



Bottega DevCamp

FULL STACK DEVELOPER

CHECKPOINT 1: EJERCICIOS TEÓRICOS

Fecha: 30 de enero de 2025

Autor: Domínguez Becerril, Jorge

DNI: 72607195J

# Índice

<b>1. Cuestiones teóricas</b>	<b>1</b>
1.1. ¿Cuáles son algunas de las cosas que hacen que SCSS sea diferente de CSS?	1
1.2. ¿Qué es una variable SCSS? . . . . .	3
1.3. ¿Qué es un SCSS Mixin? . . . . .	4
1.4. ¿Qué significa Unidad fraccionaria (fr) con CSS Grid? . . . . .	6

# 1. Cuestiones teóricas

## 1.1. ¿Cuáles son algunas de las cosas que hacen que SCSS sea diferente de CSS?

CSS (Cascading Style Sheets, por sus siglas en inglés) es un lenguaje que se utiliza para definir los estilos de un documento HTML. Por otro lado, SCSS es un lenguaje de preprocesado de scripts que extiende las capacidades de CSS permitiendo la utilización de variables, anidados, mixins y condicionales, entre otras muchas cosas. Esto evita la iteración de líneas de código que habitualmente se produce en documentos escritos en CSS y, además, facilita la escritura de código escalable.

Por ejemplo, en la figura 1(a) se muestra un fragmento de código de CSS. Como se puede ver, hay varios elementos que se repiten en ambas etiquetas `div`, el color del fondo (`background-color`) y el margen inferior (`margin-bottom`). Este código se puede reescribir utilizando las variables de SCSS, tal y como puede ver en la figura 1(b). Las variables utilizadas son `$background-color` y `$margin-bottom`. Gracias a esto, si en un futuro se desea cambiar el color del fondo o el margen inferior tan solo hay que cambiar el valor de la variable y no hay que ir cambiando uno por uno los valores dentro de cada selector.

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  background-color: green;  
  margin-bottom: 100px;  
}  
  
.content-wrapper {  
  display: grid;  
  grid-template-columns: 1fr 3fr;  
  margin-bottom: 100px;  
  background-color: green;  
}
```

(a)

```
$background-color: green;  
$margin-bottom: 100px;  
  
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  background-color: $background-color;  
  margin-bottom: $margin-bottom;  
}  
  
.content-wrapper {  
  display: grid;  
  grid-template-columns: 1fr 3fr;  
  margin-bottom: $margin-bottom;  
  background-color: $background-color;  
}
```

(b)

Figura 1: Fragmento de código en CSS (a) y reescrito en SCSS utilizando variables (b).

Otra de las diferencias de SCSS con respecto a CSS son las anidaciones. En la figura 2(a) se recoge un fragmento de código de CSS. Se puede ver que hay dos selectores, **.container** y **.content-wrapper**, siendo **.container** el padre y **.content-wrapper** el hijo. Este código se puede reescribir con SCSS utilizando anidaciones, tal y como se muestra en la figura 2(b). La diferencia radica en que el selector hijo está dentro de los corchetes del selector padre y no en un bloque aparte como en CSS. Las anidaciones muestran de forma cristalina la jerarquía entre las distintas etiquetas de un documento escrito en lenguaje HTML y requieren escribir menos, con el consiguiente ahorro temporal y dinerario.

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  background-color: green;  
  margin-bottom: 100px;  
}  
  
.container > .content-wrapper {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  background-color: green;  
}
```

(a)

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  background-color: green;  
  margin-bottom: 100px;  
  .content-wrapper {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    background-color: green;  
  }  
}
```

(b)

Figura 2: Fragmento de código en CSS (a) y reescrito en SCSS utilizando anidaciones (b).

## 1.2. ¿Qué es una variable SCSS?

Una variable en SCSS es un contenedor que almacena una característica, estilo o propiedad y que después se incluye dentro de un selector. Por ejemplo, en la figura 3 se han definido las variables **\$background-color** y **\$margin-bottom**, que almacenan, respectivamente, el color *green* y la longitud de 100px. Estas variables se utilizan posteriormente dentro de los selectores **.container** y **.content-wrapper** para definir los estilos del color del fondo y el margen inferior, aunque se podrían utilizar para definir otras propiedades.

