Sass.

The Future of Stylesheets

Without the Wait

Chris Eppstein

Blog - http://chriseppstein.github.com
Follow me - @chriseppstein, +chriseppstein
Architect - Caring.com
Creator - Compass Stylesheet Framework
Core Contributor - Sass

Sass.

compass

Caring.com

You're there for them. We're here for you.

A website for caregivers of the sick and elderly.

The right information at just the right time

Books:

Sass & Compass In Action

http://www.manning.com/netherland/
DISCOUNT CODE: `sass40 (save 40%)`

Pragmatic Guide to Sass

http://pragprog.com/book/pg_sass/pragmatic-guide-to-sass
DISCOUNT CODE: `SassCompass2011 (save 25%)`

Today

Rationale

Why does Sass exist?
What problems does it solve?

Syntax

What does Sass provide?

Best Practices

Tips, tricks & pitfalls

Questions?

Please ask them.

What is “Sass”?

Syntactically Awesome StyleSheets

What is CSS?

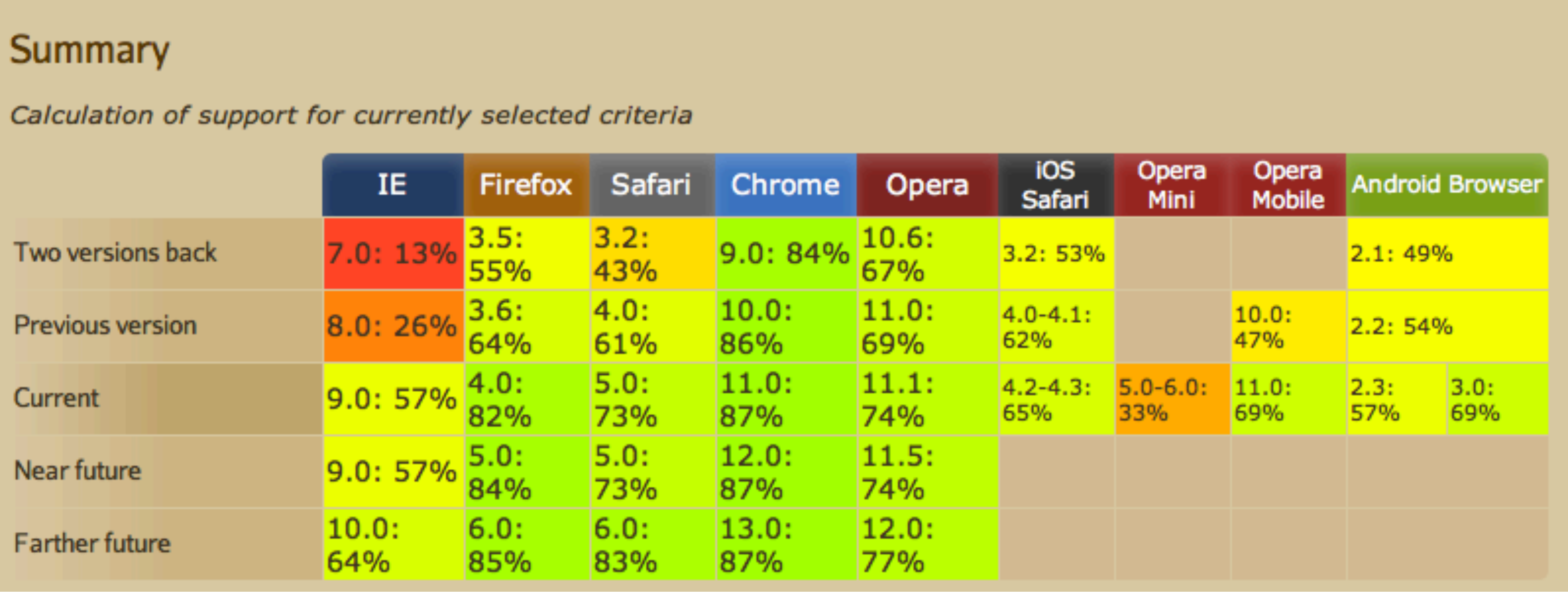
- Originated in Academia
- 14 Years Old
- Three Primitives:
  - Selectors
  - Properties
  - Values

CSS is Pretty.

Pretty dumb.

Never in the history of software have developers been asked to do so much with so little.

Browsers & Versions



Summary

Calculation of support for currently selected criteria

	IE	Firefox	Safari	Chrome	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
Two versions back	7.0: 13%	3.5: 55%	3.2: 43%	9.0: 84%	10.6: 67%	3.2: 53%			2.1: 49%
Previous version	8.0: 26%	3.6: 64%	4.0: 61%	10.0: 86%	11.0: 69%	4.0-4.1: 62%		10.0: 47%	2.2: 54%
Current	9.0: 57%	4.0: 82%	5.0: 73%	11.0: 87%	11.1: 74%	4.2-4.3: 65%	5.0-6.0: 33%	11.0: 69%	2.3: 57% 3.0: 69%
Near future	9.0: 57%	5.0: 84%	5.0: 73%	12.0: 87%	11.5: 74%				
Farther future	10.0: 64%	6.0: 85%	6.0: 83%	13.0: 87%	12.0: 77%				

Multiple Form Factors

Desktop, Mobile, Print...

Dozens of Templates

Experiment & Test

Maintain Consistency

Performant

Due Yesterday

New Hotness: CSS3

33 New Selectors
120+ New Properties
3 New @-rules
Enhanced Media Queries

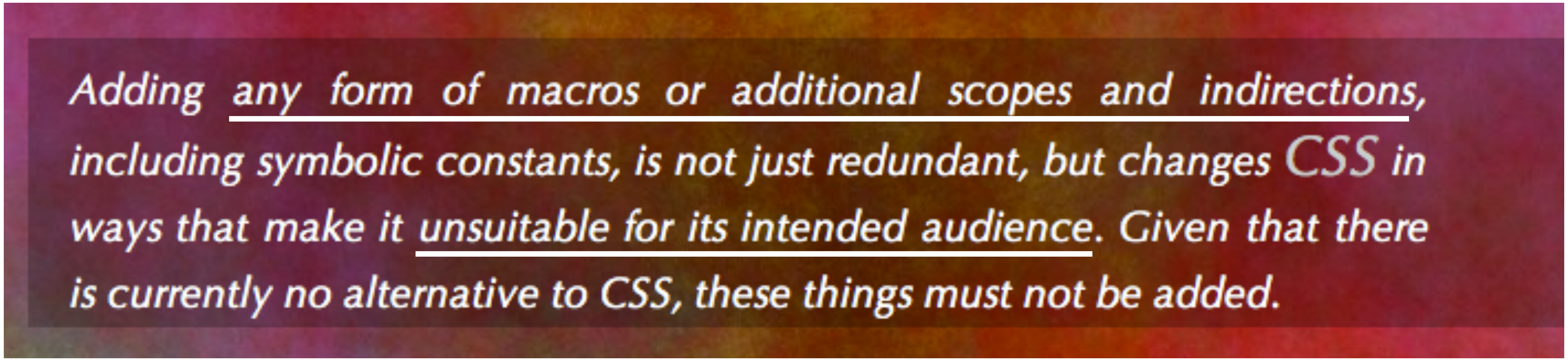
CSS3

No new syntax*

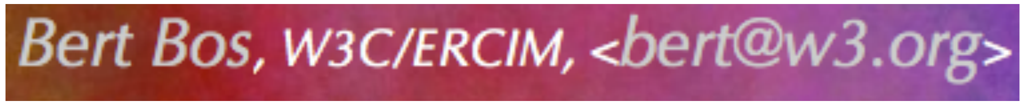
* Yet.

Sass.

compass



Adding any form of macros or additional scopes and indirections, including symbolic constants, is not just redundant, but changes CSS in ways that make it unsuitable for its intended audience. Given that there is currently no alternative to CSS, these things must not be added.



Bert Bos, W3C/ERCIM, <bert@w3.org>

CSS was designed for the Mythical Designer who is too stupid to understand High School Algebra.

If you want to make something **simple** for **beginners**...

...make it **powerful** enough for the experts to create that **simplicity**.

But...

There is an alternative.

What is Sass?

- Born from Real World Experience
- Brings Software Development Methodologies to Stylesheets:
  - Reusable, distributable libraries
  - DRY
  - Define abstractions
  - Adhere to architectural boundaries

What is Sass?

New Primitives:

- Variables
- Mixins
- Selector Inheritance
- Calculations
- Functions
- Conditionals
- Loops

What is Sass?

It’s what CSS would have been if we had known then what we know now:

Most stylesheets are written by developers.

Problem:

CSS Stylesheets

Are hard to maintain

<u>D</u>on’t
<u>R</u>epeat
<u>Y</u>ourself

(I don’t have to tell you this -- It’s not like you want to)

A **single** change.
A **single** edit.
A **single** compile.

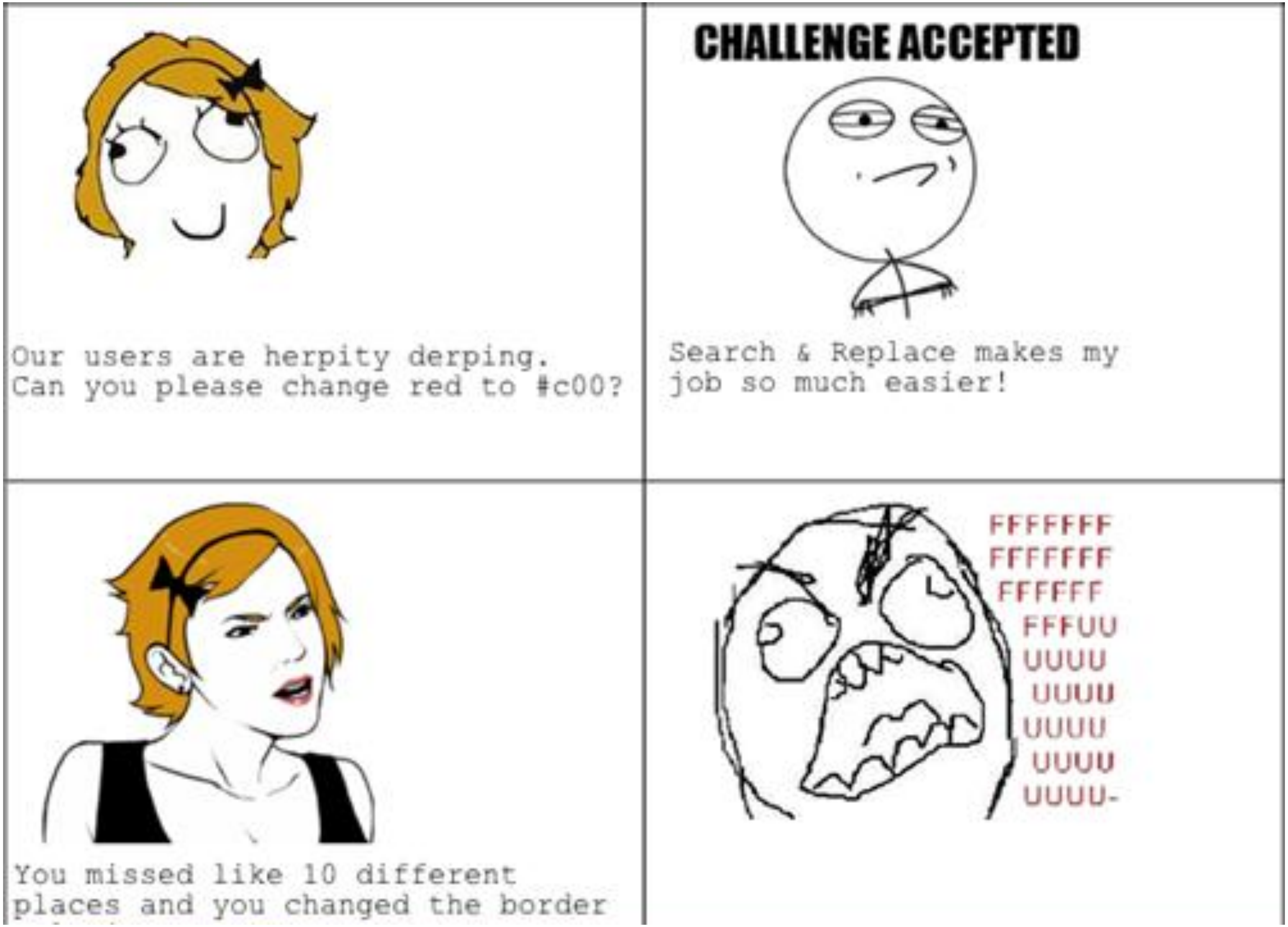
We have **machines**.
Let them do the **busy work**.

Problem:
Same Color
Many Representations

Problem:
Same Color
Different Reasons

Sass.

compass


Our users are herpity derping.
Can you please change red to #c00?
CHALLENGE ACCEPTED
Search & Replace makes my job so much easier!
You missed like 10 different places and you changed the border
FFFFFFF
FFFFFFF
FFFFFF
FFFUU
UUUU
UUUU
UUUU
UUUU
UUUU-

rgba(#f00, 1.0)

hsla(0deg, 100%, 100%, 1)

hsl(0deg, 100%, 100%)

#f00

red

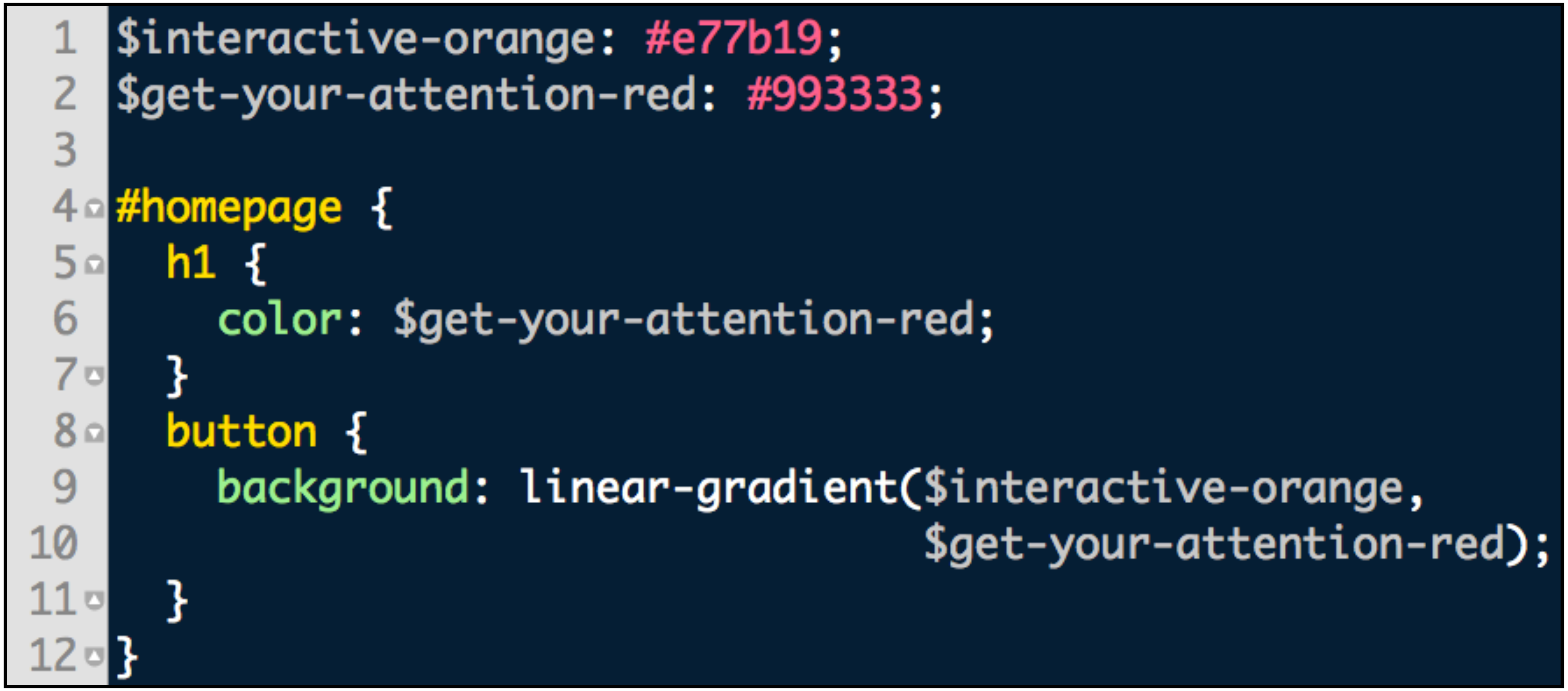
rgb(255, 0, 0)

