Sass & Compass

THE FUTURE OF STYLESHEETS Now

Brandon Mathis

CSS Today

THE GOOD, THE BAD & THE CRY FOR HELP

CSS 2.1

CSS 3

what they see

- Speed
- Works on my phone
- Woah, 3D animation!
- New Fonts
- Browser competition

what they see

what we see

- Way more complexity, same old tools.
- Selectors, properties and values. That's it?
- No variables?
- No math?

what we see

CSS is a declarative language made of selectors, properties, values and schemes for priority and inheritance, defining how styles apply.

Look, Jane.

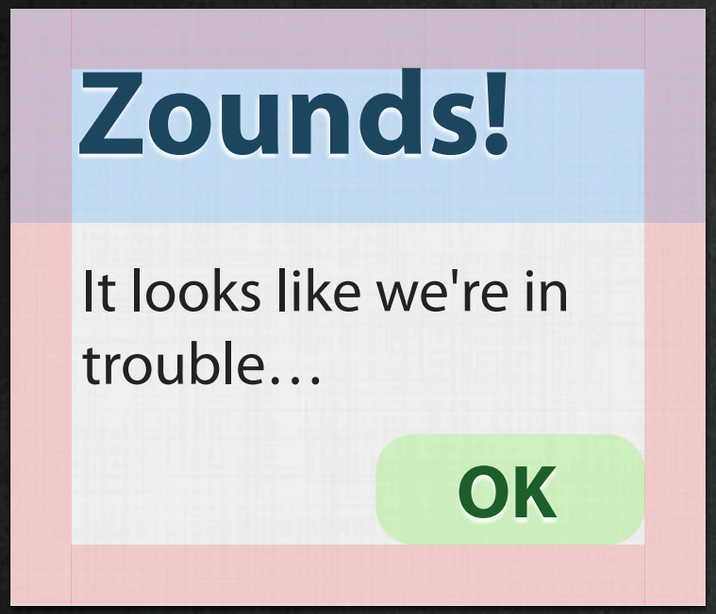
Look, look.

See Dick.

See, see.

Oh, see.

See Dick.




```
.msg { padding: 24px; }

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

Problems

- Repetition causes maintenance challenges
- Relationships are not clear
- Reasons are trapped in the mind of the designer

Dear W3C, We rely on advanced text editors, calculators, color pickers and widgets to make writing CSS bearable.

Throw us a frickin' bone.

Sass.

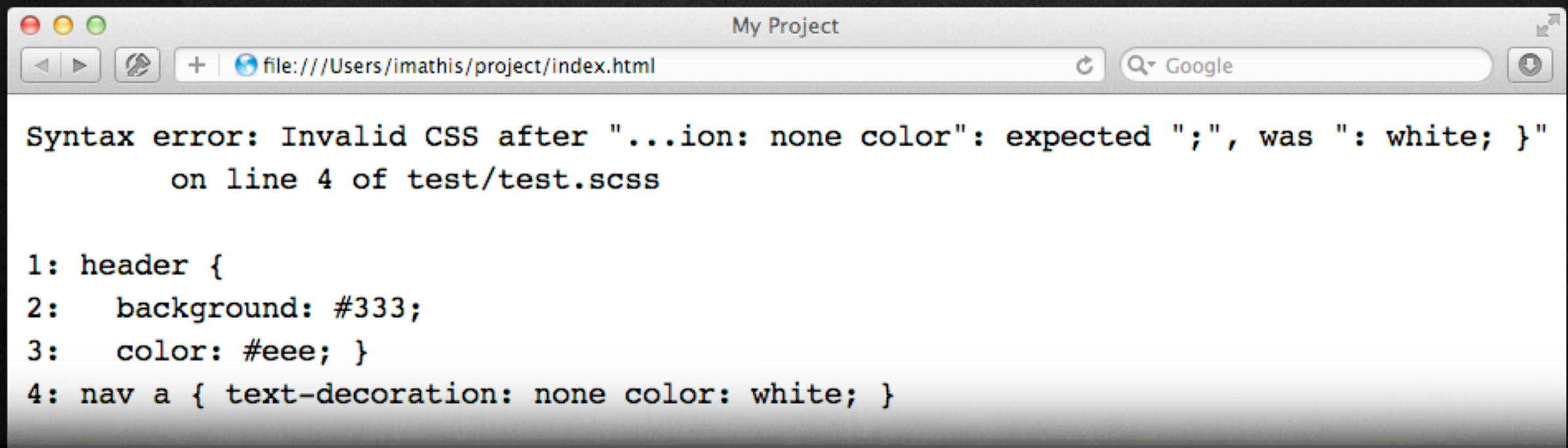
{style with attitude}

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

Setup

Bash

```
gem install sass
sass --watch screen.scss
>>> Sass is watching for changes. Press
Ctrl-C to stop.
>>> Change detected to: /Users/imathis/
project/sass/screen.scss
   overwrite ./screen.css
```



SCSS - Sassy CSS (.scss)

