

CSS Preprocessors

Extending CSS with Sass to Provide Extra Functionality

R. Scott Granneman & Jans Carton

© 2016 R. Scott Granneman

Last updated 2020-07-13

You are free to use this work, with certain restrictions.

For full licensing information, please see the last slide/page.

Notes & URLs for this presentation can be found...

- » underneath the link to this slide show on granneman.com
- » at files.granneman.com/presentations/webdev/CSS-Preprocessors.txt

Background

What's a *preprocessor*?

A program that processes input data & produces output that is then used as input to another program

Input → Preprocessor → Output/Input → Program

Input



Preprocessor



Output/Input



Program

Markdown-formatted file



Markdown converter



HTML



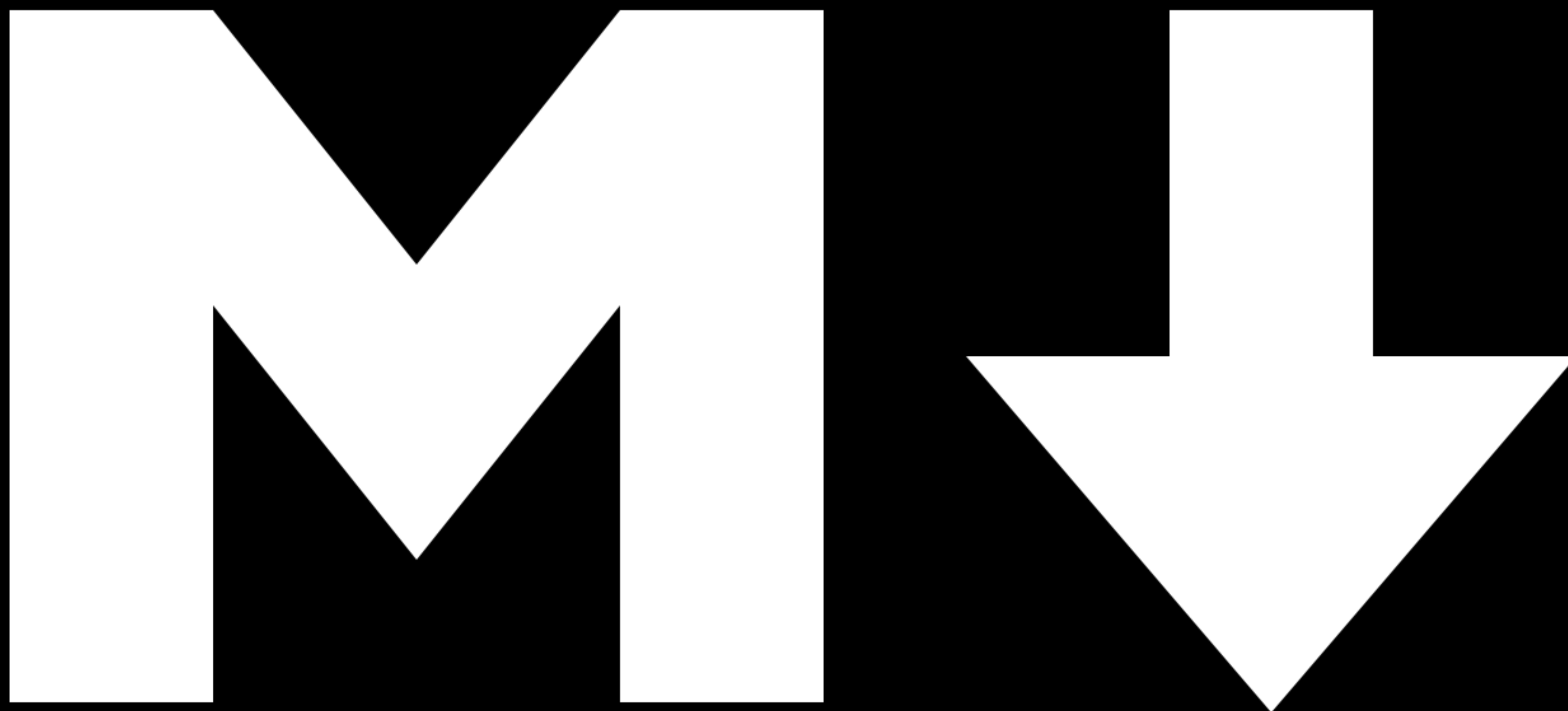
Browser rendering engine

Markdown, Textile, & AsciiDoc → HTML

TypeScript → JavaScript

Sass, Less, & PostCSS → CSS

Common pre-processors



Markdown is turned into HTML by default

Can also be exported to PDF, Word, EPUB, HTML slide shows, & much more

Some sample Markdown syntax

Paragraph

Blank lines before & after

The older matters which had made the sculptor's dream and bas-relief so significant to my uncle formed the subject of the second half of his long manuscript. Once before, it appears, Professor Angell had seen the hellish outlines of the nameless monstrosity, puzzled over the unknown hieroglyphics, and heard the ominous syllables which can be rendered only as "Cthulhu"; and all this in so stirring and horrible a connexion that it is small wonder he pursued young Wilcox with queries and demands for data.

The earlier experience had come in 1908, seventeen years before, when the American Archaeological Society held its annual meeting in St. Louis. Professor Angell, as befitted one of his authority and attainments, had had a prominent part in all the deliberations; and was one of the first to be approached by the several outsiders who took advantage of the convocation to offer questions for correct answering and problems for expert solution.

The chief of these outsiders, and in a short time the focus of interest for the entire meeting, was a commonplace-looking middle-aged man who had travelled all the way from New Orleans for certain special information unobtainable from any local source. His name was John Raymond Legrasse, and he was by profession an Inspector of Police. With him he bore the subject of his visit, a grotesque, repulsive, and apparently very ancient stone

The older matters which had made the sculptor's dream and bas-relief so significant to my uncle formed the subject of the second half of his long manuscript. Once before, it appears, Professor Angell had seen the hellish outlines of the nameless monstrosity, puzzled over the unknown hieroglyphics, and heard the ominous syllables which can be rendered only as "Cthulhu"; and all this in so stirring and horrible a connexion that it is small wonder he pursued young Wilcox with queries and demands for data.

The earlier experience had come in 1908, seventeen years before, when the American Archaeological Society held its annual meeting in St. Louis. Professor Angell, as befitted one of his authority and attainments, had had a prominent part in all the deliberations; and was one of the first to be approached by the several outsiders who took advantage of the convocation to offer questions for correct answering and problems for expert solution.

The chief of these outsiders, and in a short time the focus of interest for the entire meeting, was a commonplace-looking middle-aged man who had travelled all the way from New Orleans for certain special information unobtainable from any local source. His name was John Raymond Legrasse, and he was by profession an Inspector of Police. With him he bore the subject of his visit, a grotesque, repulsive, and apparently very ancient stone statuette whose origin he was at a loss to determine.

<p>The older matters which had made the sculptor's dream and bas-relief so significant to my uncle formed the subject of the second half of his long manuscript. Once before, it appears, Professor Angell had seen the hellish outlines of the nameless monstrosity, puzzled over the unknown hieroglyphics, and heard the ominous syllables which can be rendered only as "Cthulhu"; and all this in so stirring and horrible a connexion that it is small wonder he pursued young Wilcox with queries and demands for data.</p>

<p>The earlier experience had come in 1908, seventeen years before, when the American Archaeological Society held its annual meeting in St. Louis. Professor Angell, as befitted one of his authority and attainments, had had a prominent part in all the deliberations; and was one of the first to be approached by the several outsiders who took advantage of the convocation to offer questions for correct answering and problems for expert solution.</p>

<p>The chief of these outsiders, and in a short time the focus of interest for the entire meeting, was a commonplace-looking middle-aged man who had travelled all the way from New Orleans for certain special information unobtainable from any local source. His name was John Raymond Legrasse, and he was by profession an Inspector of Police. With him he bore the subject of his visit, a grotesque, repulsive, and apparently very ancient stone statuette whose origin he was at a loss to determine.</p>

Headings

Preface with #, ##, ###, ####, #####, or #####

OR

Underline with = for Heading 1

Underline with – for Heading 2

H.P. Lovecraft's "The Call of Cthulhu"

(Found Among the Papers of the Late Francis Wayland Thurston, of Boston)

I. The Horror in Clay.

The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents.

II. The Tale of Inspector Legrasse.

The older matters which had made the sculptor's dream and bas-relief so significant to my uncle formed the subject of the second half of his long manuscript.

III. The Madness from the Sea.

If heaven ever wishes to grant me a boon, it will be a total effacing of the results of a mere chance which fixed my eye on a certain stray piece of shelf-paper.

H.P. Lovecraft's "The Call of Cthulhu"

(Found Among the Papers of the Late Francis Wayland Thurston, of Boston)

I. The Horror in Clay.

The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents.

II. The Tale of Inspector Legrasse.

The older matters which had made the sculptor's dream and bas-relief so significant to my uncle formed the subject of the second half of his long manuscript.

III. The Madness from the Sea.

If heaven ever wishes to grant me a boon, it will be a total effacing of the results of a mere chance which fixed my eye on a certain stray piece of shelf-paper.

<h1>H.P. Lovecraft's "The Call of Cthulhu"</h1>

<p>(Found Among the Papers of the Late Francis Wayland Thurston, of Boston)</p>

<h2>I. The Horror in Clay.</h2>

<p>The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents.</p>

<h2>II. The Tale of Inspector Legrasse.</h2>

<p>The older matters which had made the sculptor's dream and bas-relief so significant to my uncle formed the subject of the second half of his long manuscript.</p>

<h2>III. The Madness from the Sea.</h2>

<p>If heaven ever wishes to grant me a boon, it will be a total effacing of the results of a mere chance which fixed my eye on a certain stray piece of shelf-paper.</p>

Bulleted Lists

Begin line with * or + or –

Indent 4 spaces or 1 tab for sub-lists

These are some of the best H.P. Lovecraft stories.

- * Cthulhu
 - * The Call of Cthulhu
 - * The Dunwich Horror
 - * The Whisperer in Darkness
- * Yog-Sothoth
 - * The Case of Charles Dexter Ward
- * Shoggoths
 - * The Shadow over Innsmouth
 - * At the Mountains of Madness
- * Nyarlathotep
 - * The Dreams in the Witch House
 - * The Rats in the Walls

These are some of the best H.P. Lovecraft stories.

- Cthulhu
 - The Call of Cthulhu
 - The Dunwich Horror
 - The Whisperer in Darkness
- Yog-Sothoth
 - The Case of Charles Dexter Ward
- Shoggoths
 - The Shadow over Innsmouth
 - At the Mountains of Madness
- Nyarlathotep
 - The Dreams in the Witch House
 - The Rats in the Walls

<p>These are some of the best H.P. Lovecraft stories.</p>

Cthulhu

The Call of Cthulhu

The Dunwich Horror

The Whisperer in Darkness

Yog-Sothoth

The Case of Charles Dexter Ward

Shoggoths

The Shadow over Innsmouth

At the Mountains of Madness

Nyarlathotep

The Dreams in the Witch House

The Rats in the Walls



MacDown

macdown.uranusjr.com

Free (& open source!)

Full-featured & actively developed



MacDown

![MacDown logo](http://macdown.uranusjr.com/static/base/img/logo-160.png)

Hello there! I'm **MacDown**, the open source Markdown editor for OS X.

Let me introduce myself.

Markdown and I

Markdown is a plain text formatting syntax created by John Gruber, aiming to provide a easy-to-read and feasible markup. The original Markdown syntax specification can be found [\[here\]\(http://daringfireball.net/projects/markdown/syntax\)](http://daringfireball.net/projects/markdown/syntax).

MacDown is created as a simple-to-use editor for Markdown documents. I render your Markdown contents real-time into HTML, and display them in a preview panel.

![MacDown Screenshot](http://d.pr/i/10UGP+)

I support all the original Markdown syntaxes. But I can do so much more! Various popular but non-standard syntaxes can be turned on/off from the [\[MacDown preference pane\]\(#markdown-pane\)](#).

MacDown



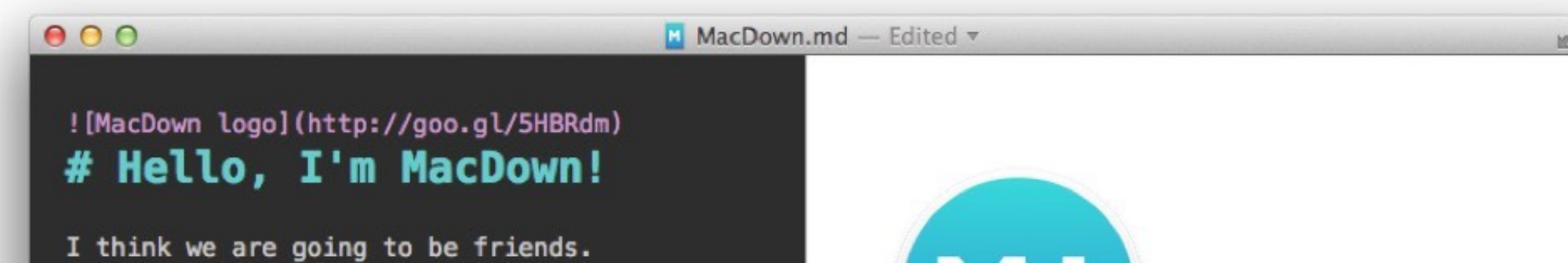
Hello there! I'm **MacDown**, the open source Markdown editor for OS X.

Let me introduce myself.

Markdown and I

Markdown is a plain text formatting syntax created by John Gruber, aiming to provide a easy-to-read and feasible markup. The original Markdown syntax specification can be found [here](#).

MacDown is created as a simple-to-use editor for Markdown documents. I render your Markdown contents real-time into HTML, and display them in a preview panel.

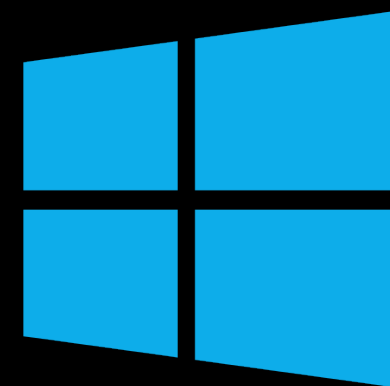


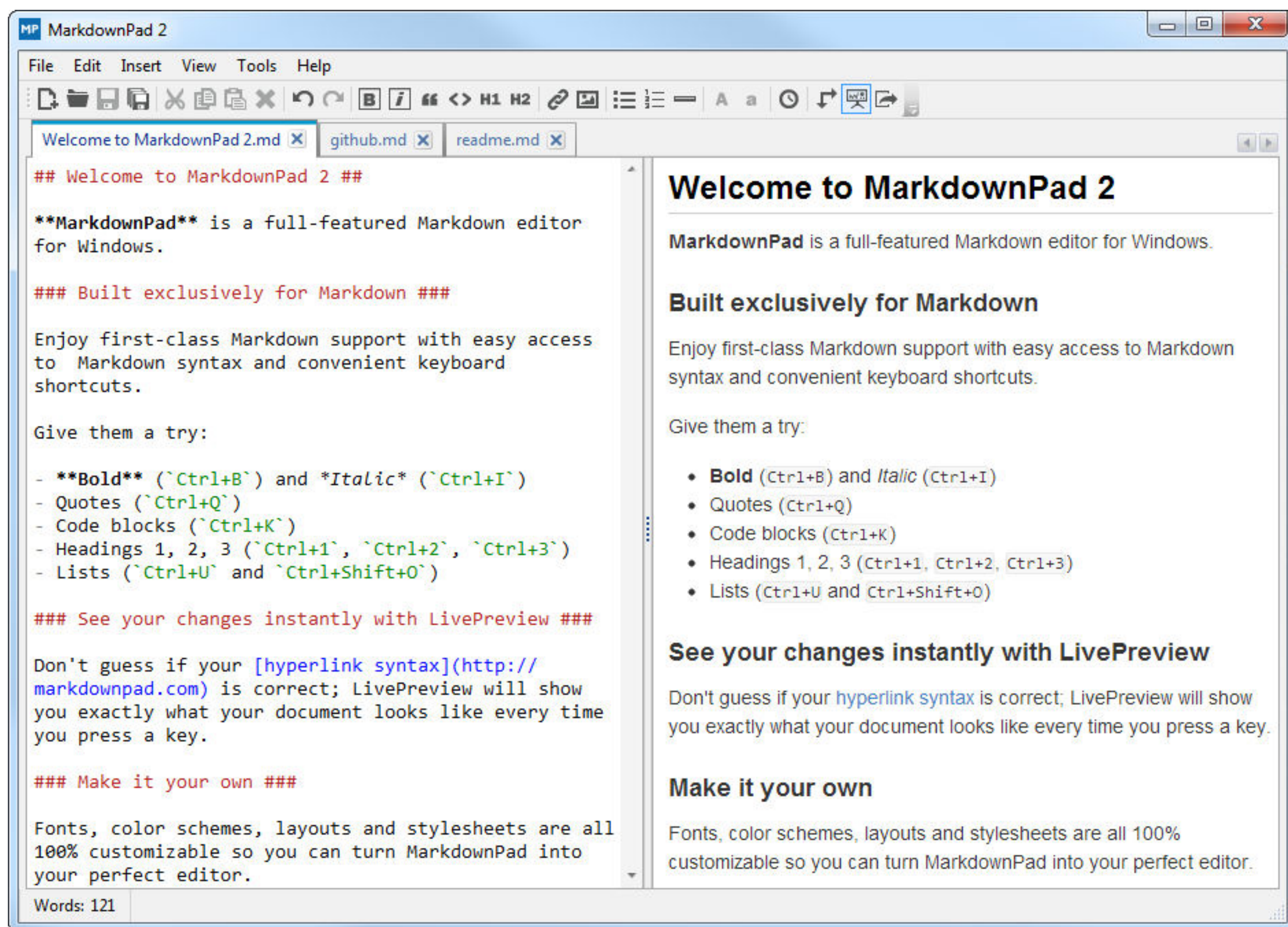
MP

MarkdownPad

markdownpad.com

Free, or \$15 gets you no usage restrictions, support for GitHub Flavored Markdown & Markdown Extra, & many other features





Welcome to MarkdownPad 2

MarkdownPad is a full-featured Markdown editor for Windows.

Built exclusively for Markdown

Enjoy first-class Markdown support with easy access to Markdown syntax and convenient keyboard shortcuts.

Give them a try:

- **Bold** (`Ctrl+B`) and *Italic* (`Ctrl+I`)
- Quotes (`Ctrl+Q`)
- Code blocks (`Ctrl+K`)
- Headings 1, 2, 3 (`Ctrl+1`, `Ctrl+2`, `Ctrl+3`)
- Lists (`Ctrl+U` and `Ctrl+Shift+O`)

See your changes instantly with LivePreview

Don't guess if your [\[hyperlink syntax\]\(http://markdownpad.com\)](http://markdownpad.com) is correct; LivePreview will show you exactly what your document looks like every time you press a key.

Make it your own

Fonts, color schemes, layouts and stylesheets are all 100% customizable so you can turn MarkdownPad into your perfect editor.

