**El esquema para cada navegación (HTML)**

El primer paso es (claro) el scaffold HTML. La navegación tienen su propia etiqueta semántica en HTML5: el elemento <nav>. Cuando construyas una navbar, empieza con esto:

<nav>

<!-- Here comes the navigation. -->

</nav>

HTML

Copy

Genial, ahora puedes agregar los enlaces de navegación. A continuación, te explicaré dos formas para hacerlo.

Opción A: puedes empezar y agregar una lista desorganizada con etiquetas a (pues técnicamente es una lista de elementos de navegación):

<nav>

<ul>

<li><a href="#">Link 1</a></li>

<li><a href="#">Link 2</a></li>

<li><a href="#">Link 3</a></li>

<li><a href="#">Link 4</a></li>

</ul>

</nav>

Nota: En este caso, acuérdate de quitarle el aspecto predeterminado a esta lista (es decir, eliminar las viñetas, mostrarla en línea, etc.)

Opción B: Por el contrario, puedes ahorrarte las molestias que traen estas listas; los elementos de anclaje estarán bien por sí solos:

<nav>

<a href="#">Link 1</a>

<a href="#">Link 2</a>

<a href="#">Link 3</a>

<a href="#">Link 4</a>

</nav>

HTML

Copy

Ningún método es mejor que otro, ambos son igualmente válidos. La opción de la lista es un poco más popular porque se cree que es mejor para los lectores de pantalla, pero la evidencia no es definitiva. Al final, es algo que no tiene mucha importancia. Desarrollaremos los componentes de ambas formas para que veas la diferencia y luego podrás elegir la que más te guste, o lo que funcione mejor para tu proyecto.

Qué bonito, ¿verdad? Abre un bin nuevo y comienza agregando el scaffold de HTML:

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">News</a></li>

<li><a href="#">Blog</a></li>

<li><a href="#">About us</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

Hasta ahora, todo muy bien. Tenemos una lista sencilla con elementos de anclaje dentro de los elementos de la misma. Sin embargo, esto me recuerda algo: no podemos olvidar el CSS reset y el cambio al modelo border-box. Agreguemos primero el reset. Incluye este enlace en la sección head:

<head>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.css">

</head>

HTML

Copy

Consejo pro: Esto vincula el reset simple de Eric Meyer desde una fuente externa para que no tengamos que agregarlo manualmente a la parte superior de nuestro CSS.

El reset está en su lugar, ahora cambiemos el modelo box-sizing. Agrega este bloque a la parte superior de tu CSS:

/\* ==== GENERAL STYLES ==== \*/

\* {

box-sizing: border-box;

}

CSS

Copy

Bien, la configuración ya está lista; podemos cambiar al CSS.

**Estilos de base**

Como estamos trabajando con el método de la lista, lo primero sería anular cualquier estilo predeterminado (como las viñetas y demás). Pero el reset de Meyer ya se encargó de eso, así que podemos omitirlo. Tampoco tenemos que cambiar la posición del nav, ya que estamos construyendo para dispositivos móviles, y esta configuración de apilación vertical es perfecta para eso.

Genial, esto significa que ya podemos saltar directo al paso dos y diseñar los elementos de navegación. Comencemos con los estilos móviles predeterminados.

**<ul>**

Comencemos con la etiqueta ul:

/\* ==== GENERAL STYLES ==== \*/

\* {

box-sizing: border-box;

}

/\* ==== A SIMPLE TOP NAVIGATION ==== \*/

nav ul {

background-color: #eee;

text-align: center;

}

CSS

Copy

+1 consejo: Observa cómo el fondo gris ocupa todo el ancho de la ventana. ¿Recuerdas por qué? Porque ul es, por defecto, un elemento en bloque.

Bueno, avancemos un poco más para estilizar los elementos de la lista.

