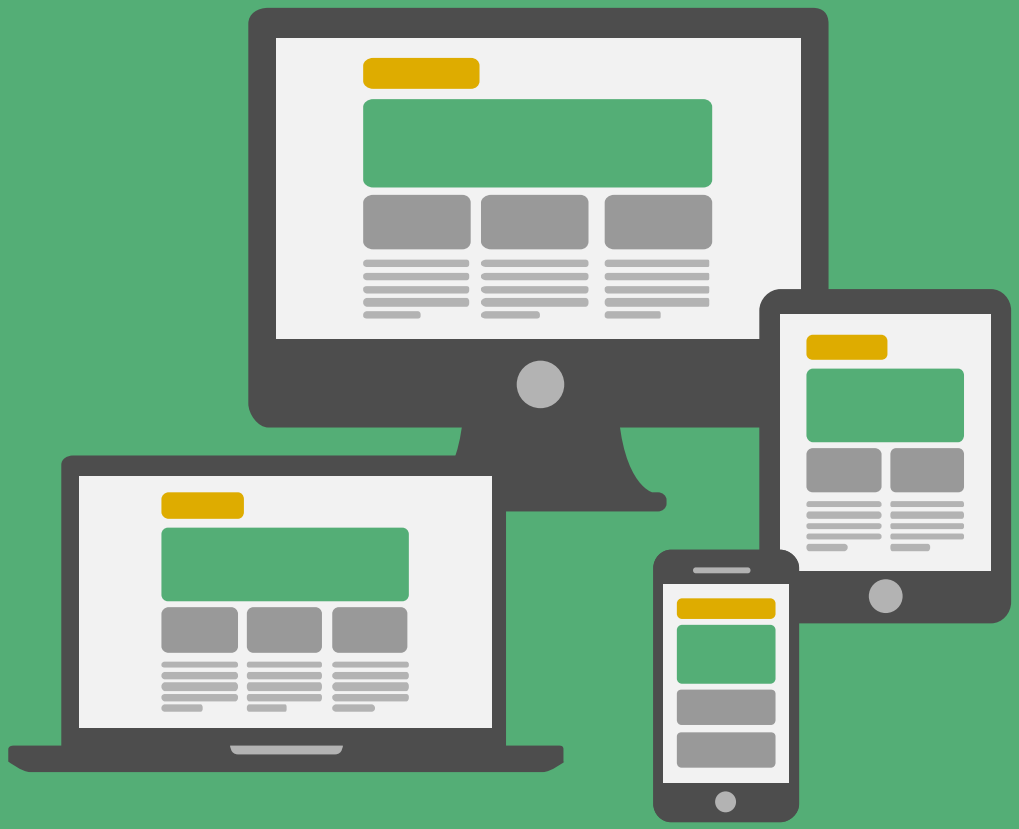
Diseño de Interfaz



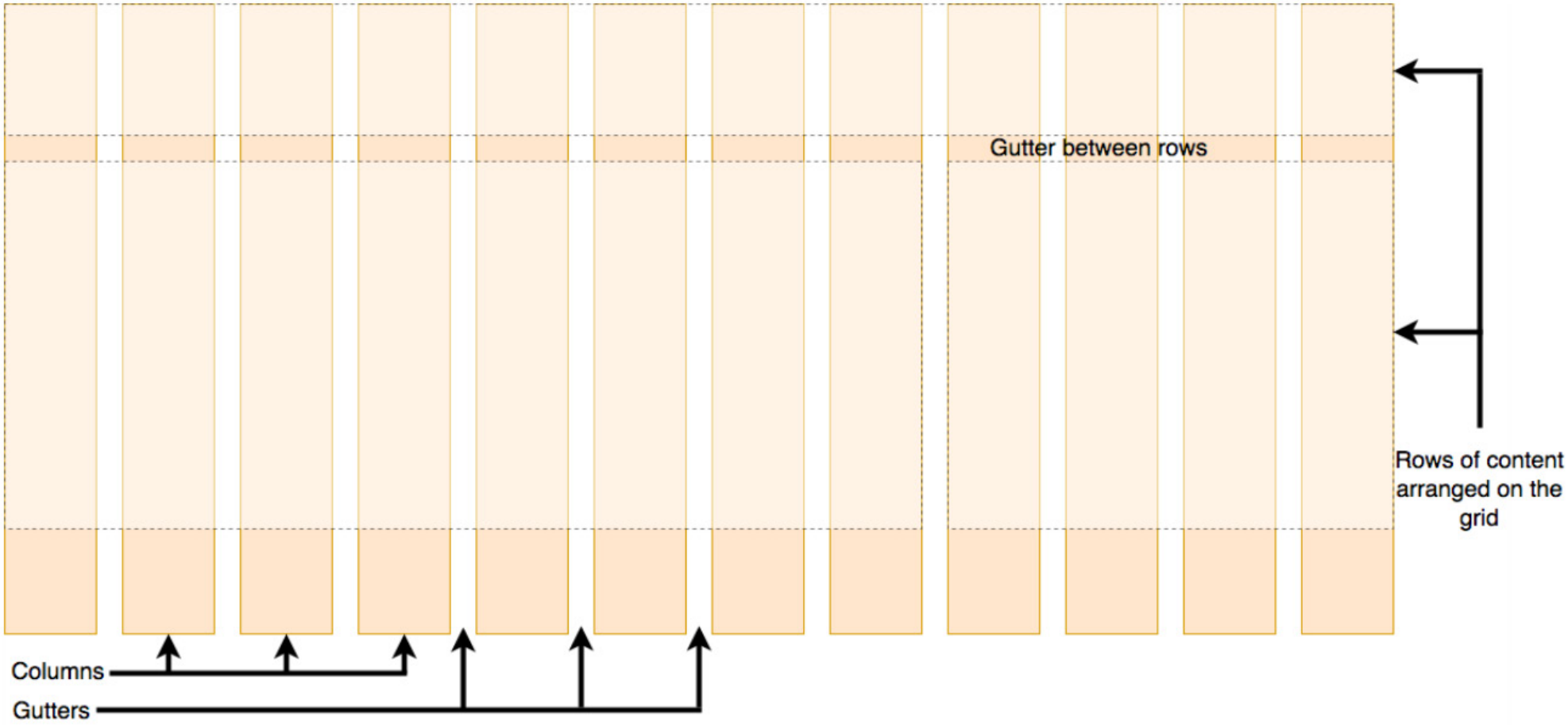
Formato visual

Layout de contenedores Display grid

La compaginación en cuadrícula o grillas con CSS (Grid) es un sistema de diseño en dos dimensiones. Permite distribuir el contenido en filas y columnas, y tiene muchas características que facilitan la creación de diseños complejos.

Una cuadrícula es un conjunto de líneas horizontales y verticales que crean un patrón sobre el que podemos alinear nuestros elementos de diseño.
