



# The dynamic stylesheet language

jsCafé vol.12

2013-08-04

Katsunori Tanaka

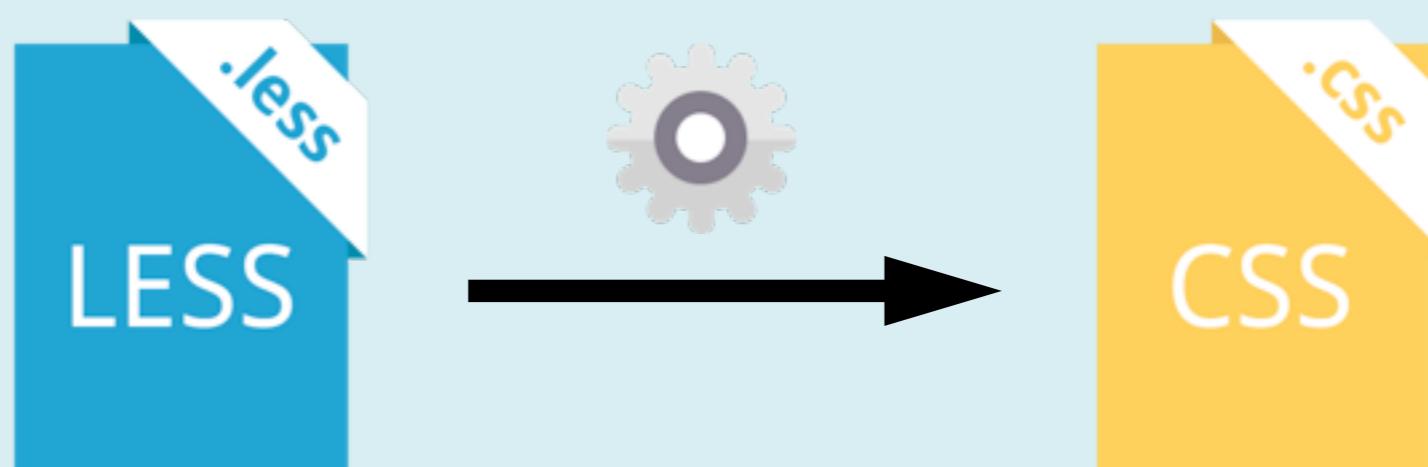
## {Outline}

1. CSS Preprocessorとは？
2. 環境構築
3. LESSの基本文法
4. ライブドリの紹介
5. LESS or SASS

# CSS Preprocessor とは？

## 導入する理由・メリット

CSSの機能を拡張したメタ言語。  
コンパイルをする事でCSSへ変換される。





## The dynamic stylesheet language.

LESS extends CSS with dynamic behavior such as variables, mixins, operations and functions.

LESS runs on both the server-side (with Node.js and Rhino) or client-side (modern browsers only).



[Download less.js](#)

version 1.4.1  
[changelog](#)

[overview](#)

[usage](#)

[language](#)

[function reference](#)

Write some LESS:

```
@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%) }
}
```

Compile to CSS:

```
npm install -g less
lessc styles.less styles.css
```

[Follow](#)

## 1.4.0

We have released 1.4.0. This includes new features such as `extends`, the `data-uri` function and more maths functions. See the [changelog](#) for a full list of changes.

# LESS

<http://lesscss.org/>

`@import-once` is removed and is now default behaviour for `@import`.  
`(~ ".myclass_{index}") { ... }` selector interpolation is deprecated, do this instead `.myclass_{index}`  
 .... This works in 1.3.1 onwards.

The browser version no longer bundles a version of es5-shim.js - the version we previously used was



# Sass.

{style with attitude}

```
$ gem install sass  
$ mv style.css style.scss  
$ sass --watch style.scss:style.css
```

[About](#)[Tutorial](#)[Documentation](#)[Blog](#)[Try Online](#)

**Latest Release:** Media Mark ([3.2.10](#))

[What's New?](#)

**Sass makes CSS fun again.** Sass is an extension of CSS3, adding [nested rules](#), [variables](#), [mixins](#), [selector inheritance](#), and [more](#). It's translated to well-formatted, standard CSS using the command line tool or a web-framework plugin.

Sass has two syntaxes. The most commonly used syntax is known as "SCSS" (for "Sassy CSS"), and is a superset of CSS3's syntax. This means that every valid CSS3 stylesheet is valid SCSS as well. SCSS files use the extension `.scss`.

The second, older syntax is known as [the indented syntax](#) (or just ".sass"). Inspired by [Haml](#)'s terseness, it's intended for people who prefer conciseness over similarity to CSS. Instead of brackets and semicolons, it uses the indentation of lines to specify blocks. Files in the indented syntax use the extension `.sass`.

While the documentation on this site is mostly in the SCSS syntax, both syntaxes are fully supported – there is no functional difference between them. Use the syntax you prefer.



[Download](#)

◆ [Editor Support](#)

◆ [Development](#)

# Sass

## Variables

<http://sass-lang.com>

Use the same color all over the place? Need to do some math with height and width and text size? Sass supports variables as well as basic math operations and [many useful functions](#).

[.SCSS](#)[.SASS](#)

## Nesting

Sass avoids repetition by nesting selectors within one another. The same thing works with properties.

[.SCSS](#)[.SASS](#)



Expressive, dynamic, robust CSS

## CSS needs a hero

```
body {  
  font: 12px Helvetica, Arial, sans-serif;  
}  
a.button {  
  -webkit-border-radius: 5px;  
  -moz-border-radius: 5px;  
  border-radius: 5px;  
}
```

# stylus

<http://learnboost.github.io/stylus/>

What if we could omit braces?

```
body  
font: 12px Helvetica, Arial, sans-serif;
```

Selectors  
Variables  
Interpolation  
Operators  
Mixins  
Functions  
Keyword Arguments  
Built-in Functions  
Rest Params  
Comments  
Conditionals  
Iteration  
@import  
@media  
@font-face  
@keyframes  
@extend  
url()  
CSS Literal  
CSS Style Syntax  
Char Escaping  
Executable  
Error Reporting  
Connect Middleware  
Introspection API  
JavaScript API  
CSS3 Extensions with Nib  
Try Stylus Online!