```
article {
  margin-bottom: 2em;
  .entry-content {
    border-top: 1px solid #eee;
  }
}
```

Indented Sass (.sass)

```
article
  margin-bottom: 2em
  .entry-content
    border-top: 1px solid #eee
```

Nesting

SIMPLE & BRILLIANT

Nesting Rules

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; }
```

Nesting Rules

scss

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

css

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

Nest 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 }
```

The Parent Selector

& references a rule's parent selectors.

scss

```
a {
  color: blue;
→ &:hover { color: red }
  display: inline-block;
  line-height: 1.8em;
}
```

css

```
a { color: blue; display: inline-block;
    line-height: 1.8em; }
→ a:hover { color: red }
```

The Parent Selector

& can add context to a selector.

scss

```
article {
  h1 { font-size: 2.4em }
→ .blog-index & {
    h1 { font-size: 2em }
  }
}
```

css

```
article h1 { font-size: 2.4em }
→ .blog-index article h1 { font-size: 2em }
```

The Parent Selector

& loves working with Modernizr.

scss

```scss
button {
  background: linear-gradient(#444,#222);
  .no-cssgradients & { background: #333 }
}
```

css

```css
button {
  background: linear-gradient(#444, #222);
}
.no-cssgradients button {
  background: #333
}
```

@media bubbling

@media rules bubble up with context.

scss

```scss
#content {
  margin: 0 1.5em;
  @media screen and (min-width: 1280px) {
    margin: 0 2.5em
  }
}
```

css

```css
#content { margin: 0 1.5em; }
@media screen and (min-width: 1280px) {
  #content { margin: 0 1.5em; }
}
```

Variables

A LANGUAGE'S BREAD & BUTTER

Using Variables

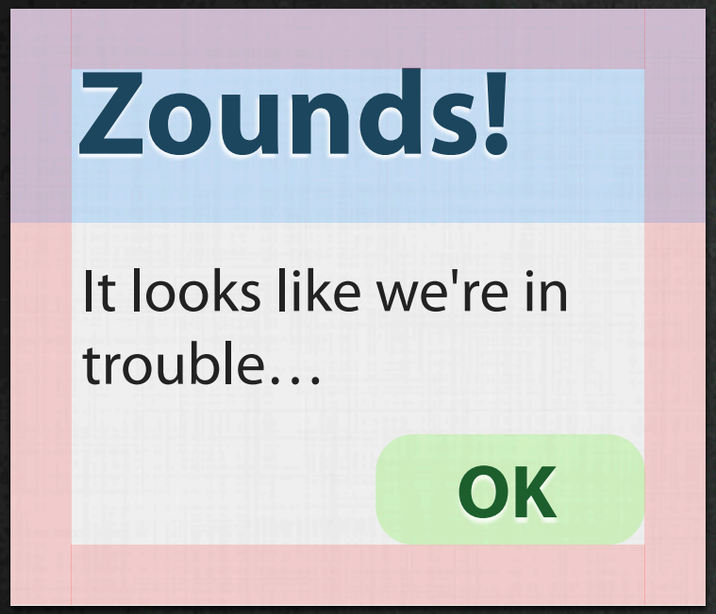
- Colors, numbers, or text.
- Understands units
- $variable: value;

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; }
```




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

With Sass…

- No unnecessary repetition, low maintenance
- The variable makes relationships clear
- The designer's reasoning is evident

@extend

THE BOMB DIGGITY OF STYLESHEET ABSTRACTION

CSS

```css
.button {
  background: blue; color: white;
  padding:0.2em 0.8em;
  border-radius:0.4em;
}

.button-delete { background: red; }
```

HTML

```html
<a class="button button-delete" href="...">
  Delete
</a>
```

Using @extend

scss

```scss
.button {
  background: blue; color: white;
  padding:0.2em 0.8em;
  border-radius:0.4em;
}
```

```scss
.button-delete {
  @extend .button;
  background: red;
}
```

```css
.button, .button-delete {
  background: blue; color: white;
  padding:0.2em 0.8em;
  border-radius:0.4em;
}

.button-delete { background: red; }

.msg .button, .msg .button-delete {
  font-size: 1.8em;
}
```

Mixins

NOW WE'RE COOKIN'

