

LESS IS MORE

2012-01-12

CSS NITE LP, DISK 26

HIROKI TANI



HIROKI TANI

WEB DEVELOPER

CYBERAGENT, INC.



LESSの過去と現在

他のCSSプリプロセッサとの違い

なぜLESSを選ぶのか

CSSプリプロセッサを控えめに使う





cloudhead.io

art code tweets



.SASS (THE INDENTED)

```
$color: #4D926F
```

```
@mixin rounded-corners ($radius: 5px)
```

```
  -webkit-border-radius: $radius
```

```
  -moz-border-radius: $radius
```

```
  border-radius: $radius
```

```
#header
```

```
  color: $color
```

```
  @include rounded-corners
```

```
#footer
```

```
  @include rounded-corners(10px)
```

.LESS

```
@color: #4D926F;

.rounded-corners (@radius: 5px) {
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
  border-radius: @radius;
}

#header {
  color: @color;
  .rounded-corners;
}

#footer {
  .rounded-corners(10px);
}
```


.SCSS (SASSY CSS)

```
$color: #4D926F;
```

```
@mixin rounded-corners ($radius: 5px) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  border-radius: $radius;  
}
```

```
#header {  
  color: $color;  
  @include rounded-corners;  
}
```

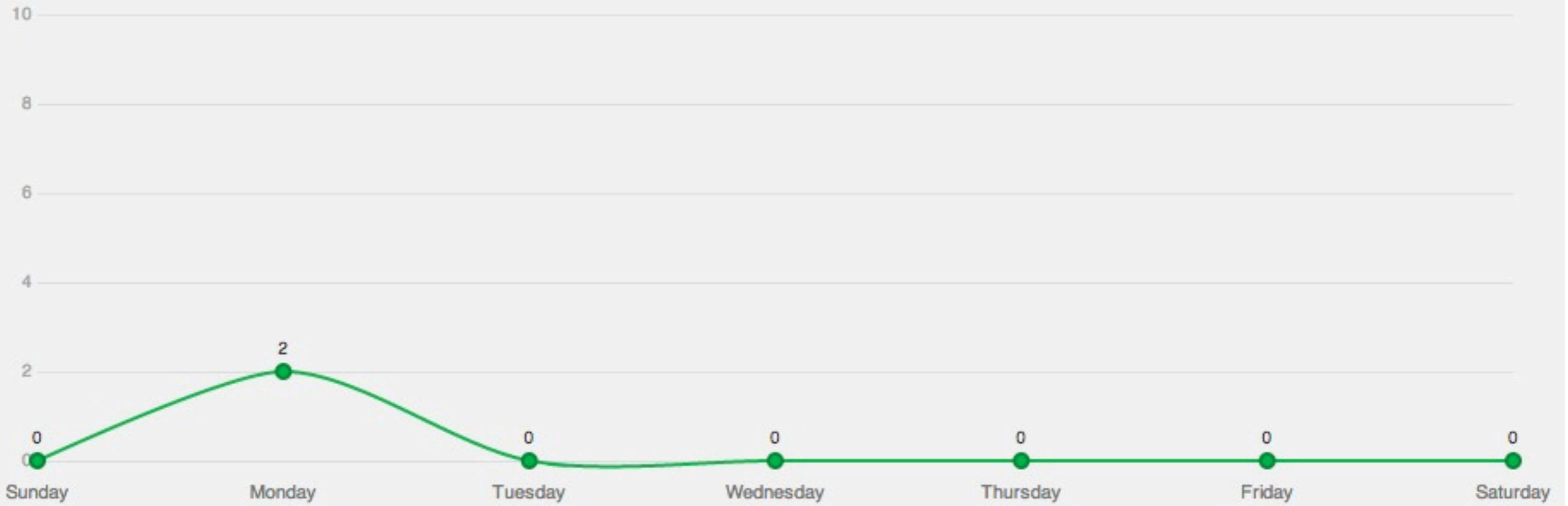
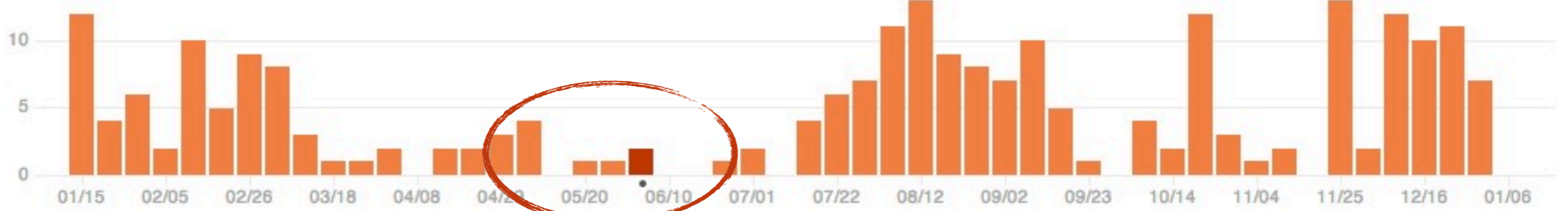
```
#footer {  
  @include rounded-corners(10px);  
}
```