# CSS Preprocessorができる事

1. 変数
2. ミックスイン
3. 入れ子ルール
4. 関数と演算
5. セレクタ継承

## 導入するメリット

1. 作業工数軽減
2. クリエイティビティの促進
3. メンテナンシビリティの向上

# 環境構築

CUI · GUI · Client-Side

# CSS Preprocessorの環境構築とは？

1. コンパイラの導入
2. ファイルの監視

# CUI usage

```
$ npm install -g less  
  
$ lessc style.less > style.css  
$ lessc style.less > style.css --compress  
$ lessc style.less > style.css --yui-compress  
$ lessc style.less > style.css --strict-units=off  
$ lessc style.less > style.css --watch
```

Buy Us A Coffee!

because we could use one

Fork me on GitHub

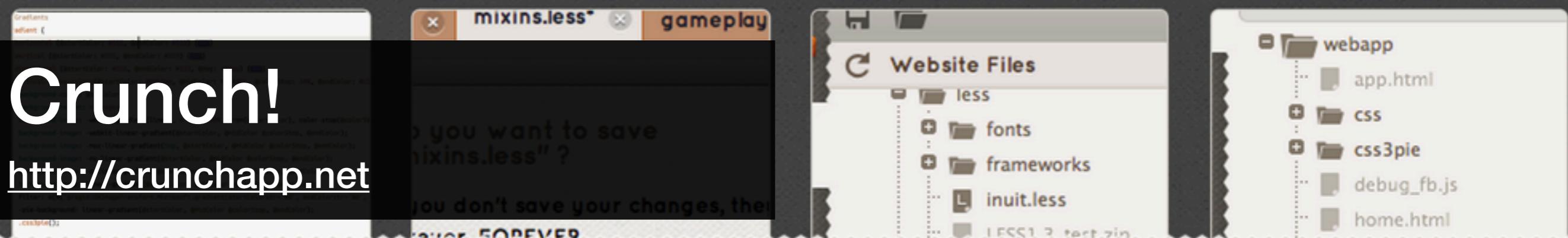


# Crunch!

The LESS editor and compiler that almost makes it too easy.



Download 1.6.4



# Crunch!

<http://crunchapp.net>

Crunch it and you're good to go

SIMPLIFYING WEBSITE PROGRAMMING

# SimpLESS compiles LESS into beautiful CSS



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

[Short intro](#)

Available for **free** for mac and pc.

[Download SimpLESS now!](#)

IT'S FREE FOR EVERY PLATTFORM

# SimpLESS

<http://wearekiss.com/simpless>

Drag & Drop

Write

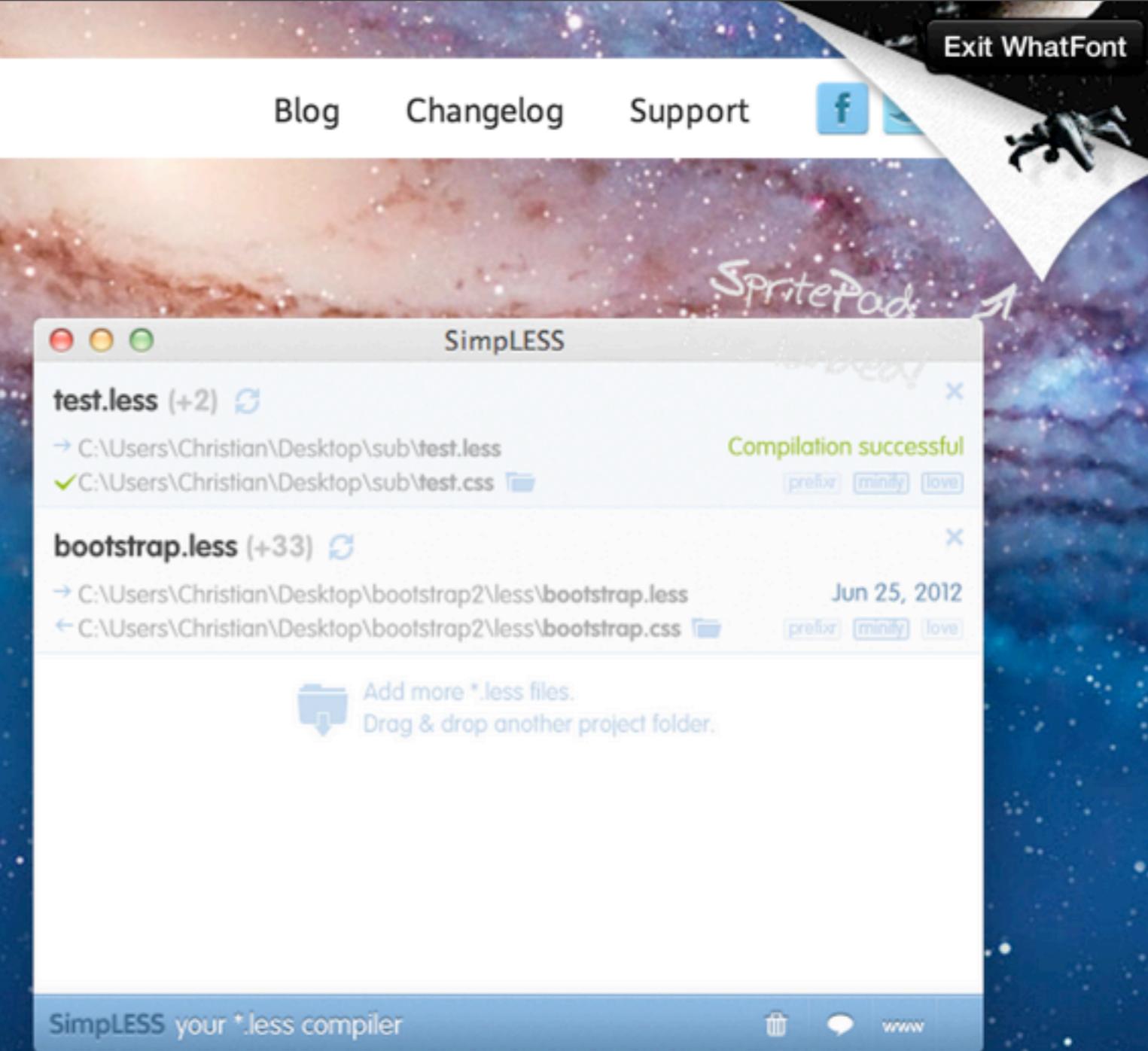
Done

```
@headline-color: #252525;
@copy-color: #565656;
```

