

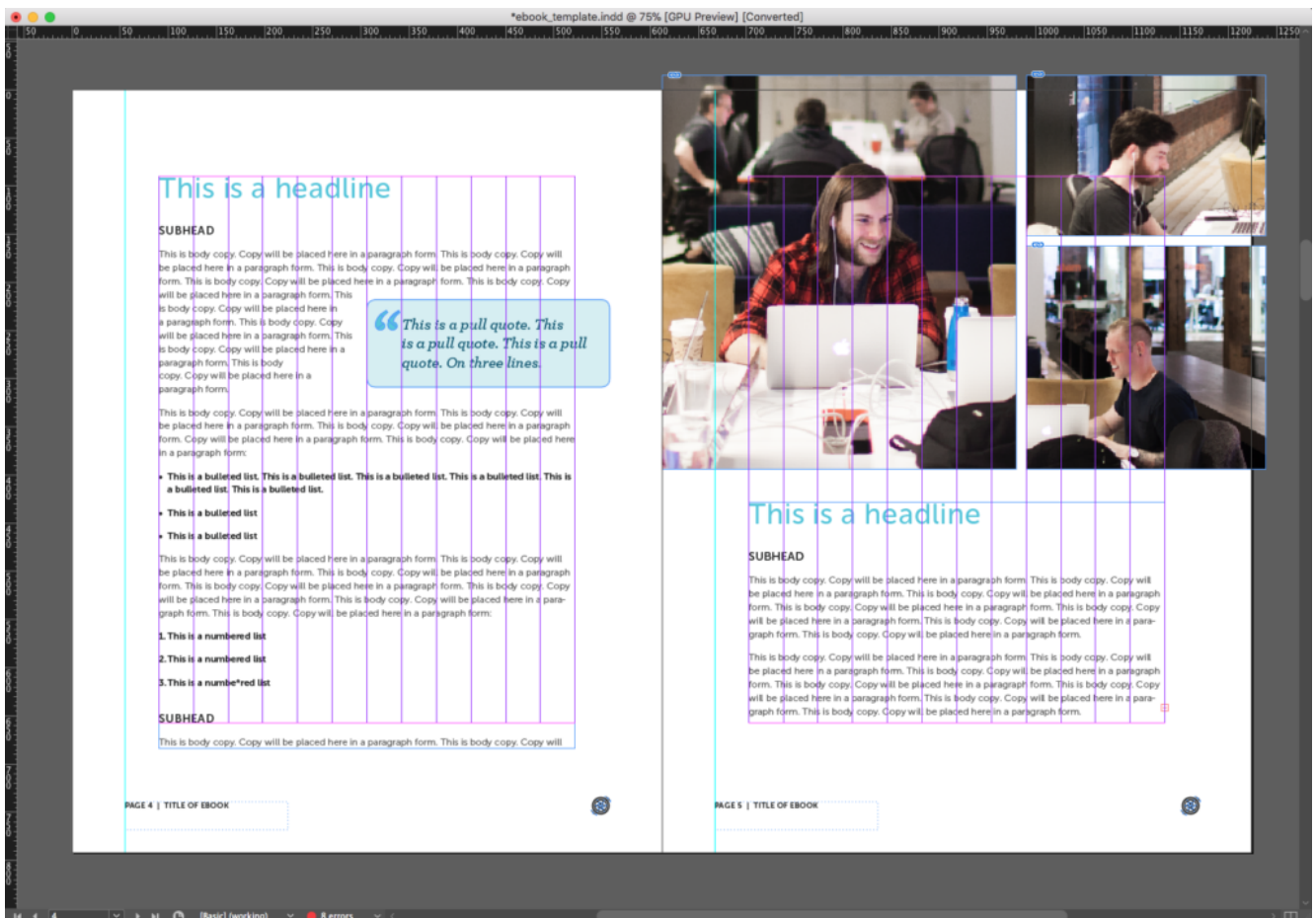


DEVELOPMENT

# How to create a simple layout with CSS Grid Layouts

Abbey Fitzgerald — December 19, 2017

Si tu has alguna vez interesado en diseño impresion, tu sabes donde la idea de  
If you've ever dabbled in print design, you know where the idea of **grid layouts** originated  
Un grid es una precisa herramienta medición usada posicionar elementos diseño sobre una pagina  
from. A grid is a precise measurement tool used to position design elements on a page. This  
es a menudo usada sobre la web hacer contenido organizado y uniforme haciendo para una  
idea is often used on the web to make content organized and uniform, making for an  
mejorada experiencia visualización para sus usuarios  
improved viewing experience for your users.