JAVASCRIPT



52 WEEK HISTORY



52 WEEK HISTORY



86

PULL REQUESTS

394

ISSUES





@MATTHEWDL

LESS大丈夫？

Sassはめっちゃ活発やで！

<https://github.com/cloudhead/less.js/issues/867>



@CLOUDHEAD

今LESSやってる暇無いねん

<https://github.com/cloudhead/less.js/issues/867>

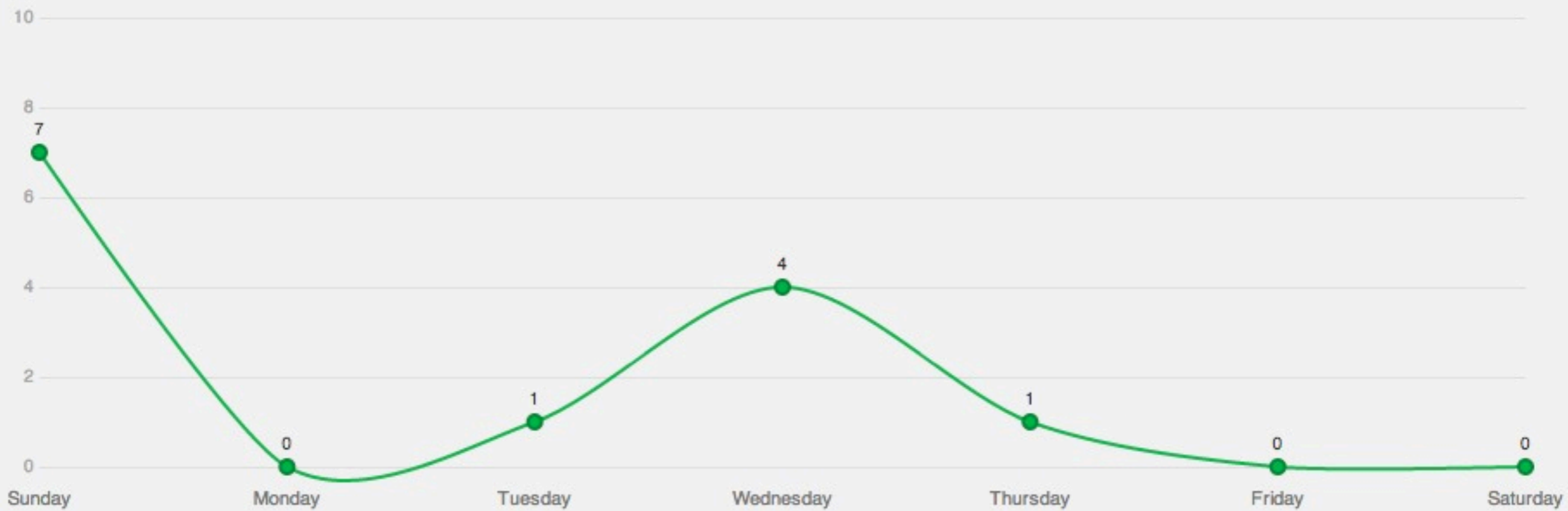


@AGATRONIC

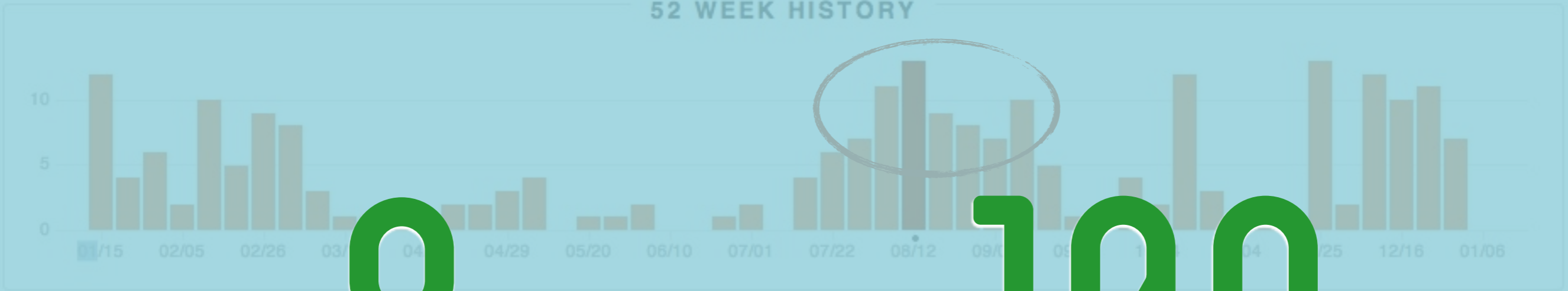
手伝うで！

<https://github.com/cloudhead/less.js/issues/867>

52 WEEK HISTORY



52 WEEK HISTORY



8

109

PULL REQUESTS

ISSUES





2012-03-10

1.3.0

2012-10-18

1.3.1

2012-12-28

1.3.2

2012-12-30

1.3.3

2013-XX-XX

1.4.0?



CLIENT SIDE
OR
SERVER SIDE

USAGE (CLIENT SIDE)

```
<head>
<link rel="stylesheet/less" type="text/css"
href="styles.less" />
<script src="less.js" type="text/javascript"></script>
<script type="text/javascript">
  less.watch();
</script>
</head>
```



[http://localhost:8000/demo/clientside/#!watch](http://localhost:8000/demo/clientside/#!<u>watch</u>)

USAGE (SERVER SIDE)

