

Características principales del CSS GRID:

1. La facilidad de posicionamiento:

CSS Grid tiene muchas opciones sintácticas sobre cómo posicionar el contenido, propiedades que toman números de línea y también pueden abarcar el contenido a través de múltiples celdas. Uno puede usar nombres de área autodefinidos para asignar contenido, como por ejemplo la propiedad "Grid-área" (Área de cuadrícula) para este propósito. Con Grid Uno puede abarcar filas y columnas, creando diseños bidimensionales, el más básico es dando una definición de punto de inicio y fin a la columna y así mismo con la fila. A continuación, se podrá evidenciar los ejemplos de posicionamiento.

```
grid-column-start: 2;  
grid-column-end: 3;  
grid-row-start: 1;  
grid-row-end: 2;  
  
#positioning-with-grid .b {  
  grid-column: 2 / 3;  
  grid-row: 2 / 3;  
}  
  
#positioning-with-grid .c {  
  grid-column: col 3;  
  grid-row: row 2;  
}  
  
#positioning-with-grid .d {  
  grid-column: 1 / 2;  
  grid-row: 1 / span 2;  
}
```

Imagen 1. Facilidad de posicionamiento (Penagos, 2021)



Imagen 1.1 Captura de salida de los ejemplos dados anteriormente (Penagos, 2021)

```

1 <div class="wrapper">
2   <div class="box a">A</div>
3   <div class="box b">B</div>
4   <div class="box c">C</div>
5   <div class="box d">D</div>
6   <div class="box e">E</div>
7   <div class="box f">F</div>
8 </div>

```

Imagen 1.2 Código simple (HTML) usado en los ejemplos anteriores (Penagos, 2021)

2.Ubicación automática:

No todos los elementos de una cuadrícula deben tener una posición definida. Esto se maneja llenando el espacio libre en el diseño de la cuadrícula con esos elementos indefinidos. Permitiendo así contenido dinámico, colocación y pedido sin ninguna intervención (**Ver imagen 1.3 y 1.4**).

3. Repetir plantilla

Las filas y columnas se pueden definir utilizando la palabra clave repeat. De esta manera se crea un ahorro de tiempo y espacio. Este es un método para colocar varias filas o columnas una al lado de la otra con el mismo tamaño (**Ver código css en la imagen 1.5, 1.3 y 1.4**)

4. posicionamiento absoluto

Los elementos se pueden colocar específicamente en una cuadrícula css. Para esto se elige una posición de anclaje al que está conectado el elemento. El atributo en cuestión es "position: absolute"; el ancla se puede elegir con, por ejemplo, las propiedades "top", "bottom", "left" o "right". (**Ver código css en la imagen 1.6 imagen posicionada en una cuadrícula, 1.3 captura de pantalla de salida y código html correspondiente 1.4**).

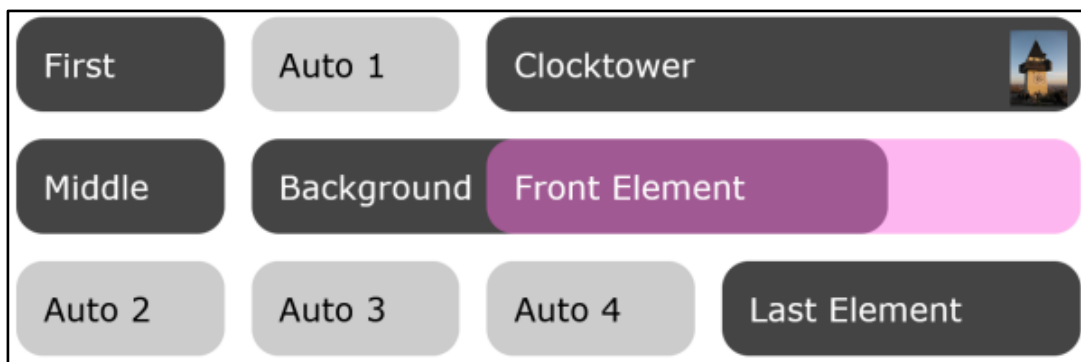


Imagen 1.3 Captura de pantalla de salida (Penagos, 2021)

```

1 <div class="wrapper">
2   <div class="box a">First</div>
3   <div class="box b">Clocktower <br /></div>
4   <div class="box c">Middle</div>
5   <div class="box d">Background</div>
6   <div class="box e">Last Element</div>
7   <div class="box f">Front Element</div>
8   <div class="box g">Auto 1</div>
9   <div class="box h">Auto 2</div>
10  <div class="box i">Auto 3</div>
11  <div class="box j">Auto 4</div>
12 </div>

