# Cómo uso CSS Grid y Flexbox para crear un sitio web de una página

<https://medium.com/swlh/how-i-use-css-grid-and-flexbox-to-create-a-one-page-website-262aeea3592a>

Contenido

[Cómo uso CSS Grid y Flexbox para crear un sitio web de una página 1](#_Toc117242766)

[CSS Grid para diseño de página y Flexbox para elementos de interfaz de usuario 2](#_Toc117242767)

[Paso 1: planifica antes de codificar 2](#_Toc117242768)

[Determinar el número de filas 2](#_Toc117242769)

[Determinar el número de columnas 3](#_Toc117242770)

[Crea una guía 3](#_Toc117242771)

[Paso 2: crea el archivo HTML 4](#_Toc117242772)

[Paso 3: crea un archivo CSS 4](#_Toc117242773)

[Configuración 5](#_Toc117242774)

[Crea la cuadrícula 5](#_Toc117242775)

[Ranura de elementos en celdas de la cuadrícula 5](#_Toc117242776)

[Creando flex-container y flex-child 6](#_Toc117242777)

[Vista móvil 7](#_Toc117242778)

[Resumen 9](#_Toc117242779)

[Grid 9](#_Toc117242780)

[Flexbox 9](#_Toc117242781)

[Cómo uso CSS Grid y Flexbox para crear un sitio web de una página - Parte 2 9](#_Toc117242782)

[Configuración 10](#_Toc117242783)

[Paso 1. Planificación antes de codificar 11](#_Toc117242784)

[Paso 2. Modificar HTML y CSS 11](#_Toc117242785)

[HTML 12](#_Toc117242786)

[CSS 12](#_Toc117242787)

[Revisar 17](#_Toc117242788)

[Vista móvil 19](#_Toc117242789)

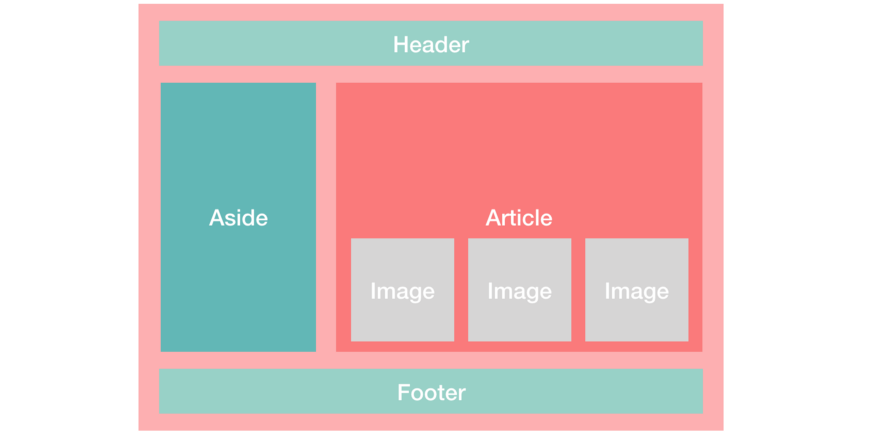
[Segunda revisión 20](#_Toc117242790)

[Conclusión 21](#_Toc117242791)

[Recursos 22](#_Toc117242792)

## CSS Grid para diseño de página y Flexbox para elementos de interfaz de usuario

Si bien podemos usar [CSS Grid](https://medium.com/swlh/how-i-use-css-grid-to-create-a-one-page-website-ffc97668d33a) y [Flexbox](https://medium.com/swlh/how-i-use-flexbox-to-create-a-one-page-website-81db78e61738) para un diseño de página web, CSS Grid tiende a usarse para diseños de página, mientras que Flexbox tiende a usarse para alinear elementos de la interfaz de usuario. Creemos una página web para ver cómo funciona esto en código real. Usaré un diseño común como se muestra a continuación:



# Paso 1: planifica antes de codificar

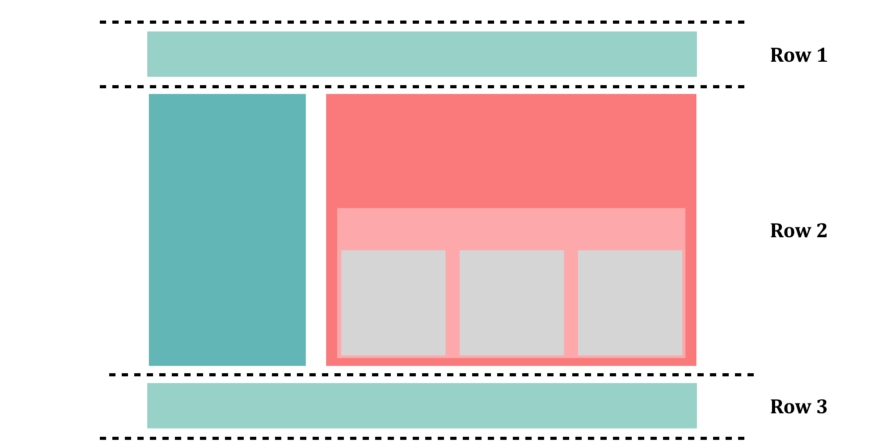
El truco para codificar diseños es planificarlos de antemano.

Primero identificaré los elementos padre e hijo según el método de diseño. A continuación, se muestran mis trabajos basados en el diseño.