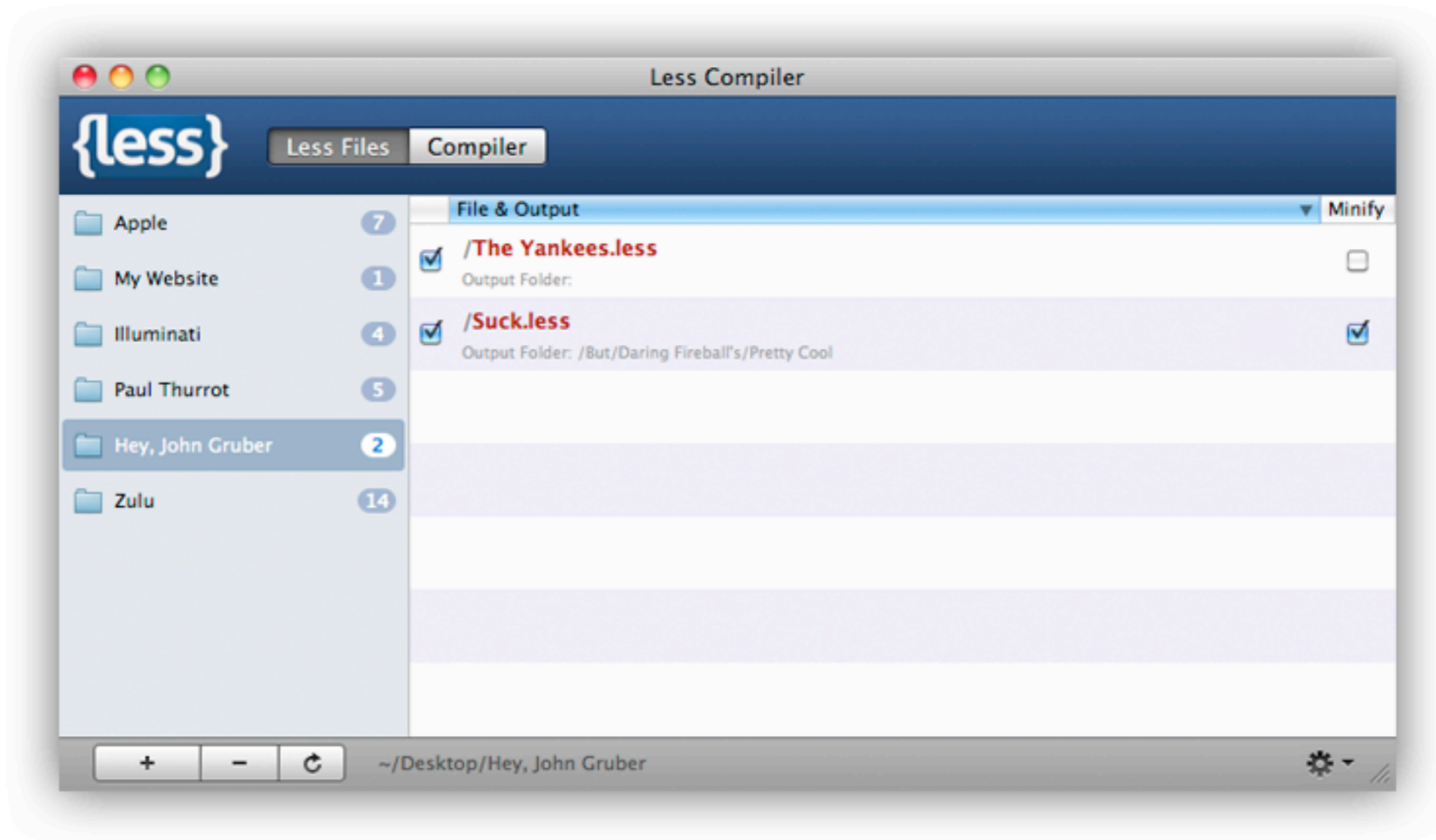
```
$ npm install -g less
```

```
$ lessc styles.less > styles.css
```

```
$ lessc styles.less > styles.css --compress
```

```
$ lessc styles.less > styles.css --yui-compress
```

```
$ lessc styles.less > styles.css --watch
```



The only thing easier is making fun of Internet Explorer

LESS.app

<http://incident57.com/less/>

{Less} extends CSS with variables, nested rules, operators and more.

