



GUÍA PASO POR PASO

CSS

44 minutos

Construir su propia página web
desde cero

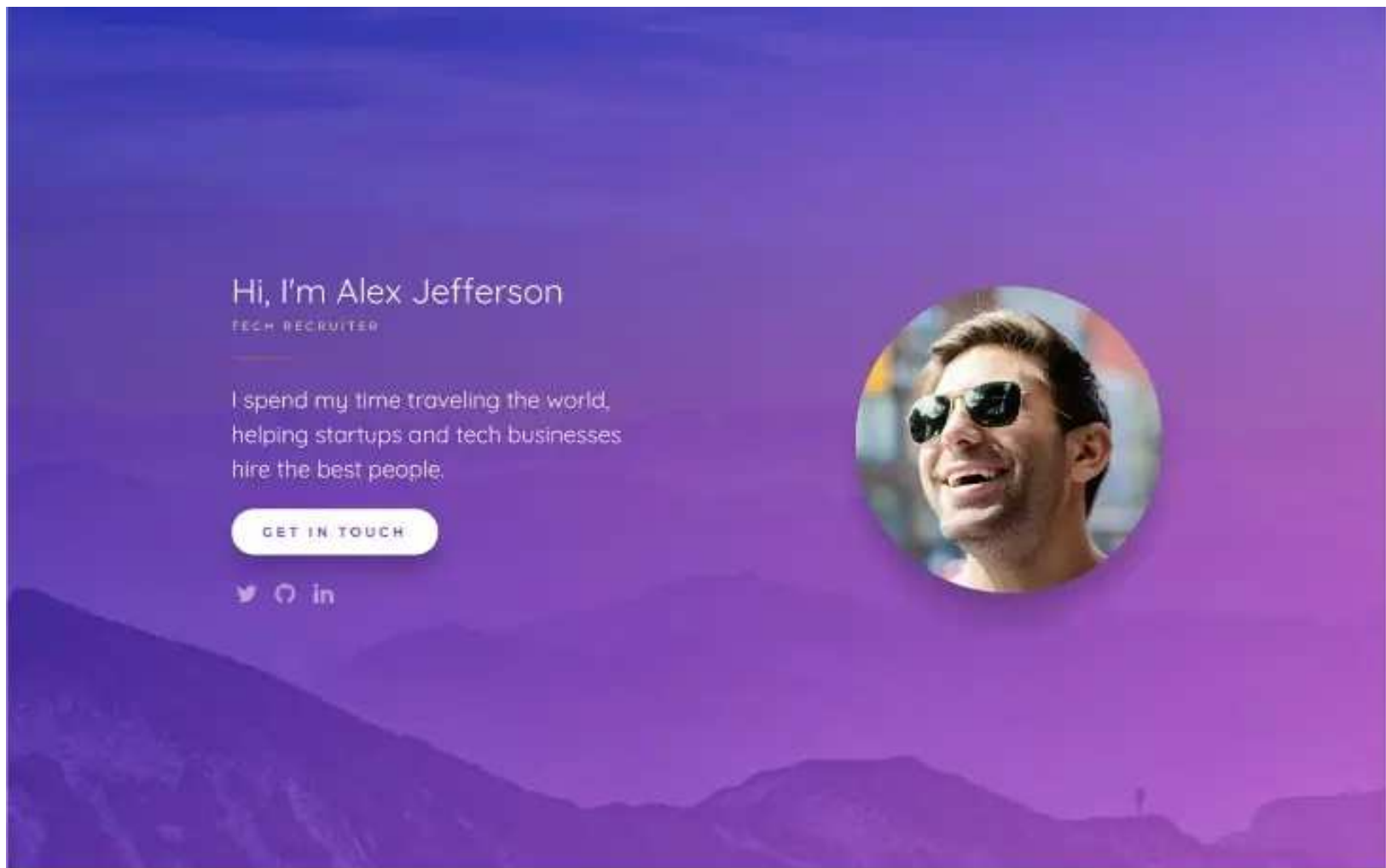


Escrito por

Jeremy Thomas

1 Introducción

Este libro es una **paso a paso** guía que le enseñará cómo construir esta página web desde cero:



Es la página web de un reclutador técnico ficticio llamado Alex Jefferson. Todas las imágenes proceden de [Unsplash](#) . A medida que siga esta guía, no dude en cambiar los estilos y el contenido de la forma que desee.

Para quién es este libro

Cualquiera puede leer este libro! No se requieren conocimientos de desarrollo web o programación previa, ya que voy a decirle lo que debe hacer, línea por línea.

Lo que va a construir

Esta página web va a utilizar los siguientes archivos:

- 1 HTML5 archivos
- CSS 3 archivo
- 4 imágenes (incluido) 1
- archivo JavaScript

También hará uso de dos servicios de terceros: [fuente impresionante](#) y [Fuentes de google](#) .

Lo que vas a aprender

A través de esta guía, te voy a enseñar cómo:

- Configurar una válida **HTML5** documento de escritura **semántico** Insertar
- marcado sensible **Retina-amigable** imágenes Aprenda cómo **estructura**
- CSS Diseño correctamente un 100% **sensible** Comprender cómo la
- página **preguntas de los medios** Uso de trabajo CSS **Flexbox** para
- diseñar los componentes del estilo de su texto con **tipografía** Añadir
- propiedades retroalimentación visual con CSS **transiciones**
-
-
-
- Traiga su página a la vida con CSS **animaciones**
- Hacer uso de **Fuentes de google**
- Incluir y estilo **fuentes impresionantes** iconos

Lo que hay que empezar

Sólo es necesario instalar 2 programas:

- un decente **editor de texto** o IDE con resaltado de sintaxis. yo recomiendo [Texto sublime](#) pero se puede usar Notepad ++, Vim, Emacs, IntelliJ, Átomo un moderno ... **navegador web**. yo suelo [Google Chrome](#) pero se puede utilizar Firefox, Safari, Opera o Edge.
-

También es necesario un par de archivos que he proporcionado:

- 1 archivo CSS: **minireset.min.css**
- 7 imágenes:
 - alex.jpg
 - alex@2x.jpg
 - alex@3x.jpg
 - austria.jpg 1x.png
 - 2x.png 3x.png
 -
 -

Tenerlos disponibles y listos para ser utilizados.

2 Escribir el contenido HTML5

Abra su editor de texto, cree un nuevo archivo y pega este fragmento de código:

```
<!DOCTYPE html>
< html > < cabeza > < meta charset = "UTF-8" > < meta http-equiv = "X-UA-Compatible" contenido = "IE = borde" > < meta name = "Ventana"
    contenido = "Width = dispositivo de ancho, inicial escala = 1" > < título > Alex Jefferson - Tech reclutador </ título >

</ cabeza > < cuerpo > < h1 > ¡Hola

    mundo! </ h1 >

</ cuerpo >
</ html >
```

Guarde este archivo como **index.html** , abrirlo en el navegador, y verá la página siguiente:



Asegúrese de tener la `.html` extensión de archivo. He aquí cómo hacerlo en [ventanas](#) y [Mac OS](#) .

Un documento HTML5 válido y sensible

Esta página es una **página HTML5 sensible válida** porque satisface los siguientes requisitos.

```
<!DOCTYPE html>
```

Esta línea indica al navegador que esta página web es una **documento HTML5** y debe interpretarse como tal.

```
< meta nombre = "Ventana" contenido = "Width = dispositivo de ancho, inicial escala = 1" >
```

Este es el **sensible** etiqueta meta. Se le dice al navegador para establecer la *contenido* anchura a la *ventana* uno, que le asegure, el contenido de adaptarse automáticamente. También le indica al navegador para establecer el nivel de zoom a 1 en un principio, al tiempo que permite al usuario hacer un zoom (especialmente en los teléfonos móviles).

Además, cada etiqueta HTML (`<html>` , `<Head>` , `<Body>` ...) se abre y cierra en el orden correcto. Solo el

`<Meta>` las etiquetas deben ser de cierre automático.

La adición de la CSS reset

Antes de diseñar su página, que desea iniciar con una **lienzo en blanco**. Debido a que cada navegador viene con sus propios estilos por defecto, que desea eliminar primero, de lo contrario estos estilos chocarán con la suya. Este proceso es el propósito de una **CSS reset**.

Mejores prácticas

Separar el contenido y el estilo

Usted quiere elegir una etiqueta HTML para su *significado semántico*, no su *apariencia*.

Si el cambio de etiqueta de un elemento en el código HTML cambia su apariencia, entonces estás esencialmente el estilo de su página web en el código HTML, que es lo que queremos evitar.

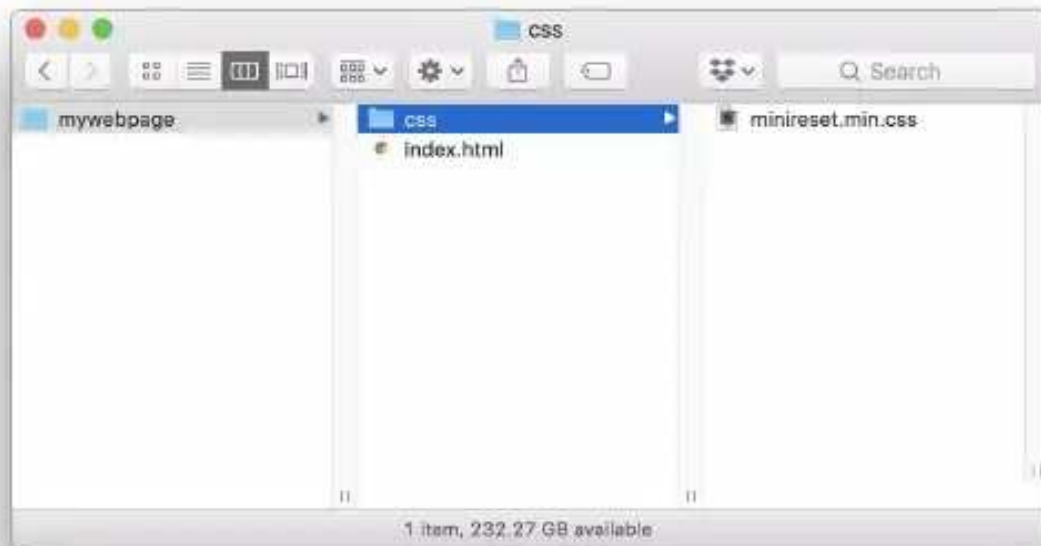
La mayoría de los restablecimientos CSS actuales son o bien complicado o anticuadas, así que creé una pequeña llamada [minireset.css](#) . Yo lo uso para todos mis proyectos, y está incluido por defecto en [Bulma](#) también.

Los únicos dos requisitos reales son los siguientes:

- eliminar los márgenes y rellenos de todo uso elementos de bloque `cuadro de`
- dimensionamiento: `frontera-box`

No dude en leer a través de la [código fuente](#) .