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; }
```

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

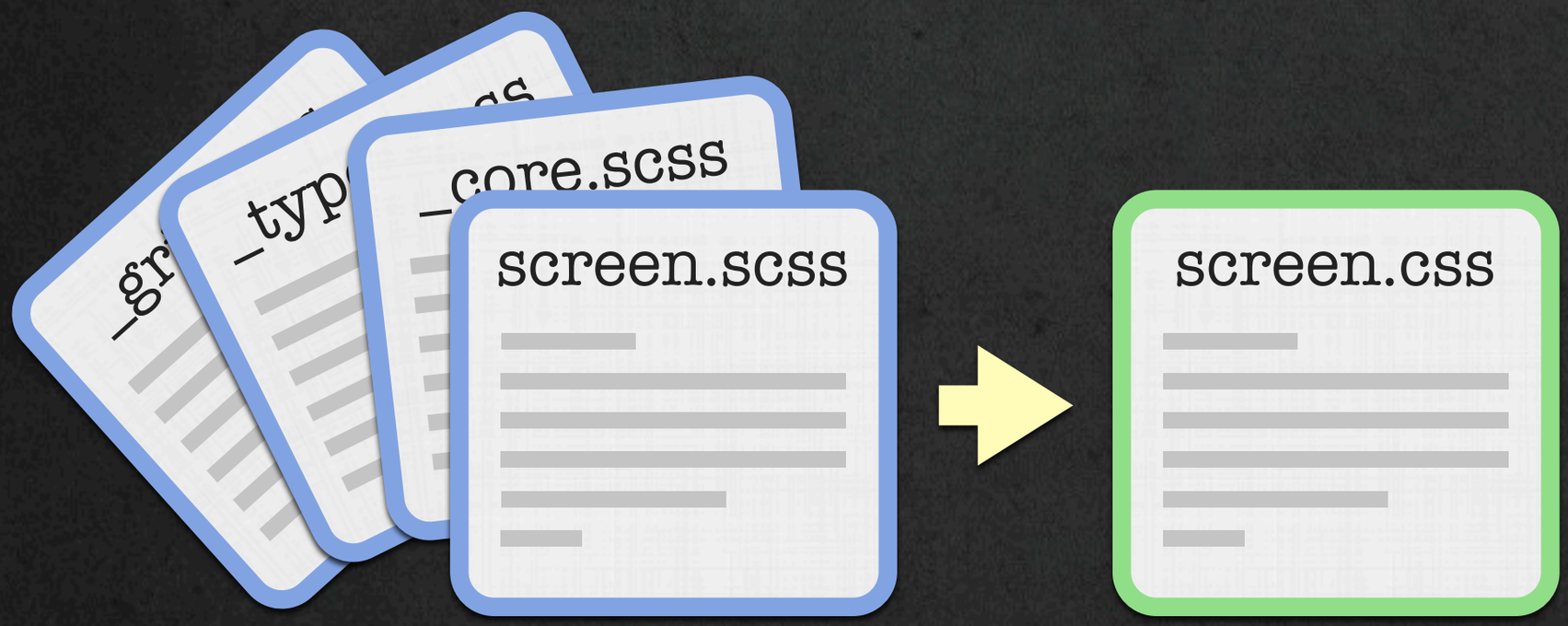
```css
a { color: blue; }
a:hover { green; }
```

Importing
THE RIGHT WAY

Combine Sass with @import.

scss

```scss
/* screen.scss */
@import "core";
@import "typography", "grid";
```



You can also nest an @import.

scss

```scss
@media screen and (min-width: 320px) {
  @import 'phone';
}

#account {
  @import 'pages/account';
}
```

Math

(2 * 5) - 8 = FINALLY

Math Operators

Math operators (+, -, *, /, %) work with numbers

scss

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

Division a special case

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
```

scss

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

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

css

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

Number Functions

scss

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

Loops & Conditions

IS IT GETTING HOT IN HERE?

Conditionals

- Logic Operators < > <= >= == !=
- @if, @else, @else if
- and, or

Relational operators (<, >, <=, >=) evaluate numbers

scss

```scss
1  <  20  // true
10 <= 20  // true
4  >  1   // true
4  >= 1   // true
```

Comparison operators (==, !=) evaluate all data types

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)
```

Conditional theming

scss

```scss
$theme: ocean;

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

The if function

scss

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

The @for loop

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; }
```

The @while loop

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; }
```

The @each loop

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'); }
```

Color

SORRY COLOR PICKER,
I'VE MET SOMEONE...

The RGBA function

scss

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

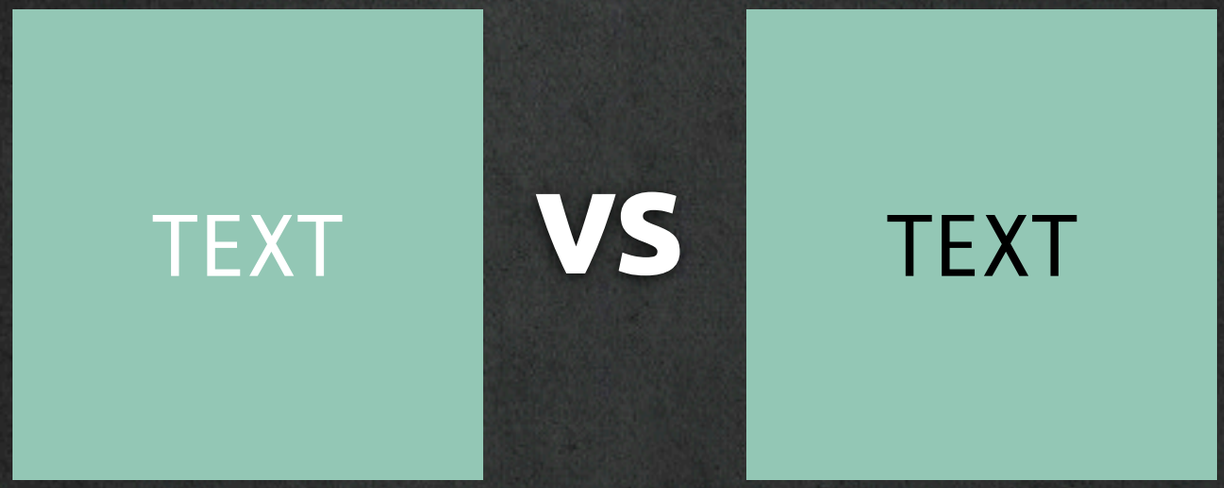
css