Cuando la practica de diseño web era nueva , distribuciones estaban simple preciosas  
 When the practice of web design was new, layouts were pretty simple. Header, sidebar, content area, and footer. Now, as the web has evolved, our layouts have become more complejas y nosotros tenemos un montón más contendere con como diseñadores web nosotros a menudo necesitamos numerosas regiones de contenido , diseño responsivo , multiples paginas layout de diseño plantillas numerous content regions, responsive design, multiple page layout template designs, entre algunas otras cosas Y e implementar el diseño y para que se muestre correctamente amongst many other things. And to implement the design and get it to display correctly, floats and positioning son requeridos suena simple pero eso puede a veces ser difícil trabajar con ello are required. Floats sound simple, but these can sometimes be tricky to work with.

afortunadamente, grid proporciona una facil camino para diseñar y mostrar contenido web ello es un alucinante  
 Luckily, the grid provides an easier way to design and display web content. It's an awesome tiempo ser un diseñador porque están rapidamente ganando tracción ello quiza sienta como time to be a designer because CSS Grid Layouts are quickly gaining traction. It may feel like grids have been a new idea in web design in recent years, but believe it or not, CSS Grid han sido una nueva idea en el diseño web en reciente años pero creer ello o no CSS Grid Layouts ha sido en in los trabajos para algunos años Navegador soportado para ello pasando muy rapidamente have been in the works for many years. Browser support for them happened very asi ahora ello puede ayudar facil algunas de las frustraciones alrededor posicionando que diseñadores quickly, so now they can help ease some of the frustration around positioning that designers han tenido tratar con para algunos años have had to deal with for many years.

ello es finalmente momento romper dentro del Grid y explorar las nuevas posibilidades diseño  
 It's finally time to break into the grid and explore the new design possibilities!

