**Librerías de desarrollo de la interfaz**

Ahora que estás familiarizado con HTML, CSS y JavaScript, mejora tus habilidades aprendiendo algunas de las librerías de interfaz más populares en la industria.

En la certificación de bibliotecas de desarrollo front-end, aprenderá cómo diseñar su sitio rápidamente con Bootstrap. También aprenderá cómo agregar lógica a sus estilos CSS y extenderlos con Sass.

Después, construirás un carrito de compras y otras aplicaciones para aprender a crear poderosas aplicaciones de página única (SPAs) con React y Redux.

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

**Bootstrap**

Bootstrap es un framework front end que se utiliza para diseñar páginas y aplicaciones web adaptativas. Tiene un enfoque de desarrollo web e incluye estilos y clases CSS predefinidos, además de algunas funcionalidades de JavaScript.

En este curso, aprenderás cómo crear sitios web adaptativos con Bootstrap, y utilizarás sus clases incluidas para estilizar botones, imágenes, formularios, navegación y otros elementos.

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

Paso 1 - Utiliza diseño adaptable con contenedores de fluido de Bootstrap

En la seccion de HTML5 y CSS de freeCodeCamp construimos un 'Cat Photo App'. Ahora volvamos a él. Esta vez, vamos a estilizarlo utilizando el popular framework CSS adaptable de Bootstrap.

Bootstrap averiguará cuan ancha es tu pantalla y responderá redimensionando tus elementos HTML - de ahí el nombre diseño adaptable.

Con diseño adaptable, no hay necesidad de diseñar una versión móvil de tu sitio web. Se verá bien en dispositivos con pantallas de cualquier ancho.

Puedes añadir Bootstrap a cualquier aplicación agregando el siguiente código en la parte superior de tu HTML:

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous"/>

Paso 2 - Haz imágenes adaptables a dispositivos móviles

Primero, agrega una nueva imagen debajo de la existente. Establece su atributo src a https://cdn.freecodecamp.org/curriculum/cat-photo-app/running-cats.jpg.

Sería genial si esta imagen pudiera tener exactamente el ancho de la pantalla de nuestro teléfono.

Afortunadamente, con Bootstrap, todo lo que necesitamos hacer es agregar la clase img-responsive a nuestra imagen. Hazlo, y la imagen debería caber perfectamente en el ancho de tu página.

Paso 3 - Centrar texto con Bootstrap

Ahora que estamos usando Bootstrap, podemos centrar nuestro encabezado para que luzca mejor. Lo único que necesitamos hacer es agregar la clase text-center a nuestro elemento h2.

Recuerda que puedes agregar varias clases al mismo elemento separando cada una de ellas con un espacio, de la siguiente manera:

<h2 class="red-text text-center">your text</h2>

Paso 4 - Crea un botón de Bootstrap

Bootstrap tiene sus propios estilos para los elementos button, que se ven mucho mejor que los de HTML puro.

Crea un nuevo elemento button debajo de la foto grande de tu gatito. Dale las clases btn y btn-default, así como el texto de Like.

Paso 5 - Crea un botón Bootstrap como elemento de bloque

Normalmente, tus elementos button con las clases btn y btn-default son tan anchos como el texto que los contiene. Por ejemplo:

<button class="btn btn-default">Submit</button>

Este botón será tan ancho como la palabra Submit.

Al convertirlos en elementos de bloque con la clase adicional btn-block, tu botón se estirará para llenar completamente el espacio horizontal de tu página y cualquier elemento siguiente se posicionará en una "nueva línea" debajo del bloque.

<button class="btn btn-default btn-block">Submit</button>

Este botón tomará el 100% del ancho disponible.

Ten en cuenta que estos botones aún necesitan la clase btn.

Agrega la clase de Bootstrap btn-block a tu botón de Bootstrap.

Paso 6 - Prueba el arcoíris de colores del botón de Bootstrap

La clase btn-primary es el color principal que usarás en tu aplicación. Es útil para resaltar acciones que deseas que tu usuario realice.

