

## Grid CSS

.

Según Wikipedia contributors. (n.d.) , CSS grid layout o CSS grid es una técnica de las Hojas de Estilo en Cascada que permite a los desarrolladores web crear diseños complejos y adaptables con mayor facilidad en todos los navegadores. Ha habido otros métodos para controlar los métodos de diseño de páginas web, como las tablas, el modelo de caja, y CSS flex. CSS grid no es actualmente un estándar oficial (es una W3C Candidate Recommendation) aunque ha sido adoptada por la mayoría de los principales navegadores.

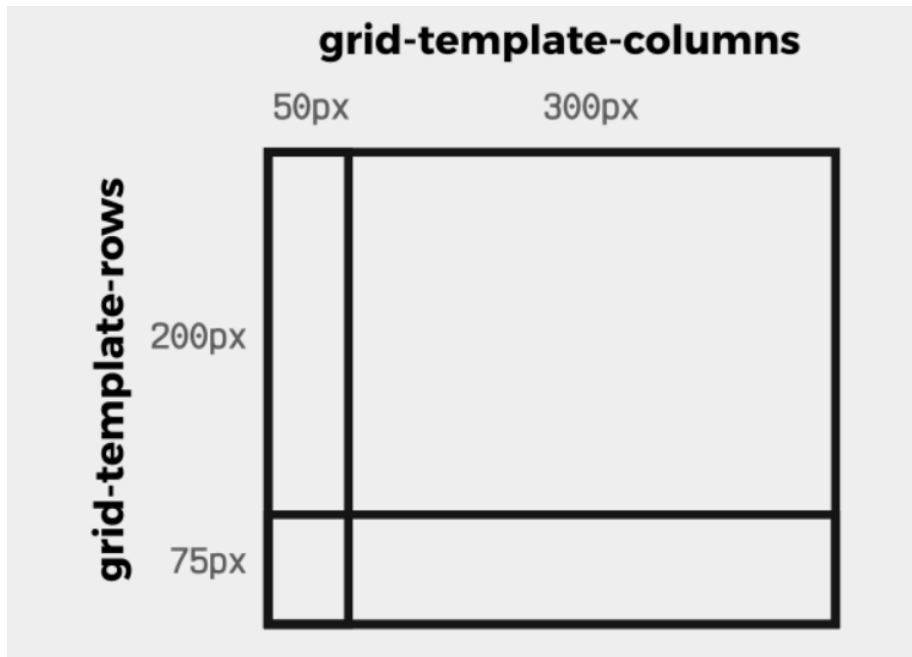
A partir de octubre de 2017, Chrome, Firefox, Safari y Edge soportan CSS grid sin prefijos.<sup>345</sup> IE 10 y 11 soportan CSS grid pero con una especificación obsoleta. En móviles, todos los navegadores modernos soportan CSS grid excepto Opera Mini y UC Browser. Los desarrolladores web orientados a navegadores más antiguos pueden utilizar Modernizr 3.5.0 para detectar y degradar con gracia la página web según sea necesario

## Grid con filas y columnas

Es posible crear cuadrículas con un tamaño explícito. Para ello, sólo tenemos que usar las propiedades CSS grid-template-columns y grid-template-rows, que sirven para indicar las dimensiones de cada celda de la cuadrícula, diferenciando entre columnas y filas. Las propiedades son las siguientes:

Conociendo estas dos propiedades, asumamos el siguiente código CSS:

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

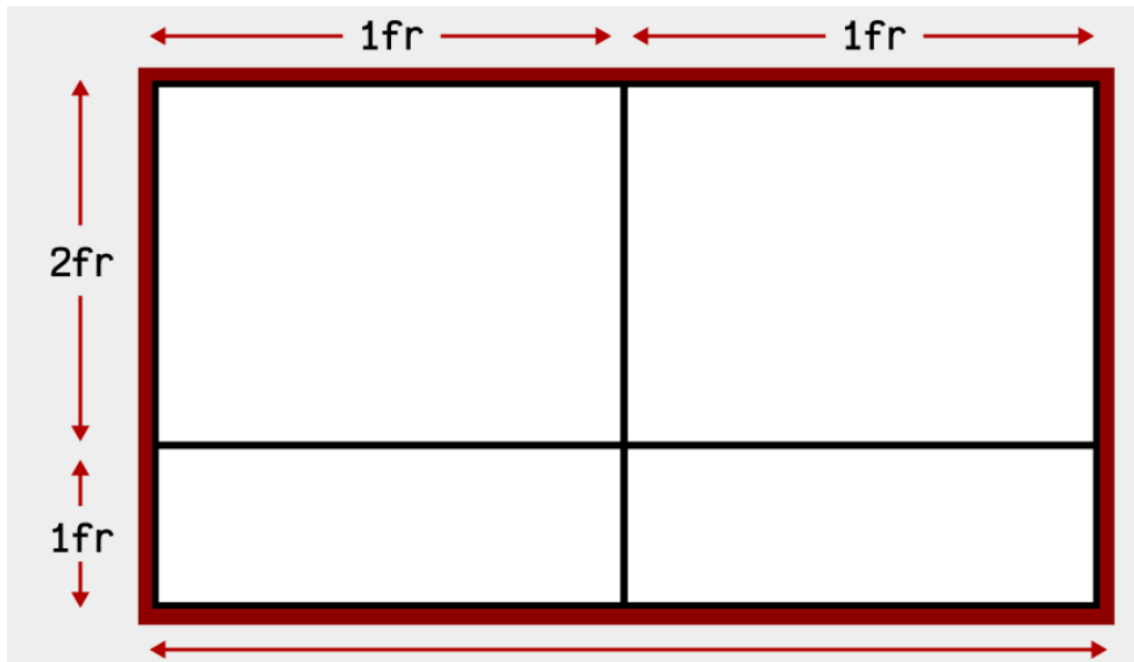


Grid-template. (s. f.). [Gráfico]. Grid CSS. <https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>

## Unidad fracción restante

También podemos utilizar otras unidades (o incluso combinarlas) como porcentajes, la palabra clave auto (que obtiene el tamaño restante) o la unidad especial de Grid CSS fr (fraction), que simboliza una fracción de espacio restante en el grid.

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



fr. (s. f.). [Gráfico]. Grid CSS. <https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>

Se pueden combinar varias unidades diferentes, pudiendo utilizar píxeles (px) y fracciones restantes (fr), porcentajes (%) y fracciones restantes (fr) o combinaciones similares.

### Filas y columnas repetitivas

En algunos casos, en las propiedades `grid-template-columns` y `grid-template-rows` podemos necesitar indicar las mismas cantidades un número alto de veces, resultando repetitivo y molesto. Se puede utilizar la expresión `repeat()` para indicar repetición de valores, indicando el número de veces que se repiten y el tamaño en cuestión.

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

En lugar de:

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

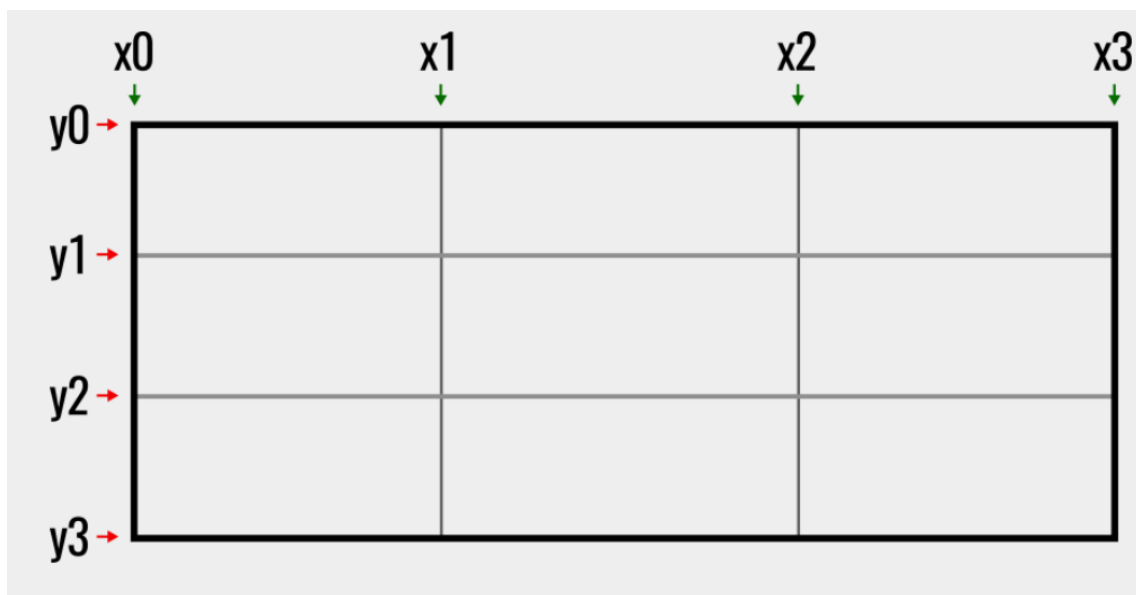
```
grid-template-columns: 100px 50px 50px 200px;  
grid-template-rows: 50px 100px 50px 100px;;  
}
```