```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
```

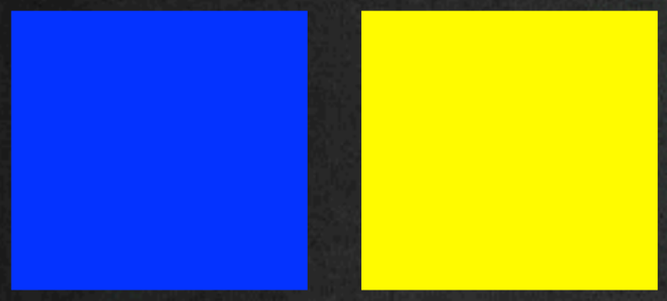


scss

```scss
$darkbg: lightness($bg) < lightness(gray);

button {
  color: if($darkbg, #fff, #000);
}
```

Manipulating Colors

```
invert(blue)
```

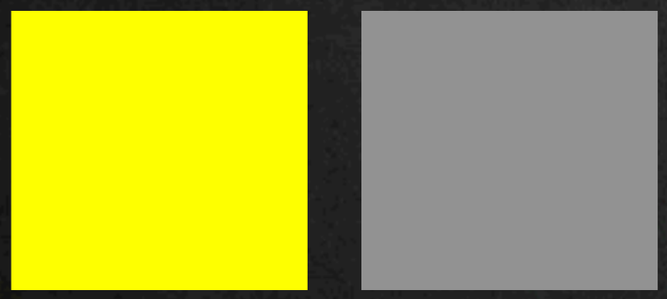


```
complement(#6cf620)
```

```
mix(red, yellow)
```

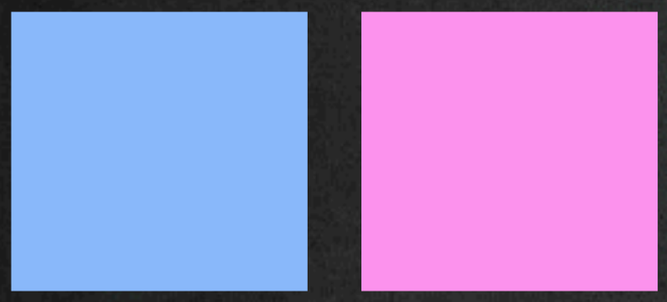
```
mix(red, yellow, 30%)
```