SimpLESS

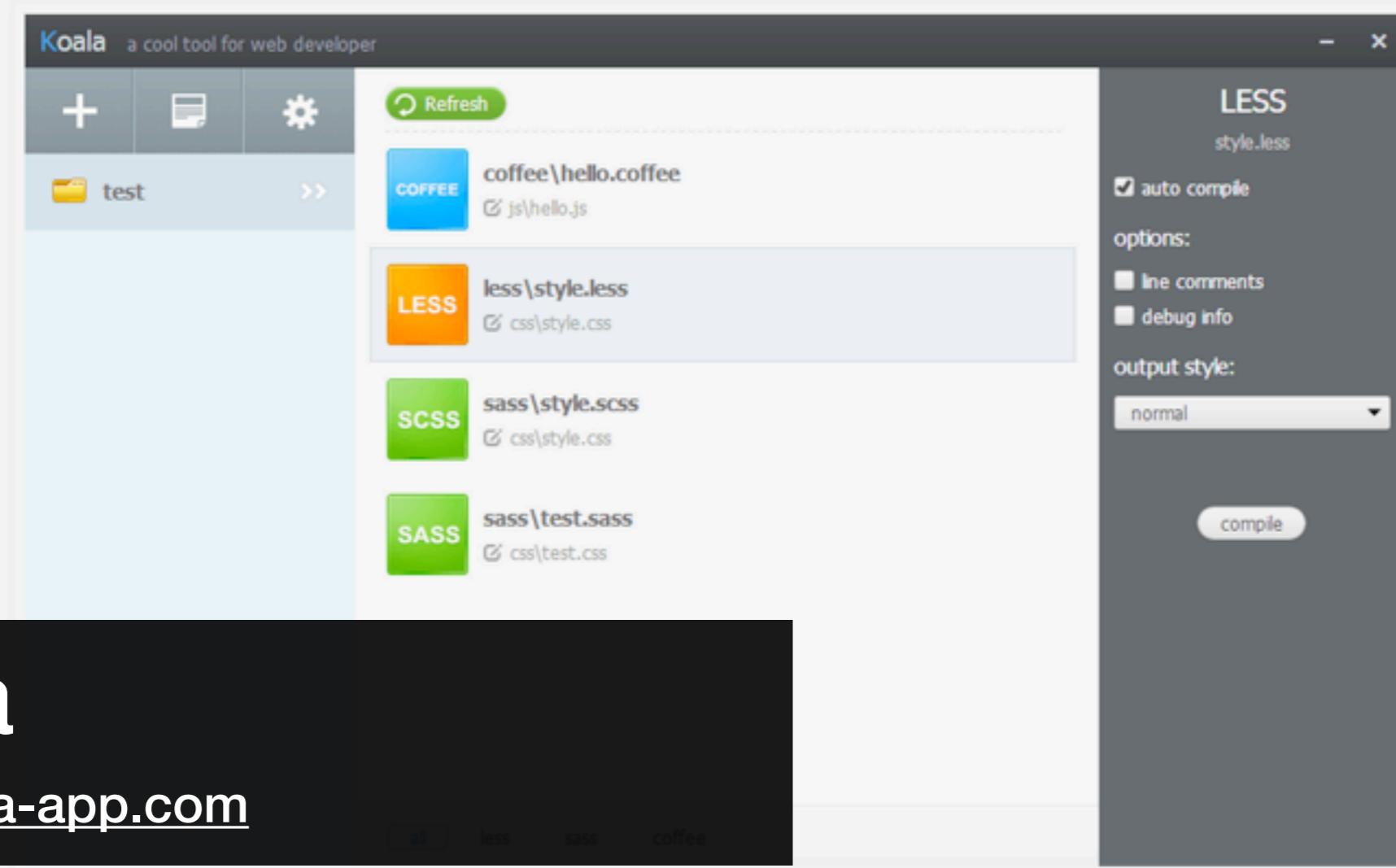


Fonts from



a gui application for Less, Sass, Compass and CoffeeScript compilation.

Koala is a GUI application for Less, Sass, Compass and CoffeeScript compilation, to help web developers to use them more efficient.Koala can run in windows, linux and mac.