If you're still building websites without it, you're an idiot. This app

Download



SIMPLIFYING WEBSITE PROGRAMMING

SimpLESS compiles LESS into beautiful CSS



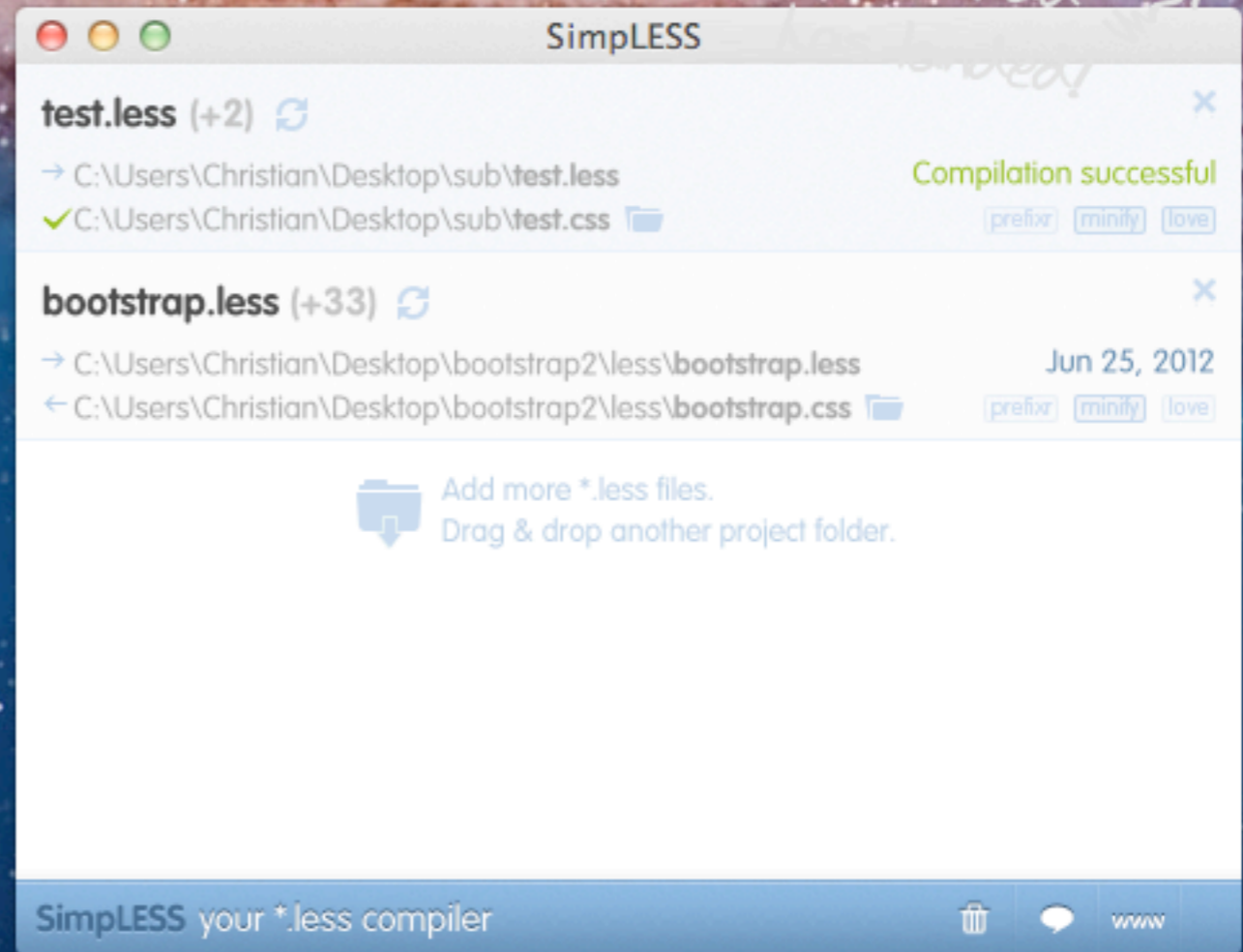
Short intro

It's easy to understand why thousands of developers use SimpLESS for their web projects. SimpLESS is your easy-to-use LESS CSS compiler.

Available for free for mac and pc.

Download SimpLESS now!

IT'S FREE FOR EVERY PLATTFORM

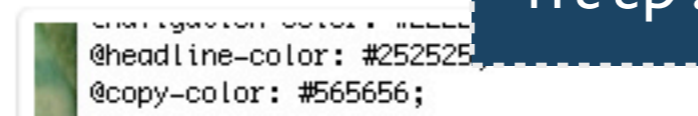


Drag & Drop

Write

SimpLESS

<http://wearekiss.com/simpless>



.SCSS

```
$color: #4D926F;
```

```
@mixin rounded-corners ($radius: 5px) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  border-radius: $radius;  
}
```

```
#header {  
  color: $color;  
  @include rounded-corners;  
  .logo {  
    float: left;  
  }  
}
```

.LESS

```
@color: #4D926F;

.rounded-corners (@radius: 5px) {
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
  border-radius: @radius;
}

#header {
  color: @color;
  .rounded-corners (10px);
  .logo {
    float: left;
  }
}
```

.SCSS (EXTEND)

```
.error {  
  border: 1px #FF0000;  
  background-color: #FFDDDD;  
}
```

```
.warning {  
  @extend .error;  
  border-width: 3px;  
}
```

.CSS (EXTEND)

```
.error, .warning {  
  border: 1px #FF0000;  
  background-color: #FFDDDD;  
}
```

```
.warning {  
  border-width: 3px;  
}
```



@HOKACCHA

Add Sass like extend.

<https://github.com/cloudhead/less.js/pull/509>

.LESS (EXTEND)

```
.error {  
  border: 1px #FF0000;  
  background-color: #FFDDDD;  
}
```

```
.warning {  
  + .error;  
  border-width: 3px;  
}
```

.LESS (EXTEND)

in 1.4.0?

```
.error {  
  border: 1px #FF0000;  
  background-color: #FFDDDD;  
}
```

```
.warning {  
  &:extend(.error);  
  border-width: 3px;  
}
```

.LESS (FUNCTIONS)

```
ceil(@number); // 小数点切り上げ
```

```
floor(@number); // 小数点切り捨て
```

```
percentage(@number); // 数値をパーセント化
```

```
lighten(@color, 10%); // 色を10%明るくする
```

```
darken(@color, 10%); // 色を10%暗くする
```

```
fadein(@color, 10%); // 色の不透明度を10%上げる
```

```
fadeout(@color, 10%); // 色の不透明度を10%下げる
```

```
multiply(@color1, @color2); // 乗算効果
```

```
screen(@color1, @color2); // スクリーン効果
```

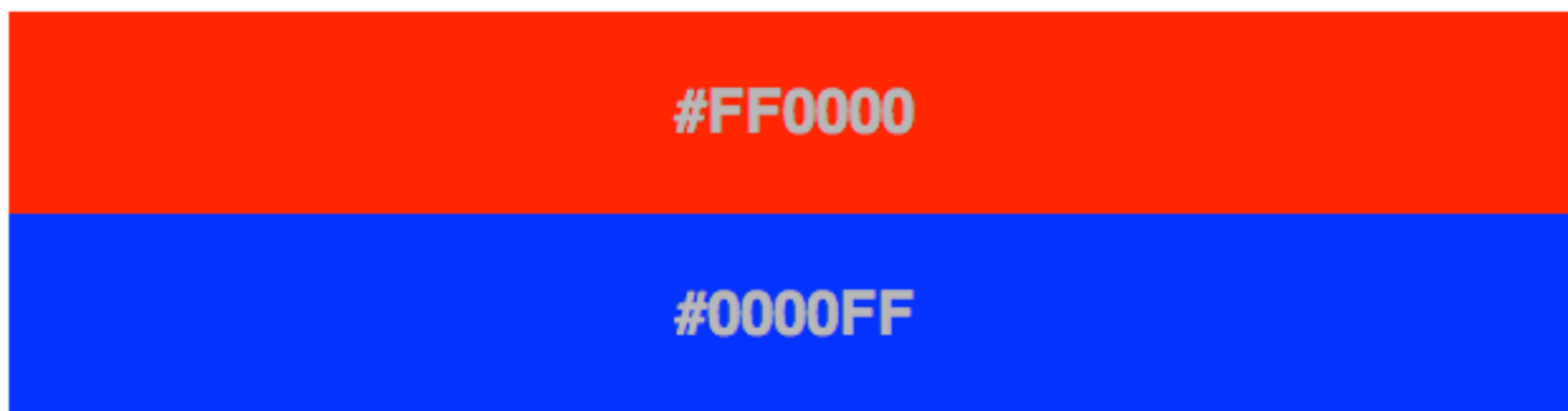
```
overlay(@color1, @color2); // オーバーレイ効果
```