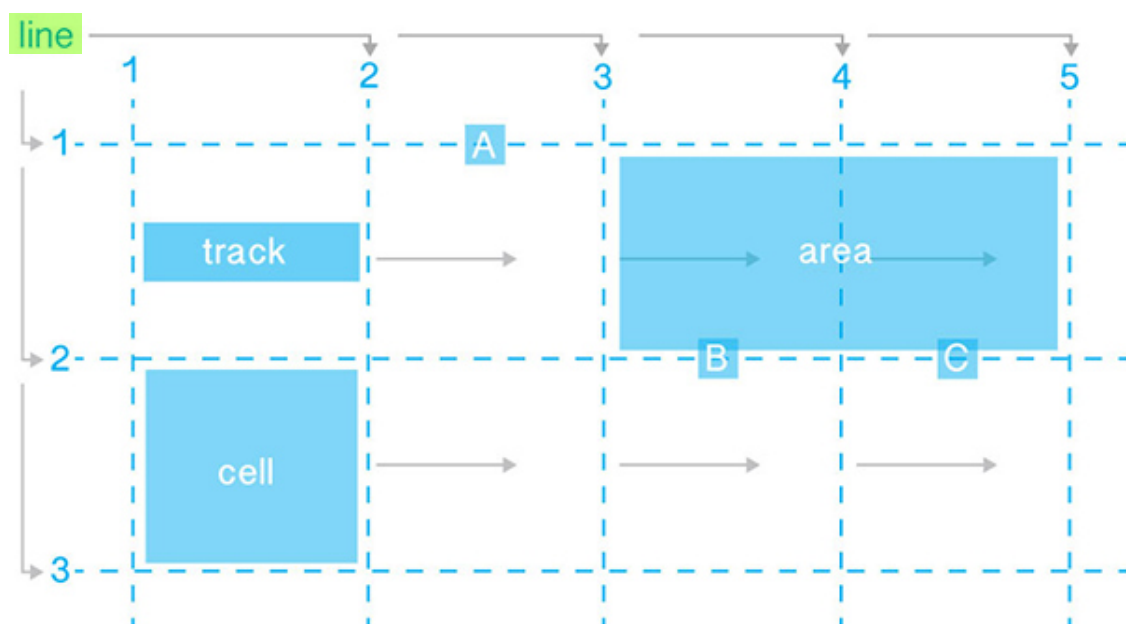
# CSS Grid support

Elo todo depende en los navegadores necesita soportar pero ahí hay probabilidad que  
It all **depends** on the **browsers you need to support**, but there's a **pretty good** chance that  
tu no tienes preocuparte sobre cuando usando puedes siempre visitar  
you do not have to worry about "flag mode" when using CSS Grids. You can always visit the  
documentacion sobre comprobacion pero soporta para tiene drasticamente  
Can I Use documentation to double check, but support for **CSS Grids** has dramatically  
incrementado y algunos diseñadores estan disponible usar en producción  
increased and many designers are able to use it in production, which is cause for celebration!

Our free local development app, [Local](#), will help you simplify your workflow and safely experiment with your site. Try it today!

## CSS Grid Layout basics

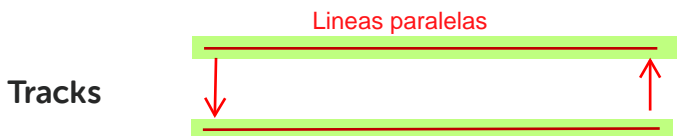
By utilizing **CSS**, **grid layouts** will help **define content areas** on **your web page**. There is a **relativamente nuevo establecido de propiedades definidas en el** relatively new set of properties defined in the **CSS Grid Layout Specifications**. This is a great **recurso a referenciar** resource to reference when learning more about this new way of layout design. **CSS Grid Layouts** **ayuda a simplificar cosas y hacer creando distribuciones faciles** help to simplify things and make creating layouts easier. Think of the grid as a **estructura donde medidas pueden ser definido** structure where measurements can be defined.



# Parts of the grid

## Lines

In this **case**, there are **five vertical lines** and **three horizontal lines**. Lines are given numbers **que comienzan desde 1** **Lineas reciben numeros** starting from one. Vertical lines are shown from left to right in this example, but this depends **Lineas verticales son mostradas desde izquierda a derecha en este ejemplo pero esto depende en la direccion escrita** on the writing direction. Because this shows reading from left to right, if it is from right to left, **Porque esto muestra leyendo desde izquierda a derecha**, things will be reversed. Lines can be given names (optional), which helps with referencing **si ello es desde der a izq** them in CSS. **Lineas pueden estar dados nombre opcional el cual ayuda con referenciando ello en CSS**



una banda es el espacio entre **2** **lineas paralelas** en el diagrama **hay 4 franjas/bandas verticales**  
A track is the space between two parallel lines. In the diagram, there are four vertical tracks **y 2 horizontales bandas/franjas** and two horizontal tracks. There is a great dual effort between lines and tracks. Lines are a **Hay una gran compartido esfuerzo entre lineas y franjas/bandas** **Lineas son una genial manera referenciar donde el contenido comienza y para** **bandas estan donde el contenido actualmente va** great way to reference where the content starts and stops. Tracks are where the content actually goes.

