





Buscador (/)

 Conéctate (/login) o Regístrate (/registro)

[Inicio \(/ \)](#) [Artículos \(/articulos \)](#) [CSS Grid VS Flexbox: ¿Cuál es más práctico? \(/articulo/css_grid_vs_flexbox:_cual_es_mas_practico_1771 \)](#)

CSS Grid VS Flexbox: ¿Cuál es más práctico?

 Gestión anuncios

CSS Grid layout

Responsive CSS

Grid HTML

Display Grid



No hace mucho tiempo, el diseño de todas las páginas HTML se hacía a través de tablas, floats y otras propiedades CSS que no eran adecuadas para diseñar páginas web complejas.

Luego vino flexbox, un modo de diseño que fue creado específicamente para crear páginas responsive robustas. Flexbox facilitó el hecho de poder alinear adecuadamente elementos y su respectivo contenido. Quizás por eso se haya convertido en el sistema CSS preferido de la mayoría de desarrolladores web.

Ahora tenemos un nuevo contendiente para el trofeo mejor-sistema-para-construir-diseños-html (sé que el título del trofeo no me lo he currado mucho). Es el potente CSS Grid, y para finales de este mes, estará disponible de forma nativa en Firefox 52 y Chrome 57, y en otros navegadores (con suerte) en breve.

Una prueba básica de diseño

Para que podamos tener una idea aproximada de cómo es desarrollar diseños con cada sistema, hemos creado la misma página HTML dos veces: una con flexbox y otra con CSS Grid.

Crearemos un diseño bastante básico. Consiste en un contenedor centrado, dentro del cual tendremos un header, una sección principal, un sidebar y un footer. Los principales "desafíos" que tendremos que resolver, siempre manteniendo el CSS y el HTML tan limpios como sea posible, son:

- Posicionar las cuatro secciones en el diseño
- Hacer que la página sea responsive (el sidebar irá del contenido principal en pantallas más pequeñas)
- Alinear el contenido del header (navegación a la izquierda y el botón a la derecha)

Como puedes ver, por el bien de la comparación, hemos hecho que todo sea muy simple. Comencemos con el primer desafío.

Desafío 1: Posicionar las cuatro secciones en el diseño

Con Flexbox

Comenzaremos con flexbox. Agregaremos `display: flex` al contenedor y dirigimos a sus hijos de forma vertical. Esto posicionará todas las secciones una debajo de otra.

```
.container {
  display: flex;
  flex-direction: column;
}
```

Ahora tenemos que crear la sección principal y el sidebar que se posicionará a su lado. Dado que los contenedores flex son generalmente unidireccionales, necesitaremos añadir un wrapper.

```
<header></header>
<div class="main-and-sidebar-wrapper">
  <section class="main"></section>
  <aside class="sidebar"></aside>
</div>
<footer></footer>
```

Ahora creamos el wrapper con `display: flex` y `flex-direction` en la dirección contraria.

```
.main-and-sidebar-wrapper {
  display: flex;
  flex-direction: row;
}
```

El último paso es definir el tamaño de la sección principal y del sidebar. Queremos que el contenido principal sea tres veces el tamaño del sidebar, lo cual no es difícil de hacer con flex o porcentajes.

```
.main {
  flex: 3;
  margin-right: 60px;
}
.sidebar {
  flex: 1;
}
```

Como puedes ver, hacer esto en Flexbox no es demasiado difícil, pero es verdad que necesitamos bastantes propiedades CSS, y muchos elementos HTML adicionales. Vamos a ver cómo hacer esto en CSS Grid.

Con CSS Grid

Existen dos métodos distintos a la hora de utilizar CSS Grid, pero nosotros utilizaremos la sintaxis `grid-template-areas`, ya que me parece más adecuada para lo que vamos a hacer.