# Koala

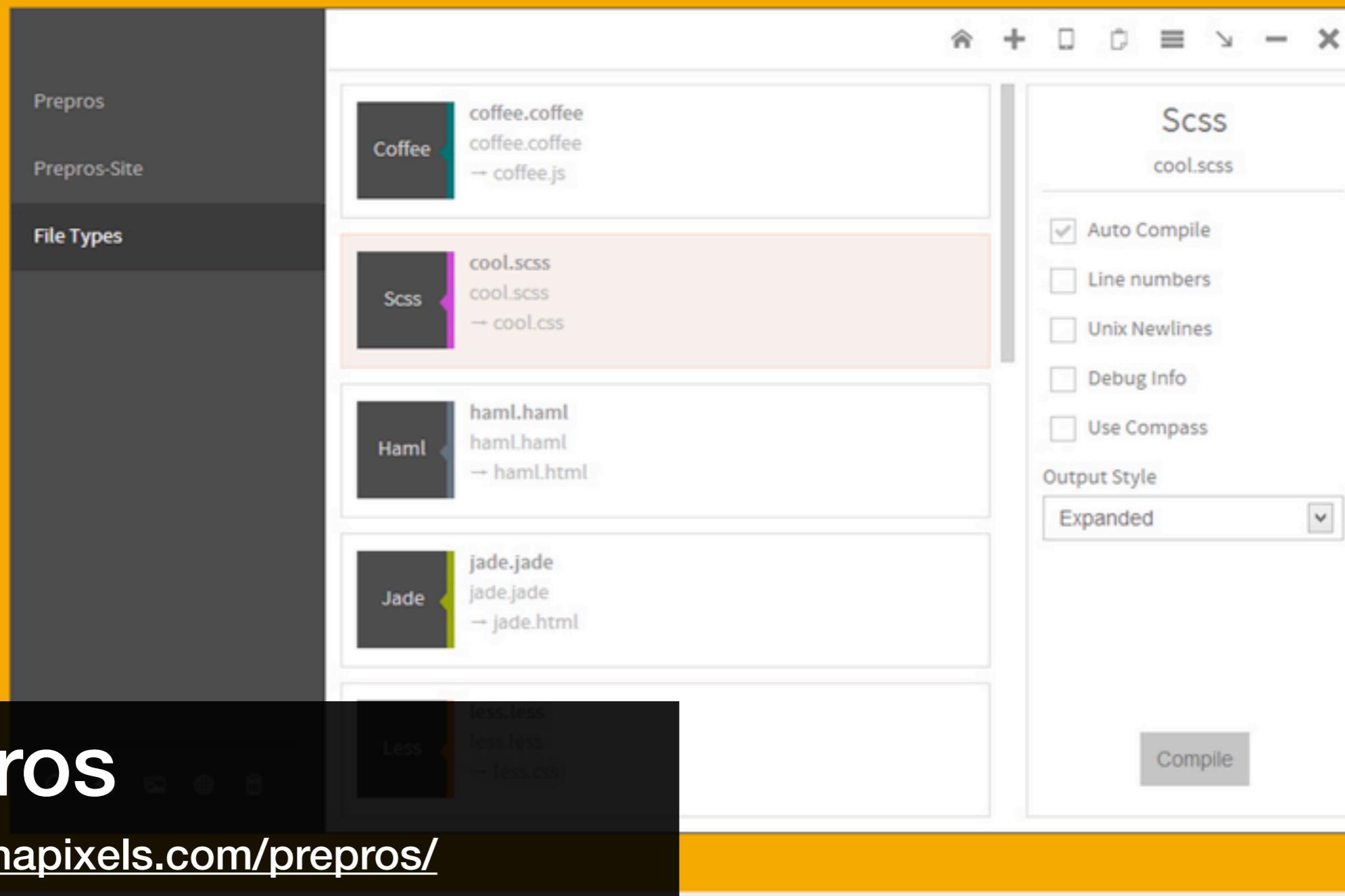
<http://koala-app.com>

all less sass coffee

## Download

# Preprocessing just got easier with Prepros

## Enjoy the dead simple design & development workflow



# Prepros

<http://alphapixels.com/prepros/>

[Prepros 2.3.2 for Windows](#)

[Prepros 2.3.2 for OSX 10.7+](#)

[Install Chrome Extension](#)

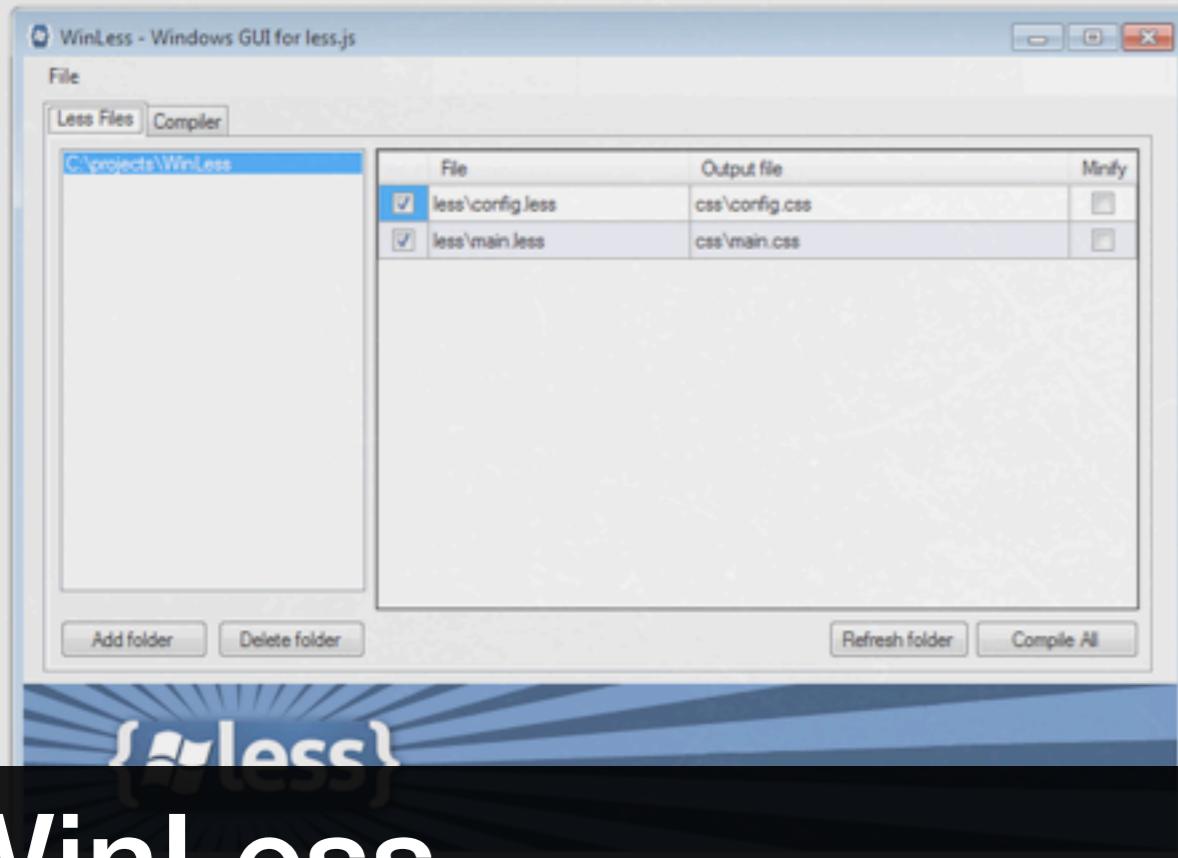


Windows GUI for LESS.js

WinLess

WinLess.lessc NuGet Package

Online Less Compiler



# WinLess

<http://winless.org>

WinLess

Usage

When you add a folder to WinLess (either via 'Add folder' or by dropping it on the folder pane),

## WinLess

WinLess is a Windows GUI for LESS.js. WinLess is a must-have for the webdeveloper who uses Windows.

