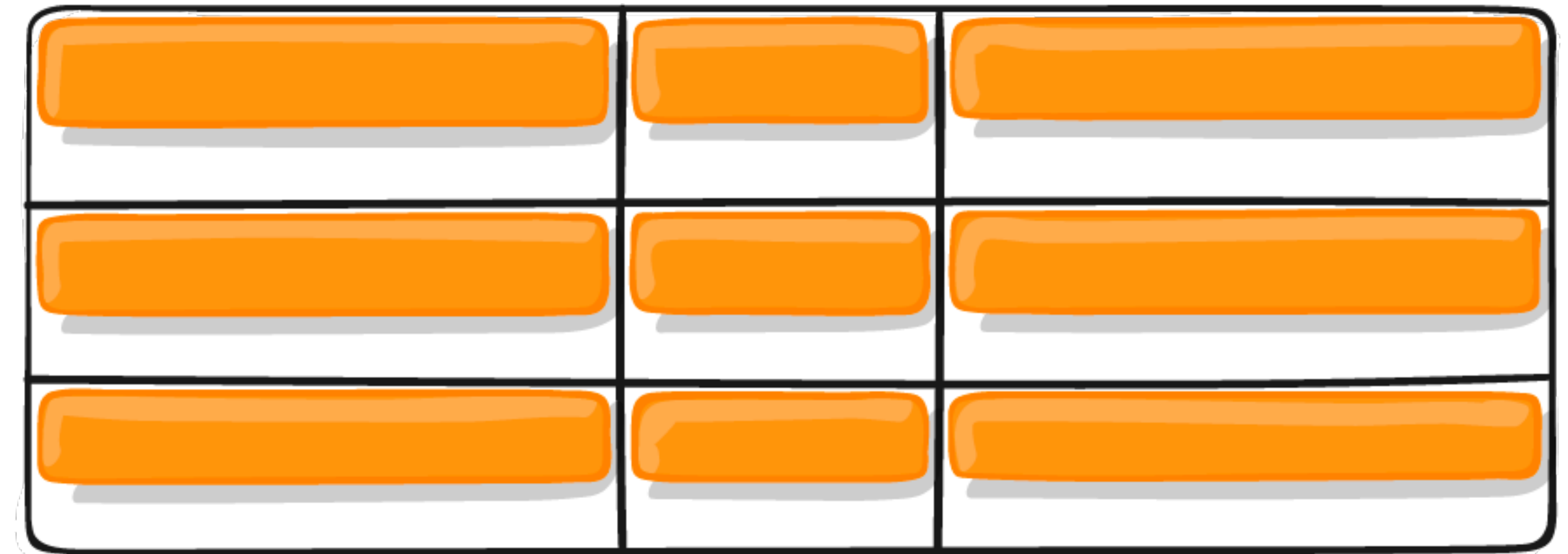
```
.container {  
  justify-items: stretch;  
}
```

### align-items

Alinea los elementos de la cuadrícula a lo largo del eje vertical. Este valor se aplica a todos los elementos de la cuadrícula.

Valores:

- start** los elementos son alineados al borde inicial de su celda
- end** los elementos son alineados al borde final de su celda
- center** alinea los elementos al centro de su celda
- stretch** llena todo el alto de la celda (valor predeterminado)



```
.container {  
  align-items: start;  
}
```

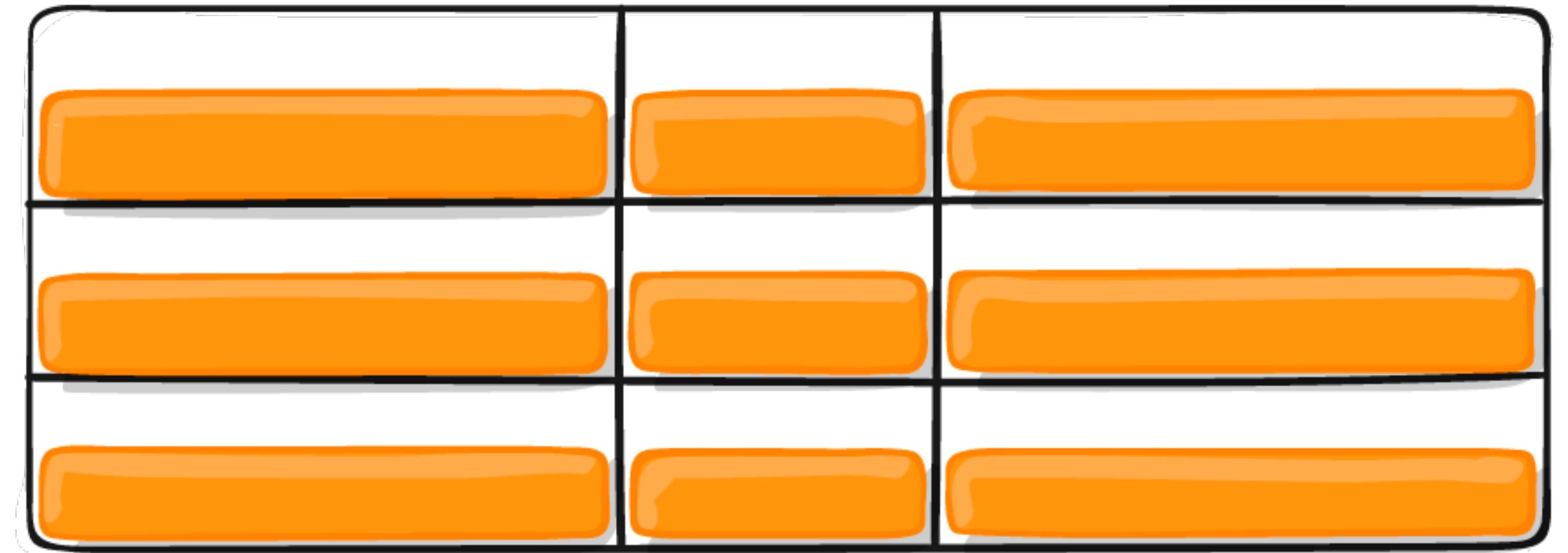


## align-items

Alinea los elementos de la cuadrícula a lo largo del eje vertical. Este valor se aplica a todos los elementos de la cuadrícula.

Valores:

- start** los elementos son alineados al borde inicial de su celda
- end** los elementos son alineados al borde final de su celda
- center** alinea los elementos al centro de su celda
- stretch** llena todo el alto de la celda (valor predeterminado)



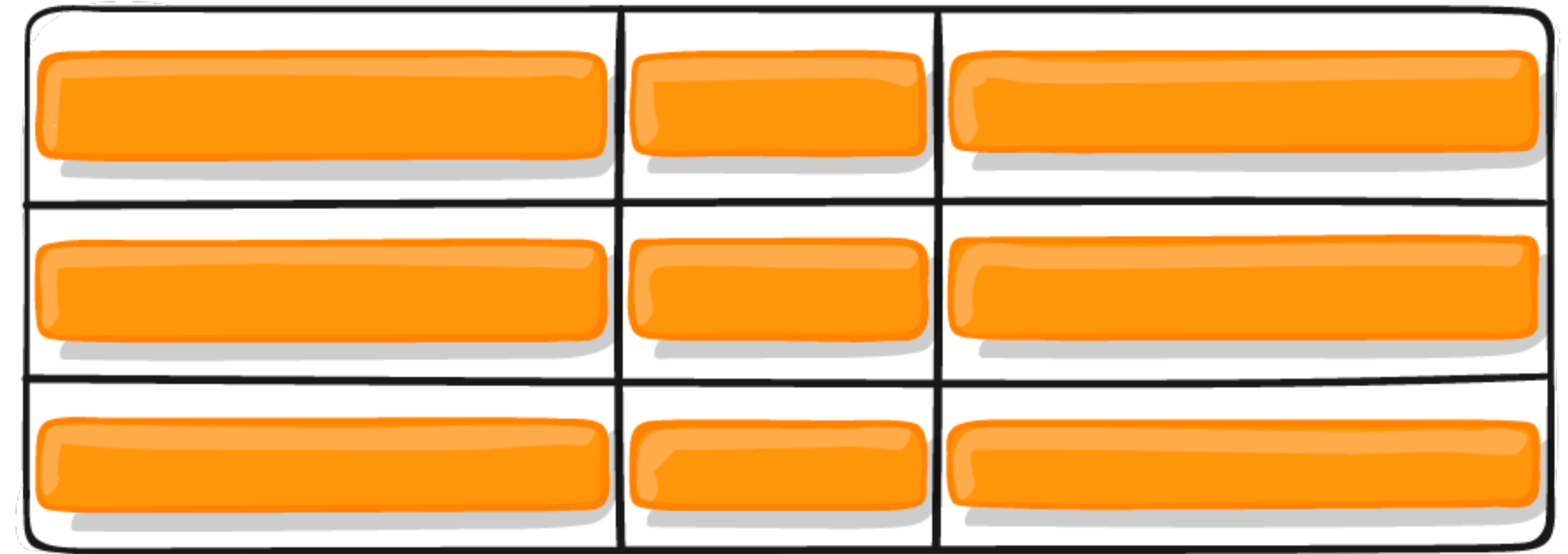
```
.container {  
  align-items: end;  
}
```

### align-items

Alinea los elementos de la cuadrícula a lo largo del eje vertical. Este valor se aplica a todos los elementos de la cuadrícula.

Valores:

- start** los elementos son alineados al borde inicial de su celda
- end** los elementos son alineados al borde final de su celda
- center** alinea los elementos al centro de su celda
- stretch** llena todo el alto de la celda (valor predeterminado)



```
.container {  
  align-items: center;  
}
```

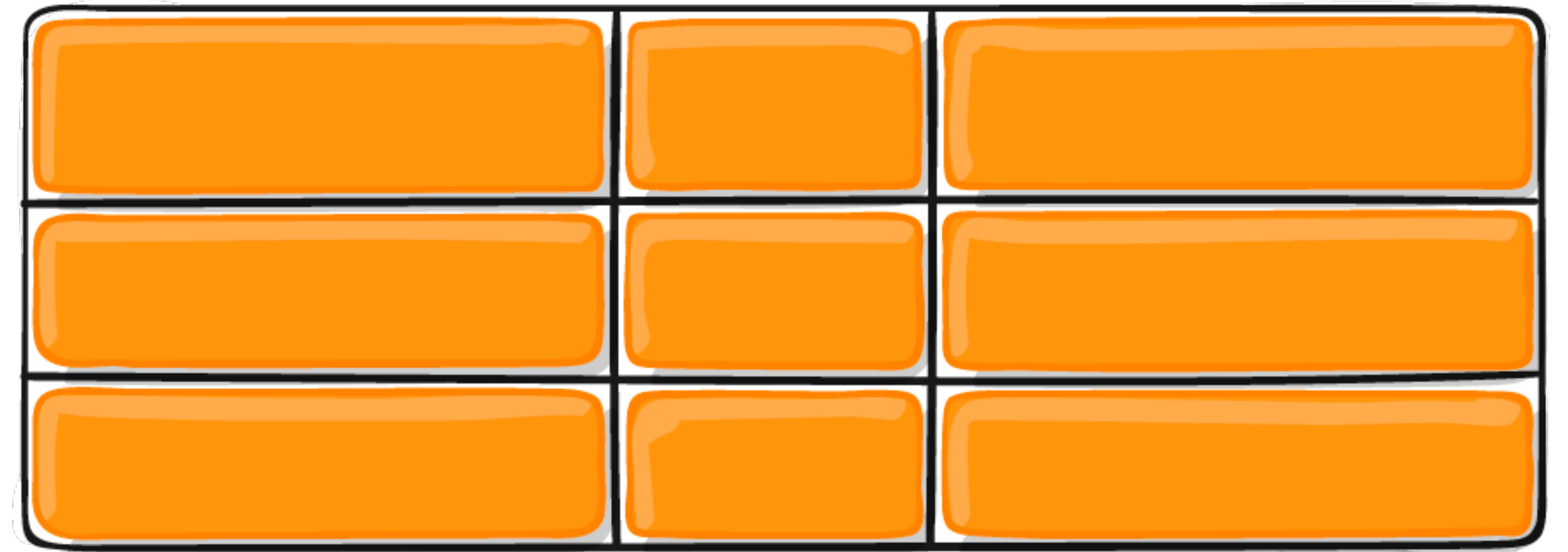


### align-items

Alinea los elementos de la cuadrícula a lo largo del eje vertical. Este valor se aplica a todos los elementos de la cuadrícula.

Valores:

- start** los elementos son alineados al borde inicial de su celda
- end** los elementos son alineados al borde final de su celda
- center** alinea los elementos al centro de su celda
- stretch** llena todo el alto de la celda (valor predeterminado)



```
.container {  
  align-items: stretch;  
}
```

### place-items

Establece las propiedades **align-items** y **justify-items** en una sola declaración

Valores: **<align-items>** / **<justify-items>**

```
.container {  
  place-items: center center;  
}
```

# justify-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje horizontal, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

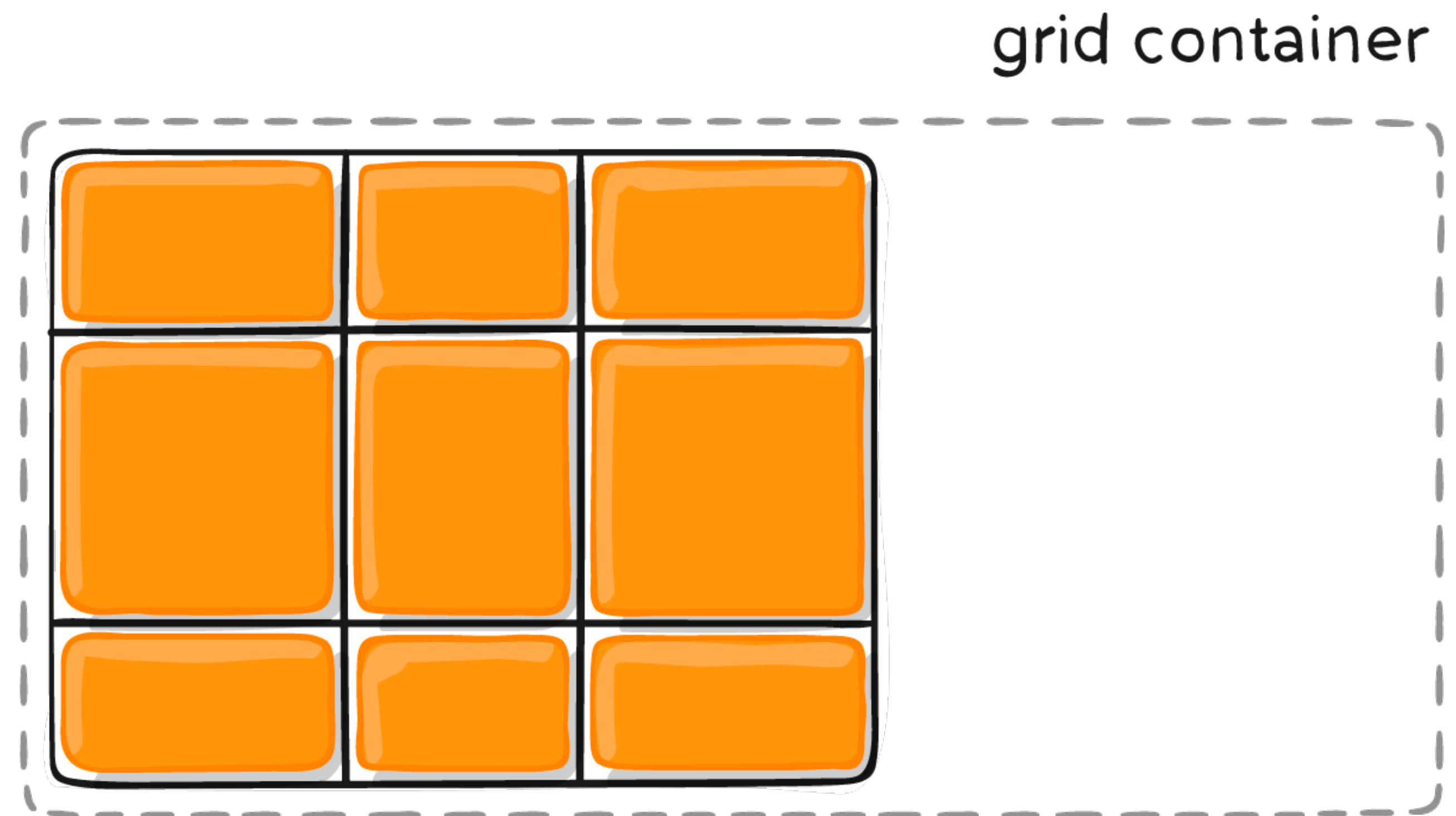
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



## justify-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje horizontal, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

