Preprocessed Stylesheets the Sassy Way

Jason Porter, @lightguardjp

Preprocessed Stylesheets the Sassy Way

Jason Porter, @lightguardjp

Brandon Mathis for some content

http://brandonmathis.com/

 @imathis


css today


CSS
IS
AWESOME

CSS

css can be

difficult

It grows

Repetition

No variables

There's no logic

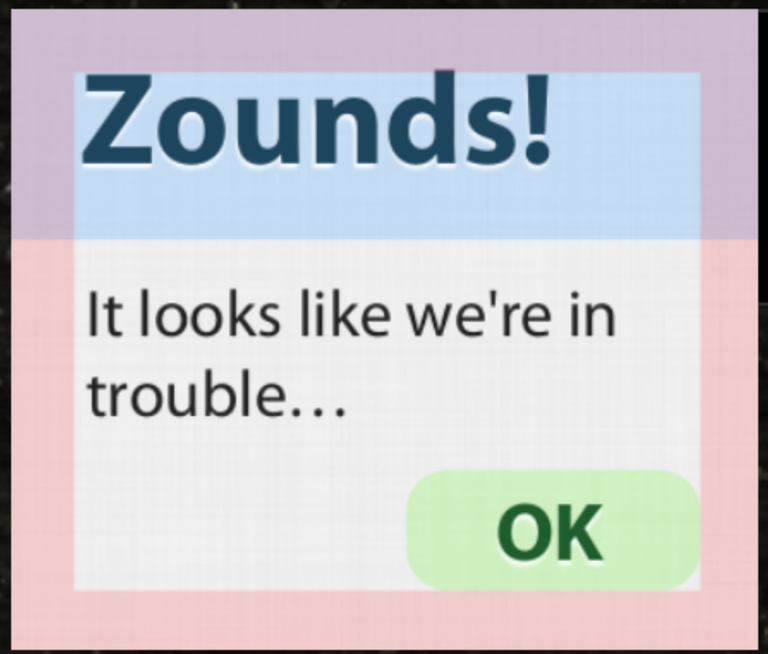
Hard to remember

No Math

CSS

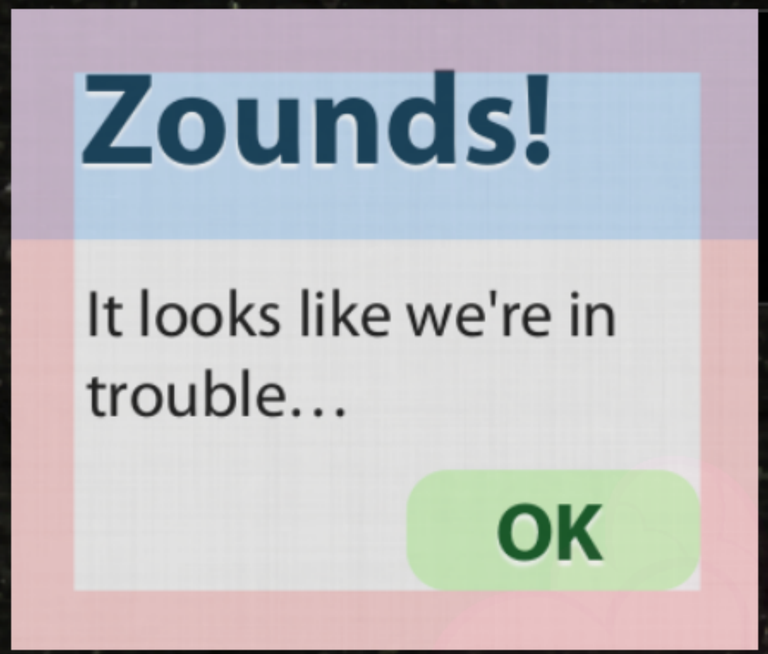
```css
@media screen and (max-width: 480px)
{
    h3 {
      font-size: 2rem;
    }
}
h1, h2, h3, h4, h5, h6 {
  font-family: "proxima-nova-soft", "Helvetica Neue", Helvetica, Arial, sans-serif;
  font-weight: 100;
  line-height: 1.5;
}
h3 {
  font-size: 3rem;
  font-weight: 800;
}
```

NOT DRY, ISOLATED, INTENT?




```
.msg { padding: 24px; }
.msg h3 {
  padding: 24px;
  margin: -24px -24px 0;
}
```

NOT DRY, ISOLATED, INTENT?




```
.msg { padding: 24px; }
.msg h3 {
  padding: 24px;
  margin: -24px -24px 0;
}
```


WAT??