Reemplaza la clase btn-default de Bootstrap por btn-primary en tu botón.

Ten en cuenta que este botón aún necesitará las clases btn y btn-block.

Paso 7 - Llama a acciones opcionales con btn-info

Bootstrap viene con varios colores predefinidos para los botones. La clase btn-info se utiliza para llamar la atención sobre las acciones opcionales que el usuario puede realizar.

Crea un nuevo botón de Bootstrap a nivel de bloque debajo de tu botón de Like con el texto Info, y agrégale la clase de Boostrap btn-info.

Ten en cuenta que estos botones todavía necesitan las clases btn y btn-block.

Paso 8 - Advierte a tus usuarios de una acción peligrosa con btn-danger

Bootstrap viene con varios colores predefinidos para botones. La clase btn-danger es el color del botón que usarás para notificar a los usuarios que el botón realiza una acción destructiva, como eliminar la foto de un gato.

Crea un botón con el texto Delete y dale la clase btn-danger.

Ten en cuenta que estos botones todavía necesitan las clases btn y btn-block.

Paso 9 - Utiliza la cuadrilla (grid) Bootstrap para poner elemento de lado a lado

Bootstrap utiliza un sistema adaptable de cuadrilla de 12 columnas, el cual hace que sea fácil poner elementos en dos filas y especificar la anchura relativa de cada elemento. La mayoría de las clases de Bootstrap pueden ser aplicadas a un elemento div.

Bootstrap tiene diferentes atributos de ancho de columna que usa dependiendo cuan ancha es la pantalla del usuario. Por ejemplo, los teléfonos tienen pantallas más angostas y las laptops tienen pantallas más anchas.

Tomemos por ejemplo la clase de Bootstrap col-md-\*. Aquí, md significa mediano y \* es un número que especifica cuántas columnas de ancho deben tener los elementos. En este caso, el ancho de columna de un elemento en una pantalla de tamaño mediano, como una laptop, está siendo especificado.

En el 'Cat Photo App' que estamos creando, vamos a usar col-xs-\*, donde xs significa extra pequeño ("extra small" como una pantalla de teléfono), y \* es el número de columnas que va a tomar el elemento en anchura.

Pon los botones Like, Info y Delete lado a lado, anidando los tres dentro de un elemento <div class="row">, luego, cada uno de ellos dentro de un elemento <div class="col-xs-4">.

La clase row es aplicada a un div, y los botones mismos pueden ser anidados dentro de él.

Paso 10 - Adiós CSS Personalizado. Hola Bootstrap

Podemos limpiar nuestro código y hacer que nuestro 'Cat Photo App' se vea más convencional utilizando los estilos incorporados de Bootstrap en lugar de los estilos personalizados que creamos antes.

No te preocupes - habrá mucho tiempo para personalizar nuestro CSS después.

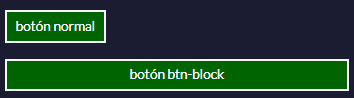
Borra las declaraciones de CSS .red-text, p y .smaller-image de tu elemento style para que las únicas declaraciones en tu elemento style sean h2 y thick-green-border.

Luego, borra el elemento p que contiene un enlace muerto. Después, borra la clase red-text de tu elemento h2 y reemplázalo con la clase de Bootstrap text-primary.

Finalmente, remueve la clase smaller-image de tu primer elemento img y reemplázalo con la clase img-responsive.

Paso 11 - Usa un span para apuntar a elementos en línea

Puedes usar spans para crear elementos en línea. ¿Recuerdas cuando usamos la clase btn-block para hacer que el botón ocupe toda la fila completa?



Eso ilustra la diferencia entre un elemento "inline" y un elemento "block".

Usando el elemento en línea span, puedes alinear varios elementos, e incluso estilizar diferentes partes de la misma línea de manera diferente.

