# FULL STACK

MEETING 02 | SASS / SCSS

# WHAT IS SASS / SCSS

**S**yntactically **A**wesome **S**tyle**S**heets: An extension for CSS

**S**assy **CSS**: Sass with CSS-like syntax (as opposed to Sass's Ruby syntax)

## FIRST STEPS

SASS can be installed as a Ruby gem:

```
gem install sass
```

**Watch file:**

To have your .sass files change into .css files for web browsers:

```
sass --watch input.scss:output.css
```

or for directories:

```
sass --watch /sass:/css
```

## THE BASICS

1. SASS uses indentations for its syntaxes and has no braces and no semicolons
2. SCSS is written in much the same was as CSS, with brackets and semicolons

## Example - SASS

CSS:

```
body {
        max-width: 650px;
        color: #444444;
        background-color: #EEEEEE;
        line-height: 1.4;
        margin: 40px auto;
}
```

SASS:

```
$max-width:  650px
$font-color:   #444444
$bg-color:     #EEEEEE

body
        max-width: $max-width
        color: $font-color
        background-color: $bg-color
        line-height: 1.4
        margin: 40px auto
```

## Example - SCSS

CSS:

```css
body {
	max-width: 650px;
	color: #444444;
	background-color: #EEEEEE;
	line-height: 1.4;
	margin: 40px auto;
}
```

SCSS:

```scss
$max-width: 650px;
$font-color:  #444444;
$bg-color:    #EEEEEE;

body {
	max-width: $max-width;
	color: $font-color;
	background-color: $bg-color;
	line-height: 1.4;
	margin: 40px auto;
}
```

# Nesting in SASS

SASS:

```
body
    max-width: 650px;
    background-color: $bg-color
    color: $bg-color
    .header
        color: $header-color
        font-family: $font-stack
```

CSS:

```
body {
    max-width: 650px;
    background-color: #FFFFFF;
    color: #000000; }
    body .header {
        color: #DDDDDD;
        font-family: “Open Sans”; }
```

## The Parent Selector (&)

```
a {
  font-weight: bold;
  text-decoration: none;
  &:hover { text-decoration: underline; }
}
```

```
a { … }
    a:hover { …}
```

```
#main {
    color: black;
    &-sidebar { border: 1px solid; }
        &:hover {darken( $sidebar-color, 10%);
    }
}
```

```
#main {...}
    #main-sidebar {...}
        #main-sidebar:hover{...}
```

# Functions built in SASS

http://sass-lang.com/documentation/Sass/Script/Functions.html

Values are comma separated:

Examples:

| | | |
|---|---|---|
| darken ($color, $value) | darken (#DD88AA, 10%) | #d2608e |
| invert ($color) | invert(#DD88AA) | #227755 |
| complement($color) | complement(#DD88AA) | #88ddbb |
| round($number) | round($margin / $colums) | 60 |

# @import

```
_default.scss:

$footer-color: #878787;

body {
        color: #DDDDDD;
        line-height: 1.4;
}
```

```
main.scss:

@import _default;

footer {
        background-color: $footer-color;
        color: white;
}
```

```
main.css:

body {
        color: #DDDDDD:
        line-height: 1.4;
}

footer {
        background-color: #878787;
        color: white;
}
```

## @include

includes are used for @mixins. Mixins can be thought of as abstract objects in CSS.

**Declaring a mixin:**

```
@mixin color ($color, $bg-color) {
	color: $color;
	background-color: $bg-color;
	line-height: 1.4;
	border: 1px solid black;
}
```

**Using a mixin:**

```
#sidebar {
	@include color (#DDDDDD, #DD88AA)
	width: 300px;
}
```

# Math and Functions

Functions are useful for math operations. Use them to avoid hard-coding things, so they can be dynamically changed. Functions can only return one value.

**Declaring a function:**

```
@function width-percent ($container, $object) {
	@return ($container / $object) * 100%;
{
```

**Using a function:**

```
.sidebar {
	width: $sidebar-width;
	&-buttons {
		width: width-percent ($sidebar-width, 60px);
	}
}
```