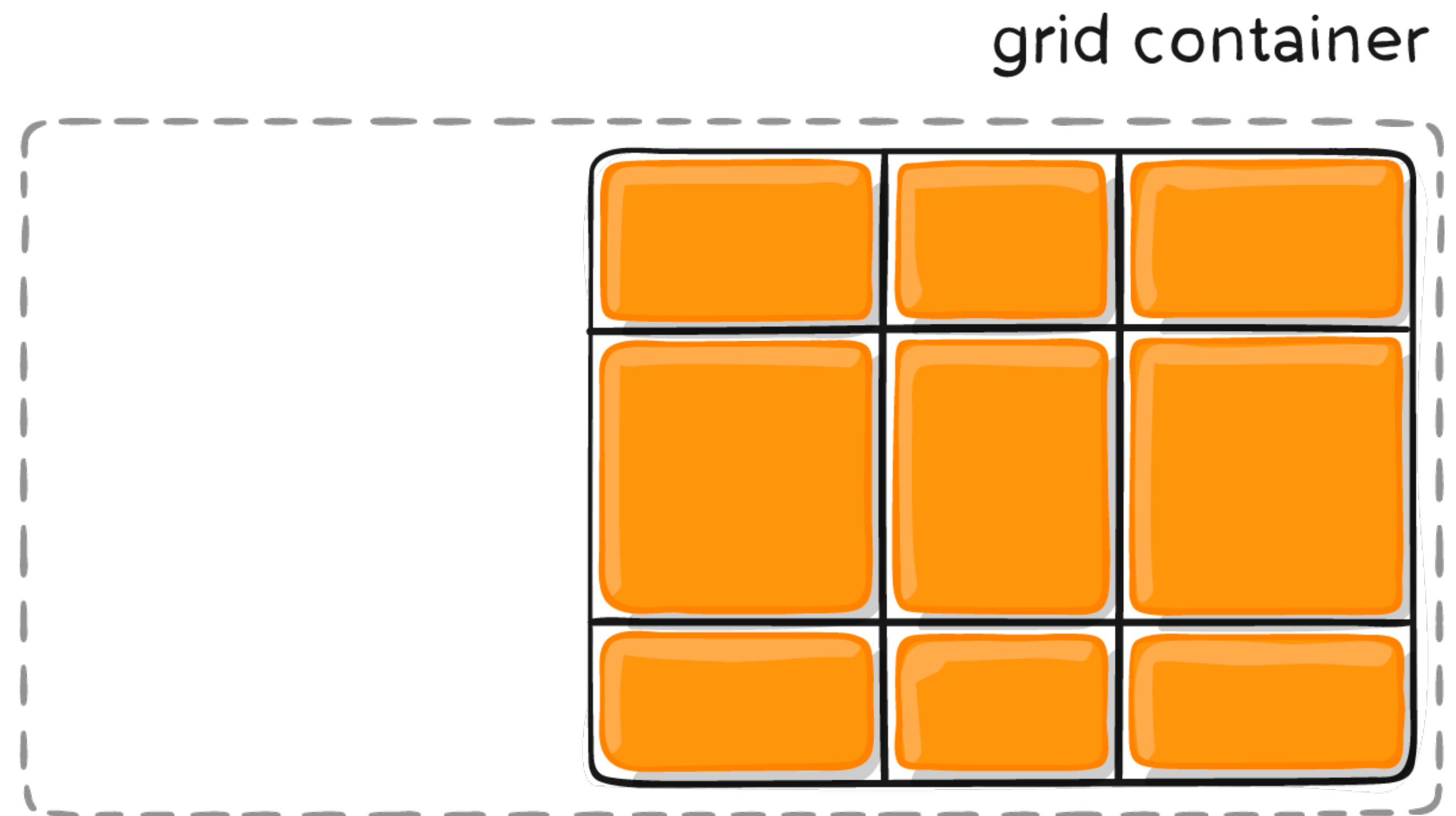
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



# justify-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje horizontal, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

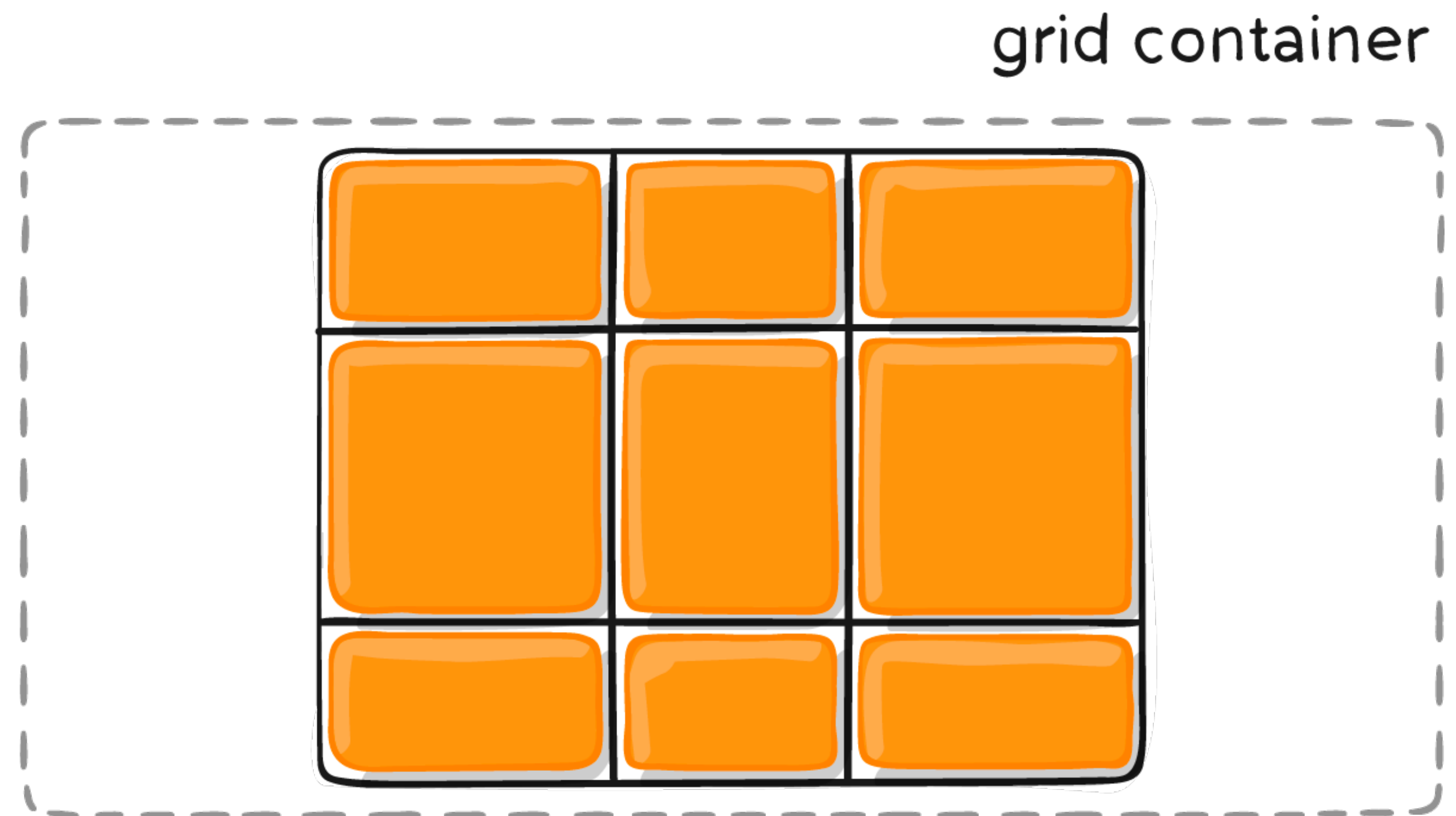
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



# justify-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje horizontal, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

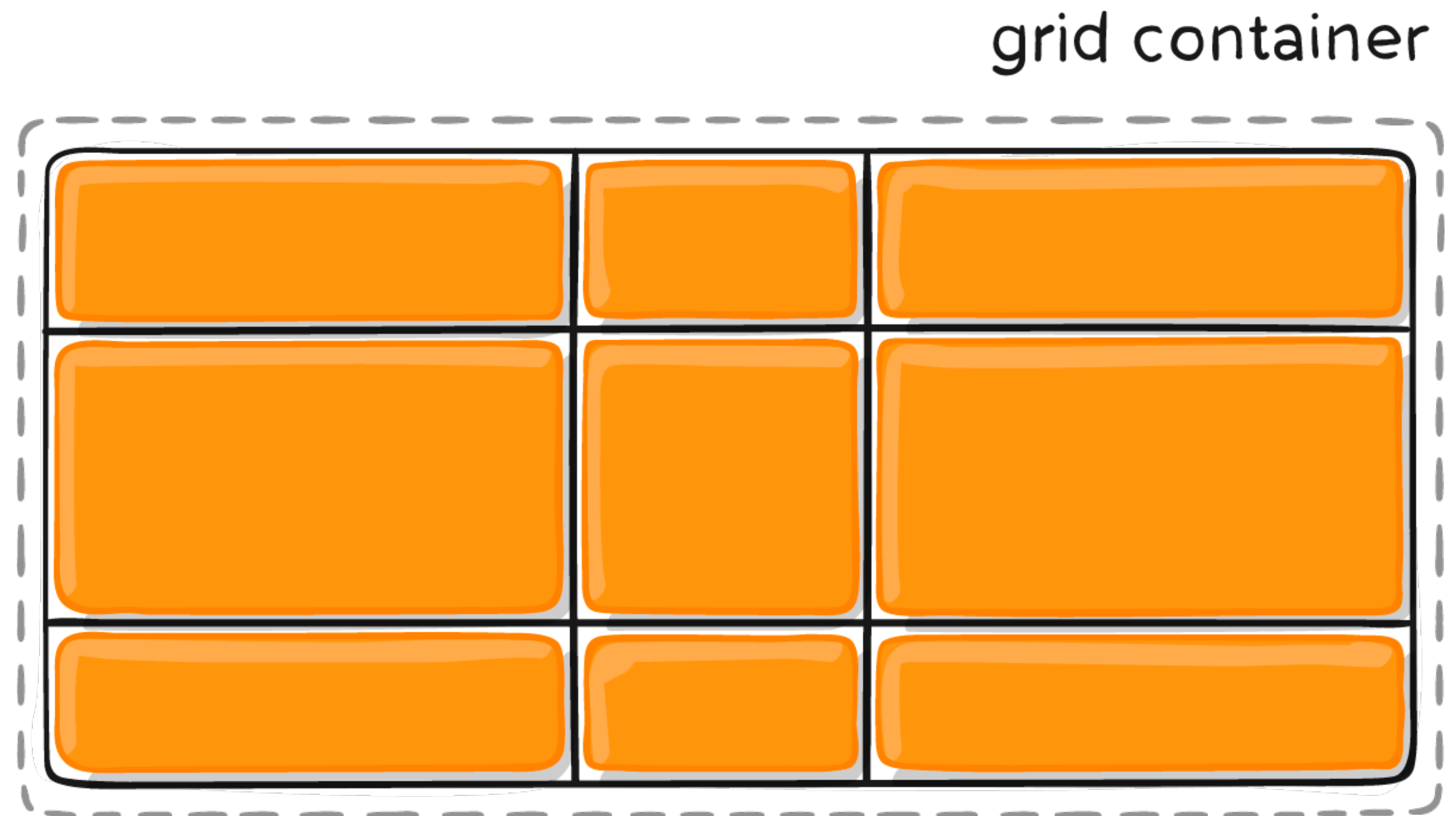
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**





## justify-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje horizontal, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

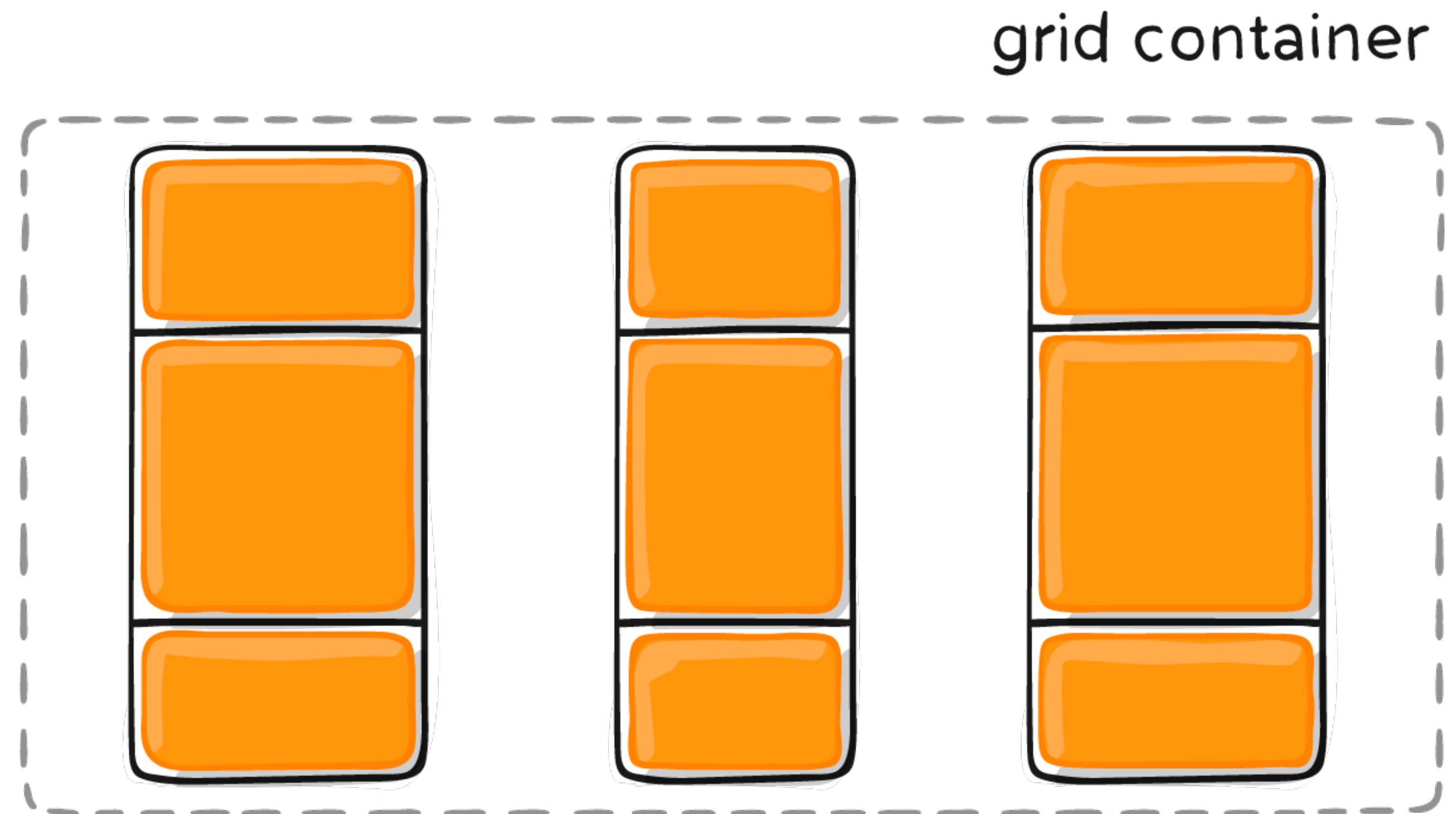
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



## justify-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje horizontal, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