Primero vamos a definir cuatro `grid-area`-s, una para cada sección de la página:

```
<header></header>
<section class="main"></section>
<aside class="sidebar"></aside>
<footer></footer>
```

```
header {
  grid-area: header;
}
.main {
  grid-area: main;
}
.sidebar {
  grid-area: sidebar;
}
footer {
  grid-area: footer;
}
```

Ahora podemos configurar nuestro grid y asignar la ubicación de cada área. El código de a continuación puede parecer bastante complicado al principio, pero una vez que conozcas el sistema de cuadrículas se vuelve muy fácil de entender.

```

.container {
  display: grid;

  /* Define the size and number of columns in our grid.
  The fr unit works similar to flex:
  fr columns will share the free space in the row in proportion to their value.
  We will have 2 columns - the first will be 3x the size of the second. */
  grid-template-columns: 3fr 1fr;

  /* Assign the grid areas we did earlier to specific places on the grid.
  First row is all header.
  Second row is shared between main and sidebar.
  Last row is all footer. */
  grid-template-areas:
    "header header"
    "main sidebar"
    "footer footer";

  /* The gutters between each grid cell will be 60 pixels. */
  grid-gap: 60px;
}

```

¡Y eso es todo! Nuestro diseño ahora seguirá la estructura anterior, y debido a la forma en que lo hemos creado, ni siquiera tendremos que lidiar con márgenes y paddings.

Desafío 2: Hacer que la página sea responsive

Con Flexbox

La ejecución de este paso está fuertemente conectado al anterior paso. Con Flexbox podemos cambiar la flex-direction del wrapper, y ajustar ciertos márgenes.

```

@media (max-width: 600px) {
  .main-and-sidebar-wrapper {
    flex-direction: column;
  }

  .main {
    margin-right: 0;
    margin-bottom: 60px;
  }
}

```

Nuestra página es muy simple y tampoco hay que currarse mucho las media.queries, pero en un diseño más complejo tendremos que redefinir cientos y cientos de cosas.

Con CSS Grid

Ya que tenemos definidas nuestras áreas, necesitamos reordenarlas en una media-query. Podemos utilizar la misma configuración de columna.

```

@media (max-width: 600px) {
  .container {
    /* Realign the grid areas for a mobile layout. */
    grid-template-areas:
      "header header"
      "main main"
      "sidebar sidebar"
      "footer footer";
  }
}

```

O, podemos redefinir el diseño entero desde cero si creemos que es una solución más limpia.

```

@media (max-width: 600px) {
  .container {
    /* Redefine the grid into a single column layout. */
    grid-template-columns: 1fr;
    grid-template-areas:
      "header"
      "main"
      "sidebar"
      "footer";
  }
}

```

Desafío 3: Alinear el contenido del header

Con Flexbox

Nuestro header incluirá ciertos enlaces para la navegación y un botón. Queremos que la navegación esté a la izquierda y el botón a la derecha. Los enlaces dentro del nav tienen que estar correctamente alineados uno al lado del otro.

```
<header>
  <nav>
    <li><a href="#"><h1>Logo</h1></a></li>
    <li><a href="#">Link</a></li>
    <li><a href="#">Link</a></li>
  </nav>
  <button>Button</button>
</header>
```

Una vez tenemos esto, hay que hacer que el header sea responsive.

```
header {
  display: flex;
  justify-content: space-between;
}
```

Ahora la navegación y el botón están correctamente alineados. Todo lo que queda por hacer es que los elementos dentro del <nav> ir estén de forma horizontal. Es más fácil usar display: inline-block aquí, pero ya que vamos full flexbox, vamos a aplicar la solución flexbox-only:

```
header nav {
  display: flex;
  align-items: baseline;
}
```

¡Solo dos líneas! No está nada mal.

