

# MODULE 1 - 004: SaSS SCSS

So far we have been through a full overview of SCSS, walked through the preprocessors step so you can see what code was written and then we looked at how it got translated into something the browser could understand.

**\* There is something that might be a little bit confusing whether it's in some type of coding interview situation, or just in general as reading documentation.**

**There are actually two different types of SASS syntax: - SASS - In which SASS was evolved: SCSS\*\***

Technically, both of those are called SASS which makes for some kind of confusing component when you're first starting to learn it.

Here we are going to talk about the differences there are between the two and what we are going to be using for the course.

## 1. SASS was the original version.

- It does not look anything like CSS.
- It cares about things such as indentation.
- When using SASS all of your code has to be segmented.
- It has to have the right number of spaces right in front of each set of style definitions.
- It doesn't have {curly braces}
- It only has plus signs and equal signs all over the place.
- If you've ever worked with *Hammill* or *Slim*, you may notice it looks very similar to that type of setup.

What you will see more often is SCSS and that's what we're going to be using in this entire course.

## Reasons to choose SCSS in production

There are many reasons why SCSS is used so widely:

### 1. Lower Learning-curve :

\* It looks exactly like CSS.

\* It uses almost the same syntax. This makes it much more familiar.

### 2. Backward Compatibility:

\* If you want to write just pure CSS code an SCSS file will still process it where the original version of SASS would actually throw an error.

I wanted to go over these differences due to the fact that during your research you are going to run into both syntax.

Without this brief overview, it may seem very confusing. \*\*\* # Video Speech: [ENG]

## Differences Between Sass and Scss

So far we have been through a full overview of SCSS, walked through the preprocessors step so you can see what

There is something that might be a little bit confusing whether it's in some type of coding interview situation

SASS was the original version. It does not look anything like CSS and it cares about things such as indentation

What you will see more often is SCSS and that's what we're going to be using in this entire course. There are

I wanted to go over these differences due to the fact that during your research you are going to run into both

---

[SPA]

## Diferencias entre Sass y Scss

Hasta ahora hemos explorado a fondo SCSS, hemos visto cómo funciona el proceso de precompilación y cómo se tr

Sin embargo, puede haber cierta confusión, especialmente en entrevistas técnicas o al leer documentación, ya

Sass fue la versión original. Su sintaxis es muy diferente a la de CSS y se basa en la indentación. Todo el c

Scss, por otro lado, es la sintaxis más común y la que utilizaremos en este curso. Su principal ventaja es qu

¿Por qué usar Scss?

Sintaxis familiar: Se parece mucho a CSS, lo que reduce la curva de aprendizaje.

Flexibilidad: Permite escribir código CSS puro dentro de archivos Scss.

Ampliamente utilizado: Es la sintaxis más popular y soportada por la mayoría de las herramientas y frameworks.

En resumen, aunque Sass y Scss son parte de la misma familia, sus sintaxis son bastante diferentes. Scss, al

¿Por qué es importante conocer esta diferencia?

Es fundamental entender las diferencias entre Sass y Scss para poder elegir la herramienta adecuada para cada caso.

En este curso, nos centraremos en Scss debido a su popularidad y facilidad de uso.

Espero que esta explicación te haya aclarado las diferencias entre Sass y Scss. ¡Continuemos explorando el mundo de Scss!