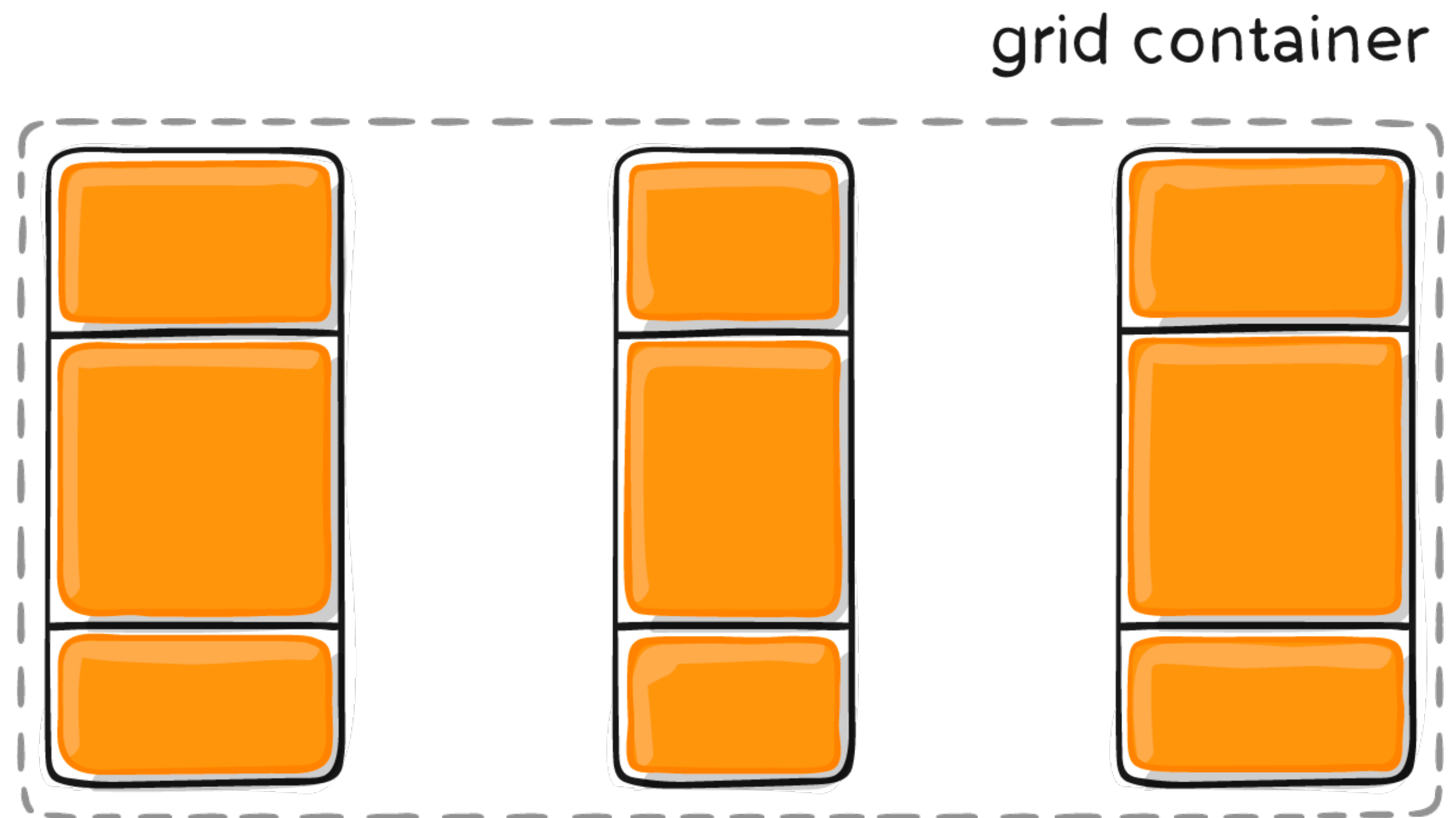
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**





## justify-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje horizontal, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

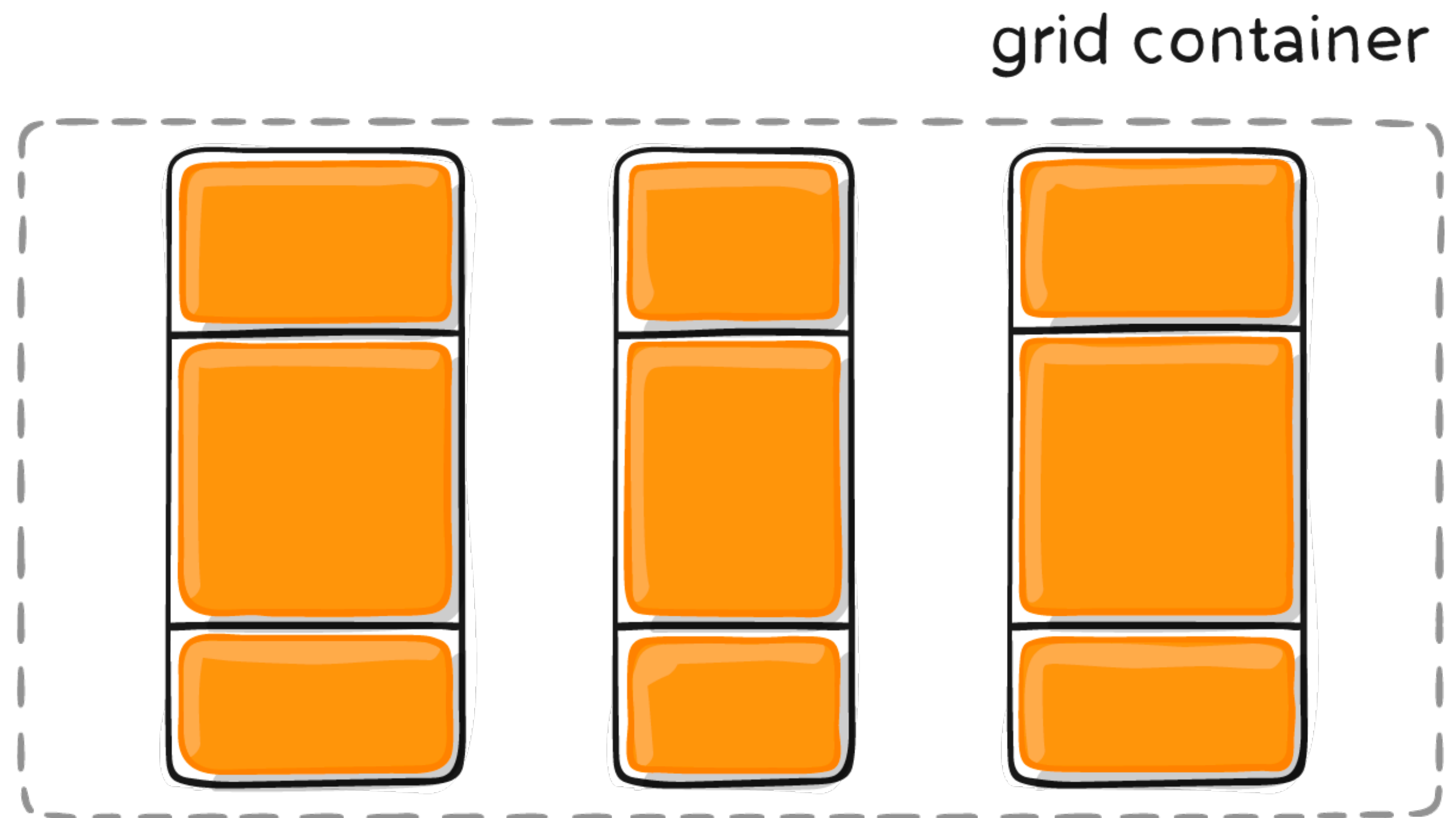
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



# align-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje vertical, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

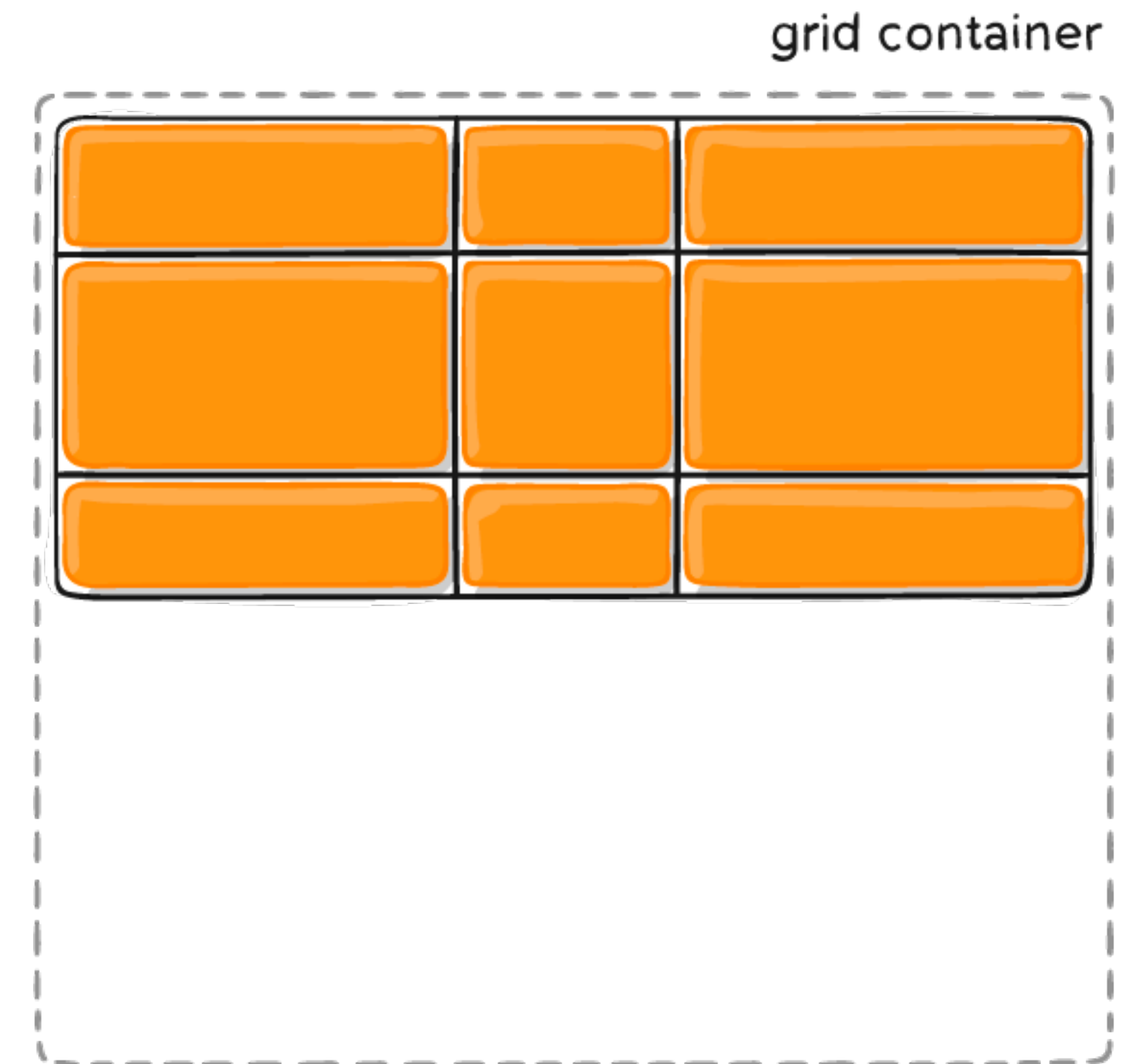
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



# align-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje vertical, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

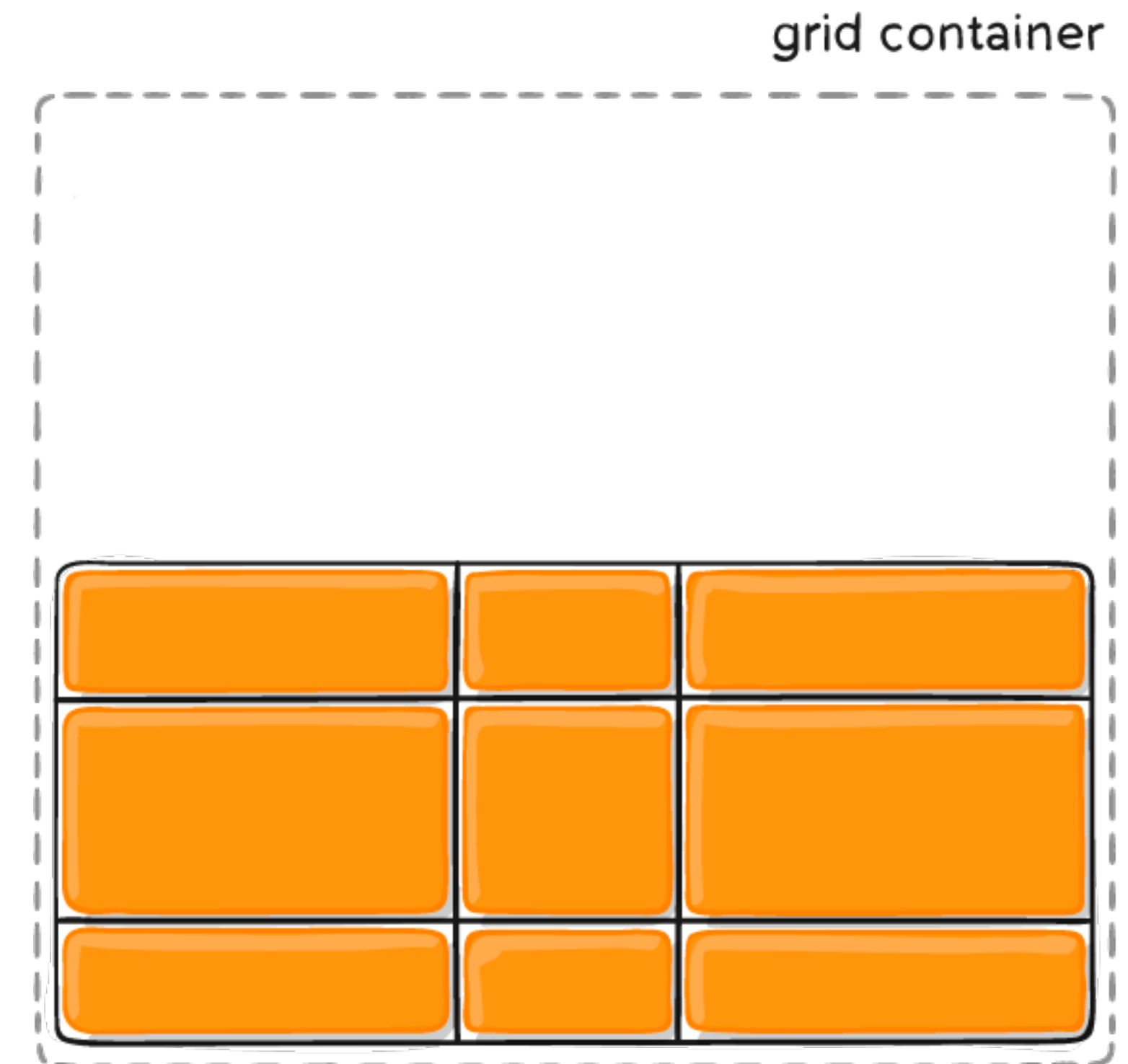
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



# align-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje vertical, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

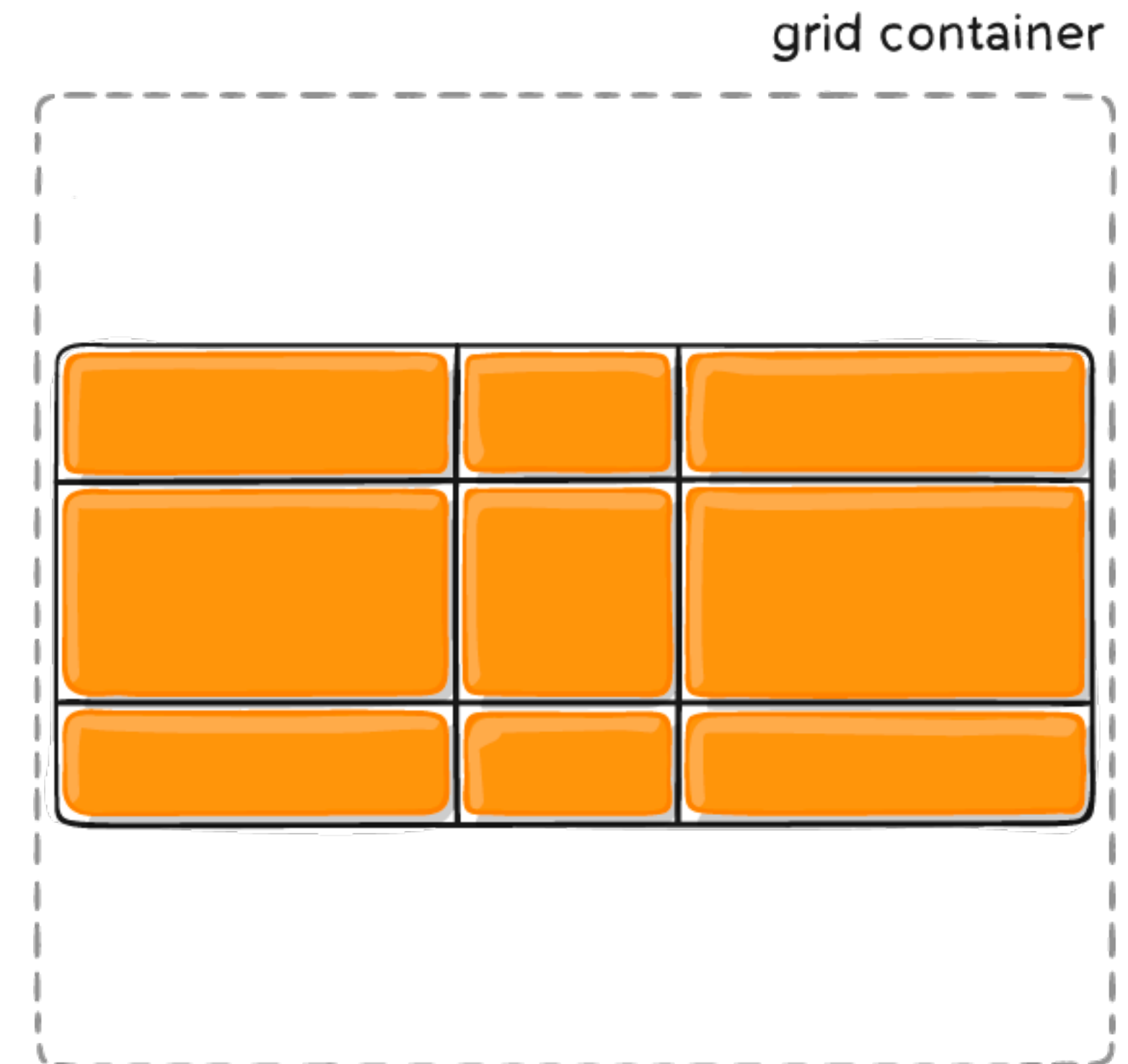
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