Usando un elemento span, anida la palabra love dentro del elemento p que actualmente tiene el texto Things cats love. Luego dale a span la clase text-danger para cambiar el texto a rojo.

Así es como harías esto para el elemento p que tiene el texto Top 3 things cats hate:

<p>Top 3 things cats <span class="text-danger">hate:</span></p>

Paso 12

Cree otro elemento p debajo de su elemento .first-paragraph y asígnele el siguiente texto:

After years - years - of pondering these two problems and how to solve them, I slipped, hit my head on the sink, and when I came to I had a revelation! A vision! A picture in my head! A picture of this! This is what makes time travel possible: the flux capacitor!

Paso 13

Agregue un tercer elemento p al final de su elemento .text y asígnele el siguiente texto:

It wasn't as dramatic as Doc's revelation in Back to the Future. It just occurred to me while I was going for a run. The revelation: the entire curriculum should be a series of projects. Instead of individual coding challenges, we'll just have projects, each with their own seamless series of tests. Each test gives you just enough information to figure out how to get it to pass. (And you can view hints if that isn't enough.)

Paso 14

Después de los tres elementos p dentro de su elemento .text, cree un elemento blockquote. Dentro de eso, agregue un elemento hr, un elemento p con class establecido en quote, y un segundo elemento hr.

Asigne al elemento .quote el texto The entire curriculum should be a series of projects.

Paso 15

Debajo de su elemento blockquote, agregue otro elemento p con el siguiente texto:

No more walls of explanatory text. No more walls of tests. Just one test at a time, as you build up a working project. Over the course of passing thousands of tests, you build up projects and your own understanding of coding fundamentals. There is no transition between lessons and projects, because the lessons themselves are baked into projects. And there's plenty of repetition to help you retain everything because - hey - building projects in real life has plenty of repetition.

Paso 16

Cree un quinto elemento p al final de su elemento .text y asígnele el siguiente texto:

The main design challenge is taking what is currently paragraphs of explanation and instructions and packing them into a single test description text. Each project will involve dozens of tests like this. People will be coding the entire time, rather than switching back and forth from "reading mode" to "coding mode".

Paso 17

Cree un elemento final p al final de su elemento .text y asígnele el siguiente texto:

Instead of a series of coding challenges, people will be in their code editor passing one test after another, quickly building up a project. People will get into a real flow state, similar to what they experience when they build the required projects at the end of each certification. They'll get that sense of forward progress right from the beginning. And freeCodeCamp will be a much smoother experience.

Paso 18

Debajo de su elemento .text, cree un nuevo elemento section y asígnele una class de text text-with-images. Dentro de eso, cree un elemento article con una class establecida en brief-history, y un elemento aside con el class establecido en image-wrapper.

Paso 19

Dentro de su elemento article, cree un elemento h3 con class establecido en list-title y el texto de A Brief History. Debajo de eso, crea un elemento p con el texto Of the Curriculum. Luego crea un elemento ul con la clase lists.

Paso 20

Dentro de su elemento ul, cree seis elementos li. Agrega un elemento h4 con una class establecida en list-subtitle y un elemento p a cada uno de tus elementos li.

Luego asigne a los elementos h4 y p el siguiente contenido de texto, en orden, con cada h4 usando lo que está en el lado izquierdo de los dos puntos, y cada p usando lo que está a la derecha:

* V1 - 2014: We launched freeCodeCamp with a simple list of 15 resources, including Harvard's CS50 and Stanford's Database Class.
* V2 - 2015: We added interactive algorithm challenges.
* V3 - 2015: We added our own HTML+CSS challenges (before we'd been relying on General Assembly's Dash course for these).
* V4 - 2016: We expanded the curriculum to 3 certifications, including Front End, Back End, and Data Visualization. They each had 10 required projects, but only the Front End section had its own challenges. For the other certs, we were still using external resources like Node School.
* V5 - 2017: We added the back end and data visualization challenges.
* V6 - 2018: We launched 6 new certifications to replace our old ones. This was the biggest curriculum improvement to date.

