SCALABLE AND MAINTAINABLE CSS/SASS

Eckhard Rotte, (@erotte)
appfertigung.com

SO, WHATS WRONG WITH SASS?

Nothing.

BUT:

- Our apps constantly increase in complexity (Blah…)
- responsive, mobile etc.
- CSS still sucks
- everybody is still doing their own thing

SASS/COMPASS

- No constraints. No rules.
- It's very easy to create complex bullshit SASS/CSS!

SHITTY CODE ALERT!

```
#page
  #list
    +column(18,true)
    #list-header
      text-align: right
      .view-actions
        float: right
        a
          +inline-block
          height: 25px
          &#action-flat
            background: transparent image-url('bg-tabs.png')
            &.active
              background-position: 0 0
            &:hover
              background-position: 0 -50px
          &#action-grouped
            background: #cde
```

```
#page #list #list-header .view-actions a#action-flat.active:hover
{
  background-position: 0 0
}
```

SMACSS

Scalable and Modular Architecture for CSS

by Jonathan Snook (@snookca)

- coding and naming conventions
- a style guide
- a guidelines framework
- classification system

CATEGORIZING CSS RULES

- Base
- Layout
- Modules
- State
- (Theme)

BASE

- basic element styles
- the defaults
- resets
- should not contain nested rules

```
html, body, form {
  margin: 0;
  padding: 0; }

input[type=text] {
  border: 1px solid #cde;
}

a { color: #039; }
a:hover { color: #03C; }
```

LAYOUT

- (outer) layout container styles
- page sections
- grid
- usually: single selectors, #ids
- may include contextual selectors

```
#header, #footer {
  width:  auto;
  margin  0 auto;
}
.l-fixed {
  #header, #footer {
    width:  960px;
  }
}
```

MODULES

- are reuseable, modular, fragments, components, sections, widgets
- e.g. buttons, a product list, sidebar nav or teaser box

STATE

- hidden or visible?
- collapsed or expanded?
- emphasized or quiet?

(THEMES)

...

EXAMPLE FILE STRUCTURE

```
+ application.sass                    // @imports
+ base/
|   _settings.css.sass                // SASS config variables
|   _reset.css.sass
|   _colors.css.sass
|   _element_defaults.css.sass
+ layout/
|   _settings.css.sass                // SASS layout/grid variables
|   _containers.css.sass
+ modules/
+ other/
```

http://railslove.com/blog/2012/03/28/smacss-and-sass-the-future-of-stylesheets/

SOME CONVENTIONS AND RULES

- use classes
- avoid #ids
- avoid element selectors
- avoid nesting
- .namespace-your-selectors
- semantic markup is overrated
- create modules
- create more modules!
- ...

EXAMPLE

```
/* a layout rule */
.layout-fixed {...};

/* you may shorten the prefix */
.l-fixed {...};

/* a module. no prefix needed! */
.tab {...};

/* a module with state */
.tab.is-active {...}
```

COMPONENTS

parts of a module

```
.message {
  border: 1px solid #333;

  .message-header {
    font-weight: bold;
  }

  .message-body
    font-size: 14px
  }
}
```

CSS

```
.searchbox {
  width: 100%;
}
.searchbox input[type=text] {
    width: 50%;
}
```

SCSS

```
.searchbox {
  width: 100%;
  input[type=text] {
    width: 50%;
  }
}
```

CHALLENGE

we need a variant of .searchbox in the Sidebar, with a full width input field.

What's your first instinct?

COMMON SOLUTION:

```
#sidebar .searchbox input[type=text]{
    width: 100%;
}
```

so far so good.

Next exercise: we need another variant with a fixed width!

NESTED RULES

SPECIFITY WARS START HERE

```
.searchbox {
  width: 100%;
  input[type=text] {
    width: 50%;
  }
}
#sidebar .searchbox input[type=text]{
    width: 100%;
}
.searchbox-fixed {
  width: 200px;
}
.searchbox-fixed input[type=text],
#sidebar .searchbox-fixed input[type=text] {
  width: 150px;
}
```

SUB-MODULES

```
.searchbox {
  width: 100%;
  input[type=text] {
    width: 50%;
  }
}
.searchbox-outlined{
    border: 1px solid #333
}
.searchbox-fixed {
  width: 200px;
}
.searchbox-fixed input[type=text] {
  width: 150px;
}
```

```
<div class="searchbox searchbox-fixed">
  <input type="text" ... />
</div>
```

- Markup moves fast. Try to avoid conditional styling based on location
- create a Sub-Module of the module instead.