**<li>**

Agrega este bloque debajo de las reglas anteriores:

nav li {

font-family: 'Anton', sans-serif;

font-size: 1.4rem;

line-height: 3rem;

height: 3rem;

border-bottom: 1px solid #ccc;

}

CSS

Copy

El menú de navegación comienza a tomar forma. Sin embargo, apareció un pequeño error: el código hace referencia a una fuente externa que no hemos vinculado hasta ahora. Arreglémoslo copiando esta línea en la sección head:

<link href="https://fonts.googleapis.com/css?family=Anton" rel="stylesheet">

HTML

Copy

Sucedieron un par de cosas interesantes en este último conjunto de reglas. Las propiedades line-height y height están configuradas intencionalmente con el mismo valor. Este pequeño truco de CSS centra verticalmente el texto dentro de los elementos. Mientras tanto, se utiliza la propiedad border-bottom para crear una separación visual entre los elementos.

Terminemos los estilos base con etiquetas de anclaje.

**<a>**

Copia este código al final de tu CSS:

nav a {

text-decoration: none;

color: #C74F78;

display: block;

}

El text-decoration: none; le quita a los enlaces su apariencia azul y subrayada, de esta manera podemos declarar un nuevo color en la siguiente línea. Sin embargo, la regla más importante es la display: block;. Gracias a ella, las anclas ahora ocupan todo el espacio dentro de las etiquetas li. Esto amplía el área para hacer clic a todo el ancho y alto del botón, lo que hace que sea más fácil darle clic a un vínculo de navegación.

Con esto terminamos los estilos base. Ahora continuemos con los estados especiales.

**Estados especiales**

Todo buen enlace necesita de un hover state:

nav a:hover {

background-color: #fff;

}

Y algunas animaciones de transición bonitas:

nav a:hover {

background-color: #fff;

transition: background-color .3s ease-in-out;

}

Es una práctica común en los sitios de páginas múltiples marcar la pestaña que está actualmente activa en la barra de navegación. Digamos que ahora mismo estamos construyendo la página de inicio. Eso hace que el enlace "Inicio" (o home, en inglés) sea el activo. Añade entonces una clase .active al elemento ancla correspondiente en tu HTML:

<nav>

<ul>

<li><a href="#" class="active">Home</a></li>

<li><a href="#">News</a></li>

<li><a href="#">Blog</a></li>

<li><a href="#">About us</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

Y añade este al final de tu CSS:

nav a.active {

background-color: #c74f78;

color: #eee;

}

**Estilos de escritorio**

Básicamente, la versión para dispositivos móviles de nuestra barra de navegación ya está lista, pero esos botones anchísimos que se encuentran apilados de manera vertical se ven realmente ridículos en una tablet o computadora. Afortunadamente, ya sabes cómo funcionan los media queries. Vamos a usar uno para transformar nuestro navegador en una barra horizontal.

Primero, iniciemos un media query y definamos un punto de ruptura:

@media (min-width: 600px) {

}

En segundo lugar, vamos a dirigirnos a los elementos de la lista y establecer el valor de display a inline-block:

@media (min-width: 600px) {

nav li {

display: inline-block;

}

}

Bueno, ya estamos más cerca, pero ahora las cosas se ven un poco atiborradas y los elementos tienen bordes inferiores innecesarios. Arreglémoslos:

@media (min-width: 600px) {

nav li {

display: inline-block;

border-bottom: none;

width: 7.5rem;

}

}

Casi perfecto. Sólo tenemos que deshacernos de los pequeños espacios en blanco entre los enlaces del nav que trajo el valor inline-block del display. La forma más sencilla para lidiar con esto es estableciendo el tamaño de fuente de todo el nav a cero píxeles. De esta forma, los espacios en blanco intermedios no se mostrarán. Agrega este bloque a la parte superior de tu CSS de navegación:

nav {

font-size: 0;

}

**Una navegación superior fija - Una posible solución**

¿Cómo te fue? ¿Fue fácil? ¿Difícil? De cualquier modo, buen trabajo. Echemos un vistazo a una posible solución.

**El scaffold HTML.**

Añadiré un simple elemento nav con algunas etiquetas a para empezar:

<nav>

<a href="#">Sign Up</a>

<a href="#">Sign In</a>

<a href="#">About</a>

<a href="#">Home</a>

</nav>

HTML

Copy

Nota: Hay algunas etiquetas p de muestra en el código HTML incrustado. Las agregaré luego, cuando termine con el nav.