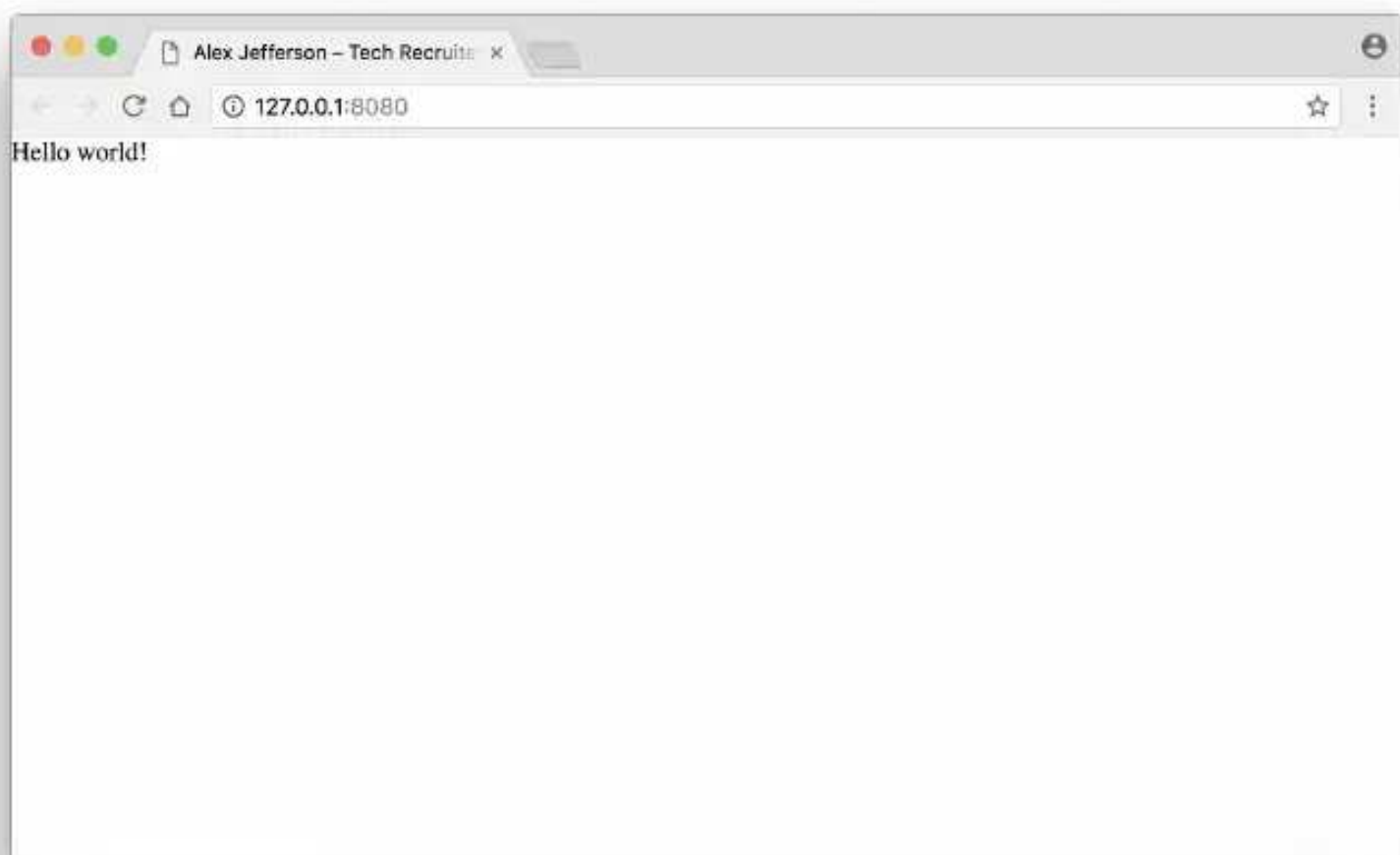
junto a su `index.html` archivo, crear una `/css` carpeta y mover el `minireset.min.css` presentar en ella.



Deje de enlazar nuestro archivo CSS *desde* el archivo HTML. Justo debajo de la `<title>`, añada esta línea:

```
< título > Alex Jefferson - Tech reclutador </ título > < link rel = "Hoja de estilo" tipo = "Text / css" href = "Css /  
minireset.min.css" >
```

El texto "**¡Hola Mundo!**" ahora está sin estilo.



Su propio CSS

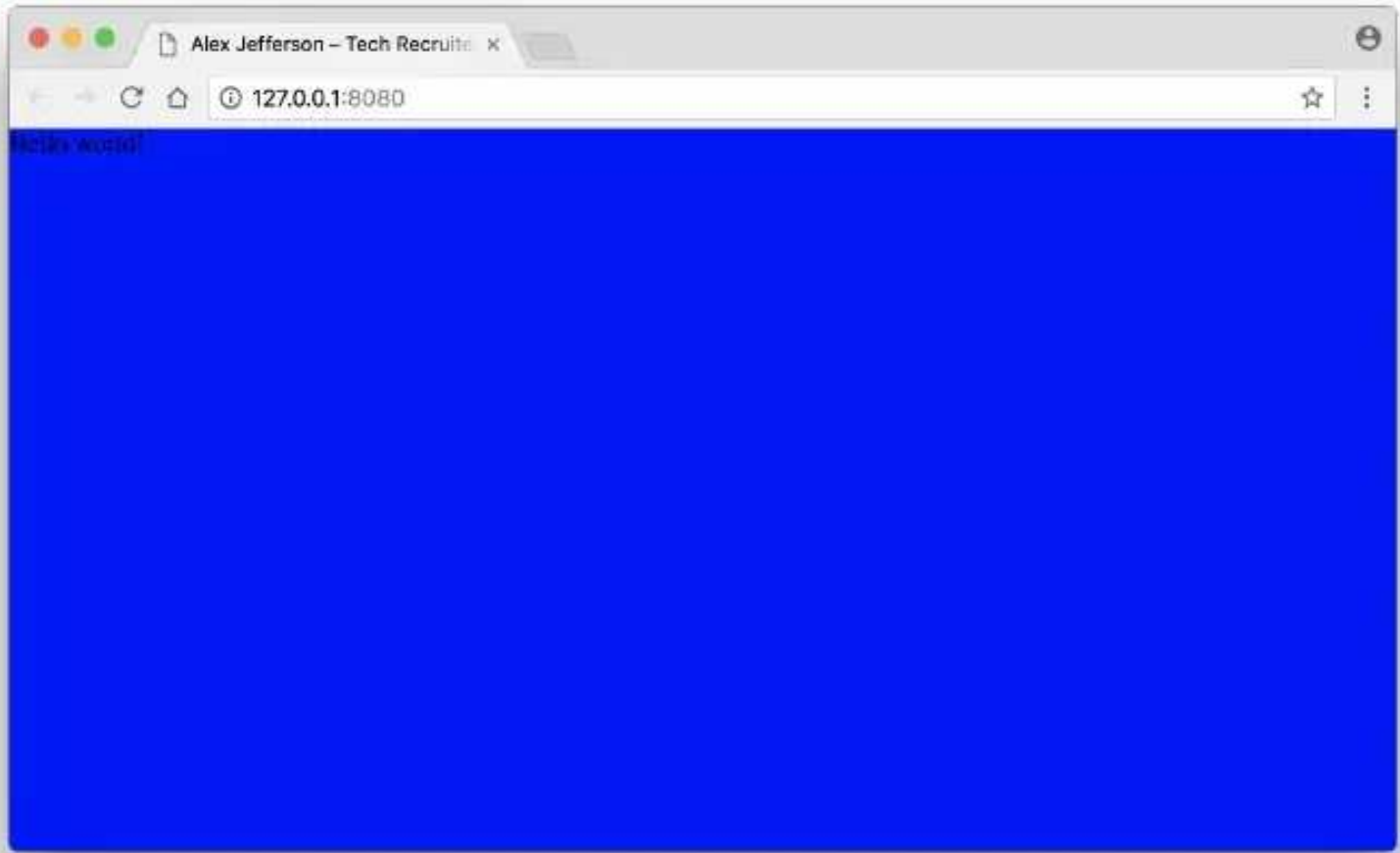
En el `/css` carpeta, cree un nuevo archivo vacío llamado `main.css`.

Ahora tiene que incluir esta CSS en su página. En `index.html` , justo debajo de la `minireset`, vincular su `main.css` archivo. Ahora tienes 2 archivos CSS:

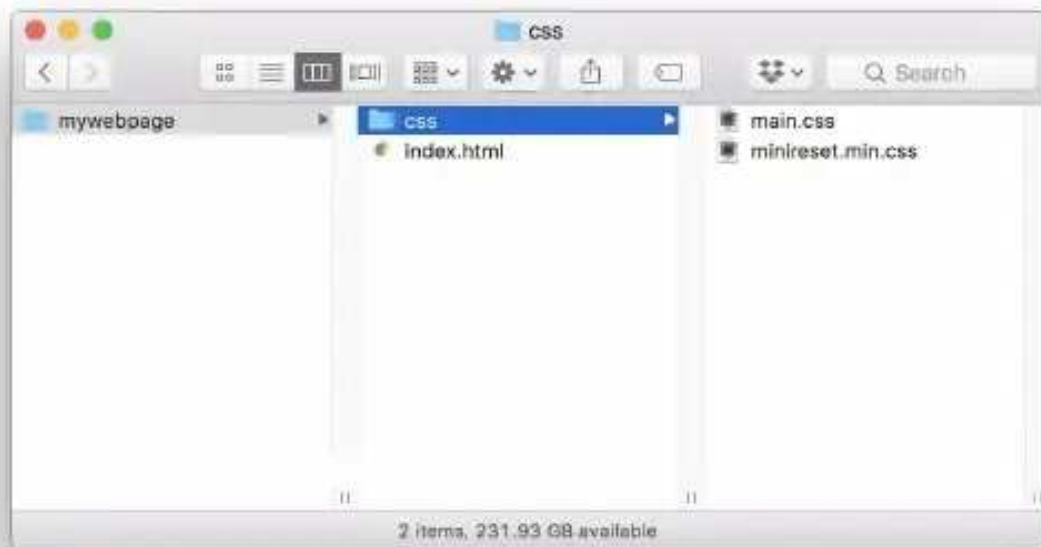
```
< link rel = "Hoja de estilo" tipo = "Text / css" href = "Css / minireset.min.css" > < link rel = "Hoja de estilo" tipo = "Text / css" href  
= "Css / main.css" >
```

Para ver si el CSS se incluyó correctamente, añadir esto a `main.css` :

```
html {  
  color de fondo : Blue; }
```



Si la pantalla no se vuelve azul, asegúrese de que su `index.html` archivo está al lado de su `/css` carpeta:

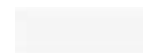


El CSS ahora trabaja para que pueda quitar el fondo azul de la CSS. Antes de escribir cualquier CSS, usted tiene que comenzar con el más importante de una página web: la **contenido**.

contenido de la escritura

El propósito de **diseño** es mejorar la presentación del contenido al que se aplica. Puede sonar obvio, pero al ser el elemento principal de un sitio web de contenido, no debe ser establecido como una ocurrencia tardía. contenido escrito constituye más del 90% de la Web.