RED

rgba(255, 0, 0, 1.0)

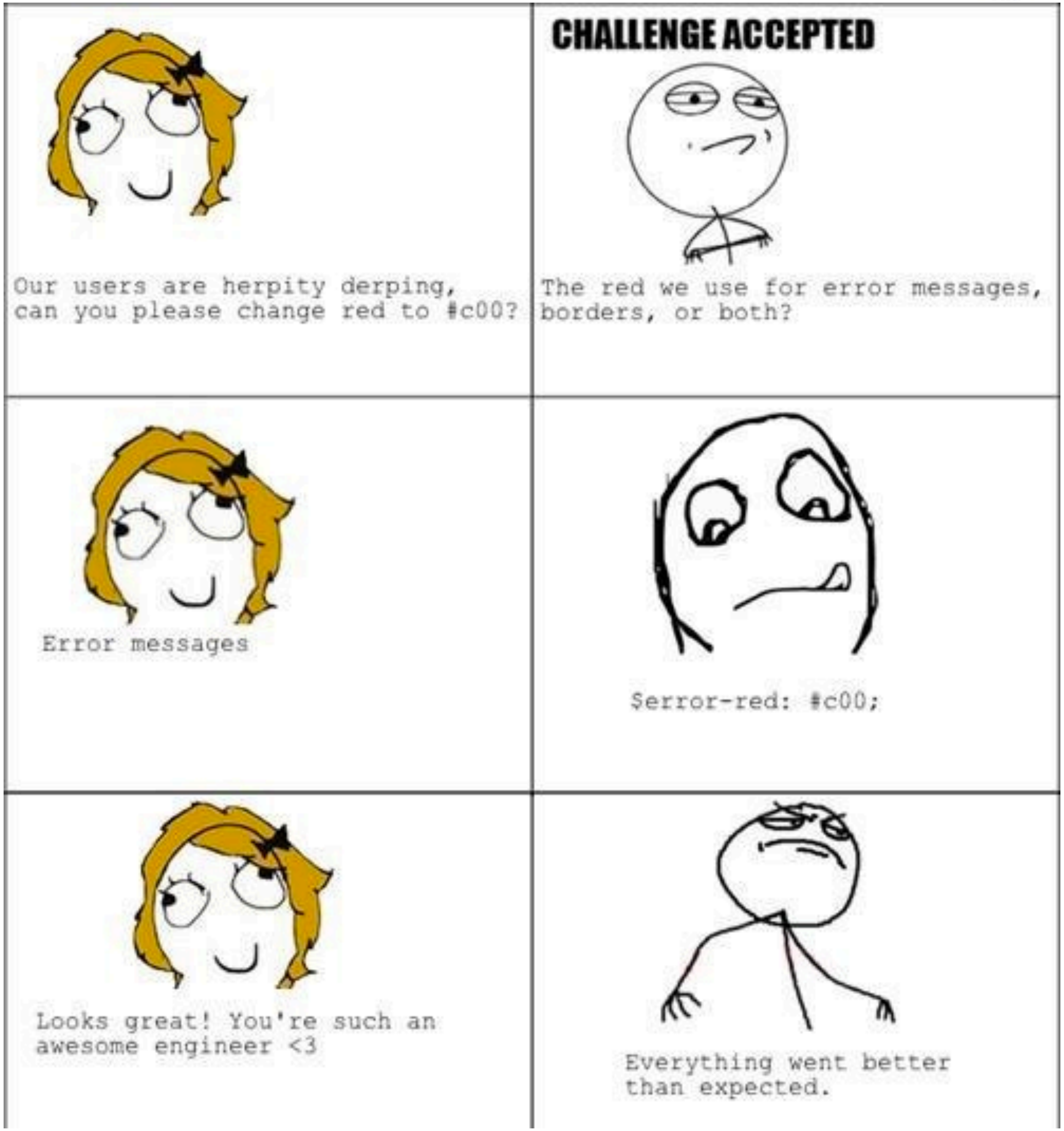
Solution:
Variables

The most requested CSS enhancement



```
$interactive-orange: #e77b19;
$get-your-attention-red: #993333;

#homepage {
  h1 {
    color: $get-your-attention-red;
  }
  button {
    background: linear-gradient($interactive-orange,
                                $get-your-attention-red);
  }
}
```

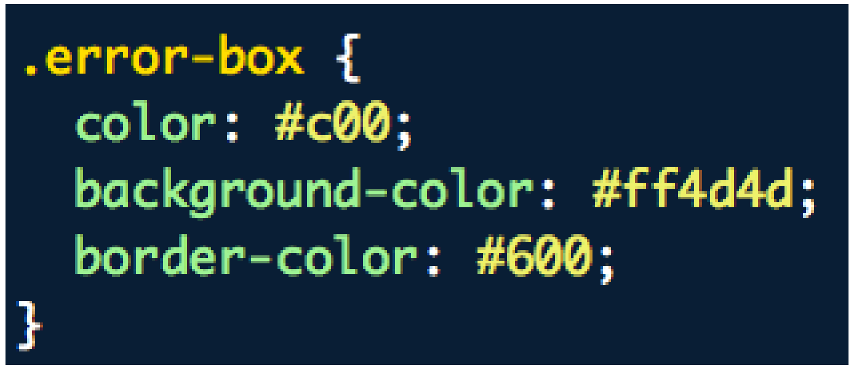
Use variables anywhere a value is allowed.


Our users are herpity derping, can you please change red to #c00?
CHALLENGE ACCEPTED
The red we use for error messages, borders, or both?
Error messages
$error-red: #c00;
Looks great! You're such an awesome engineer <3
Everything went better than expected.

Problem:

Magic Values

Lost knowledge

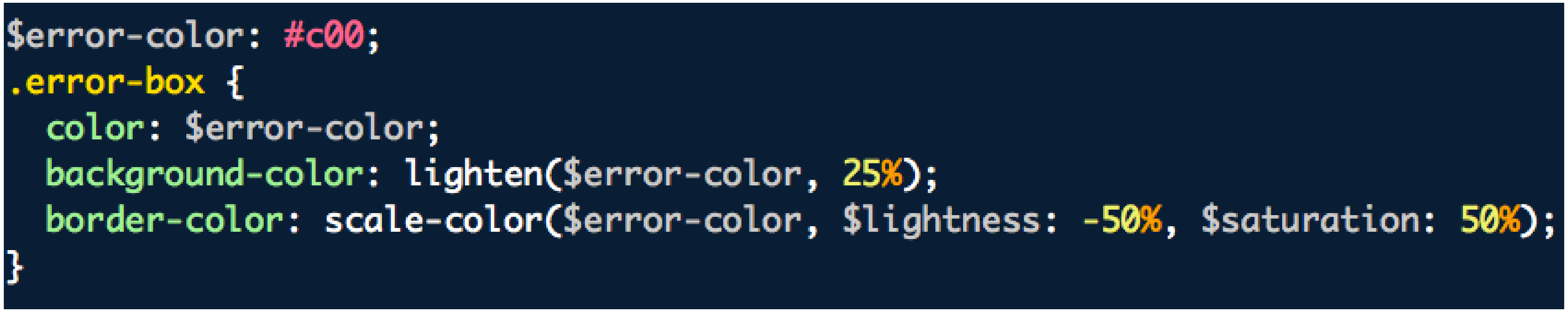


```
.error-box {
  color: #c00;
  background-color: #ff4d4d;
  border-color: #600;
}
```

Deceptively Complex

Solution:
Transformations

Derive values during compilation



```scss
$error-color: #c00;
.error-box {
  color: $error-color;
  background-color: lighten($error-color, 25%);
  border-color: scale-color($error-color, $lightness: -50%, $saturation: 50%);
}
```

As complex as it actually is.

Sass Color Functions

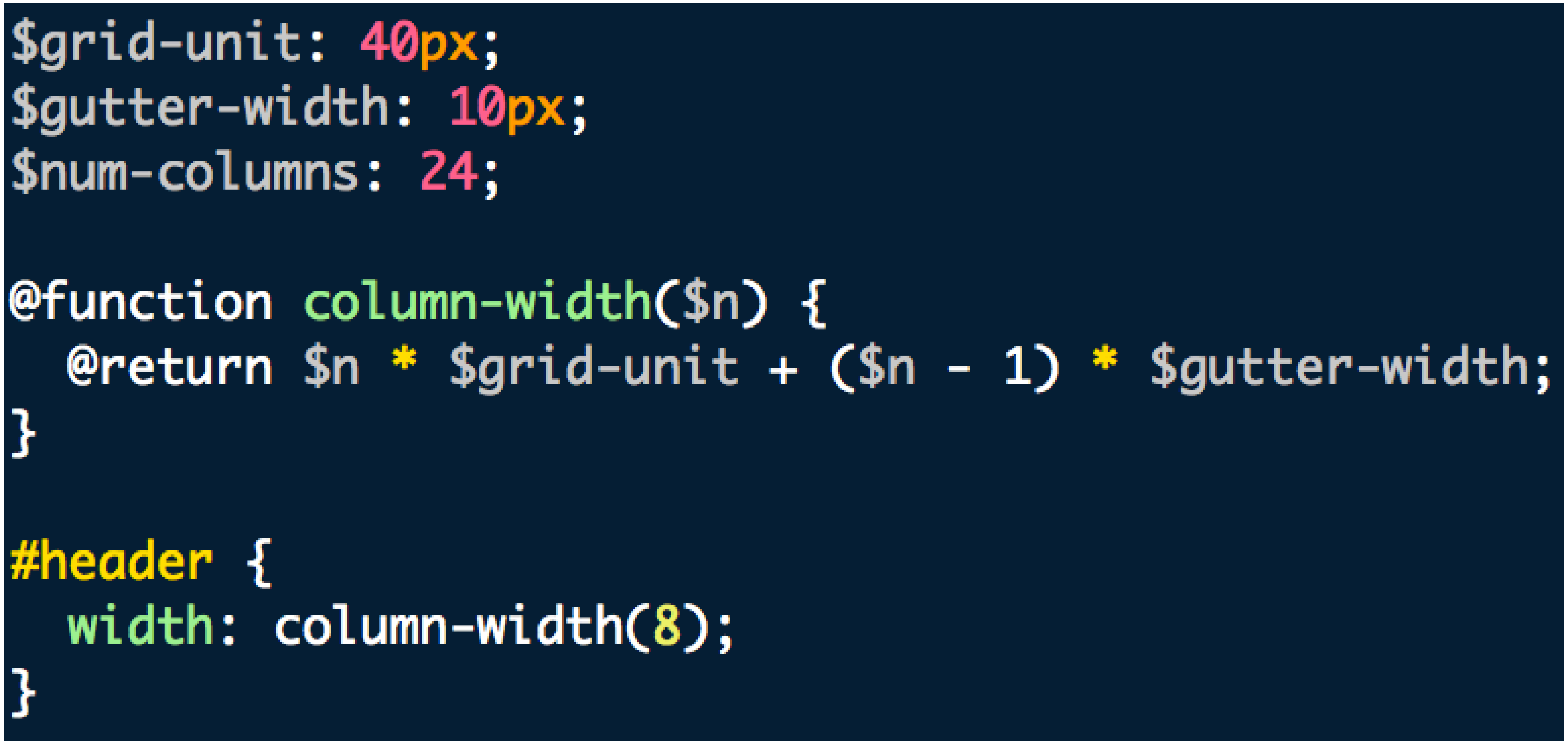
```
mix(red, blue, 40%)             => #660099
lighten(red, 10%)               => #ff3333
darken(red, 10%)                => #cc0000
desaturate(red, 10%)            => #f20d0d
saturate(#511, 10%)             => #5a0c0c
transparentize(red, 0.25)       => rgba(255, 0, 0, 0.75)
opacify(red, 0.25)              => rgba(255, 0, 0, 0.75)
grayscale(red)                  => #808080
complement(red)                 => #00ffff
```

There's more on the Sass website.

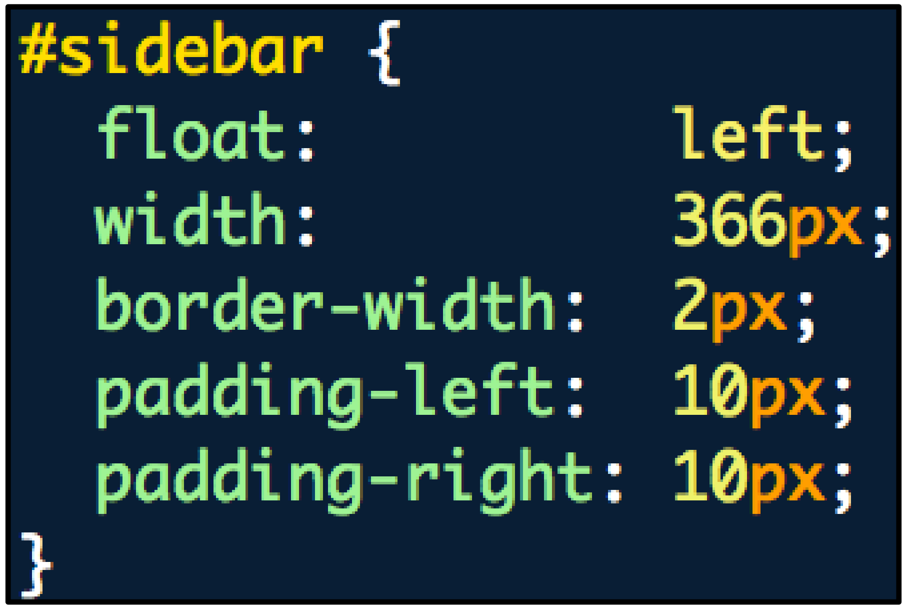
Solution:

Make Reusable Calculations

We call these things “functions”



```
$grid-unit: 40px;
$gutter-width: 10px;
$num-columns: 24;

@function column-width($n) {
  @return $n * $grid-unit + ($n - 1) * $gutter-width;
}

#header {
  width: column-width(8);
}
```

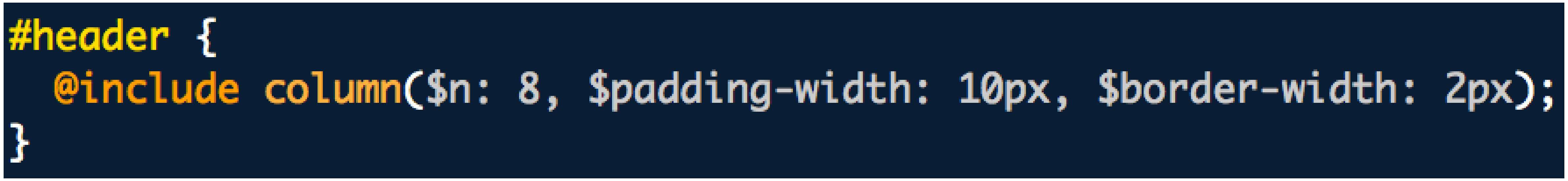


```
#sidebar {
  float:         left;
  width:         366px;
  border-width:  2px;
  padding-left:  10px;
  padding-right: 10px;
}
```