## Cells

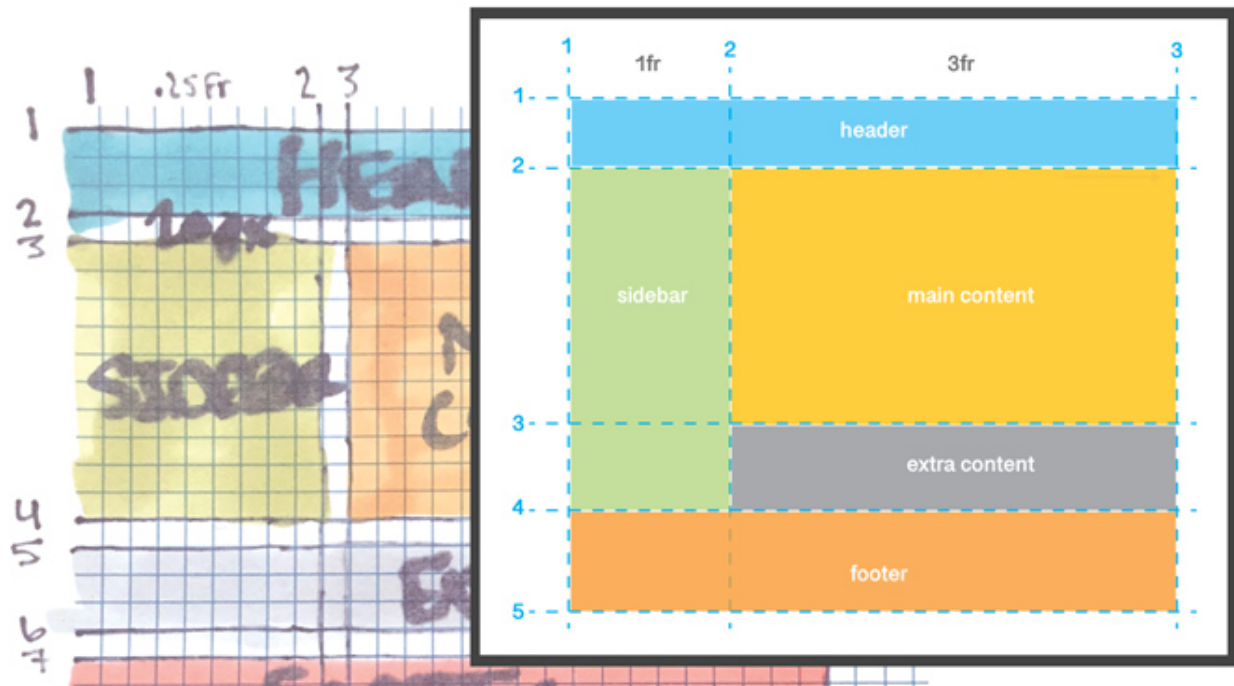
**son encontradas donde una horizontal y una vertical banda se encuentra** **Hay 8 celdas en el diagrama**  
Cells are found where a horizontal and a vertical track meet. There are eight cells in the diagram.

## Areas

**vienen entra juego cuando especifica areas** **una area es un rectangulo comparte que puede espaciar un numero de celdas**  
Cells come into play when specifying areas. An area is a rectangular shape that can span a number of cells. Like lines, areas can be optionally named. There are a few labels included in **como lineas, areas pueden ser opcionalmente nombrados, hay una pocas etiquetas incluidos in el el diagrama** the diagram: "A," "B," and "C."

## Creating a grid layout

**ello puede ser ayuda esbozar cosas antes de comenzar su diseño** **No hay sustituto para un buen viejo grafico papel**  
It can be helpful to sketch things out before starting your layout. There's no substitute for good old graph paper.



## HTML for the grid

```
&lt;div class="container"&gt;
  &lt;div class="grid header"&gt;Header&lt;/div&gt;
  &lt;div class="grid sidebar"&gt;Sidebar&lt;/div&gt;
  &lt;div class="grid content"&gt;Main Content&lt;/div&gt;
  &lt;div class="grid extra"&gt;Extra Content&lt;/div&gt;
  &lt;div class="grid footer"&gt;Footer&lt;/div&gt;
&lt;/div&gt;
```

The **container** is very important. Within the **container** are the **different content blocks** for the **website**. The **order** of the **grid items** within the **container** does not matter. Next, we'll use CSS to place them in our layout.