Eliminar "¡Hola Mundo!" desde su página, y poner el código dentro de la **< body>** :




```
< div clase = "fondo de pantalla" > </ div > < div clase = "contenido" > < aparte clase = "lado" >
  < clase figura = "imagen" > < div clase = "Foto-sombra" > </ div > < img id = "PictureImage"
  clase = "Foto-imagen"

  src = "Images / alex.jpg"
  srcset = "1x imágenes / alex.jpg,
            images / alex@2x.jpg 2x, imágenes /
            alex@3x.jpg 3x"
  alt = "Retrato de Alex Jefferson"
  anchura = "320"
  altura = "320" >
</ figura >
</ aparte > < principal clase = "acerca de" > < h1 clase
= "nombre" > Hola, soy Alex Jefferson </ h1 >
< pag clase = "trabajo" > Tech reclutador </ pag
> < hora clase = "hora" > < div clase = "descripción"
> < pag >

  Me paso el tiempo viajando por el mundo, ayudando a nuevas
  empresas y empresas de alta tecnología contratar a los
  mejores. </ pag >

</ div > < div clase = "contacto" > < un clase = "botón" href = "mailto: youremail@example.com " >
  Póngase en contacto </ un >

</ div >
</ principal >
</ div >
```

Como se puede ver, el HTML es **ruidoso**: existe una gran cantidad de código por no mucho contenido en la pantalla. Pero todas estas etiquetas HTML están aquí para *semántico* razones, y todos estos nombres de clases CSS están aquí para (futuros) *estilo* propósitos.



estructura HTML

los **fondo de pantalla** se mostrará una imagen transparente en el fondo, que cubre toda la página.

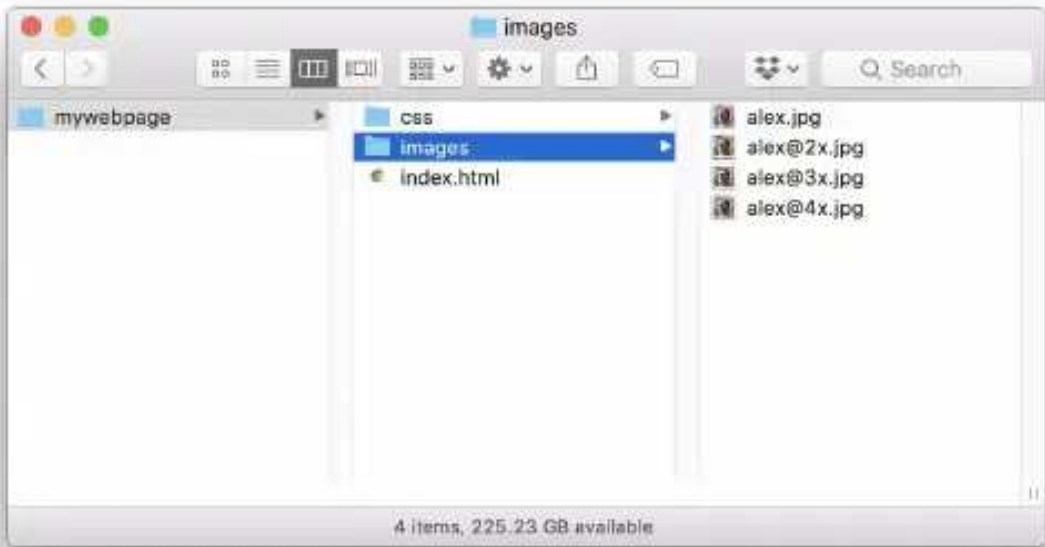
los **contenido** es el contenedor para la parte legible de la página web, y cubrirá toda la página también.

A la derecha, el **lado** mostrará el retrato imagen.

A la izquierda, el **acerca de** sección es un contenedor para todos **texto**. los **botón** actúa como un enlace para los visitantes a hacer clic, así que asegúrese de reemplazar **youremail@example.com** con su propia dirección de correo electrónico.

imágenes sensibles y Retina-amigables

Como se puede ver, ya que la `alex.jpg` la imagen no se encuentra, el navegador está mostrando el `alt` texto "Retrato de Alex Jefferson" como punto de retorno. Para solucionar este problema, crear una `/imagen` carpeta junto con el `/css` uno, y mover las 4 imágenes (`alex.jpg` , `alex@2x.jpg` , `alex@3x.jpg` y `alex@4x.jpg`) dentro:



Queremos mostrar esta imagen en un máximo de `320x320` píxeles. Pero las pantallas tienen diferentes **ratios de pixel**, especialmente los teléfonos móviles. Por ejemplo, el Sony Xperia S tiene una relación de pixel de `2` , Apple iPhone X `3` , y el Galaxy de Samsung S8 `4` .

Hemos podido demostrar la `1280x1280` versión de *todo el mundo* y cambiar su tamaño para `320x320` . Pero nos penalizaría a los usuarios que poseen un dispositivo con una proporción de píxeles de `1` porque terminarían la descarga de una imagen **10 veces** el tamaño que su dispositivo de realidad puede mostrar.

La solución es:

- mostrar al usuario una imagen de mejor calidad posible de su dispositivo puede soportar impedir que el
- navegador descargue de las otras imágenes

Esto es posible gracias a la `srcset` atributo que indica al navegador la lista de imagen *alternativas* disponible para cada densidad de píxeles, y lo dejó averiguar qué imagen se mostrará.

Si el navegador no soporta `srcset` , se utilizará `src` como punto de retorno.



Hi, I'm Alex Jefferson
Tech recruiter

I spend my time traveling the world, helping startups and tech businesses hire the best people.
[Get in touch](#)

Este es todo el contenido que usted necesita por ahora:

- imagen de un retrato de un
- nombre de un puesto de
- trabajo de un botón de
- contacto una descripción
-

Este contenido se extenderá, además, más adelante. Es el momento de añadir sus primeras líneas de **CSS**.

3 La creación de una base de CSS

Antes de modelar elementos individuales, es necesario establecer un mundial **estilo de base** para su página web.

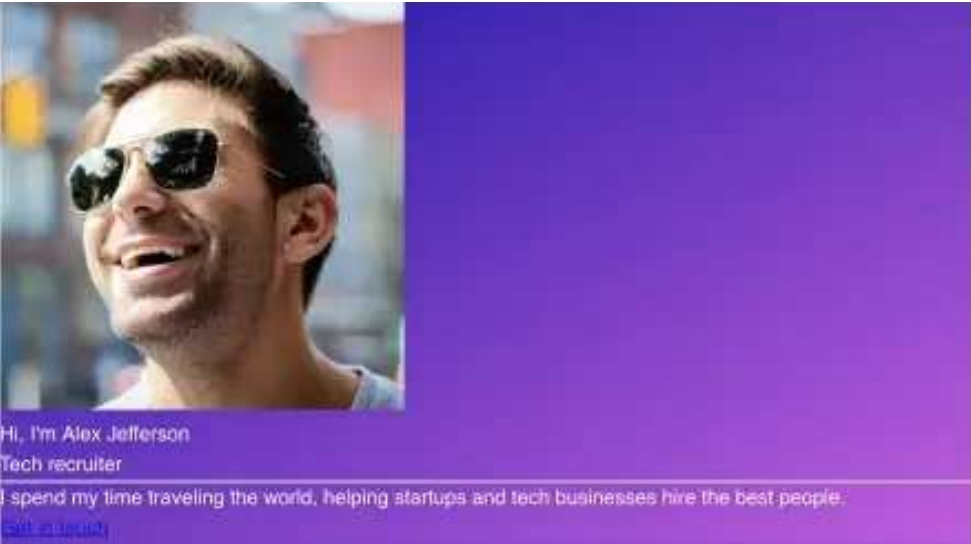
Los estilos HTML

El diseño general de una página web se realiza a través de estilos aplicados a la **html** elemento, porque la mayoría de ellos se conectan en cascada a través de todos los otros elementos secundarios.

En el vacío **main.css** archivo, agregue este código:

```
/* 1. Base /

html {
  color de fondo : # 5f45bb ;
  imagen de fondo : gradiente lineal (A la inferior derecha, # 180cac, # d054e4);
  color : #fff ;
  Familia tipográfica : "Arenas movediza" , Sans-serif;
  tamaño de fuente : 16px ;
  - moz-osx-font-suavizado : Escala de grises;
  - webkit-font-suavizado : Antialiased;
  altura de la línea : 1.5 ;
  min-altura : 100vh ;
  min-width : 300px ;
  overflow-x : oculto;
  sombra de texto : 0 3px 5px RGBA (0, 0, 0, 0,1); }
```



Este conjunto de reglas es básico, pero hace el trabajo:

- el **antecedentes** y **color** declaraciones marcan el tono general de la **fuentes** y **sombra de texto** establecer la base de la tipografía la **altura de la**
- **línea** y el navegador específica **font-alisado** declaraciones hacen la página más legible
-

- las dimensiones y valores de desbordamiento de asegurar que la página se llena toda la ventana gráfica

Mejores prácticas

Ordenar su CSS gobierna alfabéticamente

Al editar un archivo CSS, la búsqueda de la declaración de cambio puede llevar mucho tiempo. Para reducir el **carga cognitiva** cuando ambos *leyendo* y *escritura* reglas, que escriban en orden alfabético. Es la regla sólo se puede clasificar de que es a prueba de futuro porque es opinionless.

Cuando va a tener más de 10 declaraciones CSS en una regla, se le alegra saber dónde encontrar lo que estaba buscando en una fracción de segundo!

los **color de fondo** por lo general actúa como una alternativa en caso de la **imagen de fondo** no se carga, pero ya que estamos utilizando una **gradiente lineal()** casi siempre aparecerá. aparece Este gradiente *en la parte superior* de los antecedentes singlecolored, de ahí que sólo vemos el gradiente.

La propiedad font-family toma una lista de posibles familias. Si la fuente de Quicksand no está disponible, el navegador utilizará la fuente de reserva, siempre que: **sans-serif**. Esto generalmente significa Arial o Helvetica en máquinas Windows, San Francisco, el Mac OS y Ubuntu en máquinas Linux. Desde sus visitantes probablemente no tendrán esa fuente, la incluiremos usando **Fuentes de google**.

los **altura de la línea** tiene un valor sin unidades de **1.5**. Significa cada *línea* del texto será de 1,5 veces el tamaño de fuente actual del elemento. Durante la mayor parte de la página, y combinado con el **font-size: 16px** declaración, esto hará que cada línea de texto **24px** alto.

Siempre tengo una **min-width: 300px** que impide que la página sea demasiado estrecho para ser legible.

los **min-altura: 100vh** asegura la página sea al menos tan alto como 100% de la altura de visualización (**vh**).

los **desbordamiento-x: hidden** Declaración impide que la página de desplazamiento horizontal, preservando el desplazamiento vertical de costumbre.

La adición de las fuentes de Google

En **index.html**, antes de incluir **minireset.min.css**, añadir este CSS desde Google Fuentes. Ahora tienes **3 archivos CSS**:

```
< enlace href = "Https://fonts.googleapis.com/css?family=Montserrat|Quicksand" rel = "Hoja de estilo" > < link rel = "Hoja de estilo" tipo = "Text / css" href = "Css / minireset.min.css" > < link rel = "Hoja de estilo" tipo = "Text / css" href = "Css / main.css" >
```

Esto añade dos fuentes a su página: arena movediza y Montserrat, tanto en sólo el peso de la fuente normal (valor de **400**). Recargar la página para ver las fuentes en acción:



El estilo de enlaces

Los enlaces son por defecto azul y subrayado. Añadir esto en tu CSS:

```
un {
  color : Currentcolor;
  cursor : Puntero;
  text-decoration : ninguna; }
```

Utilizando **currentcolor** se prefiere porque va a recoger el juego de colores de **html** antes de: **Color: #fff**.