Sass

> Sass extends CSS3 with variables, math, mixins & more. But at its core, Sass is a layer of empathy between the designer and the stylesheets.
>
> — Brandon Mathis

WHAT SASS LOOKS LIKE

scss

```scss
article {
  margin-bottom: 2em;
  .entry-content {
    border-top: 1px solid $grey;
  }
}
```

sass

```sass
article
  margin-bottom: 2em
  .entry-content
    border-top: 1px solid $grey
```

variables

Colors, numbers, or text

Colors, numbers, or text

Understands units (em, rem, px, etc)

Colors, numbers, or text

Understands units (em, rem, px, etc)

easy — $name: value;

VARIABLES

SCSS

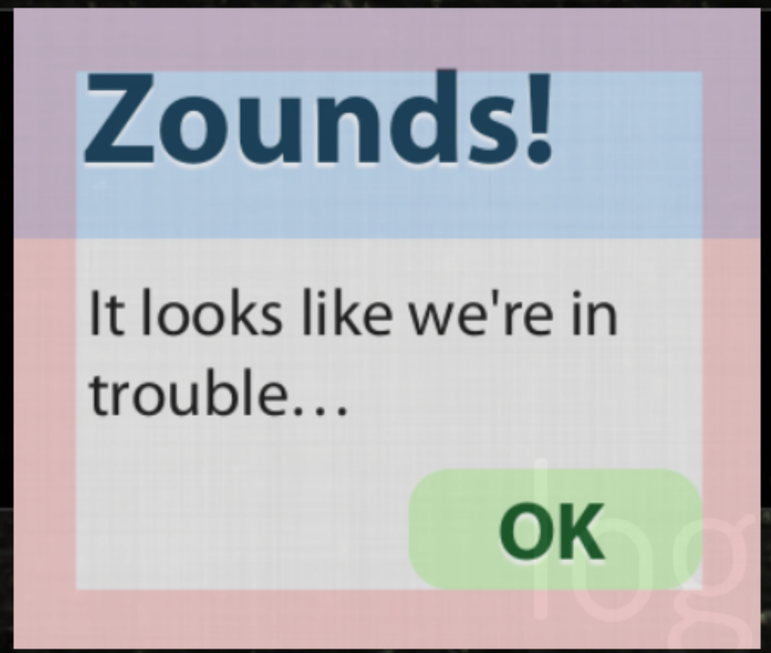
```scss
$link-color: blue;
$link-hover: red;

a {
  color: $link-color;
  &:hover { color: $link-hover; }
}
```

CSS

```css
a { color: blue; }
a:hover { color: red; }
```

We now have **context**

```scss
$msg-pad: 24px;
.msg {
  padding:$msg-pad;
  h3 {
    padding:$msg-pad;
    margin:(-$msg-pad)
           (-$msg-pad) 0;
  }
}
```



logic

RELATIONAL OPERATORS (FOR NUMBERS)

SCSS

```
1 < 20
10 <= 20
4 > 1
4 >= 1
```

< > ⇐ ⇒ == !=

@if, @else, @else if

and, or

RELATIONAL OPERATORS (FOR NUMBERS)

SCSS

```
1  < 20  // true
10 <= 20 // true
4  > 1   // true
4  >= 1  // true
5  > 5   // false
```

COMPARISON OPERATORS FOR EVERYTHING

SCSS

```scss
1 + 1 == 2    // true
small != big  // true
#000 == black // true
```

SCSS

```scss
red == #f00
red == #ff0000
red == rgb(255, 0, 0)
red == rgba(255, 0, 0, 1.0)
red == hsl(0deg, 100%, 100%)
red == hsla(0deg, 100%, 100%, 1)
```

CONDITIONALS

SCSS

```scss
$theme: ocean;
div {
  @if $theme == dusty {
    background: #c6bba9;
    color: $color;
  } @else if $theme == ocean {
    background: blue;
    color: white;
  }
}
```

TERNARY

SCSS

```scss
$main-bg: #000;
.main {
  color: if($main-bg == black, #fff, #000);
}
```

math

EXACTLY AS
YOU'D EXPECT

SCSS

```scss
1em + 1em;  // 2em
1em - 1em;  // 0em
1in + 72pt; // 2in
6px * 4;    // 24px
18  % 5;    // 3
```

SCSS

```scss
$container : 960px;
$main : 680px;
$gutter : 30px;

#sidebar {
  width: $container - $main - $gutter;
}
```

