

Sass

# ✨ What is Sass?

- ◆ Sass is a pre-processor for CSS
- ◆ We need a compiler
- ◆ Browsers currently don't support Sass/SCSS

# ✨ What can Sass do that CSS can't?

- ◆ Nested rules! (Meaning we can DRY it up!)
- ◆ Variables (That can be used in @for loops)
- ◆ Functions (You've heard me right 😎)
- ◆ Control Flow Statements
- ◆ Mixins

<https://github.com/MatheusBonavite/sass-kt>

# ✨ Nesting By Example

◆ Let's take a look at the following codepen:

<https://codepen.io/matheusbonavite/pen/g0xYMvX>

```
1 .panel{
2   font-size: 1rem;
3   border: 2px solid #f774a6;
4   border-radius: 0.25em;
5   > h2{
6     color: #f774a6;
7     font-weight: bold;
8     text-transform: uppercase;
9     font-size: 0.85em;
10    margin: 1em;
11    top: 0;
12    &::after{
13      content: " ";
14      display: table-cell;
15      border-bottom: 1px solid #f774a6;
16      width: 2.5em;
17    }
18  }
19  p{
20    margin: 1em;
21    color: #99004c;
22  }
23 }
```

*Sass*

```
1 .panel {
2   font-size: 1rem;
3   border: 2px solid #f774a6;
4   border-radius: 0.25em;
5 }
6 .panel > h2 {
7   color: #f774a6;
8   font-weight: bold;
9   text-transform: uppercase;
10  font-size: 0.85em;
11  margin: 1em;
12  top: 0;
13 }
14 .panel > h2::after {
15   content: " ";
16   display: table-cell;
17   border-bottom: 1px solid #f774a6;
18   width: 2.5em;
19 }
20 .panel p {
21   margin: 1em;
22   color: #99004c;
23 }
```



✨ ✨ There is more to nesting!

◆ Let's take a look at the following codepen:

<https://codepen.io/matheusbonavite/pen/j0LEQvY>

```
@mixin default-pink-button($default_font_color, $background_color){
  color: $default_font_color;
  background-color: $background_color;
}

$default_font_color: ■ #fff;
$default_pink: ■ rgb(209, 39, 181);
$default_background_pink: ■ rgb(221, 121, 204);

.btn{
  @include default-pink-button($default_font_color, $default_pink);
  margin: 1em;
  padding: 0.5em 1em;
  border-radius: 0.25em;
  &:hover{
    background-color: $default_background_pink;
  }
  &.btn2{
    &:hover{
      box-shadow: 5px 8px $default_background_pink;
    }
  }
}
```

*Sass*

```
.btn {
  color: ■ #fff;
  background-color: ■ #d127b5;
  margin: 1em;
  padding: 0.5em 1em;
  border-radius: 0.25em;
}

.btn:hover {
  background-color: ■ #dd79cc;
}

.btn.btn2:hover {
  box-shadow: 5px 8px ■ #dd79cc;
}
```



# ✨ Variables By Example!

◆ Let's take a look at the following codepen:

<https://codepen.io/matheusbonavite/pen/eYE0KNR>

```
1  $warning-color: #9F6000;
2  $warning-border: 1px solid #f8d7da;
3  $warning-background: #fff3f3;
4
5  $success-color: #4F8A10;
6  $success-border: 1px solid #d4edda;
7  $success-background: #d4edda;
8
9  .message{
10   border-radius: 0.25em;
11   border: $success-border;
12   background-color: $success-background;
13   > h3{
14     top: 0;
15     margin: 1em;
16     text-transform: uppercase;
17     font-size: 0.75em;
18     color: $success-color;
19   }
20   p {
21     margin: 1.25em;
22     color: $success-color;
23   }
24 }
```

*Sass*

```
1  .message {
2    border-radius: 0.25em;
3    border: 1px solid #d4edda;
4    background-color: #d4edda;
5  }
6  .message > h3 {
7    top: 0;
8    margin: 1em;
9    text-transform: uppercase;
10   font-size: 0.75em;
11   color: #4F8A10;
12 }
13 .message p {
14   margin: 1.25em;
15   color: #4F8A10;
16 }
```



# ✨ Functions By Example!

◆ Let's take a look at the following codepen:

<https://codepen.io/matheusbonavite/pen/QWMWrXW>

```
@use "sass:map";

2
3
4 @function panel-size($size: "default"){
5   $size-map: ("default": 1rem, "small": 0.75rem, "large": 2rem, "huge": 3rem);
6   @return map.get($size-map, $size);
7 }
8
9 .panel{
10   font-size: panel-size();
11   border: 2px solid #f774a6;
12   border-radius: 0.25em;
13   > h2{
14     color: #f774a6;
15     font-weight: bold;
16     text-transform: uppercase;
17     font-size: 0.85em;
18     margin: 1em;
19     top: 0;
20     &::after{
21       content: " ";
22       display: table-cell;
23       border-bottom: 1px solid #f774a6;
24       width: 2.5em;
25     }
26   }
27   p{
28     margin: 1em;
29     color: #99004c;
30   }
31 }
```

*Sass*

```
1 .panel {
2   font-size: 1rem;
3   border: 2px solid #f774a6;
4   border-radius: 0.25em;
5 }
6
7 .panel > h2 {
8   color: #f774a6;
9   font-weight: bold;
10  text-transform: uppercase;
11  font-size: 0.85em;
12  margin: 1em;
13  top: 0;
14 }
15
16 .panel > h2::after {
17   content: " ";
18   display: table-cell;
19   border-bottom: 1px solid #f774a6;
20   width: 2.5em;
21 }
22
23 .panel p {
24   margin: 1em;
25   color: #99004c;
26 }
27 }
```



# ✨ There is more to Functions!

◆ Some examples of Sass native functions are:

<code>rgb(\$red, \$green, \$blue)</code>	Creates a color from rgb values.
<code>rgba(\$red, \$green, \$blue)</code>	Creates a color from rgb values and uses alpha for opacity.
<code>red(\$color)</code>	Gets the red component of a color.
<code>green(\$color)</code>	Gets the green component of a color.
<code>blue(\$color)</code>	Gets the blue component of a color.
<code>mix(\$color1, \$color2, [\$weight])</code>	Mixes two colors together, giving their specific weights.
<code>adjust-hue(#63f, 60deg)</code>	Rotates the hue of a color by a certain number of degrees.
<code>darken(\$color, \$percent)</code>	Change the brightness of a color by subtracting a percentage.
<code>saturate(\$color, \$percent)</code>	Change the saturation of a color by a specified percentage amount.

# ✨ There is more to Functions!

◆ An example from the docs using @for:

SCSS Sass

```
@function pow($base, $exponent) {  
  $result: 1;  
  @for $_ from 1 through $exponent {  
    $result: $result * $base;  
  }  
  @return $result;  
}  
  
.sidebar {  
  float: left;  
  margin-left: pow(4, 3) * 1px;  
}
```

CSS

```
.sidebar {  
  float: left;  
  margin-left: 64px;  
}
```

◆ ...and there is still much more!

<https://sass-lang.com/documentation/modules/math>



# ✨ Mixins By Example!

◆ Let's take a look at the following codepen:

<https://codepen.io/matheusbonavite/pen/BadaqRB>

```
1 @use "sass:map";
2
3 @mixin message-border-type($type: "success"){
4   $color: ("warning": #9F6000, "success": #4F8A10);
5   $border: ("warning": 1px solid rgb(254, 239, 180), "success": 1px solid rgb(79, 138, 16));
6   $background-color: ("warning": rgba(254, 239, 180, 0.55), "success": rgba(79, 138, 16, 0.25) );
7
8   color: map.get($color, $type);
9   border: map.get($border, $type);
10  background-color: map.get($background-color, $type);
11 }
12
13 .message{
14   @include message-border-type("success");
15   border-radius: 0.25em;
16   > h3{
17     top: 0;
18     margin: 1em;
19     text-transform: uppercase;
20     font-size: 0.75em;
21     color: inherit;
22   }
23   p {
24     margin: 1.25em;
25     color: inherit;
26   }
27 }
```

```
1 .message {
2   color: #4F8A10;
3   border: 1px solid #4f8a10;
4   background-color: rgba(79, 138, 16, 0.25);
5   border-radius: 0.25em;
6 }
7
8 .message > h3 {
9   top: 0;
10  margin: 1em;
11  text-transform: uppercase;
12  font-size: 0.75em;
13  color: inherit;
14 }
15
16 .message p {
17   margin: 1.25em;
18   color: inherit;
19 }
20 }
```

*Sass*



# ✨ There is more to Mixins!

✦ An example from the docs:

SCSS

Sass

```
@mixin order($height, $selectors...) {  
  @for $i from 0 to length($selectors) {  
    #{nth($selectors, $i + 1)} {  
      position: absolute;  
      height: $height;  
      margin-top: $i * $height;  
    }  
  }  
}  
  
@include order(150px, "input.name", "input.address", "input.zip");
```

CSS

```
input.name {  
  position: absolute;  
  height: 150px;  
  margin-top: 0px;  
}  
  
input.address {  
  position: absolute;  
  height: 150px;  
  margin-top: 150px;  
}  
  
input.zip {  
  position: absolute;  
  height: 150px;  
  margin-top: 300px;  
}
```

# ✨ There is more to Mixins!

✦ An example from the docs:

SCSS

Sass

```
@mixin hover {  
  &:not([disabled]):hover {  
    @content;  
  }  
}  
  
.button {  
  border: 1px solid black;  
  @include hover {  
    border-width: 2px;  
  }  
}
```

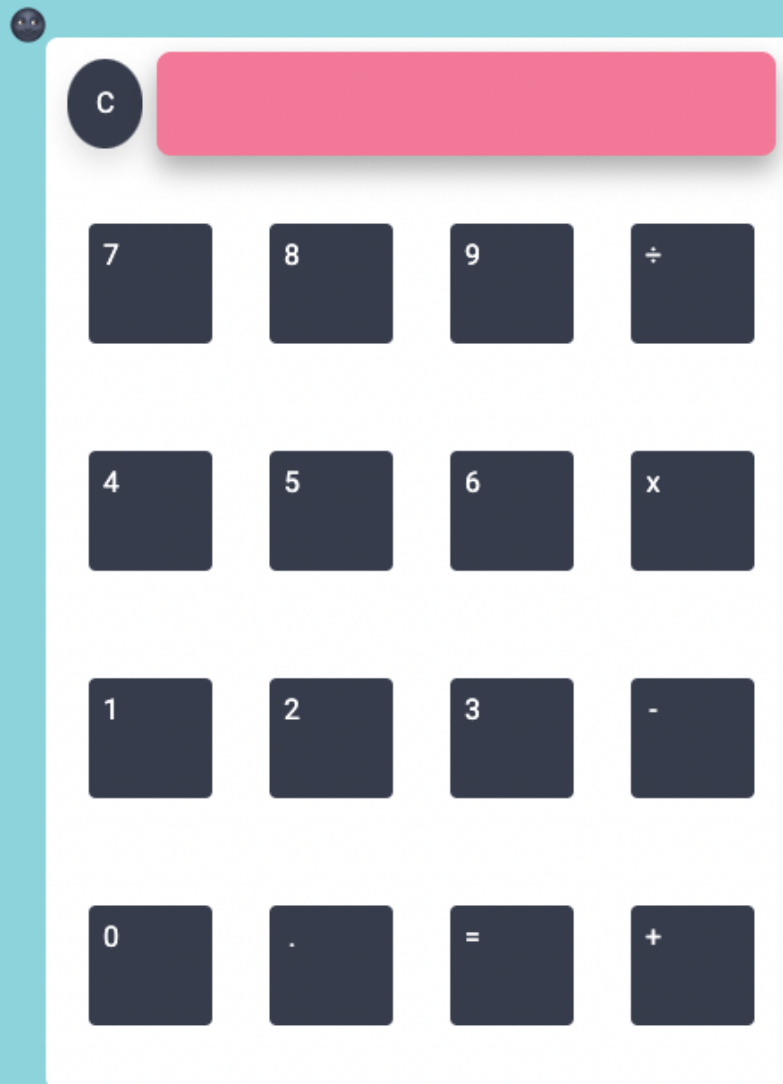
CSS

```
.button {  
  border: 1px solid black;  
}  
.button:not([disabled]):hover {  
  border-width: 2px;  
}
```

✦ ...and there is much more!

✨✨ Lets look at a final example!

✦ Taken from <https://github.com/MatheusBonavite/sass-kt>:



# ✨References:

- ◆ CSS in Depth - Keith J. Grant (2018)
- ◆ Mike North's classes on FrontendMasters
- ◆ <https://sass-lang.com/documentation>
- ◆ <https://jstutorial.medium.com/the-complete-guide-to-scss-sass-49ac053fcde5>
- ◆ <https://betterprogramming.pub/an-introduction-to-sass-scss-fdbda159b40>

✨ Thank you ☕