Words: 121

Welcome to MarkdownPad 2

MarkdownPad is a full-featured Markdown editor for Windows.

Built exclusively for Markdown

Enjoy first-class Markdown support with easy access to Markdown syntax and convenient keyboard shortcuts.

Give them a try:

- **Bold** (`Ctrl+B`) and *Italic* (`Ctrl+I`)
- Quotes (`Ctrl+Q`)
- Code blocks (`Ctrl+K`)
- Headings 1, 2, 3 (`Ctrl+1`, `Ctrl+2`, `Ctrl+3`)
- Lists (`Ctrl+U` and `Ctrl+Shift+O`)

See your changes instantly with LivePreview

Don't guess if your [hyperlink syntax](http://markdownpad.com) is correct; LivePreview will show you exactly what your document looks like every time you press a key.

Make it your own

Fonts, color schemes, layouts and stylesheets are all 100% customizable so you can turn MarkdownPad into your perfect editor.

CSS





Getting started

An overview of Less, how to download and use, examples and more.

less v2.5.3 has been released - [See what's new](#)

[Getting Started](#)[Using Less](#)[Command-line with Rhino](#)[Client-side Usage](#)[Get Less.js](#)[License FAQs](#)

Getting Started

Less is a CSS pre-processor, meaning that it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow you to make CSS that is more maintainable, themable and extendable.

Less runs inside Node, in the browser and inside Rhino. There are also many 3rd party tools that allow you to compile your files and watch for changes. The quickest place for first experiments with less is our [online editor](#).

For example:

```
@base: #f938ab;  
  
.box-shadow(@style, @c) when (iscolor(@c)) {
```

Less

Created in 2009 & inspired by Sass

Provides variables, nesting, mixins, operators, & functions

Can convert Less code on-the-fly in the client browser
using `less.js`

Open source

lesscss.org

Your laptop needs more Sass. Grab a set of Sass stickers now.



[INSTALL](#)

[LEARN SASS](#)

[BLOG](#)

[DOCUMENTATION](#)

[GET INVOLVED](#)

[LIBSASS](#)

“Syntactically Awesome Style Sheets”

CSS with superpowers



Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.

[Current Release: Selective Steve \(3.4.21\)](#)

[Release Notes](#)

[Fork on Github](#)

[Implementation Guide](#)

CSS COMPATIBLE

FEATURE RICH

MATURE

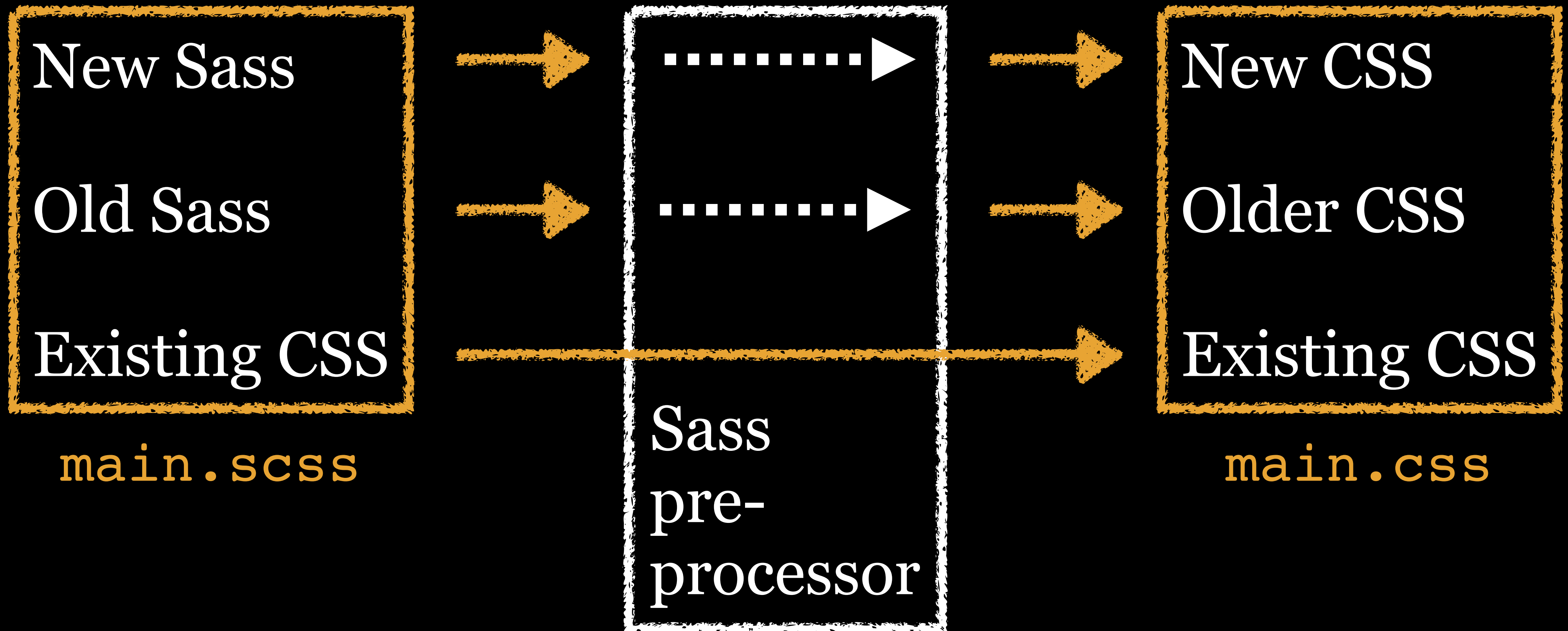
	2006	2010 (Sass 3)
Syntax	Indented	SCSS (“Sassy CSS”)
<code>.ending</code>	<code>.sass</code>	<code>.scss</code>
Uses <code>{ }</code> & <code>;</code>	No	Yes
Assignment	<code>=</code>	<code>:</code>
Mixins	<code>= & +</code>	<code>@mixin</code> & <code>@include</code>

Sass is a language with 2 different syntaxes (use SCSS)

The big difference between Sass & SCSS syntax: SCSS extends CSS, so every `.css` file is also a valid `.scss` file (but *not* the other way!)

So you've decided to start using Sass (SCSS) & you already have CSS...

1. Start with your CSS file; e.g., `main.css`
2. Change the extension to `.scss` so it's `main.scss`
3. Start converting some of the existing CSS in `main.scss` to use SCSS syntax
4. When `main.scss` is run through a preprocessor to create a new `main.css` file...
 1. any Sass is converted to CSS & inserted into `main.css`
 2. any CSS is just passed along into `main.css`



Bootstrap 3 used Less

Bootstrap 4 uses Sass (SCSS)

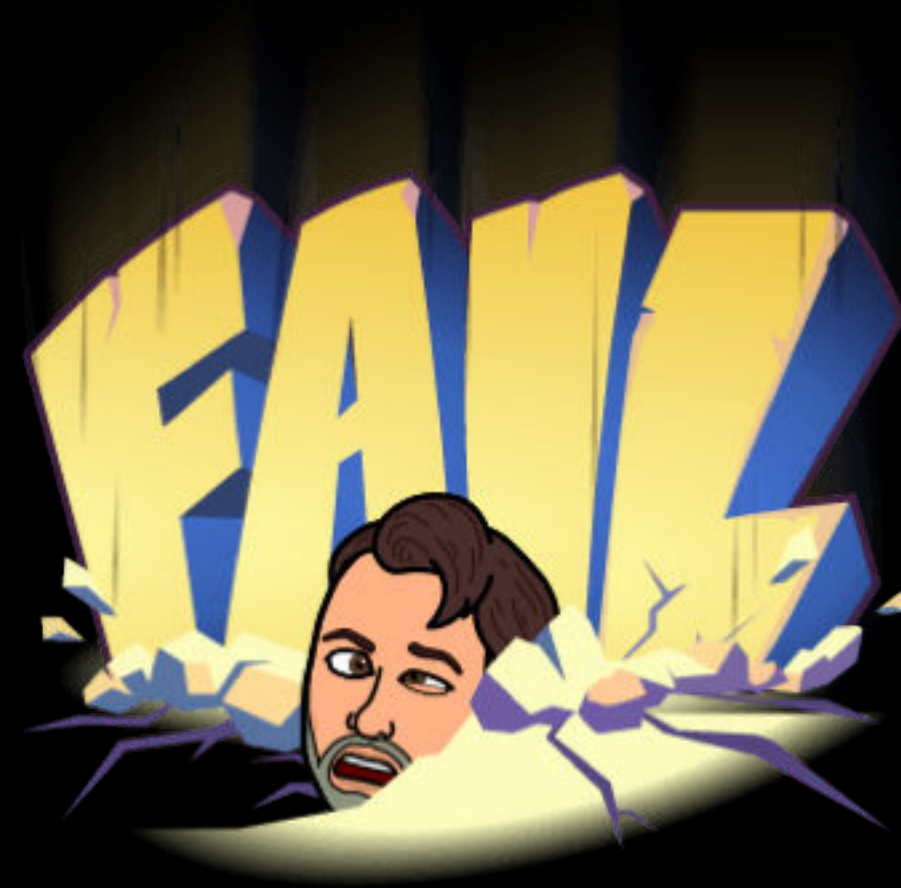
Sass seems to be “winning”

💡 PRO TIP

Never edit the CSS after it is generated by Sass!

💡 PRO TIP

Never edit the CSS after it is generated by Sass!



Sass Features

Variables & Functions

Nesting

Mixins

@import Directive

Comments

Variables

Variables allow you to set up common CSS styles

```
$accent-color: #4399FA;
```

```
$default-margin: 1em;
```

```
$sans-serif: "Source Sans Pro", Verdana, sans-  
serif;
```

```
$serif: "Georgia Pro", Georgia, serif;
```

⚙ HTML Tidy ✕

1 ▾ <h1>Dagon <small>by H.P.
Lovecraft</small></h1>
2 ▾ <p>
3 I am writing this under an
appreciable mental strain,
since by tonight I shall be no
more. Penniless, and at the end
of my supply of the drug which
alone makes life endurable, I
can bear the torture no longer;
and shall cast myself from this
garret window into the squalid
street below. Do not think from
my slavery to morphine that I
am a weakling or a degenerate.
When you have read these
hastily scrawled pages you may
guess, though never fully
realise, why it is that I must
have forgetfulness or death.
4 </p>
5 ▾ <p>
6
7

⚙ CSS (SCSS) View Compiled ✕

⚙ JS Tidy ✕

Dagon by H.P. Lovecraft

I am writing this under an appreciable mental strain, since by tonight I shall be no more. Penniless, and at the end of my supply of the drug which alone makes life endurable, I can bear the torture no longer; and shall cast myself from this garret window into the squalid street below. Do not think from my slavery to morphine that I am a weakling or a degenerate. When you have read these hastily scrawled pages you may guess, though never fully realise, why it is that I must have forgetfulness or death.

It was in one of the most open and least frequented parts of the broad Pacific that the packet of which I was supercargo fell a victim to the German sea-raider. The great war was then at its very beginning, and the ocean forces of the Hun had not completely sunk to their later degradation; so that our vessel was made a legitimate prize, whilst we of her crew were treated with all the fairness and consideration due us as naval prisoners.

H.P. Lovecraft • 10 Barnes Street • Providence, RI 02906

HTML

Tidy

5

<p>

6

It was in one of the most open and least frequented parts of the broad Pacific that the packet of which I was supercargo fell a victim to the German sea-raider. The great war was then at its very beginning, and the ocean forces of the Hun had not completely sunk to their later degradation; so that our vessel was made a legitimate prize, whilst we of her crew were treated with all the fairness and consideration due us as naval prisoners.

7

</p>

8

<footer>

9

H.P. Lovecraft • 10 Barnes Street • Providence, RI 02906

10

</footer>

CSS (SCSS)

View Compiled

JS

Tidy

Dagon by H.P. Lovecraft

I am writing this under an appreciable mental strain, since by tonight I shall be no more. Penniless, and at the end of my supply of the drug which alone makes life endurable, I can bear the torture no longer; and shall cast myself from this garret window into the squalid street below. Do not think from my slavery to morphine that I am a weakling or a degenerate. When you have read these hastily scrawled pages you may guess, though never fully realise, why it is that I must have forgetfulness or death.

It was in one of the most open and least frequented parts of the broad Pacific that the packet of which I was supercargo fell a victim to the German sea-raider. The great war was then at its very beginning, and the ocean forces of the Hun had not completely sunk to their later degradation; so that our vessel was made a legitimate prize, whilst we of her crew were treated with all the fairness and consideration due us as naval prisoners.

HTMLTidy×

CSS (SCSS)View Compiled×

```
1 $accent-color: #4399FA;
2
3 $default-padding: .5em;
4
5 $sans-serif: "Source Sans Pro",
  Verdana, sans-serif;
6
7 $serif: "Iowan Old Style",
  Georgia, serif;
8
9 html {
10   font-family: $serif;
11 }
12
13 h1 {
14   font-family: $sans-serif;
15 }
16
17 footer {
18   background-color: $accent-
    color;
19   padding: $default-padding;
20 }
```

JS Tidy×

Dagon by H.P. Lovecraft

I am writing this under an appreciable mental strain, since by tonight I shall be no more. Penniless, and at the end of my supply of the drug which alone makes life endurable, I can bear the torture no longer; and shall cast myself from this garret window into the squalid street below. Do not think from my slavery to morphine that I am a weakling or a degenerate. When you have read these hastily scrawled pages you may guess, though never fully realise, why it is that I must have forgetfulness or death.

It was in one of the most open and least frequented parts of the broad Pacific that the packet of which I was supercargo fell a victim to the German sea-raider. The great war was then at its very beginning, and the ocean forces of the Hun had not completely sunk to their later degradation; so that our vessel was made a legitimate prize, whilst we of her crew were treated with all the fairness and consideration due us as naval prisoners.

H.P. Lovecraft • 10 Barnes Street • Providence, RI 02906

HTML Tidy

CSS Compiled View Compiled

```
1 html {
2   font-family: "Iowan Old Style",
   Georgia, serif;
3 }
4
5 h1 {
6   font-family: "Source Sans Pro",
   Verdana, sans-serif;
7 }
8
9 footer {
10  background-color: #4399FA;
11  padding: 0.5em;
12 }
13
14 /* Uninteresting stuff below here
   */
15 html {
16  font-size: 24px;
17 }
18
19 footer {
```

JS Tidy

Dagon by H.P. Lovecraft

I am writing this under an appreciable mental strain, since by tonight I shall be no more. Penniless, and at the end of my supply of the drug which alone makes life endurable, I can bear the torture no longer; and shall cast myself from this garret window into the squalid street below. Do not think from my slavery to morphine that I am a weakling or a degenerate. When you have read these hastily scrawled pages you may guess, though never fully realise, why it is that I must have forgetfulness or death.

It was in one of the most open and least frequented parts of the broad Pacific that the packet of which I was supercargo fell a victim to the German sea-raider. The great war was then at its very beginning, and the ocean forces of the Hun had not completely sunk to their later degradation; so that our vessel was made a legitimate prize, whilst we of her crew were treated with all the fairness and consideration due us as naval prisoners.

H.P. Lovecraft • 10 Barnes Street • Providence, RI 02906

The Pragmatic Programmer



from journeyman
to master

Andrew Hunt
David Thomas

Foreword by Ward Cunningham

DRY: *Don't Repeat Yourself*

“Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.”


```
#toc {  
  background-color: hsl(10,  
72%, 53%);  
}  
  
.answer {  
  background-color: hsl(10,  
72%, 53%);  
}  
  
footer {  
  background-color: hsl(10,  
72%, 53%);  
}
```

CSS: *not* DRY

```
$main-bg-color: hsl(10, 72%,  
53%);  
  
#toc {  
  background-color: $main-bg-  
color;  
}  
  
.answer {  
  background-color: $main-bg-  
color;  
}  
  
footer {  
  background-color: $main-bg-  
color;  
}
```

Sass: DRY

Easier to recognize when looking through code

Which is simpler to understand: `hsl(33, 100%, 88.4%)`
or `$main-bg-color`?

Scope

Variables declared at the top of your Sass are *global*, so they can be accessed anywhere

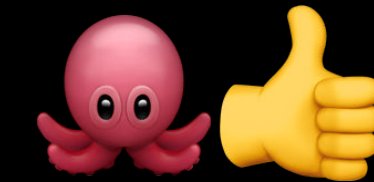
```
$main-bg-color: dodgerblue;
```

```
.cthulhu-baseball {  
  background-color: $main-bg-color;  
}
```

```
blockquote {  
  background-color: $main-bg-color;  
}
```

You can also create *local variables* that are *scoped* within a particular block of Sass by declaring the variable inside curly braces, so they can only be accessed within that block

```
.cthulhu-baseball {  
  $bg-color: dodgerblue;  
  background-color: $bg-color;  
}
```



```
.yog-sothoth-baseball {  
  background-color: $bg-color;  
}
```



(this will not work,
& will get you eaten)

Because they are separate, it's possible to declare a local variable with the same name as a global variable

The 2 are separate from each other, which ensures that someone writing a local variable doesn't accidentally stomp on a global variable they didn't know about

You can overwrite a global variable from within a local scope if you use the `!global` flag — & it will carry forward with the new value

This might help you with mixins, for instance

```
$main-bg-color: dodgerblue;
```

```
.cthulhu-baseball {  
  $main-bg-color: seagreen !global;  
  background-color: $main-bg-color;  
}
```

Resolves to `seagreen`;
Cthulhu 🐙 is happy

```
.yog-sothoth-baseball {  
  background-color: $main-bg-color;  
}
```

Also resolves to `seagreen`;
Yog-Sothoth expected `dodgerblue`;
Yog-Sothoth 🌌 eats you

Math Operators

+ Addition

– Subtraction

* Multiplication

/ Division

You can use math operations along with variables

```
$default-margin: 1em;
```

```
$default-border: $default-margin/2;
```

```
footer {  
    margin-top: $default-margin;  
    border-top: $default-border dotted gray;  
}
```



```
$default-margin: 1em;
```

```
h1, h2, h3, h4, h5, h6 {  
    margin: $default-margin*1.5 $default-margin;  
}
```

<code><absolute-size></code>	Scaling Factor	HTML <code>h_x</code>
<code>xx-large</code>	2/1	<code><h1></code>
<code>x-large</code>	3/2	<code><h2></code>
<code>large</code>	6/5	<code><h3></code>
<code>medium</code>	1	<code><h4></code>
<code>small</code>	8/9	<code><h5></code>
<code>x-small</code>	3/4	
<code>xx-small</code>	3/5	<code><h6></code>

HTML

Tidy

```
1 <pre class="code">
2   html {font-size: 24px;}   h2 1em = 24*1.5 =
   36px
3   h1 1em = 24*2 = 48px     h3 1em = 24*1.2 =
   28.8px
4 </pre>
5 <h1>Testola (48*1.5=72px)</h1>
6 <hr>
7 <h2>Testola (36*1.5=54px)</h2>
8 <hr>
9 <h3>Testola (28*1.5=42px)</h3>
```

CSS (SCSS)

View Compiled

```
1 $default-margins: 1em;
2
3 h1, h2, h3 {
4   margin: ($default-margins * 1.5) $default-
   margins;
5 }
6
7 h1, h2, h3 {
8   background-color: orange;
9 }
```