Deceptively Complex

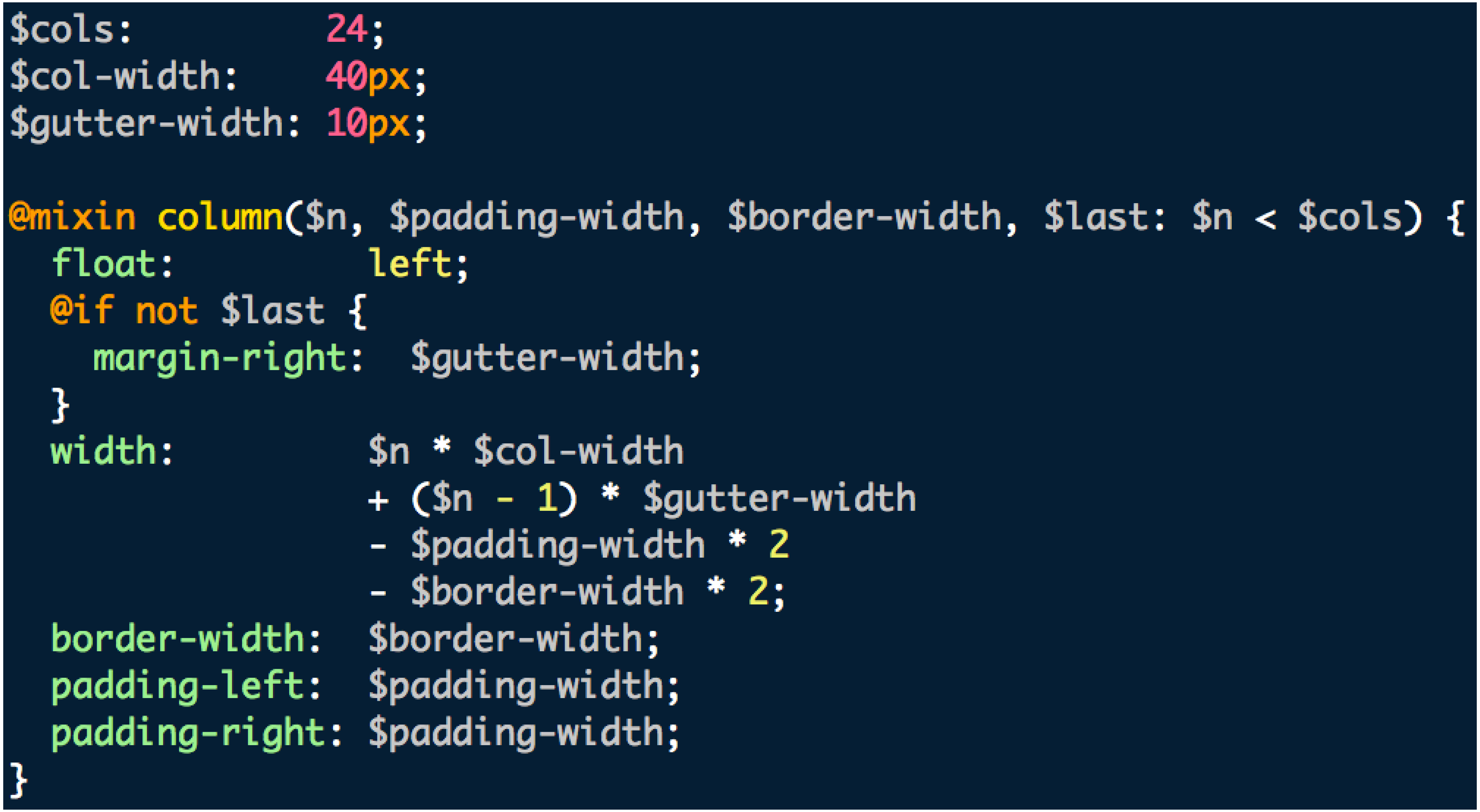
Solution:
Abstractions & Calculations

Derive values during compilation



```
#header {
  @include column($n: 8, $padding-width: 10px, $border-width: 2px);
}
```

Create a simple abstraction



```
$cols:         24;
$col-width:    40px;
$gutter-width: 10px;

@mixin column($n, $padding-width, $border-width, $last: $n < $cols) {
  float:         left;
  @if not $last {
    margin-right:  $gutter-width;
  }
  width:         $n * $col-width
                 + ($n - 1) * $gutter-width
                 - $padding-width * 2
                 - $border-width * 2;
  border-width:  $border-width;
  padding-left:  $padding-width;
  padding-right: $padding-width;
}
```

As complex as it actually is.

Best Practice:

Minimize HTTP Round Trips

Problem:
Long Stylesheets

Hard to find your styles.

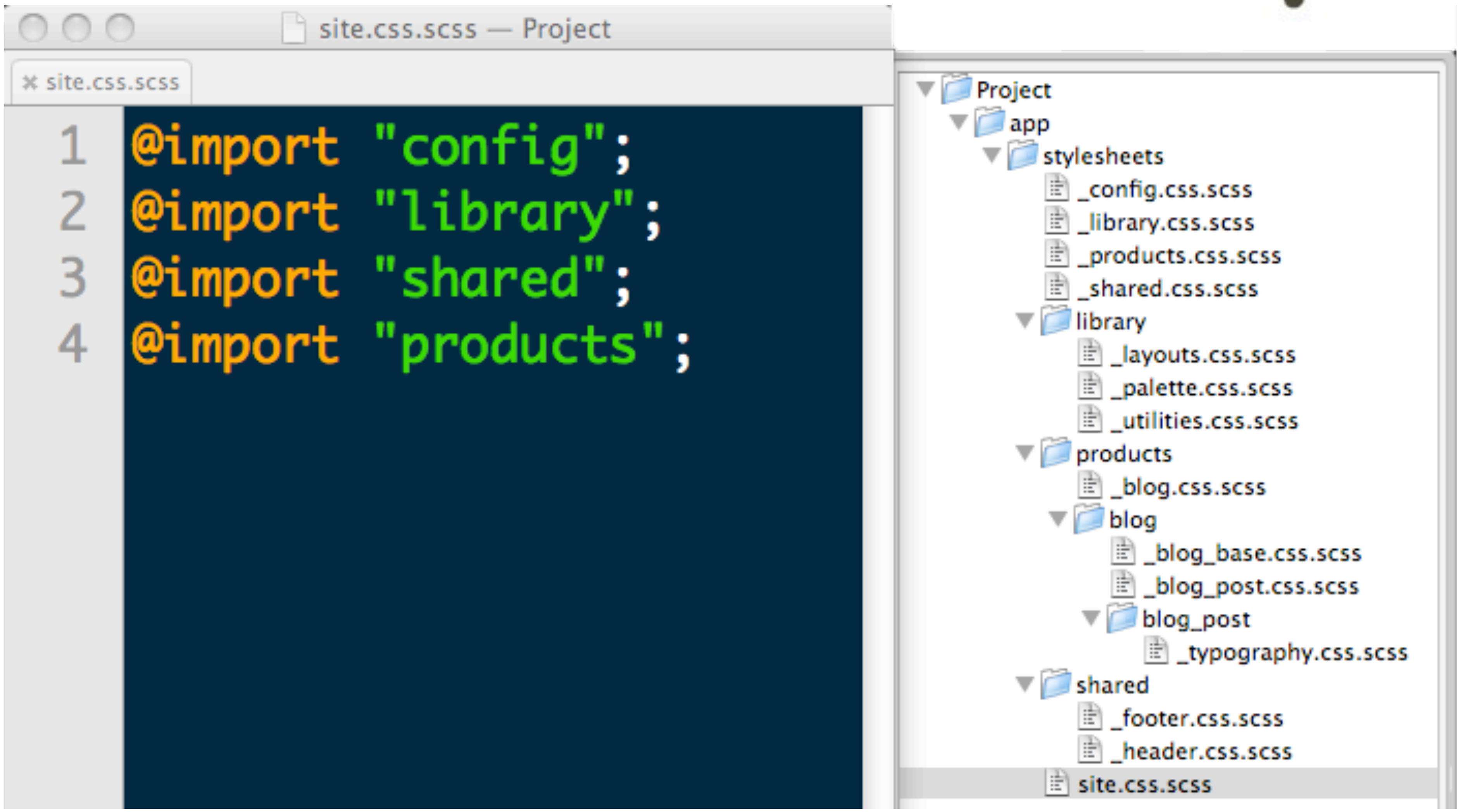
And I mean long...

Seriously, this keeps going...

Solution: Concatenation & Partials

Sass.

compass



site.css.scss — Project

× site.css.scss

```scss
@import "config";
@import "library";
@import "shared";
@import "products";
```

```
▼ Project
  ▼ app
    ▼ stylesheets
        _config.css.scss
        _library.css.scss
        _products.css.scss
        _shared.css.scss
      ▼ library
          _layouts.css.scss
          _palette.css.scss
          _utilities.css.scss
      ▼ products
          _blog.css.scss
        ▼ blog
            _blog_base.css.scss
            _blog_post.css.scss
          ▼ blog_post
              _typography.css.scss
      ▼ shared
          _footer.css.scss
          _header.css.scss
        site.css.scss
```

Partials: Just like with html templates, underscores denote a partial and can only be imported by other stylesheets.

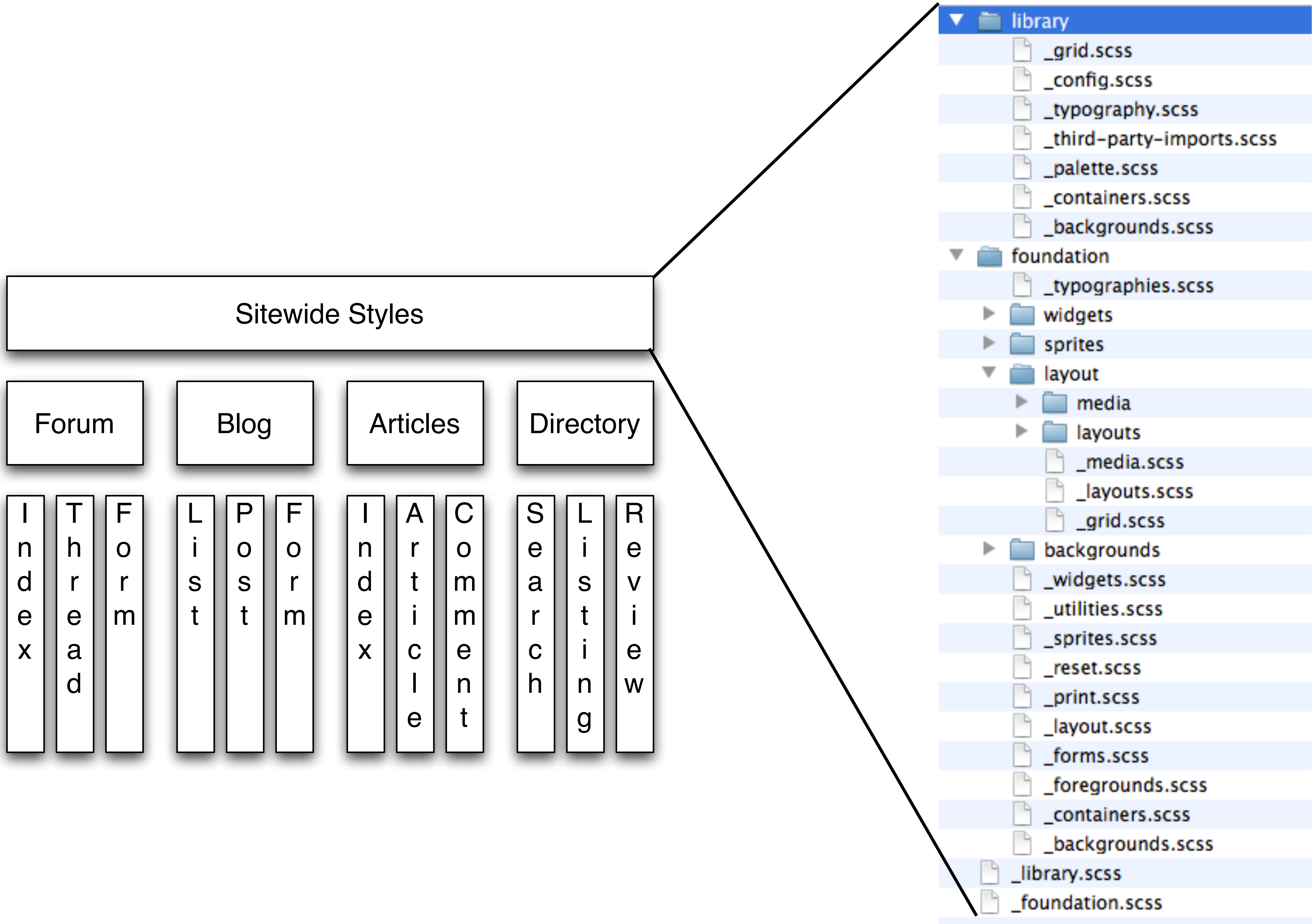
Convention: Each folder has a file of the same name that imports the files contained within it in the correct order.

Problem:
Shared Stylesheets

Styles leak into places you didn’t intend

Sass.