```

Imagen 1.4 Código html correspondiente (Penagos, 2021)

```

1 #grid-repeat .wrapper {
2   ...
3   grid-template-columns: repeat(5, [col] 12rem ) ;
4   grid-template-rows: repeat(3, [row] auto );
5 }

```

Imagen 1.5 Ejemplo donde se crea una plantilla con la palabra clave "repeat", lo que permite rápido diseño formal (Penagos, 2021)

```

1 #grid-layering .clocktower {
2   ...
3   position: absolute;
4   top: 1rem;
5   right: 1rem;
6 }

```

Imagen 1.6 Ejemplo de una imagen posicionada absolutamente en una cuadrícula CSS (Penagos, 2021)

```

1 <div class="wrapper">
2   <div class="box header">Header</div>
3   <div class="box sidebar">Sidebar</div>
4   <div class="box sidebar2">Sidebar 2</div>
5   <div class="box content">Content
6     Here is some long text that will fill this pretty broad element with stuff.</div>
7   <div class="box footer">Footer</div>
8 </div>

```

Imagen 1.7 Código HTML simple usado para los ejemplos receptivos (Penagos, 2021)

5. Area de cuadrícula

El posicionamiento también se puede hacer usando áreas. En lugar de solamente usar números para enumerar, se pueden crear nombres fijos con el atributo "grid-template-areas". Los nombres se asignan simplemente colocando los alias de una manera similar a la Cuadrícula que es muy intuitiva. Las áreas también pueden abarcar varias celdas [Andrew 2019]. **(ver imagen 1.10 para ver pantalla de salida).**

6. Columnas y filas de plantilla

El tamaño de las columnas o las filas, puede ser definido usando porcentajes del espacio disponible, o usando el atributo auto que se encarga de llenar el espacio restante. También se pueden utilizar Las fracciones para dividir el espacio en varias partes. **(ver imagen 1.8 para ver la captura de la pantalla de salida).**

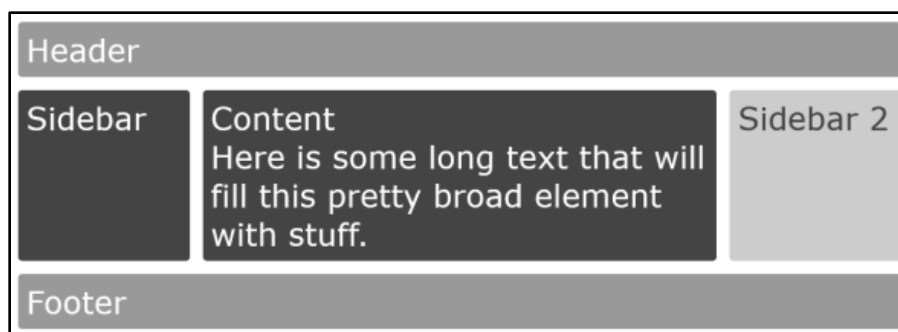


Imagen 1.8 Ejemplo de cuadrícula en el diseño de escritorio (Penagos, 2021)

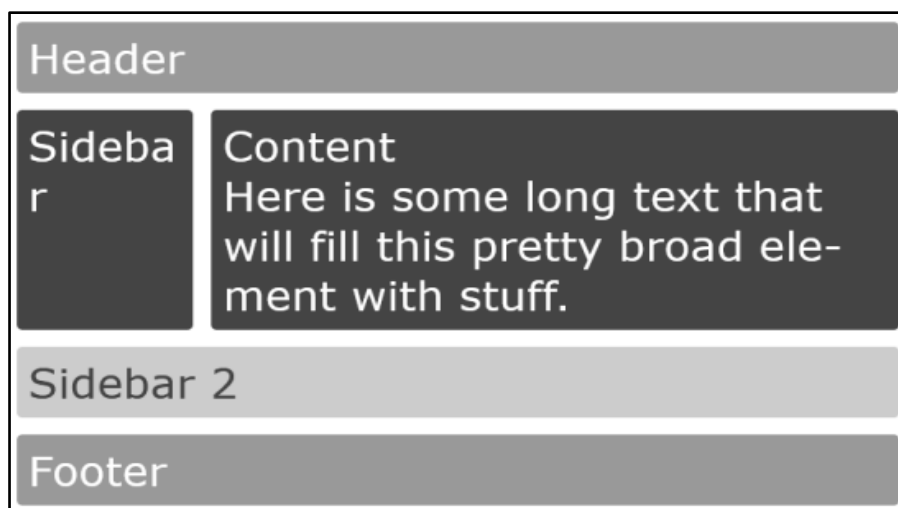


Imagen 1.9 Ejemplo de cuadrícula en el diseño de una Tablet (Penagos, 2021)