JS

Tidy

html {font-size: 24px;} h2 1em = 24*1.5 = 36px
h1 1em = 24*2 = 48px h3 1em = 24*1.2 = 28.8px

Testola (48*1.5=72px)

640.0 x 72.0 pt

Testola (36*1.5=54px)

Testola (28*1.5=42px)

HTML

Tidy

```
1 <pre class="code">
2   html {font-size: 24px;}   h2 1em = 24*1.5 =
   36px
3   h1 1em = 24*2 = 48px     h3 1em = 24*1.2 =
   28.8px
4 </pre>
5 <h1>Testola (48*1.5=72px)</h1>
6 <hr>
7 <h2>Testola (36*1.5=54px)</h2>
8 <hr>
9 <h3>Testola (28*1.5=42px)</h3>
```

CSS (SCSS)

View Compiled

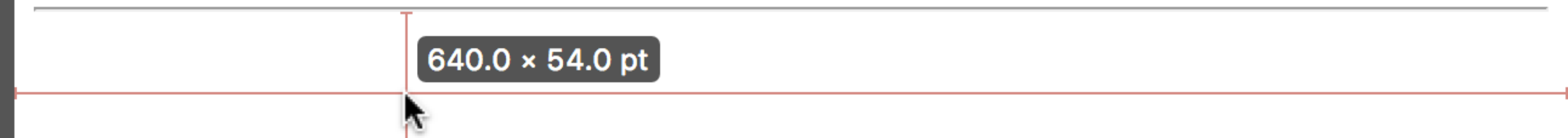
```
1 $default-margins: 1em;
2
3 h1, h2, h3 {
4   margin: ($default-margins * 1.5) $default-
   margins;
5 }
6
7 h1, h2, h3 {
8   background-color: orange;
9 }
```

JS

Tidy

html {font-size: 24px;} h2 1em = 24*1.5 = 36px
h1 1em = 24*2 = 48px h3 1em = 24*1.2 = 28.8px

Testola (48*1.5=72px)



Testola (36*1.5=54px)

Testola (28*1.5=42px)

HTML

Tidy

```
1 <pre class="code">
2   html {font-size: 24px;}   h2 1em = 24*1.5 =
   36px
3   h1 1em = 24*2 = 48px     h3 1em = 24*1.2 =
   28.8px
4 </pre>
5 <h1>Testola (48*1.5=72px)</h1>
6 <hr>
7 <h2>Testola (36*1.5=54px)</h2>
8 <hr>
9 <h3>Testola (28*1.5=42px)</h3>
```

CSS (SCSS)

View Compiled

```
1 $default-margins: 1em;
2
3 h1, h2, h3 {
4   margin: ($default-margins * 1.5) $default-
   margins;
5 }
6
7 h1, h2, h3 {
8   background-color: orange;
9 }
```

JS

Tidy

html {font-size: 24px;} h2 1em = 24*1.5 = 36px
h1 1em = 24*2 = 48px h3 1em = 24*1.2 = 28.8px

Testola (48*1.5=72px)

Testola (36*1.5=54px)

640.0 x 42.5 pt

Testola (28*1.5=42px)

HTML

Tidy



```
1 <pre class="code">
2   html {font-size: 24px;}   h2 1em = 24*1.5 =
   36px
3   h1 1em = 24*2 = 48px     h3 1em = 24*1.2 =
   28px
4 </pre>
5 <h1>Testola (48*1.5=72px)</h1>
6 <hr>
7 <h2>Testola (36*1.5=54px)</h2>
8 <hr>
9 <h3>Testola (28*1.5=42px)</h3>
```

CSS Compiled

View Compiled



```
1 h1, h2, h3 {
2   margin: 1.5em 1em;
3 }
4
5 h1, h2, h3 {
6   background-color: orange;
7 }
8
9 /* Uninteresting stuff below here */
10 html {
11   font-size: 24px;
```

JS

Tidy



html {font-size: 24px;} h2 1em = 24*1.5 = 36px
h1 1em = 24*2 = 48px h3 1em = 24*1.2 = 28px

Testola (48*1.5=72px)

Testola (36*1.5=54px)

Testola (28*1.5=42px)

Darken & Lighten Functions

Variables can use *darken* & *lighten* functions which allow you to adjust colors in a single palette

```
darken(<color>, <percentage>)  
lighten(<color>, <percentage>)
```

Do not use these functions — the colors aren't accurate

Nesting

1. Simple selectors

- » Universal
- » Type
- » Class
- » ID
- » Pseudo-classes
- » Pseudo-elements
- » Attribute

2. Compound selectors

3. Complex selectors (using combinators)

- » Descendant
- » Child
- » Adjacent sibling
- » General sibling

4. Selector list

Nesting provides very useful features

- » Create complex selectors using combinators; e.g., `blockquote > h2`
- » Create compound selectors (which glue together simple selectors); e.g., `p.intro`
- » Use `&` as a parent selector
- » Nested properties

Nesting lets you avoid repeating selectors by literally nesting them inside other declarations

Complex Selectors with Combinators

Symbol	Name	Ex.	Selected B
(space)	Descendant	$A \ B$	Any descendant of A
$>$	Child	$A > B$	Direct children of A
$+$	Adjacent Sibling	$A + B$	Next sibling after A
\sim (tilde)	General Sibling	$A \sim B$	All siblings after A

A combinator combines 2 selectors together into 1

Sass makes it easy to create descendant combinators

HTML

Tidy

1

<section>

2

<header>

3

<h1>Dagon <small>by H.P. Lovecraft</small></h1>

4

</header>

5

<p>

6

I am writing this under an appreciable mental strain, since by tonight I shall be no more. Penniless, and at the end of my supply of the drug which alone makes life endurable, I can bear the torture no longer; and shall cast myself from this garret window into the squalid street below.

7

</p>

8

<blockquote>

9

I am writing this under an appreciable mental strain, since by tonight I shall be no more.

10

</blockquote>

Dagon by H.P. Lovecraft

I am writing this under an appreciable mental strain, since by tonight I shall be no more. Penniless, and at the end of my supply of the drug which alone makes life endurable, I can bear the torture no longer; and shall cast myself from this garret window into the squalid street below.

I am writing this under an appreciable mental strain, since by tonight I shall be no more.


Do not think from my slavery to morphine that I am a weakling or a degenerate. When you have read these hastily scrawled pages you may guess, though never fully realise, why it is that I must have forgetfulness or death.

Dagon by H.P. Lovecraft

I am writing this under an appreciable mental strain, since by tonight I shall be no more.

Do not think from my slavery to morphine that I am a weakling or a degenerate. When you have read these hastily scrawled pages you may guess, though never fully realise, why it is that I must have forgetfulness or death.

 JS

Tidy


HTML Tidy X

CSS (SCSS) View Compiled X

```
1 section {
2   header {
3     color: white;
4     background: gray;
5     padding: .25em .4em;
6     margin: -.4em;
7   }
8   blockquote {
9     border-left: 8px solid gray;
10    padding-left: .5em;
11    font-style: italic;
12  }
13 }
14
15 /* Uninteresting stuff below here */
16
17 html {
18   font-size: 24px;
19   font-family: "Iowan Old Style",
20     Georgia, serif;
21 }
```

JS Tidy X

Dagon by H.P. Lovecraft

I am writing this under an appreciable mental strain, since by tonight I shall be no more. Penniless, and at the end of my supply of the drug which alone makes life endurable, I can bear the torture no longer; and shall cast myself from this garret window into the squalid street below.

I am writing this under an appreciable mental strain, since by tonight I shall be no more.

Do not think from my slavery to morphine that I am a weakling or a degenerate. When you have read these hastily scrawled pages you may guess, though never fully realise, why it is that I must have forgetfulness or death.

HTML Tidy

CSS Compiled View Compiled

```
1 section header {
2   color: white;
3   background: gray;
4   padding: .25em .4em;
5   margin: -.4em;
6 }
7 section blockquote {
8   border-left: 8px solid gray;
9   padding-left: .5em;
10  font-style: italic;
11 }
12
13 /* Uninteresting stuff below here */
14 html {
15   font-size: 24px;
16   font-family: "Iowan Old Style",
17   Georgia, serif;
18 }
19 h1 {
20   margin-top: 0;
21   margin-bottom: 0;
```

JS Tidy

Dagon by H.P. Lovecraft

I am writing this under an appreciable mental strain, since by tonight I shall be no more. Penniless, and at the end of my supply of the drug which alone makes life endurable, I can bear the torture no longer; and shall cast myself from this garret window into the squalid street below.

I am writing this under an appreciable mental strain, since by tonight I shall be no more.

Do not think from my slavery to morphine that I am a weakling or a degenerate. When you have read these hastily scrawled pages you may guess, though never fully realise, why it is that I must have forgetfulness or death.

Benefits

- » `section` selector only has to be written once
- » Immediately clear visually that both the `header` and `blockquote` styles are applied at the same level
- » Nested Sass can match HTML structure, which makes understanding how the two work together much easier
- » Much better clarity & efficiency

So remember, when you nest like this

```
footer {  
  p {  
    font-family: $serif-font  
  }  
}
```

`footer` & `p` are joined with a space, the descendant combinator

```
footer p {  
  font-family: Georgia, serif;  
}
```

You can nest with other combinators too

HTML

1

<table class="movies">

2

<thead>

3

<tr>

4

<th>Movie</th>

5

<th>Genre</th>

6

<th>Hero</th>

7

<th>Year</th>

8

</tr>

9

</thead>

10

<tbody>

11

<tr>

12

<td>Die Hard</td>

13

<td>Action</td>

14

<td>John McClane</td>

15

<td>1988</td>

16

</tr>

17

<tr>

18

<td>Star Wars</td>

19

<td>Sci-Fi</td>

20

<td>Luke Skywalker</td>

21

<td>1977</td>

22

</tr>

23

<tr>

24

<td>John Wick</td>

25

<td>Action</td>

CSS (SCSS)

JS

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

HTML

1

<table class="movies">

2

<thead>

3

<tr>

4

<th>Movie</th>

5

<th>Genre</th>

6

<th>Hero</th>

7

<th>Year</th>

8

</tr>

9

</thead>

10

<tbody>

11

<tr>

12

<td>Die Hard</td>

13

<td>Action</td>

14

<td>John McClane</td>

15

<td>1988</td>

16

</tr>

17

<tr>

18

<td>Star Wars</td>

CSS (SCSS)

1

table.movies {

2

> tbody {

3

> tr:hover {

4

background-color: pink;

5

}

6

}

7

}

8

}

JS

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

HTML

1<table class="movies">

2<thead>

3<tr>

4<th>Movie</th>

5<th>Genre</th>

6<th>Hero</th>

7<th>Year</th>

8</tr>

9</thead>

10<tbody>

11<tr>

12<td>Die Hard</td>

13<td>Action</td>

14<td>John McClane</td>

15<td>1988</td>

16</tr>

17<tr>

18<td>Star Wars</td>

CSS (SCSS)Compiled

1table.movies > tbody > tr:hover {

2background-color: pink;

3}

4

5/* Uninteresting stuff below here */

6th {

7text-align: left;

8padding: 25px 40px;

JS

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

HTML

Tidy

```
1 <h2>The Call of Cthulhu <small>by
  H.P. Lovecraft</small></h2>
2 <p>
3   The most merciful thing in the
    world, I think, is the inability
    of the human mind to correlate all
    its contents. We live on a placid
    island of ignorance in the midst
    of black seas of infinity, and it
    was not meant that we should
    voyage far. The sciences,
```

CSS (SCSS)

View Compiled

```
1
2 h2 {
3   small {
4     display: block;
5   }
6 + p::first-line {
7   font-weight: 700;
8 }
9 }
10
11
```

JS

Tidy

The Call of Cthulhu

by H.P. Lovecraft

The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents. We live on a placid island of ignorance in the midst of black seas of infinity, and it was not meant that we should voyage far. The sciences, each straining in its own direction, have hitherto harmed us little; but some day the piecing together of dissociated knowledge will open up such terrifying vistas of reality, and of our frightful position therein, that we shall either go mad from the revelation or flee from the deadly light into the peace and safety of a new dark age.

Theosophists have guessed at the awesome grandeur of the cosmic cycle wherein our world and human race form transient incidents. They have hinted at strange survivals in terms which would freeze the blood if not masked by a bland optimism. But it is not from them that there came the single glimpse of forbidden aeons which chills me when I think of it and maddens me when I dream of it. That

HTML

Tidy

1

<h2>The Call of Cthulhu <small>by

2

H.P. Lovecraft</small></h2>

3

<p>

4

The most merciful thing in the

5

world, I think, is the inability

6

of the human mind to correlate all

7

its contents. We live on a placid

8

island of ignorance in the midst

9

of black seas of infinity, and it

10

was not meant that we should

11

voyage far. The sciences,

CSS

Compiled

View Compiled

1

h2 small {

2

display: block;

3

}

4

h2 + p::first-line {

5

font-weight: 700;

6

}

7

8

/* Uninteresting stuff below here

9

*/

10

html {

11

font-size: 24px;

JS

Tidy

The Call of Cthulhu

by H.P. Lovecraft

The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents. We live on a placid island of ignorance in the midst of black seas of infinity, and it was not meant that we should voyage far. The sciences, each straining in its own direction, have hitherto harmed us little; but some day the piecing together of dissociated knowledge will open up such terrifying vistas of reality, and of our frightful position therein, that we shall either go mad from the revelation or flee from the deadly light into the peace and safety of a new dark age.

Theosophists have guessed at the awesome grandeur of the cosmic cycle wherein our world and human race form transient incidents. They have hinted at strange survivals in terms which would freeze the blood if not masked by a bland optimism. But it is not from them that there came the single glimpse of forbidden aeons which chills me when I think of it and maddens me when I dream of it. That

Anytime you have this structure in Sass

```
selectorA {  
  [combinator] selectorB {  
    property: value;  
  }  
}
```

It's going to compile in CSS as

```
selectorA [combinator] selectorB {  
  property: value;  
}
```


What if you have declarations just for selectorA as well as declarations for descendants of selectorA?

HTML

Tidy

1

<h2>The Terrible Old Man <small>[TOC]</small></h2>

2

3

4

5

6

7

8

9

10

11

CSS (SCSS)

View Compiled

1

h2 {

2

color: #2f2f2f;

3

small {

4

font-size: .6em;

5

padding-left: 1em;

6

a {

7

text-decoration: none;

8

}

9

}

10

}

11

JS

Tidy

The Terrible Old Man [\[TOC\]](#)

It was the design of Angelo Ricci and Joe Czanek and Manuel Silva to call on the Terrible Old Man. This old man dwells all alone in a very ancient house on Water Street near the sea, and is reputed to be both exceedingly rich and exceedingly feeble; which forms a situation very attractive to men of the profession of Messrs. Ricci, Czanek, and Silva, for that profession was nothing less dignified than robbery.

The inhabitants of Kingsport say and think many things about the Terrible Old Man which generally keep him safe from the attention of gentlemen like Mr. Ricci and his colleagues, despite the almost certain fact that he hides a fortune of indefinite magnitude somewhere about his musty and venerable abode. He is, in truth, a very strange person, believed to have been a captain of East India clipper ships in his day; so old that no one can remember when he

HTML

Tidy

1

<h2>The Terrible Old Man <small><a

2

href="#toc">[TOC]</small></h2>

3

<p>

4

It was the design of Angelo Ricci and Joe Czanek and Manuel Silva to call on the Terrible Old Man. This old man dwells all alone in a very ancient house on Water Street near the sea, and is reputed to be both exceedingly rich and exceedingly

CSS

Compiled

View Compiled

1

h2 {

2

color: #2f2f2f;

3

}

4

h2 small {

5

font-size: .6em;

6

padding-left: 1em;

7

}

8

h2 small a {

9

text-decoration: none;

10

}

11

JS

Tidy

The Terrible Old Man [\[TOC\]](#)

It was the design of Angelo Ricci and Joe Czanek and Manuel Silva to call on the Terrible Old Man. This old man dwells all alone in a very ancient house on Water Street near the sea, and is reputed to be both exceedingly rich and exceedingly feeble; which forms a situation very attractive to men of the profession of Messrs. Ricci, Czanek, and Silva, for that profession was nothing less dignified than robbery.

The inhabitants of Kingsport say and think many things about the Terrible Old Man which generally keep him safe from the attention of gentlemen like Mr. Ricci and his colleagues, despite the almost certain fact that he hides a fortune of indefinite magnitude somewhere about his musty and venerable abode. He is, in truth, a very strange person, believed to have been a captain of East India clipper ships in his day; so old that no one can remember when he

© 1921 H.P. Lovecraft

HTML

Tidy

9

<footer>

10

© 1921 H.P.

Lovecraft

11

</footer>

CSS (SCSS)

View Compiled

12

footer {

13

background-color: #2f2f2f;

14

padding: 1em .5em;

15

margin: 1em -.5em -.5em -.5em;

16

color: white;

17

position: fixed;

18

bottom: 0;

19

width: 100%;

20

a {

21

text-decoration: none;

22

border-bottom: 1px dotted white;

23

&:visited {

24

color: white;

25

}

26

}

27

}

JS

Tidy

The Terrible Old Man [\[TOC\]](#)

It was the design of Angelo Ricci and Joe Czanek and Manuel Silva to call on the Terrible Old Man. This old man dwells all alone in a very ancient house on Water Street near the sea, and is reputed to be both exceedingly rich and exceedingly feeble; which forms a situation very attractive to men of the profession of Messrs. Ricci, Czanek, and Silva, for that profession was nothing less dignified than robbery.

The inhabitants of Kingsport say and think many things about the Terrible Old Man which generally keep him safe from the attention of gentlemen like Mr. Ricci and his colleagues, despite the almost certain fact that he hides a fortune of indefinite magnitude somewhere about his musty and venerable abode. He is, in truth, a very strange person, believed to have been a captain of East India clipper ships in his day; so old that no one can remember when he

was born, and he is so old that few know his real name.

HTML

Tidy

9

<footer>

10

© 1921 H.P.

Lovecraft

11

</footer>

CSS

Compiled

View Compiled

12

footer {

13

background-color: #2f2f2f;

14

padding: 1em .5em;

15

margin: 1em -.5em -.5em -.5em;

16

color: white;

17

position: fixed;

18

bottom: 0;

19

width: 100%;

20

}

21

footer a {

22

text-decoration: none;

23

border-bottom: 1px dotted white;

24

}

25

footer a:visited {

26

color: white;

27

}

JS

Tidy

The Terrible Old Man [\[TOC\]](#)

It was the design of Angelo Ricci and Joe Czanek and Manuel Silva to call on the Terrible Old Man. This old man dwells all alone in a very ancient house on Water Street near the sea, and is reputed to be both exceedingly rich and exceedingly feeble; which forms a situation very attractive to men of the profession of Messrs. Ricci, Czanek, and Silva, for that profession was nothing less dignified than robbery.