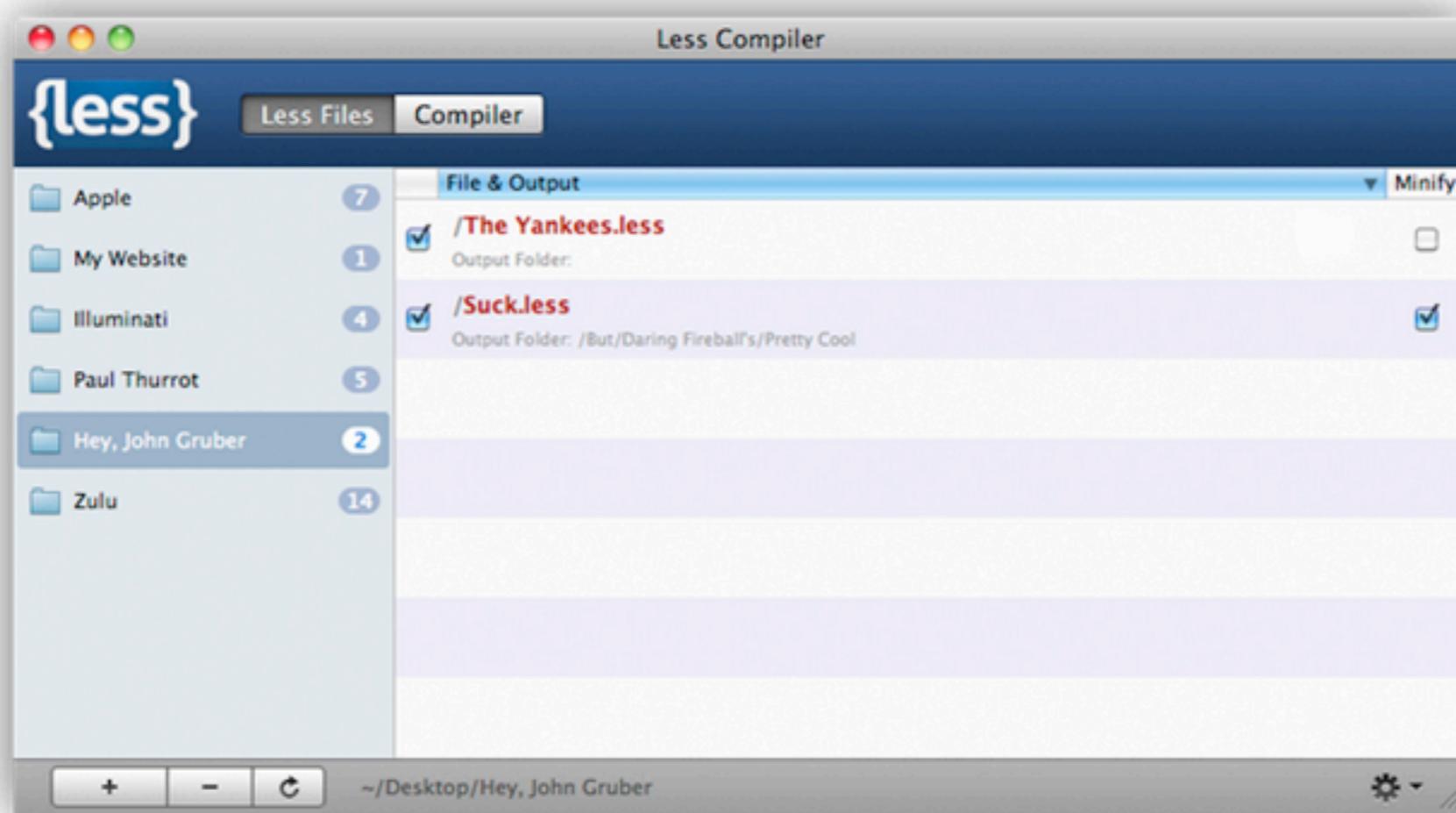


Version: 1.8.1

Tweet 258

## Features

- **NEW!** Can now be called with [command line](#)



The only thing easier is making fun of Internet Explorer.  
**less.app**

{Less} extends CSS with variables, nested rules, operators and more. If you're still building websites without it, you're an idiot. This app

makes it dead simple to use {Less} by automatically compiling \*.less files into standard CSS.

**Download**



Version: 2.8 (OS X 10.6+) | [Release Notes](#)



CodeKit

Files Log

SEARCH

PROJECTS

- Crazy Client
- My Website
- Steve
- UNC

FRAMEWORKS

- VDKFramework
- Twitter Bootstrap

All Files Styles Scripts Pages Images

/index.php

/scripts/jquery.js  
This file does not directly compile.

/scripts/pageController.js

/scripts/min/pageController-min.js

/scripts/storeController.coffee

/scripts/min/storeController.js

/store/storePage.haml

/store/storePage.html

/styles/about.sass

/css/about.css

/styles/main.less

/css/main.css

/styles/store.styl

/css/store.css

**Javascript**

Do not compile directly

Check Syntax With:

JSHint

When Processing:

Concatenate, then minify

Check Syntax

Imports:

jquery.js Prepend

Imported By:

Process

The screenshot shows the CodeKit application window. On the left, there's a sidebar with 'PROJECTS' containing 'Crazy Client', 'My Website' (which is selected), 'Steve', and 'UNC'. Below that is the 'FRAMEWORKS' section with 'VDKFramework' and 'Twitter Bootstrap'. The main area is titled 'All Files' and lists files categorized by type: PHP, JavaScript, CSS, LESS, SASS, HAML, and Stylus. A specific file, 'pageController.js', is highlighted. To the right of the file list is a configuration sidebar for 'Javascript' processing. It includes options for 'Do not compile directly', 'Check Syntax With' (set to JSHint), 'When Processing' (set to 'Concatenate, then minify'), and a 'Check Syntax' button. Below this are sections for 'Imports' (listing 'jquery.js' with a 'Prepend' dropdown) and 'Imported By' (a currently empty list). At the bottom right of the sidebar is a large green 'Process' button.

# CodeKit

<https://incident57.com/codekit/>

It's like steroids for web developers.

