**Sass: A CSS Preprocessor**

**Sass** (Syntactically Awesome Stylesheets) is a CSS preprocessor that extends the functionality of CSS by adding features like nesting, variables, mixins, and more. Sass files are compiled into standard CSS that can be used in web development.

**Key Differences Between Sass and Regular CSS:**

* **Nesting:** Sass allows you to nest selectors within other selectors, making your CSS code more organized and readable.
* **Variables:** You can define variables in Sass to store and reuse values throughout your stylesheet.
* **Mixins:** Mixins are reusable blocks of CSS code that can be included in other stylesheets.
* **Functions:** Sass provides built-in functions for common CSS tasks, such as calculating colors or converting units.

**Example Sass File:**

SCSS

$primary-color: #007bff;

.button {

background-color: $primary-color;

color: white;

padding: 10px 20px;

border: none;

cursor: pointer;

&:hover {

background-color: darken($primary-color,

10%);

}

}

@import "partials/mixins";

.container {

@include clearfix;

}

Use code [with caution.](/faq#coding)

**Explanation:**

* **Nesting:** The .button:hover selector is nested within the .button selector, making the code more concise and readable.
* **Variables:** The $primary-color variable is defined and used throughout the stylesheet.
* **Mixins:** The @import statement imports a partial file named partials/mixins (which contains a mixin named clearfix). The @include directive includes the clearfix mixin in the .container class.

**Compiled CSS:**

CSS

.button {

background-color: #007bff;

color: white;

padding: 10px 20px;

border: none;

cursor: pointer;

}

.button:hover {

background-color: #0066ff;

}

.container:before, .container:after {

content: "";

display: table;

}

.container:after {

clear: both;

}

Use code [with caution.](/faq#coding)

**Notable Differences from Standard CSS:**

* **Nesting:** Sass allows you to nest selectors, making your code more organized.
* **Variables:** Sass variables provide a way to store and reuse values.
* **Mixins:** Mixins allow you to define reusable blocks of CSS code.
* **Functions:** Sass provides built-in functions for common CSS tasks.