# Introduction to Sass

Luca Carotenuto

# Content

- Introduction to Sass
- Two Syntaxes
- Mechanisms
  - Variables
  - Nesting
  - Mixins
  - Partials
  - Inheritance
- Demo (Sass in use)

# Introduction to Sass

- **S**yntactically **A**wesome **S**tyl**e** **S**heets
- CSS preprocessor
- Compiles Sass Code into CSS Code

  => syntactic sugar
- official implementation is open-source, coded in Ruby

# Two Syntaxes

- Original Syntax: the indented syntax
  - Indention to seperate code blocks
  - Newline to seperate rules
  - => File Extension .sass
- Newer Syntax: SCSS
  - braces for code blocks
  - semicolons to seperate lines
  - => File Extension .scss

# Mechanisms of Sass

# Variables

- Variables have scope based on where they are defined!
- Date Types:
  - Numbers (units)
  - Strings (with quotes or without)
  - Colors
  - Booleans

```
$width: 5em;
$font-stack: Helvetica, sans-serif;
#primary-color: #333;

#main {
        width: $width;
        font 100% $font-stack;
        color: $primary-color;
}
```

# Nesting

- Let’s you nest CSS selectors
- same visual hierarchy of HTML
- better readability than:

body #main .value a.link … …

# Nesting

```
body #main .value {
    color: #eee;
}

body #main .value a.link {
    color: #101010;
    text-decoration: none;
}
```

=> pure CSS

```
body #main .value {
    color: #eee;

    a.link {
        color: #101010;
        text-decoration: none;
    }
}
```

=> SASS

# Partials

- contain snippets of CSS that can be included in other Sass files
- help modularizing the code
- Sass file named with leading underscore _partial.scss

  => lets Sass know that the file is a partial and should not be generated into a CSS file
- partials are used with @import

# Partials

```
stylesheets/
|
|-- partials/            # Partials
|   |-- _base.sass       # imports for all mixins + global project variables
|   |-- _buttons.scss    # buttons
|   |-- _figures.scss    # figures
|   |-- _grids.scss      # grids
|   |-- _typography.scss # typography
|   ...
`-- main.scss            # primary Sass file
```

```
// Partials
@import "partials/typography";
@import "partials/buttons";
@import "partials/figures";
@import "partials/grids";
// ...
```

# Mixins

- chunk up CSS declarations to be reusable with one reference
- Mixins can reference mixins as well
- mixin lets you make groups of CSS declarations that you want to reuse
- you can pass values to make them flexible

```
=border-radius($radius)
  -webkit-border-radius : $radius
  -moz-border-radius :    $radius
  -ms-border-radius :     $radius
  border-radius:          $radius

.box
  +border-radius(10px)
```

# Inheritance

- share CSS properties with @extend
- avoid writing same CSS code multiple times
- cleaner code

```
.message {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

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

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

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

# … and a whole lot more

- functions: ex. darken($color, $amount)
- controls
  - if
  - while
  - each
  - for
- Operators +, -, *, /, and %

# Demo

Sass in Use