## CSS styles

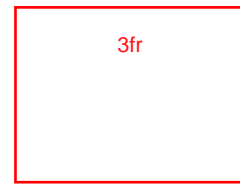
After you are done with the HTML markup, the most important declaration is done in the CSS. Within the container, you need to apply **display: grid** or **display: inline-grid**. Use **display: grid** if you would like a **block-level grid**. Use **display: inline-grid** for **inline-level grids**. To learn more, check out [this great reference guide](#).

```

.container {
  display: grid;
  grid-template-columns: 1fr 3fr;
  grid-column-gap: 10px;
  grid-row-gap: 15px;
}

.grid {
  background-color: #444;
  color: #fff;
  padding: 25px;
  font-size: 3rem;
}

```

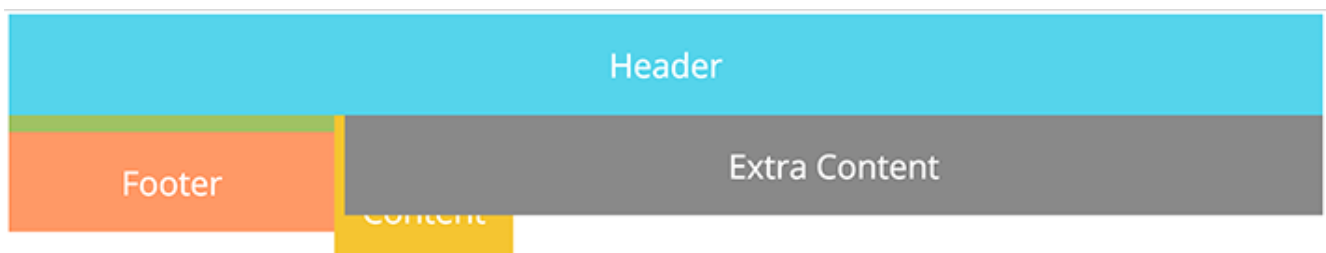


The `grid-template-columns` and `grid-template-rows` properties are used to specify the width of the rows and columns. The `grid-template-columns` have been set as `1fr` and `3fr`. This is where the grid is taking shape with the fractional units. With these values, it is apparent there are two columns and they are not equal width. The column set to `3fr` is three times wider than the other. This explains how the sidebar appears more narrow than the content area.

Hay personalizaciones responsivas como puedes crear pero usando la unidades fracción es un There are responsive customizations that can be made, but using the fractional units is a gran primer paso Espaciado entre areas puede ser controladas con el great first step. Spacing between areas can be controlled with the `grid-column-gap` and the `grid-row-gap`.

Las cosas se ven bastante ajustadas pero con un poco más especificación , cosas tomaran forma Things look pretty tight, but with a few more specifications, things will take shape.

This example starts with placing the header, but elements can be placed in whatever order trabaja la mejor para ti Si deseas comenzar con el pie , que trabajara también works best for you. If you want to start with the footer, that will work, too.



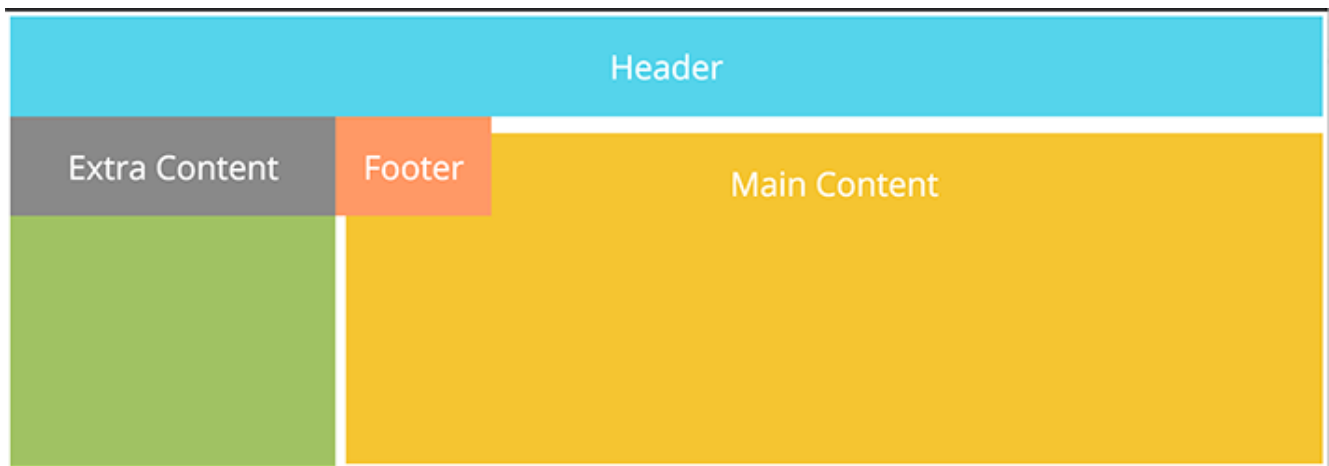
Comienza con la cabecera , la cual va desde la columna linea 1 a la columna linea 4 y linea fila 1 a la fila de la linea 2 Let's start with the header, which goes from column line 1 to column line 4 and row line 1 to row line 2. The CSS will look like this:

```
.header {
  grid-column-start: 1;
  grid-column-end: 3;
  grid-row-start: 1;
  grid-row-end: 2;
}
```

