

The background is a solid dark teal color. In the top-left corner, there is a white square. A horizontal teal line extends from the left edge of the square across the top of the frame. In the top-right corner, there is a large teal circle partially overlapping a smaller white circle with a dark teal outline. A vertical teal line runs down the right edge of the frame.

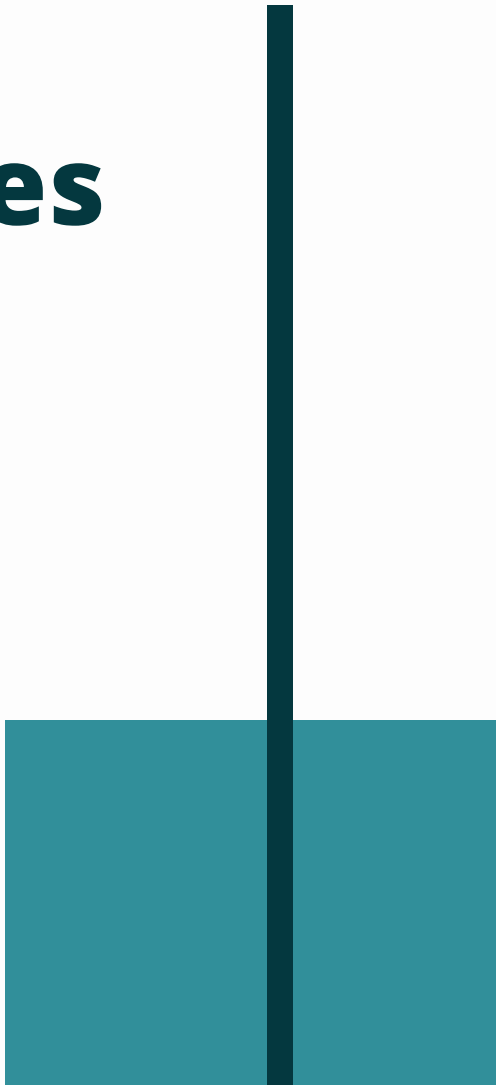
SASS

PABLO – KARINA

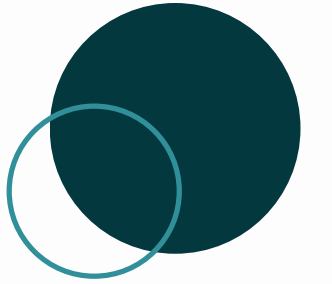
WHAT IS SASS?



- **CSS With superpowers**
- **Tool that allows you to extend the possibilities of CSS, providing various advantages for developers.**



FUNCTIONALITIES



- **It adds various functionalities on top of CSS that facilitate development, such as:**
 - **variables**
 - **Nesting**
 - **Mixins**
 - **partials**
 - **Inheritance**
 - **calculations**

LANGUAGES



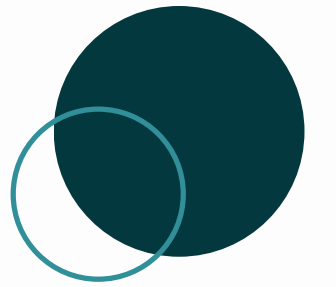
- Sass code can be provided in SCSS and Sass languages.
- The community prefers SCSS because of its very similar syntax to CSS code with its enhancements.

```
// SCSS
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

```
// SASS
$font-stack: Helvetica, sans-serif
$primary-color: #333
body
  font: 100% $font-stack
  color: $primary-color
```

VARIABLES



- You can store things like colors, font stacks, or any CSS values you think you'll want to reuse. Sass uses the \$ symbol to convert something to a variable

```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;  
  
body {  
  font: 100% $font-stack;  
  color: $primary-color;  
}
```

NESTING



- **Sass will allow you to nest your CSS selectors in a way that follows the same visual hierarchy as your HTML.**

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

PARTIALS



- **small bits of CSS that you can include in other Sass files.**
- **_partial.scss.**
- **@use rule. This rule loads another Sass file as a module**

```
// _base.scss
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}

// styles.scss
@use 'base';


.inverse {
  background-color: base.$primary-color;
  color: white;
}
```

MIXINS



A mixin allows you to make groups of CSS declarations that you want to reuse throughout your site.