**Reset y box-sizing**

Llegó la hora de enlazar el reset de Meyer:

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.css">

HTML

Copy

Y de cambiar box-sizing a border-box:

\* {

box-sizing: border-box;

}

CSS

Copy

**Estilos de base**

Estoy siguiendo el enfoque del mobile-first, lo cual significa que no tendré que preocuparme por el posicionamiento todavía. Como el comportamiento predeterminado es vertical y no es fijo, puedo seguir y estilizar los elementos.

Comencemos con una etiqueta nav:

nav {

background-color: rgba(0,0,0,.9);

padding: 1rem;

}

CSS

Copy

Elegí un color ligeramente transparente para un destello visual y el relleno está ahí para dar un poco de espacio a los elementos.

Continuemos con las anclas:

nav a {

text-decoration: none;

font-family: 'Space Mono', monospace;

text-align: center;

color: #fff;

display: block;

padding: .5rem 1rem;

}

CSS

Copy

Los elementos más importantes aquí son la regla display: block; y el relleno asimétrico. El primero expande el área para hacer clic y permite a las etiquetas a aceptar las propiedades del modelo de cajas. El segundo te ayudará a crear un aspecto alargado para el botón de "Sign Up".

Utilicé una fuente externa; vamos enlazarla en la sección head:

<link href="https://fonts.googleapis.com/css?family=Space+Mono:400,700" rel="stylesheet">

HTML

Copy

Casi terminamos con lo básico; sólo me falta un borde para mi botón "Sign Up". Me dirigiré hacia él con una clase:

<nav>

<a href="#" class="sign-up">Sign Up</a>

<a href="#">Sign In</a>

<a href="#">About</a>

<a href="#">Home</a>

</nav>

HTML

Copy

Entonces agregaré este fragmento a mi CSS:

.sign-up {

border: 1px solid hsla(0,0%,100%,.2);

border-radius: 9999px;

}

CSS

Copy

¡Bien! Ahora vamos a los estados especiales.

**Hover States**

Quiero que los enlaces cambien a un gris sutil cuando se les pase el puntero por encima, así que voy a agregar estas reglas debajo de los estilos de anclaje base:

nav a:hover {

color: hsl(0, 0%, 77%);

transition: color .15s ease-in-out;

}

CSS

Copy

También quiero que se inviertan los colores del botón "Sign Up" cuando se le pase el cursor por encima:

.sign-up:hover {

color: #000;

background-color: #fff;

transition: background-color .15s ease-in-out;

}

CSS

Copy

Y con esto, los estilos móviles ya están listos.

**Estilos para tablet y computadora**

Para las pantallas más grandes, quiero que la barra de navegación se expanda, se vuelva horizontal y se mantenga fija en la parte superior. Primero, vayamos a la parte de "fijarla":

@media (min-width: 768px) {

nav {

position: fixed;

top: 0;

left: 0;

width: 100%;

}

}

CSS

Copy

Ahora sólo tengo que hacer flotar los elementos para ponerlos en posición horizontal y en el orden correcto. En el ejemplo, el enlace "Home" está a la izquierda mientras que los otros están a la derecha. Puedo lograr esto con flotar primero cada ancla hacia la derecha y luego "Home" hacia la izquierda:

@media (min-width: 768px) {

nav a {

float: right;

}

.home {

float: left;

}

}

CSS

Copy

Nota: Por supuesto que esto necesita de una clase .home en el elemento anclaje correspondiente del HTML.

Y finalmente, en donde haya floats, también habrán clearfixes:

@media (min-width: 768px) {

/\* Clearfix \*/

nav::after {

content: "";

clear: both;

display: block;

}

}

CSS

Copy

Voilá, ya tenemos una navegación superior, hermosa y fija. :)

**Opcional: ejemplos de párrafos**

Si deseas ver el navegador en acción, agrega algunos párrafos de texto y copia este código a tu CSS:

p {

font-size: 1rem;

font-family: 'Space Mono', monospace;

}

@media (min-width: 768px) {

body {

width: 60%;

margin: 5.25rem auto 1.25rem;

}

}

p {

margin-bottom: 1.25rem;

padding: .5rem;

}

CSS

Copy

Consejo pro: Fíjate en el margen superior en el body. Esto es lo que previene que el contenido se deslice debajo de la barra de navegación.