## Determinar el número de filas

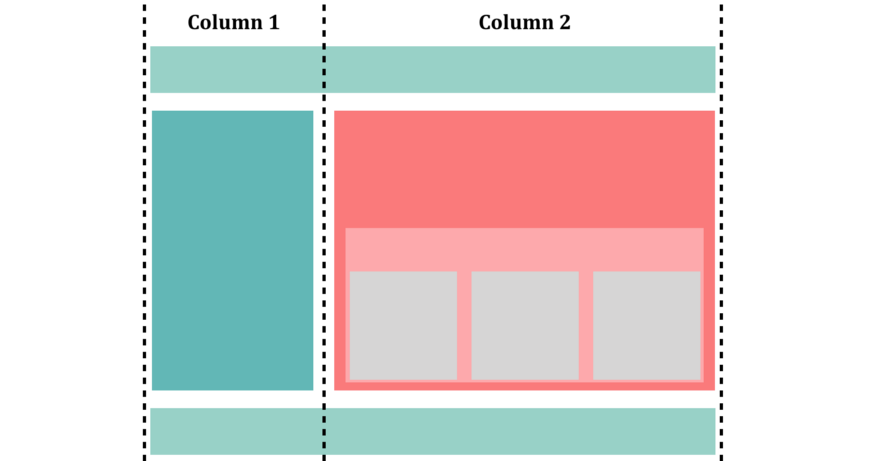
Ordenaré los elementos de la página en grupos de forma horizontal. Los elementos se representan como cuadros y los que están a lo largo de la misma línea (imaginaria) se consideran en la misma fila, por ejemplo, asidey article. Si comenzaran o terminaran en una línea diferente, los consideraría en filas diferentes.

Tenga en cuenta que los tres elementos y su contenedor interno article se ignoran aquí, ya que no forman parte de la cuadrícula.



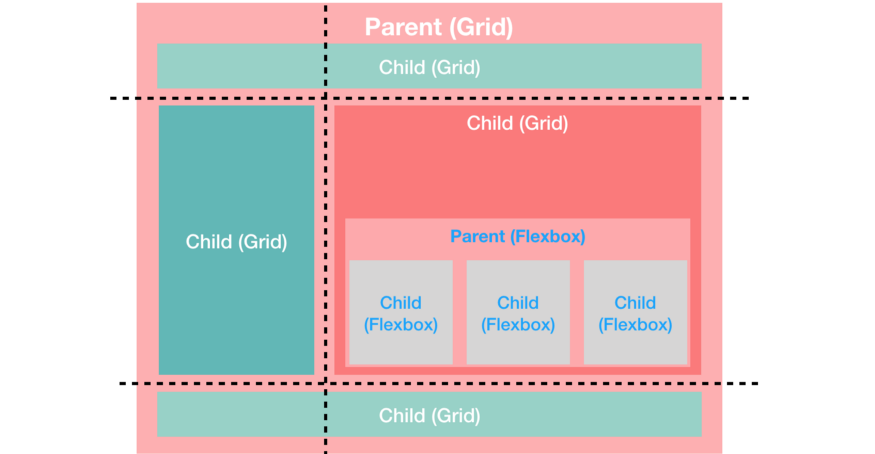
## Determinar el número de columnas

Ordenaré los elementos en grupos verticalmente. Los elementos a lo largo de la misma línea (imaginaria) se considerarán en la misma columna.



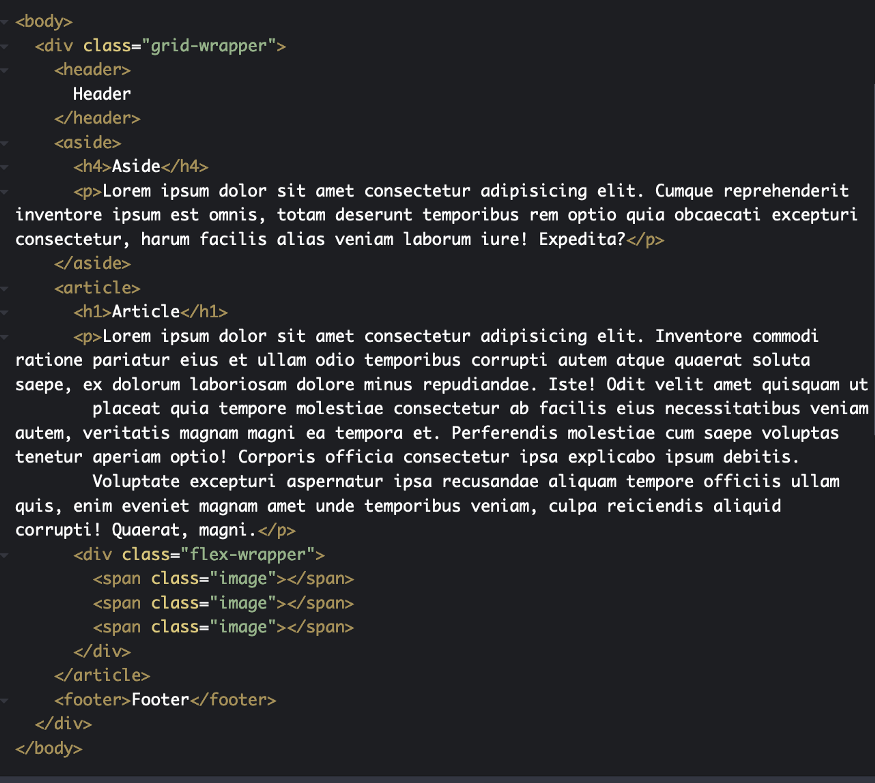
## Crea una guía

Después de determinar el número de columnas y filas, ahora tengo una guía que puedo seguir al codificar. En ella, también marcaré las líneas de la cuadrícula y anotaré los elementos padre e hijo.



# Paso 2: crea el archivo HTML

Para un diseño simple, escribiré todo el html antes de trabajar en CSS.



# Paso 3: crea un archivo CSS

Después de lo cual, crearé el archivo CSS y lo vincularé a mi archivo html.

## Configuración

Elimino el margen predeterminado y el relleno de los elementos htmly body , y añado una altura de 100%.