Las cuadrículas nos ayudan a crear diseños de página en que los elementos no saltan ni cambian de ancho cuando nos movemos de una página a otra, y así proporcionan a nuestras páginas web un aspecto más coherente.

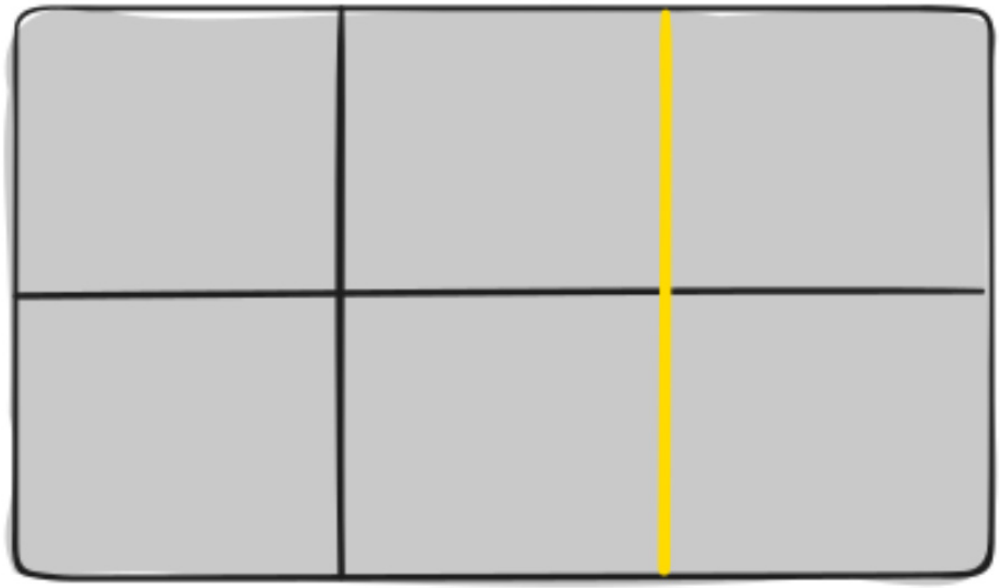
Una cuadrícula en general tiene **columnas**, **filas** y luego espacios entre cada fila y cada columna, conocidos
comúnmente como **medianiles (gutter/gap)**



