**Variables, Functions, and Inheritance in Sass**

**Sass** enhances CSS by adding features like **variables** and **functions**, which make it easier to maintain consistency across styles. **Inheritance** in Sass allows one selector to inherit properties from another, reducing repetition and making stylesheets more manageable.

**1. Variables in Sass**

In Sass, variables store reusable values such as colors, font sizes, and dimensions, allowing you to change values consistently throughout your stylesheet.

**Example of Declaring and Using Variables:**

scss

Copy code

// Declaring variables

$primary-color: #3498db;

$font-size-large: 18px;

$padding: 10px 20px;

.button {

background-color: $primary-color;

font-size: $font-size-large;

padding: $padding;

}

**Compiled CSS Output:**

css

Copy code

.button {

background-color: #3498db;

font-size: 18px;

padding: 10px 20px;

}

By using variables, changes to a value like $primary-color can be applied across all instances where it is used, keeping the design consistent.

**2. Functions in Sass**

Sass allows you to create **functions** to perform calculations or manipulate values, making your styles more dynamic. Sass also comes with built-in functions for color manipulation, math operations, and more.

**Example of Declaring and Using a Function:**

scss

Copy code

// Custom function to convert pixels to rem units

@function px-to-rem($px, $base-font-size: 16px) {

@return $px / $base-font-size \* 1rem;

}

// Using the function

.heading {

font-size: px-to-rem(32); // Converts 32px to 2rem

margin-bottom: px-to-rem(24); // Converts 24px to 1.5rem

}

**Compiled CSS Output:**

css

Copy code

.heading {

font-size: 2rem;

margin-bottom: 1.5rem;

}

This function ensures consistent units across the stylesheet and makes it easy to scale font sizes based on the base font size.

**3. Inheritance in Sass**

**Inheritance** allows one selector to inherit styles from another using the @extend directive. This prevents duplication and enables a more DRY (Don’t Repeat Yourself) approach to styling.

**Example of Inheritance:**

scss

Copy code

// Base style for all buttons

.button-base {

padding: 10px 20px;

border-radius: 5px;

font-size: 16px;

border: none;

color: white;

}

// Extend the base style for primary and secondary buttons

.primary-button {

@extend .button-base;

background-color: #3498db;

}

.secondary-button {

@extend .button-base;

background-color: #e74c3c;

}

**Compiled CSS Output:**

css

Copy code

.button-base, .primary-button, .secondary-button {

padding: 10px 20px;

border-radius: 5px;

font-size: 16px;

border: none;

color: white;

}

.primary-button {

background-color: #3498db;

}

.secondary-button {

background-color: #e74c3c;

}