Actualización del color tanto sólo se requiere un único cambio de línea.

A veces, el cursor de mano no aparece cuando se pasa de un enlace. Es por eso que siempre añadido **cursor: puntero** para forzar la mano a aparecer.

"Ponte en contacto" se ha convertido en blanco, así:



4 Definir el diseño

el fondo de pantalla

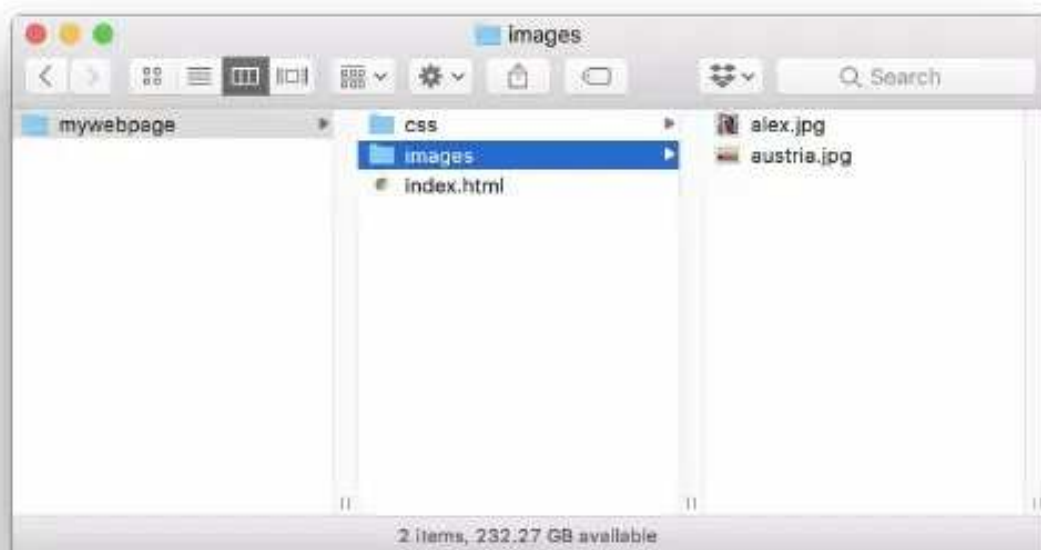
Al final de **main.css** añadir este fragmento:

```
/* 2. Compartido /  
  
. fondo de pantalla {  
    monitor : Block;  
    altura : 100% ;  
    izquierda : 0 ;  
    parte superior : 0 ;  
    anchura : 100% ;}
```

Este es el **2. compartido** sección de la CSS. Aquí, diferentes selectores compartirán la *mismo conjunto de reglas*. Vamos a ampliar esta sección, ya que necesitamos en el camino.

Por ahora, esta regla asegura que el fondo de pantalla cubre toda la página.

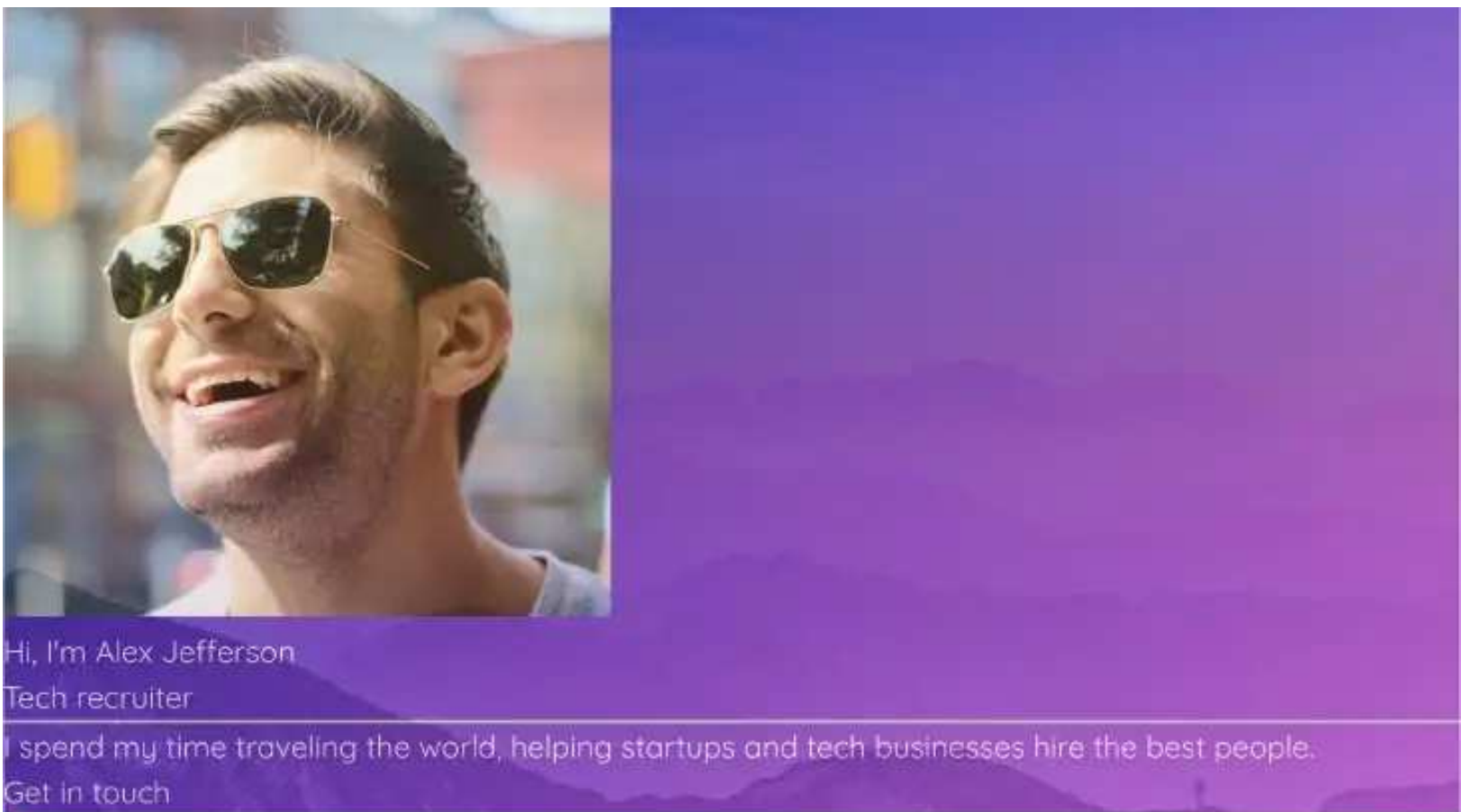
Ahora entramos en la **3. específica** sección donde el estilo de cada elemento individual. Añade el **austria.jpg** archivo a la **/ imágenes** carpeta:



A continuación, añadir este fragmento al final de **main.css** :

```
/* 3. específico */

.fondo de pantalla {
  imagen de fondo : url (" ../Images/austria.jpg" );
  background-position : Center;
  fondo de tamaño : cubrir;
  opacidad : 0.2 ;
  posición : Fijo; }
```



Esta imagen no se muestra con una `` etiqueta porque su propósito es decorativo y pertenece a la CSS.

Gracias al posicionamiento fijo, el papel tapiz no se desplazarán con la página.

Mejores prácticas

Evitar la notación abreviada

Estamos utilizando cada uno **antecedentes-*** propiedad en lugar de la taquigrafía **antecedentes** uno. Si utiliza **fondo: Rojo** , está configurando esencialmente **background-color: rojo** y resetear todas las otras propiedades a su **inicial** valor.

Se convierte en un problema cuando en realidad se anula una declaración que había escrito anteriormente, y uno se pregunta por qué su línea de CSS no funciona! Si **color de fondo** se había establecido anteriormente, habría sido deshecha aquí. Las propiedades sólo es útil taquigrafía son **margen** , **relleno** y **frontera** .

diseño principal

los **contenido** elemento es el padre de los dos:

- **lado** con la imagen del retrato
- **acerca de** con el contenido del texto

Vamos a uso **CSS Flexbox** para establecer el diseño global de la página.

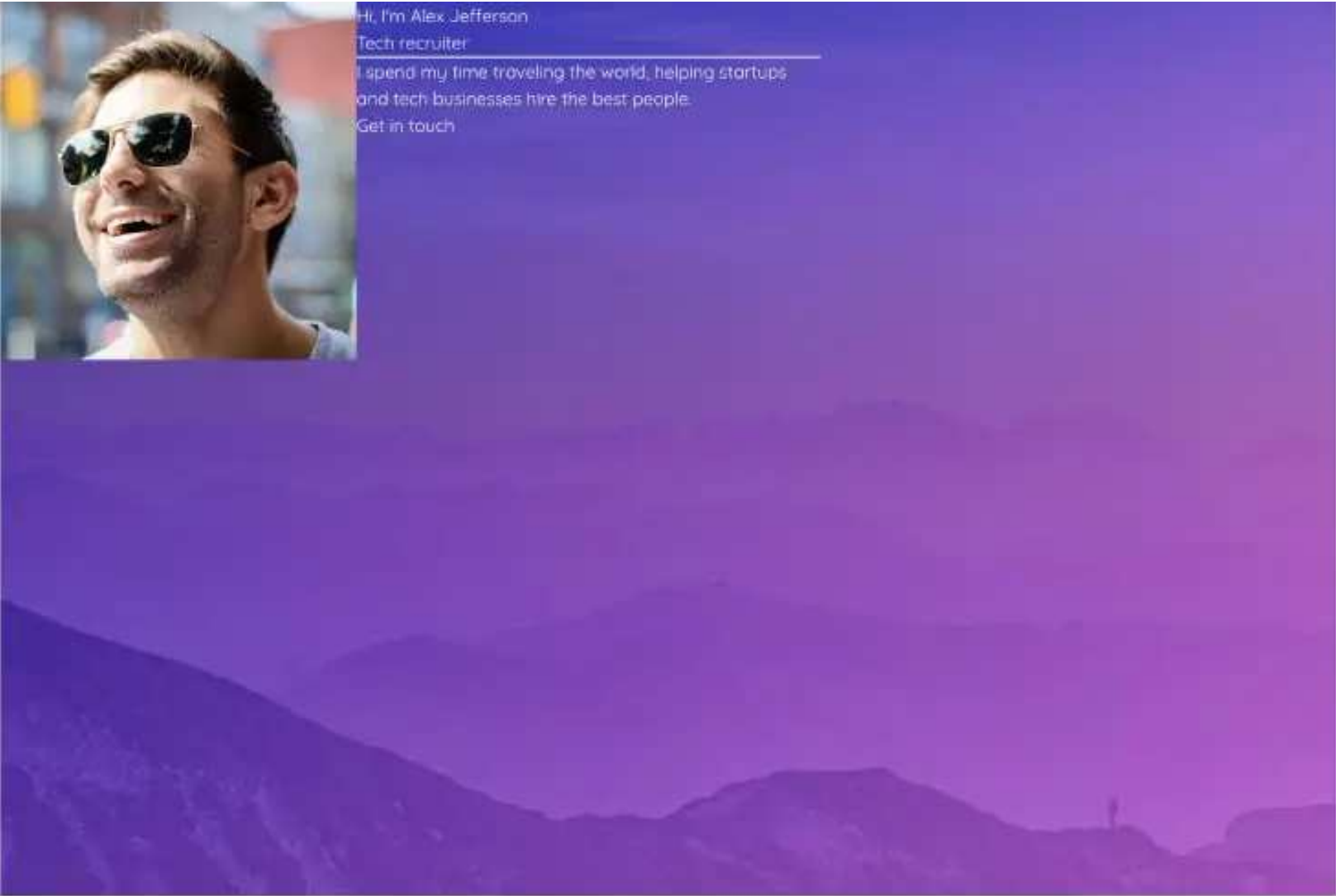
```
. contenido {
  monitor : flexionar;
  posición : relativo;
  min-altura : 100vh ; }

. lado {

  Altura máxima : 20rem ;
  anchura máxima : 20rem ; }

. acerca de {

  anchura máxima : 26rem ; }
```



Hacemos posible que en el Flexbox **. contenido** por el simple uso **display: Flex** . Esto hace que tanto **. lado** y **. acerca de** Flexbox **artículos**.

los **position: relative** permite **. contenido** a aparecer *encima* **. fondo de pantalla** , y el **min-altura** está aquí para asegurarse **. contenido** cubre toda la página.