Terminología de Grid

Grid Line

Son las líneas divisorias que componen la estructura de la cuadrícula. Pueden ser verticales (**column grid lines**) u horizontales (**row grid lines**) y corren a ambos lados de una fila o columna. Se pueden numerar

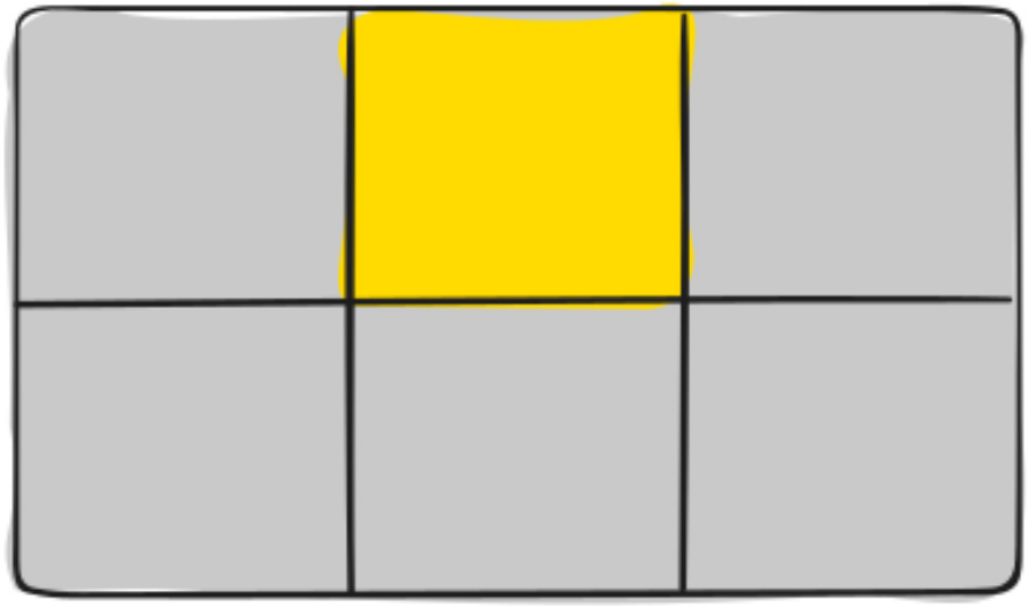


Grid Cell

Espacio formado entre 2 lineas divisorias adyacentes verticales y horizontales.

Es la unidad básica de Grid.

Es una sola “unidad” de la cuadrícula.