## Grid con líneas nombradas

Con Grid CSS también tenemos la posibilidad de usar «linenames», o lo que es lo mismo, ponerle nombre a las líneas de nuestro sistema grid. Vamos a verlo con un ejemplo, donde probablemente se vea mucho mejor. Partamos de esta estructura HTML:

```
<div class="grid">  
  <div class="header">Header</div>  
  <div class="sidebar">Sidebar</div>  
  <div class="content">Content</div>  
  <div class="footer">Footer</div>  
</div>
```

Los nombres de las clases ya dan una idea del contenido que tendrán. Ahora, mediante Grid CSS lo que haremos es darle una estructura definida. Para ello, vamos a considerar los siguientes nombres para las líneas de nuestro grid, utilizando X para las posiciones en el eje X y utilizando Y para las posiciones en el eje Y:



Estructura. (s. f.). [Gráfico]. Grid CSS. <https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>

Teniendo esto en cuenta, lo único que tenemos que hacer es indicar estos nombres entre corchetes, justo antes de la medida que establecimos como vimos en apartados anteriores. Obsérvese que también se coloca una nombre de línea final sin medida a continuación, que representa la línea final:

```
.grid {  
  display: grid;  
  grid-template-columns: [x0] 1fr [x1] 1fr [x2] 1fr [x3];  
  grid-template-rows: [y0] 1fr [y1] 1fr [y2] 1fr [y3];  
}
```

Ahora, teniendo los nombres, sólo quedaría delimitar que zonas del grid queremos que ocupe cada uno de nuestros bloques <div> del grid. Para ello utilizaremos las propiedades grid-column-start, grid-column-end y grid-row-start, grid-row-end. También podríamos utilizar sus propiedades de atajo grid-column y grid-row.

```
.header {  
  background: darkred;  
  grid-column-start: x0;  
  grid-column-end: x3;  
  
  /* Equivalente a */  
  grid-column: x0 / x3;  
}
```

```
.sidebar {  
  background: black;  
  grid-row: y1 / y2;  
  color: white;  
}
```

```
.content {  
  background: orange;  
  grid-column: x1 / x3;  
  grid-row: y1 / y3;  
}
```

```
.footer {  
  background: green;  
  grid-column: x0 / x3;  
  grid-row: y2;  
}
```

Hemos aplicado la siguiente estructura:

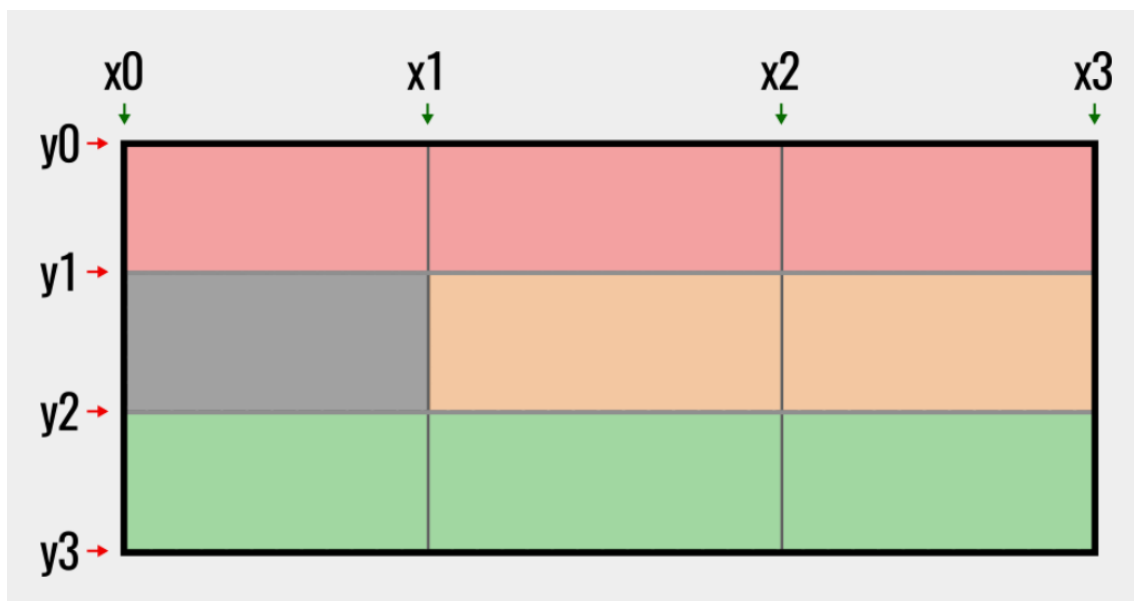
Zona .header desde la columna x0 a x3.