```
grayscale(yellow)
```



HSLA Manipulations

adjust-hue(#77a7f9,90)

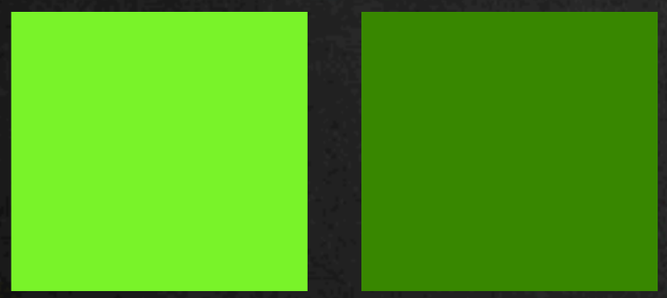


adjust-hue(#77a7f9,-90)

saturate(#9b8a60,50%)

desaturate(#d9a621,50%)

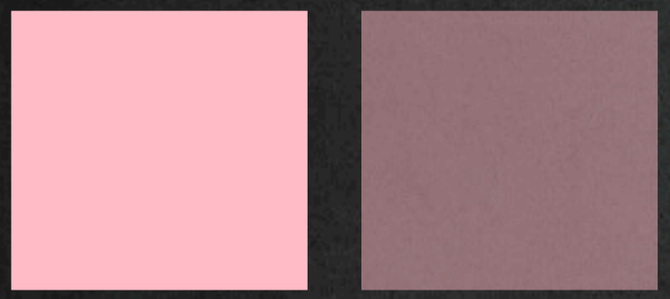
darken(#6cf620,30%)



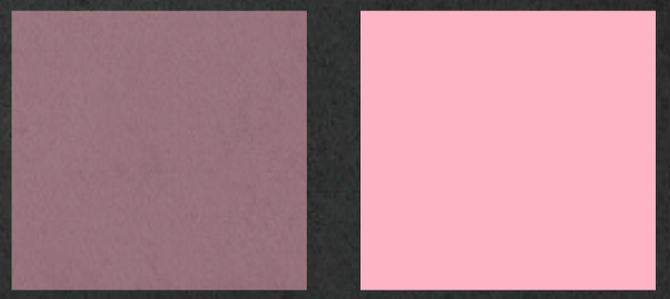
lighten(#2e7805,50%)

HSLA Manipulations

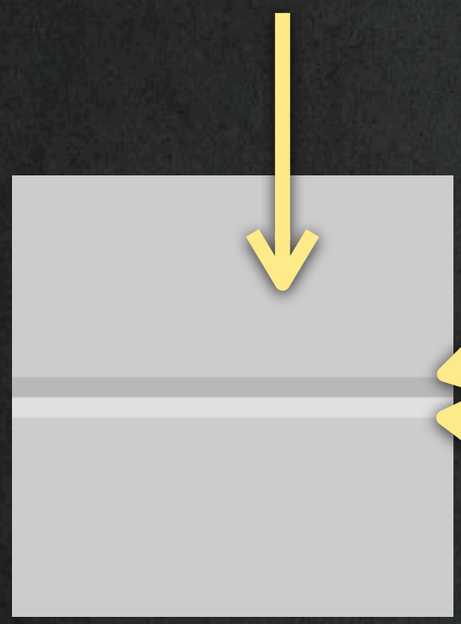
```
fade-out(#fab, .5)
```



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



```
$bg: #ccc;
```



```
darken($bg, 8%);
```

```
lighten($bg, 8%);
```

change-color

set any property of a color

scss

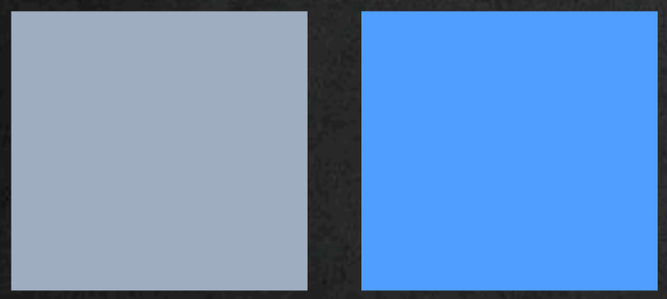
```
change-color($color, [$red], [$green], [$blue],
[$hue], [$saturation], [$lightness], [$alpha]);
```

change-color

```
change-color(#ba5637, $hue:60);
```

#19f65d

```
change-color(#8e9cb3, $saturation:100);
```



#4288ff

```
change-color(#6cf620, $green: 60, $blue: 100);
```

#6C3C64

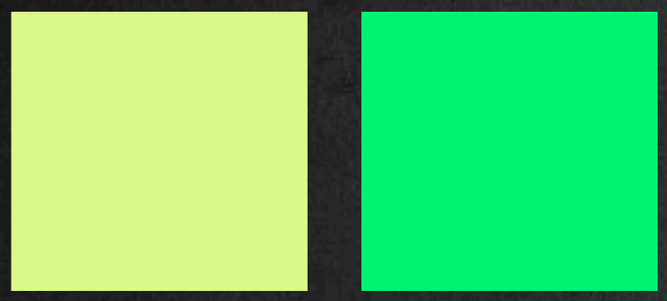
adjust-color

Incrementally manipulate any property of a color

scss

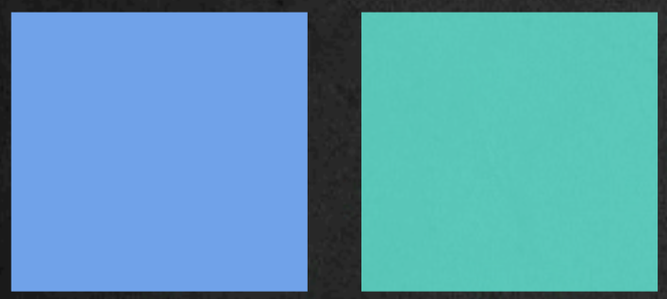
```scss
adjust-color($color, [$red], [$green], [$blue],
[$hue], [$saturation], [$lightness], [$alpha]);
```