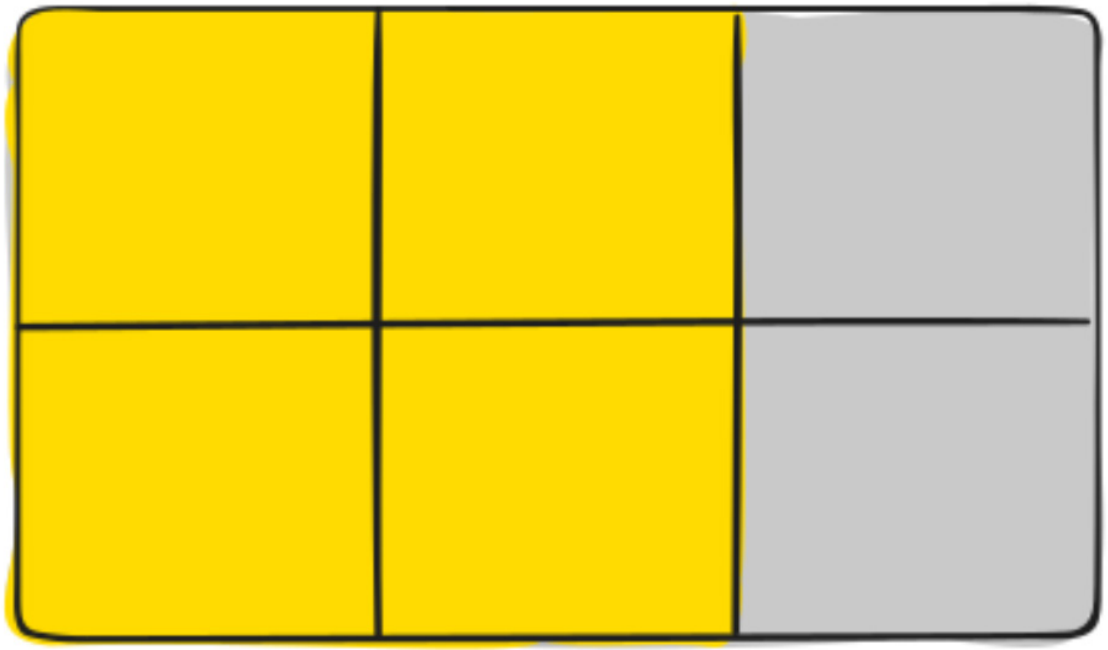
Grid Track

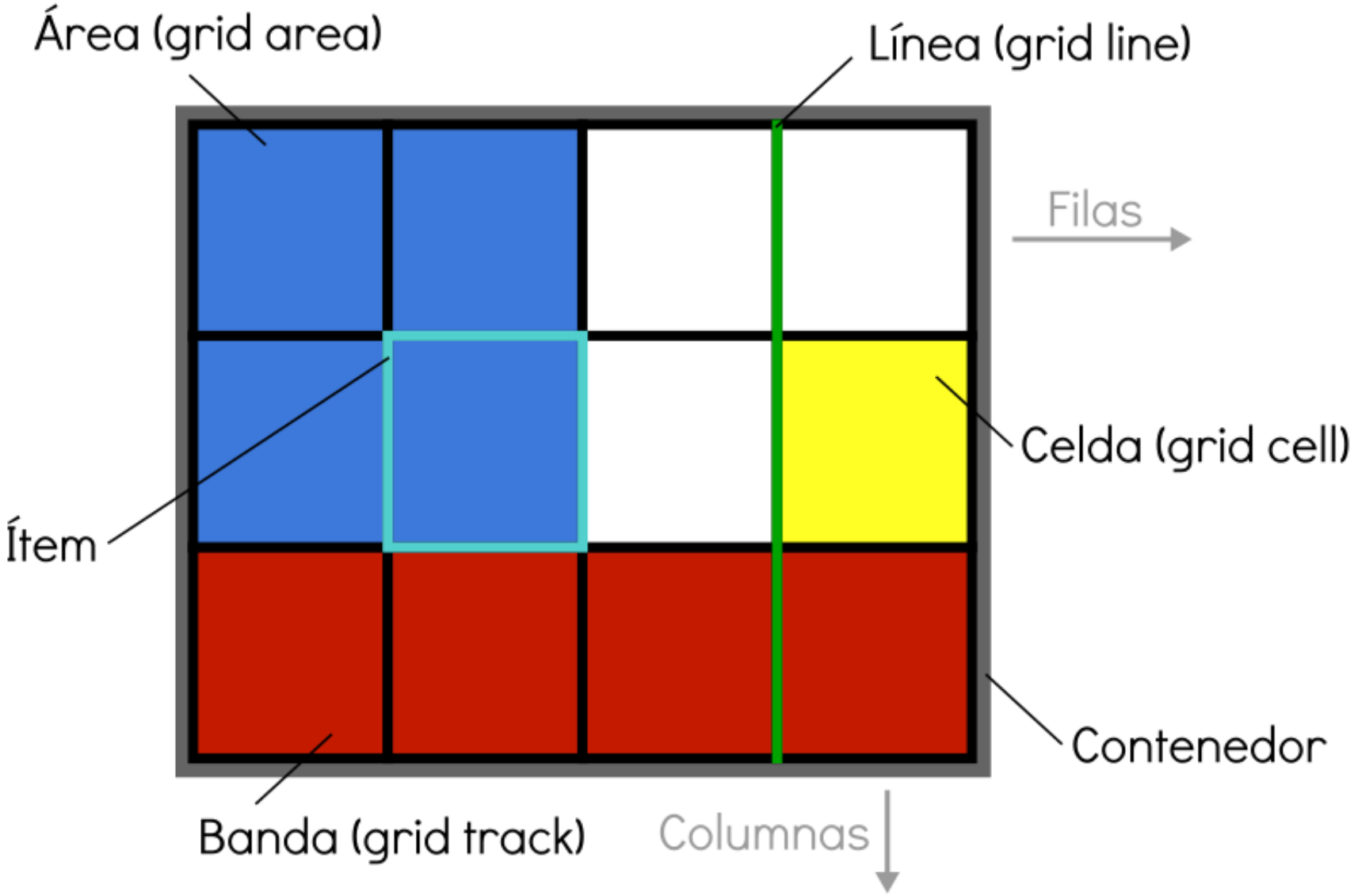
Espacio entre dos líneas divisorias adyacentes, se puede pensar como filas y columnas de la grilla.

