

CIS 371 Web Application Programming

Sass

Syntactically **A**wesome **S**tylesheets



GRAND VALLEY
STATE UNIVERSITY®

Lecturer: **Dr. Yong Zhuang**

Practice: Event Handling

Counting Click



Given the HTML and JavaScript code in the above link, implement functionality for the `<h1 id="h1counter">` element so that when it is clicked, the `<p id="outputH1counter">` displays 'Clicked ... times'.

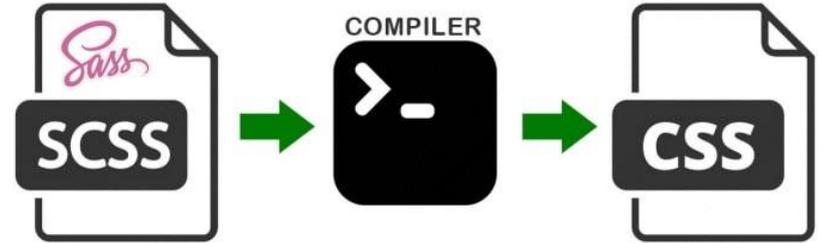
What is Sass?

- Sass stands for Syntactically Awesome Stylesheets
- A CSS preprocessor
- Extends CSS with additional features
- Compiles into standard CSS
- Common features: variables, nesting, mixins, functions



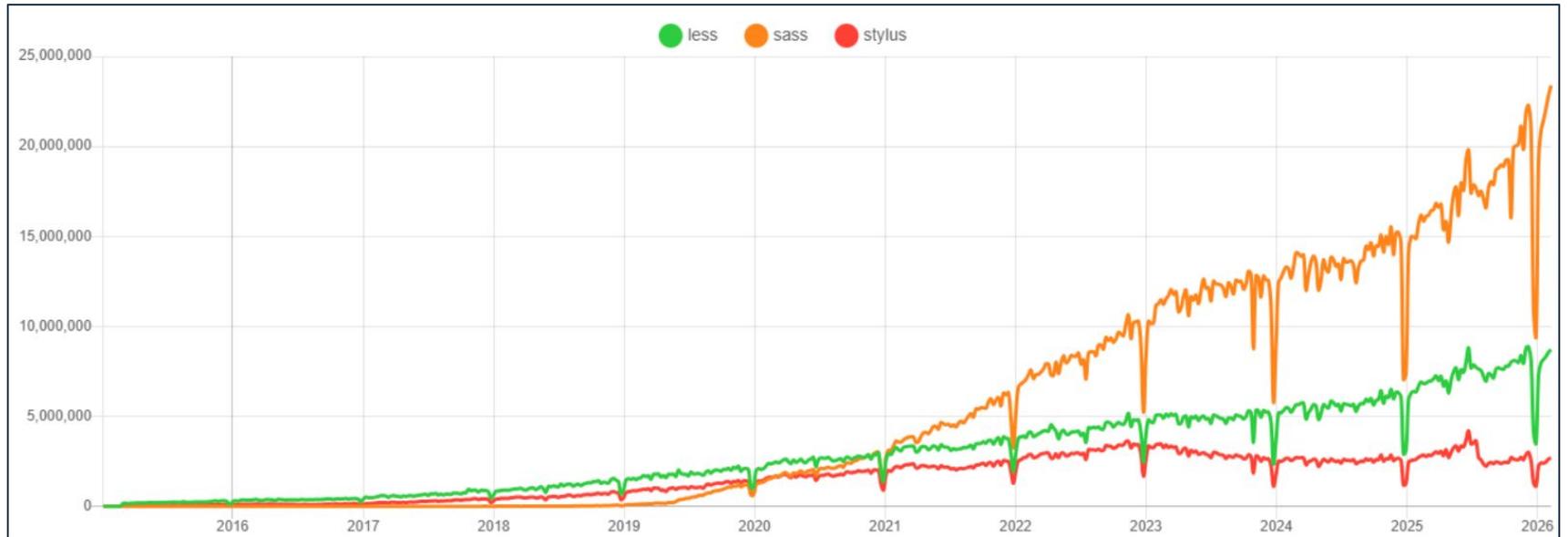
CSS Preprocessors

- CSS preprocessors take code written in the preprocessed language and then convert that code into CSS.