Quizás notes que la 'sidebar' es apretado y no podemos ver ello mucho. Nosotros necesitamos hacer una pequeña organización. En esta distribución, referenciando las líneas para obtener el establecer ayudara con la do a little rearranging. In this layout, referencing the lines to get the placement will help with the arrangement. The lines act as the guidelines. The header is located between the line at row one and the line at row two. For the sidebar, it will start at row three and end at line six. The header row ended at two, so this will be placed right under it. To view the example, see the project on [Codepen](#).

Nosotros usamos `grid-column-start` to specify the starting vertical line for an element. In this case, it would be set to three. The `grid-column-end` indicates the ending vertical line for an element. In this case, this property would be equal to four. Other row values will also be set this same way. The placement of the sidebar is there, it is just covered by the content area.

```
.sidebar {
  grid-column-start: 1;
  grid-column-end: 2;
  grid-row-start: 2;
  grid-row-end: 4;
  background: #a0c263;
}
```



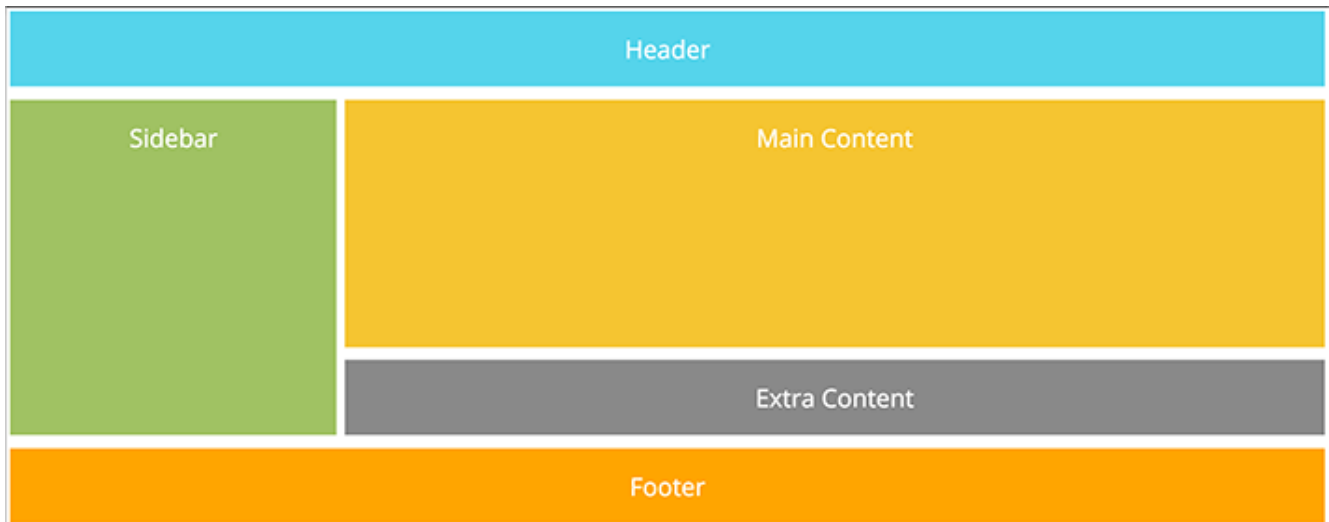
El principal contenido comienza sobre columna 3 y termina en la columna 4 El alto de la barra lateral y  
The **main content** starts at column three and ends at column four. The top of the sidebar and  
area contenido estan incluidos, asi ellos ambos tienen un **grid-row-start** of three. You may be  
preguntando como el contenido y la barra lateral son más altas quizás este  
wondering how the content and sidebar are much taller. Por poniendo un alto sobre el contenedor  
container, in this case 400px, this is now taller than the other elements.

```
.content {  
  grid-column-start: 2;  
  grid-column-end: 3;  
  grid-row-start: 2;  
  grid-row-end: 3;  
  background: #f5c531;  
  height: 400px;  
}
```