Grid Area

Epacio total entre 2 lineas horizontales y 2 verticales cualquiera.

Puede estar compuesto de cualquier número de celdas




Área (grid area)
Línea (grid line)
Filas
Celda (grid cell)
Ítem
Contenedor
Banda (grid track)
Columnas

Crear la grilla

Al trabajar con Grid, vamos a diferenciar contenedor de hijos, cada uno con sus propiedades y características para poder lograr el diseño deseado.

Contenedor de grilla (Grid container)

El elemento "**padre**" que contiene los elementos flexibles. Un contenedor flexible se define usando los valores **grid** o **inline-grid** en la propiedad display.

Elemento de grilla (Grid item)

Cada hijo de un contenedor grid se convierte en un elemento de grilla.

A diferencia del método Flex, los elementos no se ven diferentes inmediatamente.
La declaración **display: grid** proporciona una cuadrícula de una sola columna, por lo que los elementos continúan mostrándose uno debajo del otro, como lo hacen en el flujo normal.

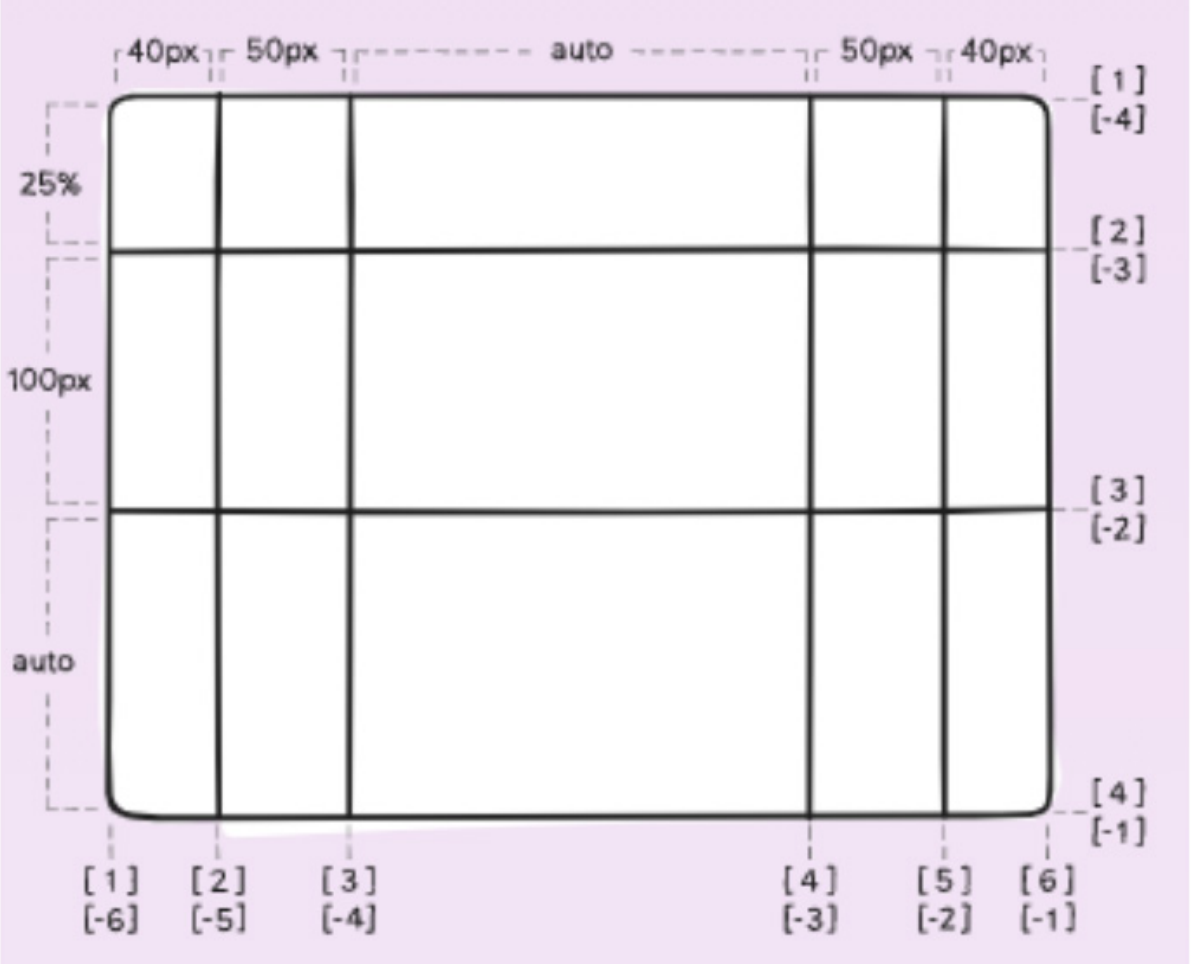
Para ver algo que se parezca más a una cuadrícula, necesitamos crear las columnas y filas en la cuadrícula.