- 3 of the more popular CSS preprocessors are Sass, LESS, and Stylus.

CSS Preprocessors



NPM Trends

Sass Online Playground

<https://sass-lang.com/playground>

Install Sass

File Extension

.sass

- Older original file type
- Allows to write css without any curly-brackets or semi-colons
- Selectors and properties are distinguished by indentation

```
npm install -D sass
# OR
yarn add -D sass
```

.scss

- More similar to regular css
- It adds in all the sass helpers like indentation, variable and more
- Any regular .css file is a valid .scss file

An Example (HTML & CSS)

Online Playground

```
<section class="promo-banner">
  <h1 class="promo-title">Summer Sale!</h1>
  <p class="promo-text">Get 50% off all items.</p>
  <button class="promo-button">Shop Now</button>
</section>

<section class="promo-banner secondary">
  <h1 class="promo-title">Clearance</h1>
  <p class="promo-text">Limited stock available.</p>
  <button class="promo-button">View All</button>
</section>
```

HTML



What is “**wrong**” with this CSS?

style.css

```
.promo-banner {
  background-color: #3498db;
  padding: 40px;
  border-radius: 8px;
}
```

```
.promo-banner .promo-title {
  color: #ffffff;
  font-size: 2.5rem;
  margin-bottom: 10px;
}
```

```
.promo-banner .promo-text {
  color: #ffffff;
  font-size: 1.2rem;
}
```

```
.promo-banner .promo-button {
  background-color: #ffffff;
  color: #3498db;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}
```

```
/* Secondary Version */
.promo-banner.secondary {
  background-color: #e67e22;
}
```

```
.promo-banner.secondary .promo-button {
  color: #e67e22;
}
```

An Example (HTML & CSS)

Hard-coded Colors

The hex codes (#3498db, #e67e22) are repeated. If the brand color changes, we have to find and replace it everywhere



What is “**wrong**” with this CSS?

```
.promo-banner {  
  background-color: #3498db;  
  padding: 40px;  
  border-radius: 8px;  
}
```

```
.promo-banner .promo-title {  
  color: #ffffff;  
  font-size: 2.5rem;  
  margin-bottom: 10px;  
}
```

```
.promo-banner .promo-text {  
  color: #ffffff;  
  font-size: 1.2rem;  
}
```

```
.promo-banner .promo-button {  
  background-color: #ffffff;  
  color: #3498db;  
  padding: 10px 20px;  
  border: none;  
  cursor: pointer;  
}
```

```
/* Secondary Version */  
.promo-banner.secondary {  
  background-color: #e67e22;  
}
```

```
.promo-banner.secondary .promo-button {  
  color: #e67e22;  
}
```

style.css

Variables

style.scss

```
<section class="promo-banner">
  <h1 class="promo-title">Summer Sale!</h1>
  <p class="promo-text">Get 50% off all items.</p>
  <button class="promo-button">Shop Now</button>
</section>

<section class="promo-banner secondary">
  <h1 class="promo-title">Clearance</h1>
  <p class="promo-text">Limited stock available.</p>
  <button class="promo-button">View All</button>
</section>
```

HTML

```
$primary-blue: #3498db;
$primary-orange: #e67e22;
$white: #ffffff;
```

```
.promo-banner {
  background-color: $primary-blue;
  padding: 40px;
  border-radius: 8px;
}
```

```
.promo-banner .promo-title {
  color: $white;
  font-size: 2.5rem;
  margin-bottom: 10px;
}
```

```
.promo-banner .promo-text {
  color: $white;
  font-size: 1.2rem;
}
```

```
.promo-banner .promo-button {
  background-color: $white;
  color: $primary-blue;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}
```

```
/* Secondary Version */
.promo-banner.secondary {
  background-color: $primary-orange;
}
```

```
.promo-banner.secondary .promo-button {
  color: $primary-orange;
}
```