元の色



- @colorA
- @colorB

それぞれに値を入れておく。



```
<div class="palette colorA">#FF0000</div>  
<div class="palette colorB">#0000FF</div>
```

```
@colorA: #FF0000;  
@colorB: #0000FF;  
  
.colorA {  
    background-color: @colorA;  
}  
.colorB {  
    background-color: @colorB;  
}  
.palette {  
    width: 100%;  
    height: 80px;  
    line-height: 80px;  
    color: #AAA;  
    text-align: center;  
    font-size: 24px;  
    font-weight: 700;  
    font-family: Helvetica,Arial;  
}
```

multiply



乗算

```
.multiply {  
    background-color:  
    multiply(@colorA,@colorB);  
}
```

LESS1.3.1からの新しい色関数のデモ

<http://inkdesign.jp/demo/less-new-colors/less.html>

.LESS (PATTERN-MATCHING)

```
// Mixin
.switch (dark, @color) {
  color: darken(@color, 10%);
}
.switch (light, @color) {
  color: lighten(@color, 10%);
}
.switch (@_, @color) {
  display: block;
}

// Load mixin
.button {
  @switch: dark;
  .switch(@switch, #888888);
}
```

```
// Output CSS
.button {
  color: #6f6f6f;
  display: block;
}
```

.LESS (GUARDS)

```
// Mixin
.checker (@color) when
(lightness(@color) >= 50%) {
  background-color: black;
}
.checker (@color) when
(lightness(@color) < 50%) {
  background-color: white;
}
.checker (@color) {
  color: @color;
}
// Load mixin
.button {
  @color: #FF0000;
  .checker(@color);
}
```

```
// Output CSS
.button {
  background-color: black;
  color: #FF0000;
}
```

Fork me on GitHub



The dynamic stylesheet language.

LESSは変数、ミックスイン、演算、そして関数のような動的な処理をCSSに追加拡張できます。クライアントサイド(IE6+、Webkit、Firefox)、またnode.js、Rhinoを利用してサーバーサイドでも動作します。

Download less.js

バージョン 1.3.1

LESSを書いてみよう:

```
@base: #f938ab;

.box-shadow(@style, @c) when (iscolor(@c)) {
  box-shadow: @style @c;
  -webkit-box-shadow: @style @c;
  -moz-box-shadow: @style @c;
}

.box-shadow(@style, @alpha: 50%) when (isnumber(@alpha)) {
  .box-shadow(@style, rgba(0, 0, 0, @alpha));
}

.box {
  color: saturate(@base, 5%);
  border-color: lighten(@base, 30%);
  div { .box-shadow(0 0 5px, 30%) }
```

less.jsをスタイルに追加する:

```
<link rel="stylesheet/less" type="text/css" href="styles.less">
<script src="less.js" type="text/javascript"></script>
```

概要 使用法 シンタックス ソース アバウト

Follow

変数

変数は広範囲で利用している値を1ヶ所で定義することができ、その値をスタイルシート内で再利用することができます。これにより全体の変更を1行のコ

```
// LESS
@color: #4D926F;
```

LESSドキュメント日本語訳
<http://less-ja.studiomohawk.com/>

LESS Input

```
1 .transition(@transition) {
2   -webkit-transition: @transition;
3   -moz-transition: @transition;
4   -o-transition: @transition;
5   transition: @transition;
6 }
7 .opacity(@opacity) {
8   opacity: @opacity / 100;
9   filter: ~"alpha(opacity=@{opacity})";
10 }
11
12 a {
13   .transition(all 0.4s);
14   &:hover {
15     .opacity(70);
16   }
17 }
18
19 @theGoodThings: ~".food, .beer, .ladies, .sleep, .javascript";
20
21 @{theGoodThings} {
22   font-weight: bold;
23 }
```

Output CSS

```
a {
  -webkit-transition: all 0.4s;
  -moz-transition: all 0.4s;
  -o-transition: all 0.4s;
  transition: all 0.4s;
}
a:hover {
  opacity: 0.7;
  filter: alpha(opacity=70);
}
.food, .beer, .ladies, .sleep, .javascript {
  font-weight: bold;
}
```

LESS2CSS

<http://less2css.org/>

WHY LESS ?

Extending Bootstrap

Extend Bootstrap to take advantage of included styles and components, as well as LESS variables and mixins.

Built with LESS >

Compiling Bootstrap >

Use as static assets >

Built with LESS

Bootstrap is made with LESS at its core, a dynamic stylesheet language created by our good friend, [Alexis Sellier](#). It makes developing systems-based CSS faster, easier, and more fun.



Why LESS?

One of Bootstrap's creators wrote a quick [blog post about this](#), summarized here:

- Bootstrap compiles faster ~6x faster with Less compared to Sass

Extending Bootstrap

<http://twitter.github.com/bootstrap/extend.html>

As an extension of CSS, LESS includes variables, mixins for reusable snippets of code, operations for simple



Why Less?