grid-template-columns
grid-template-rows

Las columnas y filas de la grilla se definen con valores separados por espacios.
Esos valores representan el ancho de los tracks.

Pueden ser en px, % o fracciones (fr) que es una unidad especifica para el manejo de grillas

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



repeat

Si las partes se repiten podemos usar **repeat**()

```
.container {
 grid-template-columns: repeat(3, 20px);
}

.container {
 grid-template-columns: repeat(3, 20px 30px);
}
```

Cuadrículas flexibles con la unidad fr

Además de crear cuadrículas con longitudes y porcentajes, podemos usar la **unidad fr** para dimensionar de manera flexible las filas y columnas de la cuadrícula.

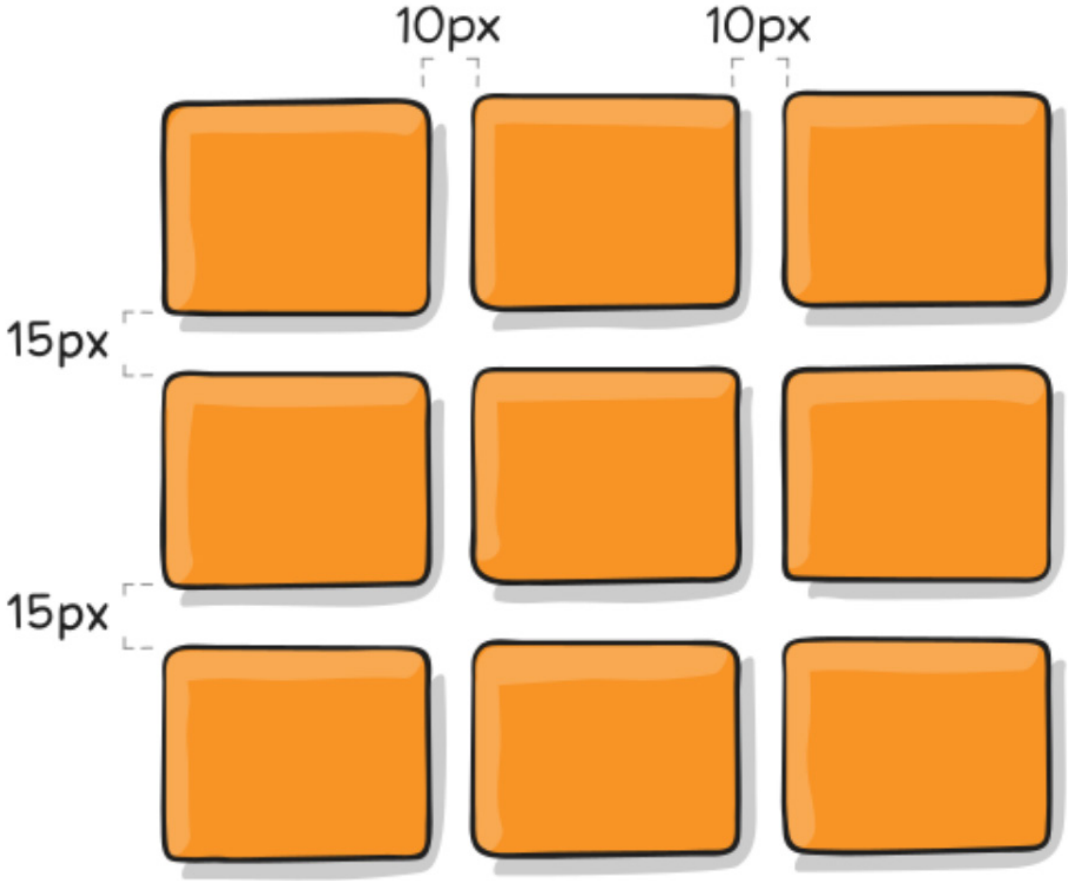
Esta unidad representa una fracción del espacio disponible en el contenedor de la cuadrícula.

```
.container {
 grid-template-columns: 1fr 2fr 1fr;
}
```