## Crea la cuadrícula

*CSS Grid puede resultar intimidante al principio, ya que hay varias formas de crear una grilla. Por lo tanto, es menos confuso ceñirse a una sintaxis en particular cuando se aprende.*

Declaro display: grid en el div con la clase de contenedor de cuadrícula. Esto transforma el div en un contenedor de rejilla. Sus hijos directos ahora son elementos de cuadrícula. Los elementos que están dentro de sus hijos directos no se verán afectados.

A continuación, especifico el ancho de las filas de la cuadrícula y la altura de las columnas de la cuadrícula.

* Filas: Dado que headery footer contienen solo el contenido, y quiero que asidey article se expandan para cubrir el resto de la página mientras empujo el footer hasta el final de la página, agrego grid-template-rows: min-content auto min-content.
* Columnas: como quiero que aside ocupe 1/4 del ancho de la página y article ocupe el resto, utilizo grid-template-columns: 1fr 3fr. La unidad fr representa una fracción del espacio disponible en el contenedor de cuadrícula. En cuanto a headery footer, no tengo que pensar en ellos, ya que se extienden por las dos columnas, ocupando un ancho de 4fr.

## **Ranura de elementos en celdas de la cuadrícula**

Especifico el tamaño y la posición de cada elemento de la cuadrícula usando las propiedades grid-columny grid-row.

El encabezado y el pie de página cubrirán el área de 2 columnas. La declaración grid-column: 1/3 es:

inicio de columna de cuadrícula: 1;

fin de columna de cuadrícula: 3;

Simplemente significa que comienza en la primera línea vertical y termina en la tercera línea vertical.

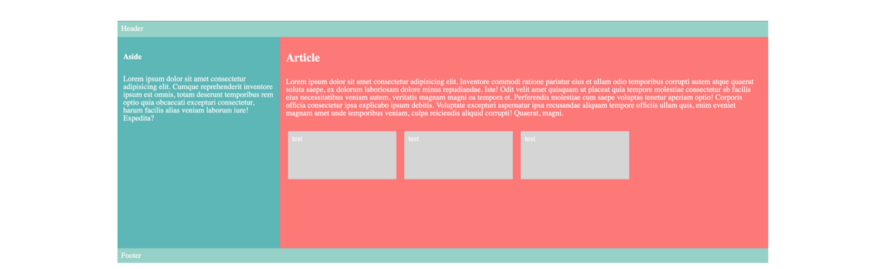
Asimismo, la declaración de aside grid-row: 2/3 significa que comienza en la segunda línea horizontal y termina en la tercera línea horizontal.

inicio de fila de cuadrícula: 2;

fin de fila de cuadrícula: 3;



A continuación se muestra el resultado final del uso de CSS Grid:



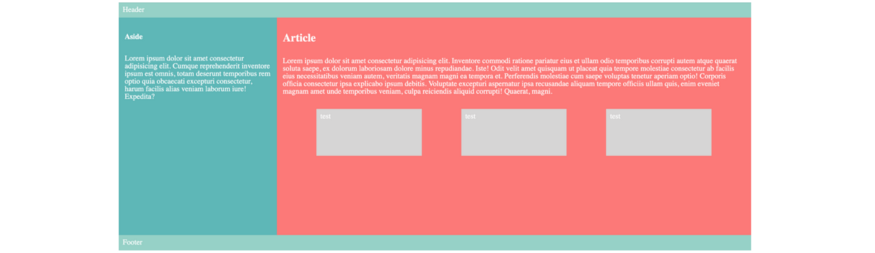
## Creando flex-container y flex-child

A continuación, trabajo en espaciar los elementos de la imagen, que se muestran como cuadros grises.

El divcon la clase de envoltura flexible se envuelve alrededor de mis imágenes. Para transformarlo en un contenedor flexible, agrego display: flex. No es necesario cambiar la dirección de los elementos flexibles, ya que se distribuirán de izquierda a derecha de forma predeterminada.

Para mejorar la capacidad de respuesta de la página, quiero que el ancho de mis elementos flexibles se ajuste de acuerdo con el ancho de la página. Esto se puede lograr configurando flex-grow: 1

para las tres imágenes.



## Vista móvil

Por último, trabajo en la vista móvil.

En dispositivos móviles, tendemos a colocar las cosas hacia abajo en lugar de hacia los lados, ya que el tamaño de la pantalla es limitado. Para lograr esto, puedo hacerlo en un paso:

He agreguado una consulta de medios para aplicar display: block en mi contenedor de cuadrícula cuando el tamaño de la pantalla es pequeño. Esa es una forma bastante ordenada, ¿verdad?



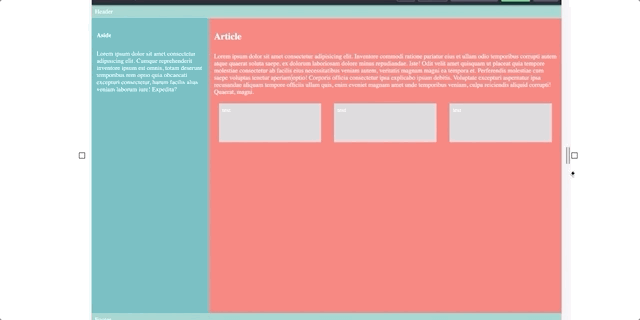
Los elementos flexibles del interior de article necesitan más trabajo, ya que todavía están dispuestos en una fila. Para cambiar de fila a columna, agrego flex-direction: column al contenedor flexible.

Para mejorar el estilo, también agrego un margin-bottom: 20px a cada elemento flexible para crear un espacio entre las imágenes.

¡Con eso, he completado mi página web!



Como siempre, el último paso será verificar si hay problemas de alineación en diferentes puntos de interrupción cambiando el ancho del navegador.