Paso 21

Dentro de su elemento aside, crear dos elementos img un elemento blockquote, y un tercer elemento img. Asigne al elemento blockquote una class establecida en image-quote.

Paso 22

Dentro del elemento.image-wrapper, asigne a su primer elemento img un src de https://cdn.freecodecamp.org/testable-projects-fcc/images/random-quote-machine.png, una alt de image of the quote machine project, una class de image-1, un atributo loading establecido en lazy, un atributo width de 600 y un atributo height de 400.

Paso 23

Dentro de su elemento .image-wrapper, asigne al segundo elemento img un src dehttps://cdn.freecodecamp.org/testable-projects-fcc/images/calc.png, un alt de image of a calculator project, un conjunto de atributos loading a lazy, una class establecida en image-2, un atributo width establecido en 400, y un atributo height establecido en 400.

Paso 24

Dentro de su elemento .image-wrapper, asigne a su tercer elemento img un src de https://cdn.freecodecamp.org/testable-projects-fcc/images/survey-form-background.jpeg, un alt de four people working on code, a loading attribute of , un loading atributo de lazy, una class establecida en image-3, un atributo width establecido en 600 y un atributo height establecido en 400.

Paso 25

Dentro de su elemento .image-quote, anide un elemento hr, un elemento p y un segundo elemento hr. Asigne al elemento p una class establecida en quote y el texto The millions of people who are learning to code through freeCodeCamp will have an even better resource to help them learn these fundamentals.

Paso 26

Iniciamos con tu CSS, normalize the CSS rules by targeting all elements with \*, including the ::before and ::after pseudo-selectors.

Set the padding and margin properties both to 0 and set the box-sizing property to border-box.

Paso 27

Cree un selector html y asígnele una propiedad font-size establecida en 62.5%. Esto establecerá el tamaño de fuente predeterminado para su página web en 10 px (el valor predeterminado del navegador es 16 px).

Esto te facilitará trabajar con unidades rem más adelante, ya que 2rem serían 20 píxeles.

Paso 28

Cree un selector body. Establezca la propiedad font-family en Baskervville, con una alternativa de serif. Establezca la propiedad color en linen y la propiedad background-color en rgb(20, 30, 40).

Paso 29

Cree un selector h1 y establezca la propiedad font-family en Anton con el respaldo de sans-serif.

Paso 30

Cree un selector h2, h3, h4, h5, h6. Dale una propiedad font-family establecida en Raleway con un respaldo de sans-serif.

Paso 31

Cree un selector a y asígnele una propiedad text-decoration establecida en none y una propiedad color establecida en linen.

Paso 32

Ahora está listo para comenzar a armar el diseño de la cuadrícula. CSS Grid ofrece un diseño bidimensional basado en cuadrículas, lo que le permite centrar elementos horizontal y verticalmente mientras conserva el control para hacer cosas como elementos superpuestos.

Comience creando un selector main y asignándole una propiedad display establecida en grid.

Paso 33

Ahora puede diseñar el diseño de su cuadrícula. CSS Grid es similar a Flexbox en que tiene una propiedad especial para los elementos padre e hijo.

En este caso, su elemento principal es el elemento main. Configure el contenido para que tenga un diseño de tres columnas agregando una propiedad grid-template-columns con un valor de 1fr 94rem 1fr. Esto creará tres columnas donde la columna del medio tiene un ancho de 94rem, y la primera y la última columna son ambas 1 fracción del espacio restante en el contenedor de cuadrícula.

Paso 34

Utiliza la función minmax para hacer tus columnas adaptables en cualquier dispositivo. La función minmax toma dos argumentos, siendo el primero el valor mínimo y el segundo el máximo. Estos valores pueden ser una longitud, un porcentaje, fr o incluso una palabra clave como max-content.