Gaps

Para crear espacios entre tracks, utilizaremos las propiedades **column-gap y row-gap**.
O **gap** para configurar ambos a la vez.

```
.container {
 gap: 15px 10px;
 }
```



Estos espacios pueden expresarse en px o % pero no en fr.

Grilla implicita y explícita

Podemos crear nuestra **grilla explícita definiendo** cuántas **columnas y filas** tendrá nuestra grilla con los grid-template-columns o rows.

Pero si definimos un solo eje de tracks (ej. solo columnas) automáticamente se generan las filas correspondientes para contener a nuestro contenido.

Por defecto los tracks que se generan en una cuadricula implícita, tienen un tamaño auto.

Tantas columnas como quepan

Podemos combinar algunos parámetros, como el repeat y el minmax, para crear patrones más útiles.
Si necesitamos colocar tantas columnas como quepan en nuestro contenedor.

Podemos establecer el grid-temmplate-columns, con el parámetro de repeat(), pero en lugar de pasar un valor, le indicamos auto-fill.

Para el segundo parámetro de la función usamos minmax() con un valor mínimo para el track, y valor máximo de 1fr.

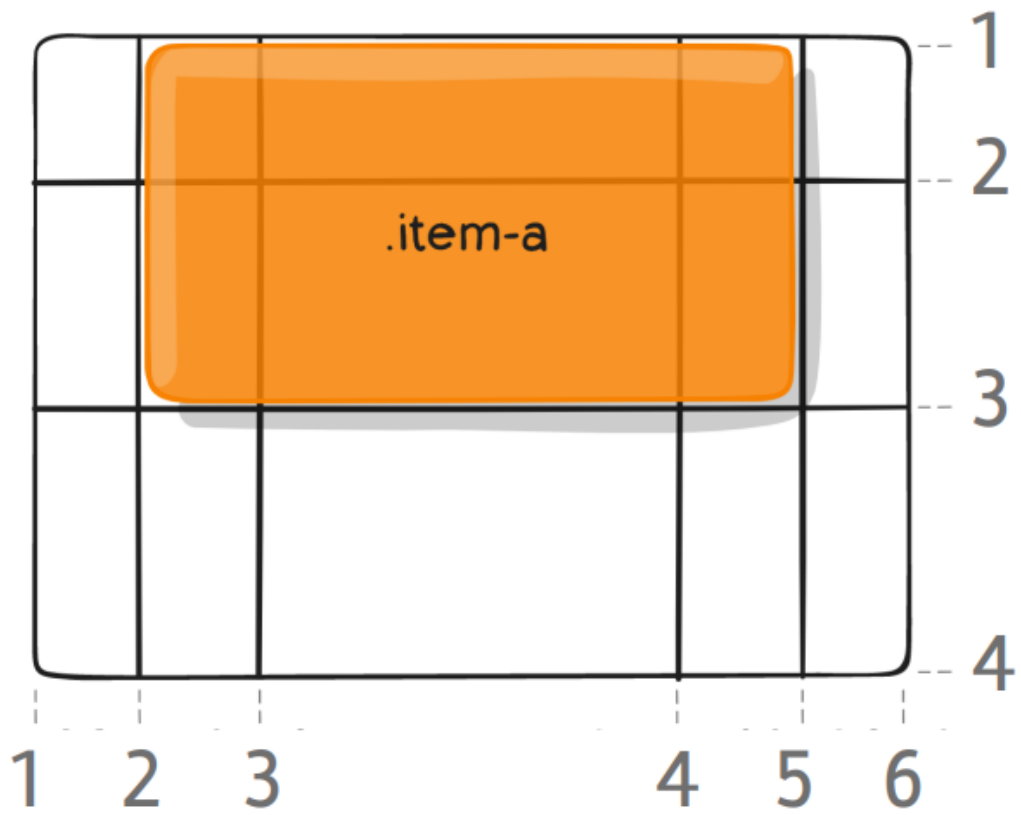
```
.container {
 grid-template-columns: repeat(auto-fill, min-
max(200px, 1fr));
}
```