Zona .sidebar desde la fila y1 a y2.

Zona .content desde la columna x1 a x3 y desde la fila y1 a y3.

Zona .footer desde la columna x0 a x3 en la fila y2.

Por lo que nuestra estructura grid quedaría así:



Estructura2. (s. f.). [Gráfico]. Grid CSS. <https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>

## Grid por áreas

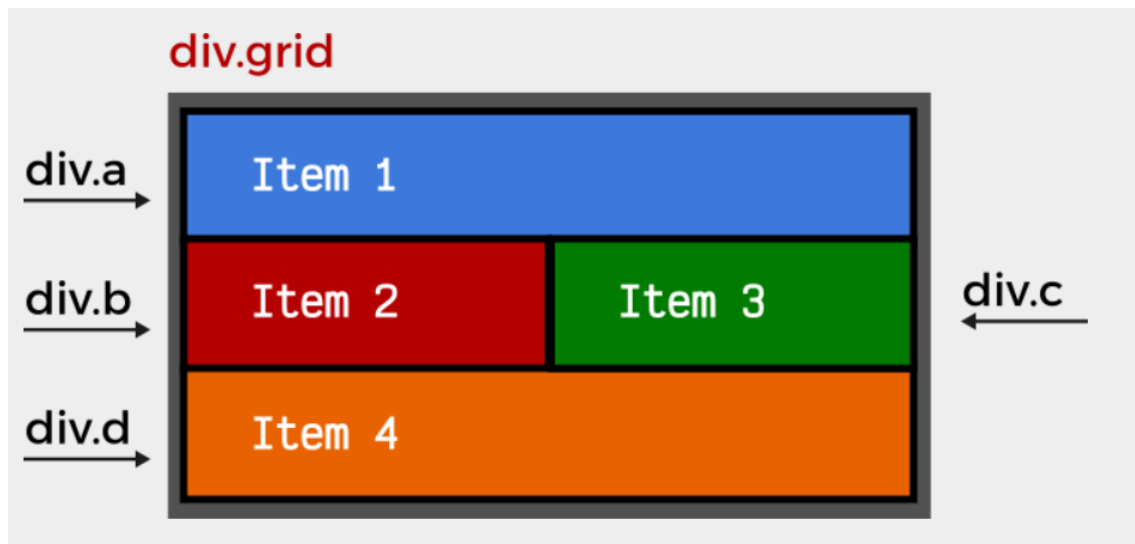
Mediante los grids CSS es posible indicar el nombre y posición concreta de cada área de una cuadrícula. Para ello utilizaremos la propiedad `grid-template-areas`, donde debemos especificar el orden de las áreas en la cuadrícula. Posteriormente, en cada ítem hijo, utilizamos la propiedad `grid-area` para indicar el nombre del área del que se trata:

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

De esta forma, es muy sencillo crear una cuadrícula altamente personalizada en apenas unas cuantas líneas de CSS, con mucha flexibilidad en la disposición y posición de cada área:

```
.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 }
```

Aplicando este código, conseguiríamos una cuadrícula donde:



Div.x. (s. f.). [Gráfico]. Grid CSS. <https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>

### Bibliografía

Grid CSS. (n.d.). Retrieved April 20, 2022, from Lenguajecss.com website: <https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>

Wikipedia contributors. (n.d.). CSS grid layout. Retrieved from Wikipedia, The Free Encyclopedia website: [https://es.wikipedia.org/w/index.php?title=CSS\\_grid\\_layout&oldid=131716163](https://es.wikipedia.org/w/index.php?title=CSS_grid_layout&oldid=131716163)