# align-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje vertical, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

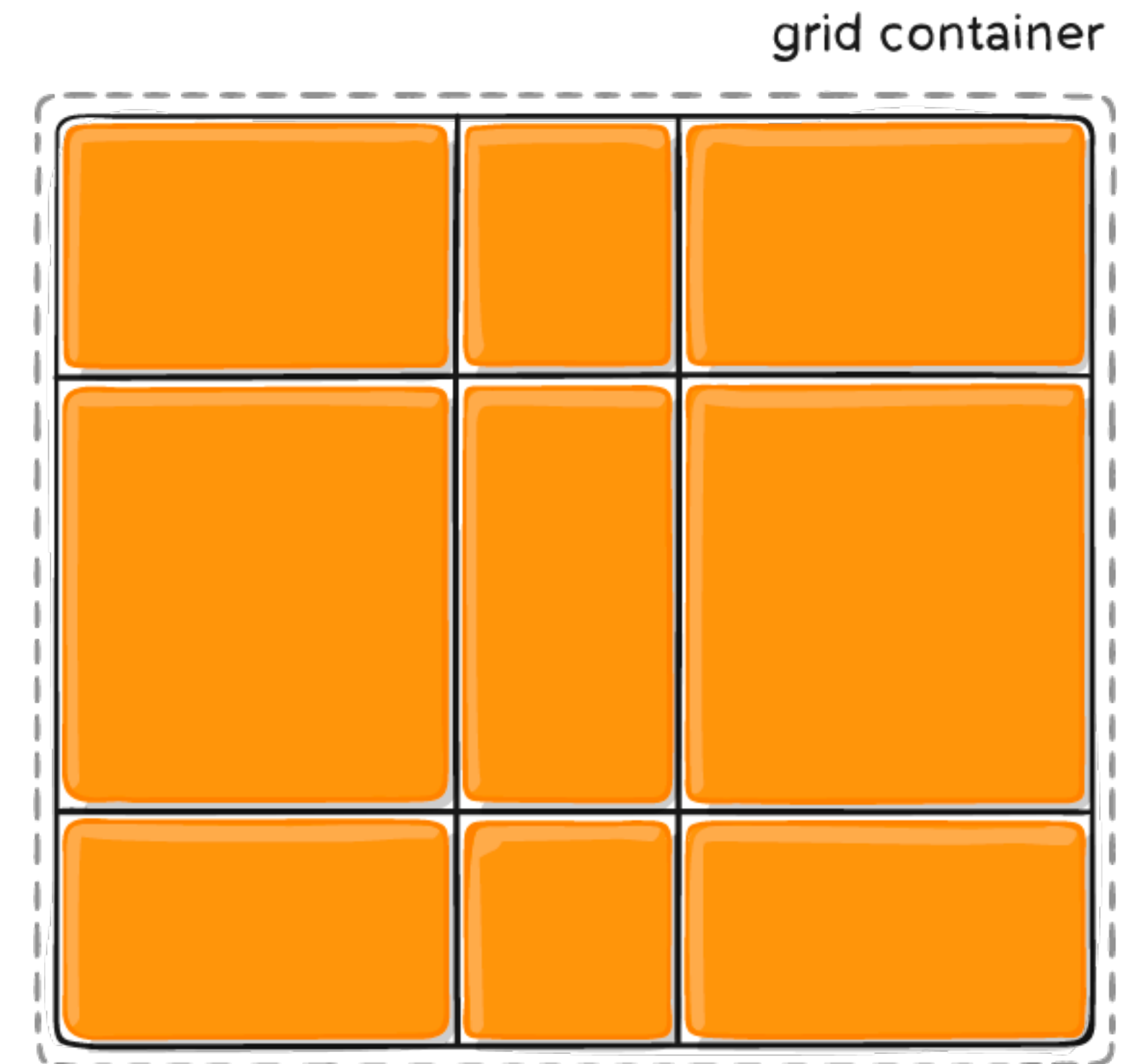
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



# align-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje vertical, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

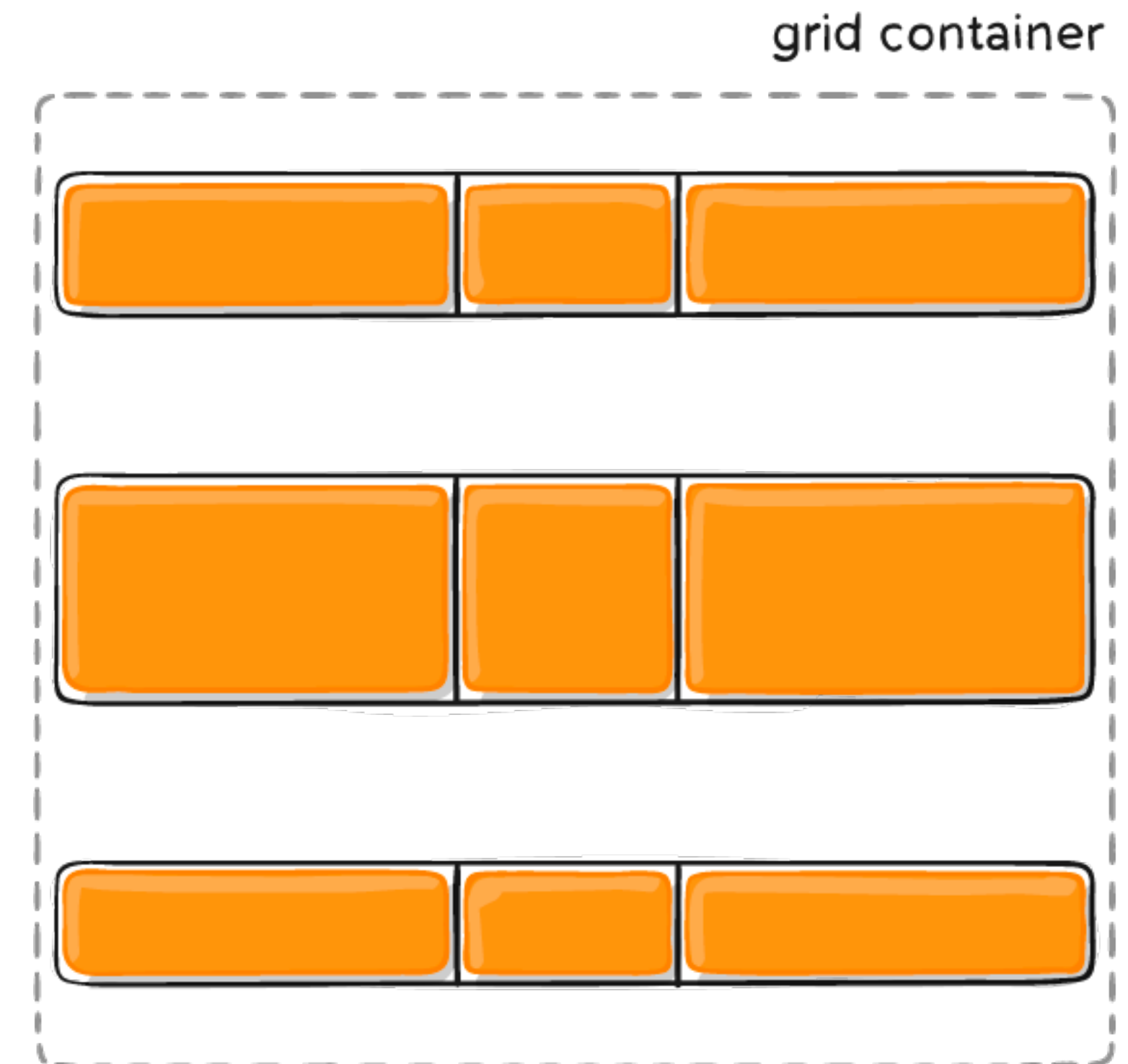
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**





# align-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje vertical, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

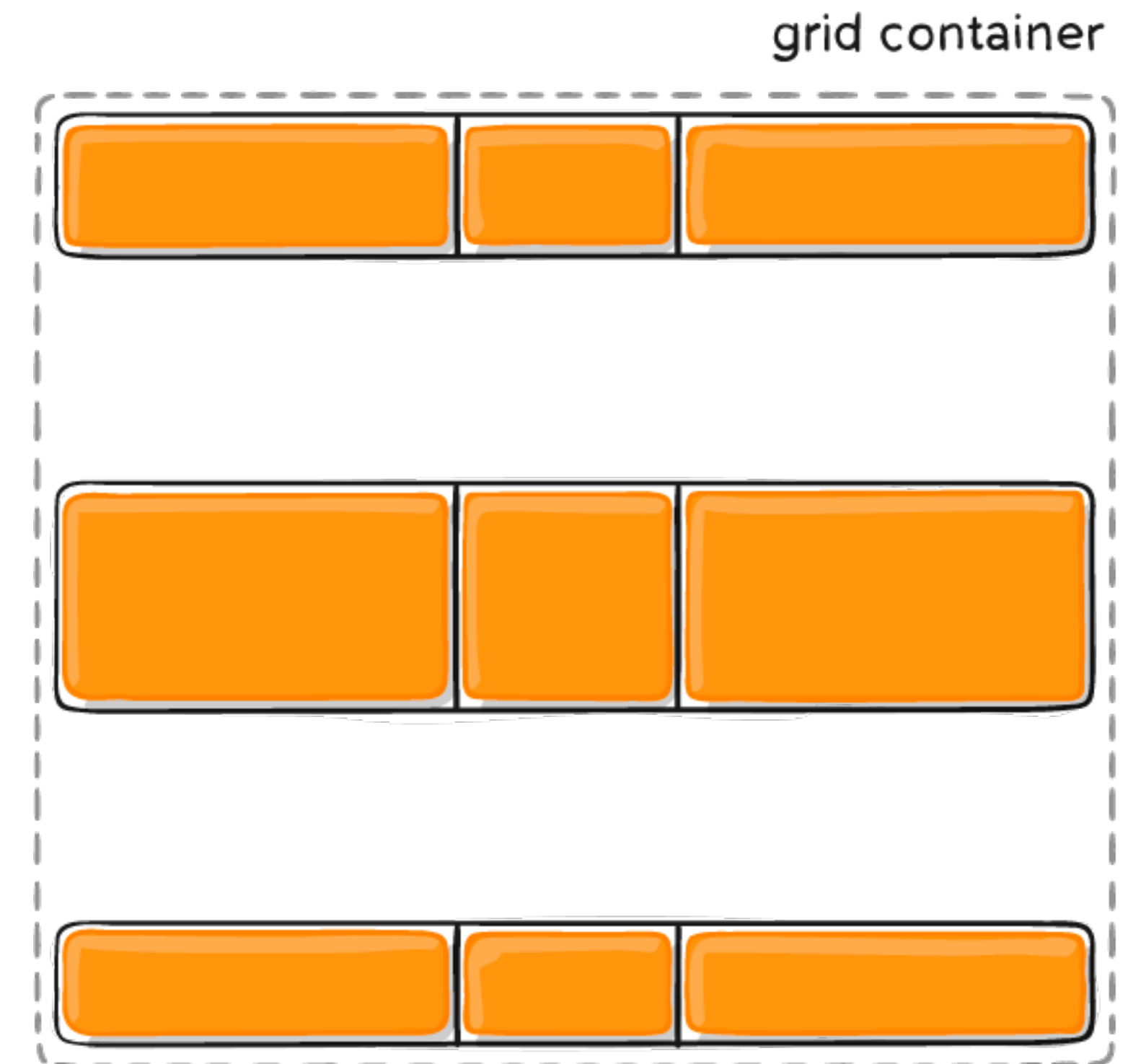
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**



## align-content

Alineación de una cuadrícula dentro de su contenedor, a lo largo del eje vertical, cuando el tamaño de la cuadrícula es menor que el de su contenedor. Puede suceder si los elementos de la cuadrícula son dimensionados con unidades no flexibles como **px**.

Valores:

**start**

**end**

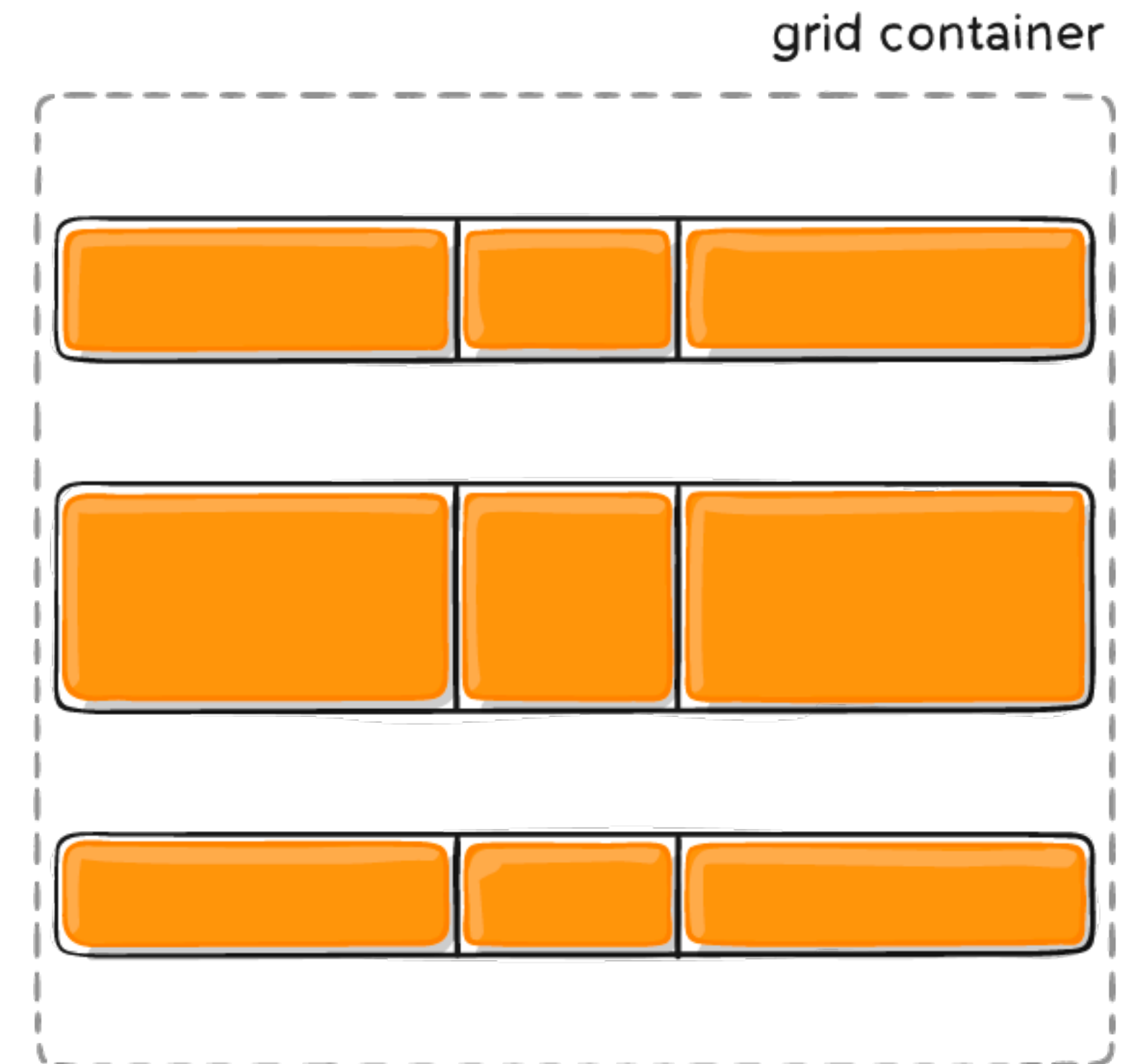
**center**

**stretch**

**space-around**

**space-between**

**space-evenly**





**grid-auto-columns**

**grid-auto-rows**

Especifica el tamaño de las pistas de cuadrícula autogeneradas (también conocidas como pistas de cuadrícula implícitas).

Las pistas implícitas se crean cuando hay más elementos que celdas en la cuadrícula o cuando un elemento de cuadrícula se coloca fuera de la cuadrícula explícita.