El uso de **anchura máxima** es muy práctico: sólo significa que en *Cualquier punto* no queremos un elemento a por más ancho que un cierto valor. Por razones de legibilidad queremos que el **. acerca de** sección para ir más allá nunca se **26rem** de ancho (que es $26 \times 16px = 676px$). Puesto que es un elemento de bloque, se utilizará todo el ancho de hasta a disposición de un cierto punto. Dependiendo de la longitud de su contenido escrito, se puede jugar con el valor aquí.

Evitar el índice z cuando sea posible

Todos los elementos de posicionado (**absoluto** , **fijo** , o **relativo**) puede tener una **z-index** valorar apilarlos uno con respecto al otro.

Pero los elementos que ya se apilan en función de su ubicación dentro del HTML *código*.

Porque **. fondo de pantalla** y **. contenido** son *hermanos*, el segundo **. contenido** aparecerá por encima (sólo si se encuentra demasiado).

Y aparecerán elementos secundarios *encima* sus padres.

Es mejor colocarlos como desee en el código HTML, de lo contrario tendrá que hacer un seguimiento de todos **z-index** los valores a través de su CSS.

Haciendo que el diseño de respuesta

Queremos que nuestro diseño sea **sensible**, lo que significa que será diferente en ventanas más pequeñas (por debajo **800px**)

que en los más amplios (**800px** y por encima):

- en **móvil**, vamos a tener un centrado *vertical* Se distribuye en **escritorio**, vamos
- a tener un centrado *horizontal* diseño

Vamos a escribir una **medios de consulta**: cualquier CSS escrito dentro de ese bloque sólo se activará si todos los elementos de la lista (consulta de **medios pantalla y (max-width: 799px)**) son **clerto**. En este caso, estamos apuntan a los dispositivos móviles que tienen una estrecha ventana o igual a **799px** .

Afortunadamente, incluso en el escritorio simplemente puede cambiar el tamaño de su navegador para verlo en acción.

/* 4. Capacidad de respuesta /

```
@medios de comunicación pantalla y (max-width: 799px) {  
  . contenido {  
    flex-dirección : Columna;  
    justificar el contenido : Center;  
    align-artículos : Center;  
    relleno : 5rem 3rem ; }  
  
  . lado {  
    margin-bottom : 3rem ;  
    anchura : 100% ; }}
```



Cambiar el tamaño de su navegador para verlo en acción.

Estamos utilizando las siguientes propiedades FlexBox:

- **flex-dirección: columna** hace que el diseño vertical
- **justificar contenido: Centro** hace que el contenido centrado en el **principal eje** (vertical)
- **align-items: Centro** hace que el contenido centrado en el **cruzar eje** (horizontal)

los **relleno** impide que el contenido de tocar los bordes de ventana gráfica, lo que le da un poco de espacio para respirar.

los **. lado** (lo que ocurra primero) tiene una **margin-bottom** para separarlo de la **. contenido** (que viene segundos).

vista de escritorio

En el escritorio, queremos que el diseño para ser **horizontal** en lugar:

```
@medios de comunicación pantalla y (min-width: 800px) {  
  
  . contenido {  
    align-artículos : Center;  
    justificar el contenido: Espacio-alrededor;  
    justificar el contenido : Espacio-de manera uniforme;  
    relleno : 4rem ; }  
  
  . lado {  
  
    flex-Grow : 0 ;  
    flex-retráctil : 0 ;  
    altura : 20rem ;  
    margin-left : 4rem ;  
    orden : 2 ;  
    anchura : 20rem ; }  
  
  . acerca de {  
    flex-Grow : 1 ;  
    flex-retráctil : 1 ; }}
```



los **. contenido** utiliza el valor por defecto de **flex-dirección** cual es **fila** , es decir, los artículos (**. lado** y **. acerca de**) se extienden a lo **horizontalmente**. **justificar el contenido** aparece dos veces porque la **espacio-uniformemente** valor no está disponible en todos los navegadores, por lo que utilizar **espacio alrededor** como punto de retorno razonable.

los **. lado** dimensiones elemento (que contiene la imagen), ha fijos de una **20rem por 20rem** cuadrado. Si hay más espacio horizontal disponible, *no hacer* quiere que crezca, por lo tanto, la **flex-crecer: 0** . Por otro lado, tampoco queremos que se encoja en absoluto, de lo contrario, la imagen se squased. Es por eso **flexionar-**

reducir el tamaño: 0 se utiliza aquí también. Básicamente, queremos que el **. lado** ser - estar **20rem** por **20rem** en todo momento. También queremos que aparezca *después* el contenido del texto, que es por eso que usamos **orden: 2**.

Para el **. acerca de**, nosotros *hacer* quiere que se utilice el espacio que queda disponible, en ambas direcciones, por lo que **flex-Grow** y **flex-retráctil** tener tanto un valor de **1**.

Mejores prácticas

En preguntas de los medios, no para deshacer, acaba de hacer

Con enfoques móviles, en primer lugar, es fácil hacer un trabajo de diseño bien en las pantallas estrechas, y luego "deshacer" la mayor parte de ella en el escritorio. Pero eso es complicado porque hay que realizar un seguimiento de lo que se ha hecho *fuera de* de preguntas de los medios, y restablecer esos valores *dentro* la consulta de medios de escritorio. También termina escribiendo un montón de CSS sólo para valores de reset, y se puede terminar dejando como CSS **margin-bottom: 0** no está seguro de qué. los **margin-bottom** establecido para el **. lado** elemento debe **solamente** aparecer en *móvil*.

En lugar de aplicar un margen por defecto en *todas* pantallas, y la eliminación en *escritorio*, que sólo se aplican en *móvil*.

5 El estilo de todos los elementos

La imagen

La sombra se separa de la imagen, ya que vamos a **animar** por separado más adelante.

En el **2. compartido** sección de la CSS, actualizar la lista de **selectores** y añadir tanto **. foto-sombra** y

. foto-imagen :

```
. fondo de pantalla ,
. foto-sombra ,
. foto-imagen {
  monitor : Block;
  altura : 100% ;
  izquierda : 0 ;
  parte superior : 0 ;
  anchura : 100% ; }
```

Al final de **3. específica** sección de la CSS, añadir los siguientes estilos:

```
. imagen {
  acolchado superior : 100% ;
  posición : relativo;
  anchura : 100% ; }

. foto-sombra {
  frontera de radio : 290486px ;

  imagen de fondo : radial-gradiente (# 000 0%, RGBA (0, 0, 0, 0) 70% );
  posición : absoluto;
  parte superior : 10% ;
}

. foto-imagen {
  frontera de radio : 290486px ;
  posición : absoluto; }
```

los **padding-top: 100%** es una técnica que hace que el **. imagen** como *alto* como están las cosas *amplio* haciendolo **cuadrado** en todo momento. También tiene una **position: relative** de modo que actúe como punto de referencia para sus dos hijos posición absoluta.

La sombra utiliza una semi-transparente **radial-gradiente ()** . Está ligeramente desplazada hacia la parte inferior con

la parte superior: 10% .

los **frontera de radio** se establece en el valor muy alto de **290486px** para asegurar los elementos para ser **redondeado**. Se puede utilizar cualquier valor extremadamente alto. Yo personalmente uso **290486px** como marca, porque es mi fecha de nacimiento. Es muy interesante ver que se muestre en **código de otras personas** !



El nombre

El nombre es la información más importante de la página. Es por eso que utiliza el **< h1>** etiqueta HTML, que tiene el valor semántico más fuerte. Para reflejar esta prominencia visual, así, que vamos a hacerlo más grande:

```
. nombre {
  tamaño de fuente : 2.25rem ;
  altura de la línea : 1.125 ;
  margin-bottom : 0.5rem ; }
```



El tamaño de fuente utiliza el **movimiento rápido del ojo** unidad: es la raíz valor, igual al tamaño de la fuente situada en el **html** elemento, que previamente hemos creado a **16px** . Entonces **2.25rem** Es esencial **36 píxeles** . Utilizando **movimiento rápido del ojo** es útil porque hace referencia a un valor común, y podemos actualizar la **html** valor al conjunto *todas* instancias de **movimiento rápido del ojo** valores.

los **altura de la línea** se establece en **1.125** Es difícil ver su propósito si el texto es demasiado corto (cambiar el tamaño de su navegador para llegar a dos líneas), pero manteniendo el valor de la página **1.5** hace que las dos líneas demasiado espaciados.

Mejores prácticas

Conjunto line-height primero, el margen / segundo a colchado

Algunos desarrolladores utilizan la línea de altura como una manera de dar espacio a un elemento. Sin embargo, como su nombre indica, es la intención de definir la altura de una *línea sola*, no el espacio *Entre* cada línea. El valor de línea de altura se debe establecer para **legibilidad** propósitos *solamente*.

Si tiene que dar más espacio para respirar a un elemento, sólo tiene que utilizar un poco de margen (o, a veces relleno), que es lo que estamos haciendo aquí con **margin-bottom** .

Título profesional

Queremos el título del trabajo que se destacan un poco. Es por eso que vamos a utilizar nuestra fuente secundaria: **Montserrat**.
Esta fuente funciona bien cuando las letras son mayúsculas y ligeramente espaciados.

En el **2. compartido** sección, añadir este fragmento:

```
.trabajo ,  
  
.botón {  
  Familia tipográfica : "Montserrat" , "Arenas movediza" , Sans-serif;  
  espaciado de letras : 0.3em ;  
  text-transform : Mayúsculas; }
```

Este estilo va a ser utilizado para nuestro botón, así que vamos a añadir los dos selectores en este momento.

