**Aprende diseño web adaptativo construyendo un piano**

El diseño adaptativo indica a tu página web cómo debe verse en pantallas de distintos tamaños.

En este curso, usarás CSS y Diseño Adaptativo para programar un piano. También aprenderás más sobre las consultas de medios y pseudoselectores.

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

Paso 1

Empieza con la estructura básica de HTML. Agrega una declaracion DOCTYPE y los elementos html, head, body y title.

Establezca el idioma de esta página a inglés. Establece el title a Piano.

Paso 2

Agrega dos etiquetas meta, una para optimizar tu página para dispositivos móviles y una para especificar el charset aceptado de la página.

Paso 3

Es hora de empezar a trabajar en el piano. Crea un elemento div dentro de tu elemento body con el id establecido en piano.

Paso 4

Anida un segundo div dentro de tu div existente y establece que el class sea keys.

Paso 5

Dentro de tu elemento .keys, agrega siete elementos div. Dales a todos la clase key.

Paso 6

Recuerda que un atributo class puede tener varios valores. Para separar tus teclas blancas de tus teclas negras, debes agregar un segundo valor al class, el valor es black--key. Agrega esto a tu segundo, tercer, quinto, sexto y séptimo elemento .key.

Paso 7

Ahora copia el conjunto de siete elementos .key, y pega otros dos conjuntos en el div .keys.

Paso 8

Agrega un elemento link dentro de tu elemento head. Para ese elemento link, establece el atributo rel en stylesheet y el atributo href en ./styles.css.

Paso 9

Los navegadores pueden aplicar valores de margen y relleno por defecto a elementos específicos. Para asegurarte de que tu piano tiene un aspecto correcto, debe reajustar el modelo de caja.

Agrega una regla que tenga el selector html a tu archivo CSS y establece la propiedad box-sizing en border-box.

Paso 10

Ahora que has restablecido el modelo de caja del html, tienes que pasarlo también a los elementos de su interior. Para ello, puedes establecer la propiedad box-sizing en inherit, lo que indicará a los elementos apuntados que utilicen el mismo valor que el elemento padre.

También tendrás que apuntar a los pseudo-elementos, que son palabras clave especiales que siguen a un selector. Los dos pseudo-elementos que usaras son ::before y ::after.

El selector ::before crea un pseudo-elemento que es el primer hijo del elemento seleccionado, mientras que el selector ::after crea un pseudo-elemento que es el último hijo del elemento seleccionado. Estos pseudo-elementos a menudo se usan para crear contenido cosmético, que verás más adelante en este proyecto.

Por ahora, crea un selector CSS para apuntar a todos los elementos con \*, e incluye los pseudo-elementos con ::before y ::after. Establece la propiedad box-sizing en inherit.

Paso 11

Ahora apunta a tu elemento #piano con un selector de id. Establece la propiedad background-color a #00471b, la propiedad width a 992px y la propiedad height a 290px.

Paso 12

Establece el margin del #piano en 80px auto.

Paso 13

Es hora de estilizar las teclas. Debajo de la regla #piano, apunta al elemento .keys con un selector de clase. Dale a la nueva regla una propiedad background-color de #040404, una propiedad width de 949px y una propiedad height de 180px.

Paso 14

Dale al .keys un padding-left de 2px.

Paso 15

Mueve las teclas a la posición correcta ajustando el selector #piano. Establece la propiedad padding a 90px 20px 0 20px.

Paso 16

Es hora de estilizar las teclas en sí. Crea un selector class para los elementos .key. Establece el background-color al valor #ffffff, la propiedad position a relative, la propiedad width a 41px y la propiedad height a 175px.

Paso 17

Dale al .key una propiedad margin establecida en 2px y una propiedad float establecida en left.

Paso 18

Ahora es el momento de utilizar los pseudo-selectores que has preparado antes. Para crear las teclas negras, agrega un nuevo selector .key.black--key::after. Esto apuntará a los elementos con la clase key black--key, y seleccionará el pseudo-elemento después de estos elementos en el HTML.

En el nuevo selector, establece el background-color en #1d1e22. También establece la propiedad content en "". Esto hará que los pseudo-elementos estén vacíos.

La propiedad content se utiliza para establecer o anular el contenido de un elemento. Por defecto, los pseudo-elementos creados por los pseudo-selectores ::before y ::after están vacíos, y los elementos no serán renderizados en la página. Establecer la propiedad content a una cadena vacía "" asegurará que el elemento se renderice en la página aunque siga estando vacío.

Si quieres experimentar, prueba eliminando la propiedad background-color y a establecer valores diferentes para la propiedad content, como "♥". Recuerde deshacer estos cambios cuando haya terminado para que las pruebas pasen.

Paso 19

Dale al .key.black--key::after una propiedad position establecida en absolute y una propiedad left establecida en -18px.

Paso 20

Para el .key.black--key::after, establece el width en 32px y el height en 100px.

Paso 21

El piano necesita el logo de freeCodeCamp para hacerlo oficial.

Agrega un elemento img antes de tu elemento .keys. Dale al img un class de logo, y establece el src en https://cdn.freecodecamp.org/platform/universal/fcc\_primary.svg. Dale un texto alt de freeCodeCamp Logo.

Paso 22

Empieza a estilizar el logo creando un selector .logo. Establece el width a 200px, un position a absolute y un top establecido en 23px.

Paso 23

El elemento img necesita que su elemento padre tenga un position establecido como un punto de referencia. Establece el position del selector #piano a relative.

Paso 24

Para suavizar los bordes afilados del piano y las teclas, empieza dando al #piano un border-radius de 10px.

Paso 25

Dale al selector .key un valor border-radius de 0 0 3px 3px.

Paso 26

Dale al selector .key.black--key::after un border-radius de 0 0 3px 3px para que coincida con las teclas.

Paso 27

La regla-at (declaración CSS que comienza con el símbolo arroba) @media, también conocida como media query, se utiliza para aplicar CSS de forma condicional. Las media queries se utilizan comúnmente para aplicar CSS basado en el ancho del viewport utilizando las propiedades max-width y min-width.

En el siguiente ejemplo, el relleno se aplica a la clase .card cuando el viewport tiene un ancho de 960px o inferior.

@media (max-width: 960px) {

.card {

padding: 2rem;

}

}

Agrega una media query que se aplicará cuando el viewport tenga 768px de ancho o menos.

Paso 28

Agrega un nuevo selector #piano dentro de tu @media query, y establece el width en 358px.

Paso 29

Dentro de la @media query, agrega un selector .keys y establece el width a 318px.

Paso 30

Ahora agrega un selector .logo a la @media query, y establece la propiedad width en 150px.

Paso 31

Puede que hayas notado que las teclas se colapsan cuando la ventana del navegador es menor de 768px. Establece el overflow a hidden en el primer selector .keys, para solucionar este problema. Esta propiedad ocultará cualquier elemento que sea empujado fuera del valor del width establecido en el .keys.

Paso 32

Es posible aumentar la complejidad de las media queries usando operadores lógicos. El operador lógico and es utilizado para combinar en una sola consulta dos características de medios.

Por ejemplo, una media query cuyo objetivo es una anchura de pantalla entre 500px y 1000px podría ser:

@media (min-width: 500px) and (max-width: 1000px){

}

Añade otra regla @media que se aplique si la pantalla del navegador tiene una anchura mayor que 769px pero menor que 1199px.

Paso 33

Para nueva regla @media, establece el width del #piano a 675px y el width del .keys a 633px.

Con esto, ¡tú piano está completo!