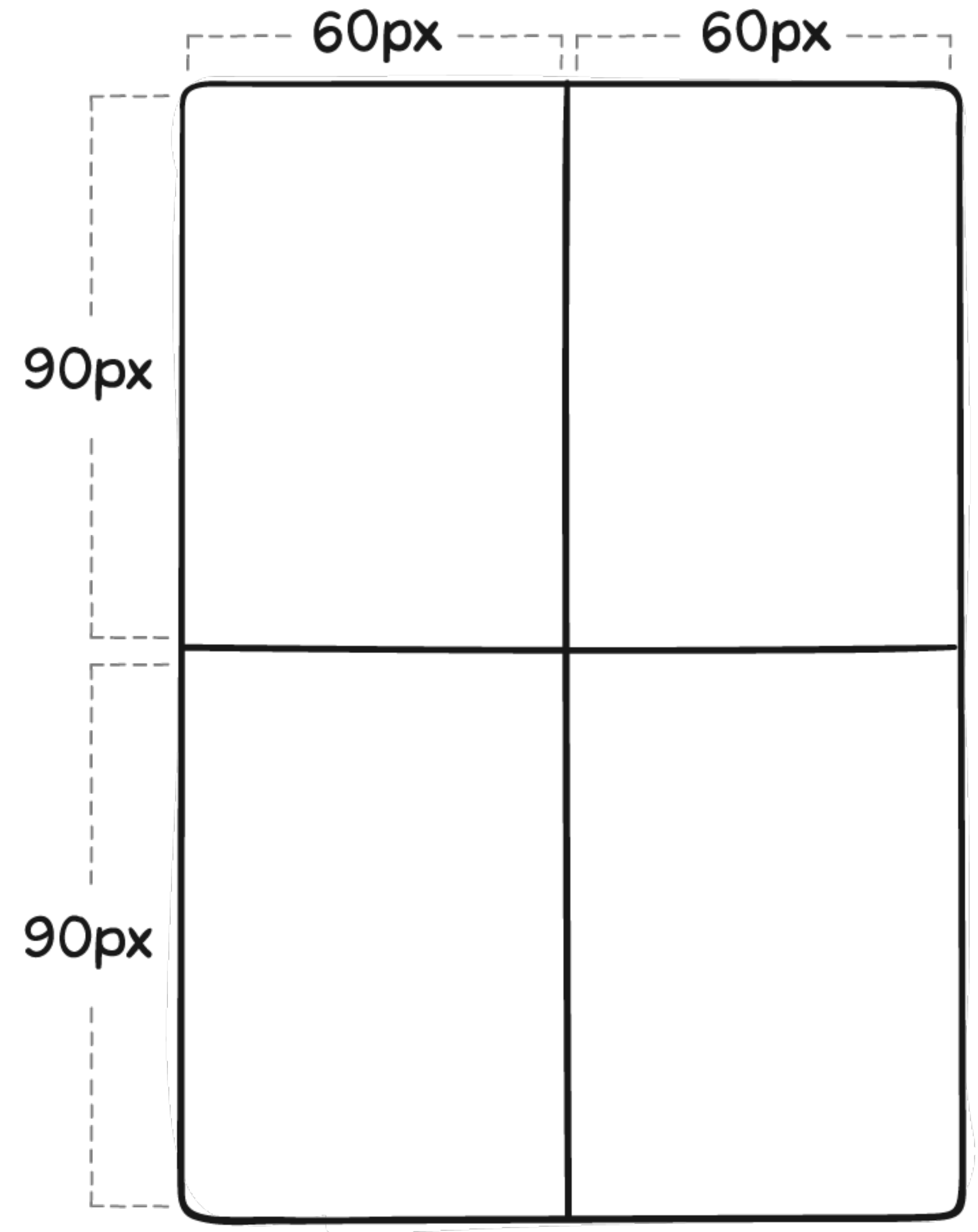
```
.container {  
  grid-auto-columns: <track-size> ...;  
  grid-auto-rows: <track-size> ...;  
}
```

**grid-auto-columns**

**grid-auto-rows**

Para ilustrar cómo se crean las pistas de cuadrícula implícitas, veamos:

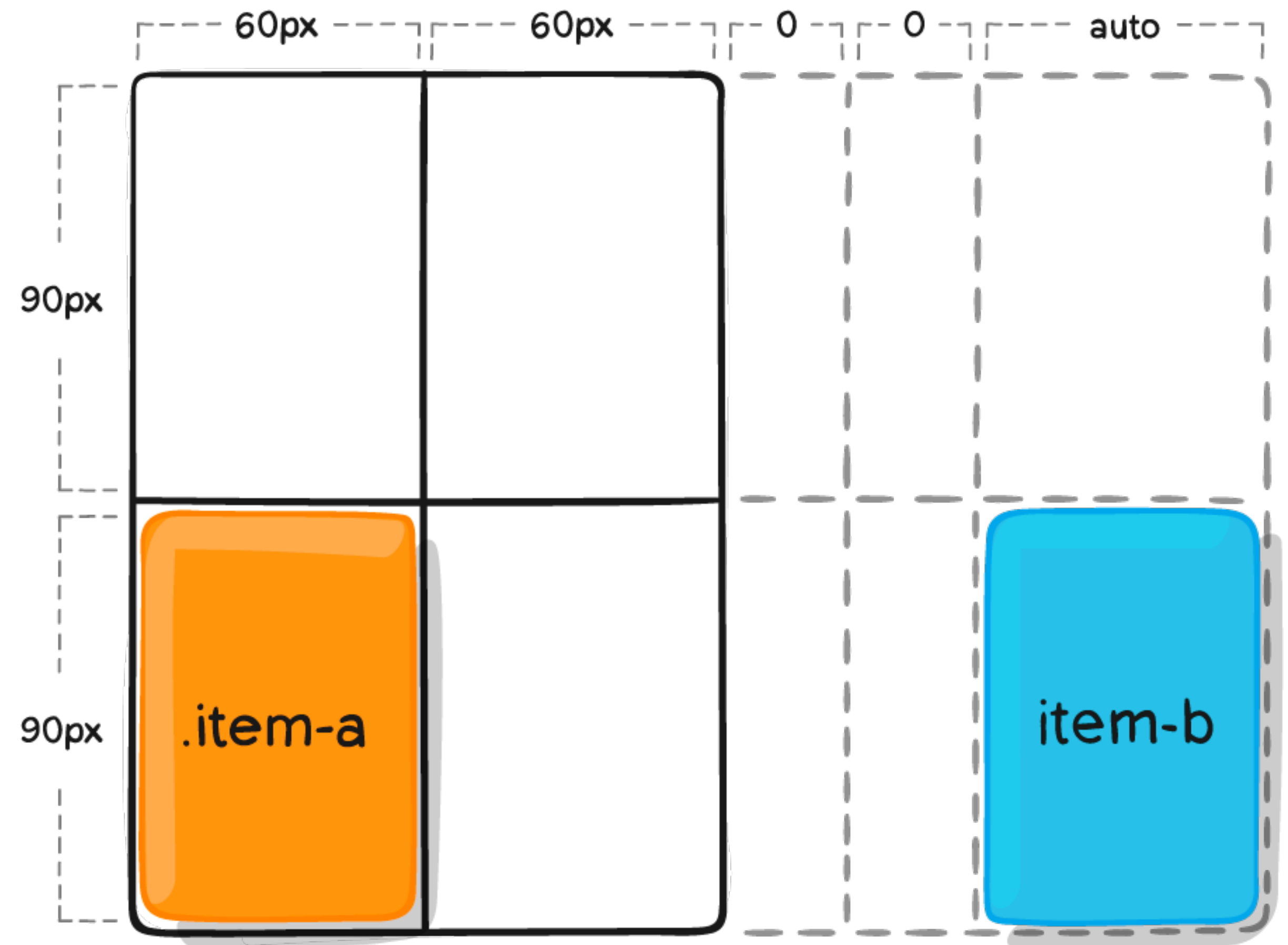
```
.container {  
  grid-template-columns:  
    60px 60px;  
  grid-template-rows:  
    60px 60px;  
}
```



## CSS Grid / Propiedades / Contenedor

Imaginemos que colocamos elementos de la cuadrícula de esta manera:

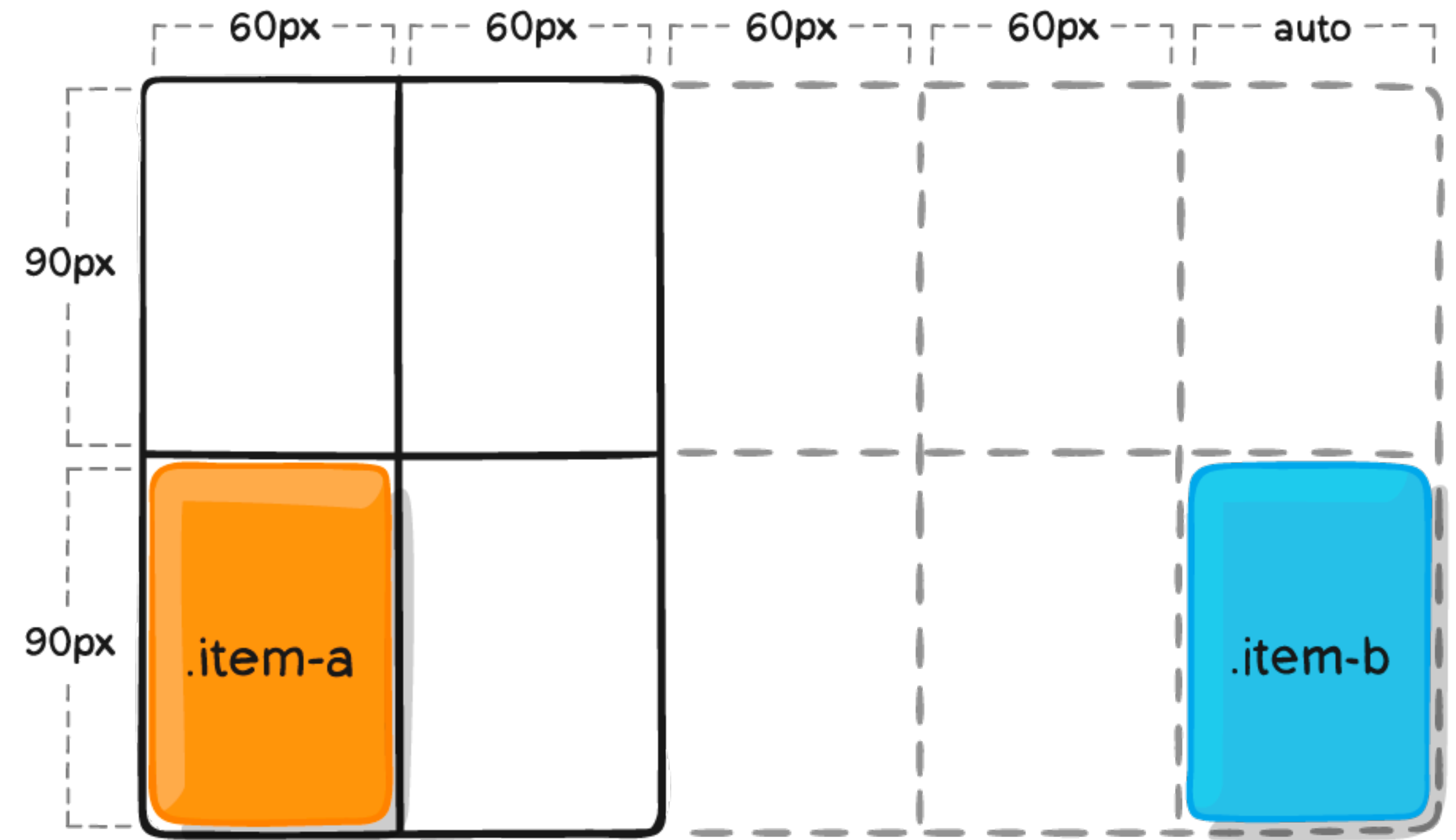
```
.item-a {  
  grid-column: 1 / 2;  
  grid-row: 2 / 3;  
}  
  
.item-b {  
  grid-column: 5 / 6;  
  grid-row: 2 / 3;  
}
```



## CSS Grid / Propiedades / Contenedor

Se Indicó que **.item-b** comenzara en **línea de columna 5** y terminara en **línea de columna 6**. Las líneas no existen: se crean pistas implícitas con ancho **cero** para rellenar huecos. Se especifican los anchos con **grid-auto-columns**:

```
.container {  
  grid-auto-columns: 60px;  
}
```



### **grid-auto-flow**

Si hay elementos de cuadrícula no emplazados explícitamente, el algoritmo de colocación automática se activa para colocarlos automáticamente. Esta propiedad controla cómo funciona ese algoritmo.

Valores:

- row** completa cada fila por turno, agregando nuevas filas según sea necesario (predeterminado)
- column** completa cada columna a su vez, agregando nuevas columnas según sea necesario
- dense** intenta rellenar agujeros en la cuadrícula si aparecen elementos más pequeños más adelante

# place-content

Establece las propiedades **align-content** y **justify-content** en una sola declaración

Valores: **<align-content>** / **<justify-content>**

```
.container {  
  place-content: center center;  
}
```

Si se omite el segundo valor, el primer valor se asigna a ambas propiedades.

# grid

Atajo para establecer las siguientes propiedades en una sola declaración:

**grid-template-rows**

**grid-template-columns**

**grid-template-areas**

**grid-auto-rows**

**grid-auto-columns**

**grid-auto-flow**



CSS Grid

# Propiedades CSS para los items



**grid-column-start**

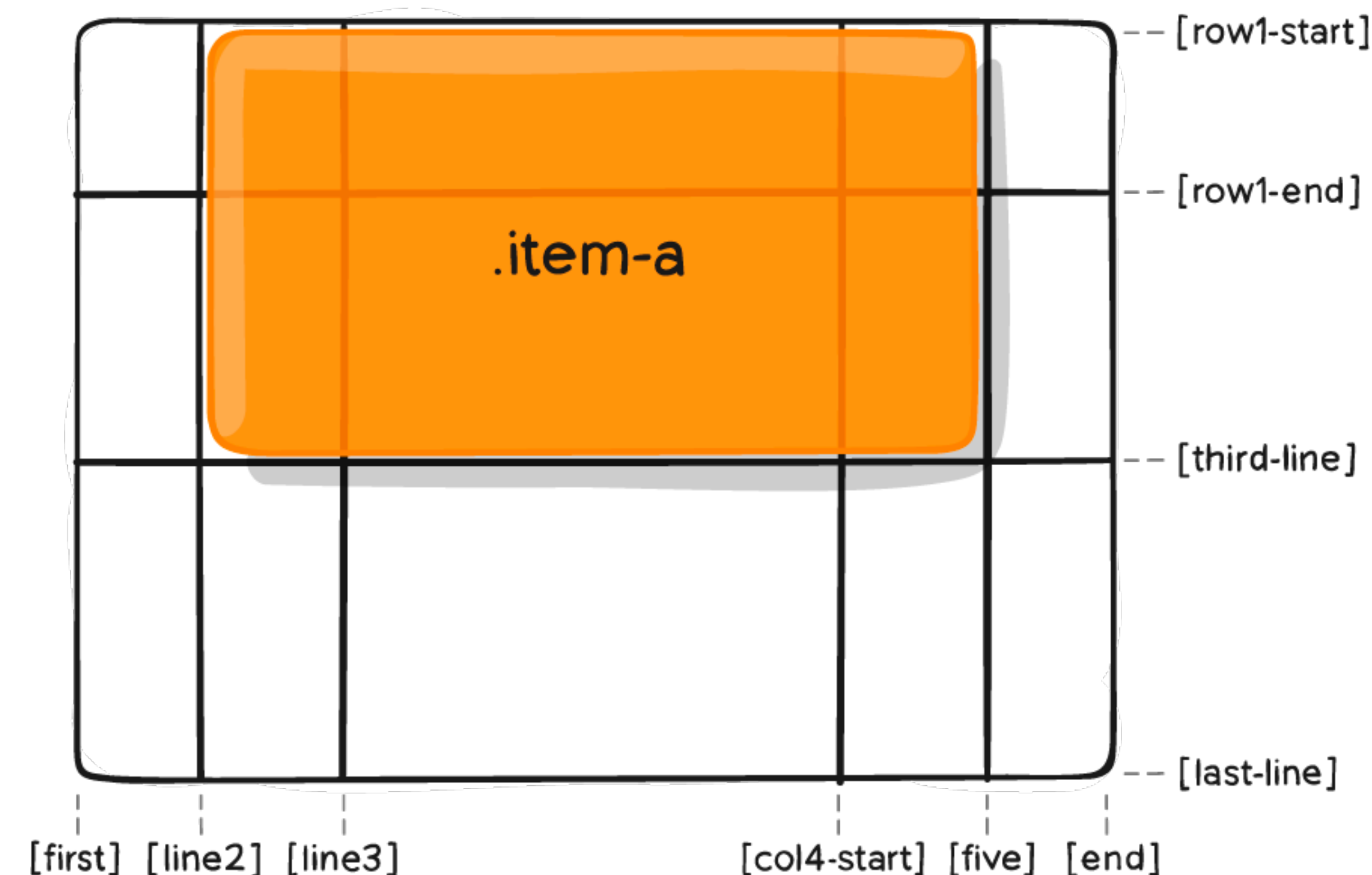
**grid-column-end**

**grid-row-start**

**grid-row-end**

Determinan la ubicación de un elemento dentro de la cuadrícula haciendo referencia a líneas de cuadrícula específicas.