En el **3. específica** , añada la siguiente:

```
.trabajo {  
  color : # ffe479 ;  
  tamaño de fuente : 0.75rem ; }
```



El estilo del texto en mayúsculas hace bastante "en su cara", por lo que estamos reduciendo el tamaño de la fuente un poco, y también la aplicación de un tono de amarillo.

La línea hr

La regla horizontal (**hora**) define un descanso semántica entre bloques de texto . Mientras se mantiene este valor semántico intacta, queremos hacer de esta línea más sutil:

```
.hora {  
  color de fondo : # ff470f ;  
  frontera : ninguna;  
  contenido : "" ;  
  altura : 1px ;  
  margin-bottom : 1.5rem ;  
  margin-top : 1.5rem ;  
  transformar origen : Centro izquierda;  
  anchura : 4rem ; }
```



Para un mayor control, estamos eliminando la **frontera** y el uso de la **color de fondo** con una altura de **1px** para definir una fina línea corta.

los **transformar origen** se utilizará cuando animar la anchura más adelante.

Descripción

La descripción sólo tiene que ser un poco más prominente. Vamos a aumentar el tamaño de letra:

```
. descripción {
  tamaño de fuente : 1.5rem ; }
```



botón de contacto

```
. contacto {
  monitor : Inline-block;
  margin-top : 1.5rem ;
  vertical-align : parte superior; }
```

Mediante el uso **display: inline-block** combinamos dos comportamientos:

- **en línea** se asegura de que la anchura es igual a su contenido (el botón)
- **bloquear** se asegura de elementos que lo rodean van a aparecer por encima y por debajo de ella también nos
- permite el uso **margin-top**

Y **vertical-align: top** asegura el elemento de utilizar sólo el espacio vertical requerido, y mantiene el estrecho espaciado.

Primero tenemos que añadir un poco de CSS en el **2. compartido** sección:

```
. botón ,
. social un {
  transformar origen : Center;
  duración de la transición : 100 ms ; }
```

Estamos estableciendo unos valores de transformación y de transición compartidas entre el botón y los vínculos sociales (que vendrán más adelante).

los **duración de la transición** es compartida por lo que sólo hay que sustituir el valor en una sola ubicación si es necesario.

Ahora nos podemos centrar en el **3. específica** sección para el propio botón, que es el elemento más elaborado que tenemos:

```
. botón {
  color de fondo : #fff ;
  frontera de radio : 290486px ;
  sombra de la caja : 0 1rem 2rem RGBA (0, 0, 0, 0,2);
  color : # 9013fe ;
  monitor : Inline-block;
  tamaño de fuente : 0.875rem ;
  altura de la línea : 1 ;
  relleno : 1.25em 2em ;

  sombra de texto : ninguna;
  transición de propiedad : Box-shadow, transformar;

  El usuario seleccione : ninguna;
  vertical-align : parte superior;
  de espacio en blanco : Nowrap;
  cambiará : Box-shadow, transformar; }
```



El botón utiliza el mismo **inline-block** técnica como su padre.

los **altura** del botón es **proporcional** al tamaño de fuente, e igual a **3.5** veces el valor de tamaño de letra:

- sin unidades del **altura de la línea** es igual a **1** o **0.875rem**
- los acolchados verticales (superior e inferior) se establecen para **1.25em** cada uno, o **1.25** veces el tamaño de la fuente

los **em** unidad es igual a la *Actual*/tamaño de fuente. En este caso, es similar al valor de altura de la línea sin unidades.

Si juegas un poco con el **tamaño de fuente** valor, por el ajuste **2rem** por ejemplo, usted notará que el botón se **cambiar el tamaño proporcionalmente**. Esto es muy útil porque sólo necesitamos una actualización *solo valor* para aumentar / disminuir el tamaño del botón, mientras se mantiene una relación perfecta.

También aplicamos algunos otros estilos:

- debido al color de fondo y el color del texto más oscuro, quitamos la sombra de texto que usamos **el usuario seleccione: ninguno** por lo que el contenido
- de texto no se puede seleccionar, lo que puede suceder cuando se hace clic repetidamente en el botón
- Para asegurarse de que el texto no se visualiza en dos líneas, utilizamos **white-space: nowrap**

los **transición de propiedad** y **cambiará** Los valores son para los estados de botón.

estados de botón

El botón es el elemento principal interacción de la página. Los visitantes pueden flotar en el botón y haga clic en él.

```
. botón :flotar {
  sombra de la caja : 0 1.5rem 3rem RGBA (0, 0, 0, 0,2);
  transformar : escala (1,02) translateY (-4px); }
```



La sombra caja es transparente por lo que puede funcionar en cualquier color de fondo. Para la transformación, aumentamos el tamaño del botón en un 2% con **escala (1,02)** y moverlo hacia arriba por unos pocos píxeles con **translateY (-4px)** .

Al hacer clic en el botón, nosotros queremos que se vea como si se ha pulsado *hacia abajo*.

```
. botón :activo {
  sombra de la caja : 0 0.5rem 1rem RGBA (0, 0, 0, 0,3);
  transformar : escala (0,98) translateY (-2px); }
```



Estamos jugando con las mismas propiedades pero con diferentes valores. La sombra es más fuerte y más pequeño debido a que el botón aparece más cerca de la "tierra".

Jugar con el **duración de la transición** valor establecido en el **2. compartido** sección para ver más lento o más rápido.

6 Importación de iconos impresionante fuente

[fuente impresionante](#) es una biblioteca de iconos que le permite incluir fácilmente los iconos en su página web. Lo que ofrece es esencialmente una **icono de la fuente** lo que simplemente es una fuente de texto en el que todos los personajes han sido reemplazados por iconos. Así que en lugar de escribir **formas de *letras*** básicamente estás escribiendo formas de *iconos*. Esto funciona porque las letras son formas de vectores propios. Sólo ocurren para poder leerlos!

Estamos incluyendo toda la biblioteca impresionante fuente directamente de la CDN. La ventaja es que todos los iconos están disponibles. La desventaja es que los archivos cargados son mucho más grandes en tamaño. Se puede utilizar un generador de fuente icono como [IcoMoon](#) o [Fontello](#) para incluir sólo los iconos que realmente necesita. Esto reducirá el tamaño de los archivos de fuentes. Otra opción es utilizar imágenes SVG directamente, pero eso es más complicado!

Importación de la biblioteca impresionante fuente

Justo antes del cierre `</ head>` etiqueta de `index.html` , incluir la siguiente secuencia de comandos:

```
< guión src Defer = "Https://use.fontawesome.com/releases/v5.0.0/js/all.js" > </ guión >
```

Justo después de la `< div class = "contacto">` , agregar estos iconos sociales:

```
< ul clase = "social" > < li > < un > < yo clase = "Fab fa-twitter" > </ yo >

    </ un >
</ li > < li > < un > < yo clase = "Fab fa-github" > </ yo >

    </ un >
</ li > < li > < un > < yo clase = "Fab fa-linkedin" > </ yo >

    </ un >
</ li >
</ ul >
```



Si los iconos no aparecen, asegúrese de que está conectado a Internet y que ha habilitado JavaScript.

El estilo de los iconos

Si vuelve a cargar la página repetidamente, puede notar que la disposición "salta" de una fracción de segundo. Eso es porque en un primer momento, los iconos no están cargados, y la página sólo muestra el texto y las imágenes. Entonces, cuando los iconos fin de carga, el pop-up, y hacen que la página se vuelve a dibujar, de ahí el salto.

Para evitar este salto, vamos a establecer un área cuadrada de **2rem** por **2rem** para cada **** elemento de la lista:

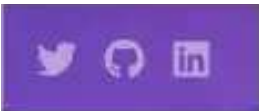
```
. social {
  monitor : flexionar;
  margin-top : 1.5rem ; }

. social li {

  altura : 2rem ;
  margin-right : 0.5rem ;
  texto alineado : Center;
  anchura : 2rem ; }
```

los **** Lista actúa como contenedor Flexbox para cada elemento de la lista cuadrado. Ahora podemos estilo de los enlaces internos:

```
. social un {
  align-artículos : Center;
  monitor : flexionar;
  tamaño de fuente : 1.5rem ;
  altura : 2rem ;
  justificar el contenido : Center;
  opacidad : 0.5 ;
  transición de propiedad : Opacidad, transformar;
  anchura : 2rem ;
  cambiará : Opacidad, transformar; }
```



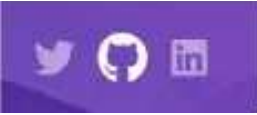
Cada elemento de enlace es un cuadrado, así, en la que el icono de fuente impresionante es tanto vertical como horizontalmente **centrado**.

Debido a que los iconos son *texto*, podemos utilizar el **tamaño de fuente** la propiedad de aumentar el tamaño del icono. Y ya que hemos puesto todo **<a>** etiquetas para utilizar el **currentcolor**, que recoge el color blanco fijo de la página.

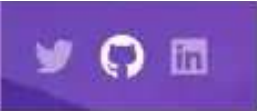
También estamos establecer la opacidad al medio transparente, que es conveniente porque es válida para cualquier color usaríamos.

En cuanto a la tecla, vamos a añadir un poco de vuelo estacionario y estilos activos:

```
. social un :flotar {  
  opacidad : 1 ;  
  transformar : escala (1,25); }
```



```
. social un :activo {  
  opacidad : 1 ;  
  transformar : escala (1,1); }
```



7 Creación de animaciones CSS

La animación en el CSS está cambiando básicamente un conjunto de *valores* terminado *hora*. Por ejemplo, el desvanecimiento de un elemento se hace estableciendo el *opacidad* a cero *0* , y gradualmente incrementarlo (*0.1* , *0.2* , *0.3* ...) hasta llegar finalmente al valor de *1* .

Se necesitaría un tiempo para averiguar toda la *intermedio* valores. Por suerte en el CSS, sólo necesita conjunto **2 valores**:

- los valores de inicio, con la palabra clave *desde*
- los valores finales, con la palabra clave *a*

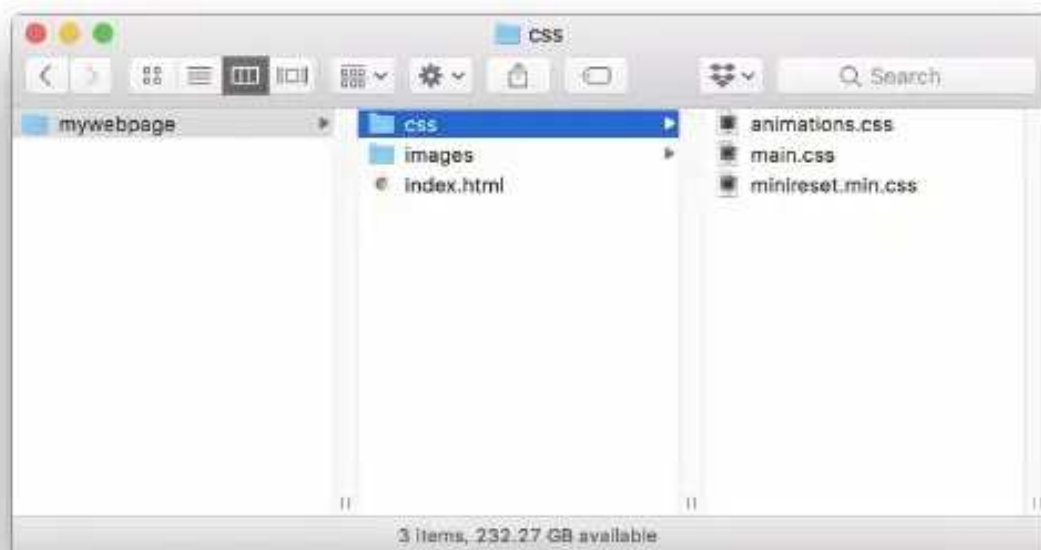
Todo lo demás será determinado por:

- el *duración de la animación* : cuánto tiempo se tarda en ir desde el principio hasta el final *animación-timing-function* : cómo se
- calculan los valores intermedios (basado en una curva)

Las animaciones CSS se activan cuando se carga la página, o cuando cambia un nombre de clase.