The inhabitants of Kingsport say and think many things about the Terrible Old Man which generally keep him safe from the attention of gentlemen like Mr. Ricci and his colleagues, despite the almost certain fact that he hides a fortune of indefinite magnitude somewhere about his musty and venerable abode. He is, in truth, a very strange person, believed to have been a captain of East India clipper ships in his day; so old that no one can remember when he

Compound Selectors

A compound selector consists of a chain of simple selectors (`*`, `p`, `.class`, `#id`, `:pseudo-class`, `::pseudo-element`, or `p[attribute]`) connected together without a combinator, e.g.:

- » `p.class`
- » `.intro::first-line`
- » `a[title^="Lovecraft"]`

& replicates the nested rule's parent

Most commonly used when you need to glue selectors together (pseudo-classes, pseudo-elements, attribute selectors) to create compound selectors; e.g., `p:foo` or `a[href*=cthulhu]`

& should never be followed by a space if you're building a compound selector, or you don't need it

.class	:hover	:only-child
#id	:lang()	:only-of-type
::after	:last-child	:root
::before	:last-of-type	:target
::first-letter	:link	:visited
::first-line	:not(s)	[foo]
:active	:nth-child(n)	[foo*="bar"]
:disabled	:nth-last-	[foo^="bar"]
:empty	child(n)	[foo="bar"]
:enabled	:nth-last-of-	[foo ="en"]
:first-child	type(n)	[foo~="bar"]
:first-of-type	:nth-of-	[foo\$="bar"]
:focus	type(n)	

HTML

1<table class="movies">

2<thead>

3<tr>

4<th>Movie</th>

5<th>Genre</th>

6<th>Hero</th>

7<th>Year</th>

8</tr>

9</thead>

10<tbody>

11<tr>

12<td>Die Hard</td>

13<td>Action</td>

14<td>John McClane</td>

15<td>1988</td>

16</tr>

17<tr>

18<td>Star Wars</td>

19<td>Sci-Fi</td>

20<td>Luke Skywalker</td>

21<td>1977</td>

22</tr>

23<tr>

24<td>John Wick</td>

25<td>Action</td>

CSS (SCSS)Compiled

JS

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

⚙ HTML ▾

```
1 <table class="movies">
2   <thead>
3     <tr>
4       <th>Movie</th>
5       <th>Genre</th>
6       <th>Hero</th>
7       <th>Year</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>Die Hard</td>
13      <td>Action</td>
14      <td>John McClane</td>
15      <td>1988</td>
16    </tr>
```

⚙ CSS (SCSS) ▾

```
1 table.movies {
2   tbody {
3     tr {
4       &:nth-child(odd) {
5         background-color: pink;
6       }
7     }
8   }
9 }
```

⚙ JS ▾

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

⚙ HTML ▾

```
1 <table class="movies">
2   <thead>
3     <tr>
4       <th>Movie</th>
5       <th>Genre</th>
6       <th>Hero</th>
7       <th>Year</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>Die Hard</td>
13      <td>Action</td>
14      <td>John McClane</td>
15      <td>1988</td>
16    </tr>
17    <tr>
18      <td>Star Wars</td>
19      <td>Sci-Fi</td>
20      <td>Luke Skywalker</td>
21      <td>1977</td>
22    </tr>
```

⚙ CSS (SCSS) Compiled ▾

```
1 table.movies tbody tr:nth-child(odd) {
2   background-color: pink;
3 }
```

⚙ JS ▾

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

```
<p class="center note">
```

```
.center {  
  text-align: center;  
}
```

Selects elements with the
`.center` class

```
.note {  
  font-size: .9em;  
}
```

Selects elements with the
`.note` class

```
.center.note {  
  font-style: italic;  
}
```

Selects elements with the
`.center` *and* `.note` classes

Sass makes it easy to select multiple classes

```
.foo {  
  background-color: green;  
  &.bar {  
    background-color: red;  
  }  
}
```

Sass



```
.foo {  
  background-color: green;  
}
```

CSS

```
.foo.bar {  
  background-color: red;  
}
```

& as Sass Parent Selector

HTML Tidy X

```
1 <table class="movies">
2   <thead>
3     <tr>
4       <th>Movie</th>
5       <th>Genre</th>
6       <th>Hero</th>
7       <th>Year</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>Die Hard</td>
```

CSS (SCSS) View Compiled X

```
1 table.movies {
2   tbody {
3     tr {
4       &:nth-child(odd) {
5         background-color: cornsilk;
6       }
7     }
8   }
9 }
10
```

JS Tidy X

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

& actually reproduces the *outer selector*

`tr` is outside the `&`, & `tbody` is outside `tr`, & `table.movies` is outside `tbody`, so that why `table.movies tbody tr` is the *outer selector*

So far we've used `&` to create compound (`p.foo`) selectors by putting it at the *beginning* of nested lines, e.g.:

```
p {  
  &::first-line {  
    font-weight: bold;  
  }  
}
```

But you can put the `&` in other places besides the beginning of lines, & then `&` becomes even more useful

When you use the & on a line, the outer selector is inserted wherever that & is

You might expect this:

```
table.movies {  
  > tbody {  
    > tr:hover {  
      color: red;  
    }  
  }  
}  
  
.sidebar & {  
  color: blue;  
}  
}
```

To produce this:

```
table.movies > tbody >  
tr:hover .sidebar  
table.movies > tbody >  
tr:hover
```

That is not what happens

Instead, this:

```
table.movies {  
  > tbody {  
    > tr:hover {  
      color: red;  
    }  
  }  
  
  .sidebar & {  
    color: blue;  
  }  
}
```

Actually produces this:

```
.sidebar table.movies  
> tbody > tr:hover
```

Why? Because the outer selector (`table.movies > tbody > tr:hover`) is inserted wherever the `&` is

```
1 <article>
2   <h1></h1>
5   <p></p>
8   <h2>
9     Yog-Sothoth
10  </h2>
11  <p></p>
14  <h2>
15    Nyarlathotep
16  </h2>
17  <p></p>
20 </article>
21
22 <aside>
23   1.2
24
```

```
1 h1, h2 {
2   font-family: sans-serif;
3
4   aside & {
5     color: green;
6   }
7 }
```

Cthulhu Mythos

The Cthulhu Mythos is a shared fictional universe, based on the work of American horror writer H. P. Lovecraft.

Yog-Sothoth

“Imagination called up the shocking form of fabulous Yog-Sothoth—only a congeries of iridescent globes, yet stupendous in its malign suggestiveness.”

Nyarlathotep

“And it was then that Nyarlathotep came

August Derleth

Best remembered
as the first
publisher of the
writings of H. P.
Lovecraft, and for
his own
contributions to
the Cthulhu
Mythos genre of
horror, as well as
his founding of the
publisher Arkham
House.


```
HTML
1 <article>
2   <h1></h1>
5   <p></p>
8   <h2>
9     Yog-Sothoth
10  </h2>
11  <p></p>
14  <h2>
15    Nyarlathotep
16  </h2>
17  <p></p>
20 </article>
21
22 <aside>
23  </aside>

CSS (SCSS) Compiled
1 h1, h2 {
2   font-family: sans-serif;
3 }
4 aside h1, aside h2 {
5   color: green;
6 }
7

JS
```

Cthulhu Mythos

The Cthulhu Mythos is a shared fictional universe, based on the work of American horror writer H. P. Lovecraft.

Yog-Sothoth

“Imagination called up the shocking form of fabulous Yog-Sothoth—only a congeries of iridescent globes, yet stupendous in its malign suggestiveness.”

Nyarlathotep

“And it was then that Nyarlathotep came

August Derleth

Best remembered as the first publisher of the writings of H. P. Lovecraft, and for his own contributions to the Cthulhu Mythos genre of horror, as well as his founding of the publisher Arkham House.

Nested Properties

Some CSS properties use the same base

`font-family`, `font-size`, & `font-weight` all start with `font-`

`padding-top`, `padding-right`, `padding-bottom`, & `padding-left` all start with `padding-`

Why type `font-` or `padding-` over & over?

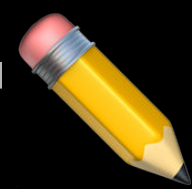
This is a lot of CSS:

```
font-size: 1.25em;  
line-height: 1.5;  
font-weight: 700;  
font-style: italic;  
font-variant: small-caps;  
font-family: "Georgia Pro", serif;
```

Using CSS *shorthand properties*, we can do this:

```
font: 700 italic small-caps 1.25em/1.5 "Georgia Pro",  
serif;
```

You can emulate CSS shorthand properties in Sass



SIDE NOTE

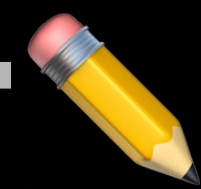
Other CSS shorthand properties:

```
padding: 1em 1.5em;
```

```
border: 2px dotted darkgray;
```

```
margin: 1em;
```

animation • background • border-bottom • border-color • border-left • border-radius • border-right • border-style • border-top • border-width • column-rule • columns • flex • flex-flow • grid • grid-area • grid-column • grid-row • grid-template • list-style • offset • outline • overflow • place-content • place-items • place-self • text-decoration • transition



SIDE NOTE

Worst. Shorthand. Ever.

```
background: url(http://placeholder.it/150) 20px  
20px / auto auto no-repeat scroll padding-box  
border-box indigo;
```

Order matters in several places!

And I left several values out!




```
HTML
1 <div class="attention">
2   <p>
3     Do not try to feed Cthulhu!
4   </p>
5 </div>
6 <h2>The Call of Cthulhu</h2>
7 <p>
8   The most merciful thing in the world, I
   think, is the inability of the human mind
   to correlate all its contents. We live on
   a placid island of ignorance in the midst
   of black seas of infinity, and it was not
   meant that we should voyage far. The

CSS (SCSS)
1 .attention {
2   color: red;
3   font: {
4     family: Verdana, sans-serif;
5     size: 1.5em;
6     weight: bold;
7   }
8 }
9

JS
```

Do not try to feed Cthulhu!

The Call of Cthulhu

The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents. We live on a placid island of ignorance in the midst of black seas of infinity, and it was not meant that we should voyage far. The sciences, each straining in its own direction, have hitherto harmed us little; but some day the piecing together of dissociated knowledge will open up such terrifying vistas of reality, and of our frightful position therein, that we shall either go mad from the revelation or flee from the deadly light into the peace and safety of a new dark age.

Note the **:** after **font**!

HTML

Tidy

1

<div class="attention">

2

<p>

3

Do not try to feed Cthulhu!

4

</p>

5

</div>

6

<h2>The Call of Cthulhu</h2>

7

<p>

8

The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents. We live on a placid island of ignorance in the midst of black seas of infinity, and it was not meant that we should voyage far. The

CSS

Compiled

View Compiled

1

.attention {

2

color: red;

3

font-family: Verdana, sans-serif;

4

font-size: 1.5em;

5

font-weight: bold;

6

}

7

8

/* Uninteresting stuff below here */

9

html {

JS

Tidy

Do not try to feed Cthulhu!

The Call of Cthulhu

The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents. We live on a placid island of ignorance in the midst of black seas of infinity, and it was not meant that we should voyage far. The sciences, each straining in its own direction, have hitherto harmed us little; but some day the piecing together of dissociated knowledge will open up such terrifying vistas of reality, and of our frightful position therein, that we shall either go mad from the revelation or flee from the deadly light into the peace and safety of a new dark age.

Warning!

Think before nesting more than 3 levels deep

```
.lovecraft {  
  > .cthulhu {  
    > li {  
      // Stop! (Maybe?)  
    }  
  }  
}
```


If you're deeper than 3 levels, you are creating a very specific selector

Very reliant on HTML structure, so fragile

Probably also going to be difficult to understand the generated CSS

Just think first

Mixins

Mixins allow you to write a set of declarations once & reuse them in any of your stylesheets

Should've been called *scoped includes*

When do you use a mixin?

When you keep repeating a chunk of code over & over again

Box Styles

Styles for `<div class="modal">`:

```
.modal {  
  background-color: HSL(0, 0%, 50%);  
  border: 4px solid black;  
  border-radius: 10px;  
  padding: .75em 1.25em;  
}
```

A `<button>` inside `<div class="modal">` will use the same styles as `.modal` *and* will also have its own styles too

Time for a mixin!

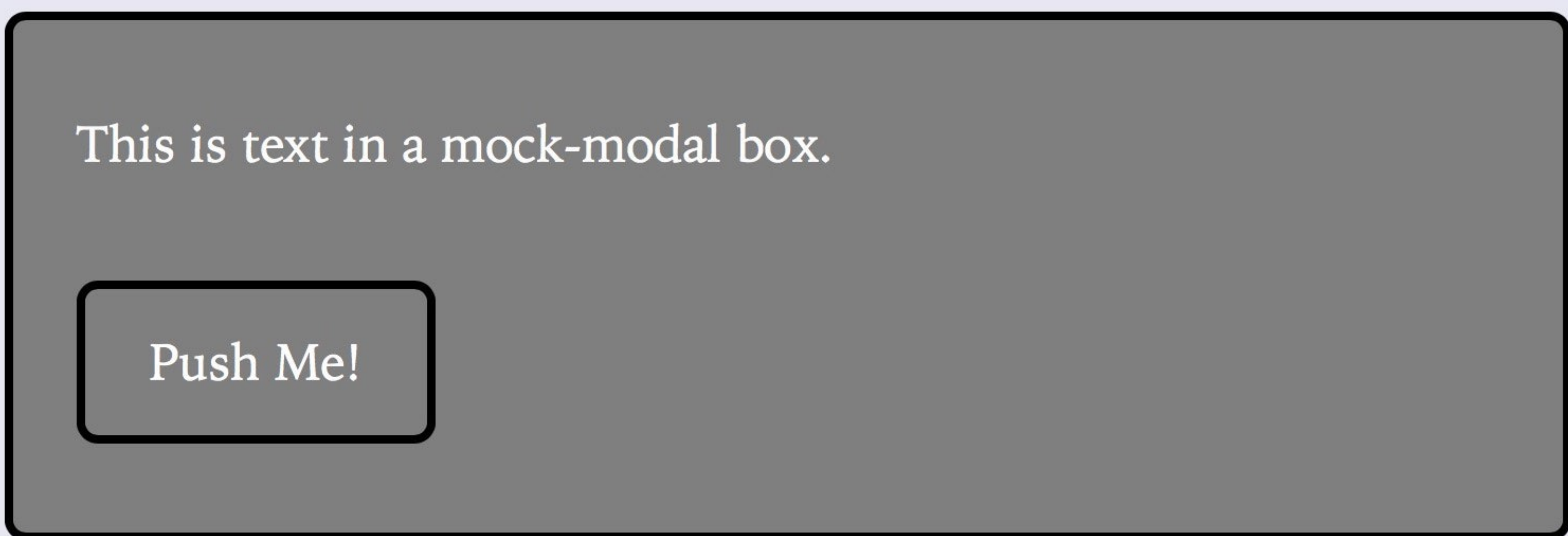
⚙ HTML

1 <div class="modal">
2 <p>
3 This is text in a mock-modal
box.
4 </p>
5 <button>
6 Push Me!
7 </button>
8 </div>

⚙ CSS (SCSS)

19
20 /* uninteresting stuff below here */
21
22 html {
23 font-size: 24px;
24 font-family: "Iowan Old Style",
Georgia, serif;
25 background-color: HSL(240, 22%,
92%);
26 }
27

⚙ JS



HTML

1

<div class="modal">

2

<p>

3

This is text in a mock-modal box.

CSS (SCSS)

1

@mixin box-style {

2

background-color: HSL(0, 0%, 50%);

3

border: 4px solid black;

4

border-radius: 10px;

5

padding: 0.75em 1.25em;

6

}

7

8

.modal {

9

@include box-style;

10

color: HSL(0, 0%, 99%);

11

> button {

12

@include box-style;

13

margin: 1em 0;

14

font-size: 1rem;

15

font-family: inherit;

16

color: inherit;

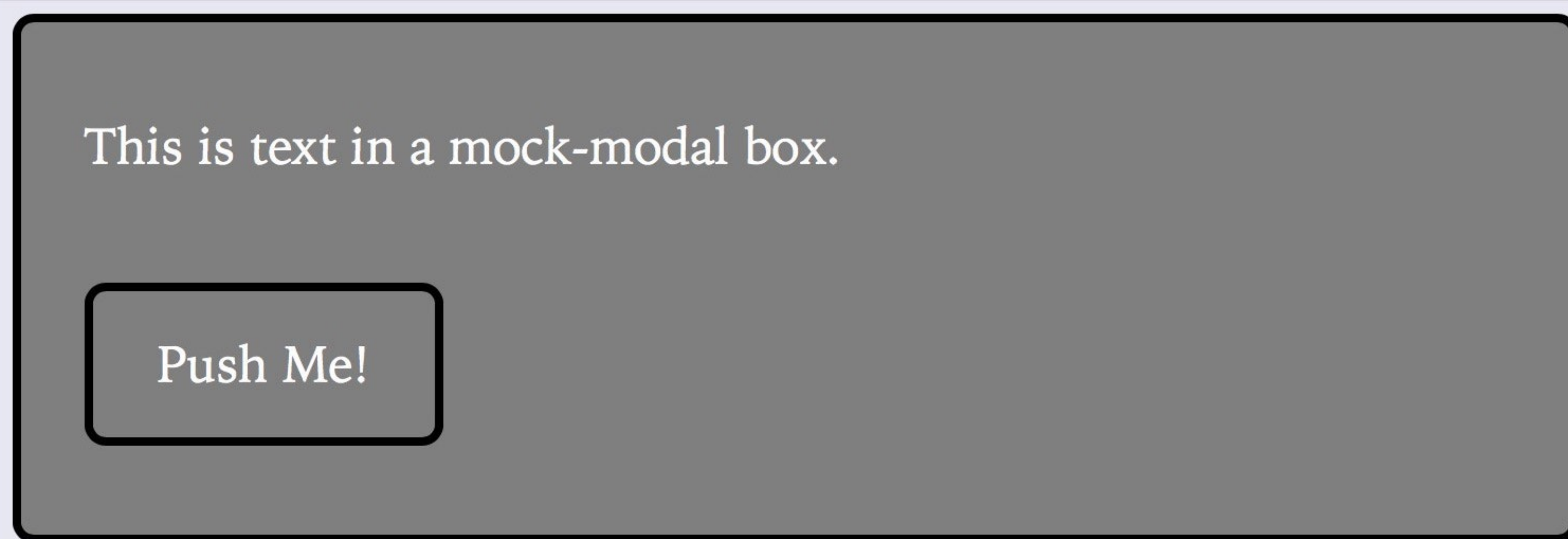
17

}

18

}

JS



HTML

1

<div class="modal">

2

<p>

3

This is text in a mock-modal box.