March 8th 2012

Since first open sourcing bootstrap one question Mark and I have heard a *lot* is: “why Less?” If you’re not familiar, Less is a dynamic stylesheet language meant to extend CSS with things like variables, mixins, operations and functions.

The question is a really good one and something we struggle with a lot ourselves. So much so, Mark asked me to write a little bit about some of the conversations we’ve had around our choice and why we’ve made it.

Going forward, I’m going to try to use this as a faq/wiki for common questions around our decision to use Less. Feel free to ask Mark or I more about any of these questions and I’ll update the post here. Thanks!

Why not Sass?

Sass is great. It ships with rails

Sass because:

- **Bootstrap compiles ~6x f**
measured the time it took to

port). Our Less version compiled in a little under 90 ms on average, while the Sass version compiled around

Why LESS?

<http://www.wordsbyf.at/2012/03/08/why-less/>

WHY NOT SASS ?

WHY NOT SASS?

“BOOTSTRAP COMPILES ~ 6X FASTER WITH LESS THAN SASS”

Sassよりコンパイルが6倍速いから

“LESS IS IMPLEMENTED IN JAVASCRIPT”

JavaScriptで実装されているから

“LESS IS SIMPLE”

シンプルだから

Bootstrap

Sleek, intuitive, and powerful front-end framework for
assembling simpler web development.

540 ms

BOOTSTRAP-SASS

90 ms

BOOTSTRAP

Star 42,661

Fork 11,325

Follow @twbootstrap 46.6K followers

Tweet 17.1K

Introducing Bootstrap.

WHY NOT SASS?

“BOOTSTRAP COMPILES ~ 6X FASTER WITH LESS THAN SASS”

Sassよりコンパイルが6倍速いから

“LESS IS IMPLEMENTED IN JAVASCRIPT”

JavaScriptで実装されているから

“LESS IS SIMPLE”

シンプルだから

WHY NOT SASS?

“BOOTSTRAP COMPILES ~ 6X FASTER WITH LESS THAN SASS”

Sassよりコンパイルが6倍速いから

“LESS IS IMPLEMENTED IN JAVASCRIPT”

JavaScriptで実装されているから

“LESS IS SIMPLE”

シンプルだから

.LESS

```
@color: #4D926F;

.rounded-corners (@radius: 5px) {
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
  -ms-border-radius: @radius;
  -o-border-radius: @radius;
  border-radius: @radius;
}

#header {
  color: @color;
  .rounded-corners;
}

#footer {
  .rounded-corners(10px);
}
```

“LESS” PREPROCESSOR

AVOID OVERLY SPECIFIC SELECTORS

過剰なネストは避ける


```
.content {
  .main {
    .entries {
      .entry {
        width: 960px;
        .entry-header {
          h1 {
            font-size: 16px;
          }
          .pubdate {
            font-size: 12px;
          }
          .tags {
            > li {
              display: inline;
            }
          }
        }
        .entry-body {
          p {
            font-size: 14px;
            small {
              font-size: 12px;
            }
          }
        }
      }
    }
  }
}
```

```
.content .main .entries .entry {
  width: 960px;
}
.content .main .entries .entry .entry-header h1 {
  font-size: 16px;
}
.content .main .entries .entry .entry-header .pubdate {
  font-size: 12px;
}
.content .main .entries .entry .entry-header .tags > li {
  display: inline;
}
.content .main .entries .entry .entry-body p {
  font-size: 14px;
}
.content .main .entries .entry .entry-body p small {
  font-size: 12px;
}
.content .main .entries .entry .entry-body .features .feature-content {
  background: url(img/features.png);
}
.content .main .entries .entry .entry-body figure {
  float: left;
}
.content .main .entries .entry .entry-body figure figcaption {
  text-align: center;
}
.content .main .entries .entry .entry-body figure img {
  border: 3px solid #FFFFFF;
}
```

AVOID TOO COMPLEX MIXINS

複雑すぎるMixinsは避ける

```
@sprite-image: '/example.png';
```

```
@sprite-grid: 50px;
```

```
.sprite(@x, @y, @sprite-image:@sprite-image, @sprite-grid:@sprite-grid) {  
  .sprite-image(@sprite-image);  
  .sprite-pos(@x, @y, @sprite-grid);  
}
```

```
.sprite-image(@sprite-image:@sprite-image) {  
  background-image: url(@sprite-image);  
  background-repeat: no-repeat;  
}
```

```
.sprite-pos(@x, @y, @sprite-grid:@sprite-grid) {  
  background-position: -(@x*@sprite-grid*1px) -(@y*@sprite-grid*1px);  
}
```

AVOID BLOAT CSS

Mixinsの多用は避ける

```
.ellipsis {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```

```
.username {  
  .ellipsis;  
  font-size: 12px;  
}
```

```
.nav-list {  
  .ellipsis;  
  display: block;  
  background: #EEEEEE;  
}
```

```
.news-item {  
  .ellipsis;  
  padding: 10px;  
}
```

```
.ellipsis {
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
}

.username {
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  font-size: 12px;
}

.nav-list {
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  display: block;
  background: #EEEEEE;
}

.news-item {
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  padding: 10px;
}
```

AVOID TOO MUCH UGLY SELECTORS

Extendsの多用は避ける


```
.clearfix {
  *zoom: 1; /* for IE 6/7 only */
  &:before,
  &:after {
    content: " ";
    display: table;
  }
  &:after {
    clear: both;
  }
}
.header {
  @extend .clearfix;
}
.container {
  @extend .clearfix;
}
.globalnav {
  @extend .clearfix;
}
.features {
  @extend .clearfix;
}
.search {
  @extend .clearfix;
}
.footer {
  @extend .clearfix;
}
```

