Task 4-Variables, Functions, and Inheritance (Gemini)

**Using Variables and Functions in Sass**

**Variables**

Sass variables allow you to store reusable values, such as colors, fonts, or spacing. This promotes consistency and makes it easier to update styles globally.

**Example:**

$primary-color: #333;

$secondary-color: #f0f0f0;

body {

  background-color: $secondary-color;

  color: $primary-color;

}

**Functions**

Sass functions allow you to create custom functions to manipulate values or generate CSS. This can help reduce code repetition and make your stylesheets more modular.

**Example:**

@function lighten($color, $amount) {

  @return lighten($color, $amount \* 1%);

}

.button {

  background-color: lighten($primary-color, 20);

}

**Inheritance**

Sass inheritance allows you to create parent-child relationships between selectors. This makes it possible to share styles between different elements, reducing code duplication.

**Example:**

.base-style {

  font-size: 16px;

  font-family: Arial, sans-serif;

}

.heading {

  @extend .base-style;

  font-weight: bold;

  font-size: 24px;

}

.paragraph {

  @extend .base-style;

  line-height: 1.5;

}

**Combining Variables, Functions, and Inheritance**

$primary-color: #333;

$secondary-color: #f0f0f0;

@function lighten($color, $amount) {

  @return lighten($color, $amount \* 1%);

}

.base-style {

  font-size: 16px;

  font-family: Arial, sans-serif;

}

.heading {

  @extend .base-style;

  font-weight: bold;

  font-size: 24px;

  color: lighten($primary-color, 20);

}

.paragraph {

  @extend .base-style;

  line-height: 1.5;

  color: $secondary-color;

}

In this example:

* We've defined two variables: $primary-color and $secondary-color.
* We've created a lighten function to lighten a color by a specific amount.
* We've defined a .base-style class with basic font properties.
* The .heading and .paragraph classes inherit from the .base-style class and add specific styles.
* We've used the lighten function to dynamically generate a lighter shade of the primary color for the heading.

By leveraging variables, functions, and inheritance, you can create more efficient, maintainable, and scalable Sass stylesheets.