La adición de un nuevo archivo CSS

Las animaciones CSS escritura toma un montón de espacio, así que vamos a crear un nuevo archivo en el */css* carpeta y llamarla *animations.css* .



Añadirlo a la lista de CSS incluido. Ahora debe tener **4 hojas de estilo**:

```
< enlace href = "Https://fonts.googleapis.com/css?family=Montserrat|Quicksand" rel = "Hoja de estilo" > < link rel = "Hoja de estilo" tipo = "Text / css" href = "Css / minireset.min.css" > < link rel = "Hoja de estilo" tipo = "Text / css" href = "Css / main.css" > < link rel = "Hoja de estilo" tipo = "Text / css" href = "Css / animations.css" >
```

fotogramas clave

UN **@ fotogramas clave** declaración contiene una lista de fotogramas clave, que son todos los pasos intermedios de una animación.

Vamos a utilizar la versión simple, donde sólo tenemos que configurar dos fotogramas clave:

- **desde** es el fotograma clave en el inicio de la animación
- **a** es el fotograma clave al final de la animación

Puede tener más de 2 fotogramas clave, si se utiliza **porcentajes** en lugar.

Aquí está el primer fotograma clave que vamos a utilizar. Añadirlo a **animations.css** :

```
@ fotogramas clave bounceIn {
  desde {
    opacidad : 0 ;
    transformar : escala (0,5); }

  a {
    opacidad : 1 ;
    transformar : escala (1); }}
```

Esta **bounceIn** animación completa dos cambios de forma simultánea:

- el aumento gradual, desde completamente transparente a completamente opaco ampliación de
- escala, de la mitad del tamaño al tamaño original

Vamos a escribir 7 declaraciones de fotogramas clave en total:

- **disminuir el zoom** para el fondo de pantalla
- **picImage** para la imagen foto
- **picShadow** que la sombra de imagen
- **bajar deslizándose** para el nombre
- **deslizar hacia arriba** para todos los demás elementos de texto
- **llena** para la línea hr
- **bounceIn** para el botón de contacto

declaraciones generales

Al igual que con nuestro CSS principal, vamos a compartir algunas declaraciones entre varios selectores.

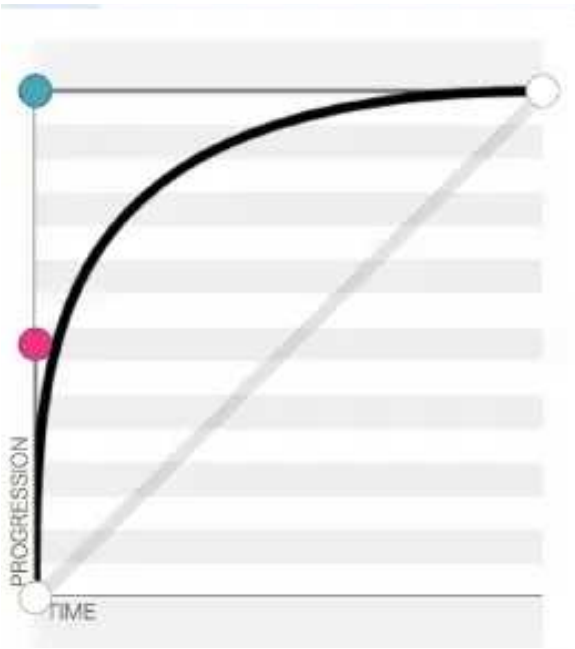

```
. fondo de pantalla ,
. foto-sombra ,
. foto-imagen ,
. nombre ,
. trabajo ,
. hora ,
. descripción ,
. contacto ,
. social li {

    duración de la animación : 1s ;
    animación-timing-function : cúbicas Bézier (0, 0,5, 0, 1);
    animación-fill-mode : ambos; }
```

La duración se establece en un segundo (1s).

los **animación-fill-mode** ajustado a **ambos** indica al navegador que utilice el **desde** valores *antes de* la animación, y para mantener la **a** valores *después* la animación ha terminado. De lo contrario, el elemento volvería a sus estilos no animado.

Para hacer la animación más interesante y más ágil, estamos utilizando una costumbre **Bezier** curva. [Verlo en acción!](#)



Animación del botón de contacto

Para una animación que se active automáticamente, sólo se necesita la duración y el nombre. Vamos a añadir el nombre:

```
. contacto {
    animación-nombre : BounceIn; }
```

[Haga clic para ver el botón Bouncein en la acción](#)

otros fotogramas clave

Aquí están los otros 6 estados de fotogramas clave vamos a necesitar:

```
@ fotogramas clave llena {  
  desde {  
    transformar : scaleX (0); }  
  
  a {  
    transformar : scaleX (1); }}
```

```
@ fotogramas clave picImage {  
  desde {  
    opacidad : 0 ;  
    transformar : escala (1,2) translateY (-1rem); }  
  
  a {  
    opacidad : 1 ;  
    transformar : escala (1) translateY (0); }}
```

```
@ fotogramas clave picShadow {  
  desde {  
    opacidad : 0 ;  
    transformar : escala (1,2) translateY (4rem); }  
  
  a {  
    opacidad : 1 ;  
    transformar : escala (1,1) translateY (0); }}
```

```
@ fotogramas clave bajar deslizándose {  
  desde {  
    opacidad : 0 ;  
    transformar : translateY (-1rem); }  
  
  a {  
    opacidad : 1 ;  
    transformar : translateY (0); }}
```

```
@ fotogramas clave deslizar hacia arriba {  
  desde {  
    opacidad : 0 ;  
    transformar : translateY (1rem); }  
  
  a {  
    opacidad : 1 ;  
    transformar : translateY (0); }}
```

```
@fotogramas clave disminuir el zoom {  
  desde {  
    opacidad : 0 ;  
    transformar : escala (1,05); }  
  
  a {  
    opacidad : 0.2 ;  
    transformar : escala (1); }}
```

otras animaciones

Tanto la sombra de la imagen de una necesidad imagen ligeramente *Más rápido* animación, y una curva de animación diferente:

```
. foto-sombra ,  
. foto-imagen {  
  duración de la animación : 750 ms ;  
  animación-timing-function : cúbicas Bézier (0, 0.5, 0.25, 1.25); }
```

Ahora podemos desencadenar todas las animaciones mediante el uso de **animación-nombre** para todos ellos, al tiempo que algunos otros ajustes también:

```
. fondo de pantalla {  
  animación-nombre : disminuir el zoom;  
  animación-timing-function : Facilitarse; }  
  
. foto-sombra {  
  animación-nombre : PicShadow; }  
  
. foto-imagen {  
  animación-nombre : PicImage; }  
  
. nombre {  
  animación-nombre : bajar deslizándose; }  
  
. trabajo {  
  animación-nombre: deslizar hacia arriba; }  
  
. hora {  
  animación-nombre : llena; }  
  
. descripción {  
  animación-nombre: deslizar hacia arriba; }
```

```
. social li {  
  duración de la animación : 500 ms ;  
  animación-nombre : deslizar hacia arriba;  
  animación-timing-function : cúbricas Bézier (0,5, 0, 0,5, 1,5); }
```

los **. fondo de pantalla** no utiliza una curva de Bezier, sino más bien la **Facilitarse** palabra clave.

Los elementos sociales tienen una animación aún más rápido (500 milisegundos, o la mitad de un segundo) y diferente curva demasiado.

Actualizar su página y ver todas las animaciones activadas a la vez!

Haga clic para ver la acción simultánea animaciones en

El retraso de cada animación

Debido a que todas las animaciones están sucediendo *simultaneamente*, es difícil ver el efecto de cada uno de ellos. Lo que simplemente tenemos que hacer, es poner en funcionamiento las **en secuencia**.

Mediante el uso de un disparador **retrasar**, podemos lograr mucho!

```
. nombre {  
  animación de retardo : 100 ms ; }  
  
. trabajo {  
  animación de retardo : 200ms ; }  
  
. hora {  
  animación de retardo : 300ms ; }  
  
. descripción {  
  animación de retardo : 400ms ; }  
  
. foto-imagen {  
  animación de retardo : 500 ms ; }  
  
. foto-sombra {  
  animación de retardo : 500 ms ; }  
  
. contacto {  
  animación de retardo : 600ms ; }
```

Para los iconos sociales, queremos retrasar de forma individual. Podemos utilizar el **:nth-child** la pseudo selector para seleccionarlos:

```
. social li : Nth-child (1) {  
  animación de retardo : 800ms ; }
```

```
. social li : Nth-child (2) {  
  animación de retardo : 900ms ; }
```

```
. social li : Nth-child (3) {  
  animación de retardo : 1s ; }
```

```
. social li : Nth-child (4) {  
  animación de retardo : 1.1s ; }
```

```
. social li : Nth-child (5) {  
  animación de retardo : 1.2s ; }
```

Si usted tiene más de 5 iconos, simplemente aumentará en 0.1s cada vez.

Actualizar su página y ver su página entera animado!

[Haga clic para ver la Secuenciado animaciones en la acción](#)

8 Desencadenando animaciones con JavaScript

Puesto que usted ha venido desarrollando a nivel local, ambas imágenes (alex.jpg y austria.jpg) carga instantáneamente. Pero en un servidor de producción, estas imágenes serán unos segundos en cargar. Dado que la animación se inicia al instante, y sólo toma 1 segundo para completar, se habrá terminado *antes de* la imagen se ha cargado.