```
adjust-color(#d3fa7b, $hue:60, $lightness: -20%);
```



#19f65d

```
adjust-color(#5f8fe3, $green:100, $alpha: -0.25);
```



rgba(95, 255, 227, 0.75);

scale-color

fluidly scale any percentage based property of a color

scss

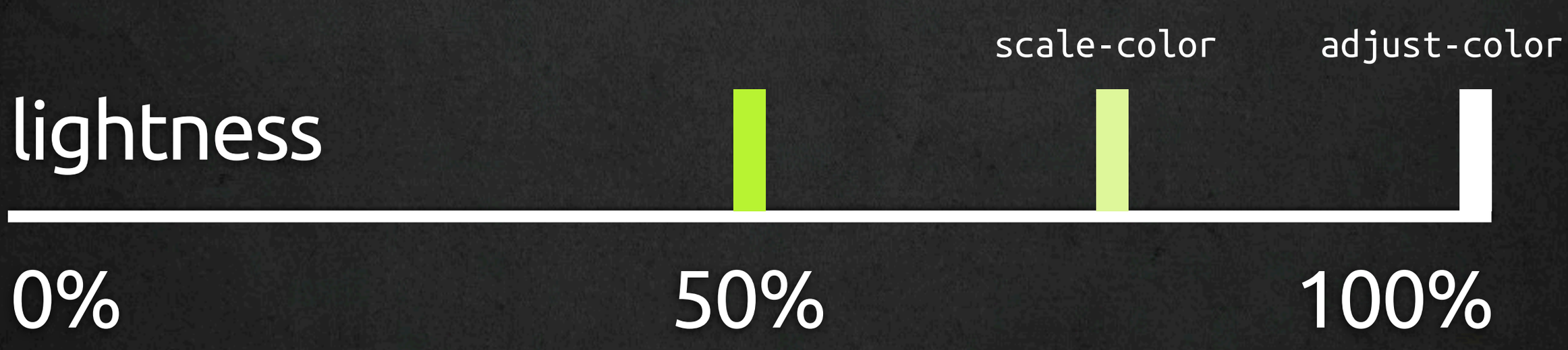
```
scale-color($color, [$red], [$green], [$blue],
[$saturation], [$lightness], [$alpha]);
```

```
scale-color(#adf609, $lightness:50%);
```

#d6fa84

```
adjust-color(#adf609, $lightness:50%);
```

white



Custom Functions

PUT A CHERRY ON TOP

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; }
```

scss

```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); }
```

compass

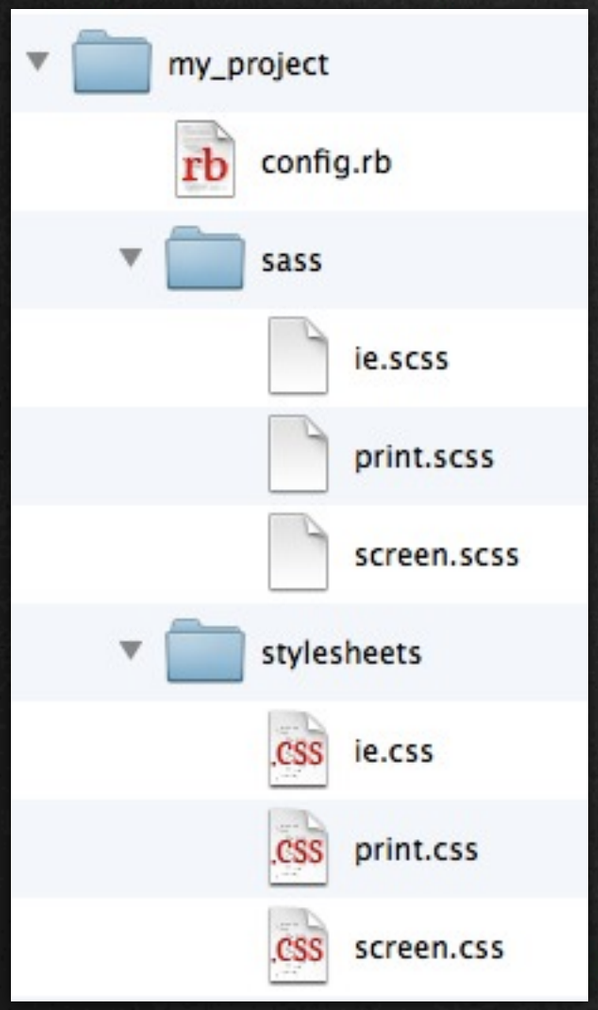
What we get?

- Mixin library
- Sass functions
- Environmental awareness
- Extensions

Setup

Bash

```
gem install compass
compass create my_project
```



Configuration

Ruby

```
http_path = "/"
css_dir = "stylesheets"
sass_dir = "sass"
images_dir = "images"
fonts_dir  = "fonts"
javascripts_dir = "javascripts"
output_style = :compressed
```

Helper functions

scss

```
adjust-lightness, scale-lightness
adjust-saturation, scale-saturation
image-url
image-height
image-width
inline-image
font-url
inline-font-files
pi
sin
cos
tan
more...
```

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
Browser Hacks, Clearfixes, Resets

Element styles
Links, Lists, Float, Tables, Text

Design patterns
Tag Cloud, Sticky footer, Vertical rhythm

CSS3
appearance, background, gradients, background-clip
background-origin, background-size, border-radius
box, box-shadow,box-sizing, CSS3 PIE, columns,
font-face, opacity, transform, transition, more...