Con CSS Grid

Para dividir la navegación y el botón tendremos que poner el header a display:grid y configurar una cuadrícula de 2 columnas. También necesitaremos dos líneas adicionales de CSS para ajustarlos a sus respectivos bordes.

```
header{
  display: grid;
  grid-template-columns: 1fr 1fr;
}
header nav {
  justify-self: start;
}
header button {
  justify-self: end;
}
```

Con esto los enlaces no estarán correctamente alineados. Para ello, tenemos que definir otro subgrid.

```
header nav {
  display: grid;
  grid-template-columns: auto 1fr 1fr;
  align-items: end;
}
```

Está claro que CSS Grid ha tenido problemas con esta parte del diseño. Esto no me sorprende, su enfoque es alinear los contenedores, no tanto el contenido dentro de ellos. Simplemente no está destinado a hacer esos toques finales.

La verdad es que no hay un sistema mejor que otro. Tanto flexbox como Grid CSS son buenos en distintos aspectos y deben utilizarse juntos, y no como alternativas entre sí.

Fuente: [tutorialzine.com \(http://tutorialzine.com/2017/03/css-grid-vs-flexbox/\)](http://tutorialzine.com/2017/03/css-grid-vs-flexbox/)

COMPARTE ESTE ARTÍCULO

✉ ENVIAR A UN AMIGO

f COMPARTIR EN FACEBOOK

🐦 COMPARTIR EN TWITTER

⌘ COMPARTIR EN GOOGLE +

ARTÍCULO ANTERIOR

Cómo crear una galería de imágenes dinámica con jQuery, PHP y MySQL
(/articulo/como_crear_una_galeria_de_imagenes_dinamica_con_jquery-_php_y_mysqli_1770)

SIGUIENTE ARTÍCULO

Cómo crear un menú lateral con slide para diseños responsive
(/articulo/como_crear_un_menu_lateral_con_slide_para_disenos_responsive_1773)

syredes-programacionnet&utm_medium=referral&utm_content=thumbnails-b:Below Article Thumbnails:)
syredes-programacionnet&utm_medium=referral&utm_content=thumbnails-b:Below Article Thumbnails:)
syredes-programacionnet&utm_medium=referral&utm_content=thumbnails-b:Below Article Thumbnails:)
Recomendado Para Ti

(https://article.everquote.com/?h1=startup&h2=brilliant_company&auuid=27b4cb79-e1cf-4df2-b07d-47fb496b2ce6&&tid=584&subid=2014&utm_medium=mediosyredes-programacionnet&utm_title=San+Francisco%2C+California%3A+This+Brilliant+Company+Is+Disrupting+a+%24200+Billion+Industry&utm_thumbnail=https%3A%2F%2Fs3.amazonaws.com%20a30-442c-9695-a0f9382e91b8.jpg)

San Francisco, California: This Brilliant Company Is Disrupting a \$200 Billion Industry

EverQuote Insurance Quotes

(https://article.everquote.com/?h1=startup&h2=brilliant_company&auuid=27b4cb79-e1cf-4df2-b07d-47fb496b2ce6&&tid=584&subid=2014&utm_medium=mediosyredes-programacionnet&utm_title=San+Francisco%2C+California%3A+This+Brilliant+Company+Is+Disrupting+a+%24200+Billion+Industry&utm_thumbnail=https%3A%2F%2Fs3.amazonaws.com%20a30-442c-9695-a0f9382e91b8.jpg)

(http://gundrymd.com/cmd.php?ad=822792&utm_source=taboola&utm_medium=referral)

How To Fix Dark Spots

Gundry MD

(http://gundrymd.com/cmd.php?ad=822792&utm_source=taboola&utm_medium=referral)

(https://plarium.com/play/en/vikings/005_village_anim_g?publisherID=53442276&placement=mediosyredes-programacionnet&plid=101491&pxl=taboola_fr)

Play this for 1 minute and see why everyone is addicted

Vikings: Free Online Game

(https://plarium.com/play/en/vikings/005_village_anim_g?publisherID=53442276&placement=mediosyredes-programacionnet&plid=101491&pxl=taboola_fr)