Online Doc

An Example (HTML & CSS)

style.scss

Lack of Hierarchy
The CSS doesn't visually reflect the nested structure of the HTML.

```
$primary-blue: #3498db;  
$primary-orange: #e67e22;  
$white: #ffffff;
```

```
.promo-banner {  
  background-color: $primary-blue;  
  padding: 40px;  
  border-radius: 8px;  
}
```

```
.promo-banner .promo-title {  
  color: $white;  
  font-size: 2.5rem;  
  margin-bottom: 10px;  
}
```

```
.promo-banner .promo-text {  
  color: $white;  
  font-size: 1.2rem;  
}
```

```
.promo-banner .promo-button {  
  background-color: $white;  
  color: $primary-blue;  
  padding: 10px 20px;  
  border: none;  
  cursor: pointer;  
}
```

```
/* Secondary Version */  
.promo-banner.secondary {  
  background-color: $primary-orange;  
}
```

```
.promo-banner.secondary .promo-button {  
  color: $primary-orange;  
}
```



What still smells **bad** here?

Nesting

```
<section class="promo-banner">
  <h1 class="promo-title">Summer Sale!</h1>
  <p class="promo-text">Get 50% off all items.</p>
  <button class="promo-button">Shop Now</button>
</section>

<section class="promo-banner secondary">
  <h1 class="promo-title">Clearance</h1>
  <p class="promo-text">Limited stock available.</p>
  <button class="promo-button">View All</button>
</section>
```

HTML

Key Point: Keep nesting shallow (ideally 2–3 levels) to maintain low specificity and reusable components.

[Online Doc](#)

```
$primary-blue: #3498db;
$primary-orange: #e67e22;
$white: #ffffff;
```

style.scss

```
.promo-banner {
  background-color: $primary-blue;
  padding: 40px;
  border-radius: 8px;
}

.promo-title {
  color: $white;
  font-size: 2.5rem;
  margin-bottom: 10px;
}

.promo-text {
  color: $white;
  font-size: 1.2rem;
}

.promo-button {
  background-color: $white;
  color: $primary-blue;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}

/* Secondary Version */
&.secondary {
  background-color: $primary-orange;
}

.promo-button {
  color: $primary-orange;
}
}
```

An Example (HTML & CSS)

Scalability

If we add more color themes, the file size and complexity grow significantly.



What still smells **bad** here?

```
$primary-blue: #3498db;
$primary-orange: #e67e22;
$white: #ffffff;
```

style.scss

```
.promo-banner {
  background-color: $primary-blue;
  padding: 40px;
  border-radius: 8px;
}
```

```
.promo-title {
  color: $white;
  font-size: 2.5rem;
  margin-bottom: 10px;
}
```

```
.promo-text {
  color: $white;
  font-size: 1.2rem;
}
```

```
.promo-button {
  background-color: $white;
  color: $primary-blue;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}
```

```
/* Secondary Version */
&.secondary {
  background-color: $primary-orange;

  .promo-button {
    color: $primary-orange;
  }
}
```

@mixin and @include

```
<section class="promo-banner">
  <h1 class="promo-title">Summer Sale!</h1>
  <p class="promo-text">Get 50% off all items.</p>
  <button class="promo-button">Shop Now</button>
</section>

<section class="promo-banner secondary">
  <h1 class="promo-title">Clearance</h1>
  <p class="promo-text">Limited stock available.</p>
  <button class="promo-button">View All</button>
</section>
```

HTML

```
$primary-blue: #3498db;
$primary-orange: #e67e22;
$white: #ffffff;
```

style.scss

```
@mixin banner-theme($theme-color) {
  background-color: $theme-color;

  .promo-button {
    color: $theme-color;
  }
}

.promo-banner {
  @include banner-theme($primary-blue);
  padding: 40px;
  border-radius: 8px;

  .promo-title {
    color: $white;
    font-size: 2.5rem;
    margin-bottom: 10px;
  }

  .promo-text {
    color: $white;
    font-size: 1.2rem;
  }

  .promo-button {
    background-color: $white;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
  }

  /* Secondary Version */
  &.secondary {
    @include banner-theme($primary-orange);
  }
}
```