CSS

Compiled

1

.modal {

2

background-color: HSL(0, 0%, 50%);

3

border: 4px solid black;

4

border-radius: 10px;

5

padding: 0.75em 1.25em;

6

color: HSL(0, 0%, 99%);

7

}

8

.modal > button {

9

background-color: HSL(0, 0%, 50%);

10

border: 4px solid black;

11

border-radius: 10px;

12

padding: 0.75em 1.25em;

13

margin: 1em 0;

14

font-size: 1rem;

15

font-family: inherit;

16

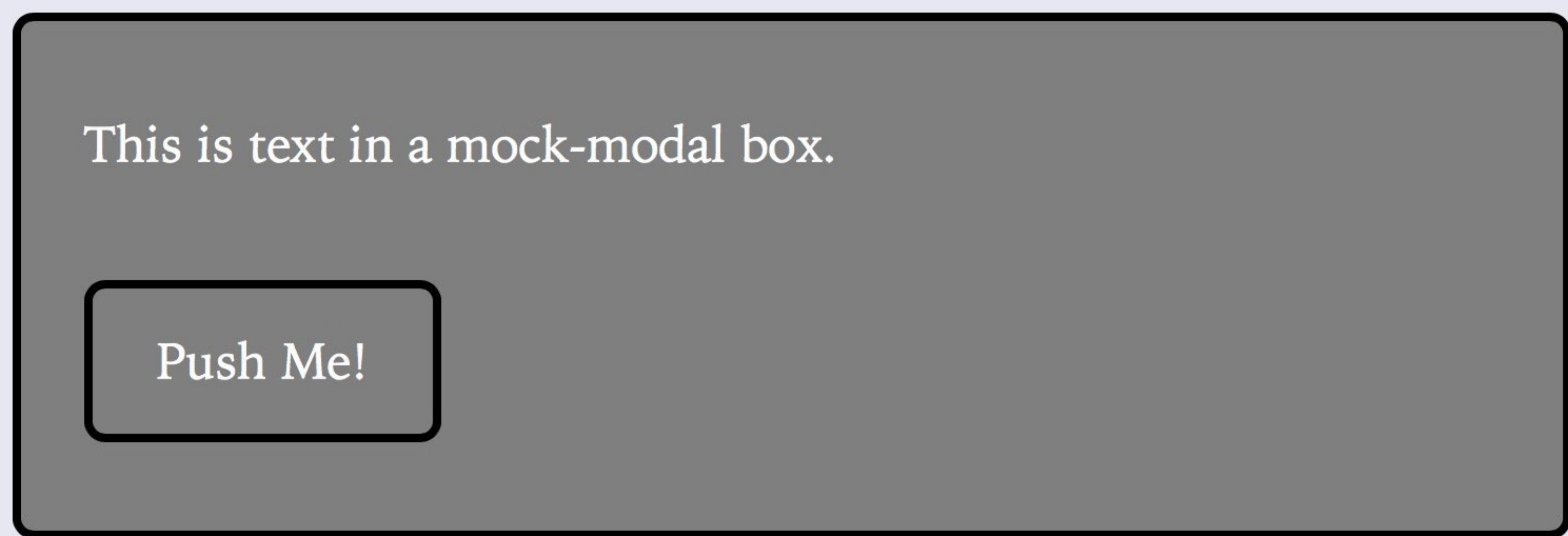
color: inherit;

17

}

18

JS



HTML

1

<div class="modal">

2

<p>

3

This is text in a mock-modal box.

CSS (SCSS)

8

.modal {

9

@include box-style;

10

color: HSL(0, 0%, 99%);

11

> button {

12

@include box-style;

13

margin: 1em 0;

14

font-size: 1rem;

15

font-family: inherit;

16

color: inherit;

17

}

18

}

19

20

/* uninteresting stuff below here */

21

22

html {

23

font-size: 24px;

24

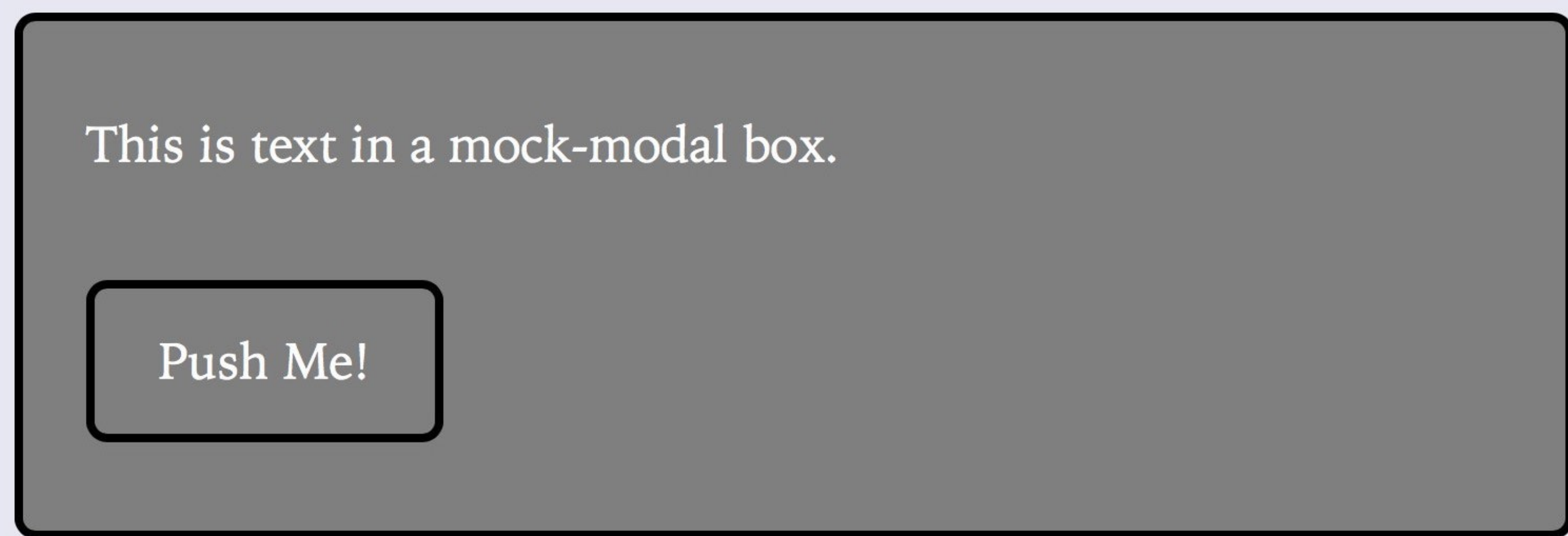
font-family: "Iowan Old Style",

Georgia, serif;

25

background-color: HSL(240, 22%,

JS



Movies & TV

HTML

Tidy

1<table class="movies">

2<thead>

3<tr>

4<th>Movie</th>

5<th>Genre</th>

6<th>Hero</th>

7<th>Year</th>

8</tr>

9</thead>

10<tbody>

11<tr>

12<td>Die Hard</td>

13<td>Action</td>

14<td>John McClane</td>

15<td>1988</td>

16</tr>

17<tr>

18<td>Star Wars</td>

19<td>Sci-Fi</td>

20<td>Luke Skywalker</td>

21<td>1977</td>

22</tr>

...

A table with the **movies** class

CSS (SCSS)

View Compiled

JS

Tidy

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

TV Show	Genre	Hero	Network
The Walking Dead	Horror	Daryl Dixon	AMC
Episodes	Comedy	Matt LeBlanc	Showtime
Supernatural	Action	Sam & Dean	The CW

HTML

Tidy

31

<table class="tv">

32

<thead>

33

<tr>

34

<th>TV Show</th>

35

<th>Genre</th>

36

<th>Hero</th>

37

<th>Network</th>

38

</tr>

39

</thead>

40

<tbody>

41

<tr>

42

<td>The Walking Dead</td>

43

<td>Horror</td>

44

<td>Daryl Dixon</td>

45

<td>AMC</td>

46

</tr>

47

<tr>

48

<td>Episodes</td>

49

<td>Comedy</td>

50

<td>Matt LeBlanc</td>

51

<td>Showtime</td>

52

</tr>

53

CSS (SCSS)

View Compiled

JS

Tidy

A table
with the
tv class

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

TV Show	Genre	Hero	Network
The Walking Dead	Horror	Daryl Dixon	AMC
Episodes	Comedy	Matt LeBlanc	Showtime
Supernatural	Action	Sam & Dean	The CW

HTML

Tidy

CSS (SCSS)

View Compiled

1

@mixin table-base {

2

th {

3

text-align: left;

4

}

5

td,

6

th {

7

padding: .25em .4em;

8

}

9

}

10

11

@mixin table-data {

12

th {

13

background-color: maroon;

14

color: white;

15

}

16

td {

17

border-bottom: 1px solid maroon;

18

}

19

}

20

21

.movies,

22

.tv {

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

Mixins
get re-
used
later

JS

Tidy

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

TV Show	Genre	Hero	Network
The Walking Dead	Horror	Daryl Dixon	AMC
Episodes	Comedy	Matt LeBlanc	Showtime
Supernatural	Action	Sam & Dean	The CW

HTML

Tidy

CSS (SCSS)

View Compiled

1

padding: 10px 10px;

2

3

4

5

6

7

8

}

9

}

10

11

@mixin table-data {

12

th {

13

background-color: maroon;

14

color: white;

15

}

16

td {

17

border-bottom: 1px solid maroon;

18

}

19

}

20

21

.movies,

22

.tv {

23

@include table-base;

24

}

25

26

.tv {

27

@include table-data;

28

}

29

JS

Tidy

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

TV Show	Genre	Hero	Network
The Walking Dead	Horror	Daryl Dixon	AMC
Episodes	Comedy	Matt LeBlanc	Showtime
Supernatural	Action	Sam & Dean	The CW

HTMLTidy

CSS (SCSS)View Compiled

```
1 @mixin table-base {
2   th {
3     text-align: left;
4   }
5   td,
6   th {
7     padding: .25em .4em;
8   }
9 }
10
11 @mixin table-data {
12   th {
13     background-color: maroon;
14     color: white;
15   }
16   td {
17     border-bottom: 1px solid maroon;
18   }
19 }
20
21 .movies,
22 .tv {
23   @include table-base;
```

JS

The
table-
base
mixin
gets re-
used

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

TV Show	Genre	Hero	Network
The Walking Dead	Horror	Daryl Dixon	AMC
Episodes	Comedy	Matt LeBlanc	Showtime
Supernatural	Action	Sam & Dean	The CW

HTML

Tidy

CSS

Compiled

View Compiled

1

.movies th,

2

.tv th {

3

text-align: left;

4

}

5

.movies td,

6

.movies th,

7

.tv td,

8

.tv th {

9

padding: .25em .4em;

10

}

11

12

.tv th {

13

background-color: maroon;

14

color: white;

15

}

16

.tv td {

17

border-bottom: 1px solid maroon;

18

}

19

20

/* Uninteresting stuff below here */

21

body {

22

font-family: "Iowan Old Style", Georgia,

Movie

Genre

Hero

Year

Die Hard

Action

John McClane

1988

Star Wars

Sci-Fi

Luke Skywalker

1977

John Wick

Action

John Wick

2014

TV Show

Genre

Hero

Network

The Walking Dead

Horror

Daryl Dixon

AMC

Episodes

Comedy

Matt LeBlanc

Showtime

Supernatural

Action

Sam & Dean

The CW

```
@mixin table-base {  
  th {  
    text-align: left;  
  }  
  td,  
  th {  
    padding: .25em .4em;  
  }  
}
```

```
.movies,  
.tv {  
  @include table-base;  
}
```

```
.movies th,  
.tv th {  
  text-align: left;  
}  
.movies td,  
.movies th,  
.tv td,  
.tv th {  
  padding: .25em .4em;  
}
```



```
@mixin table-data {  
  th {  
    background-color:  
maroon;  
    color: white;  
  }  
  td {  
    border-bottom: 1px  
solid maroon;  
  }  
}
```

```
.tv {  
  @include table-data;  
}
```

```
.tv th {  
  background-color:  
maroon;  
  color: white;  
}  
.tv td {  
  border-bottom: 1px  
solid maroon;  
}
```



Mixins Using @if Operatives

Repeating breakpoints

```
@media (max-width: 767px) {  
  img.hero {  
    display: none;  
  }  
}
```

```
@media (max-width: 767px) {  
  .multi-column {  
    column-count: 1;  
  }  
}
```

Not DRY! Time for a mixin!

⚙ HTML ▾

1 ▾

<p>

2

Ex nihilo nihil fit.

3

</p>

⚙ CSS (SCSS) ▾

⚙ JS ▾

Ex nihilo nihil fit.

Ex nihilo nihil fit.

1200 × 324 px

Ex nihilo nihil fit.

991 × 324 px

```
@mixin breakpoint($point) {  
    @if($point==sm) {  
        @media(min-width: 768px) {  
            @content  
        }  
    }  
    @if($point==md) {  
        @media(min-width: 992px) {  
            @content  
        }  
    }  
    @if($point==lg) {  
        @media(min-width: 1200px) {  
            @content  
        }  
    }  
}  
  
body {  
    background-color: pink;  
    @include breakpoint(md) {  
        background-color:  
lightgreen;  
    }  
}
```



```
@mixin breakpoint($point) {  
    @if($point==sm) {  
        @media(min-width: 768px) {  
            @content  
        }  
    }  
    @if($point==md) {  
        @media(min-width: 992px) {  
            @content  
        }  
    }  
    @if($point==lg) {  
        @media(min-width: 1200px) {  
            @content  
        }  
    }  
}
```

```
body {  
    background-color: pink;  
    @include breakpoint(md) {  
        background-color:  
lightgreen;  
    }  
}
```

```
@mixin breakpoint($point) {  
    @if($point==sm) {  
        @media(min-width: 768px) {  
            @content  
        }  
    }  
    @if($point==md) {  
        @media(min-width: 992px) {  
            @content  
        }  
    }  
    @if($point==lg) {  
        @media(min-width: 1200px) {  
            @content  
        }  
    }  
}
```

```
body {  
    background-color: pink;  
    @include breakpoint(md) {  
        background-color:  
lightgreen;  
    }  
}
```

```
@mixin breakpoint($point) {  
    @if($point==sm) {  
        @media(min-width: 768px) {  
            @content  
        }  
    }  
    @if($point==md) {  
        @media(min-width: 992px) {  
            @content  
        }  
    }  
    @if($point==lg) {  
        @media(min-width: 1200px) {  
            @content  
        }  
    }  
}
```

```
body {  
    background-color: pink;  
    @include breakpoint(md) {  
        background-color:  
lightgreen;  
    }  
}
```

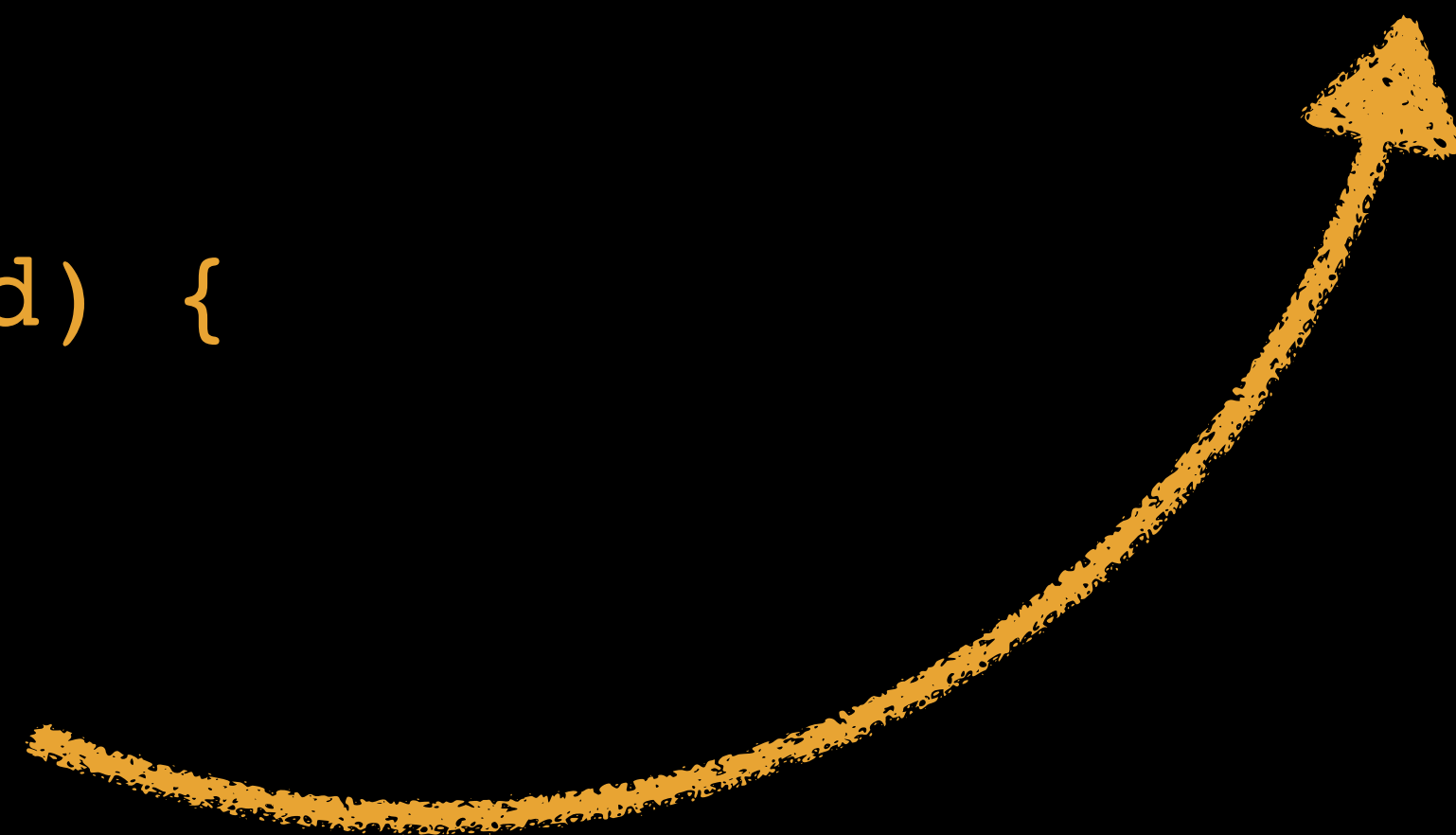
```
@mixin breakpoint($point) {  
  @if($point==md) {  
    @media(min-width: 992px) {  
      @content  
    }  
  }  
}
```

```
body {  
  background-color: pink;  
}
```

```
@include breakpoint(md) {  
  body {  
    background-color:  
lightgreen;  
  }  
}
```