(https://autooverload.com/comical-photos-from-walmart/?utm_source=taboola&utm_medium=taboola&utm_campaign=tbshop6c)

Walmart Cameras Captured These Photos (10 Photos)

Auto Overload

(http://autooverload.com/comical-photos-from-walmart/?utm_source=taboola&utm_medium=taboola&utm_campaign=tbshop6c)

(http://tophatter.com?source=taboola&campaign=580941&ad_group=Desktop_Tophatther_ElectrnoicsShoppers&ad_content=51396848&ad_widget=mediosyredes-programacionnet)

Shoppers Are Getting Unbelievable Deals With This Little-Known Site

Tophatter

(http://tophatter.com?source=taboola&campaign=580941&ad_group=Desktop_Tophatther_ElectrnoicsShoppers&ad_content=51396848&ad_widget=mediosyredes-programacionnet)

(http://click.sleepgram.com/c906a5d2-b2af-4aff-a47e-abdf940d8cf9?SID=tabdesk9&utm_source=taboola&utm_medium=referral&utm_site=mediosyredes-programacionnet&utm_thumbnail=http%3A%2F%2Fcdn.taboola.com%2Fliibtrc%2Fstatic%2Fthumbnails%2Fb4f666d91c6c95977fdf3aef8b54ae53.png&utm_title=The+Best+Pillow+I+Have+Ever

The Best Pillow I Have Ever Slept On...

Sleepgram

(http://click.sleepgram.com/c906a5d2-b2af-4aff-a47e-abdf940d8cf9?SID=tabdesk9&utm_source=taboola&utm_medium=referral&utm_site=mediosyredes-programacionnet&utm_thumbnail=http%3A%2F%2Fcdn.taboola.com%2Fliibtrc%2Fstatic%2Fthumbnails%2Fb4f666d91c6c95977fdf3aef8b54ae53.png&utm_title=The+Best+Pillow+I+Have+Ever

(http://d0ehy.voluumtrk.com/7132ccc9-efb1-4cc9-ac2d-3440b114efc7?utm_source=taboola&utm_medium=referral)

Dr’s Shocked By US Dad’s Simple Glaucoma “Fix”

contactlensesplus.com

(http://d0ehy.voluumtrk.com/7132ccc9-efb1-4cc9-ac2d-3440b114efc7?utm_source=taboola&utm_medium=referral)

(http://thinninghairfix.com/split.php?utm_source=taboola&utm_medium=referral)

Want A Gorgeous Head Of Hair Again? Don’t Use This Popular Product

JuveTress

(http://thinninghairfix.com/split.php?utm_source=taboola&utm_medium=referral)

¡SÉ EL PRIMERO EN COMENTAR!

Conéctate (/login) o Regístrate (/registro) para dejar tu comentario.

Secciones

- Artículos (/articulos)
- Tutoriales y código fuente (/codigos)
- Foros (/foros)
- Eventos (/eventos)
- Empleo (/empleo)

Lenguajes Destacados

- PHP (/php)
- Java (/java)

[ASP \(/asp\)](#)
[Bases de datos \(/bases-de-datos\)](#)
[C \(/c\)](#)

Información

[Datos Legales \(/datos_legales\)](#)
[Política de privacidad \(/politica_de_privacidad\)](#)
[Publicidad \(/publicidad\)](#)

Contacto

[Contacte con nosotros \(/contacto\)](#)
[Publicidad \(/publicidad\)](#)



<https://www.facebook.com/programacionencastellano>



<https://twitter.com/noprog>

Diseño web y desarrollo web (<https://colorvivo.com>). Un proyecto de los hermanos Carrero (<https://carrero.es>).

Alojado en cloud privado StackScale (<https://www.stackscale.es/>)

Más internet: Password (<https://password.es>) | Favicon (<http://genfavicon.com>) | Crear un Avatar (<http://face.co>)