Online Doc

Partials & Modules

style.scss

```
$primary-blue: #3498db;  
$primary-orange: #e67e22;  
$white: #ffffff;
```

```
@mixin banner-theme($theme-color) {  
  background-color: $theme-color;  
  
  .promo-button {  
    color: $theme-color;  
  }  
}  
  
.promo-banner {  
  @include banner-theme($primary-blue);  
  padding: 40px;  
  border-radius: 8px;  
  
  .promo-title {  
    color: $white;  
    font-size: 2.5rem;  
    margin-bottom: 10px;  
  }  
  
  .promo-text {  
    color: $white;  
    font-size: 1.2rem;  
  }  
  
  .promo-button {  
    background-color: $white;  
    padding: 10px 20px;  
    border: none;  
    cursor: pointer;  
  }  
  
  /* Secondary Version */  
  &.secondary {  
    @include banner-theme($primary-orange);  
  }  
}
```



As projects grow, how do we keep our code clean?
Can we put all variables into a separate file?

Partials & Modules

```
$primary-blue: #3498db;  
$primary-orange: #e67e22;  
$white: #ffffff;
```

`_variables.scss`

- As projects grow, one giant Sass file becomes a mess. We split them into **Partials** (files starting with `_`) and load them using **Modules**.

Key Point: Partials don't compile into their own CSS files; they are only used to organize the main file.

Online Doc

`style.scss`

```
/* Load the partial */  
@use 'variables';  
  
@mixin banner-theme($theme-color) {  
  background-color: $theme-color;  
  
  .promo-button {  
    color: $theme-color;  
  }  
}  
  
.promo-banner {  
  @include banner-theme(variables.$primary-blue);  
  padding: 40px;  
  border-radius: 8px;  
  
  .promo-title {  
    color: variables.$white;  
    font-size: 2.5rem;  
    margin-bottom: 10px;  
  }  
  
  .promo-text {  
    color: variables.$white;  
    font-size: 1.2rem;  
  }  
  
  .promo-button {  
    background-color: variables.$white;  
    padding: 10px 20px;  
    border: none;  
    cursor: pointer;  
  }  
  
  /* Secondary Version */  
  &.secondary {  
    @include banner-theme(variables.$primary-orange);  
  }  
}
```

Inheritance



If we have the same alignment and padding in `.promo-title` and `.promo-text`. Can we extract a 'Super Style' to manage these in one place?

style.scss

```
/* Load the partial */
@use 'variables';

@mixin banner-theme($theme-color) {
  background-color: $theme-color;

  .promo-button {
    color: $theme-color;
  }
}

.promo-banner {
  @include banner-theme(variables.$primary-blue);
  padding: 40px;
  border-radius: 8px;

  .promo-title {
    color: variables.$white;
    font-size: 2.5rem;
    margin-bottom: 10px;
  }

  .promo-text {
    color: variables.$white;
    font-size: 1.2rem;
  }

  .promo-button {
    background-color: variables.$white;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
  }
}

/* Secondary Version */
&.secondary {
  @include banner-theme(variables.$primary-orange);
}
}
```

Extend/Inheritance



What's the color of "Summer Sale"?

```
<section class="promo-banner">
  <h1 class="promo-title">Summer Sale!</h1>
  <p class="promo-text">Get 50% off all items.</p>
  <button class="promo-button">Shop Now</button>
</section>

<section class="promo-banner secondary">
  <h1 class="promo-title">Clearance</h1>
  <p class="promo-text">Limited stock available.</p>
  <button class="promo-button">View All</button>
</section>
```

HTML

```
/* This CSS will print because it is extended. */
%super-text-style {
  text-align: center;
  padding: 10px;
  color: black;
}

/* This CSS won't print because it is never extended.*/
%another-text-style {
  text-align: left;
  padding: 2px;
}
```

```
.promo-banner {
  @include banner-theme(variables.$primary-blue);
  padding: 40px;
  border-radius: 8px;
}
```

```
.promo-title {
  color: variables.$white;
  font-size: 2.5rem;
  margin-bottom: 10px;
  @extend %super-text-style;
}
```

```
.promo-text {
  color: variables.$white;
  font-size: 1.2rem;
  @extend %super-text-style;
}
```

```
.promo-button {
  background-color: variables.$white;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}
```

```
/* Secondary Version */
&.secondary {
  @include banner-theme(variables.$primary-orange);
}
}
```