Envuelve cada uno de tus valores ya definidos de la propiedad grid-template-columns en una función minmax, usando cada valor como segundo argumento. El primer argumento debe ser 2rem, min-content y 2rem respectivamente.

Paso 35

Para agregar espacio entre filas en el diseño de cuadrícula, puede usar la propiedad row-gap. Dale al selector main una propiedad row-gap de 3rem.

Paso 36

Su revista tendrá tres secciones principales. Ya configuró el diseño general en la regla main, pero puede ajustar la ubicación en las reglas secundarias.

Una opción es la propiedad grid-column, que es la abreviatura de grid-column-start y grid-column-end. La propiedad grid-column le dice al elemento de la cuadrícula en qué línea de cuadrícula debe comenzar y terminar.

Cree una regla .heading y establezca la propiedad grid-column en 2 / 3. Esto le indicará al elemento .heading que comience en la línea de cuadrícula 2 y finalice en la línea de cuadrícula 3.

Paso 37

Cree un selector .text y asígnele una propiedad grid-column establecida en 2 / 3.

Paso 38

Para un control adicional sobre el diseño de su contenido, puede tener una cuadrícula CSS dentro de una cuadrícula CSS.

Establezca la propiedad display de su selector .heading en grid.

Paso 39

Ahora puede diseñar el contenido del elemento .heading con CSS Grid.

La función CSS repeat() se usa para repetir un valor, en lugar de escribirlo manualmente, y es útil para los diseños de cuadrícula. Por ejemplo, establecer la propiedad grid-template-columns en repeat(20, 200px) crearía 20 columnas cada una de 200px de ancho.

Dale a tu elemento .heading una propiedad grid-template-columns establecida en repeat(2, 1fr) para crear dos columnas de igual ancho.

Paso 40

Dale a tu selector .heading una propiedad row-gap establecida en 1.5rem.

Paso 41

Recuerde que la propiedad grid-column determina en qué columnas comienza y termina un elemento. Puede haber momentos en los que no esté seguro de cuántas columnas tendrá su cuadrícula, pero desea que un elemento se detenga en la última columna. Para hacer esto, puede usar -1 para la columna final.

Cree un selector .hero y asígnele una propiedad grid-column establecida en 1 / -1. Esto le indicará al elemento que abarque todo el ancho de la cuadrícula.

Paso 42

Dale al selector .hero una propiedad position establecida en relative.

Paso 43

Debe eliminar el atributo temporal width antes de escribir el CSS para su .hero-img.

Paso 44

Cree un selector img y asígnele una propiedad width establecida en 100%, y un conjunto de propiedades object-fit para cover.

La propiedad object-fit le dice al navegador cómo colocar el elemento dentro de su contenedor. En este caso, cover configurará la imagen para llenar el contenedor, recortándola según sea necesario para evitar cambiar la relación de aspecto.

Paso 45

Cree un selector .hero-title y asígnele una propiedad text-align establecida en center, una propiedad color establecido en orangered y una propiedad font-size establecida en 8rem.

Paso 46

El subtítulo también necesita ser diseñado. Cree un selector .hero-subtitle y asígnele una propiedad font-size establecida en 2.4rem, un color propiedad establecida en orangered, y una propiedad text-align establecida en center.

Paso 47

Cree un selector .author y asígnele una propiedad font-size establecida en 2rem y una propiedad font-family establecido en Raleway con un respaldo de sans-serif.

Paso 48

Cree un selector .author-name a:hover y asígnele una propiedad background-color establecida en #306203.

Esto creará un efecto de desplazamiento solo para el elemento a dentro de .author-name, mostrando el freeCodeCamp verde original en el fondo.

Paso 49

Cree un selector .publish-date y asígnele una propiedad color de rgba(255, 255, 255, 0.5).

Paso 50

Cree un selector .social-icons. Dale una propiedad display establecida en grid y una propiedad font-size establecida en 3rem.