compass



Sitewide Styles

Forum

Blog

Articles

Directory

Index

Thread

Form

List

Post

Form

Index

Article

Comment

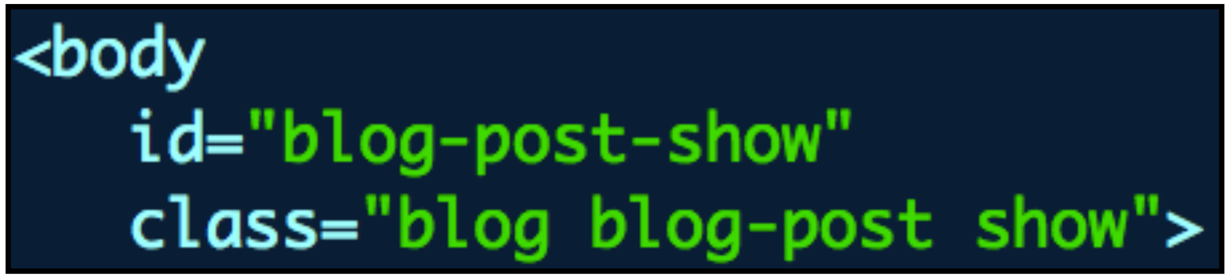
Search

Listing

Review

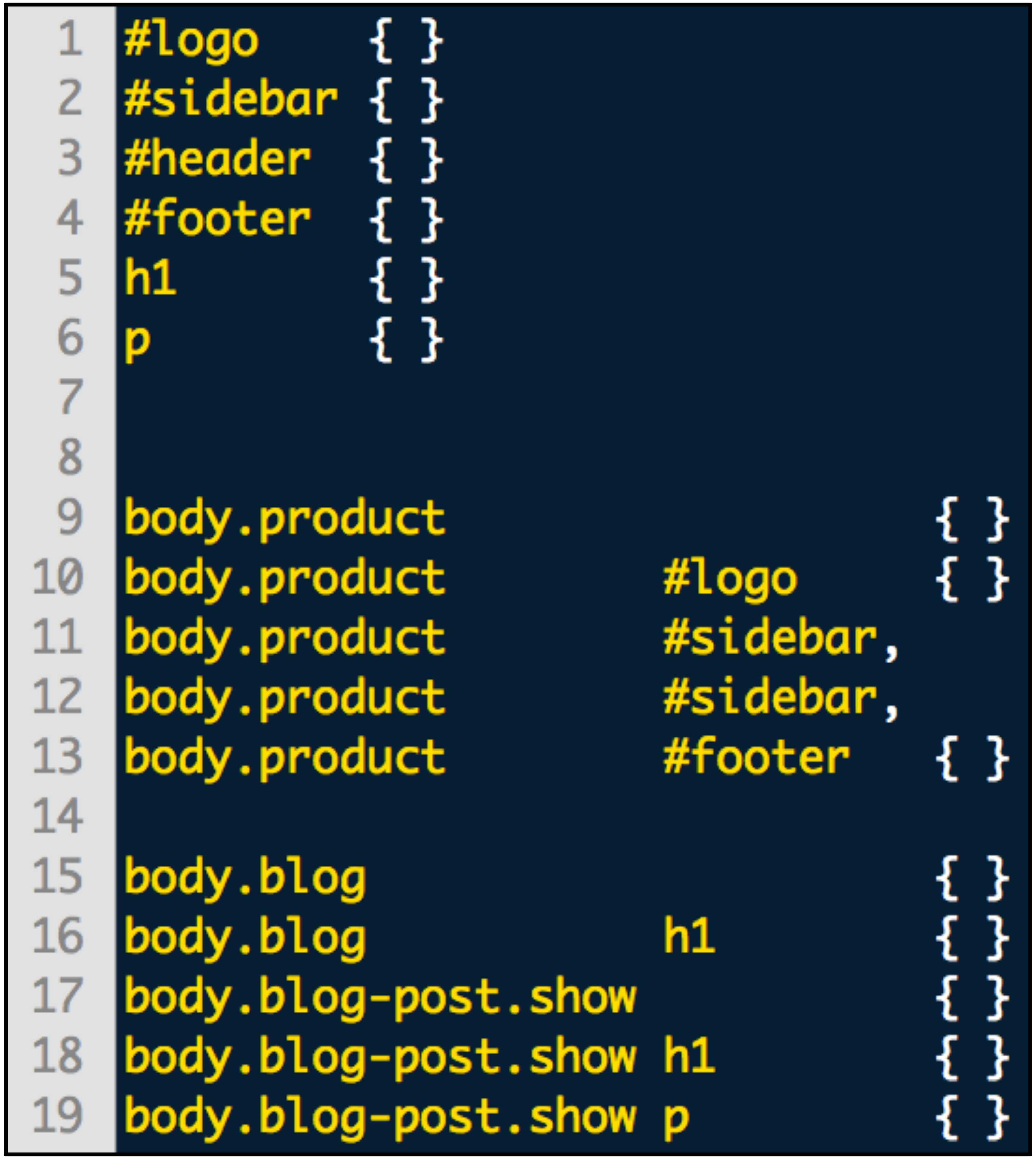
```
▼ library
      _grid.scss
      _config.scss
      _typography.scss
      _third-party-imports.scss
      _palette.scss
      _containers.scss
      _backgrounds.scss
▼ foundation
      _typographies.scss
   ▶ widgets
   ▶ sprites
   ▼ layout
      ▶ media
      ▶ layouts
         _media.scss
         _layouts.scss
         _grid.scss
   ▶ backgrounds
      _widgets.scss
      _utilities.scss
      _sprites.scss
      _reset.scss
      _print.scss
      _layout.scss
      _forms.scss
      _foregrounds.scss
      _containers.scss
      _backgrounds.scss
   _library.scss
   _foundation.scss
```

Solution:
Scoped Selectors



```
<body
   id="blog-post-show"
   class="blog blog-post show">
```

compass

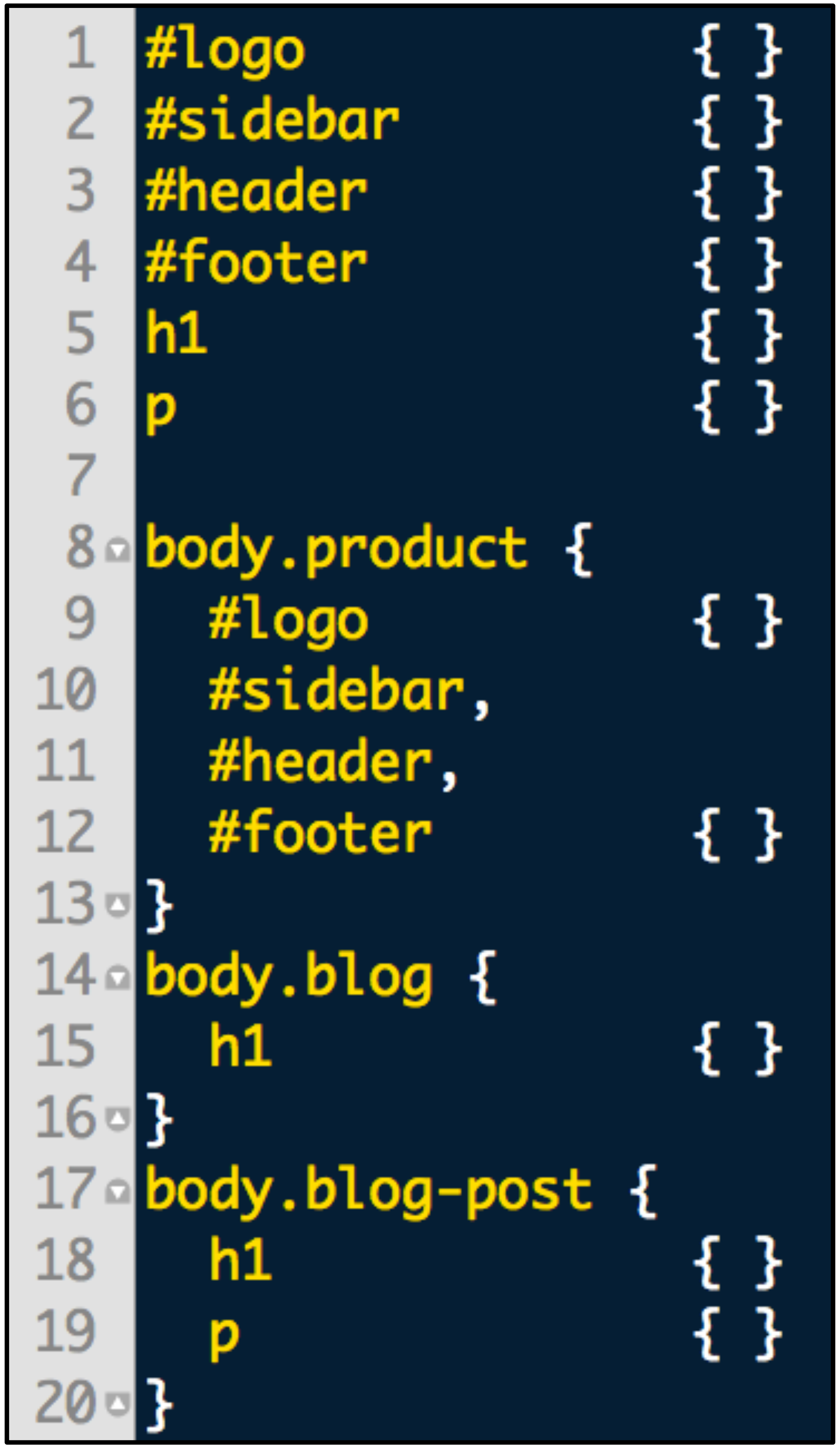


```
#logo    { }
#sidebar { }
#header  { }
#footer  { }
h1       { }
p        { }


body.product                  { }
body.product        #logo     { }
body.product        #sidebar,
body.product        #sidebar,
body.product        #footer   { }

body.blog                     { }
body.blog           h1        { }
body.blog-post.show           { }
body.blog-post.show h1        { }
body.blog-post.show p         { }
```

CSS gives you carpal tunnel from typing your scopes OVER & OVER AGAIN

Sass.

compass

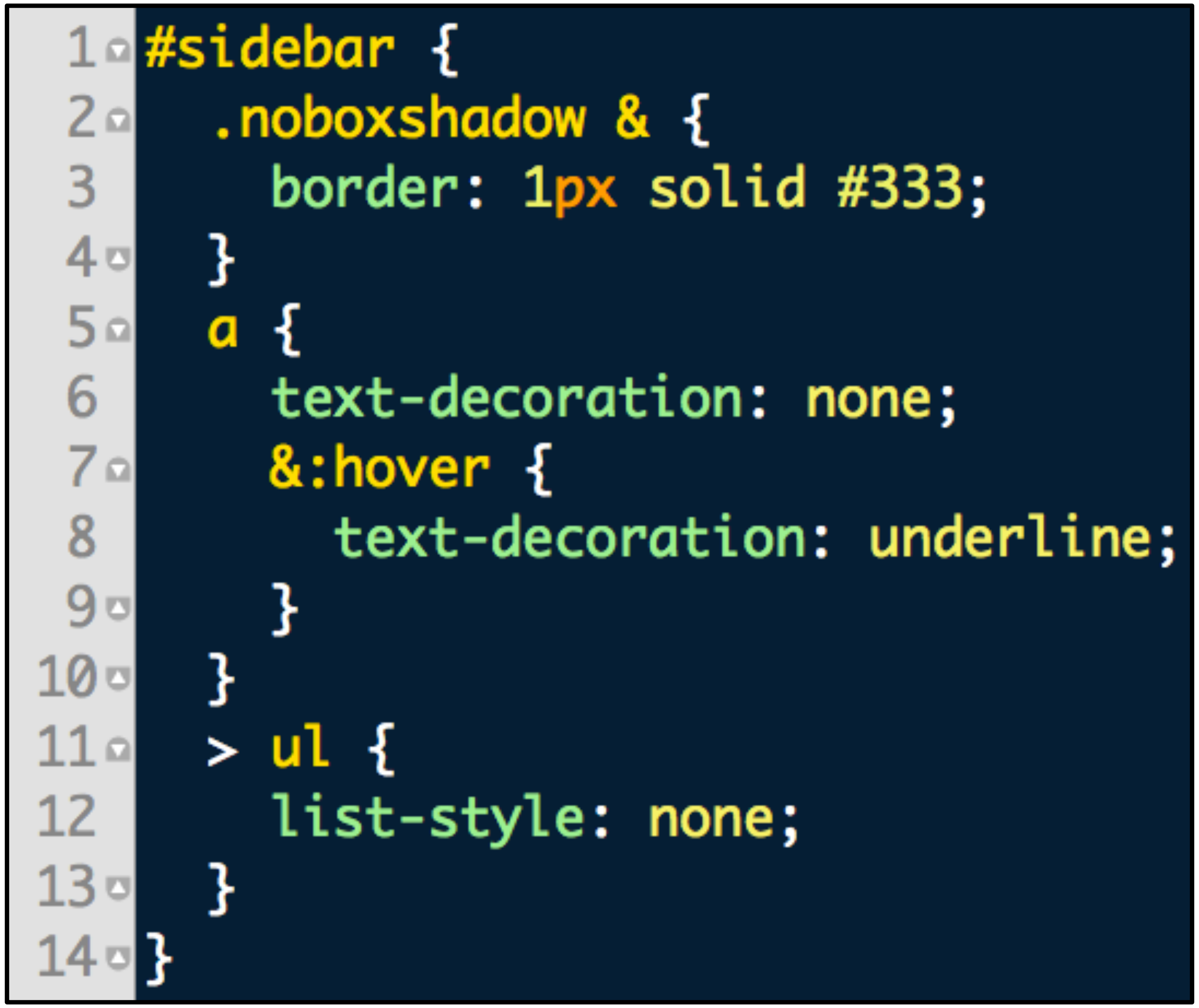


```
#logo            { }
#sidebar         { }
#header          { }
#footer          { }
h1               { }
p                { }

body.product {
  #logo          { }
  #sidebar,
  #header,
  #footer        { }
}
body.blog {
  h1             { }
}
body.blog-post {
  h1             { }
  p              { }
}
```

Sass Provides
DRY Selectors
via **nesting**

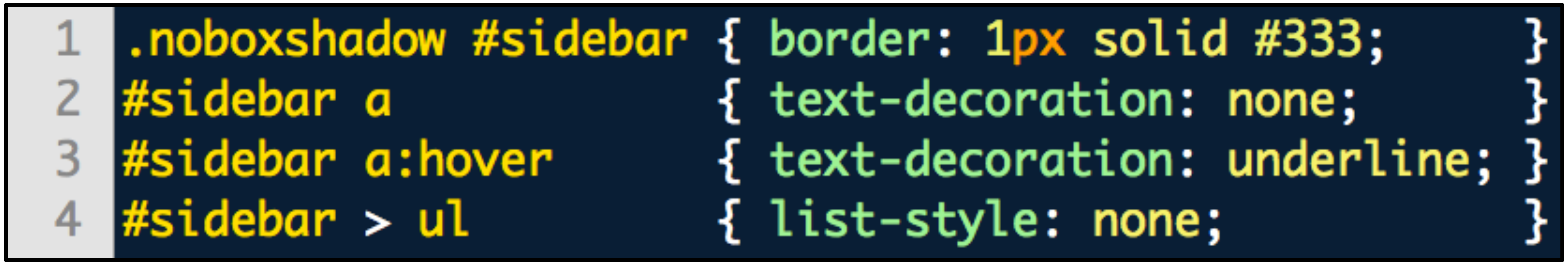
Nested Selectors

- Default: descendant
- & - context reference
- Works with combinators



```
#sidebar {
  .noboxshadow & {
    border: 1px solid #333;
  }
  a {
    text-decoration: none;
    &:hover {
      text-decoration: underline;
    }
  }
  > ul {
    list-style: none;
  }
}
```

Compiles to

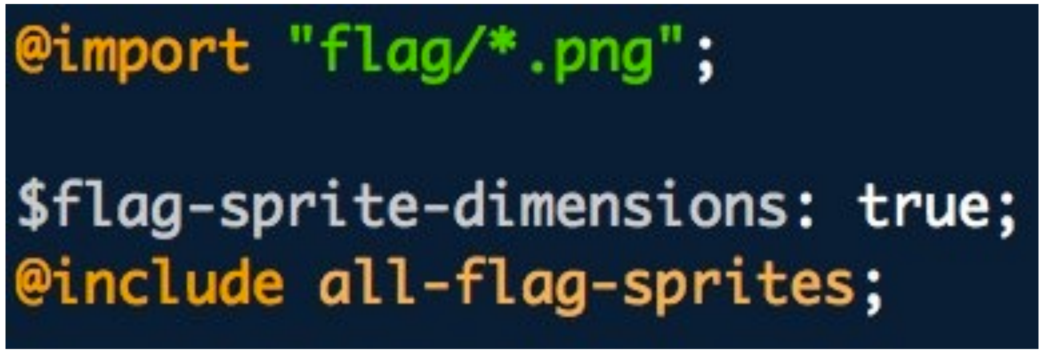


```
.noboxshadow #sidebar { border: 1px solid #333;     }
#sidebar a            { text-decoration: none;      }
#sidebar a:hover      { text-decoration: underline; }
#sidebar > ul         { list-style: none;           }
```

