Sass Variable

Variables are a way to store information that you can re-use later.

With Sass, you can store information in variables, like:

- strings
- numbers
- colors
- booleans
- lists
- nulls

Sass uses the $ symbol, followed by a name, to declare variables:

Sass Variable Syntax:

$variablename: *value*;

Sass Variable Scope

Sass variables are only available at the level of nesting where they are defined.

Look at the following example:

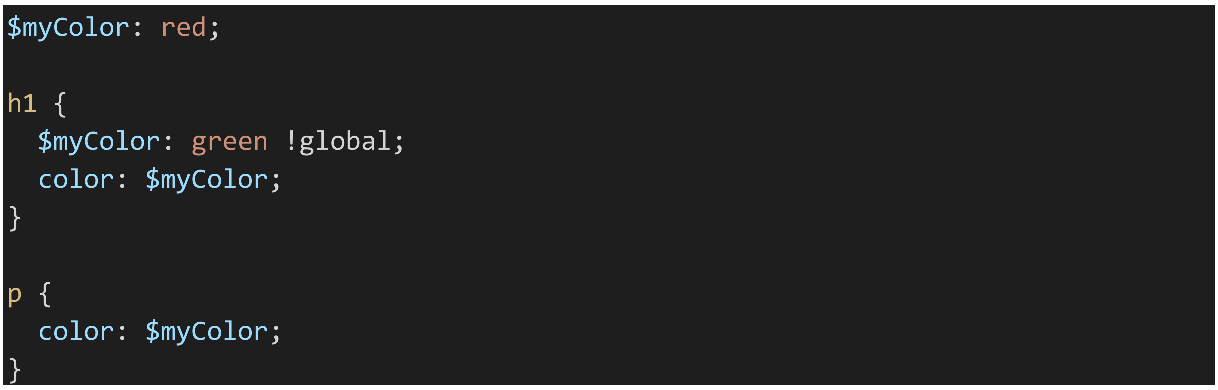
```
$myColor: red;

h1 {
  $myColor: green;
  color: $myColor;
}

p {
  color: $myColor;
}
```

Using Sass !global

The default behavior for variable scope can be overridden by using the !global switch.

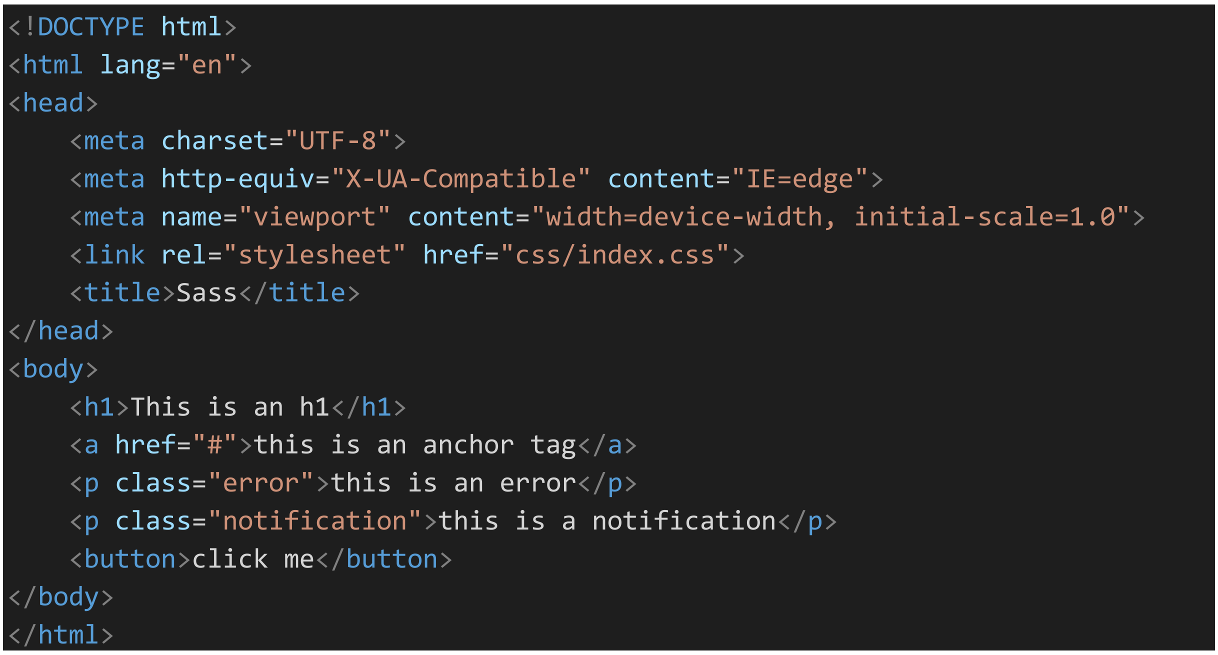
!global indicates that a variable is global, which means that it is accessible on all levels.