The last two content areas are the extra content area and the footer.

```
.extra {  
  grid-column-start: 2;  
  grid-column-end: 3;  
  grid-row-start: 3;  
  grid-row-end: 4;  
  background: #898989;  
}  
  
.footer {  
  grid-column-start: 1;  
  grid-column-end: 3;  
  grid-row-start: 3;  
  grid-row-end: 4;  
  background: #FFA500;  
}
```





Final y comienzo declaracion puede ser escrito mas eficientemente

End and start declarations can be written more efficiently. You may see `grid-column-start: 1;` and `grid-column-end: 3;` written in shorthand. This would look like `grid-column: 1 / 3;`. This same idea can be used to declare column row information.

## Responsive advantages

Ahora que una distribución ha sido creada , ello vera muy "escritorio"

Qué sobre dispositivos tablets y moviles

Now that a layout has been created, it seems very "desktop." What about tablets and mobile

devices? CSS Grid Layouts work well with media queries so the design can adapt to different

screen sizes. What's really cool is that you can change the content areas at these different

media query breakpoints. As a designer, this means you choose what is best for your layout

at different breakpoints. For example, if you want the "extra content" to be placed above the "content" area, you can specify the correct columns and rows.

```
/* For mobile phones: */
.header {
  grid-column-start: 1;
  grid-column-end: 4;
  grid-row-start: 1;
  grid-row-end: 2;
}

.extra {
  grid-column-start: 1;
  grid-column-end: 4;
  grid-row-start: 3;
  grid-row-end: 4;
}
```

```

.content {
  grid-column-start: 1;
  grid-column-end: 4;
  grid-row-start: 5;
  grid-row-end: 6;
  background: #f5c531;
  height: 400px;
}

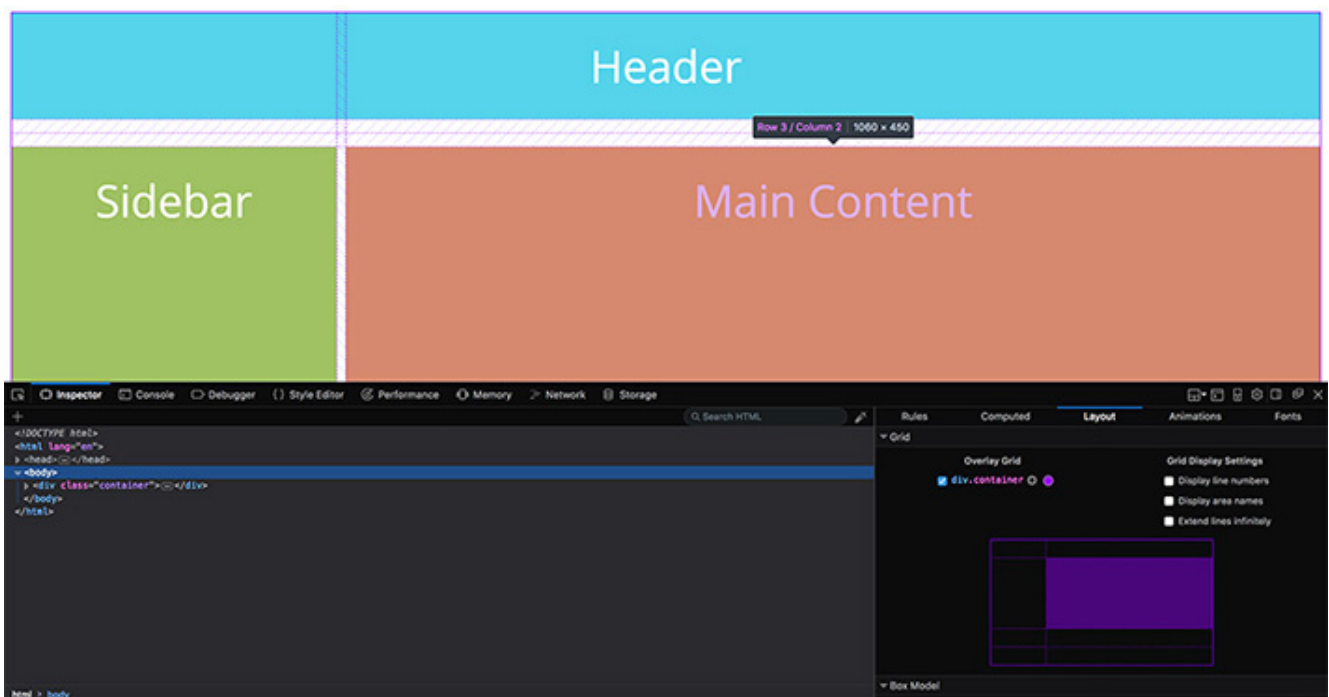
```