Problem:
Dozens of Images

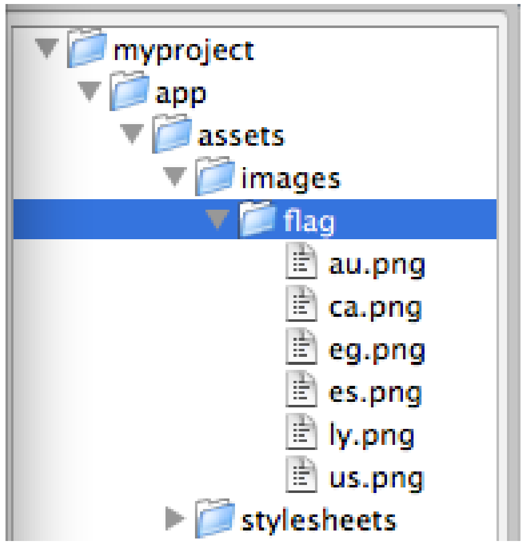
Spriting is annoying

Solution:
Compass Sprites

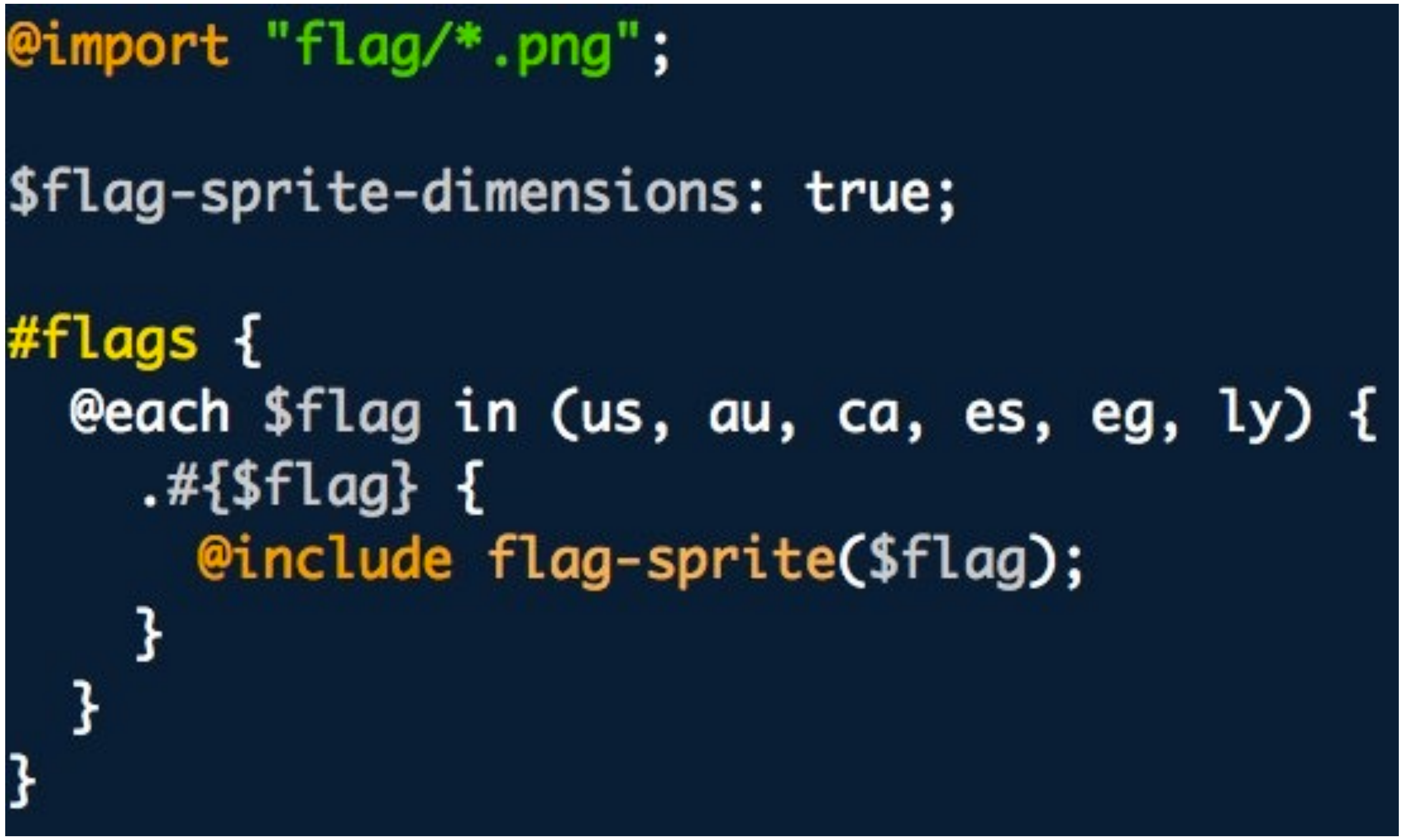


```
@import "flag/*.png";

$flag-sprite-dimensions: true;
@include all-flag-sprites;
```

Simple Sprites (Compass makes classes for you)

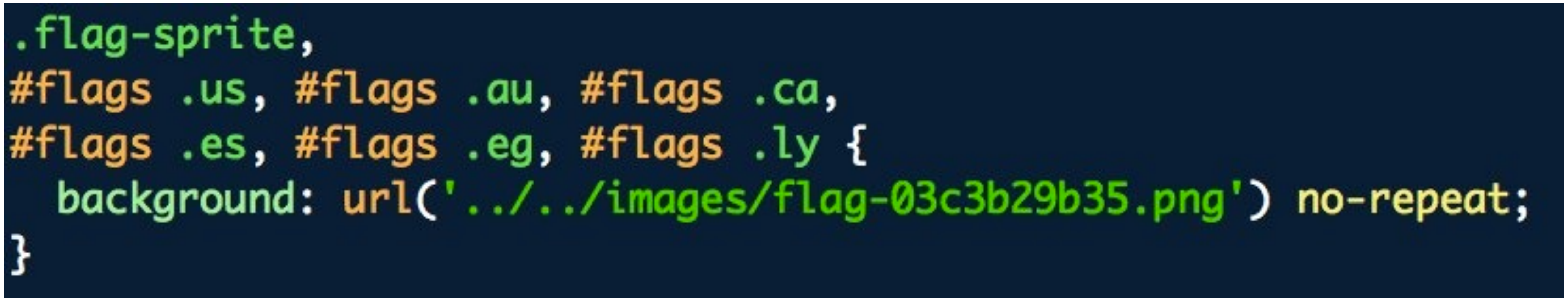


```
▼ myproject
  ▼ app
    ▼ assets
      ▼ images
        ▼ flag
            au.png
            ca.png
            eg.png
            es.png
            ly.png
            us.png
      ▶ stylesheets
```

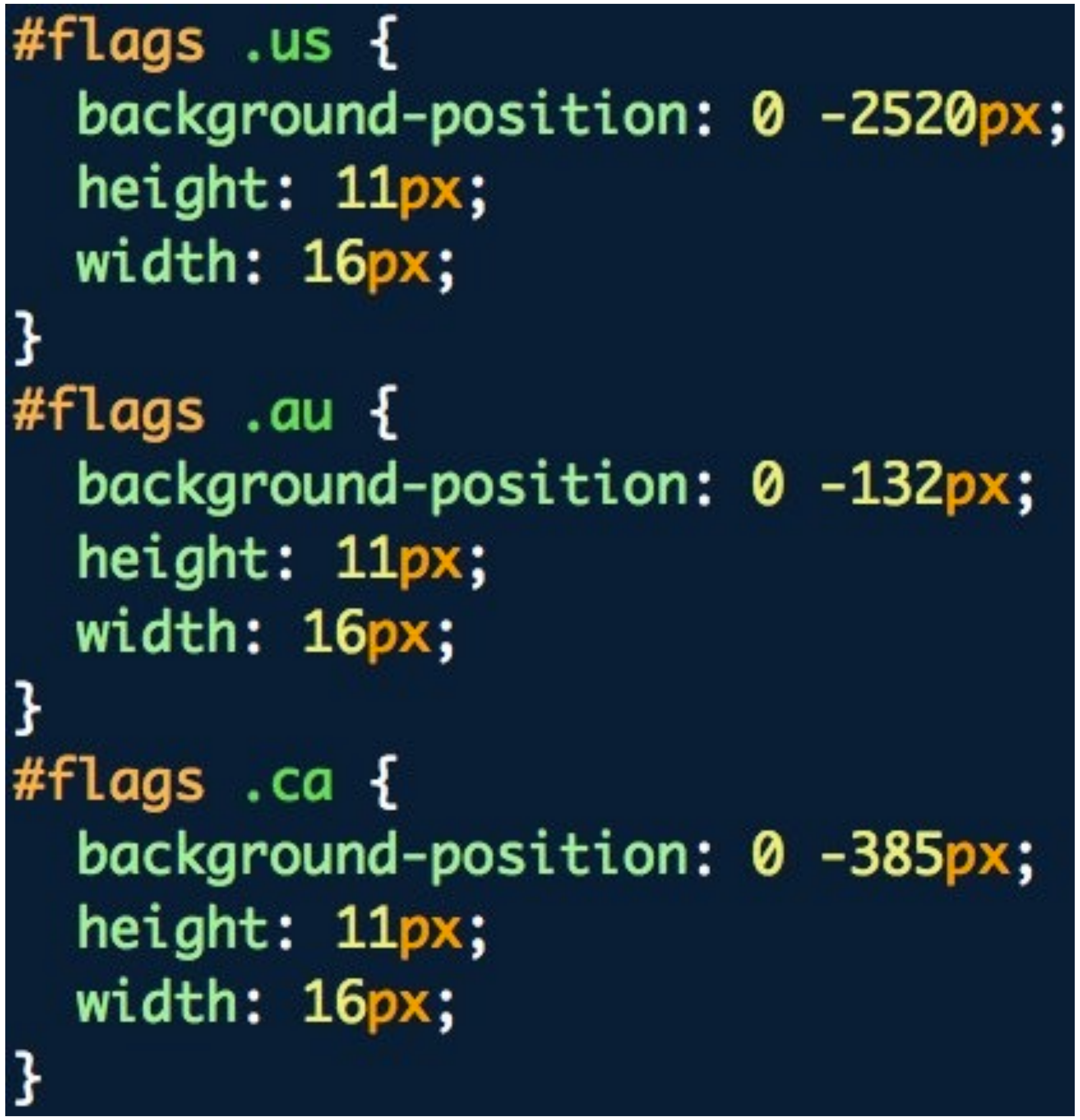


```
@import "flag/*.png";

$flag-sprite-dimensions: true;

#flags {
  @each $flag in (us, au, ca, es, eg, ly) {
    .#{$flag} {
      @include flag-sprite($flag);
    }
  }
}
```

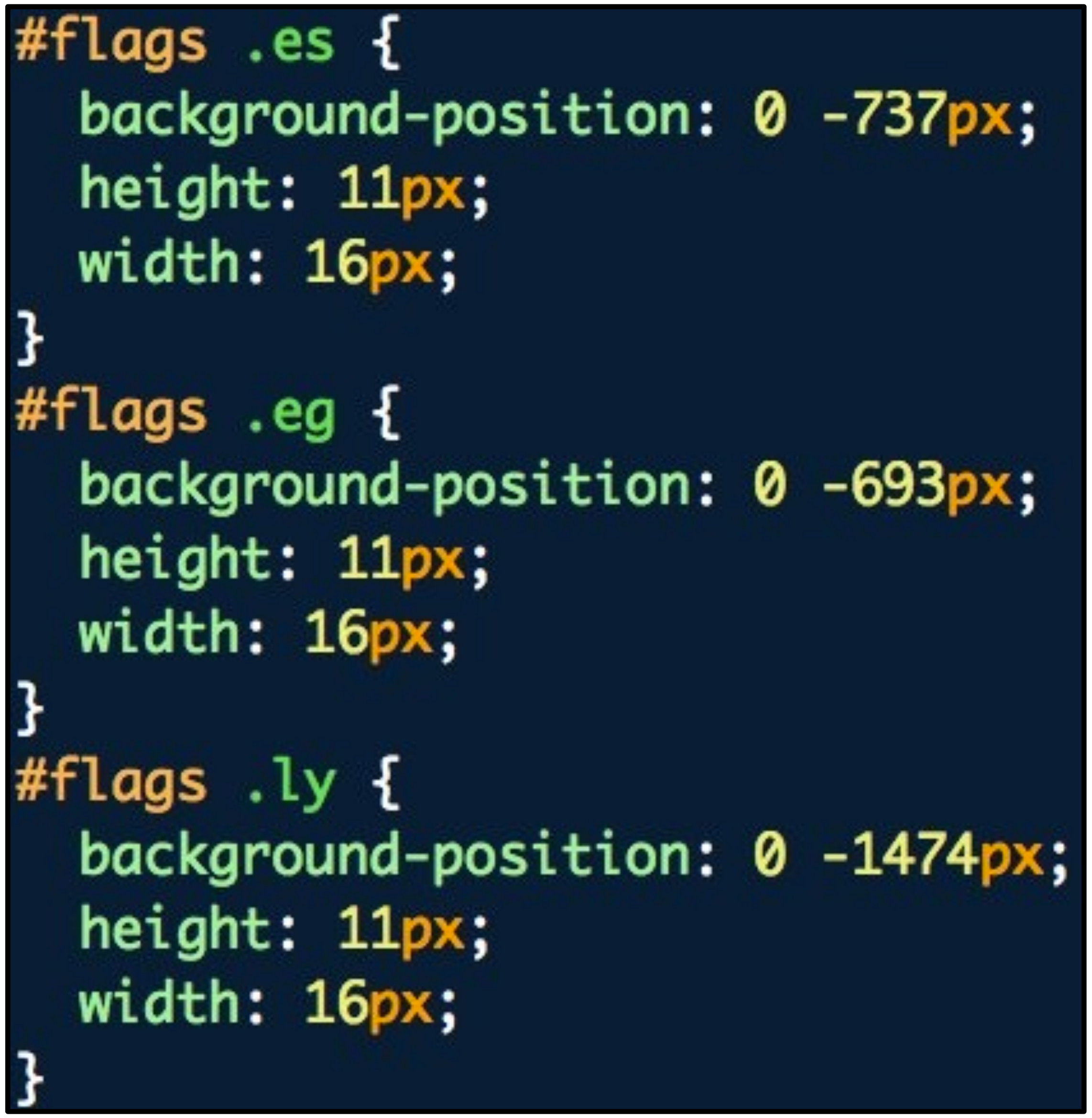
Or you can define your own selectors.



```
.flag-sprite,
#flags .us, #flags .au, #flags .ca,
#flags .es, #flags .eg, #flags .ly {
  background: url('../../images/flag-03c3b29b35.png') no-repeat;
}
```



```
#flags .us {
  background-position: 0 -2520px;
  height: 11px;
  width: 16px;
}
#flags .au {
  background-position: 0 -132px;
  height: 11px;
  width: 16px;
}
#flags .ca {
  background-position: 0 -385px;
  height: 11px;
  width: 16px;
}
```



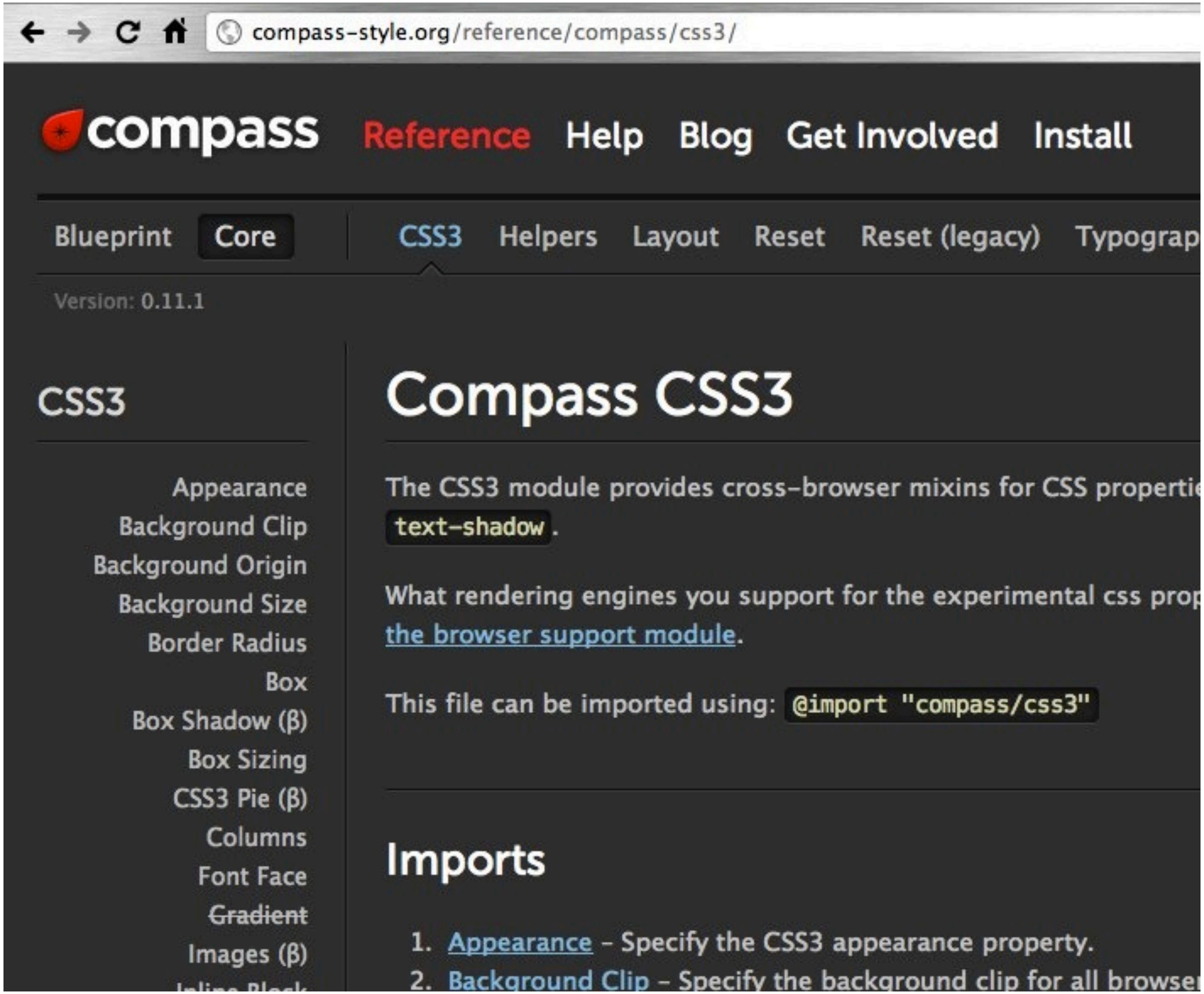
```
#flags .es {
  background-position: 0 -737px;
  height: 11px;
  width: 16px;
}
#flags .eg {
  background-position: 0 -693px;
  height: 11px;
  width: 16px;
}
#flags .ly {
  background-position: 0 -1474px;
  height: 11px;
  width: 16px;
}
```