GUI Tools	LESS version	Platform
Crunch!	1.3.3	Win & Mac
SimpLESS	1.3.0	Win & Mac
Koala	1.4.0	Win & Mac
Prepros	1.4.1	Win & Mac
WinLess	1.4.1	Windows
less.app	1.3.3	Mac
CodeKit	1.4.1	Mac

2013-08-04現在

Online Debugging Tools	LESS version
LESS2CSS	1.4.1
CODEPEN	1.4.x
jsdo.it	less than 1.4x
JS Bin	less than 1.4x

2013-08-04現在

# Client-side usage

```
<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();
    // or append '#!watch' to the browser URL, then refresh the page.
  </script>
</head>
```



I can't access to any local files

```
$ /Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome
--allow-file-access-from-files
```



技術部追い剥ぎペンギン » CSS »

# LESS.jsをchromeのローカルでも使用する方法。 と、その為のアプリ作った!

on 2013年3月15日 18:40 · in css

2 ツイート 2 いいね! 10 +1 0

CSSのコーディングをスピードアップする方法で、LESSやSassのようなCSS拡張メタ言語があります。

最近になって使い始めたのですが超便利ですね。

参考 : CSSの記述が3倍速くなる「LESS」の使い方

## chrome-for-less

<http://www.kigurumi.asia/tech/?p=365> ません。

Firefoxでは普通に動くのに。

Search



中の人

新宿御苑前付近に出没のWEBクリエイター。日々の作業中に備忘的に更新してます。本館は[こちらから](#)



ナカシマ

@kanakogiさんをフォロー

おすすめ記事(本館含む)



書籍代を大幅削減  
「その本、図書館にあります。」  
[713 users](#)

[1184 tweets](#)



普通のブログなのに表示されるエロいバナー広告は完全にアウトだと思う件

# LESSの基本文法

## SASSとの比較を交えて

# Variables in LESS

```
@font-color: #143352;  
@selector: body;  
@base-url: "http://root.com";  
@property: background;  
  
{@selector} {  
    color: @font-color;  
    background: url("@{base-url}/img/bg.png")  
    //not possible to use any variables for the property.  
}
```

{less}

# Variables in SCSS

```
$font-color: #143352;  
$selector: body;  
$base-url: "http://root.com";  
$property: background;  
  
#{$selector} {  
    color: $font-color;  
    #{$property}: url("#{ $base-url }/img/bg.png");  
}
```

Sass.

# compiles to CSS

```
body {  
    color: #143352;  
    background-image: url("http://root.com/img/bg.png");  
}
```



# Scope in LESS

```
@var: 0;

.class1 {
  @var: 1;

.class {
  @var: 2;
  three: @var;
  @var: 3;
}

  one: @var;
}
```

{less}

```
.class1 {
  one: 1;
}
.class1 .class {
  three: 3;
}
```



# Scope in SCSS

```
$var: 0;  
  
.class1 {  
  $var: 1;  
  
.class {  
  $var: 2;  
  two: $var;  
  $var: 3;  
}  
  
three: $var;  
}
```

Sass.

```
.class1 {  
  three: 3;  
}  
.class1 .class {  
  two: 2;  
}
```



# Mixins in LESS

```
.bordered-top {  
    border-top: dotted 1px black;  
}  
.border-btm() {  
    border-bottom: solid 2px black;  
}  
.rounded-corners (@radius: 5px) {  
    -webkit-border-radius: @radius;  
    -moz-border-radius: @radius;  
    -ms-border-radius: @radius;  
    -o-border-radius: @radius;  
    border-radius: @radius;  
}  
  
h3 {  
    .bordered-top;  
    .border-btm;  
    .rounded-corners(3px);  
}
```

{less}

# compiles to CSS

```
.bordered-top {  
    border-top: dotted 1px black;  
}  
  
h3 {  
    border-top: dotted 1px black;  
    border-bottom: solid 2px black;  
    -webkit-border-radius: 3px;  
    -moz-border-radius: 3px;  
    -ms-border-radius: 3px;  
    -o-border-radius: 3px;  
    border-radius: 3px;  
}
```



# Mixins in SCSS

```
@mixin border-top {  
  border-top: dotted 1px black;  
  border-bottom: solid 2px black;  
}  
  
@mixin border-btm {  
  border-bottom: solid 2px black;  
}  
  
@mixin rounded-corners ($radius: 5px) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  -o-border-radius: $radius;  
  border-radius: $radius;  
}  
  
h3 {  
  @include border-top;  
  @include border-btm;  
  @include rounded-corners(3px);  
}
```

Sass.

# compiles to CSS

```
h3 {  
    border-top: dotted 1px black;  
    border-bottom: solid 2px black;  
    -webkit-border-radius: 3px;  
    -moz-border-radius: 3px;  
    -ms-border-radius: 3px;  
    -o-border-radius: 3px;  
    border-radius: 3px;  
}
```



# Namespaces in LESS

```
#bundle {  
  .button() {  
    display: block;  
    border: 1px solid black;  
    background-color: grey;  
  }  
}  
  
header a {  
  #bundle > .button();  
}
```

{less}

```
header a {  
  display: block;  
  border: 1px solid black;  
  background-color: grey;  
}
```



# Selector Inheritance in SCSS

```
.error {  
  border: 1px #f00;  
  background: #fdd;  
  .misc {  
    color: darkblue;  
  }  
}  
.error.intrusion {  
  font-size: 1.3em;  
  font-weight: bold;  
}  
  
.badError {  
  @extend .error;  
  border-width: 3px;  
}
```

Sass.

```
.error,  
.badError {  
  border: 1px #f00;  
  background: #fdd;  
}  
.error .misc,  
.badError .misc {  
  color: darkblue;  
}  
.error.intrusion,  
.badError.intrusion {  
  font-size: 1.3em;  
  font-weight: bold;  
}  
.badError {  
  border-width: 3px;  
}
```



# Selector Inheritance in LESS (with “all”)

```
.error {  
    border: 1px #f00;  
    background: #fdd;  
    .misc {  
        color: darkblue;  
    }  
}  
.error.intrusion {  
    font-size: 1.3em;  
    font-weight: bold;  
}  
  
.badError:extend(.error all) {  
    border-width: 3px;  
}
```

{less}

```
.error,  
.badError {  
    border: 1px #f00;  
    background: #fdd;  
}  
.error .misc,  
.badError .misc {  
    color: darkblue;  
}  
.error.intrusion,  
.badError.intrusion {  
    font-size: 1.3em;  
    font-weight: bold;  
}  
.badError {  
    border-width: 3px;  
}
```



# Selector Inheritance in LESS (without “all”)

```
.error {  
  border: 1px #f00;  
  background: #fdd;  
  .misc {  
    color: darkblue;  
  }  
}  
.error.intrusion {  
  font-size: 1.3em;  
  font-weight: bold;  
}  
  
.badError {  
  &:extend(.error);  
  border-width: 3px;  
}
```

{less}

```
.error,  
.badError {  
  border: 1px #f00;  
  background: #fdd;  
}  
.error .misc {  
  color: darkblue;  
}  
.error.intrusion {  
  font-size: 1.3em;  
  font-weight: bold;  
}  
.badError {  
  border-width: 3px;  
}
```



# Selector Inheritance in SCSS (with %)

```
%clearfix {  
  *zoom: 1;  
  &:before,  
  &:after {  
    display: table;  
    content: "";  
    line-height: 0;  
  }  
  &:after {  
    clear: both;  
  }  
}  
  
nav ul {  
  @extend %clearfix;  
  li {  
    float: left;  
  }  
}
```

Sass.

```
nav ul {  
  *zoom: 1;  
}  
nav ul:before,  
nav ul:after {  
  display: table;  
  content: "";  
  line-height: 0;  
}  
nav ul:after {  
  clear: both;  
}  
  
nav ul li {  
  float: left;  
}
```



# Nesting Rules

```
header {  
  color: black;  
  
  nav {  
    font-size: 12px;  
    @media only screen and (min-width: 600px) {  
      font-size: 16px;  
    }  
  }  
  .logo {  
    width: 300px;  
    &:hover { text-decoration: none }  
  }  
}
```

{less} Sass.