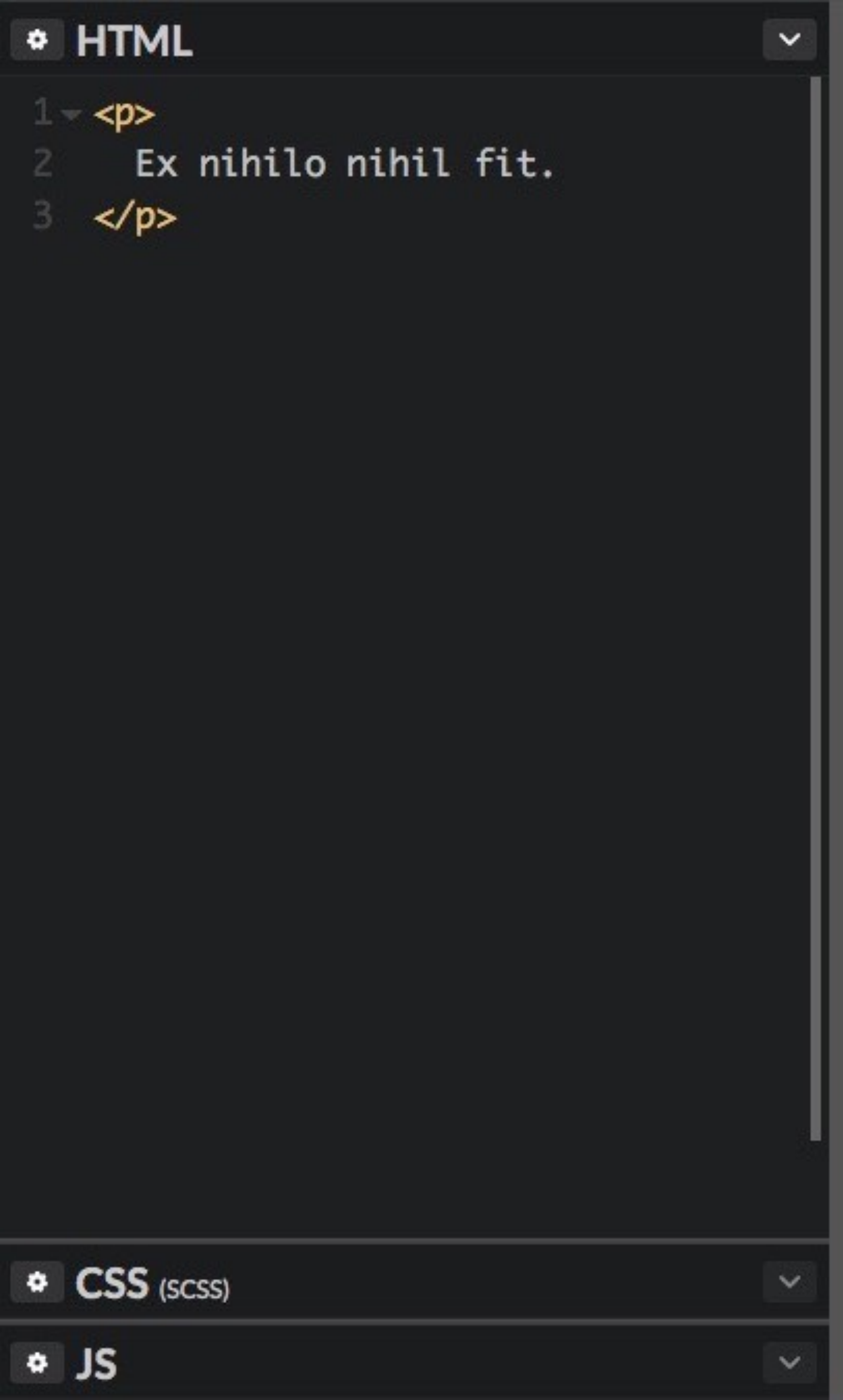
```
body {  
  background-color: pink;  
}
```

```
@media (min-width: 992px) {  
  body {  
    background-color:  
lightgreen;  
  }  
}
```



Ex nihilo nihil fit.

What about the centering?



Ex nihilo nihil fit.

```
html {  
  font: 24px "Iowan Old  
Style", serif;  
  height: 100%;  
}
```

Needed to vertically center
the `<p>`

Needed to vertically center
the `<p>`

```
body {  
  font-size: 4em;  
  min-height: 100%;  
  display: flex;  
  justify-content:  
center;  
  align-items: center;  
}
```

Flexbox!

Flexbox; horizontally centers
`<p>`

Flexbox; vertically centers
`<p>`

Why `height` on `<html>` & `min-height` on `<body>`?

html and body element - height:100% ?

DEC 3, 2014



absolutholz

Follow

| [absolutholz's Posts](#)

Something that has bothered me for quite some time, and maybe even some other poor soul out there. Is there a way to ensure that the height of the body and html elements of a webpage is at least as tall as the viewport and at most that of the content. Now I'm sure there are some use cases out there where the height of body and html should not be the same (position:fixed elements perhaps?) but for what I want to do this was not the case.

HTML

CSS (SCSS)

```
1 // Bootstrap breakpoints
2 @mixin breakpoint($point){
3   @if ($point == lg){
4     @media (min-width: 1200px){ @content }
5   }
6   @if ($point == md){
7     @media (min-width: 992px){ @content }
8   }
9   @if ($point == md-only){
10    @media (min-width: 992px) and (max-width: 1199px){ @content }
11  }
12  @if ($point == sm){
13    @media (min-width: 768px){ @content }
14  }
15  @if ($point == sm-only){
16    @media (min-width: 768px) and (max-width: 991px){ @content }
17  }
18  @if ($point == xs-only){
19    @media (max-width: 767px){ @content }
20  }
21 }
22
23 p {
24   padding: 1em;
25   background-color: yellow;
```

JS

I'm a little teapot

HTML

CSS (SCSS)

```
2 @mixin breakpoint($point){
3   @if ($point == lg){
4     @media (min-width: 1200px){ @content }
5   }
6   @if ($point == md){
7     @media (min-width: 992px){ @content }
8   }
9   @if ($point == md-only){↔}
12  @if ($point == sm){
13    @media (min-width: 768px){ @content }
14  }
15  @if ($point == sm-only){↔}
18  @if ($point == xs-only){↔}
21 }
22
23 p {
24   padding: 1em;
25   background-color: yellow;
26   font-size: 24px;
27   @include breakpoint(sm) {
28     background-color: red;
29   }
30   @include breakpoint(md) {
31     background-color: green;
32 }
```

JS

I'm a little teapot

HTML

CSS (SCSS)

```
2 @mixin breakpoint($point){
3   @if ($point == lg){
4     @media (min-width: 1200px){ @content }
5   }
6   @if ($point == md){↔}
9   @if ($point == md-only){↔}
12  @if ($point == sm){↔}
15  @if ($point == sm-only){↔}
18  @if ($point == xs-only){↔}
21 }
22
23 p {
24   padding: 1em;
25   background-color: yellow;
26   font-size: 24px;
27   @include breakpoint(sm) {
28     background-color: red;
29   }
30   @include breakpoint(md) {
31     background-color: green;
32   }
33   @include breakpoint(lg) {
34     background-color: blue;
35     color: white
36  }
```

JS

I'm a little teapot

HTML

CSS (SCSS)

```
1 // Bootstrap breakpoints
2 @mixin breakpoint($point){
3   @if ($point == lg){↔}
6   @if ($point == md){↔}
9   @if ($point == md-only){↔}
12  @if ($point == sm){
13    @media (min-width: 768px){ @content }
14  }
15  @if ($point == sm-only){↔}
18  @if ($point == xs-only){↔}
21 }
22
23 p {
24   padding: 1em;
25   background-color: yellow;
26   font-size: 24px;
27   @include breakpoint(sm) {
28     background-color: red;
29   }
30   @include breakpoint(md) {↔}
33   @include breakpoint(lg) {↔}
37 }
```

JS

I'm a little teapot

HTML

CSS (SCSS)

```
6  @if ($point == md){
7    @media (min-width:
  992px){ @content }
8  }
9  @if ($point == md-
only){↔}
12 @if ($point == sm){↔}
15 @if ($point == sm-
only){↔}
18 @if ($point == xs-
only){↔}
21 }
22
23 p {
24   padding: 1em;
25   background-color:
yellow;
26   font-size: 24px;
27   @include
breakpoint(sm) {↔}
30   @include
breakpoint(md) {
31     background-color:
green;
32   }
```

JS

I'm a little teapot

HTML

CSS (SCSS)

```
1 // Bootstrap breakpoints
2 @mixin breakpoint($point){
3   @if ($point == lg){
4     @media (min-width: 1200px){
5       @content }
6   }
7   @if ($point == md){↔}
8   @if ($point == md-only){↔}
9   @if ($point == sm){↔}
10  @if ($point == sm-only){↔}
11  @if ($point == xs-only){↔}
12 }
13
14
15 p {
16   padding: 1em;
17   background-color: yellow;
18   font-size: 24px;
19   @include breakpoint(sm) {↔}
20   @include breakpoint(md) {↔}
21   @include breakpoint(lg) {
22     background-color: blue;
23     color: white
24   }
25 }
```

JS

I'm a little teapot

I'm a little teapot

Sass *functions* are called without any special keyword & return a value

```
background-color: darken($main-bg-color, 20%);  
→ background-color: hsl(32, 100%, 29%);
```

Sass *mixins* are called with `@include` & inject their contents within the scope in which they're called

Advantages of mixins

- » Efficient & clean code repetitions
- » Greater consistency
- » Fewer errors
- » Easy alteration of code: only have to update changes in one place (like CSS itself!)

Collections of Mixins



Bourbon

[GITHUB](#)[DOCS](#)

A simple and lightweight mixin library for Sass.

```
ol {  
  > li {  
    background-color: #F4F5F7;  
    @include background-image( linear-gradient(to left, #FFF, #F4F5F7) );  
    float: left;  
    font-family: $lucida-grande;  
    position: relative;  
    @include transition( all 0.8s );  
  
    &::after {
```

 GITHUB

EXAMPLES

DOCS

A **lightweight** semantic grid
framework for **Sass** and **Bourbon**.

```
<!-- HTML markup for the section right below this code block -->
```

```
<section>
```




Compass is an open-source *CSS Authoring Framework*.

♥ Why designers love Compass.

1. Experience cleaner markup without presentational classes.
2. It's chock full of the web's best [reusable patterns](#).
3. It makes creating [sprites](#) a breeze.
4. Compass mixins make [CSS3](#) easy.
5. Create beautiful [typographic rhythms](#).

✓ Compass uses Sass.

[Sass](#) is an extension of CSS3 which adds nested rules, variables, mixins, selector inheritance, and more. Sass generates well formatted CSS and makes your stylesheets easier to organize and maintain.



useful sass mixins



Images Videos

Useful SASS Mixins | SachaGreif.com

Region

I'm a big fan of **SASS** and Compass, two tools which help you write CSS better and faster. I previously wrote about managing color variables, and today I'd like to ...

sachagreif.com/useful-sass-mixins/

5 Useful Sass Mixins in Bootstrap - SitePoint

Reginald Dawson has some go-to **mixins** that he's uncovered in Bootstrap's **Sass** port. Here he describes 5 of them and how they can be used in your Bootstrap projects.

sitepoint.com/5-useful-sass-mixins-bootstrap/

Useful SASS Mixins | Code Responsible

Helpful **SASS Mixins** to help speed up your CSS and website development process.

coderesponsible.com/useful-sass-mixins/

Useful Sass mixins — Medium

Why **Sass**? **Sass** boasts more features and abilities than any other CSS extension language out there and **mixins** are one of the most powerful features of **Sass**.

medium.com/@vilcins/usefull-sass-mixins-7a68c5491bda

AsyncNick/SASS-Useful-Mixins · GitHub

SASS-Useful-Mixins - **Useful SASS Mixins**. Skip to content. Sign up Sign in. This repository. Explore; Features; Enterprise; Pricing; Watch 1 Star 1 Fork 8

github.com/AsyncNick/SASS-Useful-Mixins

#132: A Quick Useful Case for Sass Math and Mixins | CSS-Tricks

Send feedback

@import Directive

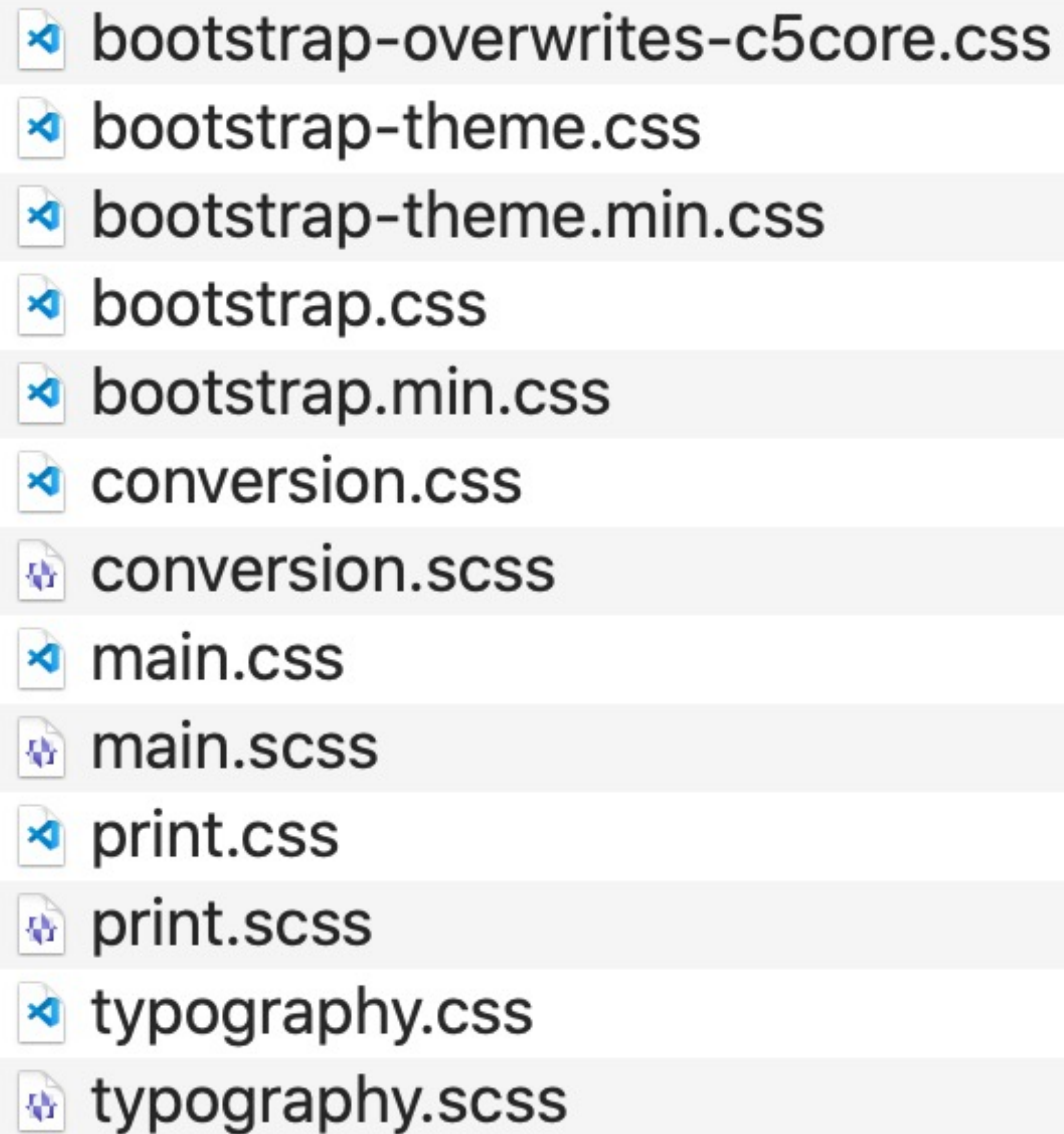
In CSS, `@import` allows authors to import other CSS files

Sass extends `@import` to import Sass files before compiling the merged Sass file into CSS

In `main.scss`:

```
@import "typography.scss";  
@import "navigation.scss";  
@import "footer.scss";
```

All those files get merged into `main.scss` in that order,
which is then compiled to `main.css`

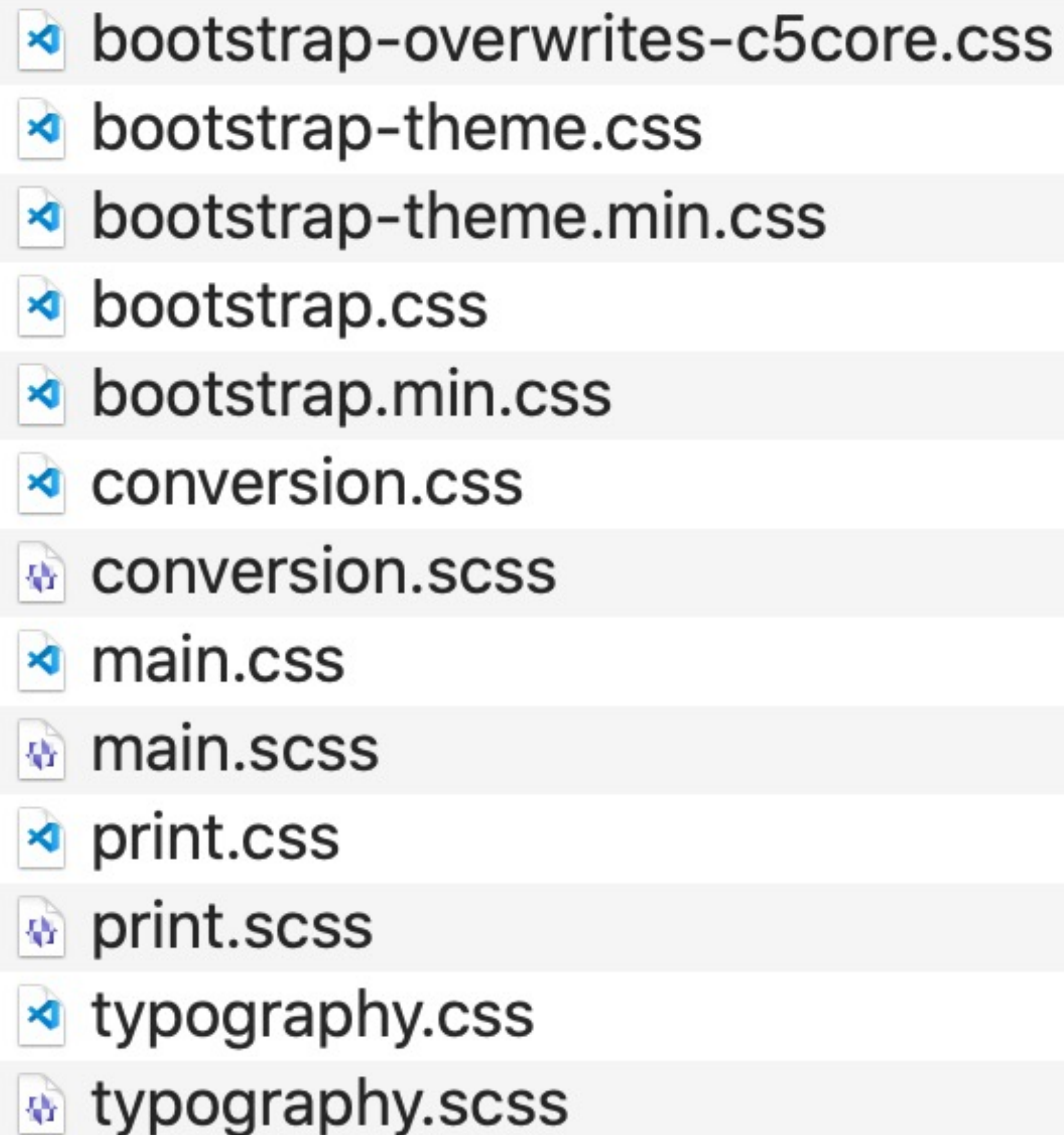


A screenshot of a file explorer window showing a list of files. The files are arranged in a list with alternating light and dark gray background rows. Each file has a small icon to its left: a blue icon with a white 'X' for CSS files and a purple icon with a white 'X' for SCSS files. The files listed are:

- bootstrap-overwrites-c5core.css
- bootstrap-theme.css
- bootstrap-theme.min.css
- bootstrap.css
- bootstrap.min.css
- conversion.css
- conversion.scss
- main.css
- main.scss
- print.css
- print.scss
- typography.css
- typography.scss

Sass files go in the same directory as the CSS*

*OK, not really, but it's far easier



A screenshot of a file explorer window showing a list of files. The files are arranged in a list with alternating light and dark gray background rows. Each file has a small icon to its left: a blue icon with a white 'x' for CSS files and a purple icon with a white 'x' for SCSS files. The files listed are:

- bootstrap-overwrites-c5core.css
- bootstrap-theme.css
- bootstrap-theme.min.css
- bootstrap.css
- bootstrap.min.css
- conversion.css
- conversion.scss
- main.css
- main.scss
- print.css
- print.scss
- typography.css
- typography.scss

The `.scss` files are imported into `main.scss`, which is then compiled into `main.css`

Notice that every `.scss` file is also compiled to `.css`

Why create all those unnecessary `.css` files?