Problem:
Dozens More Images

Solution:
Use CSS3 Instead

Abstract vendor prefixes away

compass



Compass CSS3

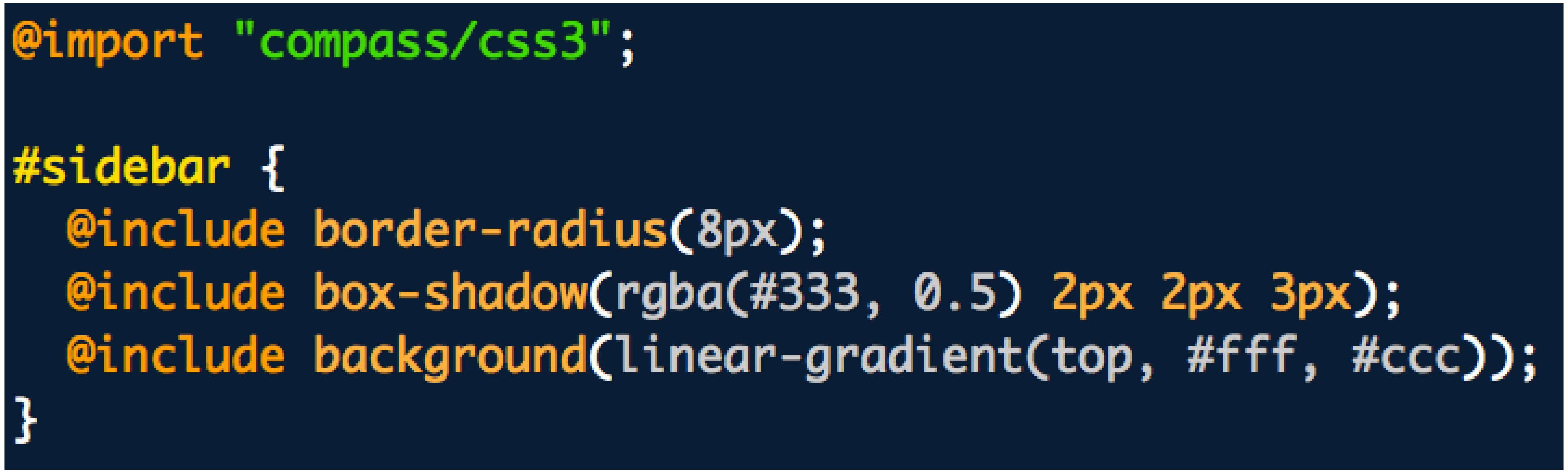
The CSS3 module provides cross-browser mixins for CSS properti
`text-shadow`.

What rendering engines you support for the experimental css prop
the browser support module.

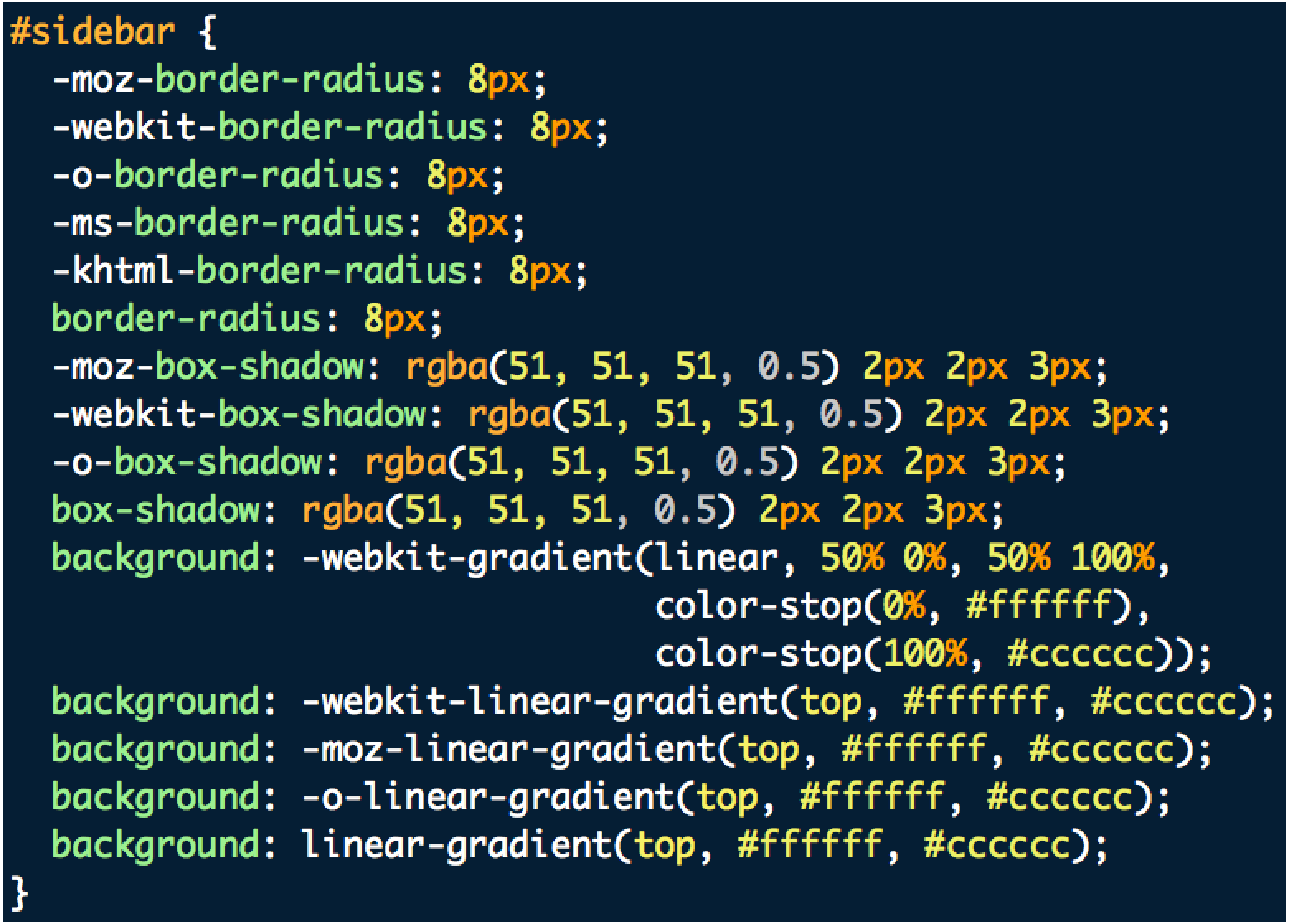
This file can be imported using: `@import "compass/css3"`

Imports

1. Appearance - Specify the CSS3 appearance property.
2. Background Clip - Specify the background clip for all browse



```
@import "compass/css3";

#sidebar {
  @include border-radius(8px);
  @include box-shadow(rgba(#333, 0.5) 2px 2px 3px);
  @include background(linear-gradient(top, #fff, #ccc));
}
```



```
#sidebar {
  -moz-border-radius: 8px;
  -webkit-border-radius: 8px;
  -o-border-radius: 8px;
  -ms-border-radius: 8px;
  -khtml-border-radius: 8px;
  border-radius: 8px;
  -moz-box-shadow: rgba(51, 51, 51, 0.5) 2px 2px 3px;
  -webkit-box-shadow: rgba(51, 51, 51, 0.5) 2px 2px 3px;
  -o-box-shadow: rgba(51, 51, 51, 0.5) 2px 2px 3px;
  box-shadow: rgba(51, 51, 51, 0.5) 2px 2px 3px;
  background: -webkit-gradient(linear, 50% 0%, 50% 100%,
                               color-stop(0%, #ffffff),
                               color-stop(100%, #cccccc));
  background: -webkit-linear-gradient(top, #ffffff, #cccccc);
  background: -moz-linear-gradient(top, #ffffff, #cccccc);
  background: -o-linear-gradient(top, #ffffff, #cccccc);
  background: linear-gradient(top, #ffffff, #cccccc);
}
```

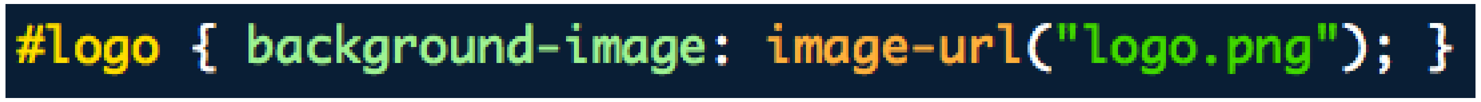
Best Practice:
Use a CDN / Asset Hosts

Load assets faster

Problem:
Image URLs Change

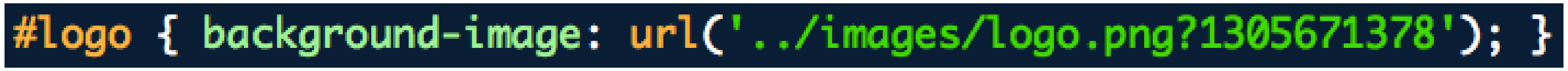
Solution:

Use URL Helpers

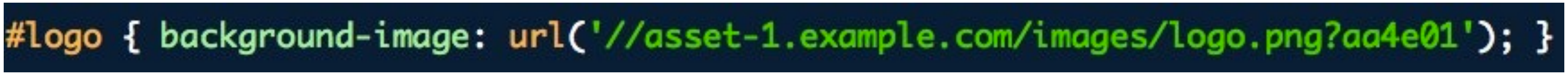


```
#logo { background-image: image-url("logo.png"); }
```

Compiles to



```
#logo { background-image: url('../images/logo.png?1305671378'); }
```

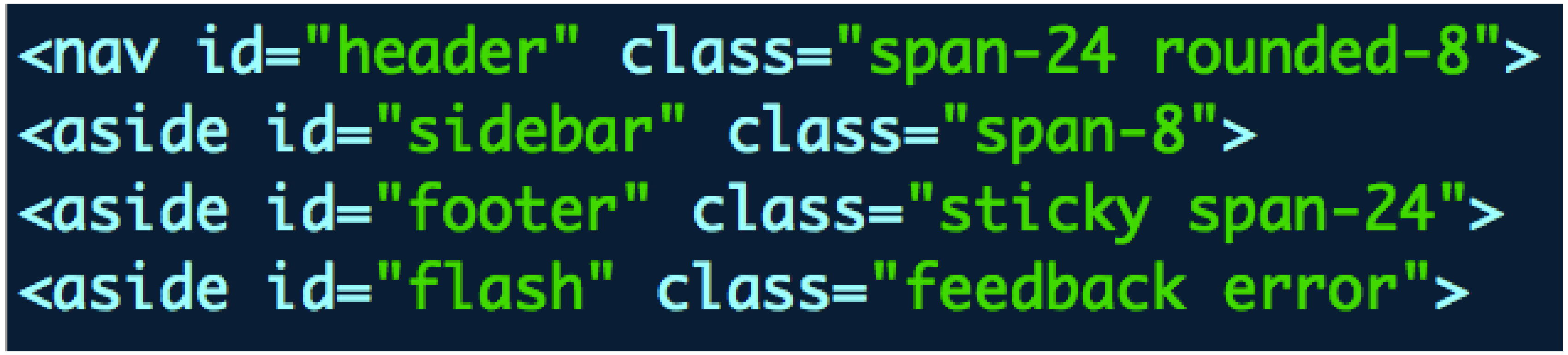


```
#logo { background-image: url('//asset-1.example.com/images/logo.png?aa4e01'); }
```

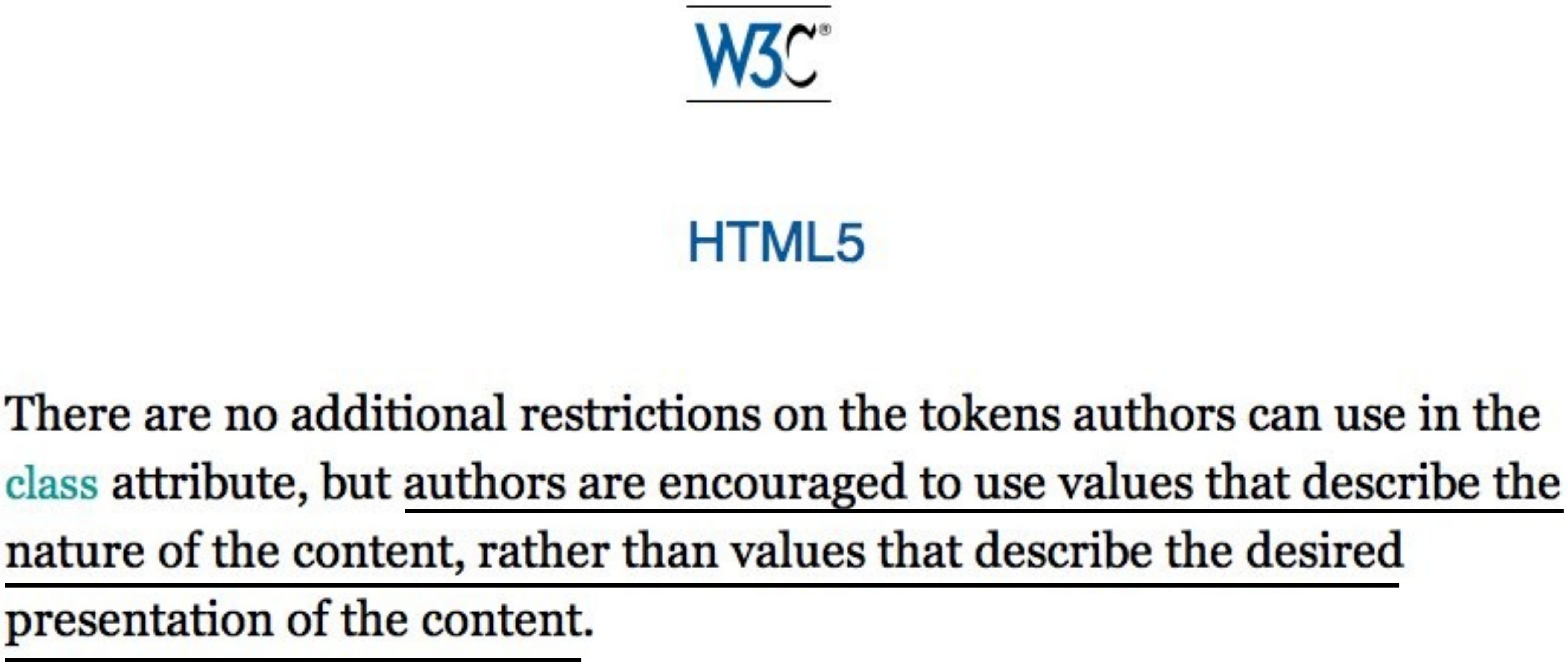
Best Practice:

Separate Content & Presentation

Problem:
Classitis



```
<nav id="header" class="span-24 rounded-8">
<aside id="sidebar" class="span-8">
<aside id="footer" class="sticky span-24">
<aside id="flash" class="feedback error">
```



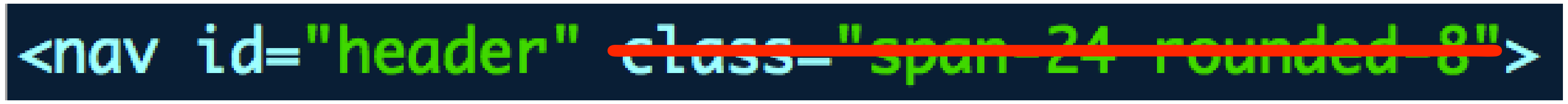
HTML5

There are no additional restrictions on the tokens authors can use in the class attribute, but authors are encouraged to use values that describe the nature of the content, rather than values that describe the desired presentation of the content.