```
.clearfix, .header, .container, .globalnav, .features,  
.search, .footer {  
  *zoom: 1;  
  /* for IE 6/7 only */  
}
```

```
.clearfix:before, .header:before, .container:before,  
.globalnav:before, .features:before, .search:before,  
.footer:before, .clearfix:after, .header:after,  
.container:after, .globalnav:after, .features:after,  
.search:after, .footer:after {  
  content: " ";  
  display: table;  
}
```

```
.clearfix:after, .header:after, .container:after,  
.globalnav:after, .features:after, .search:after,  
.footer:after {  
  clear: both;  
}
```

SYMPTOMS

Styles on a webpage are missing or look incorrect when the page loads in the versions of Microsoft Internet Explorer that are listed in the "Applies to" section.

Note This problem can occur whether the webpage uses an inline style sheet or points to a cascading style sheet.

You may also receive the following error message:

The page you are looking for might have been moved or had its name changed.

[↑ Back to the top](#) | [Give Feedback](#)

CAUSE

This problem occurs because the following conditions are true in Internet Explorer:

- All style tags after the first 31 style tags are not applied.
- All style rules after the first 4,095 rules are not applied.
- On pages that uses the @import rule to continuously import external style sheets that import other style sheets, style sheets that are more than three levels deep are ignored.

4,095

rules are not applied.

[↑ Back to the top](#) | [Give Feedback](#)

MORE INFORMATION

A network monitor tool may indicate that a TCP reset occurs when Internet Explorer is receiving the page that is experiencing the problem. Internet Explorer then generates another POST request if the original request was a POST request. Or, Internet Explorer may send a GET request instead.

This article describes the problem that occurs when you use Internet Explorer to view a webpage that uses CSS styles.

A webpage that uses CSS styles does not render correctly in Internet Explorer

<http://support.microsoft.com/kb/262161/en-us>

If

Wh

The page cannot be displayed



Because after all these years, IE **still** sucks.

With modernizer & CSS3PIE you thought you had this one all figured out.

Think again. CSS selector limit. Boom.

So, what is it?

Bless provides an elegant solution to a lesser-known bug in Internet Explorer which causes CSS to be completely ignored. It runs server-side and on your local machine with [Node.js](#)

Say what?

IE versions 6, 7, 8 & 9 all have a limit on the number of selectors allowed in a single CSS file. Once the limit is reached, IE silently fails and just ignores any further CSS in the file leaving parts of your site totally unstyled.

File size, number of lines or amount of whitespace are irrelevant. Minifying CSS will not help you here.

I've never come across this before...

Chances are, so far you've worked on smaller scale sites and/or separated out CSS across multiple files. Once a project reaches a certain size however, there's no going back.



Bless for Mac

Basic drag and drop app

Bless.js

[View the project on GitHub](#)

Example.css

[Download and try it out!](#)

bless

<http://blesscss.com/>

THINK ABOUT YOUR OUTPUT CSS

コンパイルされたCSSのことを考える

THINK ABOUT YOUR OUTPUT CSS

CSSのファイルサイズ

THINK ABOUT YOUR OUTPUT CSS

~~CSSのファイルサイズ~~

THINK ABOUT EDITING OUTPUT CSS

コンパイル後のCSSを

編集することを考える

SINGLE-CLASS

VS

MULTI-CLASS



...things by Nicolas Gallagher

Blog [Archive](#) [Lab](#) [About](#)

MARCH 15, 2012

About HTML semantics and front-end architecture

A collection of thoughts, experiences, ideas that I like, and ideas that I have been experimenting with over the last year. It covers HTML semantics, components and approaches to front-end architecture, class naming patterns, and HTTP compression.

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.

About HTML semantics and front-end architecture

<http://nicolasgallagher.com/about-html-semantics-front-end-architecture/>

邦訳 : <http://enja.studiomohawk.com/2012/03/20/about-html-semantics-and-front-end-architecture/>

semantics is the study of the relationships between signs and symbols and

and the way they are used to represent things, ideas, or feelings.

Default

Primary

Success

Warning

Danger

Default

Primary

Success

Warning

Danger

Default

Primary

Success

Warning

Danger

```
<nav>
  <ul>
    <li><a href="#" class="btn">Default</a></li>
    <li><a href="#" class="btn-primary">Primary</a></li>
    <li><a href="#" class="btn-success">Success</a></li>
    <li><a href="#" class="btn-warning">Warning</a></li>
    <li><a href="#" class="btn-danger">Danger</a></li>
  </ul>
  <ul>
    <li><a href="#" class="btn-large">Default</a></li>
    <li><a href="#" class="btn-primary-large">Primary</a></li>
    <li><a href="#" class="btn-success-large">Success</a></li>
    <li><a href="#" class="btn-warning-large">Warning</a></li>
    <li><a href="#" class="btn-danger-large">Danger</a></li>
  </ul>
  <ul>
    <li><a href="#" class="btn-small">Default</a></li>
    <li><a href="#" class="btn-primary-small">Primary</a></li>
    <li><a href="#" class="btn-success-small">Success</a></li>
    <li><a href="#" class="btn-warning-small">Warning</a></li>
    <li><a href="#" class="btn-danger-small">Danger</a></li>
  </ul>
</nav>
```

```
.btn {
  display: inline-block;
  margin-bottom: 0;
  padding: 4px 12px;
  border: 1px solid;
  border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
  border-bottom-color: #A2A2A2;
  border-radius: 4px;
  background-color: whiteSmoke;
  background-image: linear-gradient(to bottom, white, #E6E6E6);
  background-repeat: repeat-x;
  box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
  color: #333333;
  vertical-align: middle;
  text-align: center;
  text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
  font-size: 14px;
  line-height: 20px;
  cursor: pointer;
}
```

