**General Information**

SCSS (Sassy CSS) is a superset of CSS (Cascading Style Sheets) that adds several enhancements and features to traditional CSS. SCSS retains the same basic syntax as CSS, making it easy for developers to transition from writing regular CSS to using SCSS.

With SCSS, you can write CSS code with additional features like variables, nesting, mixins, and functions. These features make it easier to organize and maintain CSS code, improve code reusability, and enhance the overall development workflow.

Here are some key features of SCSS:

**Variables**: SCSS allows you to define variables, which can store reusable values such as colors, font sizes, or dimensions. Using variables helps maintain consistency and allows you to easily update values across the entire stylesheet.

**Nesting**: SCSS provides nesting support, allowing you to nest selectors within one another. This helps in organizing and structuring your CSS code, making it more readable and reducing repetition.

**Mixins**: Mixins are reusable blocks of code that can be included in multiple selectors. They enable you to define a set of styles and then apply them to various elements, reducing code duplication.

**Functions**: SCSS supports custom functions, which allow you to perform calculations, manipulate colors, and perform other operations within your stylesheets. This adds flexibility and extends the capabilities of CSS.

**Partials and Imports**: SCSS allows you to split your CSS code into smaller, modular files called partials. You can then import these partials into a main SCSS file, making it easier to manage and organize your styles.

**Inheritance**: SCSS supports inheritance, where one selector can inherit styles from another. This can help reduce code repetition and make it easier to define variations of styles.

To use SCSS in your projects, you'll need to compile the SCSS code into regular CSS code using a preprocessor or a build tool. There are various tools available, such as Sass (the original Ruby-based implementation), Node-sass, Dart-sass, or using build tools like webpack or Gulp.

Overall, SCSS enhances CSS by providing additional features and improving the development experience, making it a popular choice for styling web applications and websites.

**Note:**

In SCSS, variables are used to store and reuse values throughout your stylesheets. However, variables in SCSS are evaluated at compile-time, not runtime. So, changing the value of a variable dynamically based on a class or state change is not possible using pure SCSS.