CSS

```css
#sidebar {
  width: 250px;
}
```

```scss
$font: 18px / 1.45em;  // 18px/1.45em
$font: (20px / 3);     // 4px
$font: 20px / 5 + 1px; // 5px
$font: $base / 5;      // 4px
$font: 20px / 5;       // 4px
```

except for division

SCSS

```scss
font  : 18px / 1.45em; // 18px/1.45em
font  : (20px / 5);    // 4px
font  : 20px / 5 + 1;  // 5px
font  : $base / 5;     // 4px
$size : 20px / 5;      // 4px
```

NUMBER FUNCTIONS

SCSS

```
percentage(13/25) // 52%
round(2.4)        // 2
ceil(2.2)         // 3
floor(2.6)        // 2
abs(-24)          // 24
```

looping

@FOR

SCSS

```scss
@for $level from 0 to 5 {
  .tag-#{$level + 1} {
    font-size: .7em + ($level * .5em);
  }
}
```

CSS

```css
.tag-1 { font-size: 0.7em; }
.tag-2 { font-size: 1.2em; }
.tag-3 { font-size: 1.7em; }
.tag-4 { font-size: 2.2em; }
.tag-5 { font-size: 2.7em; }
```

@WHILE

SCSS

```scss
$level: 0;

@while $level < 5 {
  .tag-#{$level + 1} {
    font-size: .7em + ($level * .5em);
  }
  $level: $level + 1;
}
```

CSS

```css
.tag-1 { font-size: 0.7em; }
.tag-2 { font-size: 1.2em; }
.tag-3 { font-size: 1.7em; }
.tag-4 { font-size: 2.2em; }
.tag-5 { font-size: 2.7em; }
```

@EACH

SCSS

```scss
$animals: puma, crab, emu, duck;

@each $animal in $animals {
  .#{$animal}-icon {
    background: url('/images/#{$animal}.png');
  }
}
```

CSS

```css
.puma-icon { background: url('/images/puma.png'); }
.crab-icon { background: url('/images/crab.png'); }
.emu-icon { background: url('/images/emu.png'); }
.duck-icon { background: url('/images/duck.png'); }
```

nesting

SCSS

```scss
article {
  border-top: 1px dashed #eee;
  header { margin-bottom: 1.5em; }
}
```

CSS

```css
article { border-top: 1px dashed #eee; }
article header { margin-bottom: 1.5em; }
```

SCSS

```scss
article {
  header, section { margin-bottom: 1.5em; }
}
```

CSS

```css
article header, article section {
  margin-bottom: 1.5em;
}
```

NESTING (SYMBOL SELECTORS)

SCSS

```scss
article {
  > h2 { border-top: 1px dashed #eee }
  ~ article { padding-top: 1.5em }
  + footer { margin-top: 0 }
  * { color: #000 }
}
```

CSS

```css
article > h2 { border-top:1px dashed #eee }
article ~ article { padding-top: 1.5em }
article + footer { margin-top: 0 }
article * { color: #000 }
```

PARENT SELECTOR

SCSS

```scss
a {
  color: blue;
  &:hover { color: red } (1)
  display: inline-block;
  line-height: 1.8em;
}
```

1selector: &

CSS

```css
a { color: blue; display: inline-block; line-height: 1.8em; }
a:hover { color: red } (1)
```

1output

mixins

Reusable chunks of css

Doesn't output anything in not used

May contain arguments

BASIC MIXIN

SCSS

```scss
@mixin hover-link {
  text-decoration: none;
  &:hover { text-decoration: underline; }
}
nav a { @include hover-link; }
```

CSS

```css
nav a { text-decoration: none; }
nav a:hover { text-decoration: underline; }
```

MIXIN WITH ARGS

SCSS

```scss
@mixin border-radius($amount) {
  border-radius: $amount;
  -webkit-border-radius: $amount;
  -moz-border-radius: $amount;
}
.msg { @include border-radius(5px); }
```

CSS

```css
.msg {
  border-radius: 5px;
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
}
```

DEFAULTS AND NAMED ARGUMENTS

SCSS

```scss
@mixin link-color($text:blue, $hover:red) {
  color: $text;
  &:hover { color: $hover; }
}
a {
  @include link-colors($hover: green);
}
```

CSS

```css
a { color: blue; }
a:hover { green; }
```

MIXINS AND BLOCKS

SCSS

```scss
@mixin apply-to-ie6-only {
  * html {
    @content;
  }
}
@include apply-to-ie6-only {
  #logo {
    background-image: url(/logo.gif);
  }
}
```

CSS

```css
* html #logo {
  background-image: url(/logo.gif);
}
```

importing

extends CSS **`@import`**
by default looks for sass file
may fall back to css import — see docs
multiples on one statement