Look at the following example (same as above; but with !global added):



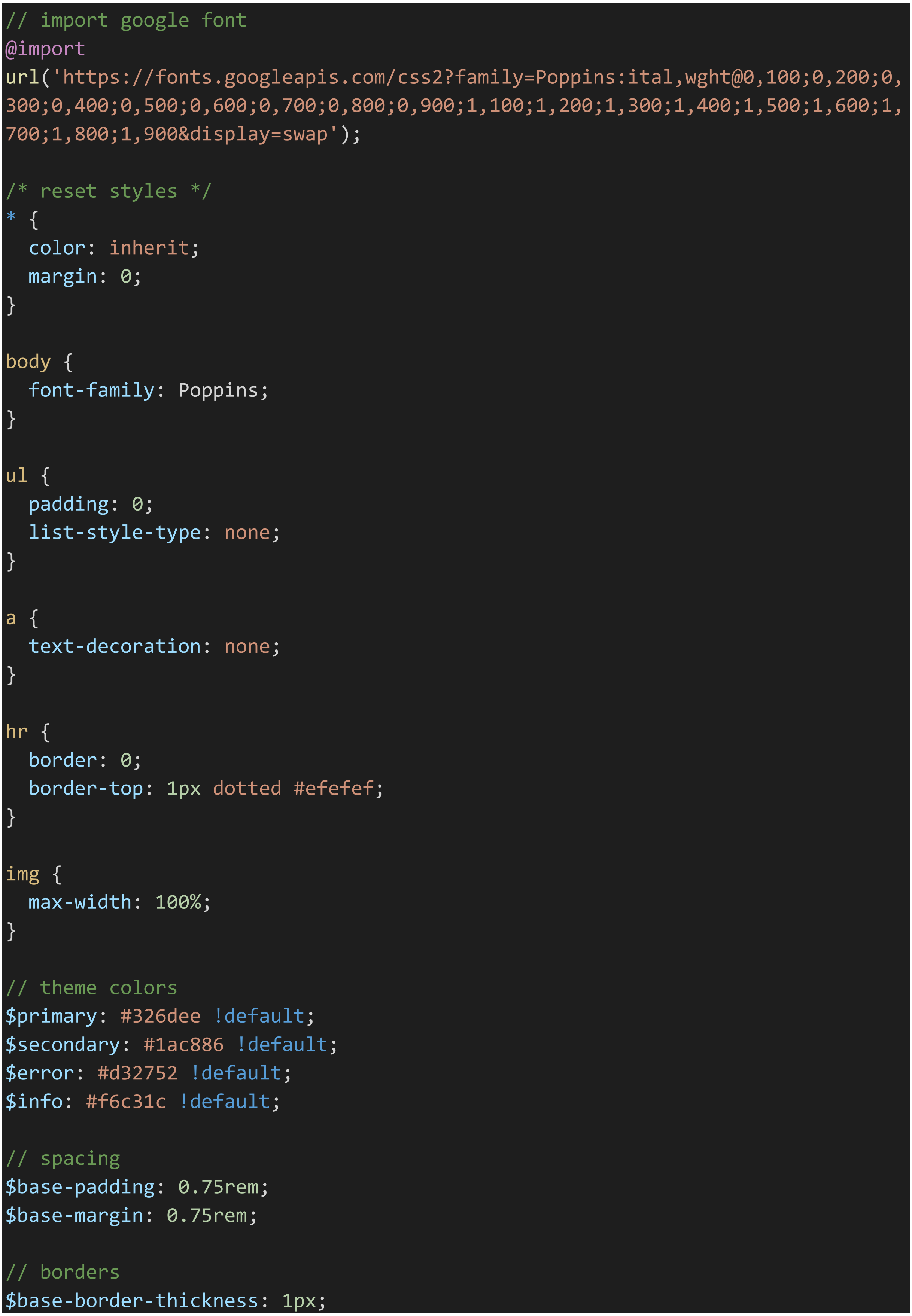
```scss
$myColor: red;

h1 {
  $myColor: green !global;
  color: $myColor;
}

p {
  color: $myColor;
}
```