# Resumen

En resumen, las propiedades de diseño que hemos utilizado en este artículo son:

## Grid

* display: grid
* grid-template-columnsy grid-template-rows (tenga en cuenta la forma plural)
* grid-column y grid-row (nota la forma singular)

## Flexbox

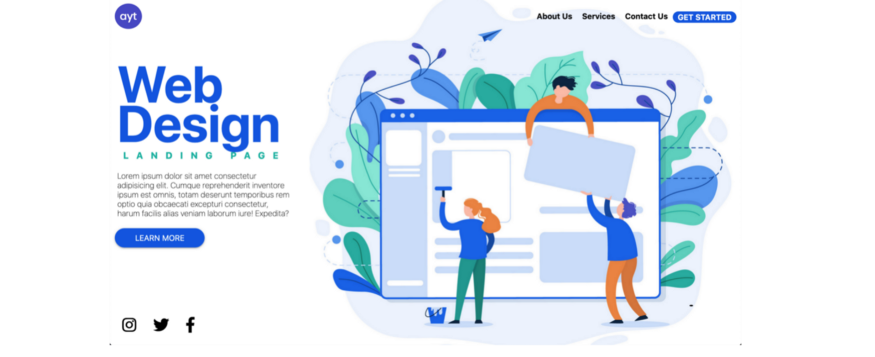
* display: flex
* flex-direction
* flex-grow

# Cómo uso CSS Grid y Flexbox para crear un sitio web de una página - Parte 2

https://betterprogramming.pub/how-i-use-css-grid-and-flexbox-to-create-a-one-page-website-part-2-e575edb422fe

Crear un sitio web desde cero puede llevar mucho tiempo. Sin embargo, dado que muchos sitios web tienen un aspecto estándar, podemos reducir el tiempo de desarrollo creando nuestras propias plantillas que podemos diseñar y modificar cuando sea necesario.

En este artículo, modificaré una de mis plantillas para crear un sitio web a partir de una página. A continuación se muestra la captura de pantalla del sitio web que se quiere crear:



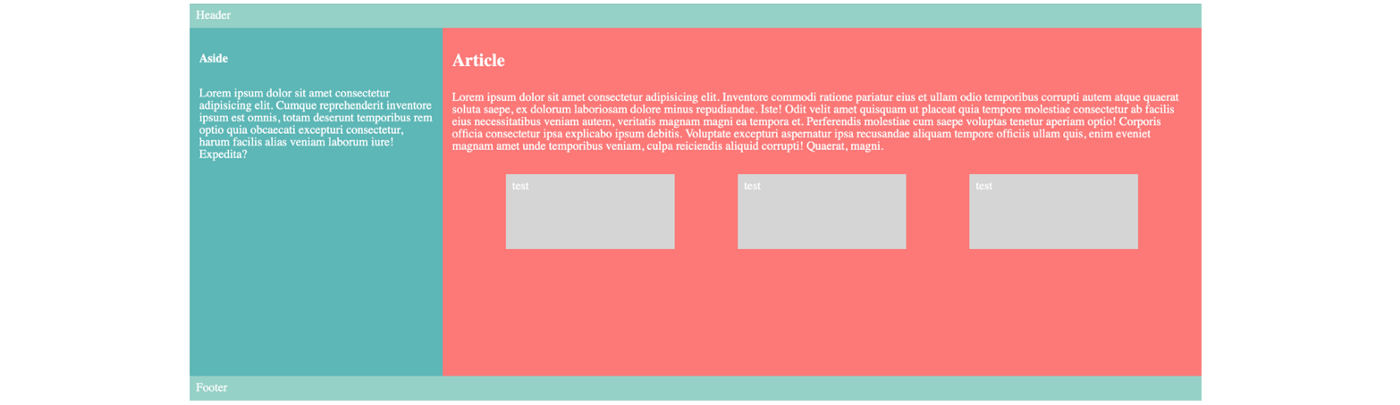
El diseño se basa libremente en lo siguiente:



Diseñado por [Freepik](http://www.freepik.com/)

# Configuración

La plantilla que estoy usando es de mi [último artículo sobre cómo uso CSS Grid y Flexbox para crear un sitio web de una página](https://medium.com/swlh/how-i-use-css-grid-and-flexbox-to-create-a-one-page-website-262aeea3592a).



# Paso 1. Planificación antes de codificar

Necesito modificar la plantilla, ya que no coincide completamente con el diseño del sitio web que tengo en mente. A continuación se muestran los cambios que planeo realizar:

header

* Agregar un componente de navegación y enlaces de navegación en la esquina derecha.
* Cambiar la navegación a un contenedor flexible para que los enlaces de navegación se muestren en una fila en una pantalla grande y en una columna en una pantalla pequeña.
* Agregar un logotipo en la esquina superior izquierda.

aside

* Conservar asidepara mi contenido destacado.
* Agregar un subtítulo con texto en mayúsculas.
* Cambiar a un lado a un contenedor flexible para que pueda usar flex-grow para llenar el espacio.

article

* Cambiar article a div para mantener mi imagen destacada.
* Agregar una imagen usando la propiedad background-image, ya que me gustaría que cambiara de tamaño según el tamaño de la pantalla.
* Elimina todo el contenido.