style.scss

Online Doc

Extend/Inheritance



What's the color of "Summer Sale"?

```
<section class="promo-banner">
  <h1 class="promo-title">Summer Sale!</h1>
  <p class="promo-text">Get 50% off all items.</p>
  <button class="promo-button">Shop Now</button>
</section>

<section class="promo-banner secondary">
  <h1 class="promo-title">Clearance</h1>
  <p class="promo-text">Limited stock available.</p>
  <button class="promo-button">View All</button>
</section>
```

HTML

Key Point: It depends on where `%super-text-style` placeholder was originally defined. If it was defined before `.promo-title`, then white color will override it. Otherwise, the color defined in `%super-text-style` will take precedence.

Online Doc

```
/* This CSS will print because it is extended. */
%super-text-style {
  text-align: center;
  padding: 10px;
  color: black;
}
/* This CSS won't print because it is never extended.*/
%another-text-style {
  text-align: left;
  padding: 2px;
}
```

```
.promo-banner {
  @include banner-theme(variables.$primary-blue);
  padding: 40px;
  border-radius: 8px;

  .promo-title {
    color: variables.$white;
    font-size: 2.5rem;
    margin-bottom: 10px;
    @extend %super-text-style;
  }

  .promo-text {
    color: variables.$white;
    font-size: 1.2rem;
    @extend %super-text-style;
  }

  .promo-button {
    background-color: variables.$white;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
  }

  /* Secondary Version */
  &.secondary {
    @include banner-theme(variables.$primary-orange);
  }
}
```

style.scss

@mixin vs. @extend



What's the difference between them?

@mixin

@extend

Operators



Can we use relative values for the up-down and left-right padding of our button?

```
/* This CSS will print because it is extended. */
%super-text-style {
  text-align: center;
  padding: 10px;
  color: black;
}
/* This CSS won't print because it is never extended.*/
%another-text-style {
  text-align: left;
  padding: 2px;
}

.promo-banner {
  @include banner-theme(variables.$primary-blue);
  padding: 40px;
  border-radius: 8px;

  .promo-title {
    color: variables.$white;
    font-size: 2.5rem;
    margin-bottom: 10px;
    @extend %super-text-style;
  }

  .promo-text {
    color: variables.$white;
    font-size: 1.2rem;
    @extend %super-text-style;
  }

  .promo-button {
    background-color: variables.$white;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
  }

  /* Secondary Version */
  &.secondary {
    @include banner-theme(variables.$primary-orange);
  }
}
```

Operators

[Online Doc](#)

```
$button-padding-y: 10px;
/* This CSS will print because it is extended. */
%super-text-style {
  text-align: center;
  padding: 10px;
  color: black;
}
/* This CSS won't print because it is never extended.*/
%another-text-style {
  text-align: left;
  padding: 2px;
}

.promo-banner {
  @include banner-theme(variables.$primary-blue);
  padding: 40px;
  border-radius: 8px;

  .promo-title {
    color: variables.$white;
    font-size: 2.5rem;
    margin-bottom: 10px;
    @extend %super-text-style;
  }

  .promo-text {
    color: variables.$white;
    font-size: 1.2rem;
    @extend %super-text-style;
  }

  .promo-button {
    background-color: variables.$white;
    padding: $button-padding-y ($button-padding-y * 2);
    border: none;
    cursor: pointer;
  }

  /* Secondary Version */
  &.secondary {
    @include banner-theme(variables.$primary-orange);
  }
}
```

Practice: Sass

Use the online playground to convert the CSS code into Sass. Implement the following features:

- Inheritance: Create a `%flex-layout` placeholder to share the `display`, `align-items`, and `padding` between both boxes.
- Mixins: Create a `@mixin alert-theme($bg, $text)` to handle the background and text colors in one line.
- Nesting: Nest the `.alert-title` inside the `.alert-box`.
- Operators: Use a `$base-unit: 5px`; and set the `border-radius` and `margin-right` using multiplication.