Solution:

Move Presentation Concerns

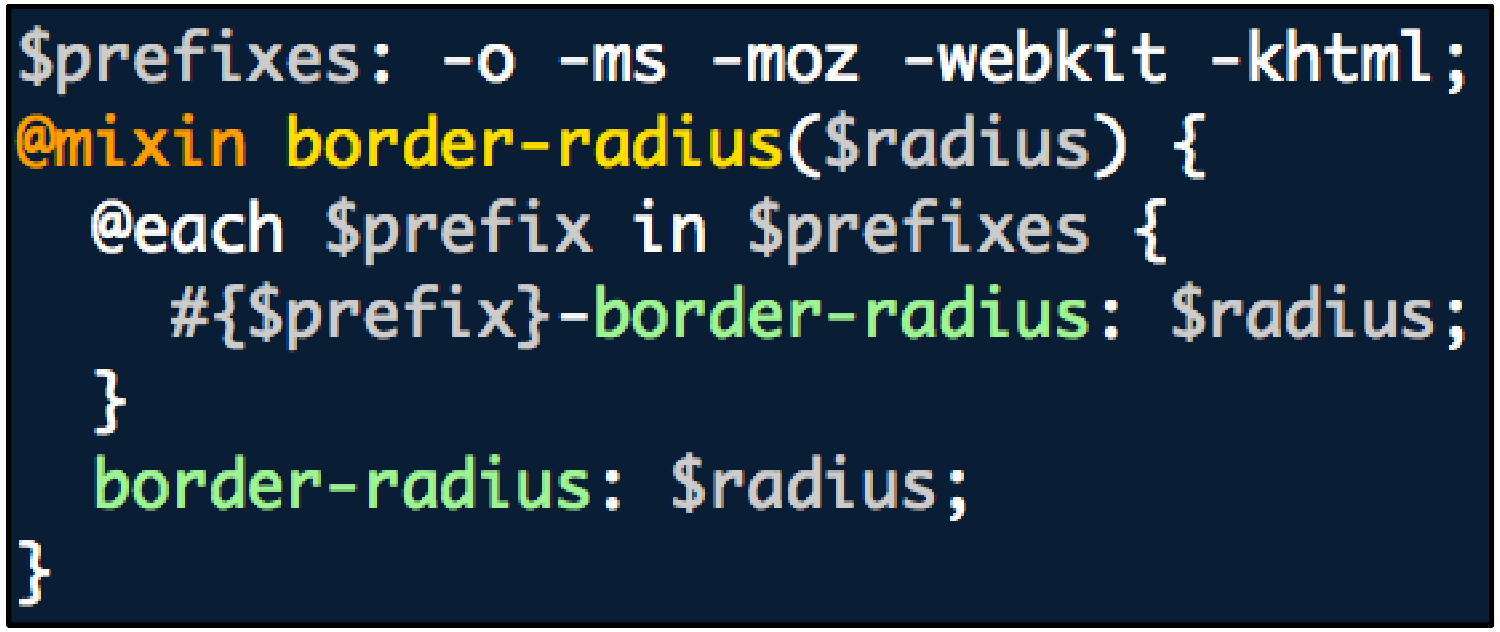
They belong in your stylesheets



<nav id="header" ~~class="span-24 rounded-8"~~>

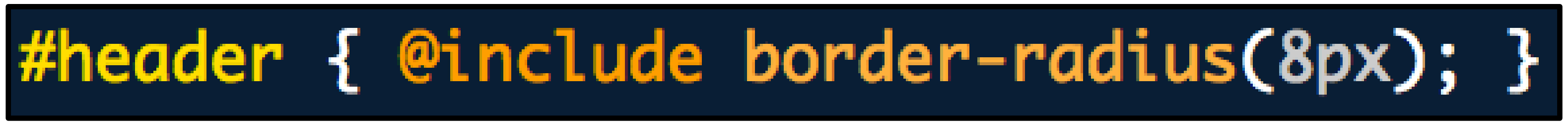
This is all you should need.

Mixins



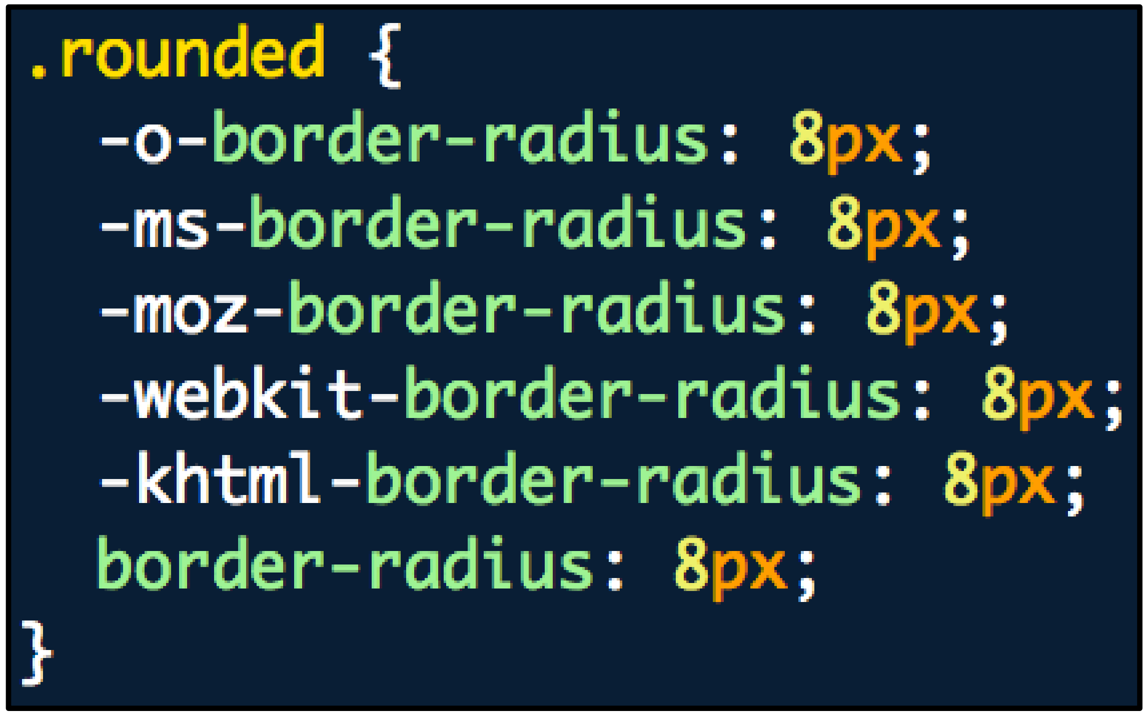
```
$prefixes: -o -ms -moz -webkit -khtml;
@mixin border-radius($radius) {
  @each $prefix in $prefixes {
    #{$prefix}-border-radius: $radius;
  }
  border-radius: $radius;
}
```

+



```
#header { @include border-radius(8px); }
```

Compiles to

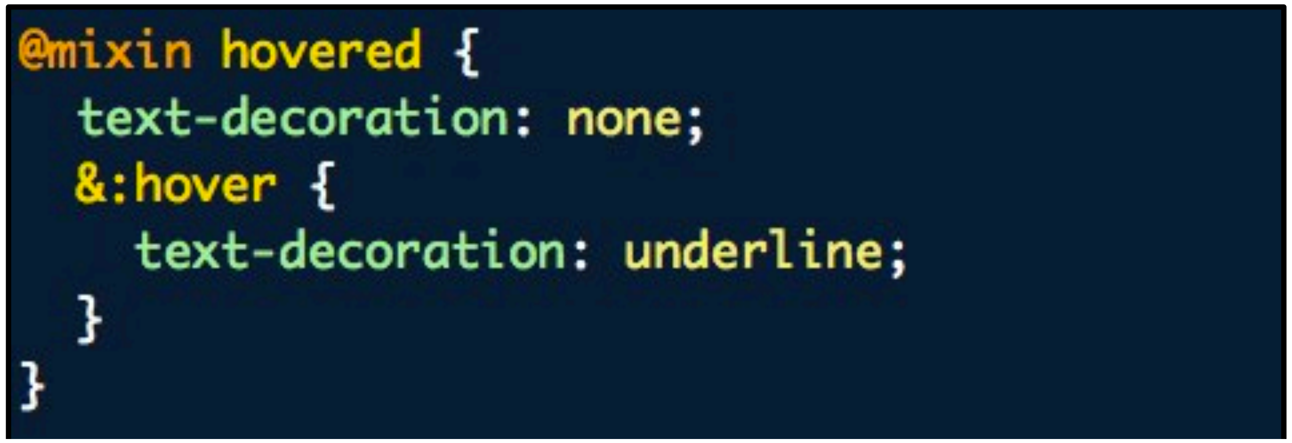


```
.rounded {
  -o-border-radius: 8px;
  -ms-border-radius: 8px;
  -moz-border-radius: 8px;
  -webkit-border-radius: 8px;
  -khtml-border-radius: 8px;
  border-radius: 8px;
}
```

Mixins

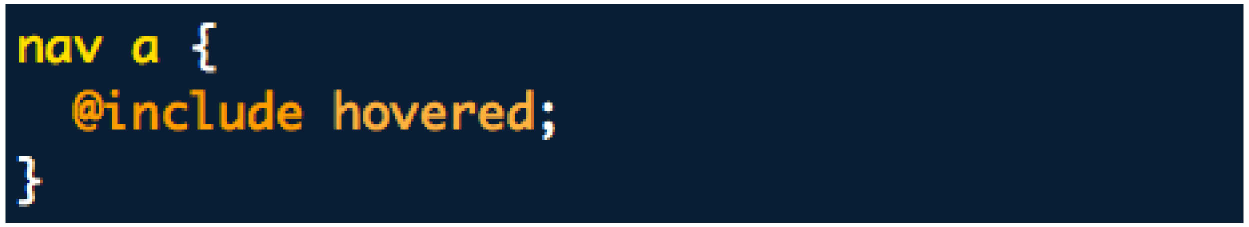
Create a design vocabulary of common patterns independent from your markup's selector domain

More Mixins



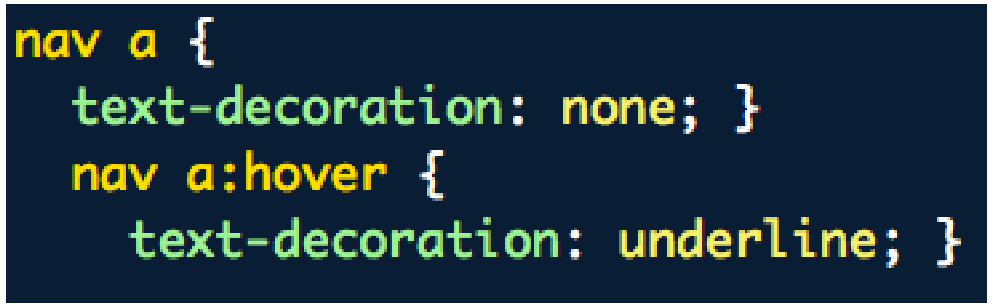
```
@mixin hovered {
  text-decoration: none;
  &:hover {
    text-decoration: underline;
  }
}
```

+



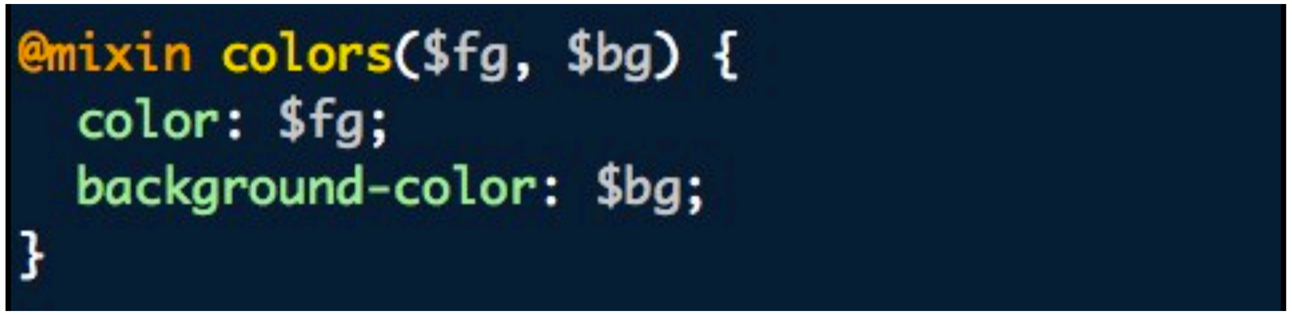
```
nav a {
  @include hovered;
}
```

Compiles to



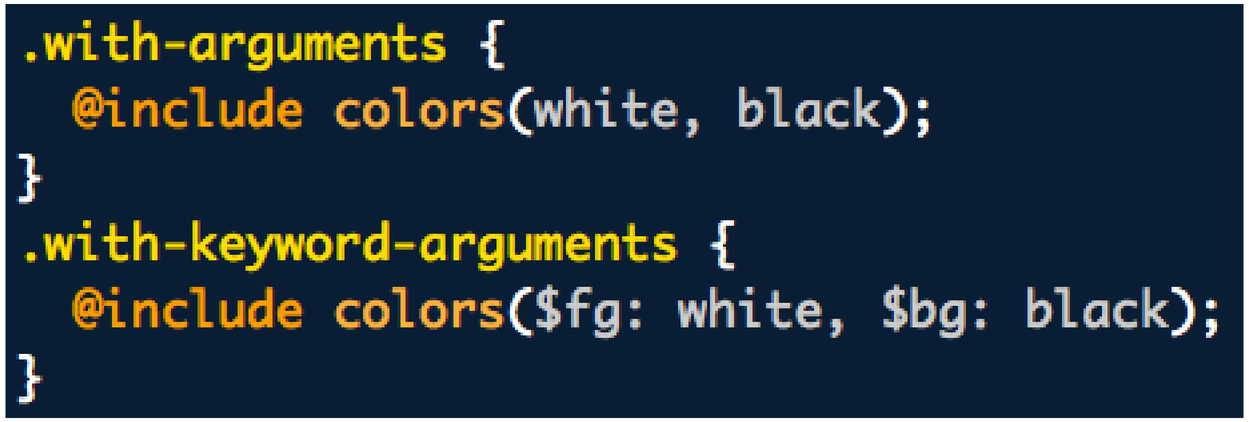
```
nav a {
  text-decoration: none; }
  nav a:hover {
    text-decoration: underline; }
```