# compiles to CSS

```
header {  
  color: black;  
}  
header nav {  
  font-size: 12px;  
}  
@media only screen and (min-width: 600px) {  
  header nav {  
    font-size: 16px;  
  }  
}  
header .logo {  
  width: 300px;  
}  
header .logo:hover {  
  text-decoration: none;  
}
```



# Nesting Rules

```
.fluid-divided-module-width (@width, @col, @target) {  
    width: @width;  
    .lt-ie8 & {  
        width: @width - (0.5 / unit(@target) * 100%);  
    }  
}  
  
nav ul li {  
    .fluid-devided-module-width(25%, 4, 320px);  
}
```

{less}

```
nav ul li {  
    width: 25%;  
}  
  
.lt-ie8 nav ul li {  
    width: 24.84375%;  
}
```



# Guard Expression in LESS

```
.set-color (@color) when (lightness(@color) >= 50%) {  
  background-color: black;  
}  
.set-color (@color) when (lightness(@color) < 50%) {  
  background-color: white;  
}  
.set-color (@color) {  
  color: @color;  
}  
  
body {  
  .set-color(#777);  
}
```

{less}

```
body {  
  background-color: white;  
  color: #777777;  
}
```



# @if Directive in SCSS

```
@mixin set-color($color) {  
  
  @if lightness($color) >= 50% {  
    background-color: black;  
  } @else if (lightness($color) < 50%) {  
    background-color: white;  
  }  
  color: $color;  
  
}  
body {  
  @include set-color(#777);  
}
```

Sass.

```
body {  
  background-color: white;  
  color: #777777;  
}
```



# Pattern-matching in LESS

```
.case (dark, @color) {  
  color: darken(@color, 10%);  
}  
.case (light, @color) {  
  color: lighten(@color, 10%);  
}  
.case (@_, @color) {  
  display: block;  
}  
  
@switch: light;  
.class {  
  .case(@switch, #888);  
}
```

{less}

```
.class {  
  color: #a2a2a2;  
  display: block;  
}
```



# @if Directive in SCSS

```
@mixin case($c, $color){  
  @if $c == dark {  
    color: darken($color, 10%);  
  }@else if $c == light {  
    color: lighten($color, 10%);  
  }  
  display: block;  
}  
  
$switch: light;  
.class {  
  @include case($switch, #888);  
}
```

Sass.

```
.class {  
  color: #a2a2a2;  
  display: block;  
}
```



# Loop in LESS

```
//Guard
.loop(@i) when (@i > 0){

    .item-@{i} {
        width: 2em * @i;
    }
    //Recursive mixin
    .loop(@i - 1);
}

//Pattern-matching
.loop(0){}

@i:6;
.loop(@i);
```

{less}

```
.item-6 {
    width: 12em;
}
.item-5 {
    width: 10em;
}
.item-4 {
    width: 8em;
}
.item-3 {
    width: 6em;
}
.item-2 {
    width: 4em;
}
.item-1 {
    width: 2em;
}
```



# @for Directive in SCSS

```
@for $i from 1 through 6 {  
  .item-#{$i} {  
    width: 2em * $i;  
  }  
}
```

Sass.

```
.item-1 {  
  width: 2em;  
}  
.item-2 {  
  width: 4em;  
}  
.item-3 {  
  width: 6em;  
}  
.item-4 {  
  width: 8em;  
}  
.item-5 {  
  width: 10em;  
}  
.item-6 {  
  width: 12em;  
}
```



# Functions & Operations

```
@baseColor : #1d4000;

.set-fontSize(@pxsize) when ( ispixel(@pxsize) ){
  @remsize: unit(@pxsize, rem) * 0.1;
  font-size: @pxsize;
  font-size: @remsize;
}

aside {
  .set-fontSize(16px);
  background: lighten(@baseColor, 2.5%);
}
```

{less}

```
aside {
  font-size: 16px;
  font-size: 1.6rem;
  background: #234d00;
}
```



# Escaping in LESS

```
@midium-width: 670px;  
@midium: ~"only screen and (min-width: #{@midium-width});  
  
.box-shadow(@shadow) {  
  -webkit-box-shadow: @shadow;  
  box-shadow: @shadow;  
}  
  
@media @midium {  
  body {  
    font-size: 18px;  
    font-size: 1.8rem;  
  }  
  #content {  
    .box-shadow(~"7px 0px 10.5px rgba(0,0,0,.34), -7px 0px 10.5px  
    rgba(0,0,0,.34)");  
  }  
}
```

{less}

# compiles to CSS

```
@media only screen and (min-width: 670px) {  
  body {  
    font-size: 18px;  
    font-size: 1.8rem;  
  }  
  #content {  
    -webkit-box-shadow: 7px 0px 10.5px rgba(0,0,0,.34), -7px 0px 10.5px  
    rgba(0,0,0,.34);  
    box-shadow: 7px 0px 10.5px rgba(0,0,0,.34), -7px 0px 10.5px  
    rgba(0,0,0,.34);  
  }  
}
```



# JavaScript evaluation in LESS

```
.box-shadow(...) {  
  
    @processing:  
        ~`function () {var arg = "@{arguments}".replace("[",  
"").replace("]", "") || "none";  
            if (!/^#?\w*%?([ X])/ .test(arg)) {  
                arg = arg.replace(/,(?=[^()]*\))/g, '--').replace(/,/g,  
"").replace(/--/g, ',');  
            }  
            return arg;  
        })();  
  
    -webkit-box-shadow: @processing;  
    box-shadow: @processing;  
}  
  
#content {  
    .box-shadow(7px 0px 10.5px rgba(0,0,0,.34),  
    -7px 0px 10.5px rgba(0,0,0,.34));  
}
```