Sin **grid-column-end** / **grid-row-end**, el elemento abarcará una pista.



```
.item-a {  
  grid-column-start: 2;  
  grid-column-end: five;  
  grid-row-start: row1-start;  
  grid-row-end: 3;  
}
```

**grid-column-start**

**grid-column-end**

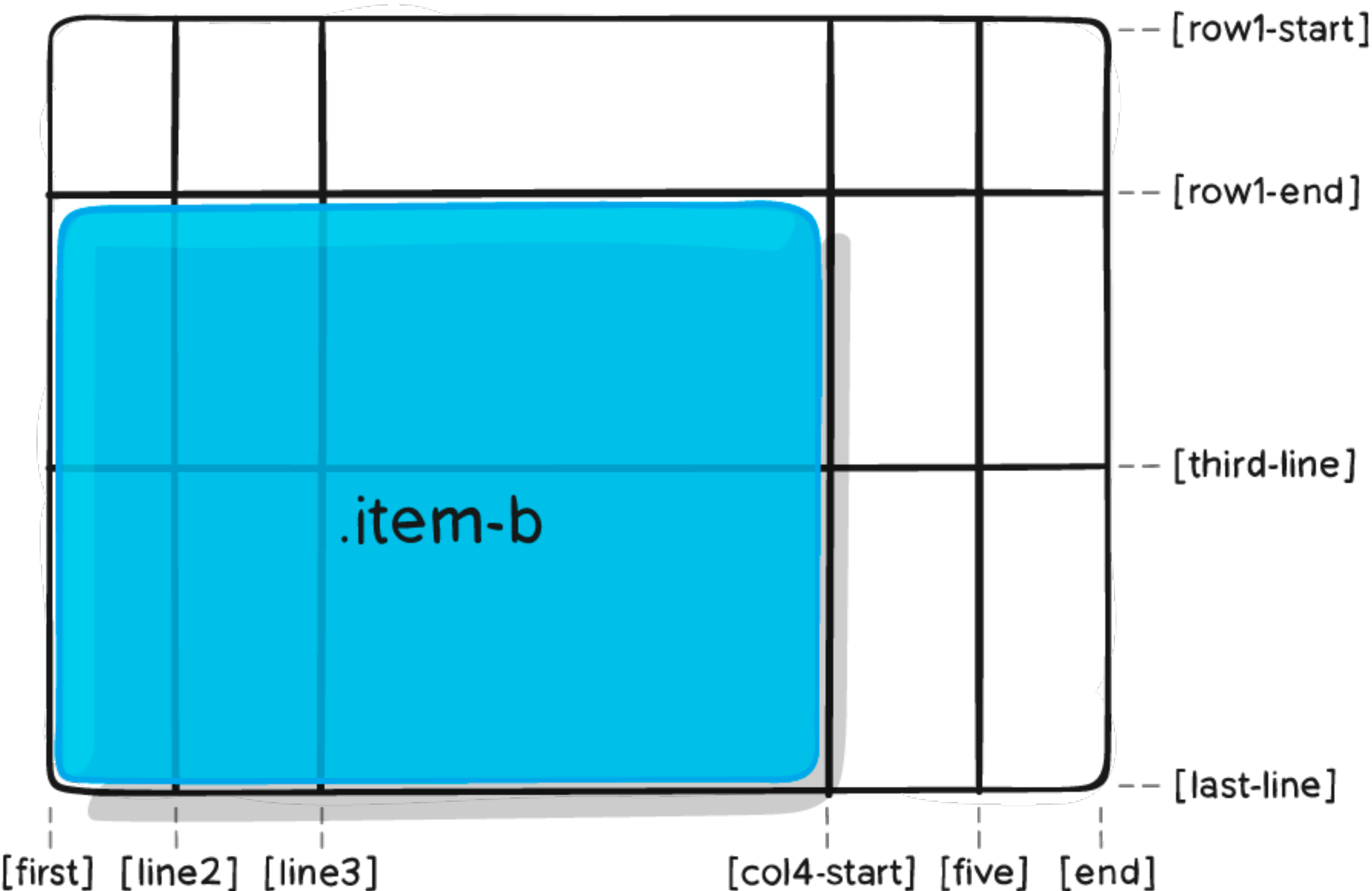
**grid-row-start**

**grid-row-end**

Determinan la ubicación de un elemento dentro de la cuadrícula haciendo referencia a líneas de cuadrícula específicas.

Los elementos pueden superponerse.

Con **z-index** se controla su orden.



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

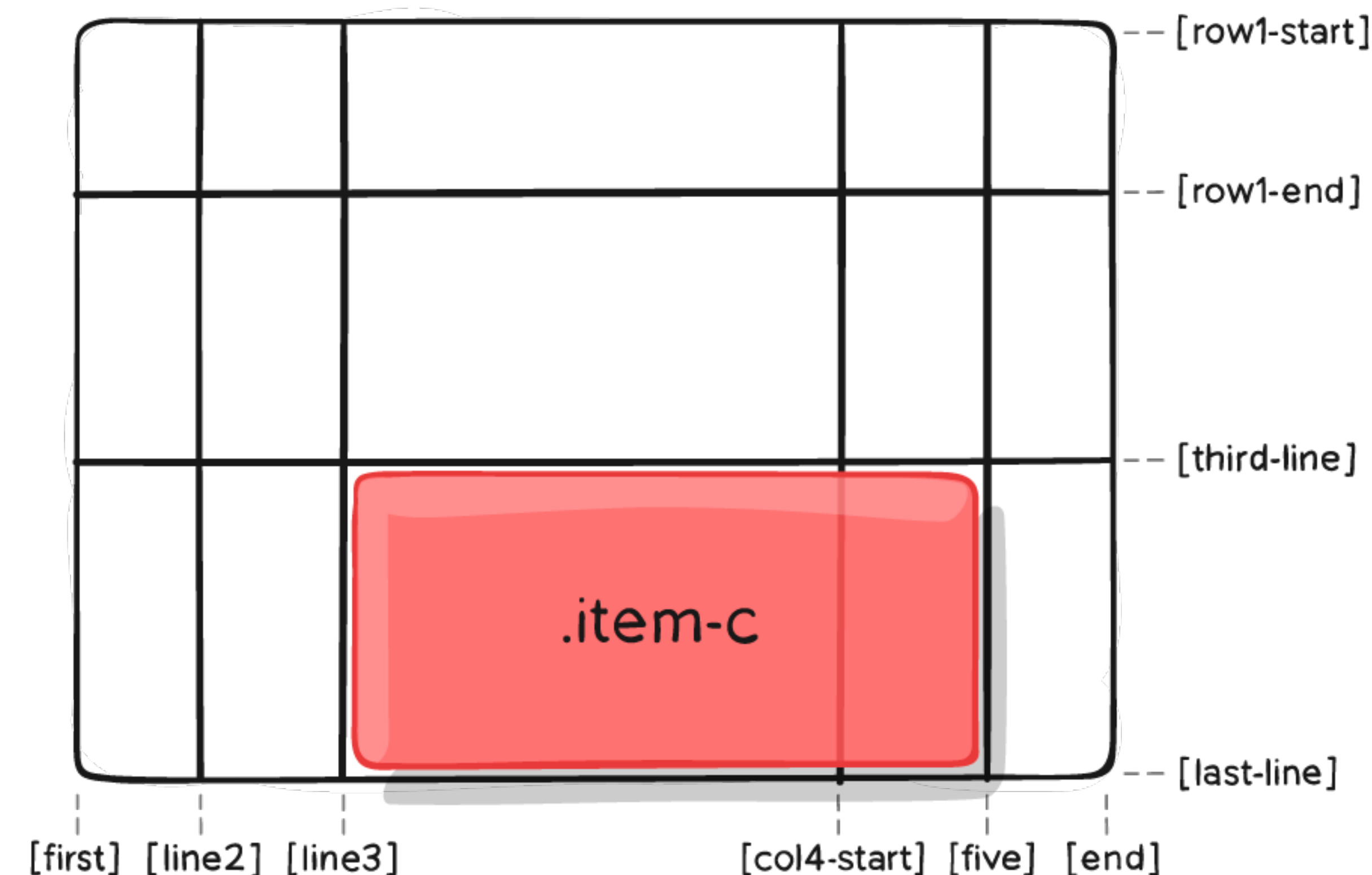
**grid-column**

**grid-row**

**grid-column-start + grid-column-end**

**grid-row-start + grid-row-end**

Sin valor de línea final, el elemento abarcará  
1 pista de forma predeterminada



```
.item-c {  
  grid-column: 3 / span 2;  
  grid-row: third-line / 4;  
}
```

### **grid-area**

- Da nombre a un elemento para que puede ser referenciado por una plantilla creada con **grid-template-areas**.
- Atajo aún más corto para **grid-row-start + grid-column-start + grid-row-end + grid-column-end**.

Valores:

**<nombre>** cualquier nombre

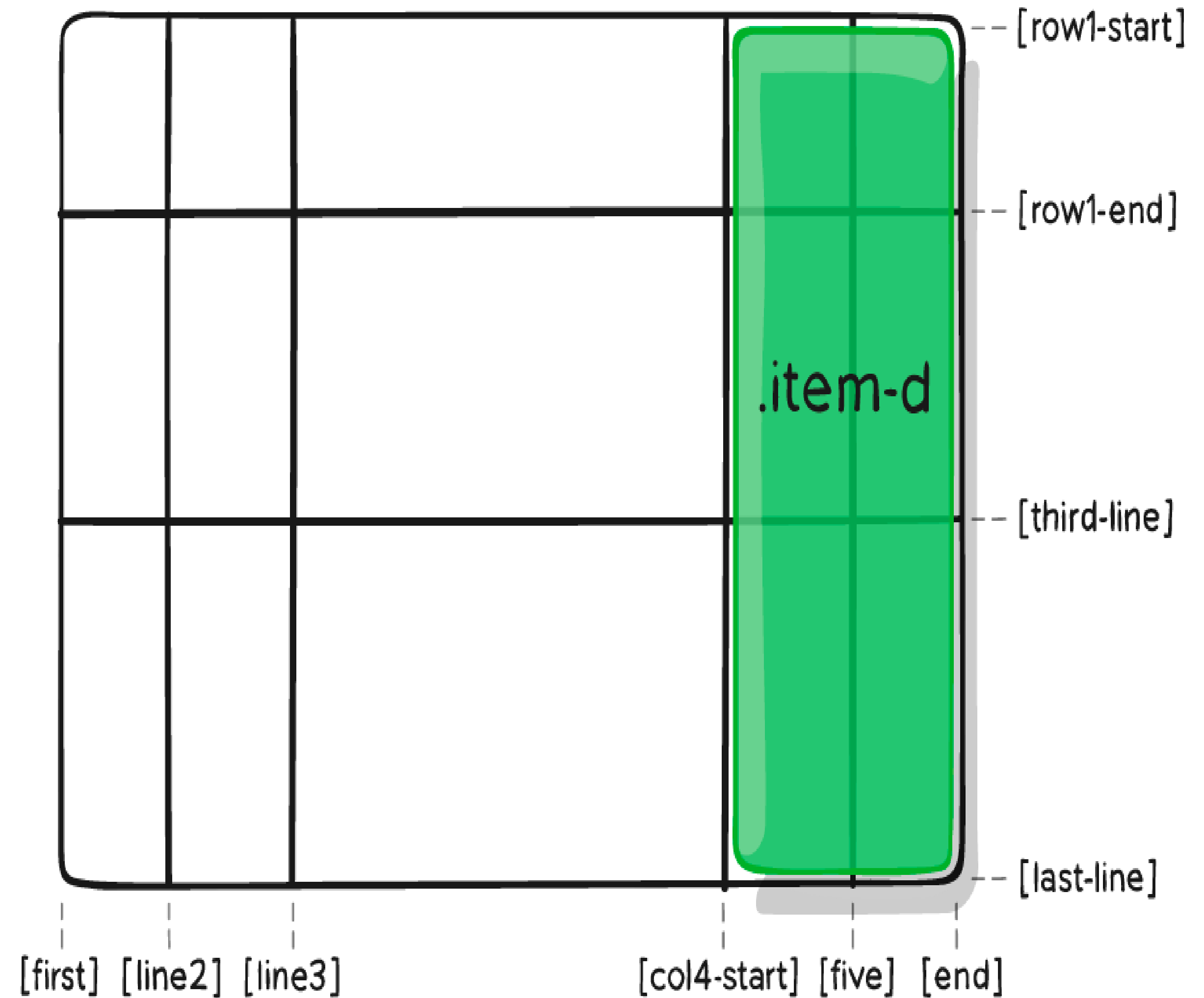
**<row-start> / <column-start> / <row-end> / <column-end>**

pueden ser números o líneas con nombre

## grid-area

```
.item-d {  
  grid-area: header;  
}
```

```
.item-d {  
  grid-area: 1 / col4-start /  
  last-line / 6;  
}
```



### **justify-self**

Alinea un elemento en una celda según el eje horizontal

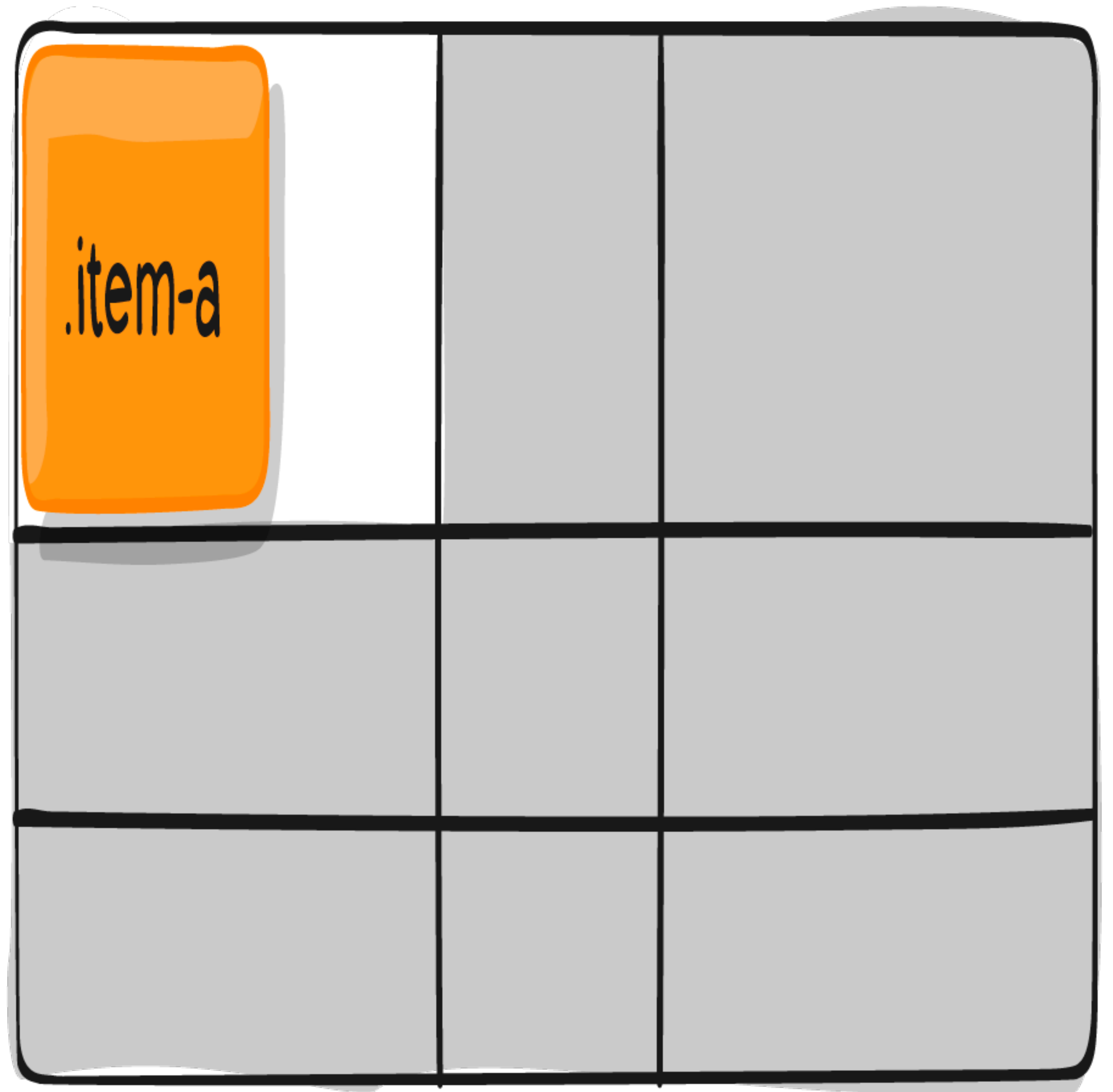
Valores:

**start**

**end**

**center**

**stretch**



### **justify-self**

Alinea un elemento en una celda según el eje horizontal

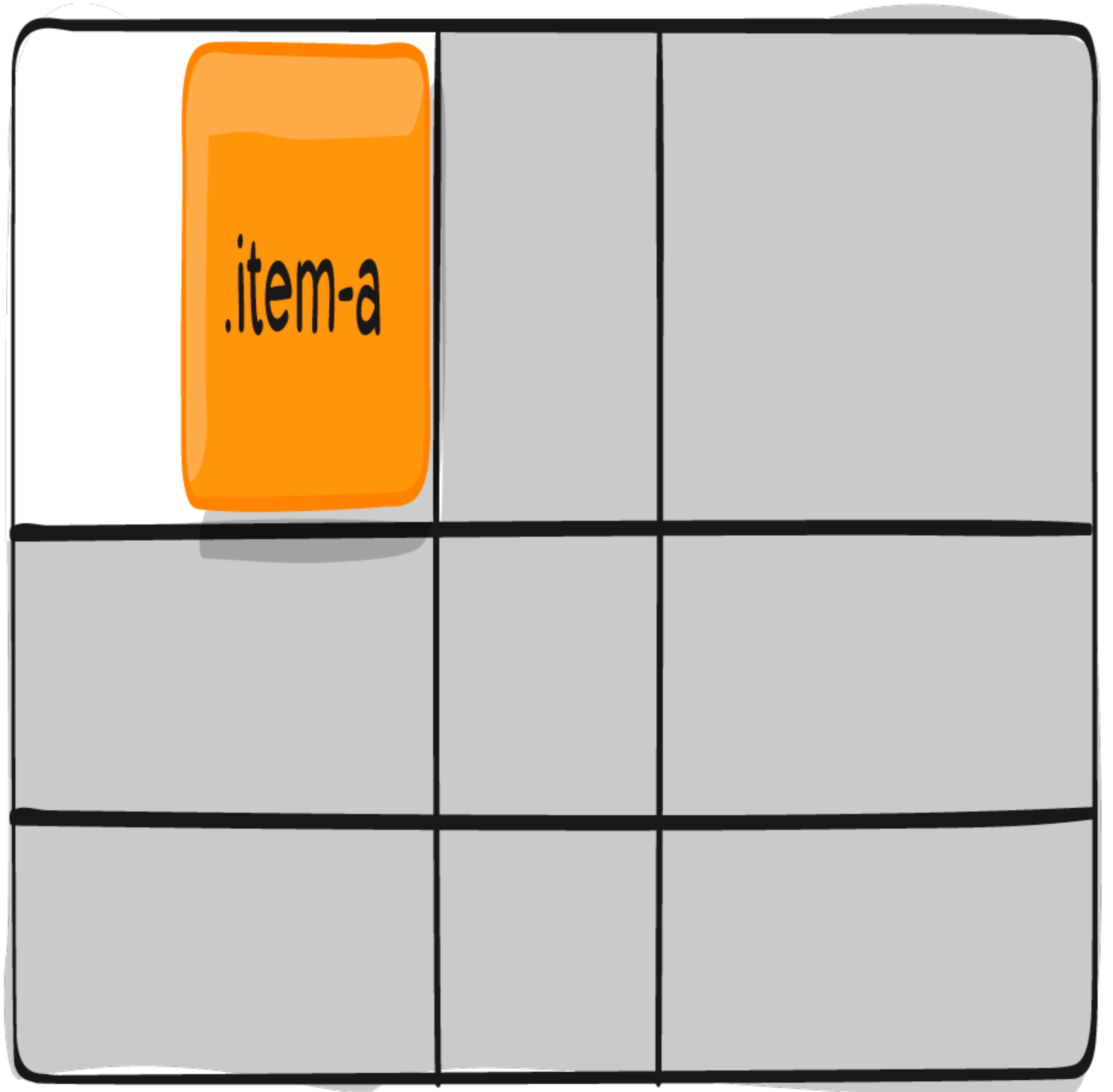
Valores:

**start**

**end**

**center**

**stretch**



## justify-self

Alinea un elemento en una celda según el eje horizontal

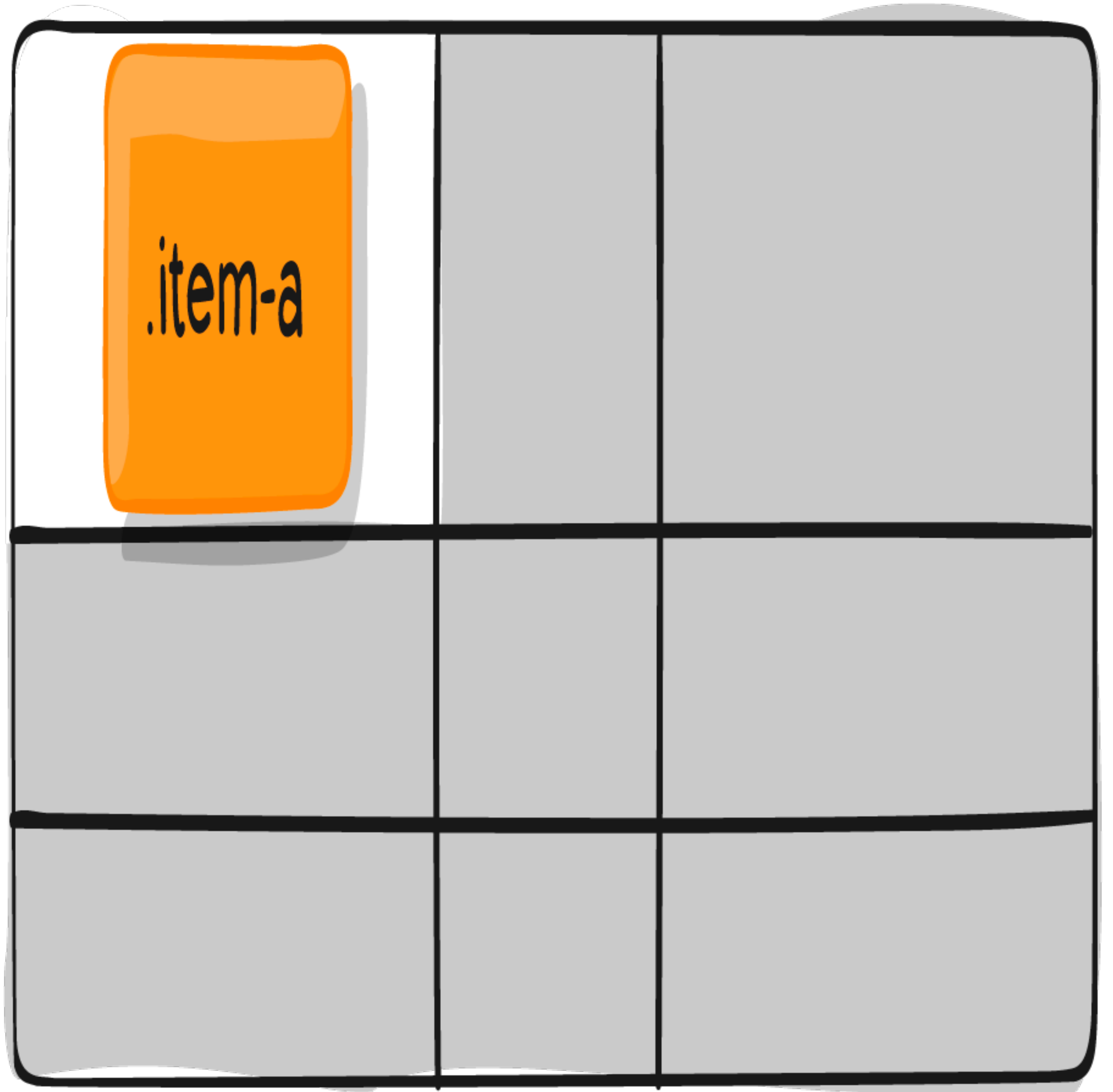
Valores:

**start**

**end**

**center**

**stretch**





### **justify-self**

Alinea un elemento en una celda según el eje horizontal

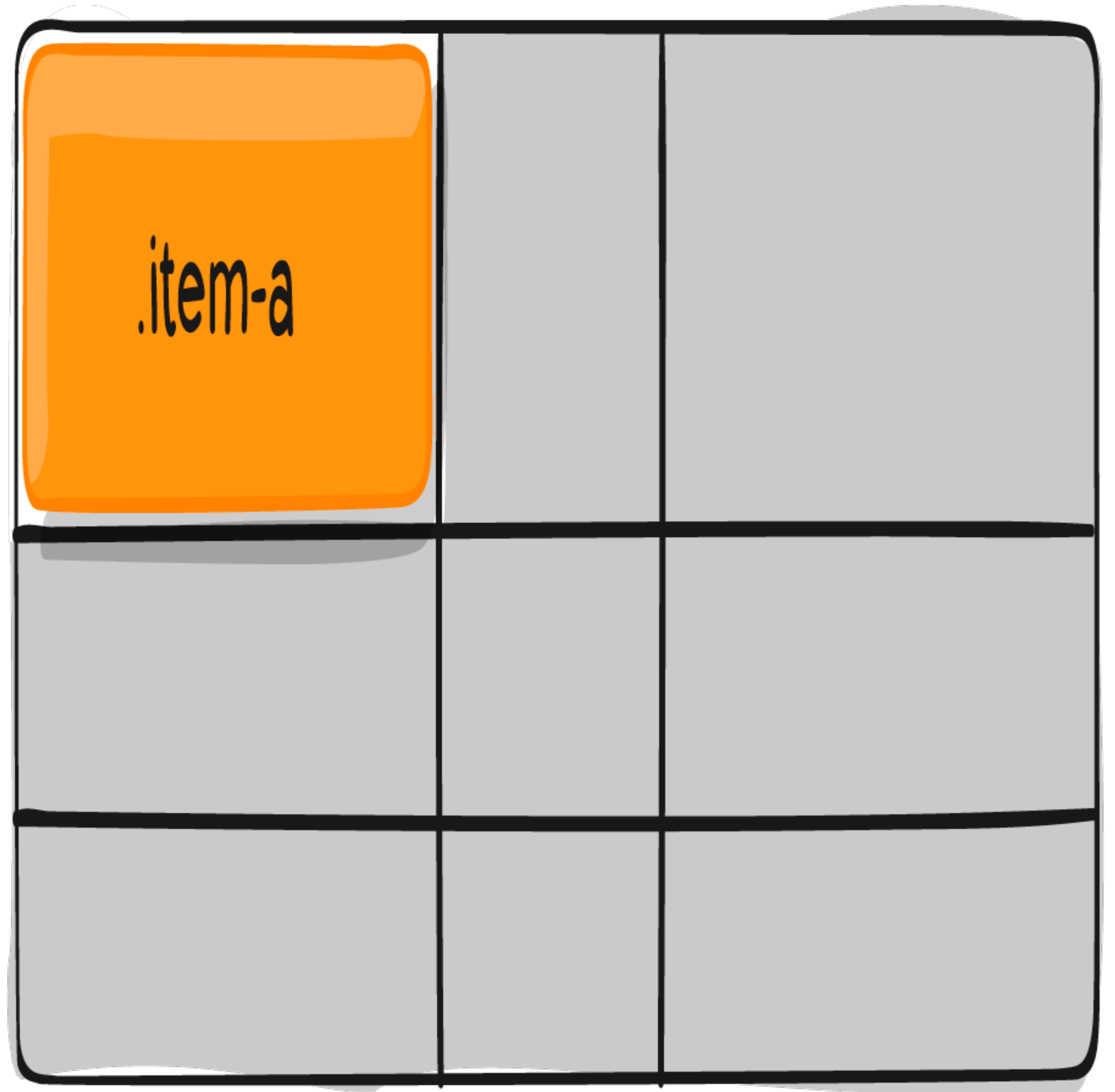
Valores:

**start**

**end**

**center**

**stretch**



## **align-self**

Alinea un elemento en una celda según el eje vertical

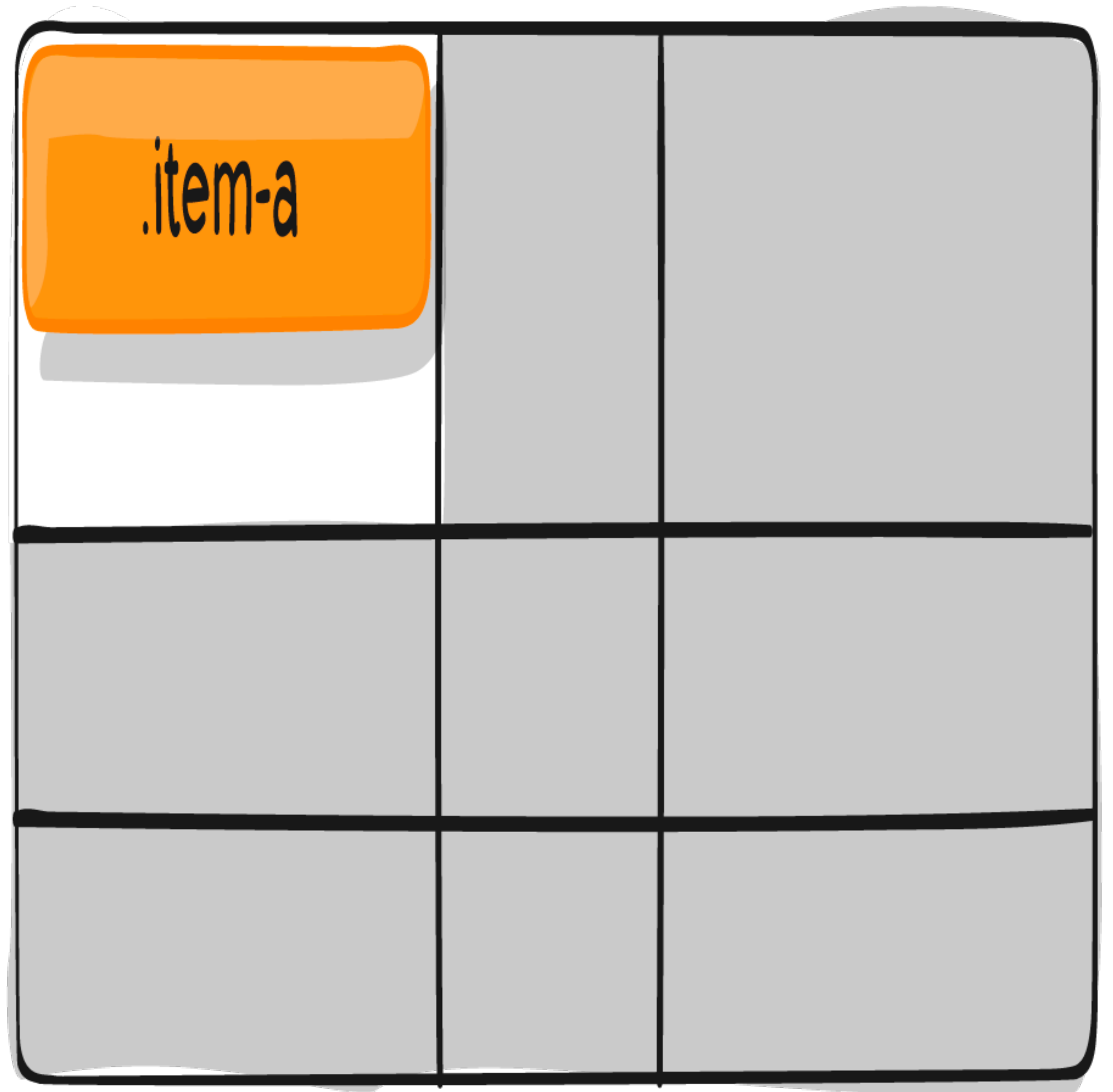
Valores:

**start**

**end**

**center**

**stretch**



## **align-self**

Alinea un elemento en una celda según el eje vertical

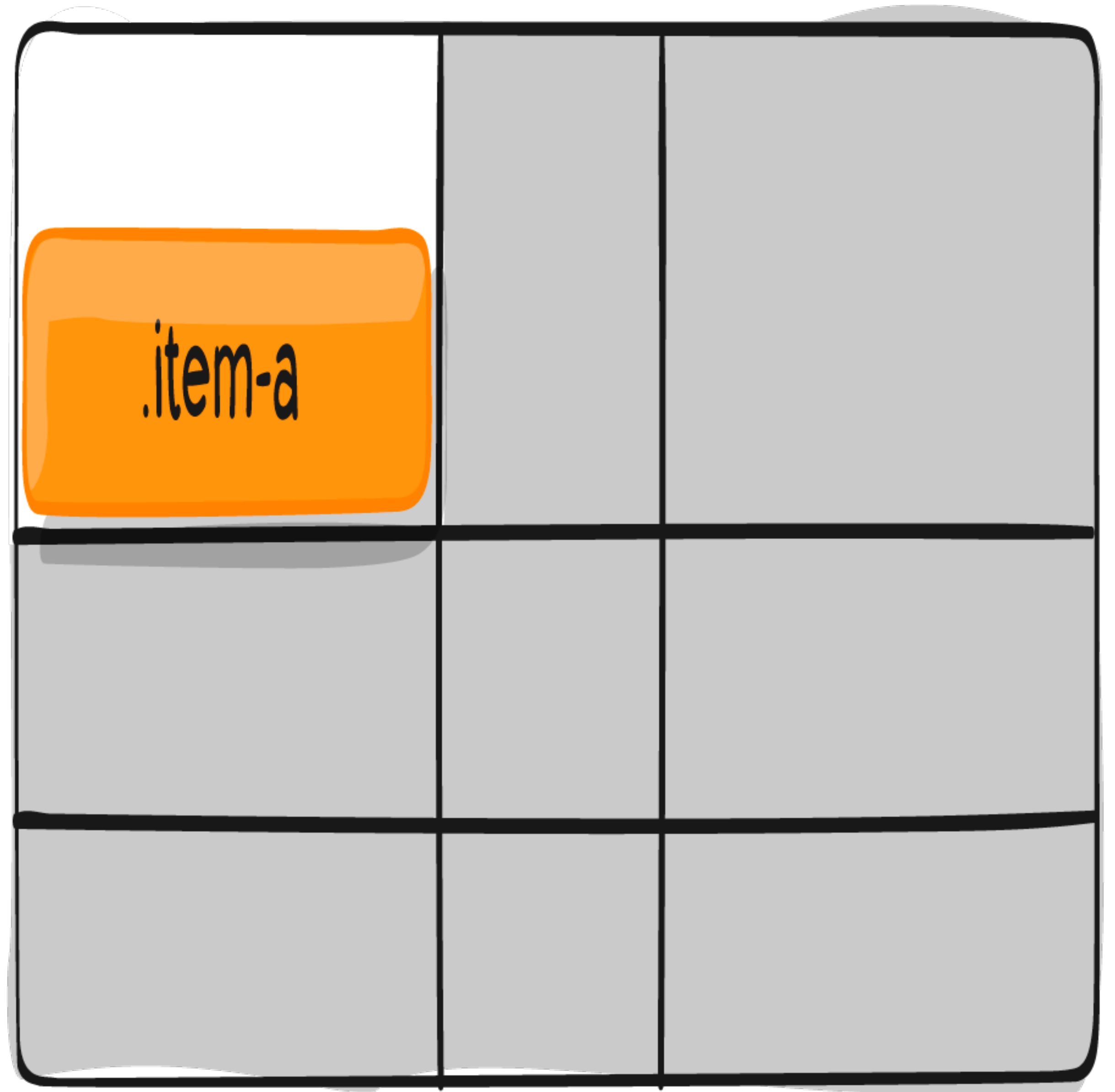
Valores:

**start**

**end**

**center**

**stretch**



## **align-self**

Alinea un elemento en una celda según el eje vertical

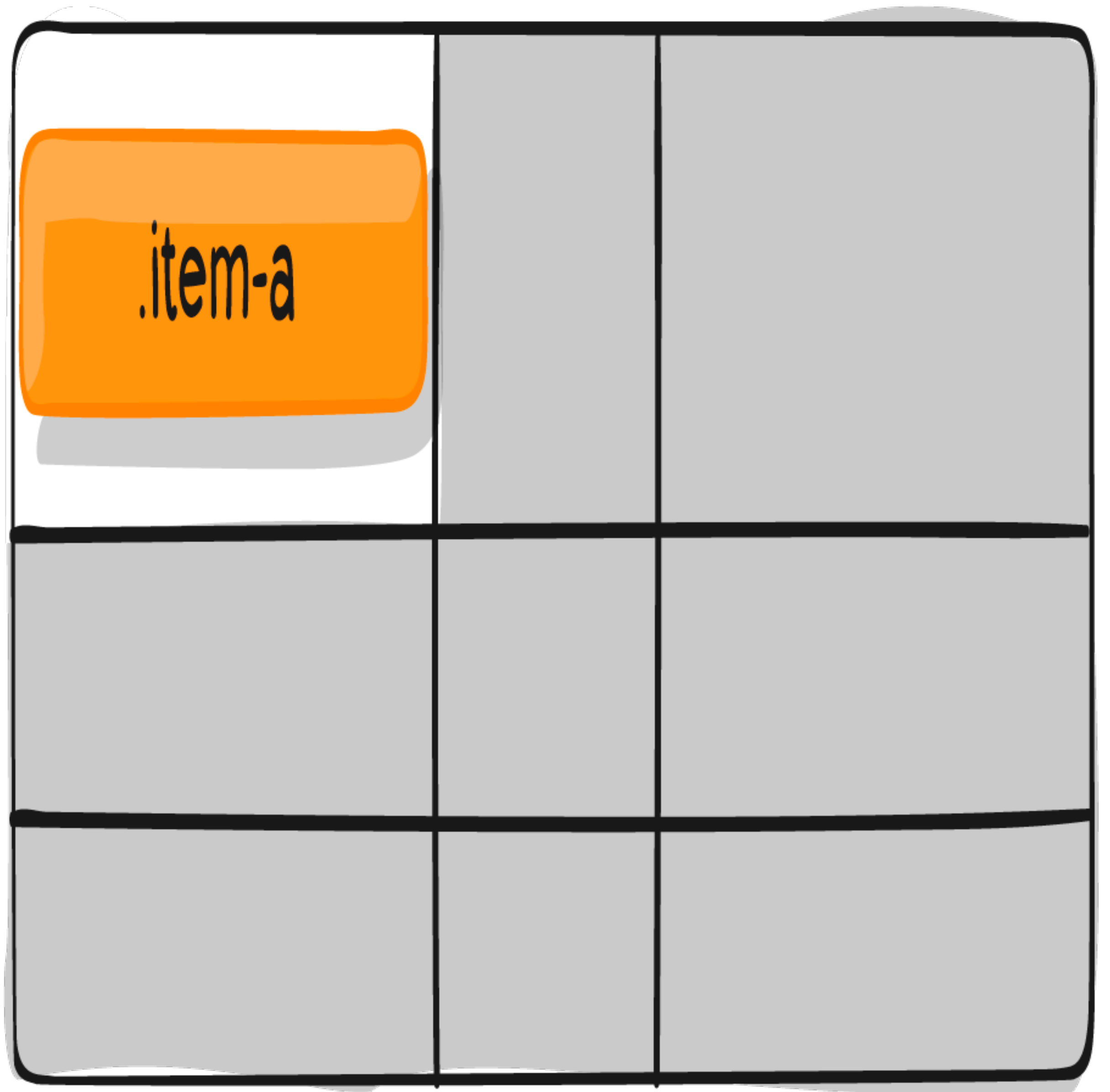
Valores:

**start**

**end**

**center**

**stretch**



## **align-self**

Alinea un elemento en una celda según el eje vertical

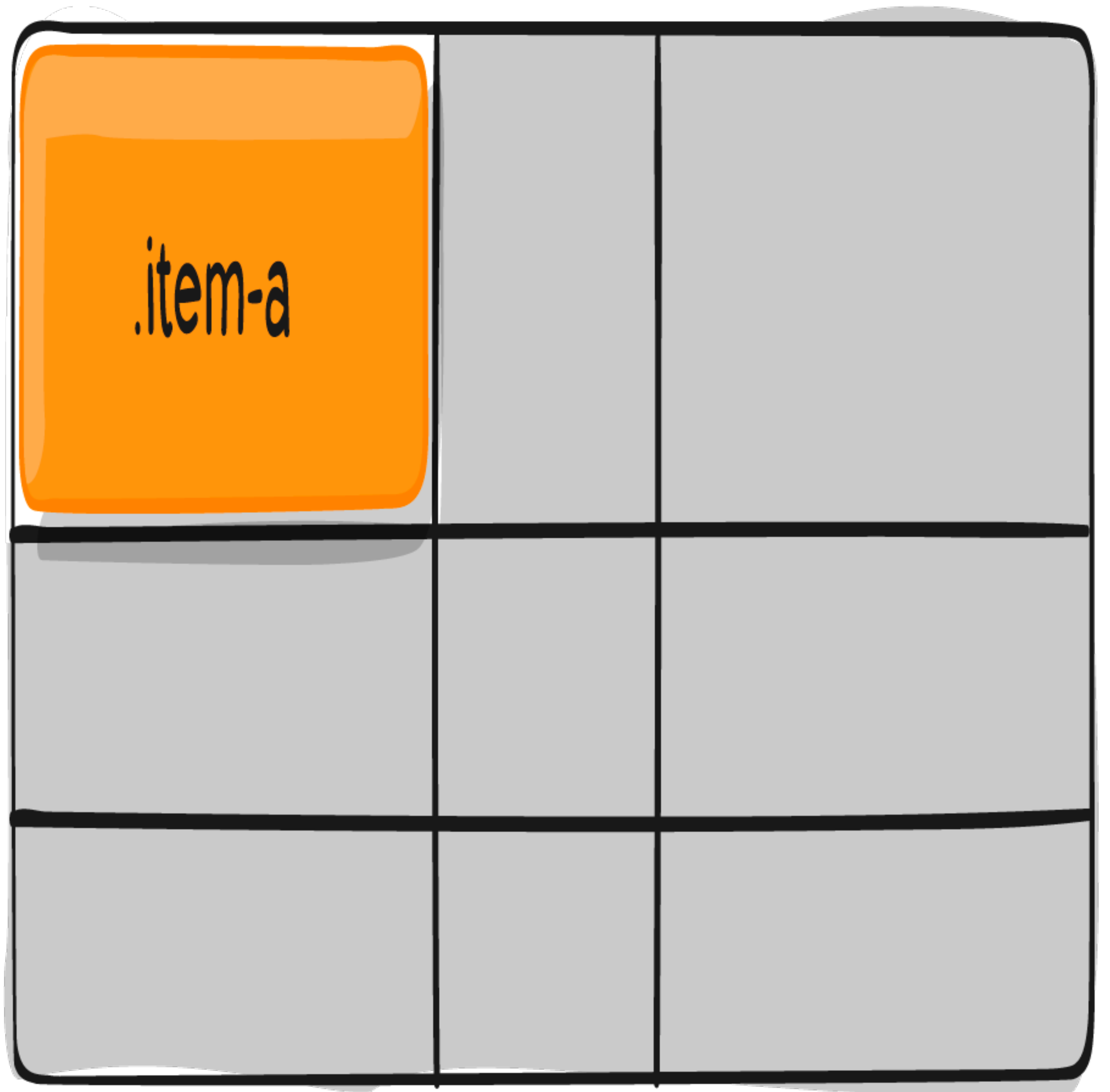
Valores:

**start**

**end**

**center**

**stretch**

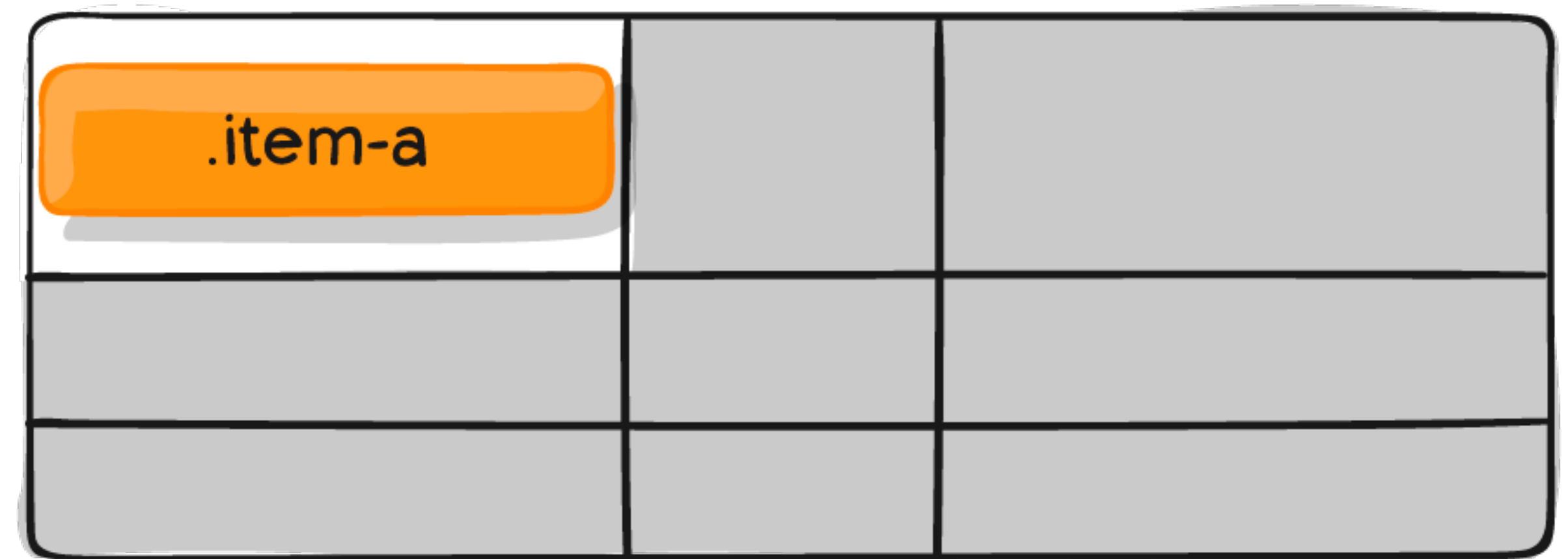
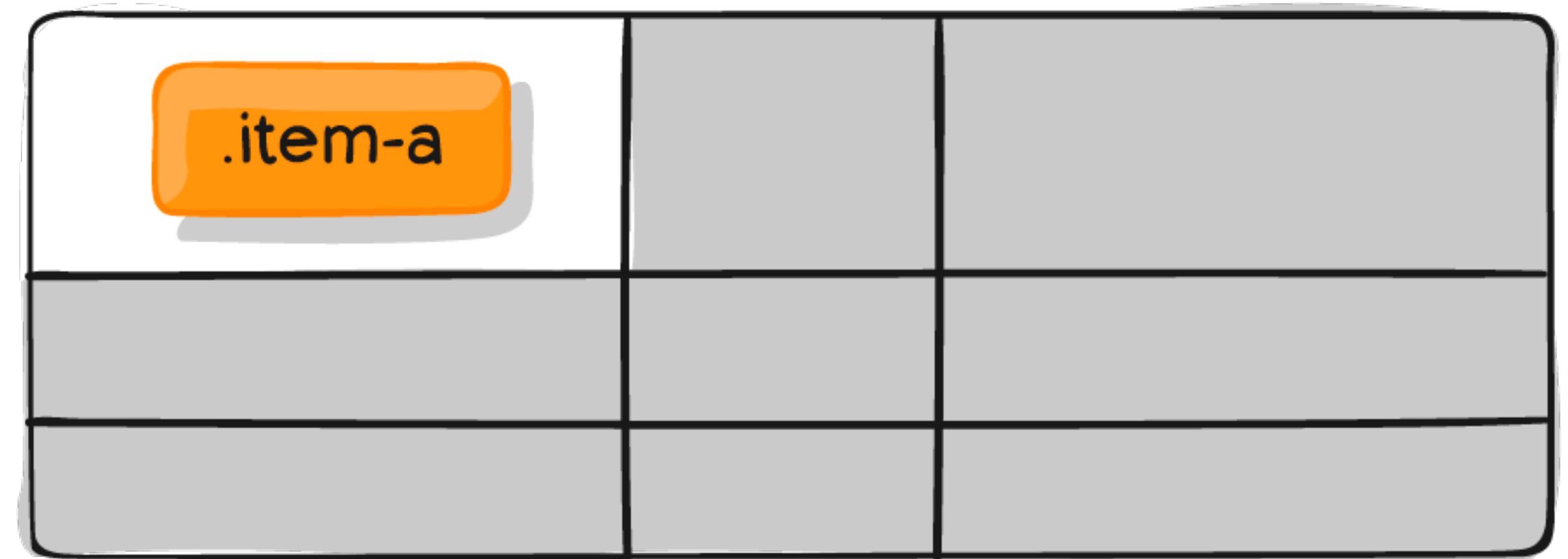


## place-self

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

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

## align-self + justify-self





CSS Grid

# Funciones especiales y palabras clave



## CSS Grid / Funciones especiales y palabras clave

- Se pueden utilizar todas las unidades que se suelen utilizar (**px**, **em**, **rem**, **%**, etc.). Pero hay palabras clave como **min-content**, **max-content**, **auto**, y unidades fraccionales: **.grid-template-columns: 200px 1fr 2fr min-content;**
- Se puede establecer ancho de una columna en **1fr**, pero que no sea menor a **200px**, por ejemplo: **grid-template-columns: 1fr minmax(200px, 1fr);**
- La función **repeat()** ahorra escritura, por ejemplo, para 10 columnas: **grid-template-columns: repeat(10, 1fr);**
- Se pueden combinar, como: **grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));**

### **minmax()**

El tamaño de una pista de cuadrícula variará de un rango mínimo a máximo

### **max-content**

La columna de la cuadrícula será tan amplia como el elemento más ancho que contiene

### **min-content**

La columna de la cuadrícula será tan ancha como el elemento más angosto que contiene, de forma que no produzca un desbordamiento

### **auto-fill**

Llena el contenedor si sobran elementos. Crea columnas vacías adicionales si hay espacio disponible.

### **auto-fit**

Extiende el último elemento de la cuadrícula al extremo del contenedor, independientemente si hay espacio para otra columna.



Una guía de

# CSS Grid