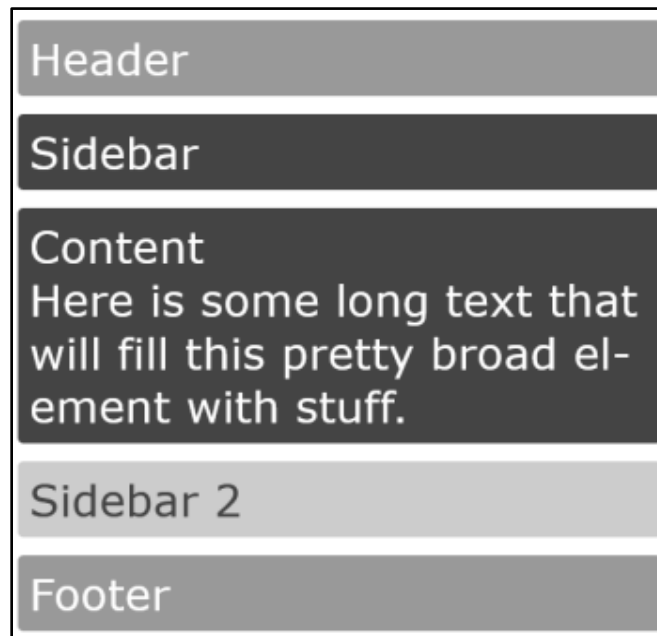


Imagen 1.10 Ejemplo de cuadrícula en el diseño de un móvil (Penagos, 2021)

7. Consultas multimedia y cuadrícula CSS

Las consultas multimedia se pueden ser utilizadas para comprobar el tamaño de la pantalla del cliente y hacer que el diseño responda. Esto se hace mediante la comprobación del ancho mínimo y luego definiendo un nuevo diseño de cuadrícula. El diseño predeterminado simplemente se sobrescribe con el nuevo. Esto permite un diseño drástico e incluso cambios de orden de contenido de una manera muy fácil. Los sitios web se pueden instalar en diferentes dispositivos como móviles y de escritorio sin mucho trabajo ni cambios en el HTML. (Niess, Roubal, Thurner & Barba, 2019) **(ver las imágenes 1.8, 1.9, y 1.10 para ver las capturas de pantallas de salida).**

8. Capas

Los elementos de una cuadrícula pueden ser superpuestos entre sí con la propiedad "z-index". Esto permite añadir una tercera dimensión a los diseños. La posición posterior más lejana estaría en cero, y cada aumento por uno es una capa más hacia el frente. También se puede combinar con transparencias. **(ver Imagen 1.3 para ver la captura de pantalla de salida).**

9. Soporte del navegador

Soporte para Grid hay en Chrome, Firefox y Safari desde marzo de 2017 y en octubre de este mismo año se envió soporte para Edge. Hay navegadores

antiguos que no son compatibles con Grid, porque simplemente ignoran las propiedades en CSS que no comprenden. Para garantizar soporte en navegadores más antiguos solo es necesario agregar consultas de características a las propiedades que no son exclusivas de CSS Grid. En la mayoría de los casos solo será necesario sobre escribir propiedades como el ancho o margen. **(ver Imagen 1.11 para ver una descripción general de la compatibilidad de Grid).**

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android
		2-39	4-28									
		40-51	29-56		10-27							
6-9	12-15	52-53	57	3.1-10	28-43	3.2-10.2						
10	16-17	54-70	58-77	10.1-12.1	44-63	10.3-13.1		2.1-4.4.4	12-12.1			
11	18	71	78	13	64	13.2	all	76	46	78	68	12.12
	76	72-73	79-81	TP		13.3						

Imagen 1.11 Descripción general de la compatibilidad de Grid (Penagos, 2021)

Tomado de:

Niess, G., Roubal, A., Thurner, S. y Roque, EB (2019). Diseños de cuadrícula CSS.
<https://courses.isds.tugraz.at/iaweb/surveys/ws2019/iaweb-ws2019-g1-survey-css-grid.pdf>

Bibliografía

Ochoa Torres, J. (04,12,2018). O.V.I. Definición de HTML5. [Archivo de Video]. Recuperado de <http://hdl.handle.net/10596/22593>

Recio García, J. A. (2016). HTML5, CSS3 y JQuery: curso práctico. RA-MA Editorial. (Pág. 52 a 142) Recuperado de <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/106494?page=52>

Celaya Luna, A. (2014). Creación de páginas web: HTML 5 (pág. 1 a 60). Recuperado de <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/56045>