Paso 51

La configuración predeterminada para CSS Grid creará filas adicionales según sea necesario, a diferencia de Flexbox. Asigne al selector .social-icons una propiedad grid-template-columns establecida en repeat(5, 1fr) para organizar los íconos en una sola fila.

Paso 52

Si desea agregar más íconos sociales, pero mantenerlos en la misma fila, deberá actualizar grid-template-columns para crear columnas adicionales. Como alternativa, puede utilizar la propiedad grid-auto-flow.

Esta propiedad toma row o column como primer valor, con un segundo valor opcional de dense. grid-auto-flow utiliza un algoritmo de colocación automática para ajustar el diseño de la cuadrícula. Establecerlo en column le indicará al algoritmo que cree nuevas columnas para el contenido según sea necesario. El valor dense permite que el algoritmo retroceda y rellene los huecos en la cuadrícula con elementos más pequeños, lo que puede provocar que los elementos aparezcan desordenados.

Para su selector .social-icons, establezca la propiedad grid-auto-flow en column.

Paso 53

Ahora, el algoritmo de colocación automática se activará cuando agregue un nuevo elemento de icono. Sin embargo, el algoritmo establece por defecto que el ancho de la nueva columna sea auto, que no coincidirá con sus columnas actuales.

Puede anular esto con la propiedad grid-auto-columns. Asigne al selector .social-icons una propiedad grid-auto-columns establecida en 1fr.

Paso 54

Al igual que Flexbox, con CSS Grid puedes alinear el contenido de los elementos de la cuadrícula con align-items y justify-items. align-items alineará los elementos secundarios a lo largo del eje de la columna, y justify-items alineará los elementos secundarios a lo largo del eje de la fila.

Asigne al selector .social-icons una propiedad align-items establecida en center.

Paso 55

Asigne al selector .text una propiedad font-size establecida en 1.8rem y una propiedad letter-spacing establecida en 0.6px.

Paso 56

Su elemento .text no es una cuadrícula CSS, pero puede crear columnas dentro de un elemento sin usar Grid usando la propiedad column-width.

Dale a tu selector .text una propiedad column-width establecida en 25rem.

Paso 57

Las revistas a menudo usan texto justificado en su contenido impreso para estructurar su diseño y controlar el flujo de su contenido. Si bien esto funciona en forma impresa, el texto justificado en los sitios web puede ser un problema de accesibilidad, por ejemplo, presenta desafíos para las personas con dislexia.

Para hacer que su proyecto parezca una revista impresa, asigne al selector .text una propiedad text-align establecida en justify.

Paso 58

El pseudo-selector ::first-letter le permite apuntar a la primera letra en el contenido de texto de un elemento.

Cree un selector .first-paragraph::first-letter y establezca la propiedad font-size en 6rem. También asígnele una propiedad color establecida en orangered para que se destaque.

Paso 59

El otro texto se ha desplazado fuera de lugar. Muévalo a su posición dándole al selector .first-paragraph::first-letter una propiedad float establecida en left y un margin-right establecida en 1rem.

Paso 60

Cree un selector hr y asígnele una propiedad margin establecida en 1.5rem 0.

Paso 61

Para darle un color a hr, debe ajustar la propiedad border. Dale al selector hr una propiedad border establecida en 1px solid rgba(120, 120, 120, 0.6).

Paso 62

Cree un selector .quote. Asígnele una propiedad color establecida en #00beef, una propiedad font-size establecida en 2.4rem y una Propiedad text-align establecida en center.

Paso 63

Para que el texto de la cita se destaque más, asigne al selector .quote una propiedad font-family establecida en Raleway con un respaldo de sans-serif.

Paso 64

Una cita no es realmente una cita sin las comillas adecuadas. Puede agregarlos con pseudo-selectores de CSS.

Cree un selector .quote::before y establezca la propiedad content en " con un espacio a continuación.

Además, cree un selector .quote::after y establezca la propiedad content en " con un espacio antes.