footer

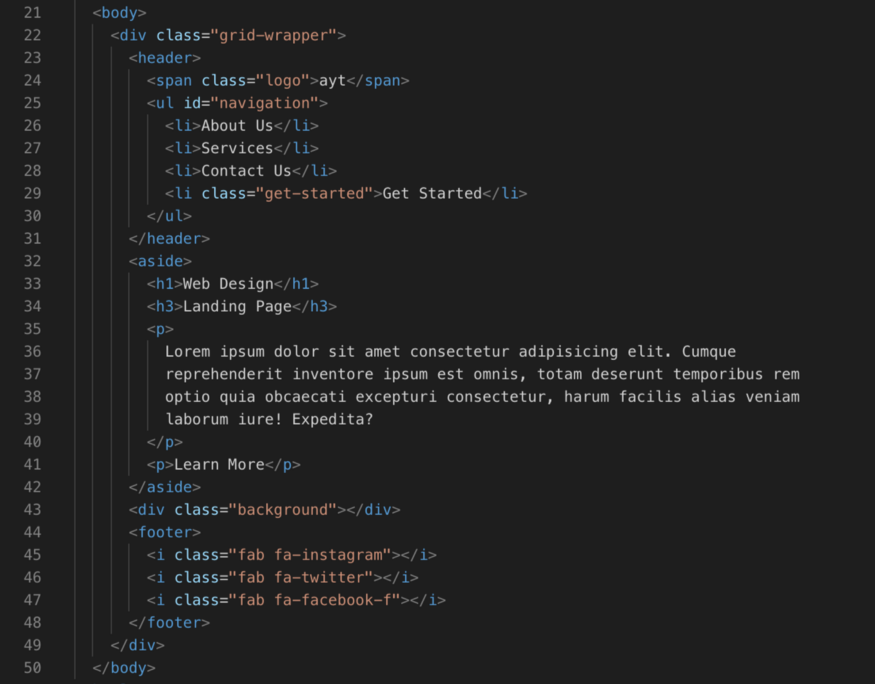
* Agregar tres iconos y alinearlos en el lado izquierdo.

# Paso 2. Modificar HTML y CSS

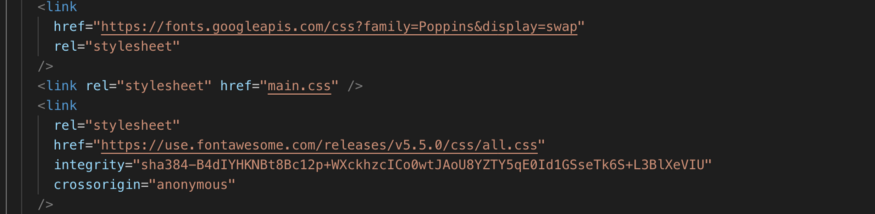
Con los cambios anteriores en mente, veamos cómo modifico mi HTML y CSS.

## HTML

La captura de pantalla a continuación muestra el HTML actualizado con un logotipo, enlaces de navegación, publicaciones destacadas e iconos de redes sociales.



Se agregan dos etiquetas link a [Google Font](https://fonts.google.com/) y [Font Awesome](https://fontawesome.com/?from=io) para que pueda usar la fuente seleccionada de Google Font y los iconos de Font Awesome.



## CSS

A continuación, empiezo modificando los estilos de arriba a abajo.



body

En lugar de usar la fuente predeterminada en el navegador, prefiero el aspecto de las fuentes del sistema y puedo usarla declarando:

font-family: -apple-system, BlinkMacSystemFont, “Segoe UI”, Roboto, Oxygen, Ubuntu, Cantarell, “Open Sans”, “Helvetica Neue”, sans-serif;

En el selector de cuerpo. Como las propiedades de la fuente son propiedades heredadas, esto aplicará la fuente especificada en toda la página web a menos que la anule.

header

* Agregar un logo en la esquina superior izquierda.

Después de agregar mi logotipo y el menú de navegación dentro de la etiqueta del encabezado en HTML, para crear un diseño donde quiero que estos dos elementos se coloquen en extremos opuestos, transformo mi encabezado en un contenedor de cuadrícula.

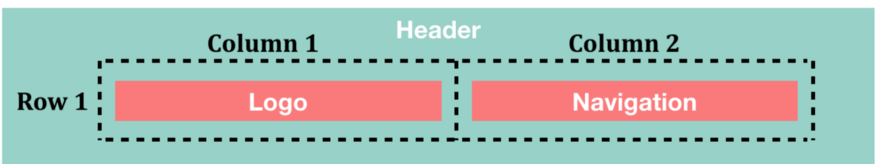
header {

display: grid;

grid-template-columns: 1fr 1fr;

}

Los elementos dentro del encabezado, el logotipo y los enlaces de navegación ahora son elementos de cuadrícula.



* Cambiar la navegación a un contenedor flexible para que los enlaces de navegación se muestren en una fila en una pantalla grande y en una columna en una pantalla pequeña.

A continuación, cambio el ul con el ID de navegación a un contenedor flexible declarando display: flex en él.

* Agregar contenedor de navegación y enlaces de navegación en la esquina derecha.

Agrego justify-content: flex-end para empujar los enlaces de navegación al final de la fila.

* Estilismo

Elimino el estilo de lista en los enlaces de navegación con la declaración list-style-type:none.

aside

* Agrega un subtítulo con texto en mayúsculas.