*OK, not really, but it's far easier

Sass supports `@import` *partials*

Sass files that start with `_` are not compiled into CSS,
but are instead imported as Sass into the merged SCSS
file

`_variables.scss`

Store your Sass variables in 1 file

When imported, all other Sass files can use those variables

You obviously don't want this compiled into CSS; you want it as a separate file that can merged into


















`main.scss`

So if you want `main.scss` to be compiled to `.css` & no other file, in `main.scss` you would have:

```
@import "_navigation.scss";  
@import "_footer.scss";  
@import "_typography.scss";
```

None of the files starting with `_` are converted to `.css`

However, all files starting with `_` get merged into `main.scss` & then compiled to `main.css`

-  _fonts.scss
-  _images.scss
-  _nav.scss
-  _print.scss
-  _syllabus.scss
-  _tables.scss
-  _typography.scss
-  _variables.scss
-  bootstrap-overwrites-c5core.css
-  bootstrap-theme.css
-  bootstrap-theme.min.css
-  bootstrap.css
-  bootstrap.min.css
-  conversion.css
-  conversion.scss
-  main.css
-  main.scss

Contents of `main.scss`:

```
@import "_variables.scss";  
  
@import "_fonts.scss";  
  
@import "_typography.scss";  
  
@import "_nav.scss";  
  
@import "_syllabus.scss";  
  
@import "_tables.scss";  
  
@import "_images.scss";  
  
@import "_print.scss";
```

Comments

CSS comments are single line or multi-line:

```
/* foo */
```

Sass comments

» single line: `/* foo */` or `// foo`

» multi-line: `/* foo */`

`/* foo */` comments *are compiled* into CSS

`// foo` comments are *not compiled* into CSS


```
.foo {  
  // That's it??  
  font-family: serif;  
  .bar {  
    /* Corp. colors */  
    // Hideous!! 🤢  
    background-color:  
blue;  
    color: orange;  
  }  
}
```

Sass

```
.foo {  
  font-family: serif; }  
.foo .bar {  
  /* Corp. colors */  
  background-color:  
blue;  
  color: orange; }
```

Compiled CSS

Autoprefixer

Autoprefixer has nothing to do with Sass, but is often used along with it — it's all part of automating the process

New features come out in CSS, but sometimes the property or value isn't yet nailed down

Browser makers guess what the final answers will be, & use *vendor prefixes* so developers can test & experiment now

When the W3C releases a final spec, all browsers standardize on that

When flexbox was first introduced, it wasn't certain what the value for `display` would be

```
.featured {  
  display: -webkit-box;  
  display: -webkit-flex;  
  display: -ms-flexbox;  
  display: flex;  
}
```

Always put the final value specified by the W3C last!

<code>-moz-</code>	Mozilla Firefox Gecko-based browsers
<code>-ms-</code>	Microsoft Internet Explorer
<code>-webkit-</code>	Apple Safari for macOS & iOS Chromium/Google Chrome WebKit-based browsers

<code>-webkit-box</code>	Safari on macOS 3.1–6 Safari on iOS 3.2–6.1
<code>-webkit-flex</code>	Safari on macOS 6.1–8 Safari on iOS 7–8.4 Chrome 21–28
<code>-ms-flexbox</code>	Internet Explorer 10
<code>-moz-box</code>	Firefox 2–21
<code>flex</code>	Firefox 21+ Chrome 29+ IE 11+/Edge Safari on macOS & iOS 9+

Only include vendor prefixes for browsers you want to still support

You should revisit your code as browser support improves & remove old prefixes for browsers you don't care about targeting any longer

Firefox 21 came out in April 2013 & is used by 0.01% of the world, so I can probably remove `-moz-box`

Who has time to keep track of all this?!



This repository

Search

Pull requests

Issues

Marketplace

Explore



postcss / autoprefixer

Watch

393

Star

14,533

Fork

839

Code

Issues 8

Pull requests 0

Projects 0

Wiki

Insights

Parse CSS and add vendor prefixes to rules by Can I Use <https://twitter.com/autoprefixer>

css

postcss-plugins

prefixer

2,003 commits

2 branches

93 releases

125 contributors

MIT

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



ntwb authored and ai committed a day ago Add missing space in `autoprefixer -info` CLI docs (#978)

Latest commit 5d94aec a day ago

bin	Add autoprefixer-info CLI tool	2 months ago
data	Add grid-template-areas support (#948)	2 months ago
lib	Fix value prefixing	19 days ago
test	Fix value prefixing	19 days ago
.editorconfig	Use standard PostCSS .editorconfig	9 months ago
.gitignore	Add code coverage, tests of initializing and remove not used code (#928)	3 months ago
.npmignore	Add code coverage, tests of initializing and remove not used code (#928)	3 months ago
.travis.yml	Use Jest 22	a month ago

Run autoprefixer, & it...

- » adds prefixes where necessary
- » scans your CSS & removes outdated prefixes

You tell autoprefixer which browsers (& therefore prefixes) it should support, e.g.:

- » `last 2 versions`
- » `last 2 Chrome versions`
- » `Firefox > 20`
- » `last 2 iOS major versions`
- » `> 5%` (global)
- » `> 3% in US`
- » `last 2 years`
- » `> 1%, last 2 versions, Firefox ESR` (default)

`autoprefixer` is a command-line tool, but you can:

- » automate running it with gulp or your favorite task runner
- » install plugins for Visual Studio Code, Brackets, Sublime Text, & other editors

Vendor prefixes are a PITA, bloat stylesheets, & require a lot of maintenance

Browser makers are moving away from vendor prefixes in favor of other methods

- » @supports

- » Browser Advanced Settings (AKA *flags*)

@supports is a conditional group rule that lets you test if a browser supports CSS `property: value` pairs

```
@supports (display: grid) {  
  .grid-container {  
    display: grid;  
  }  
}
```

If the browser supports `display: grid`, use it

But what if it does not?

The `not` keyword lets you specify what will happen if the browser does *not* support the `property: value` pair




```
@supports not (display: grid) {  
  .grid-container {  
    float: right;  
  }  
}
```

If the browser does *not* support `display: grid`, use `float: right` instead

You can also chain multiple property checks together using **or** or **and**

```
@supports (display: -ms-grid) or  
  (display: grid) {  
  .grid-container {  
    display: grid;  
  }  
}
```


Best of all, autoprefixer is now automatically taking care of `@supports` for you too!

					ios		
@supports	—	12	22	9	9	28	4.4

Works with 96.43% of global browsers (July 2020)

Chrome: `chrome://flags`

Firefox: `about:config`

Safari: Develop > Experimental Features

Edge (pre-Chromium): `about:config`

Sass Tools

Visual Studio Code

Live Sass Compiler

Compiles Sass files to CSS on Save

Turn it on

- » Click Watch Sass on status bar
- » Invoke the Command Palette & enter **Live Sass:**
Watch Sass

For more, see

- » github.com/ritwickdey/vscode-live-sass-compiler
- » [marketplace.visualstudio.com/items?
itemName=ritwickdey.live-sass](https://marketplace.visualstudio.com/items?itemName=ritwickdey.live-sass)

CodePen

CodePen allows users to enter Sass (Sass or SCSS, Less, & others) in the CSS box, which is then compiled into CSS

⚙

HTML

Tidy

✕

1

⚙

CSS

Tidy

View Compiled

✕

1

⚙

JS

Tidy

✕

1





HTML

Tidy



1



CSS

Tidy

View Compiled



1



JS

Tidy



We don't need JS, so we close it





HTML

Tidy



1



CSS

Tidy

View Compiled



JS

Tidy



Click on the action
button to open
CSS Settings





HTML

1

CSS

1

JS

Tidy



Pen Settings

HTML CSS JavaScript Behavior

CSS Preprocessor



None



CSS Base



☐ NORMALIZE ☐ RESET ☒ NEITHER

Vendor Prefixing



☒ AUTOPREFIXER ☐ PREFIXFREE ☐ NEITHER

Add External CSS



These stylesheets will be added in this order and before the code you write in the CSS editor. You can also add another Pen here, and it will pull the CSS from it. Try typing "font" or "ribbon" below.



https://yourwebsite.com/style.css



http://codenpen.io/username/pen/aBcDef



Public



Private

PRO



PEN TITLE

PEN DESCRIPTION

Explain what's going on in your Pen here. This text is searchable, so it can also help others find your work. Remember to credit others where credit is due. Markdown supported.

TAGS

COMMA SEPARATED, MAX OF FIVE

☐ TEMPLATE?



Save & Close

Last saved 1 minute ago

Share

Export



HTML

1

Pen Settings

HTML CSS JavaScript Behavior

CSS Preprocessor

- ✓ None
- LESS
- SCSS
- Sass
- Stylus
- PostCSS

Vendor Prefixing

☒ AUTOPREFIXER ☐ PREFIXFREE ☐ NEITHER

Add External CSS

These stylesheets will be added in this order and before the code you write in the CSS editor. You can also add another Pen here, and it will pull the CSS from it. Try typing "font" or "ribbon" below.

-
-

Public ☒ Private **PRO**

PEN TITLE

PEN DESCRIPTION

Explain what's going on in your Pen here. This text is searchable, so it can also help others find your work. Remember to credit others where credit is due. Markdown supported.

TAGS

COMMA SEPARATED, MAX OF FIVE

☐ TEMPLATE?

CSS

1

JS

Tidy

Save & Close

HTML

1

Pen Settings

HTML CSS JavaScript Behavior

CSS Preprocessor

SCSS

Need an add-on?

?

⌵

CSS Base

☐ NORMALIZE

☐ RESET

☒ NEITHER

?

Vendor Prefixing

☒ AUTOPREFIXER

☐ PREFIXFREE

☐ NEITHER

?

Add External CSS

These stylesheets will be added in this order and before the code you write in the CSS editor. You can also add another Pen here, and it will pull the CSS from it. Try typing "font" or "ribbon" below.

≡

https://yourwebsite.com/style.css

×

≡

×

CSS (SCSS)

1

Public ☒ Private **PRO** ?

PEN TITLE

PEN DESCRIPTION

Explain what's going on in your Pen here. This text is searchable, so it can also help others find your work. Remember to credit others where credit is due. Markdown supported.

TAGS

COMMA SEPARATED, MAX OF FIVE

☐ TEMPLATE?

?

JS

Tidy ×

Save & Close

Go to # on this page

Console Assets Embed Comments × Delete

Last saved 1 minute ago

Share Export



HTML

1

Pen Settings

HTML CSS JavaScript Behavior

CSS Preprocessor

Need an add-on?



SCSS



ADD-ONS FOR SASS (.SCSS)

Just like a real project, use these lines at the top of your CSS to import them as a dependency.

```
@import "susy";  
@import "breakpoint";  
@import "compass";  
  @import "color-schemer"; /* depends on above */  
@import "bourbon";  
  @import "neat"; /* depends on above */  
@import "modular-scale";
```



CSS Base



JS

Tidy



Save & Close

Public



Private

PRO



PEN TITLE

PEN DESCRIPTION

Explain what's going on in your Pen here. This text is searchable, so it can also help others find your work. Remember to credit others where credit is due. Markdown supported.

TAGS

COMMA SEPARATED, MAX OF FIVE

☐ TEMPLATE?



We don't need any...
just looking

Collections

Console

Assets

Embed

Comments

× Delete

Last saved 1 minute ago

Share

Export

HTML

1

Pen Settings

HTML CSS JavaScript Behavior

CSS Preprocessor

Need an add-on?

SCSS

ADD-ONS FOR SASS (.SCSS)

Just like a real project, use these lines at the top of your CSS to import them as a dependency.

```
@import "susy";
```

```
@import "breakpoint";
```

```
@import "compass";
```

```
@import "color-schemer" /* deprecated */;
```

```
@import "bourbon";
```

```
@import "normalize";
```

```
@import "modern-normalize";
```

CSS Base

ABOUT BOURBON

Bourbon is "a simple and lightweight mixin library for Sass." It helps with things like making sure the selectors, properties, and values are output correctly for the best cross-browser support as well as other helper functions

Public ☒ Private **PRO**

PEN TITLE

PEN DESCRIPTION

Explain what's going on in your Pen here. This text is searchable, so it can also help others find your work. Remember to credit others where credit is due. Markdown supported.

TAGS COMMA SEPARATED, MAX OF FIVE

☐ TEMPLATE?

Click to learn more

JS

Tidy

Save & Close



HTML

1

Pen Settings

HTML CSS JavaScript Behavior

CSS Preprocessor

Need an add-on?



SCSS



CSS Base



☐ NORMALIZE ☐ RESET ☒ NEITHER

Vendor Prefixing



☒ AUTOPREFIXER ☐ PREFIXFREE ☐ NEITHER

Add External CSS



These stylesheets will be added in this order and before the code you write in the CSS editor. You can also add another Pen here, and it will pull the CSS from it. Try typing "font" or "ribbon" below.



https://yourwebsite.com/style.css



Public



Private

PRO



PEN TITLE

PEN DESCRIPTION

Explain what's going on in your Pen here. This text is searchable, so it can also help others find your work. Remember to credit others where credit is due. Markdown supported.

TAGS

COMMA SEPARATED, MAX OF FIVE

☐ TEMPLATE?



Nice!

JS

Tidy



Save & Close



Collections

Console

Assets

Embed

Comments

× Delete

Last saved 1 minute ago

Share

Export



HTML

1

Pen Settings

HTML CSS JavaScript Behavior

CSS Preprocessor

Need an add-on?



SCSS



CSS Base



☐ NORMALIZE ☐ RESET ☒ NEITHER

Vendor Prefixing



☒ AUTOPREFIXER ☐ PREFIXFREE ☐ NEITHER

Add External CSS



These stylesheets will be added in this order and before the code you write in the CSS editor. You can also add another Pen here, and it will pull the CSS from it. Try typing "font" or "ribbon" below.



https://yourwebsite.com/style.css



JS

Tidy



Save & Close



Collections

Console

Assets

Embed

Comments


× Delete


Last saved 1 minute ago

Share

Export




 HTML Tidy ×

 CSS (SCSS) View Compiled ×

1



 JS Tidy ×



HTML

Tidy



```
1 <section>
2   <header>
3     <h2>Heading Text</h2>
4   </header>
5   <p>
6     Normal text
7   </p>
8 </section>
```

CSS (SCSS)

View Compiled



```
1 body {
2   section {
3     header {
4       color: red;
5     }
6     p {
7       color: black;
8     }
9   }
10 }
```

JS

Tidy



Heading Text

Normal text



HTML

Tidy



CSS (SCSS)

View Compiled



```
1 body {  
2   section {  
3     header {  
4       color: red;  
5     }  
6   p {  
7     color: black;  
8   }  
9 }  
10 }
```

Don't really need
to see the HTML...

JS

Tidy



Heading Text

Normal text





HTML

Tidy



CSS (SCSS)

View Compiled



Toggle Compiled View

```
1 body {  
2   section {  
3     header {  
4       color: red;  
5     }  
6     p {  
7       color: black;  
8     }  
9   }  
10 }
```

Heading Text

Normal text

We typed in SCSS...
let's see the resulting CSS

JS

Tidy





HTML

Tidy



CSS Compiled

View Compiled



```
1 body section header {  
2   color: red;  
3 }  
4 body section p {  
5   color: black;  
6 }  
7
```

Heading Text

Normal text

Notice it changed from
SCSS to Compiled

JS

Tidy





HTML

Tidy



CSS (SCSS)

View Compiled



```
1 body {  
2   section {  
3     header {  
4       color: red;  
5     }  
6     p {  
7       color: black;  
8     }  
9   }  
10 }
```

Heading Text

Normal text

JS

Tidy



Task Runners

You're working on a project & after every edit to a Sass file you want to:

- » Check your Sass file for errors (*lint* it)
- » Compile Sass to CSS
- » Insert vendor prefixes for the last 2 browser versions
- » Minimize your CSS
- » View the HTML file with the new CSS in a browser

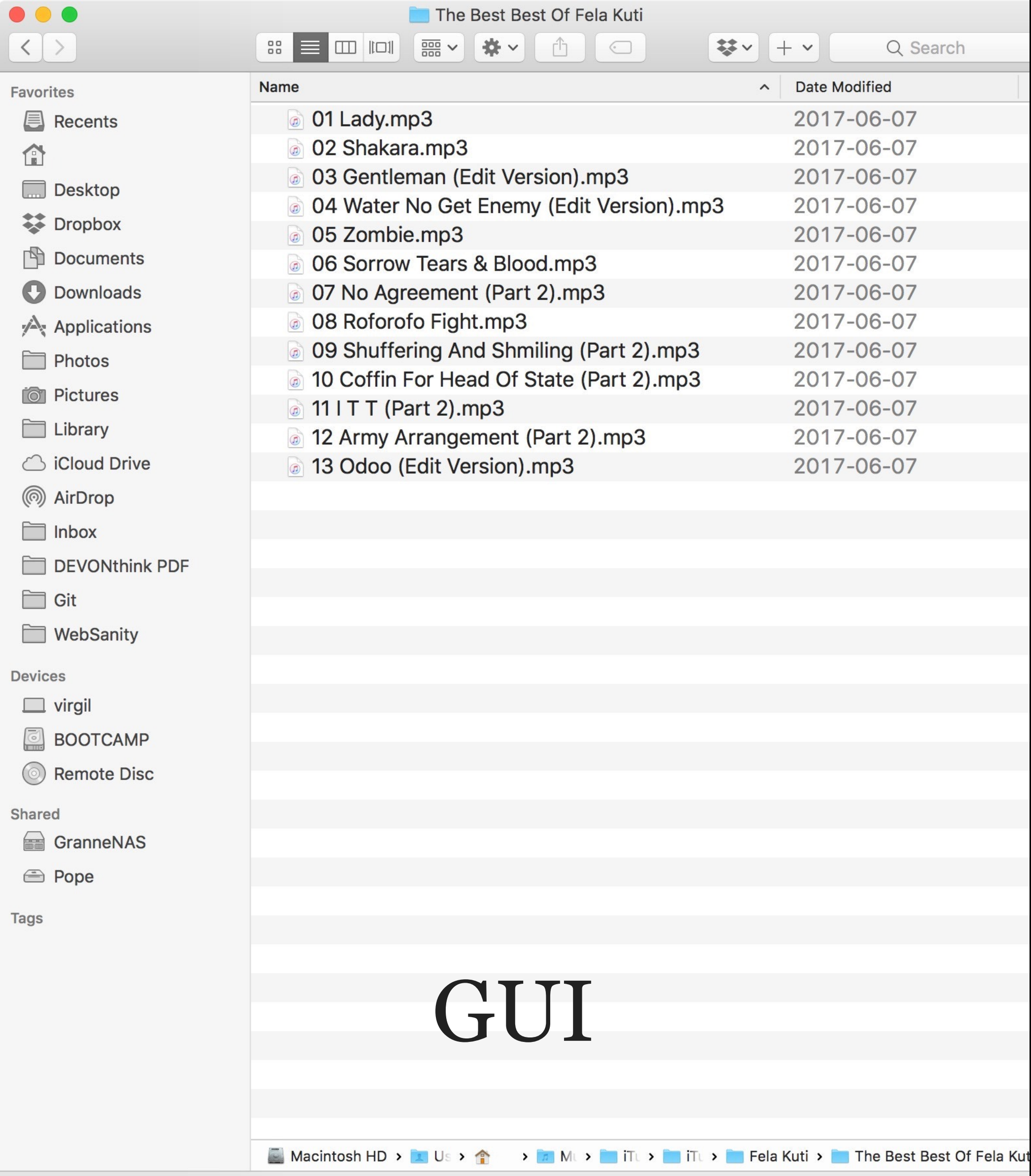
That's a lot of work to do manually!