```
$background-color: green;
$margin-bottom: 100px;

.container {
  display: grid;
  grid-template-columns: 1fr 1fr;
  background-color: $background-color;
  margin-bottom: $margin-bottom;
}

.content-wrapper {
  display: grid;
  grid-template-columns: 1fr 3fr;
  margin-bottom: $margin-bottom;
  background-color: $background-color;
}
```

Figura 3: Fragmento de código de SCSS donde se definen las variables **\$background-color** y **\$margin-bottom**.

El uso de variables permite agrupar todas aquellas propiedades compartidas por varios selectores, como en la figura 3, donde los selectores comparten el color del fondo y el margen inferior. Esto permite que, si en un futuro, se desea cambiar el valor de alguna de las propiedades compartidas, tan solo hay que cambiar el valor de la variable. En el ejemplo, si se quisiera cambiar el color del fondo a rojo, solo habría que cambiar la primera línea de código por **.\$background-color: red;**. Esto permite que el código sea escalable, una propiedad altamente deseable.

### 1.3. ¿Qué es un SCSS Mixin?

Un mixin es un conjunto de estilos y propiedades que acepta argumentos y permite la reutilización de código. Se define con la palabra clave `@mixin` seguida del nombre y sus posibles parámetros. En la figura 4 se muestra un mixin de nombre `cuadrícula` que toma como argumentos las variables `$background-color` y `$margin-bottom` y que contiene unas líneas de código que definen una cuadrícula (grid) de dos columnas del mismo tamaño (1fr) con un color de fondo determinado por la variable `$background-color` y un margen inferior determinado por `$margin-bottom`.

```
@mixin cuadrícula($background-color,$margin-bottom) {
  display: grid;
  grid-template-columns: 1fr 1fr;
  background-color: $background-color;
  margin-bottom: $margin-bottom;
}
```

Figura 4: Mixin de SCSS.

Para utilizar el mixin dentro de un selector se debe utilizar la palabra clave `@include` seguida del nombre del mixin y sus posibles argumentos. En la figura 5 se muestra cómo se utiliza el mixin dentro del selector `.container`.

```
@mixin cuadrícula($background-color,$margin-bottom) {
  display: grid;
  grid-template-columns: 1fr 1fr;
  background-color: $background-color;
  margin-bottom: $margin-bottom;
}

.container {
  @include cuadrícula(green,100px)
}
```

Figura 5: Mixin de SCSS.

Los mixins son muy útiles cuando varios selectores comparten estilos y propiedades comunes. Por ejemplo, el código de CSS de la figura 1(a) tiene dos selectores que comparten el mismo código, de manera que se puede reescribir el código CSS de la figura 1(a) utilizando un mixin, concretamente el mixin `cuadrícula` definido en la figura 4. En la figura 6 se muestra como queda el código reescrito con el mixin.

```
@mixin cuadrícula($background-color,$margin-bottom) {
  display: grid;
  grid-template-columns: 1fr 1fr;
  background-color: $background-color;
  margin-bottom: $margin-bottom;
}

.container {
  @include cuadrícula(green,100px)
}

.content-wrapper {
  @include cuadrícula(green,100px)
}
```

Figura 6: Código de CSS reescrito utilizando un mixin.

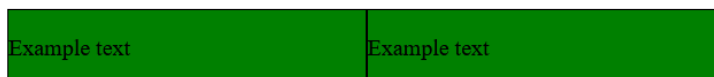
A priori parece que no supone una gran ventaja en términos del número de líneas de código, pero hay que tener en cuenta que en el ejemplo solo hay dos selectores. En un proyecto más grande el uso de mixins evita la repetición de muchas líneas de código, ahorrando tiempo.

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

La unidad fraccionaria (fr) es una unidad que representa la fracción del espacio disponible en el contenedor grid. Por ejemplo, en la figura 7(a) se muestra el código CSS para un grid con dos columnas de la misma anchura. Dentro del selector, en la propiedad **grid-template-columns**, es donde se define el número de columnas y su anchura. En este caso se tienen dos columnas con una anchura de 1fr, de manera que cada columna ocupa la mitad del espacio disponible para el grid. En la figura 7(b) se puede ver el resultado final.

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
}
```

(a)



(b)

Figura 7: Código de CSS para un grid con dos columnas de la misma anchura (a) y resultado (b).

En la figura 8(a) se muestra el código para otro grid o cuadrícula con dos columnas, pero esta vez de distinta anchura. La primera columna tiene una anchura de 1fr mientras que la segunda columna tiene una anchura de 3fr. Por lo tanto, la segunda columna tiene una anchura 3 veces mayor que la primera columna. En la figura 8(b) se puede ver el resultado final.

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 3fr;  
}
```

(a)



(b)

Figura 8: Código de CSS para un grid con dos columnas de distinta anchura (a) y resultado (b).