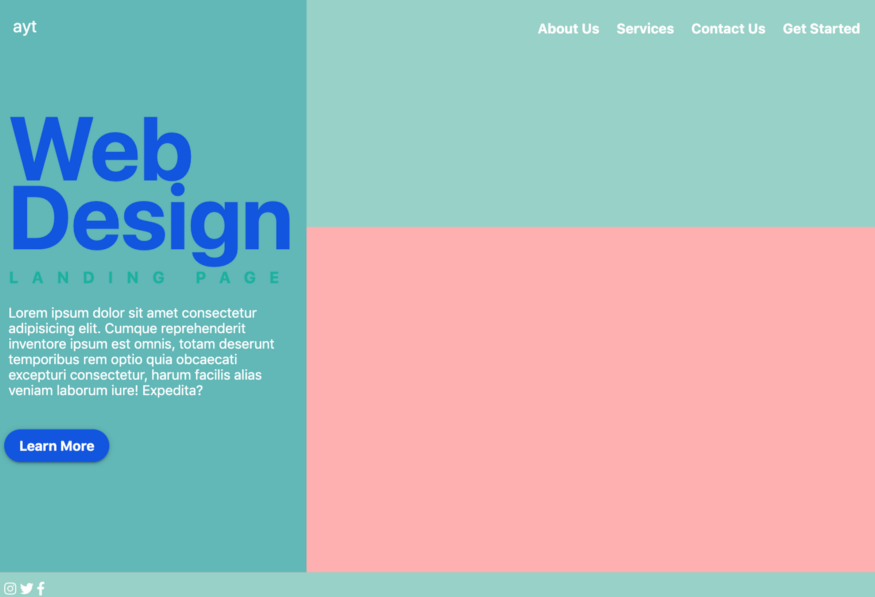
Para el subtítulo, declaro text-transform: uppercase para cambiar el texto a mayúsculas. Si bien puedo escribir directamente el texto en mayúsculas en el archivo HTML, esa no es la forma correcta, ya que diseñamos nuestro contenido con CSS.

* Estilismo

Aumento el tamaño de fuente predeterminado del título y subtítulo para que se destaquen en la página.

En cuanto al color, reutilizo los colores más oscuros de la imagen destacada para los colores de fuente para mantener la paleta de colores consistente.

Se ha extendido la posición de aside para superponerlo con el encabezado cambiando la propiedad grid-column. Esto es para subir el contenido ya que el diseño parecería desequilibrado de otra manera.



.background

* Cambiamos articlea div para mantener mi imagen destacada.

Reemplazo el selector article con el selector .background, ya que se ha cambiado la etiqueta HTML article por div.

* Agreguamos una imagen usando la propiedad  background-image, ya que queremos que cambie de tamaño según el tamaño de la pantalla.

Se ha seleccionado una imagen de fondo para el sitio.



Diseñado por [Freepik](http://www.freepik.com/)

Se ha agregado la imagen de arriba como valor de la propiedad background-image.

.background {

background-image: url ("/ background2.png");

tamaño de fondo: cubierta;

posición de fondo: centro izquierdo;

repetición de fondo: no repetición;

}

El valor de cover para background-size escala la imagen sin estirarla. Luego juego con el posicionamiento de la imagen antes de decidir seguir background-position: left center.

* Estilismo

Cambio el  .background-position del nuevo selector en la cuadrícula para que la imagen se superponga con headery footer.

footer

* Agregar tres iconos y alinearlos en el lado izquierdo.

De forma predeterminada, después de agregar los iconos en HTML, se organizan en el lado izquierdo de la página.

Solo necesito espaciarlos aumentando los márgenes entre los iconos.

.fab {

tamaño de fuente: 1.5rem;

margen: 10px;

}

# Revisar

Aquí hay una vista previa del sitio, basada en los cambios hasta ahora. Se tiene que hacer más trabajo antes de que pueda dar por terminado el día.



Revisión del sitio web

Propiedades colory background-color. Eliminamos los colores de fondo en la plantilla original y cambiamos el color de la fuente a negro declarando color: black en el selector de cuerpo.

Nuevamente, utilizo propiedades heredadas donde el color de la fuente será heredado por los hijos del elemento del cuerpo (y nietos y bisnietos… .etc.).



Un enlace con el estilo de un botón

En el paso anterior, recuerde que cambiamos la propiedad color a negro. Esto cambió el color de mi botón falso. Por lo tanto, tengo que agregar color: white para volver a poner este texto blanco.

Agrego estilos similares al enlace de navegación para Comenzar.

También agrego una altura y un ancho a mi logotipo y lo convierto en un círculo usando la propiedad border-radius.

Siempre que la altura y el ancho sean de la misma longitud y el radio del borde tenga un valor del 50% o más, el elemento se transformará de un cuadrado a un círculo.