Example with 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; }
```

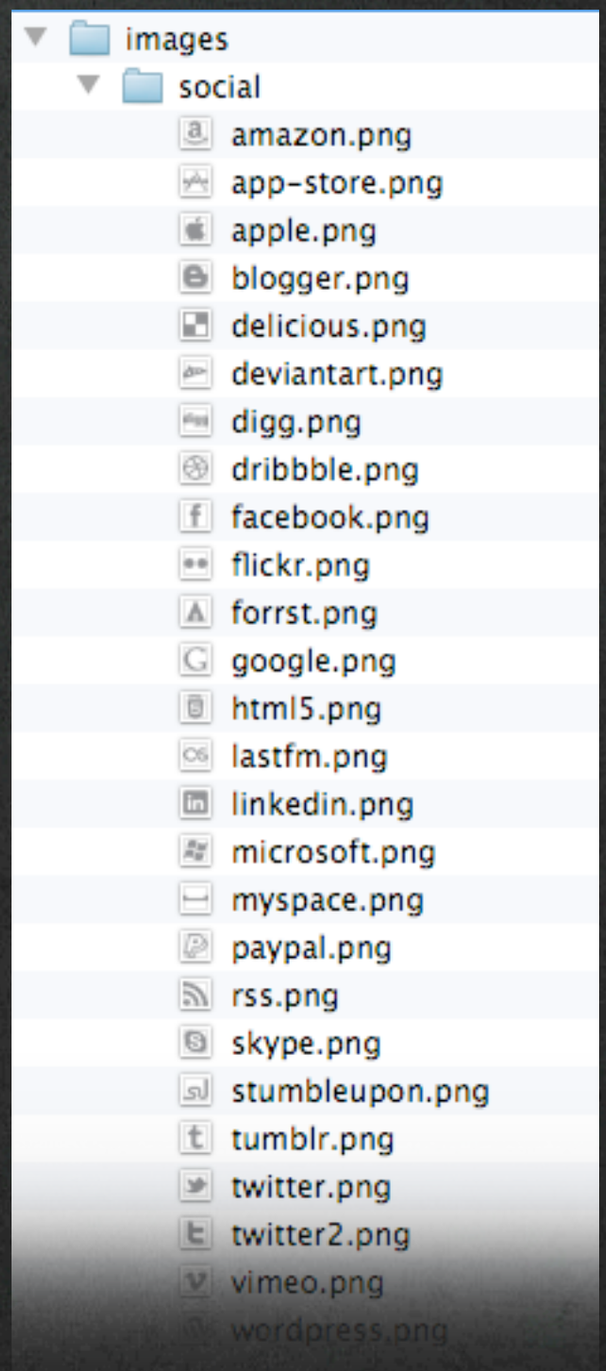
Using Sprites

Pros:

- Speed
- Server load

Cons:

- Creating sprite maps
- Writing the CSS
- Maintenance



Compass magic sprites

- Generates the sprite map
- Generates the CSS
- Easy to configure classes, spacing, etc
- Add a new image? updates automatically

```
@import "social/*.png";
@include all-social-sprites($dimensions:true);
```

```
.social-sprite, .social-amazon, .social-app-store, .social-apple, .social-blogger, .social-
delicious, .social-deviantart, .social-digg, .social-dribbble, .social-facebook, .social-flickr, .social-
forrst, .social-google, .social-html5, .social-lastfm, .
myspace, .social-paypal, .social-rss, .social-skype, .so
twitter, .social-twitter2, .social-vimeo, .social-wordpr
url('../images/social-sfdebe26bed.png') no-repeat; }
.social-amazon { background-position: 0 0; height: 32px;
.social-app-store { background-position: 0 -32px; height
.social-apple { background-position: 0 -64px; height: 32
.social-blogger { background-position: 0 -96px; height:
.social-delicious { background-position: 0 -128px; heigh
.social-deviantart { background-position: 0 -160px; heig
.social-digg { background-position: 0 -192px; height: 32
.social-dribbble { background-position: 0 -224px; height
.social-facebook { background-position: 0 -256px; height
.social-flickr { background-position: 0 -288px; height:
.social-forrst { background-position: 0 -320px; height:
.social-google { background-position: 0 -352px; height:
.social-html5 { background-position: 0 -384px; height: 3
.social-lastfm { background-position: 0 -416px; height:
.social-linkedin { background-position: 0 -448px; height
.social-microsoft { background-position: 0 -480px; heigh
.social-myspace { background-position: 0 -512px; height:
.social-paypal { background-position: 0 -544px; height:
.social-rss { background-position: 0 -576px; height: 32p
.social-skype { background-position: 0 -608px; height: 3
.social-stumbleupon { background-position: 0 -640px; hei
.social-tumblr { background-position: 0 -672px; height:
.social-twitter { background-position: 0 -704px; height:
.social-twitter2 { background-position: 0 -736px; height
.social-vimeo { background-position: 0 -768px; height: 3
.social-wordpress { background-position: 0 -800px; height: 32px; width: 32px; }
.social-yahoo { background-position: 0 -832px; height: 32px; width: 32px; }
.social-youtube { background-position: 0 -864px; height: 32px; width: 32px; }
```

Extensions

- Easy to write and share
- Sass, Images, HTML, JS, Fonts, etc
- Zip or Ruby gems

Plugins & Extensions

- Fancy Buttons, Sassy Buttons - easy CSS3 buttons
- Animate - CSS3 animation library
- RGBApng - Generate alpha pngs from Sass
- Compass Magick - Render CSS3 Gradients to png
- Many more...

UNITED MITOCHONDRIAL DISEASE FOUNDATION®

HOPE. ENERGY. LIFE.

umdf.org/compass

Color Hacker

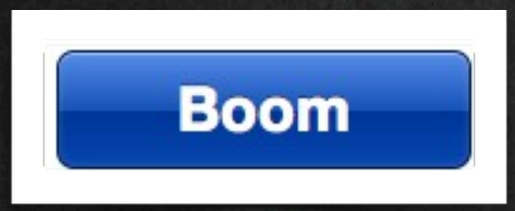
CRACK THE COLOR SHCEME

Reverse engineering color

scss

