1. ¿Cuáles son algunas de las cosas que hacen que SCSS sea diferente de CSS? (Coloca ejemplos)

SCSS (Syntactically Awesome Style Sheets): Es una extensión de CSS es de gran ayuda por hace que la codificación de estilos sea más flexible cuando usamos, CSS. Aquí están algunas de las diferencias clave entre SCSS y CSS:

* Anidamiento: SCSS permite anidar selectores, lo que hace que el código sea más limpio y comprensible.

Ejemplo en SCSS:

.container {

width: 100%;

.header {

background-color: blue;

color: white;

}

.footer {

background-color: grey;

color: black;

}

}

CSS equivalente:

.container {

width: 100%;

}

.container .header {

background-color: blue;

color: white;

}

.container .footer {

background-color: grey;

color: black;

}

* Variables: Cuando usamos SCSS, nos permite definir variables, para almacenar valores que podemos volver a utilizar, como los colores, tamaños de fuente, etc.
* Mixins y funciones: SCSS permite la reutilizar bloques de código mediante mixins y el uso de funciones, lo que facilita mantenga el código. Es como una plantilla que se crea y luego puede agregarla cuando lo necesites
* Operaciones matemáticas: SCSS permite realizar operaciones matemáticas directamente en el CSS, como sumar o restar valores.

Ejemplo de operación en SCSS:

.box {

width: 100px + 20px; Esto es igual 120px

height: 50px \* 2; Esto es igual 100px

}

2. ¿Qué es una variable SCSS? (¿Por qué crees que debes utilizarla? Pon un ejemplo de una variable, escribe una variable y cómo se pondría para utilizarla)

La variable SCSS

Es una manera de almacenar un valor que luego puedo volver a utilizar, en todo el código de CSS. Es como una “etiqueta”, para un valor y se puede usar una y otra vez, nos permite tener una codificación más limpia, organizada y fácil de modificar.

Ejemplo:

En este caso las variables serán el color y el tamaño de la fuente.

$primary-color: #3498db;

$font-size: 16px;

Y luego definimos el lugar, en el que vamos a utilizar estas variables. Se pueden aplicar de manera general o para un sitio especifico de la codificación.

body {

font-size: $font-size;

color: $primary-color;

}

button {

background-color: $primary-color;

color: white;

}

¿Por qué utilizarlas?

* Hacen que sea más fácil modificar el diseño: Esto se traduce que puedo cambiar variables, como el color, tamaño de la fuente, etc., en un solo lugar y no en cada parte del código del estilo (CSS).
* Reciclable: Puedes reutilizar las mismas variables en varios lugares.

3. ¿Qué es un SCSS Mixin? (¿Por qué crees que debes utilizarla? Pon un ejemplo de un mixin, escribe cómo se crea y cómo se pondría para utilizarla)

Un SCSS mixin es una especie de "plantilla" o "bloque de código reutilizable". Permite agrupar propiedades y valores CSS que luego puedes incluir en otros selectores. Es útil cuando tienes reglas de estilo que se repiten y quieres evitar la duplicación de código.

Ejemplo:

// Definir un mixin

@mixin border-radius($radius) {

-webkit-border-radius: $radius;

-moz-border-radius: $radius;

border-radius: $radius;

}

// Utilizar el mixin

.box {

@include border-radius(10px);

background-color: #f0f0f0;

}

.button {

@include border-radius(5px);

padding: 10px 20px;

background-color: #3498db;

}

¿Por qué utilizarlas?

* Reutilización: Evita la repetición de código.
* Mantenibilidad: Si necesitas cambiar el valor de una propiedad en varios lugares, solo tienes que actualizar el mixin en un lugar.
* Flexibilidad: Puedo pasar parámetros a un mixin y personalizar las reglas de estilo sin duplicar el código.

CSS compilado:

El código SCSS anterior se compilaría a este CSS:

.button {

background-color: #3498db;

color: white;

padding: 10px 20px;

border-radius: 10px;

box-shadow: 0 4px 6px rgba(44, 62, 80, 0.2);

}

.card {

background-color: #fff;

padding: 20px;

border-radius: 15px;

box-shadow: 0 4px 6px rgba(231, 76, 60, 0.2);

}

4. ¿Qué significa Unidad fraccionaria (fr) con CSS Grid?

La unidad fr (fracción) en CSS Grid, es una especie de fracción de espacio disponible que está en una cuadricula.

Y al usar fr, le decimos al navegador que divida el espacio restante disponible entre las columnas o filas que usan esa unidad. Se divide el espacio de forma flexible y se distribuye el espacio, no es necesario definir valores fijos como píxeles o porcentajes.

Ejemplo con CSS Grid:

.container {

display: grid;

grid-template-columns: 1fr 2fr 1fr;

gap: 10px;

}

.item {

background-color: #3498db;

color: white;

padding: 20px;

}

En este ejemplo:

* La primera columna ocupa 1 fracción del espacio
* La segunda columna ocupa 2 fracciones, y será el doble que la primera.
* La tercera columna ocupa 1 fracción de espacio y tendrá el mismo tamaño que la primera

La unidad fr es muy útil cuando no conoces el tamaño exacto del espacio disponible, pero quieres que los elementos se ajusten de manera flexible y proporcional.