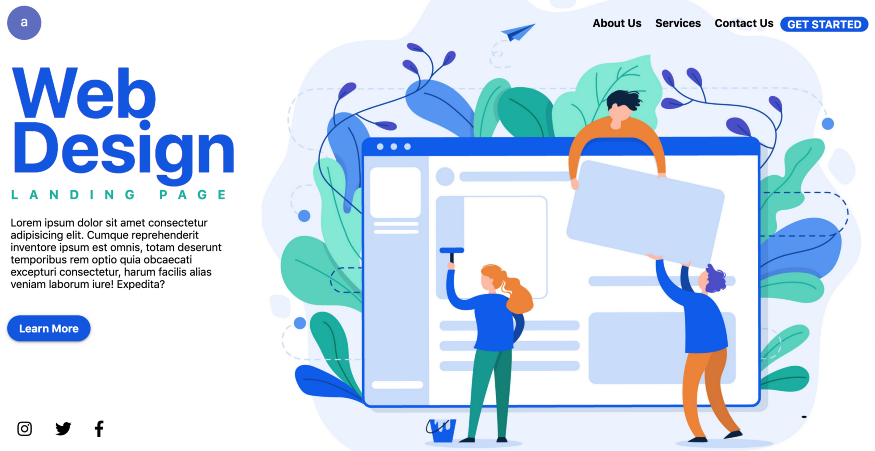
Mi logo

Propiedad z-index

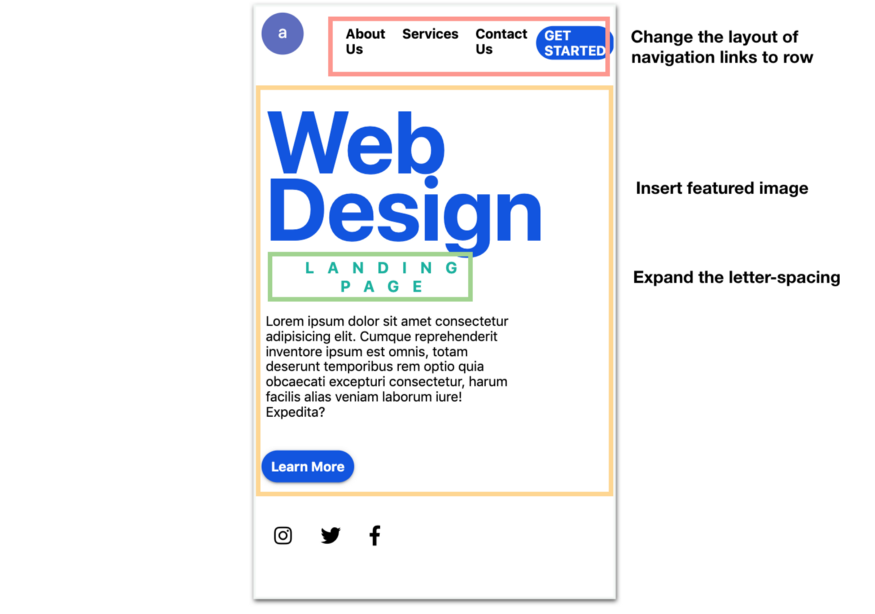
En la captura de pantalla que muestra una revisión del sitio web, la imagen destacada se superpone a los enlaces de navegación. Necesito cambiar el valor de la propiedad z-index del encabezado, que al mismo tiempo establece el índice z de sus hijos, es decir, los enlaces de navegación.

Como los elementos superpuestos con un índice z más grande cubren a los que tienen uno más pequeño, le doy un valor de 1 para que sea más alto que el valor de la imagen.

Este es el producto final de mi vista de escritorio. Hurra…



Me quedo con la vista móvil. Así es como se ve ahora.



# Vista móvil

¿Notas que la imagen destacada desaparece en la vista móvil?

Esto sucede porque eliminé la cuadrícula cuando la página se muestra en una pantalla pequeña, y el div que contiene la imagen ahora tiene una altura de 0 ya que no declaré una altura ni puse ningún contenido.

En lugar de agregarle algo de altura, prefiero rediseñar la vista móvil.

Agrego la imagen usando la propiedad background-image del selector .grid-wrapper y cambio el diseño de navegación de una fila a otra.

Observe que utilicé otra imagen en la que no eliminé los bordes. La imagen anterior no se puede usar debido a la propiedad background-blend-mode que usaré.

.grid-wrapper {

    background-image: url("./background.jpg");

    background-size: cover;

    background-position: center;

    background-color: lightgray;

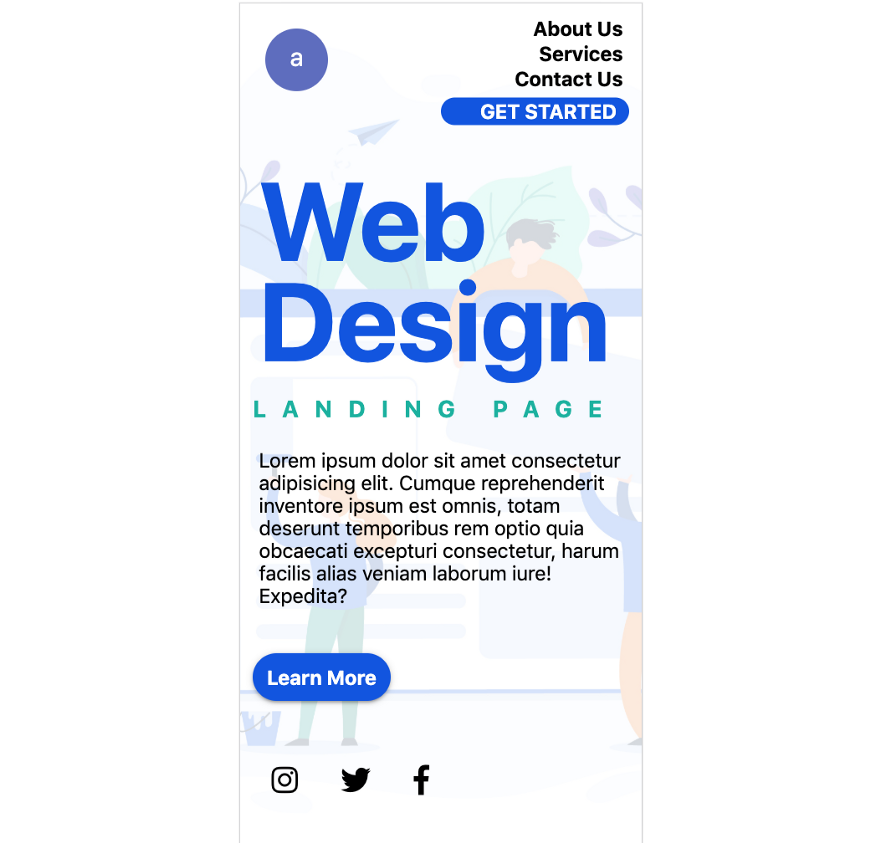
    background-blend-mode: screen;

  }

Ahora, el texto no se puede leer fácilmente contra la imagen debido al choque de colores. Por lo tanto, la salida más fácil es difuminar los colores de la imagen de fondo. No puedo usar la propiedad opacity ya que eso también desvanece el texto. Por lo tanto, tengo que usar otro método que acabo de aprender:

Agrego un color de fondo más claro a la imagen de fondo y mezcle ambos aplicando la declaración background-blend-mode: screen.