More Mixins



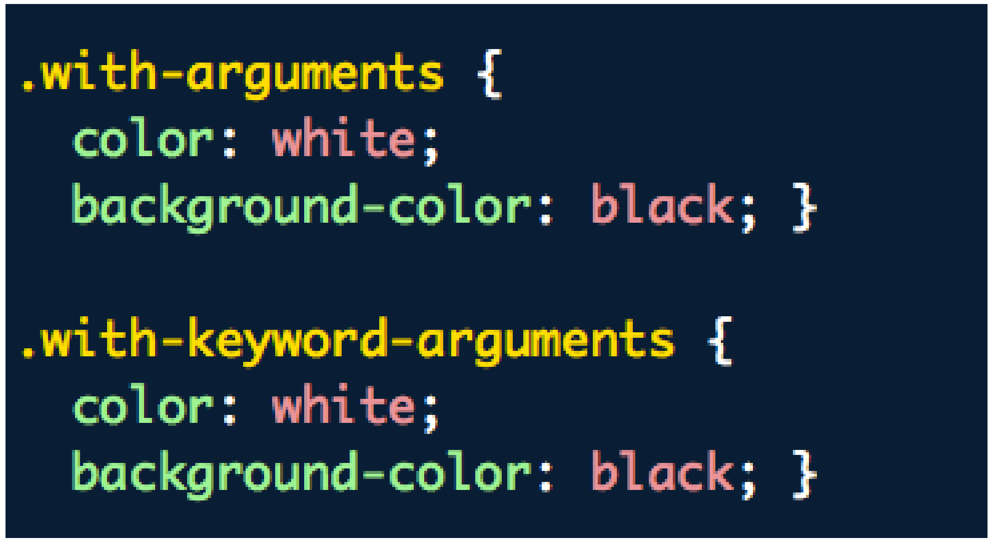
```
@mixin colors($fg, $bg) {
  color: $fg;
  background-color: $bg;
}
```

+



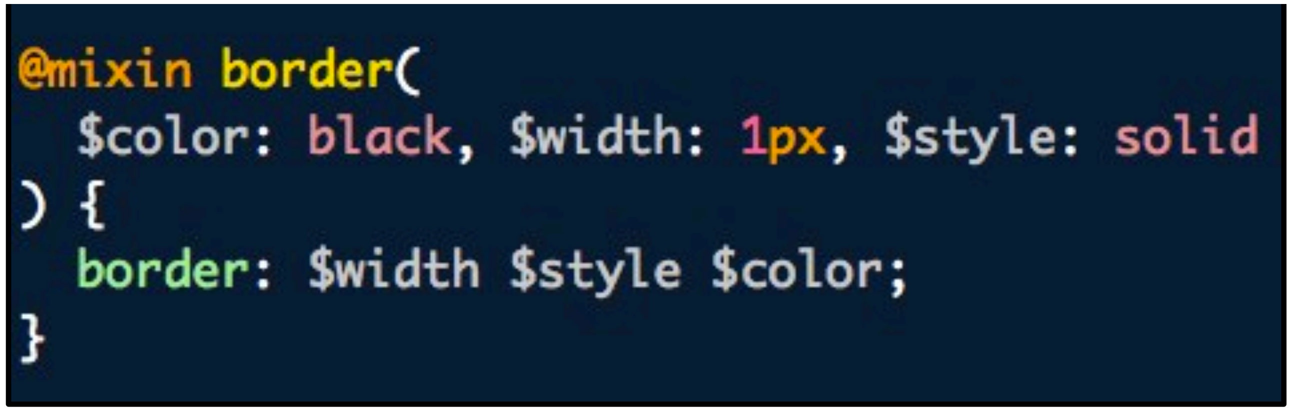
```
.with-arguments {
  @include colors(white, black);
}
.with-keyword-arguments {
  @include colors($fg: white, $bg: black);
}
```

Compiles to →



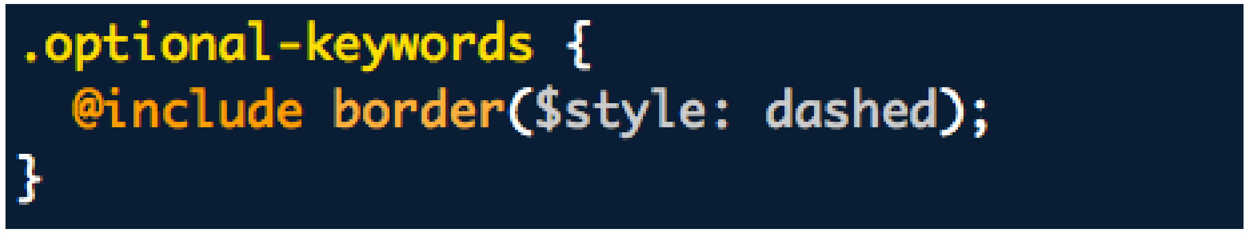
```
.with-arguments {
  color: white;
  background-color: black; }

.with-keyword-arguments {
  color: white;
  background-color: black; }
```

More Mixins



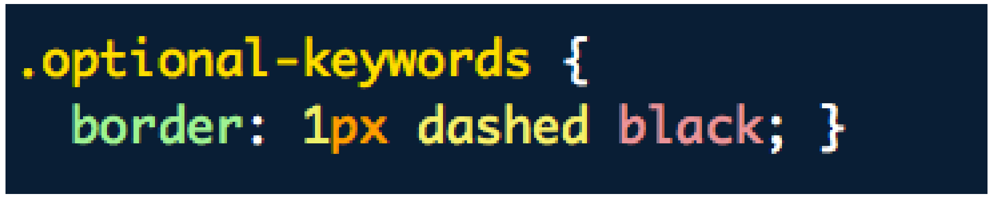
```
@mixin border(
  $color: black, $width: 1px, $style: solid
) {
  border: $width $style $color;
}
```

+

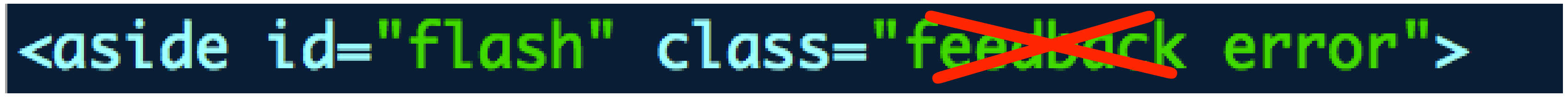


```
.optional-keywords {
  @include border($style: dashed);
}
```

Compiles to →

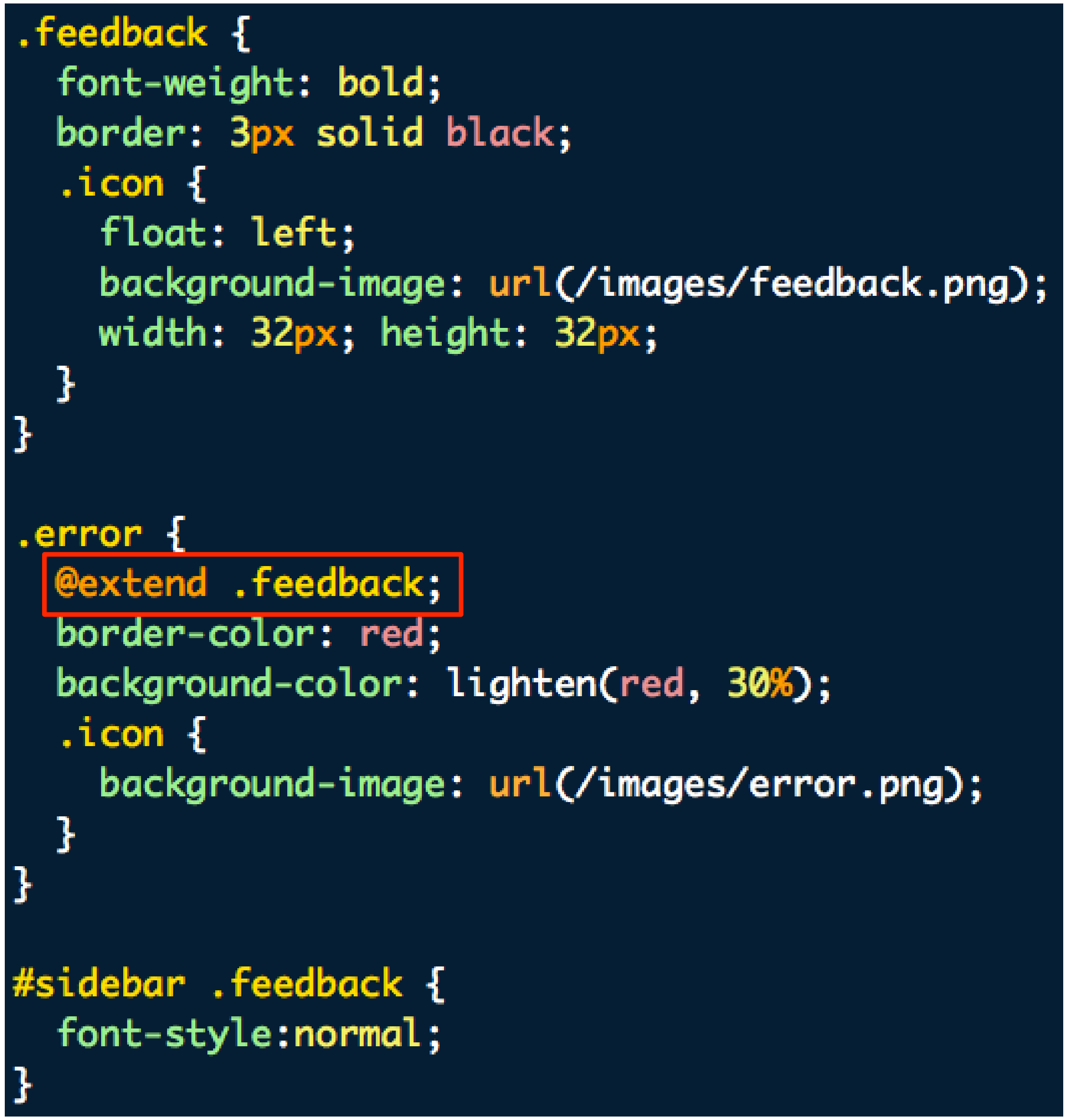


```
.optional-keywords {
  border: 1px dashed black; }
```



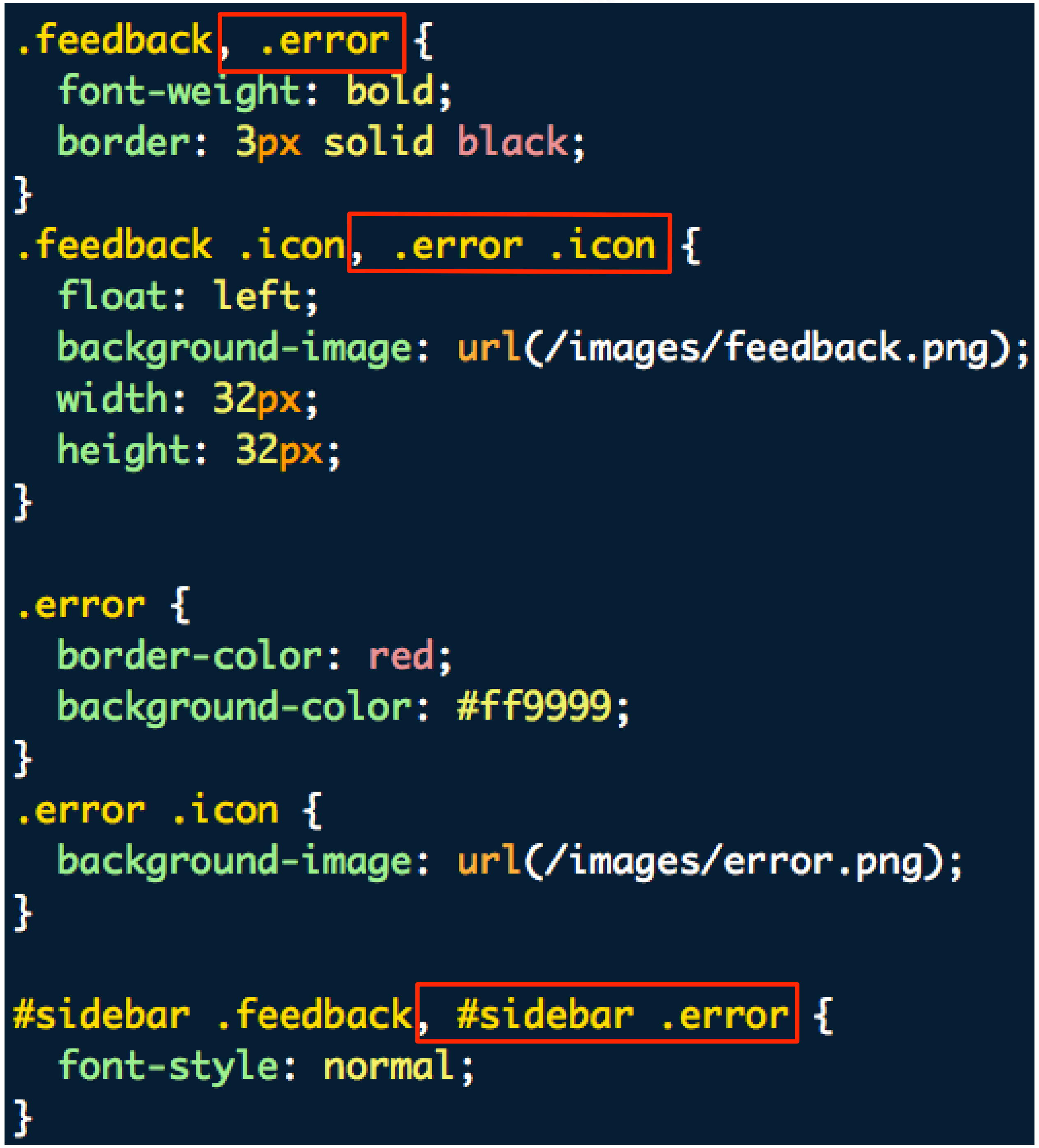
```
<aside id="flash" class="~~feedback~~ error">
```

This is all you should need.



```
.feedback {
  font-weight: bold;
  border: 3px solid black;
  .icon {
    float: left;
    background-image: url(/images/feedback.png);
    width: 32px; height: 32px;
  }
}

.error {
  @extend .feedback;
  border-color: red;
  background-color: lighten(red, 30%);
  .icon {
    background-image: url(/images/error.png);
  }
}

#sidebar .feedback {
  font-style:normal;
}
```

Compiles to →



```
.feedback, .error {
  font-weight: bold;
  border: 3px solid black;
}
.feedback .icon, .error .icon {
  float: left;
  background-image: url(/images/feedback.png);
  width: 32px;
  height: 32px;
}

.error {
  border-color: red;
  background-color: #ff9999;
}
.error .icon {
  background-image: url(/images/error.png);
}

#sidebar .feedback, #sidebar .error {
  font-style: normal;
}
```

Selector Inheritance

We’ve Only Scratched the Surface

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

My Teammates

Sass Core Team

Nathan Wiezenbaum (@nex3)

Hampton Catlin (@hcatlin)

The Compass Core Team

Eric Meyer (@eriiicam)

Brandon Mathis (@imathis)

Scott Davis (@jetviper21)

Nico Hagenburger (@hagenburger)

Sass.

compass

Compass is Charityware

UNITED
MITOCHONDRIAL
DISEASE
FOUNDATION.

HOPE. ENERGY. LIFE.

If you use Compass and it makes your life better, please help make someone else’s life better by making a tax-deductible donation to the United Mitochondrial Disease Foundation.
http://umdf.org/compass