Index.html file

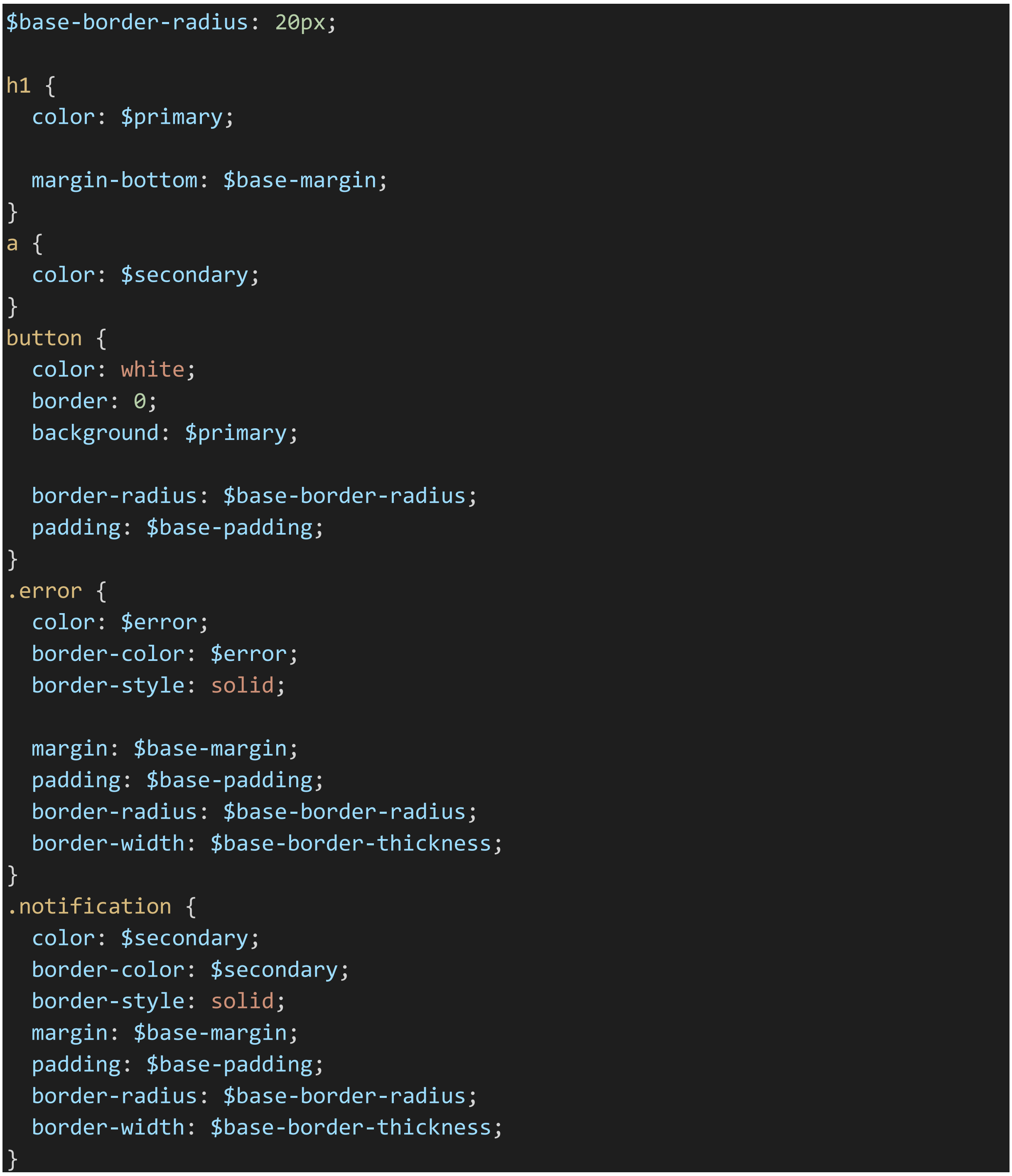


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/index.css">
    <title>Sass</title>
</head>
<body>
    <h1>This is an h1</h1>
    <a href="#">this is an anchor tag</a>
    <p class="error">this is an error</p>
    <p class="notification">this is a notification</p>
    <button>click me</button>
</body>
</html>
```

Index.sass file

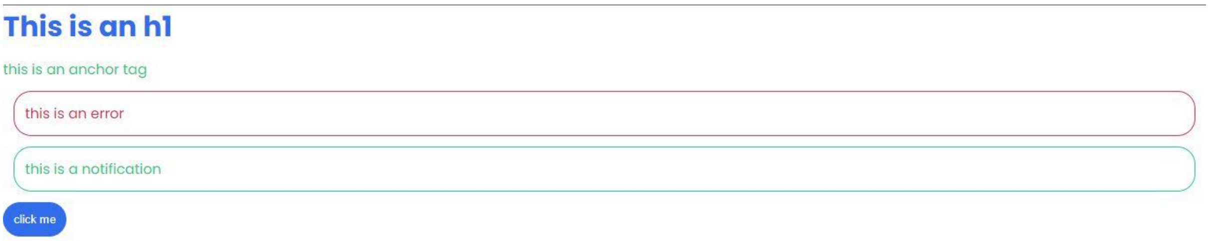


```
// import google font
@import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,
300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,
700;1,800;1,900&display=swap');

/* reset styles */
* {
  color: inherit;
  margin: 0;
}

body {
  font-family: Poppins;
}

ul {
  padding: 0;
  list-style-type: none;
}

a {
  text-decoration: none;
}

hr {
  border: 0;
  border-top: 1px dotted #efefef;
}

img {
  max-width: 100%;
}

// theme colors
$primary: #326dee !default;
$secondary: #1ac886 !default;
$error: #d32752 !default;
$info: #f6c31c !default;

// spacing
$base-padding: 0.75rem;
$base-margin: 0.75rem;

// borders
$base-border-thickness: 1px;
```



```scss
$base-border-radius: 20px;

h1 {
  color: $primary;

  margin-bottom: $base-margin;
}
a {
  color: $secondary;
}
button {
  color: white;
  border: 0;
  background: $primary;

  border-radius: $base-border-radius;
  padding: $base-padding;
}
.error {
  color: $error;
  border-color: $error;
  border-style: solid;

  margin: $base-margin;
  padding: $base-padding;
  border-radius: $base-border-radius;
  border-width: $base-border-thickness;
}
.notification {
  color: $secondary;
  border-color: $secondary;
  border-style: solid;
  margin: $base-margin;
  padding: $base-padding;
  border-radius: $base-border-radius;
  border-width: $base-border-thickness;
}
```

Output:



This is an h1

this is an anchor tag

this is an error

this is a notification