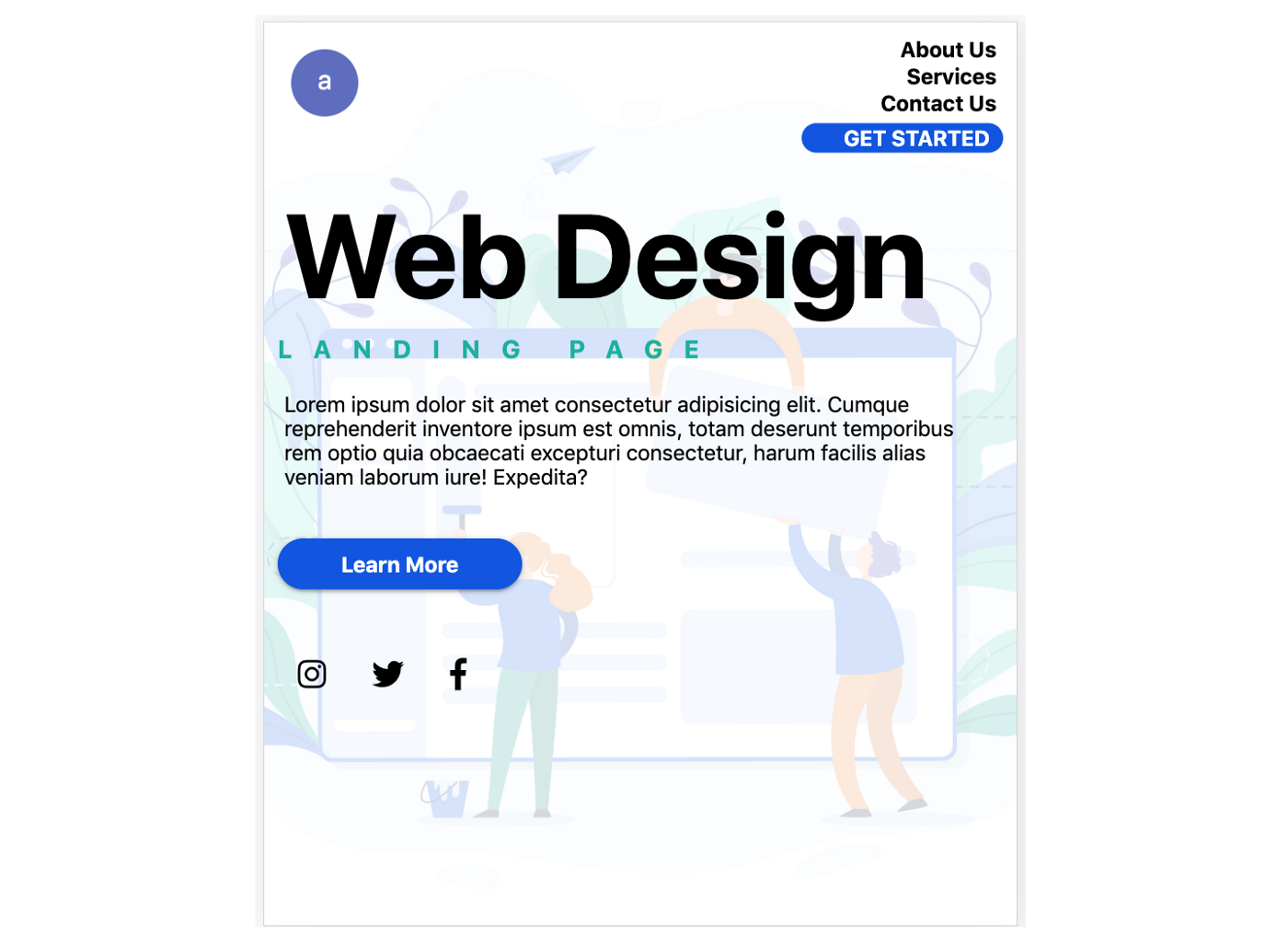
Aún en el texto, reduzco el espacio entre letras del subtítulo “Página de destino” para que se pueda mostrar en una sola línea.



# Segunda revisión

A continuación, [verifico](https://developers.google.com/web/tools/chrome-devtools) otros puntos de interrupción al verificar los diferentes tamaños de pantalla usando Chrome [DevTools](https://developers.google.com/web/tools/chrome-devtools).

Mi pie de página ya no está fijo en la parte inferior de la página después de que elimino la cuadrícula en la vista móvil. Se ve bien pero soy un poco quisquilloso.



Para resolver este problema, cambio .grid-wrapper a un contenedor flexible y agrego un crecimiento flexible aside para empujar el pie de página hacia la parte inferior.

.grid-wrapper {

display: flex;

flex-direction: column;

}

aside {

flex-grow: 1;

}

Última comprobación ... Sí ... Si desea ver el sitio real, puede visitar [mi sitio web](http://aliceyt.com/medium_articles/gridflex-demo/) .

# Conclusión

Crear un sitio web es mucho más fácil que escribir sobre cómo crearlo. Sin embargo, este ha sido un buen ejercicio ya que puedo escribir exactamente lo que quiero lograr y cómo hacerlo. Espero que este artículo sea útil para alguien.

Si desea utilizar este diseño de sitio web, puede obtenerlo [de mi repositorio de GitHub](https://github.com/tiffam/medium_articles/tree/master/gridflex-demo). Reordene las reglas de estilos en el archivo CSS como las ordené según el flujo de este artículo.

De lo contrario, puede divertirse creando su propio sitio. Obtendrá una gran sensación de logro una vez que lo complete, independientemente del enfoque que adopte.

# Recursos

* [Mi repositorio de Github con código para este artículo](https://github.com/tiffam/medium_articles/tree/master/gridflex-demo)

* [Cómo uso CSS Grid y Flexbox para crear un sitio web de una página (Parte 1)](https://medium.com/swlh/how-i-use-css-grid-and-flexbox-to-create-a-one-page-website-262aeea3592a)