SUB-CLASSING

CSS

```
.searchbox.searchbox-fixed {
  width: 200px;
}
```

SCSS

```
.searchbox {
  &.searchbox-fixed {
    width: 200px;
  }
}
```

SASS

```
.searchbox
  &.searchbox-fixed
    width: 200px;
```

A SASS MODULE

```
.searchbox
  width: 100%

  input[type=text]
    width: 50%

  &.searchbox-fixed
    width: 200px
    input[type=text]
      width: 150px

  &.searchbox-outlined
    border: 1px solid red
```

SMURF

Scalable, Modular, reUsable Rails Frontends

by Jakob Hilden (@jkwebs)

- additional coding conventions for SMACSS
- a Ruby gem called "Smurfville" for generating live styleguides
- more accurate definitions of submodules, components and modifiers

EXAMPLE

```
// -- module --
.m-box
  // components
  .m-box--header
  .m-box--body

  // modifiers
  &.no-border
  &.right

  // states
  &.is-disabled

// -- submodule --
.m-box_attention
// additional component, that only exists in the submodule
  .m-box_attention--teaser
```

http://railslove.com/blog/2012/11/09/taking-sass-to-the-next-level-with-smurf-and-extend

WORKING WITH SASS EXTENDABLES

collecting selectors

SASS

```
.button
  color: #333
.button-red
  @extend .button
  background: red
  color: white
```

generated CSS

```
.button, .button-red {
  color: #333333; }

.button-red {
  background: red;
  color: white; }
```

@extend

- + you can re-use code
- + smaller CSS output
- – may blow up your selectors to gigantic, comma separated selector stacks
- – is not transparent, everything may get inherited
- – has some strange side effects with complex selectors

%PLACEHOLDER SELECTOR

won't be compiled until it's @extended

```
%clearfix
  overflow: none;
  *zoom: 1;

aside, footer
  @extend %clearfix;

#grid-container
  @extend %clearfix;
```

produces

```
aside, footer, #grid-container {
  overflow: none;
  *zoom: 1;
}
```

SASS VARIABLES

SHITTY CODE ALERT AGAIN!

```
// - buttons
$button-font-size: $base-font-size;
$button-light-color: #5f737d;
$button-background-color: #1a3744;
$button-text-color: white;
$search-button-background-color: $accent-color;
$search-button-light-color: #48BFE3;
$search-button-text-color: white;
$admin-button-light-color: #adbabe;
$admin-button-background-color: #8c9ea4;
$admin-button-text-color: white;
//
$neighbours-background-color: $button-background-color;
$neighbours-separator-color: $button-light-color;
$footer-text-color: $medium-gray;
$net-rating-border-color: #d2d2d2;
//
$pager-text-color: #666;
$pager-color: #b3b3b3;
$pager-highlight-color: #4c4c4c;
```

SASS/COMPASS COLOR MANIPULATION IS GREAT!

```
// Makes a color lighter.
+lighten($color, $amount)

// Makes a color darker.
+darken($color, $amount)

+saturate($color, $amount)
+desaturate($color, $amount)
+mix($color-1, $color-2, [$weight])
```

You need to now details about the design- and color concept!

TRY TO DEFINE BASE VARIABLES ONLY.

move specific variables into the modules

```
$primary-color
$secondary-color
$spot-color
$text-color

$base-gap
$grid-width
```

You need to now details about the design- and color concept!

2. USE LOCAL VARIABLES

```
// config

// somewhere in the grids...
$base-gap: 10px

// colors
$ci-blue: #0F559A
$spotcolor: #faaaee
```

```
%button
  $bg-blue: $ci-blue
  $bg-blue-dark: darken($ci-blue, 20)
  padding: 2px $base-gap*1.5
  +background-image(linear-gradient($bg-blue, $bg-blue-dark))

.button
  @extend %button
  &.button-red
    $red-start: shade($spot-color, 16)
    $red-stop: darken($spot-color, 26)
    +background-image(linear-gradient($red-start, $red-stop))
```

IT MAY GET A KIND OF DOCUMENTATION

```
.my-class {
  border: 1px red solid;
  height: 244px;
  padding: 6px;
  width: 368px;
}
```

vs.

```
.my-class {
  $border-width: 1px;
  $padding: 6px;
  $height: 256px
  $width: 380px;

  border: $border-width red solid;
  height: $height - 2 * ($padding + $border-width);
  padding: $padding;
  width: $width - 2 * ($padding + $border-width);
}
```

http://www.hagenburger.net/BLOG/Document-Your-CSS-Sass-With-Variables.html

JS BINDINGS, THE RAILS WAY

(lazy slide)

```
= link_to 'Do it!', '#', class: 'btn btn-red',
  data: {behaviour: 'open-dialog'}
```

```
<a href="#" class="btn btn-red" data-behavior="open-dialog">
```

```
// jQuery helper
$.behaviour = function(behaviour, context) {
  return $('[data-behaviour~='+behaviour+']', context)
}
```

```
$.behaviour('open-dialog').click(function (event) {
  event.preventDefault();
  //... open the dialog!
}));
```

KUDOS TO

Jonathan Snook(@snookca)

Jakob Hilden(@jkwebs)

Nico Hagenburger(@Hagenburger)

Examples in this Presentation are mostly shameless clones or stolen!

QUESTIONS?