Esto funciona porque la cuadrícula crea tantas columnas de 200 píxeles como caben en el contenedor, luego comparte el espacio restante entre todas las columnas

Propiedades de los hijos

grid-column
grid-row

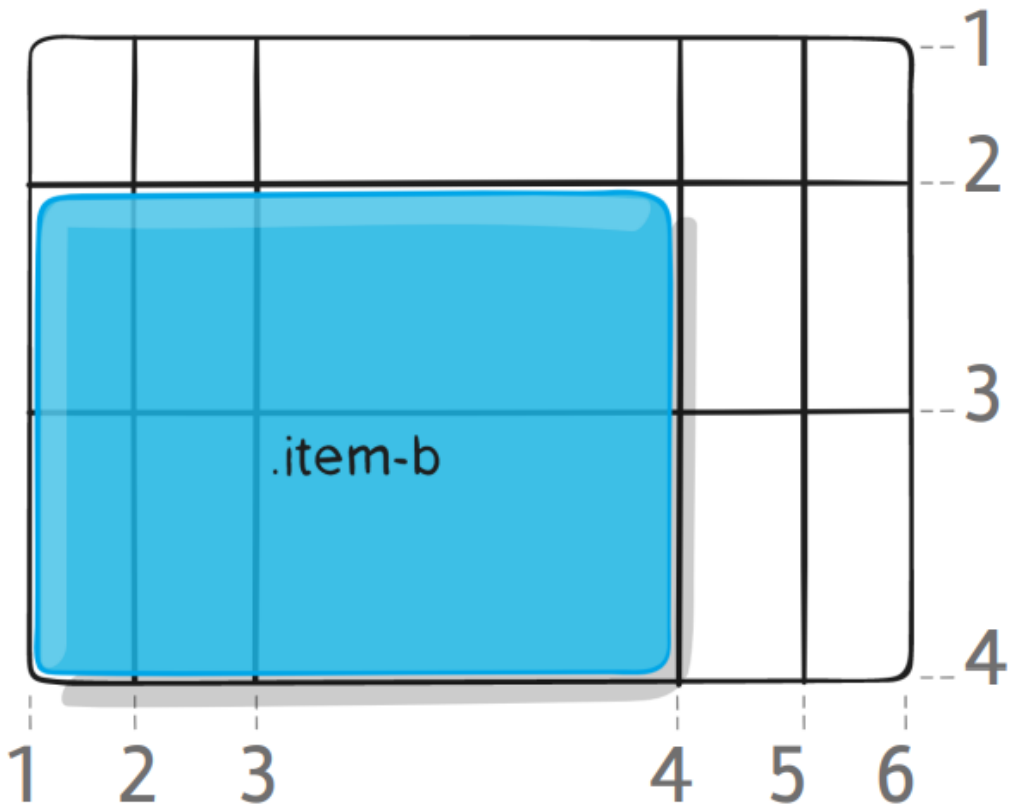
Determina la ubicación de una celda dentro de la grilla, refiriéndose a las líneas, desde dónde arranca y hasta dónde llega.




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

También podemos indicar cuántos tracks ocupa, con el parámetro span y la cantidad.

Si no se indica una línea de final, el item ocupará un solo espacio




```
.item-b {
 grid-column: span 3;
 grid-row: 2/ span 2;
}
```