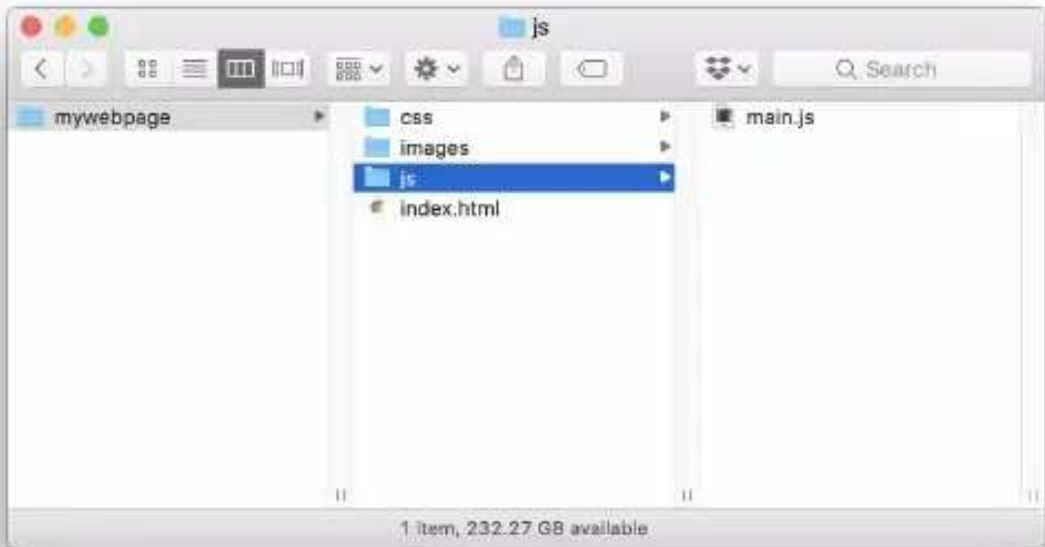
Para verlo en acción en la máquina local, simular una conexión lenta usando las herramientas de Google Chrome Dev :

- abre el " Herramientas de desarrollo" (con Ctrl + Shift + I o Cmd + Opt + I en Mac) hasta llegar a la " Red" habilitar la
- pestaña " Desactivar el caché" casilla de verificación en el " En línea" desplegable elegir " 3G lenta"
-
-

Actualizar su página y ver cómo las imágenes toman un tiempo para la carga.

Adición de un archivo JavaScript

Crear una nueva carpeta llamada / js , y en el interior crea un archivo llamado main.js .



En main.js , escriba lo siguiente:

```
alerta( "Hola JS!" );
```

En el < head> de su HTML, antes de que el guión impresionante fuente, incluir su propio guión. Ahora tienes 2 guiones:

```
< gui3n tipo Defer = "Text / javascript" src = "JS / main.js" > </ gui3n > < gui3n src Defer = "Https://use.fontawesome.com/releases/v5.0.0/js/all.js" > </ gui3n >
```

Actualizar su p1gina. Debería ver aparecer un cuadro de alerta!



Tu puedes ahora eliminar el `alerta()` funci3n.

Modificar el c3digo HTML

Tenemos que preparar nuestro HTML para interactuar con nuestros JS, mediante el uso de:

- **carne de identidad** Atributos, para seleccionar elementos dentro de nuestro c3digo JS
- **datos** Atributos, para pasar datos *desde* el c3digo HTML *a* la JS

localizar el **fondo de pantalla** elemento y a1adir la **carne de identidad** y **dataimagen** atributos:

```
< div id = "fondo de pantalla" clase = "fondo de pantalla" dataimagen = "Images / austria.jpg" > </ div >
```

A1adir una **carne de identidad** al **imagen** elemento:

```
< Identificaci3n figura = "imagen" clase = "imagen" >
```

Agrega otro **carne de identidad** al **foto-imagen** elemento:

```
< img id = "PictureImage"
      clase = "Foto-imagen"
      src = "Images / alex.jpg"

      srcset = "1x im1genes / alex.jpg,
               images / alex@2x.jpg 2x, im1genes /
               alex@3x.jpg 3x"
      alt = "Retrato de Alex Jefferson"
      anchura = "320"
      altura = "320" >
```

Al a1adir una clase CSS en la carga

Vamos a crear una funci3n JS que:

- obtener un elemento HTML
- hacer el origen de la imagen adjunta a ese elemento a crear una nueva
- imagen desde que la carga fuente de esa imagen
-
- cuando se carga la imagen, a1adir una nueva clase de CSS para el elemento

```
función cargar imagen (Id, targetId) {
  var EL = documento .getElementByld (id);
  var targetEl = targetId? documento .getElementByld (targetId): EL;
  var imageToLoad;

  Si (El.dataset.image) {
    imageToLoad = el.dataset.image; } else if ( tipo de el.currentSrc === 'Indefinido' )
  {ImageToLoad = el.src; } más {

    imageToLoad = el.currentSrc; }

  Si (ImageToLoad) {
    var img = nuevo Imagen(); img.src =
    imageToLoad; img.onload = función () {

      targetEl.classList.add ( 'está cargado' );
    }; }}
}
```

Vamos a través de este paso a paso la función. En primer lugar, se define una función que puede tomar dos parámetros llamados

carne de identidad y targetId :

```
función cargar imagen (Id, targetId) {}
```

A continuación, buscar el elemento HTML utilizando el valor de carne de identidad :

```
var EL = documento .getElementByld (id);
```

Nos manera similar buscar un elemento de destino para agregar una clase a, pero sólo si una targetId está provisto. De lo contrario, volvemos al elemento anterior:

```
var targetEl = targetId? documento .getElementByld (targetId): EL;
```

Creemos una instancia del imageToLoad variable:

```
var imageToLoad;
```

Hay 3 (mutuamente excluyentes) casos para establecer el valor de imageToLoad :

- Si dataimagen se proporciona, se utiliza ese valor si el navegador no soporta srcset y currentSrc , usamos la src valor de la
-
- de lo contrario simplemente podemos usar currentSrc


```
Si (El.dataset.image) {  
    imageToLoad = el.dataset.image; } else if ( tipo de el.currentSrc === 'Indefinido' )  
{ImageToLoad = el.src; } más {  
  
    imageToLoad = el.currentSrc; }
```

Comprobamos si **imageToLoad** mantiene un valor:

```
Si (ImageToLoad)
```

entonces se crea una nueva imagen a partir de la cual el **src** es el valor de la **imageToLoad** :

```
var img = nuevo Imagen(); img.src =  
imageToLoad;
```

Por último, cuando la imagen está completamente cargado por el navegador, añadimos la clase CSS **está cargado** al elemento de destino:

```
img.onload = función () {  
    targetEl.classList.add ( 'está cargado' );  
};
```

Ahora tenemos que añadir los **está cargado** estilos.

Configuración de los estilos cargados

Las animaciones para la **fondo de pantalla** y el **imagen-*** elementos todavía se activan de forma automática, así que vamos a **eliminar** ellos primero.

```
. fondo de pantalla {  
    animación-nombre : disminuir el zoom;  
    animación-timing-function : Facilitarse; }  
  
. foto-sombra {  
    animación-nombre : PicShadow; }  
  
. foto-imagen {  
    animación-nombre : PicImage; }
```

Y **reemplazar** con lo siguiente:

```
. fondo de pantalla {  
  animación-timing-function : Facilitarse; }  
  
. wallpaper.is-cargado {  
  animación de retardo : 1s ;  
  animación-nombre : disminuir el zoom; }  
  
. picture.is-cargado .picture-sombra {  
  animación-nombre : PicShadow; }  
  
. picture.is-cargado .picture-imagen {  
  animación-nombre : PicImage; }
```

También tenemos que establecer el **estado inicial** de esos elementos, ocultándolos en la salida. Agregue la siguiente:

```
. fondo de pantalla ,  
. foto-sombra ,  
. foto-imagen {  
  opacidad : 0 ; }
```

Si ahora vuelve a cargar la página, la imagen de fondo y la imagen son **oculto**:

- la opacidad se establece en cero **0**
- la animación que se desvanece en es *no* desencadenada

Ahora vamos a desencadenar las animaciones con JS.

Uso de la función

Hemos establecido la CSS para activar la animación cuando el **está cargado** . Y hemos definido una **cargar imagen()** función, pero no hemos llamado todavía. Vamos a probarlo!

Al final de **main.js** , llamar a la función doble:

```
documento .addEventListener ( 'DOMContentLoaded' , función () {  
  cargar imagen( 'fondo de pantalla' ); cargar imagen( 'PictureImage'  
  , 'imagen' );  
});
```

Estamos esperando a que todo el documento para ser cargado antes de llamar al **cargar imagen()** dos veces, con diferentes parámetros.

La página debería trabajar perfectly! Tratar de establecer la conexión de red "**3G lenta**" de nuevo, y el aviso cómo las animaciones esperan a que las imágenes de la carga antes de ser disparado.

caso extremo: sin guión

Si por alguna razón el visitante no tiene JS activado, aún quieren mostrarles las imágenes. La experiencia no será tan grande, pero es mejor que no muestra imágenes en absoluto.

En el código HTML, justo antes de la `</body>` etiqueta de cierre, puede añadir este `<noscript>` etiqueta:

```
< noscript > < tipo de estilo = "Text / css" >

    . fondo de pantalla {
        animación-nombre : disminuir el zoom; }

    . imagen {
        animación-nombre : entrar un momento; } </ estilo
    >

</ noscript >
```

Si JS está desactivada, simplemente disparar el animaciones. Ahora tiene la **Lo mejor de ambos mundos**:

- si es JS *habilitado*, espera el navegador para que las imágenes se cargan antes de las animaciones de disparo si es JS *discapacitado*, el navegador
- sigue mostrando las imágenes mediante la activación de las animaciones

9 Próximos pasos

Explora HTML y CSS

Si bien hemos cubierto el 90% de lo que se necesita para construir páginas web con HTML y CSS, hay otras etiquetas y funciones disponibles. Puede encontrarlos en el [Mozilla Developer Network](#) . Si quieres una versión más simple, tratar los recursos gratuitos que he construido: [Referencia HTML](#) y [Referencia CSS](#) .

aprender Sass

[Hablar con descaro a](#) es un CSS **preprocesador** que permite a CSS escribir más fácilmente. Viene con un montón de características, pero sobre todo **variables** (para establecer valores como colores, tamaños, duraciones ...), (**mixins** como funciones JS pero para su CSS) o **se extiende** (compartir declaraciones de que se extiende otro selector). Puede leer mi [capítulo libre MarkSheet en Sass](#) Como un inicio.

aprender JavaScript

Si bien hemos cubierto un poco de JS, hay una *lote* más que aprender. Hay un montón de recursos gratuitos en línea, al igual [elocuente JavaScript](#) , [JavaScript Jardín](#) o [Codecademy](#) .

Para las interfaces más complejas, se puede utilizar una **marco JS** me gusta [Reaccionar](#) , [AngularJS](#) , [Ember.js](#) o [Vue.js](#) .

Aprender el desarrollo backend

Nosotros sólo hemos cubierto *lado del cliente* desarrollo (o "frontend"). Sin embargo, para los sitios web más complejos, que tendrá que aprender *lado del servidor* desarrollo (o "back-end"). Esto requiere un aprendizaje **lenguaje de programación** (como JavaScript, Python, Ruby, PHP ...). De nuevo, [Codecademy](#) es un gran recurso libre.

Pruebe Bulma

Con base en el conocimiento adquirido a través de 10 años de experiencia, he creado un marco de código abierto llamado CSS [Bulma](#) .

Se puede utilizar directamente a las interfaces de construcción, simplemente mediante el uso de las clases CSS previstos. Es modular, por lo que puede incluir sólo lo que necesita. Pero ya que es probable que desee escribir su propio CSS después de haber leído este libro, también es un buen punto de partida para cualquier tipo de trabajo. Siempre lo uso en mis proyectos personales y profesionales de alguna manera.

Es también un buen recurso para el aprendizaje. Al pasar por el código fuente, que podría terminar la implementación de nuevas características, escribir extensiones, o simplemente corrigiendo errores! Es una base de código muchos desarrolladores se han inspirado.

¡Gracias por leer!