Paso 65

Ahora es el momento de diseñar tu tercera section. Tenga en cuenta que tiene los valores text y text-with-images para el atributo class, lo que significa que ya hereda los estilos de su regla .text.

Cree un selector .text-with-images y establezca la propiedad display en grid.

Paso 66

Deberá tener una columna para el texto y una columna para las imágenes. Asigne al selector .text-with-images una propiedad grid-template-columns establecida en 1fr 2fr. También establezca la propiedad column-gap en 3rem para proporcionar más espacio entre las columnas.

Paso 67

Dale al selector .text-with-images una propiedad margin-bottom establecida en 3rem.

Paso 68

Cree un selector .lists y establezca la propiedad list-style-type en none. Esto eliminará las viñetas en los elementos de la lista.

Paso 69

Asigne al selector .lists una propiedad margin-top establecida en 2rem.

Paso 70

Cree una regla .lists li para apuntar a los elementos de la lista dentro de su elemento .lists. Dale una propiedad margin-bottom establecida en 1.5rem.

Paso 71

Cree un selector .list-title, .list-subtitle y establezca la propiedad color en #00beef.

Paso 72

Es hora de diseñar la última sección de la revista: las imágenes.

Las imágenes se envuelven con un elemento aside utilizando la clase image-wrapper, así que crea un selector .image-wrapper. Establezca la propiedad display en grid.

Paso 73

Las imágenes deben estar dentro de un diseño de dos columnas y tres filas.

Asigne al selector .image-wrapper una propiedad grid-template-columns establecida en 2fr 1fr y grid-template-rows propiedad establecida en repeat(3, min-content). Esto le dará a nuestra cuadrícula filas que se ajustan en altura según el contenido, pero columnas que mantienen un ancho fijo según el contenedor.

Paso 74

La propiedad gap es una forma abreviada de establecer el valor de column-gap y row-gap al mismo tiempo. Si se le da un valor, establece column-gap y row-gap en ese valor. Si se le dan dos valores, establece el row-gap en el primer valor y el column-gap en el segundo.

Dale al selector .image-wrapper una propiedad gap establecida en 2rem.

Paso 75

La propiedad place-items se puede usar para establecer los valores align-items y justify-items al mismo tiempo. La propiedad place-items toma uno o dos valores. Si se proporciona un valor, se utiliza para las propiedades align-items y justify-items. Si se proporcionan dos valores, el primer valor se usa para la propiedad align-items y el segundo valor se usa para la propiedad justify-items.

Asigne al selector .image-wrapper una propiedad place-items establecida en center.

Paso 76

Cree una regla .image-1, .image-3 y asígnele una propiedad grid-column establecida en 1 / -1. Esto permitirá que la primera y la tercera imagen abarquen todo el ancho de la cuadrícula.

Paso 77

Ahora que el diseño de la revista está terminado, debe hacerlo adaptable.

Comience con una consulta @media para only screen con un max-width de 720px. Dentro, crea un selector .image-wrapper y dale una propiedad grid-template-columns de 1fr.

Esto colapsará las tres imágenes en una columna en pantallas más pequeñas.

Paso 78

Crear otro @media consulta only screen con un max-width de 600px. Dentro, cree una regla .text-with-images y asígnele una propiedad grid-template-columns de 1fr.

Esto colapsará el área de texto inferior en una sola columna en pantallas más pequeñas.

Paso 79

Cree una tercera consulta @media para only screen con un max-width de 550px. Dentro, crea un selector .hero-title con un font-size establecido en 6rem, un .hero-subtitle, .author, .quote, .list-title selector con un font-size establecido en 1.8rem, un selector .social-icons con un font-size establecido en 2rem, y un selector .text con un font-size establecido en 1.6rem.

Paso 80

Cree una consulta final de @media para only screen con un max-width de 420px. Dentro, cree un selector .hero-title con una propiedad font-size establecida en 4.5rem.

¡Felicidades! Su revista ahora está completa.