```
@mixin theme($theme: DarkGray) {  
    background: $theme;  
    box-shadow: 0 0 1px rgba($theme, .25);  
    color: #fff;  
}  
  
.info {  
    @include theme;  
}  
  
.alert {  
    @include theme($theme: DarkRed);  
}  
  
.success {  
    @include theme($theme: DarkGreen);  
}
```



INHERITANCES

- Using **@extend** allows you to share a set of CSS properties from one selector to another.

```
/* This CSS will print because %message-shared is extended. */
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

// This CSS won't print because %equal-heights is never extended.
%equal-heights {
  display: flex;
  flex-wrap: wrap;
}

.message {
  @extend %message-shared;
}

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

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

.warning {
  @extend %message-shared;
  border-color: yellow;
}
```

OPERATORS



```
@use "sass:math";

.container {
  display: flex;
}

article[role="main"] {
  width: math.div(600px, 960px) * 100%;
}

aside[role="complementary"] {
  width: math.div(300px, 960px) * 100%;
  margin-left: auto;
}
```

ADOPTION



- **State of css 2022 - Most used preprocessors**

PRE-/POST-PROCESSORS

Main Answers

Other Pre-/Post-processors

Get JSON Data

Get GraphQL Query

Label	% of survey respondents	% of question respondents	Count
Sass	57.4%	87.5%	8206
PostCSS	36%	54.9%	5146
Less	15.6%	23.8%	2225
Stylus	4.2%	6.4%	597
Assembler CSS	0.4%	0.6%	47

Negative aspects:

- **Compile to CSS**
- **Complexity**

Positive Aspects:

- **Better maintainability**
- **Reuse components**
- **Maintenance improvements**

The word 'Sass' is written in a thick, pink, cursive script. The 'S' is large and loops around the 'a', which is also cursive. The 's' and 's' are smaller and follow the 'a'. The overall style is elegant and handwritten.

INSTALATION



- Two ways to use Sass
- Using Node.js via the console
- Via a vscode extension

Command to compile Sass

```
sass origen/index.scss build/index.css
```

```
npm install -g sass
```



Live Sass Compiler v5.5.1

Glenn Marks | 643,794 | ★★★★★ (48)

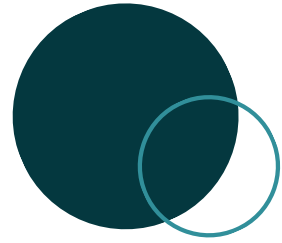
Compile Sass or Scss to CSS at realtime.

Disable



Uninstall



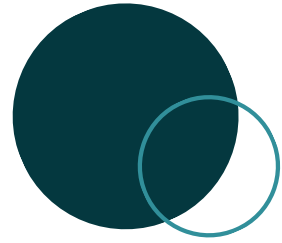


FLOW CONTROL DIRECTIVES

- @if – This directive allows you to write conditionals inside your Sass code. For example: @if \$variable == true { ... }
-
- @else – This directive is used in conjunction with @if to specify the actions that should occur if the condition is false. For example: @if \$variable == true { ... } @else { ... }

EXAMPLE

```
$color: blue;
.class {
  @if $color == blue {
    color: $color;
  } @else {
    color: red;
  }
}
```



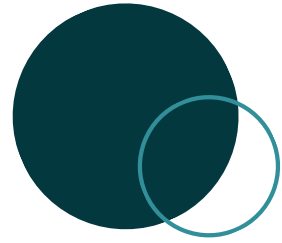
FLOW CONTROL DIRECTIVES

- @for – Allows you to iterate over a specified range of numbers.
-

For example: @for \$i from 1 through 10 { ... }

EXAMPLE

```
@for $i from 1 through 10
{
  .item-#{$i} {
    width: 100px * $i;
  }
}
```



FLOW CONTROL DIRECTIVES

- @each: allows you to iterate over a list or a map.
-

For example: @each \$item in \$list { ... }

EXAMPLE

```
$colors: red, blue, green;

@each $color in $colors {
  .#{$color}-bg {
    background-color: $color;
  }
}
```




FLOW CONTROL DIRECTIVES

- @while – Allows you to write a while loop in Sass.

•
For example: @while
\$i > 0 { \$i: \$i - 1; }

EXAMPLE

```
$i: 5;

@while $i > 0 {
  .item-#{ $i } {
    width: 100px * $i;
  }
  $i: $i - 1;
}
```

```
.item-5 {
  width: 500px;
}
.item-4 {
  width: 400px;
}
.item-3 {
  width: 300px;
}
.item-2 {
  width: 200px;
}
.item-1 {
  width: 100px;
}
```