```
.btn-primary {
  display: inline-block;
  margin-bottom: 0;
  padding: 4px 12px;
  border: 1px solid;
  border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
  border-radius: 4px;
  background-color: #006dcc;
  background-image: linear-gradient(to bottom, #0088cc, #0044cc);
  background-repeat: repeat-x;
  box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
  color: #ffffff;
  vertical-align: middle;
  text-align: center;
  ...
}
```

```
.btn {  
  display: inline-block;  
  margin-bottom: 0;  
  padding: 4px 12px;  
  border: 1px solid;  
  border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);  
  ...  
}
```

```
.btn-small {  
  padding: 2px 10px;  
  font-size: 11.9px;  
  border-radius: 3px;  
}
```

```
.btn-primary {  
  @extend .btn;  
  @extend .btn-small;  
  border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);  
  background-color: #006dcc;  
  background-image: linear-gradient(to bottom, #0088cc, #0044cc);  
  ...  
}
```

```
.btn {
  background-color: gray;
}
.btn-primary {
  @extend .btn;
  background-color: blue;
}
.btn-success {
  @extend .btn;
  background-color: green;
}
.btn-warning {
  @extend .btn;
  background-color: yellow;
}
.btn-danger {
  @extend .btn;
  background-color: red;
}

.btn-small {
  @extend .btn;
  font-size: small;
}
.btn-small-primary {
  @extend .btn;
  @extend .btn-small;
  @extend .btn-primary;
}
.btn-small-success {
  @extend .btn;
  @extend .btn-small;
  @extend .btn-success;
}
.btn-small-warning {
  ...
```

```
.btn { ... }  
.btn-primary { ... }  
.btn-success { ... }  
.btn-warning { ... }  
.btn-danger { ... }  
  
.btn-small { ... }  
.btn-small-primary { ... }  
.btn-small-success { ... }  
.btn-small-warning { ... }  
.btn-small-danger { ... }  
  
.btn-large { ... }  
.btn-large-primary { ... }  
.btn-large-success { ... }  
.btn-large-warning { ... }  
.btn-large-danger { ... }
```



```
<nav>
  <ul>
    <li><a href="#" class="btn">Default</a></li>
    <li><a href="#" class="btn btn-primary">Primary</a></li>
    <li><a href="#" class="btn btn-success">Success</a></li>
    <li><a href="#" class="btn btn-warning">Warning</a></li>
    <li><a href="#" class="btn btn-danger">Danger</a></li>
  </ul>
  <ul>
    <li><a href="#" class="btn btn-large">Default</a></li>
    <li><a href="#" class="btn btn-large btn-primary">Primary</a></li>
    <li><a href="#" class="btn btn-large btn-success">Success</a></li>
    <li><a href="#" class="btn btn-large btn-warning">Warning</a></li>
    <li><a href="#" class="btn btn-large btn-danger">Danger</a></li>
  </ul>
  <ul>
    <li><a href="#" class="btn btn-small">Default</a></li>
    <li><a href="#" class="btn btn-small btn-primary">Primary</a></li>
    <li><a href="#" class="btn btn-small btn-success">Success</a></li>
    <li><a href="#" class="btn btn-small btn-warning">Warning</a></li>
    <li><a href="#" class="btn btn-small btn-danger">Danger</a></li>
  </ul>
</nav>
```

```
.btn {  
  background-color: gray;  
}  
.btn-primary {  
  background-color: blue;  
}  
.btn-success {  
  background-color: green;  
}  
.btn-warning {  
  background-color: yellow;  
}  
.btn-danger {  
  background-color: red;  
}  
.btn-small {  
  font-size: small;  
}  
.btn-large {  
  font-size: large;  
}
```

```
.btn { ... }  
.btn-primary { ... }  
.btn-success { ... }  
.btn-warning { ... }  
.btn-danger { ... }
```

```
.btn-small { ... }
```

```
.btn-large { ... }
```

15 CLASSES

SINGLE-CLASS

```
.btn { ... }  
.btn-primary { ... }  
.btn-success { ... }  
.btn-warning { ... }  
.btn-danger { ... }  
  
.btn-small { ... }  
.btn-small-primary { ... }  
.btn-small-success { ... }  
.btn-small-warning { ... }  
.btn-small-danger { ... }  
  
.btn-large { ... }  
.btn-large-primary { ... }  
.btn-large-success { ... }  
.btn-large-warning { ... }  
.btn-large-danger { ... }
```

7 CLASSES

MULTI-CLASS

```
.btn { ... }  
.btn-primary { ... }  
.btn-success { ... }  
.btn-warning { ... }  
.btn-danger { ... }  
  
.btn-small { ... }  
  
.btn-large { ... }
```

```
%_btn {
  border: 1px solid;
}
%_btn-primary {
  @extend %_btn;
  border-color: blue;
  background-color: skyblue;
}
%_btn-success {
  @extend %_btn;
  border-color: green;
  background-color: limegreen;
}

.register {
  @extend %_btn-primary;
}
.save {
  @extend %_btn-success;
}
```

```
.register, .save {
  border: 1px solid;
}
.register {
  border-color: blue;
  background-color: skyblue;
}
.save {
  border-color: green;
  background-color: limegreen;
}
```

CSS PREPROCESSOR IS JUST A TOOL

ただのツール、使い方次第



“IT DEPENDS.”

時と場合による

Jeremy Keith - Caveat

<http://adactio.com/journal/4437/>

@CSSRADAR

ちゃんと

CSSを書け！



THANK YOU!

@HILOKI

INKDESIGN.JP