The same sort of multi-step process is true for HTML, JavaScript, images, & many other kinds of files

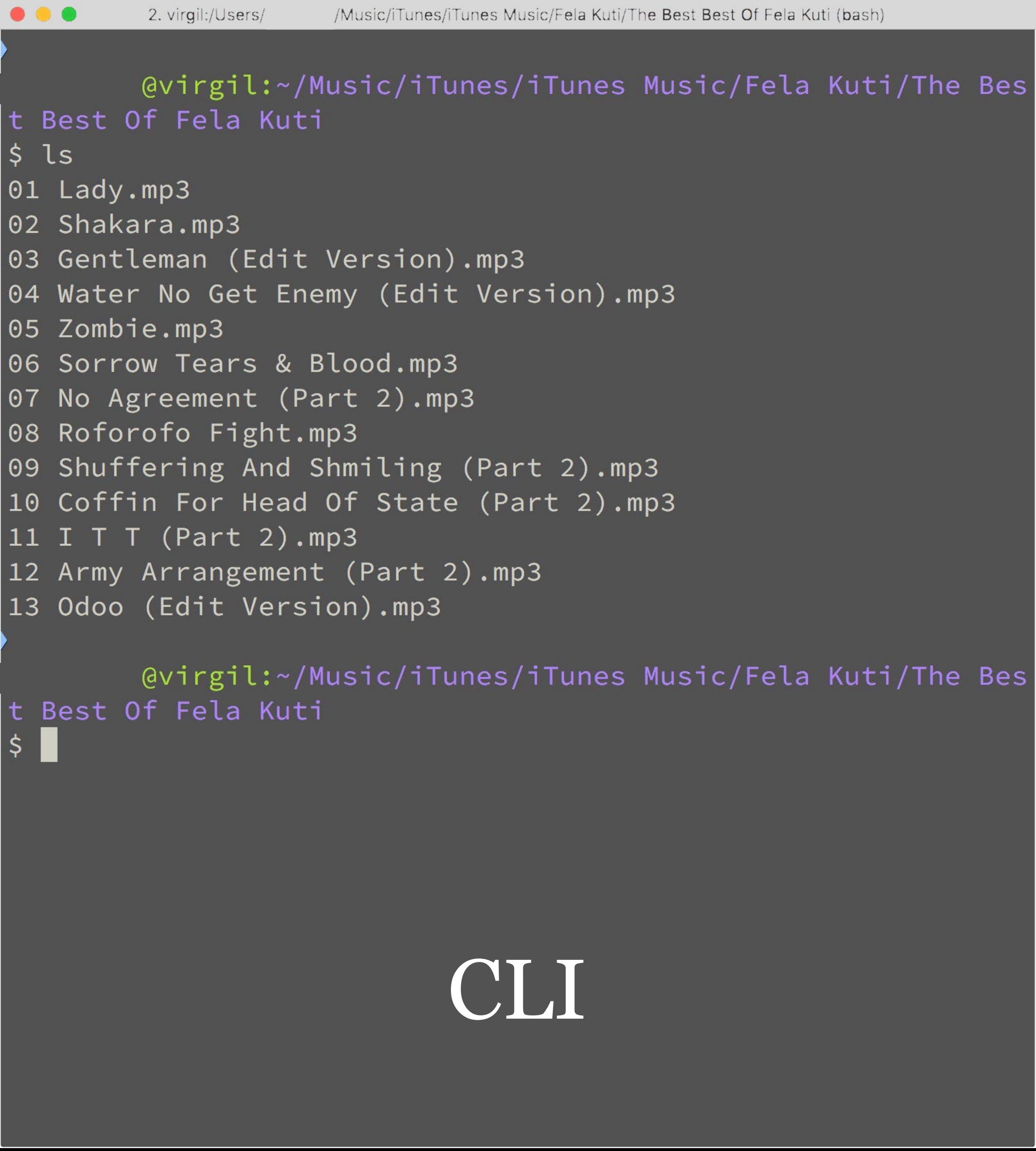
We need to automate the process as much as possible!

A task runner is a tool that automatically performs frequent tasks you need done as part of a build process

Can use either a CLI (*command-line interface*) or a GUI (*graphical user interface*)



GUI



CLI

CLI Task Runners

2 very popular task runners: Grunt & Gulp

Both allow you to create *watched folders*

When something happens in a watched folder (e.g., a file is saved), then certain tasks are automatically run

<> Code

🔔 Issues 23

🔗 Pull requests 2

📁 Projects 0

⚡ Pulse

📊 Graphs

Branch: master ▾

gulp / docs / recipes / rebuild-only-files-that-change.md

Find file

Copy path



danielbayerlein gulp-watch v3.0.0 API

c9563c7 on Dec 19, 2014

2 contributors



17 lines (13 sloc) | 355 Bytes

Raw

Blame

History



Rebuild only files that change

With `gulp-watch`:

```
var gulp = require('gulp');
var sass = require('gulp-sass');
var watch = require('gulp-watch');

gulp.task('default', function() {
  return gulp.src('sass/*.scss')
    .pipe(watch('sass/*.scss'))
    .pipe(sass())
    .pipe(gulp.dest('dist'));
});
```

Grunt



GRUNT

The JavaScript Task Runner

Latest Version

- Stable: [v1.0.1](#) (npm)
- Development: [master](#) (GitHub)

Ads by [Bocoup](#).

Latest News

[Grunt 1.0.0 released](#)

April 04, 2016

Why use a task runner?

In one word: automation. The less work you have to do when performing repetitive tasks like minification, compilation, unit testing, linting, etc, the easier your job becomes. After you've configured it through a [Gruntfile](#), a task runner can do most of that mundane work for you—and your team—with basically zero effort.

Why use Grunt?






The Grunt ecosystem is huge and it's growing every day. With literally hundreds of plugins to choose from, you can use Grunt to automate just about anything with a minimum of effort. If someone hasn't already built what you need, authoring and publishing your own Grunt plugin to npm is a breeze. See how to [get started](#).

Available Grunt plugins

Free & open source JavaScript task runner

6,000+ plugins available via npm (*Node Package Manager*)

Used internally by Adobe, Twitter, Mozilla, Bootstrap, WordPress, Walmart, & Microsoft

Plugin	Downloads last 30 days
 contrib-concat by Grunt Team Concatenate files.	693577
concurrent by Sindre Sorhus Run grunt tasks concurrently	295295
 contrib-compress by Grunt Team Compress files and folders	201229
 contrib-imagemin by Grunt Team Minify images	165796
autoprefixer by Dmitry Nikitenko Parse CSS and add vendor-prefixed CSS properties using the Can I Use database. Based on Autoprefixer.	161251
 contrib-htmlmin by Grunt Team Minify HTML	158614
angular-templates by Eric Clemmons Grunt build task to concatenate & register your AngularJS templates in the \$templateCache	154487
 contrib-coffee by Grunt Team Compile CoffeeScript files to JavaScript	127306

Gulp

Automate and enhance your workflow

```
$ npm install gulp-cli -g  
$ npm install gulp -D  
$ touch gulpfile.js  
$ gulp --help
```


Free & open source JavaScript task runner

3,600+ plugins available via npm

Requires knowledge of JavaScript to define tasks

●●● gulpfile.js

```
var gulp = require('gulp');
var pug = require('gulp-pug');
var less = require('gulp-less');
var minifyCSS = require('gulp-cssso');

gulp.task('html', function(){
  return gulp.src('client/templates/*.pug')
    .pipe(pug())
    .pipe(gulp.dest('build/html'));
});

gulp.task('css', function(){
  return gulp.src('client/templates/*.less')
    .pipe(less())
    .pipe(minifyCSS())
    .pipe(gulp.dest('build/css'));
});

gulp.task('default', [ 'html', 'css' ]);
```

Sample `gulpfile.js`

Jans' gulpfile.js

```
// Sass configuration
var gulp = require('gulp');
var sass = require('gulp-sass');
var autoprefixer = require('gulp-autoprefixer');

var sassOptions = {
  errLogToConsole: true,
  outputStyle: 'expanded'
};

var autoprefixerOptions = {
  overrideBrowserslist: ['last 2 versions', '> 5%', 'Firefox ESR']
};

gulp.task('sass', function(cb) {
  gulp
    .src(['**/*.scss', '!**/bootstrap/**'])
    .pipe(sass(sassOptions).on('error', sass.logError))
    .pipe(autoprefixer(autoprefixerOptions))
    .pipe(
      gulp.dest(function(f) {
        return f.base;
      })
    );
  cb();
});

gulp.task(
  'default',
  gulp.series('sass', function(cb) {
    gulp.watch(['**/*.scss', '!**/bootstrap/**'], gulp.series('sass'));
    cb();
  })
);
```

From Sass
Back to CSS

The growing influence of Sass (& Less) was noticed by the W3C, so they're incorporating ideas from pre-processors into CSS

Custom Properties

CSS *custom properties* (formerly called *CSS variables*) are very similar in some ways to Sass variables — they're variables, after all — but with big differences too

```
$main-color: orange;
```

```
footer {  
  background-color:  
$main-color;  
}
```

Sass

```
:root {  
  --main-color:  
orange;  
}
```

```
footer {  
  background-color:  
var(--main-color);  
}
```

CSS custom properties

Sass

```
$main-color: orange;
```

CSS

```
:root {  
  --main-color: orange;  
}
```

Define a variable (Sass) or custom property (CSS)

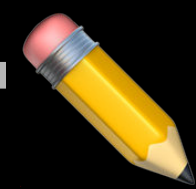
Sass

```
footer {  
  background-color: $main-color;  
}
```

CSS

```
footer {  
  background-color: var(--main-color);  
}
```

Use a variable (Sass) or custom property (CSS)

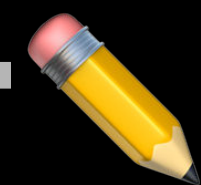


SIDE NOTE

Why the `--` & `var()` syntax? Why not `$` as in Sass or Less?

Tab Atkins, author of W3C's CSS Custom Properties: “If we use `$foo` for variables, we'll be unable to use it for future ‘variable-like’ things.”

Please explain further...



SIDE NOTE

“For example, if we do define an alternate form that are more SASS-like (can be used anywhere, but are global; more ‘macros’ than ‘variables’) we’d have to use some other glyph [besides `$`] for them [if we already used `$`]. That’s suboptimal.

More specifically, if we ever do some sort of ‘variables’ in selectors, we must use a compact form like `$foo` or something.

...As another example, ... a proper macro system for CSS; in other words, exactly what SASS’s variables currently do. I assert that we’d definitely want a nice short ‘`$foo`’ or similar syntax for this as well.”

Like Sass, a custom property can derive its value from another custom property

No math operators yet, though

HTML

4

</p>

5

<blockquote>

6

<p>

7

Lorem ipsum, dolor sit amet consectetur
adipisicing elit. Vero minima repudiandae
magni perspiciatis blanditiis beatae
voluptate ducimus, dolor nulla non, ex
exercitationem officiis. Eveniet est ipsum
iste sint rem accusantium.

8

</p>

9

<p>

10

CSS

1

:root {

2

--default-margin: 1rem;

3

--bq-margin: var(--default-margin) 3rem;

4

--bq-padding: .1rem var(--default-margin);

5

}

6

7

blockquote {

8

margin: var(--bq-margin);

9

padding: var(--bq-padding);

10

}

JS

Custom Properties

Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatum necessitatibus ipsum vero in unde impedit veniam laborum explicabo sit est dolorum reiciendis sint, id aliquid quae cumque debitis nihil odio.

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Vero minima repudiandae magni perspiciatis blanditiis beatae voluptate ducimus, dolor nulla non, ex exercitationem officiis. Eveniet est ipsum iste sint rem accusantium.

Pariatur assumenda ab soluta et libero ad suscipit molestias, rem expedita provident minus accusamus maiores quae, obcaecati voluptas praesentium corporis fugit totam

Can't use a variable as part of a value, so this is not allowed:

```
.foo {  
  --gap: 20;  
  margin-top: var(--gap)px;  
}
```

`calc()` to the rescue!

```
.foo {  
  --gap: 20;  
  margin-top: calc(var(--gap) * 1px);  
}
```

`var()` supports *fallback values*, in case a custom property isn't specified:

```
background-color: var(--main-color, yellow,  
cornsilk);
```

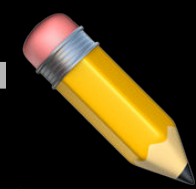
```
font-family: var(--body-fonts, Helvetica,  
Arial, sans-serif);
```

```
padding: var(--default-padding, 1rem 1.5rem);
```


Components can have fallbacks in case the component's container doesn't specify a value

```
/* Component styles */
.component .alert {
  background-color: var(--bg-color, #ccc);
  color: var(--alert-color, red);
}

/* Page/App styles */
.component {
  --bg-color: #bbb;
  /* No --alert-color set, so red used by component */
}
```



SIDE NOTE

CSS properties are *not* case sensitive, so `font:` & `Font:` are the same

Custom properties *are* case-sensitive, so `--foo:` & `--Foo:` are different

Advantages of CSS Custom Properties

No pre-processor required, since support is built in


```
:root {  
  --color: yellow;  
}
```

```
aside {  
  --color: orange;  
}
```

```
aside > h2 {  
  color: var(--color);  
}
```

CSS custom variables scoping



PRO TIP

A lot of developers use `:root` for global variables & then use scoped variables as needed

Why `:root`?

`:root` means your CSS can be used with SVG & XML

With Sass, the variables are static: you can't change values once the CSS has been rendered, or inside media queries

With custom properties, when the values change (e.g., via media queries or JavaScript), the browser repaints

```
:root {  
  --base-margin: 20px;  
}
```

```
@media (max-width: 640px) {  
  :root {--base-margin: 10px;}  
}
```

```
div {  
  margin: var(--base-margin);  
}
```


Bonus: JavaScript can now change variables without having to write to the `style` attribute on elements

In fact, while not valid as CSS, this is a valid custom property that could be manipulated by JavaScript:

```
--foo: if(x > 5) this.width = 10;
```

CSS Custom Properties (CSS Variables) Sample

Available in [Chrome 49+](#) | [View on GitHub](#) | [Browse Samples](#)

Background

CSS custom properties allow you to store and retrieve values from properties you define yourself.

They follow the same rules as other CSS properties, so you're able to define and use them at multiple levels, following standard CSS cascading and specificity rules.

Combining custom variables with `calc()`, in particular, allows for some very interesting possibilities.

Live Output

CSS Variable Demo

Spacing unit:

Columns:

Margins:

Red

Click the buttons on the cards to set the

Pink

The colors on the cards are not affected

Purple

Use the controls above to adjust some

💡 PRO TIP

Custom properties can really be handy for i18n

```
:root,  
:root:lang(en) {--external-link: "external  
link";}
```

```
:root:lang(fr) {--external-link: "lien  
externe";}
```

```
a[href^="http"]::after {  
  content: " (" var(--external-link) ")"  
}
```

					iOS		
--*	—	16	31	9.1	9.3	49	49
var()	—	16	31	9.1	9.3	49	49

Works with ~94% of global browsers (July 2020)

Thank you!

scott@granneman.com

www.granneman.com

ChainsawOnATireSwing.com

@scottgranneman

jans@websanity.com

websanity.com

CSS Preprocessors

Extending CSS with Sass to Provide Extra Functionality

R. Scott Granneman & Jans Carton

© 2016 R. Scott Granneman

Last updated 2020-07-13

You are free to use this work, with certain restrictions.

For full licensing information, please see the last slide/page.

Changelog

2020-07-13 1.8: Added *&* as *Sass Parent Selector* to *Nesting*, with examples & details; added Jans' *gulpfile.js*; minor formatting; added section on *Scope*; dumped darken & lighten functions, as they are unreliable; minor fixes & corrections

2020-02-15 1.7: Completely re-did & expanded section on custom properties & section introducing Sass; changed SCSS to Sass where appropriate; added note re: using HSL with Darken & Lighten Functions

Changelog

2018-06-22 1.6: Added VSCode under Tools; added screenshots of Jans' Bootstrap 3 breakpoints `@mixin`; minor changes to Compound selectors; minor fixes

2018-01-31 1.5: Added Task Runners & CLI Task Runners sections under Tools; moved CodeKit, Prepros, & Koala into new GUI Task Runners section under Tools; added example to Comments; ~~TODO CSS Variables~~; TODO Sass Variables Scope

Changelog

2018-01-30 1.4: Fixed minor formatting issues; better examples for nesting to create compound selectors; minor wording changes; added Autoprefixer section

2018-01-30 1.3: Improved Background section with better pre-processor info; applied Granneman 1.4 theme; fixed formatting issues; added Box Styling example to Mixins; added detail in Nested Properties; touched up `@import` Directive & Comments; changed Sass to SCSS where appropriate; added many screenshots of CodeKit, Prepros, & Koala

Changelog

2017-10-19 1.2: Updated Markdown examples; removed Mou & added MacDown; SCSS is winning; added Pro Tip re: editing CSS

2017-01-31 1.1: Updated theme & fixed minor formatting issues; added more examples & explanation to Mixins; added `@import`; fleshed out Brackets config files a *lot*; updated CodeKit home page image; added OS support info to Prepros

2016-01-19 1.0: Created slides

Licensing of this work

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>.

You are free to:

- » *Share* — copy and redistribute the material in any medium or format
- » *Adapt* — remix, transform, and build upon the material for any purpose, even commercially

Under the following terms:

Attribution. You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. Give credit to:

Scott Granneman • www.granneman.com • scott@granneman.com

Share Alike. If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions. You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Questions? Email scott@granneman.com