Por teniendo el comienzo columna sobre una y final sobre 4 nosotros hemos creado el contenido areas ancho completo  
 By having the columns start at one and end at four, we've made the content areas full width.  
 La filas cuadrículas estan colocadas sobre el cual el esto ahora sobre el contenido  
 The grid rows are placed so that the "extra content" is now above the "content."

Herramientas navegador para CSS Grid Layout

## Browser tool for CSS Grid Layout

Como tu estas diseñando layout or inspeccionado otros s ofrece una gran herramienta para  
 As you're designing layouts, or inspecting other websites, Firefox offers a great tool for  
 trabajandoo con y ello hace examinando la  
 working with CSS Grid Layouts. It is called **CSS Grid Playground** and it makes examining the  
 muy intuitiva Yendo al inspeccionador hay una opción en la distribución pestaña visualmente ver  
 grid very intuitive. By going to the inspector there's an option in the layout tab to visually see  
 como la cuadrícula esta contruida ello hara el proceso diseño facilmente tener esta herramienta explorar  
 how the grid is constructed. It will make the design process easier to have this tool to explore  
 como diferente propiedades cuadrículas a efectos el general  
 how different grid properties affect the overall grid layout and design.



CSS Grid Layouts are a new, exciting way to create web layouts. As you can see, it is pretty **facil suceder y corriendo cuando creando una simple pagina distribución** easy to be up and running when creating a simple page layout. This simple example is a good foundation for how to make more complex layouts. As this gains in popularity, there will be an advantage to this technology when designing for various devices and breakpoints with the layout customizations that can be made.

## Want more design tips? Subscribe to The Layout!

We'll send the latest design tutorials and inspiration straight to your inbox, once a week.

Yes Please!

*This article was originally published on August 3rd, 2016. It was last updated on December 6th, 2017.*

*Share this article*



HTML & CSS

WordPress 101



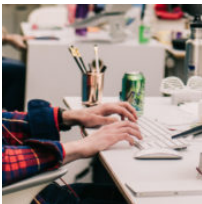
---

### *Similar articles*



#### DEVELOPMENT

39 of the best JavaScript libraries and frameworks to try in 2020



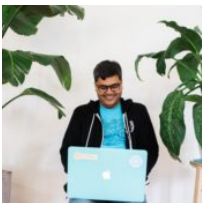
#### DEVELOPMENT, WORDPRESS

15 steps to take before launching an eCommerce store



#### DEVELOPMENT, WORDPRESS

How to add a custom scrollbar to WordPress with jQuery



#### DEVELOPMENT

7 best practices to create sustainable websites



#### DEVELOPMENT

How to use CSS3 transitions and transforms to create animations



#### DEVELOPMENT

If your site's design isn't broke, fix it anyway

---

Get the inspiration you need to do  
your best work, every Sunday!

Email address

Hook me up!

---



## Layout

Design

Development

WordPress

Marketing

Business

## Resources

Free ebooks

Discounts

Design Trends e-course

## Our products

WordPress Hosting

Local by Flywheel

White Label

## Flywheel

[About](#)

[Careers](#)

[Life at Flywheel](#)

[Product Updates](#)