```scss
@debug(hack-colors(#d3643b #edebe6 #d6e1c7));


$key: #d3643b;
$color-2: scale-color(
  adjust-hue($key, 26.673deg),
  $saturation: -74.296%, $lightness: 82%);
$color-3: scale-color(
  adjust-hue($key, 69.2deg),
  $saturation: -52%, $lightness: 64.167%);
```

Pick gradients



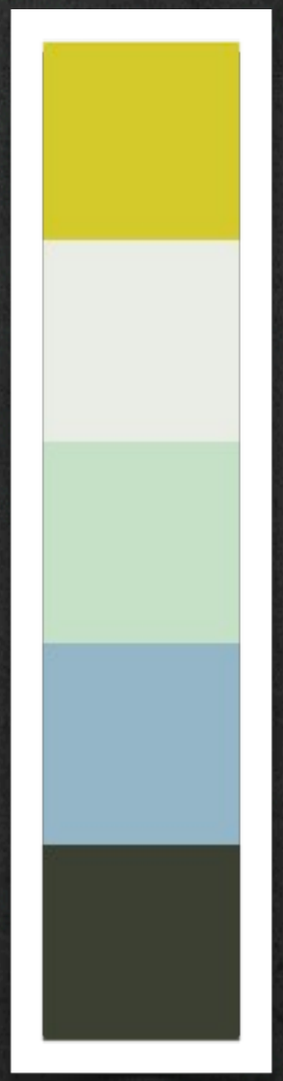

```
button { background:linear-gradient(#588EE7
0%, #1A55B5 50%, #003895 50%, #0D4AAD 100%);
border: 1px solid #001F53;
}
```

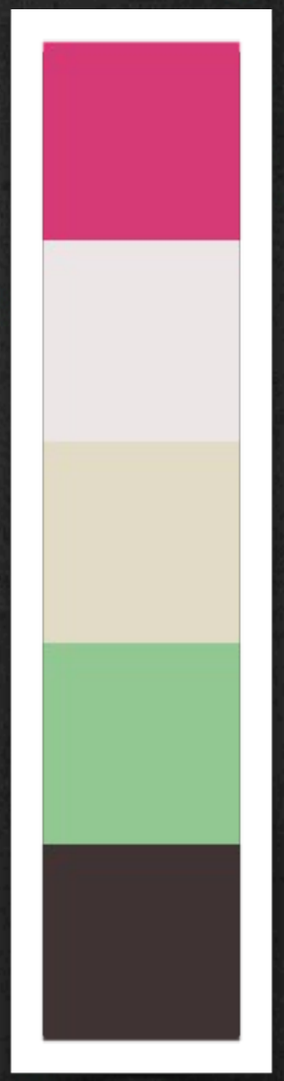
```
$button: #003895 #588EE7 #1A55B5 #0D4AAD #001F53;
@debug(color-hack($button));

button { @include background(linear-gradient(
  $color-2 0%, $color-3 50%, $key 50%, $color-4 100%));
  border: 1px solid $color-5;
}
```

Tweak color schemes

```
$scheme: #d3643b #edebe6 #d6e1c7 #94c7b6 #403b33;
```

```
adjust-hue($key, 32deg);
adjust-hue($key,-32deg);
```





j.mp/color-hacker

`gem install color-hacker`

Thank You

YOU'RE ALL LOVELY PEOPLE

Brandon Mathis

twitter + github @imathis

Talk links:

j.mp/imathis-fowd-links