SCSS

🧠 What is SCSS?

SCSS stands for **Sassy CSS** — it's a **CSS preprocessor syntax** used by **Sass** (Syntactically Awesome Style Sheets).

👉 In short:
SCSS = Advanced CSS with superpowers 💪

It allows you to write **cleaner**, **reusable**, and **more organized** styles.
Later, SCSS is **compiled** into normal CSS that browsers understand.

🌈 Why Use SCSS?

Here's why developers love SCSS:

✅ **Variables** — store colors, fonts, sizes, etc.
✅ **Nesting** — write CSS inside CSS (like HTML structure).
✅ **Mixins** — reusable code blocks (like CSS functions).
✅ **Inheritance / Extends** — share styles between selectors.
✅ **Partials & Imports** — split your CSS into multiple files.
✅ **Functions & Math** — perform calculations right inside CSS.

🧩 1. Variables Example

Normal CSS:

```css
body {
  color: #333;
  background-color: #f5f5f5;
}
h1 {
  color: #333;
}
```

SCSS:

```scss
$primary-color: #333;
$background: #f5f5f5;

body {
  color: $primary-color;
  background-color: $background;
}

h1 {
  color: $primary-color;
}
```

🎯 Benefit: If you ever want to change your main color — just change $primary-color once!

🧱 2. Nesting

CSS:

```css
nav ul {
  list-style: none;
}
nav ul li {
  display: inline-block;
}
nav ul li a {
  color: blue;
}
```

SCSS:

scss

```scss
nav {
  ul {
    list-style: none;

    li {
      display: inline-block;

      a {
        color: blue;
      }
    }
  }
}
```

🎯 Cleaner, and matches HTML structure — easier to read and manage.

⚙️ 3. Mixins (Reusable Code Blocks)

SCSS:

scss

```scss
@mixin flex-center {
  display: flex;
  justify-content: center;
  align-items: center;
}

.container {
  @include flex-center;
  height: 100vh;
}
```

🎯 @mixin defines reusable styles, and @include calls them — like a function in programming.

👭 4. Inheritance (Extends)

SCSS:

```scss
.message {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.success {
  @extend .message;
  border-color: green;
}

.error {
  @extend .message;
  border-color: red;
}
```

🎯 Avoid repeating base styles — share common rules easily.

📁 5. Partials & Imports

You can split SCSS into multiple files for better organization.

Example:

```css
styles/
  _variables.scss
  _mixins.scss
  main.scss
```

_variables.scss

```scss
$primary-color: #4a90e2;
```

main.scss

```scss
@import 'variables';

body {
  color: $primary-color;
}
```

🎯 The underscore _ means it's a **partial** file (not compiled alone).

🧮 6. Functions and Math

You can do math right in your styles!

```scss
$base-size: 16px;

p {
  font-size: $base-size * 1.2;
  margin-top: $base-size / 2;
}
```

⚡ Final Output (Compiled CSS)

The SCSS code is **compiled** into regular CSS:

```css
p {
  font-size: 19.2px;
  margin-top: 8px;
}
```

✅ Works in **all browsers** — they only see the final CSS.

🧰 How to Use SCSS

You can use SCSS in a few ways:

1. **VS Code plugin** like "Live Sass Compiler"
2. **Command Line** using:

```bash
npm install -g sass
sass style.scss style.css
```

→ It converts your .scss file into .css