PARTIALS

source tree

```
+--- style
|          _partials
|                _grid.scss
|                _typography.scss
|                _nav.scss
|          style.scss
```

style.scss

```
@import "_partials/grid", "_partials/typeograph", "_partials/nav"
```

output tree

```
+--- style
|          style.css
```

colors

RGBA FUNCTION

SCSS

```scss
a { color: rgba(blue, .75) }
p { background: rgba(#ffa, .25) }
```

CSS

```css
a { color: rgba(255, 255, 170, 0.25) }
p { background: rgba(255, 255, 170, 0.25) }
```

INSPECTING COLORS

SCSS

```scss
$color: red;

hue($color); // 0deg
saturation($color); // 100%
lightness($color); // 50%

red($color); // 100
green($color); // 0
blue($color); // 0

alpha($color); // 100
```

MANIPULATING COLORS

invert(blue)

complement(#6cf620)

adjust-hue(#77a7f9, 90)

adjust-hue(#77a7f9, -90)

mix(red, yellow)

mix(red, yellow, 30)

saturate(#9b8a60, 50%)

desaturate(#d9a621, 50%)

grayscale(yellow)

darken(#6cf620, 30%)

lighten(#2e7805, 30%)

fade-in(rgba(#fab,.5), .5)

fade-out(#fab, .5)

functions

SCSS

```scss
@function pxem($px, $context: 16px) {
  @return ($px / $context) * 1em;
}

article h2 { font-size: pxem(45px); }
```

CSS

```css
article h2 { font-size: 2.813em; }
```

FUNCTIONS IN PRACTICE

SCSS

```
@function text-contrast($bg,
  $light:#fff, $dark:#000) {
  $darkbg:lightness($bg) < lightness(gray);
  @return if($darkbg, $light, $dark);
}
@mixin easy-button($bg){
  color: text-contrast($bg);
  background: linear-gradient(
  lighten($bg, 8), darken($bg, 8));
}
button { @include easy-button(blue); }
```

debugging

USE THE
SOURCE [MAPS]

Must use 3.3.0 (currently at RC 2)

`--sourcemap`

Web-kit browsers

FireSass in Firefox

compass


WHAT DO WE
GET?

Mixin library

Extensions

Sass functions

Environment awareness

adjust-lightness, scale-lightness
adjust-saturation, scale-saturation
image-url
image-height
image-width
inline-image
font-url

HELPER FUNCTIONS (CONT)

inline-font-files
pi
sin
cos
tan
more...

HELPER FUNCTIONS IN PRACTICE

SCSS

```scss
header {
  background: image-url('hbg.png');
  h1 {
    width: image-width('logo.png');
    height: image-height('logo.png');
    background: inline-image('logo.png')
  }
}
```

CSS

```css
header {
  background: url('/images/hbg.png?1351...');
}
header h1 {
  width: 220px; height: 100px;
  background: url('data:image/png;base64...
}
```

Compass Mixins

General Utilities

Element Styles

Design Patters

CSS3

CSS3 MIXINS

SCSS

```scss
.msg {
  @include background(linear-gradient(#fff, #eee));
  @include border-radius(5px);
}
```

CSS

```css
.msg {
  background: -webkit-gradient(linear, 50% 0%, 50% 100%,
  color-stop(0%, #ffffff), color-stop(100%, #eeeeee));
  background: -webkit-linear-gradient(#ffffff, #eeeeee);
  background: -moz-linear-gradient(#ffffff, #eeeeee);
  background: -ms-linear-gradient(#ffffff, #eeeeee);
  background: linear-gradient(#ffffff, #eeeeee);
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
  -ms-border-radius: 5px;
  border-radius: 5px;
}
```

Sprites

pngs
Based on image name
Does all the work for you

SPRITES IN ACTION

SCSS

```scss
@import "compass/utilities/sprites";
@import "my-icons/*.png";
@include all-my-icons-sprites;
```

CSS

```css
.my-icons-sprite,
.my-icons-delete,
.my-icons-edit,
.my-icons-new,
.my-icons-save   { background: url('/images/my-icons-s34fe0604ab.png') no-repeat; }

.my-icons-delete { background-position: 0 0; }
.my-icons-edit   { background-position: 0 -32px; }
.my-icons-new    { background-position: 0 -64px; }
.my-icons-save   { background-position: 0 -96px; }
```

q & a

http://sass-lang.com

http://compass-style.org

Resources

http://sass-lang.com
http://compass-style.org