{less}

# compiles to CSS

```
#content {  
    -webkit-box-shadow: 7px 0px 10.5px rgba(0,0,0,.34), -7px 0px 10.5px  
    rgba(0,0,0,.34);  
    box-shadow: 7px 0px 10.5px rgba(0,0,0,.34), -7px 0px 10.5px  
    rgba(0,0,0,.34);  
}  
}
```



# Importing in LESS

```
// In LESS, you don't need to begin with an underscore  
// in order that the files be "partial" in SASS.  
@import "libs/preboot";  
  
//Importing css file as less.  
@import (less) "../css/normalize.css";  
@import (less) "../css/main.css";  
  
//Importing preboot v1 in bootstrap.less.  
@import "bootstrap/less/variables";  
@import "bootstrap/less/mixins";
```

{less}

[MSDN Blogs](#) > [IEInternals](#) > Stylesheet Limits in Internet Explorer



EricLaw [ex-MSFT] 14 May 2011 2:14 PM

14

[KB 262161](#) outlines the maximum number of stylesheets and rules supported by Internet Explorer 6 to 9.

- A sheet may contain up to 4095 rules
- A sheet may @import up to 31 sheets
- @import nesting supports up to 4 levels deep

Some folks have [wondered](#) about the math that underlies these numbers. The root of the limitations is that Internet Explorer uses a 32bit integer to identify, sort, and apply the cascading rules. The integer's 32bits are split into five fields: four sheetIDs of 5 bits each, and one 12bit ruleID. The 5 bit sheetIDs results in the 31 @import limit, and the 12 bit ruleID results in the 4095 rules-per-sheet limitation. While these limits are entirely sufficient for most sites, there are some sites (particularly when using [frameworks](#) and [controls](#)) that can encounter the limits, requiring workarounds.

There's a simple test page for the limits [here](#).

-Eric

**Update:** IE10 Platform Preview #2 significantly [raises the limits](#) described above. In IE10 (any browser/document mode):

- A sheet may contain up to 65534 rules
- A document may use up to 4095 stylesheets
- @import nesting is limited to 4095 levels (due to the 4095 stylesheet limit)

# IEInternals

<http://blogs.msdn.com/b/ieinternals/archive/2011/05/14/internet-explorer-stylesheet-rule-selector-import-sheet-limit-maximum.aspx>



Comments



Translate This Page

Translate this page

Klingon



Microsoft® Translator



Please read my blog's [comment policy](#) [here](#).

Options

Blog Home

Email Blog Author

Share this

RSS for posts

Atom

RSS for comments

Search

Search MSDN with Bing



Search this blog  Search all blogs

Tags

ActiveX add-ons Best-Practices

BetterInIE10 BetterInIE9

caching design dev fixes http https

ie9 interop limitations



# 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?

# Bless

IE v6, v7, v8, v9 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 broken.

<http://blesscss.com>

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

[Downloads](#)



[Bless for Mac](#)

Basic drag and drop app

[Bless.js](#)

View the project on GitHub

[Example.css](#)

Download and try it out!

# ライブラリの紹介

# LESS Hat. A bunch of smart LESS mixins.

Kick-ass universal mixin library from makers of [CSS Hat](#).

 Download LESS Hat 1.1.2

 [Spread the word!](#)

Introduction

Configuration

Design considerations

FAQ

Roadmap

Drawbacks

Feedback

.animation

.keyframes

EN ROUTE

appearance

# LESS Hat

<http://lesschat.com>

.background-origin

.background-size

## Introduction

Why LESS Hat? We were searching for LESS mixins library that we could use in [CSS Hat](#) plugin for exporting Photoshop layer styles into CSS.

**We required the mixins to be:**

- **Universal:**

If it's possible in CSS, it must be possible to be done with a mixin. Unlimited number of anything, shadows, gradients, gradient swatches.

- **Cross-browser:**

The library should export CSS for all available browsers. Old gradient syntax for iOS4? Totally!

- **Configurable:**

Building a webkit app? Turn off all other browsers and get just CSS3 and -webkit- extensions without the clutter.

We looked at existing libraries, and found out they are not matching our criteria:

# Clearless

A collection of reusable LESS mixins



[Download .zip](#)



[Download .tar.gz](#)



[View on GitHub](#)

## ClearLess

A reuseable collection of carefully-considered **Less** mixins, or **YALML** (Yet Another Less Mixin Library).

The core tenets of this mixin library are to *avoid output bloat wherever possible* (via duplicated properties etc) and to *provide flexible, configurable solutions* to the problems that are addressed by the library (i.e. by using Modernizr classes, browser hacks or not, etc). The aim is to give the author the benefits of reusable shortcuts without obliterating personal style and generating bloated stylesheets.

**Before diving in** it is strongly recommended that you peruse the **notes on usage and best practices** at the end of this document, which gives an overview of how you can take full advantage of ClearLess without compromising the generated CSS output.

## Clearless

Simply @import the `mixins/all.less` file into the top of your main Less file, and then

<http://clearleft.github.io/clearless/>

The `mixins/all.less` file itself simply imports all the individual Less files into one place.

The mixins and settings for these individual files are documented under in their various

# Preboot

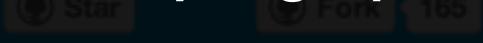
A collection of LESS  
mixins and variables for  
writing better CSS from  
[@mdo](#).

[Download Preboot](#)

[View on GitHub](#)

# Preboot

<http://getpreboot.com>



[Tweet](#) 423

## Introducing Preboot

Preboot is a comprehensive and flexible collection of LESS utilities. Its original variables and mixins became the precursor to Bootstrap. Since then, it's all come full circle.

After a two-year hiatus, Preboot has been reborn with many of the LESS variable and mixin improvements from Bootstrap, along with some new enhancements, too. It's one of the lightest and most powerful tools for CSS development without any prebuilt components.

Need something more powerful? Use [Bootstrap](#).

[Download Preboot 2](#)

[GitHub project](#)

## What's included

Here's the rundown of what you can find in Preboot. Jump to a section for some lightweight documentation and code snippets.

[Variables](#)

[Vendor mixins](#)

[Utility mixins](#)

# Bootstrap

Sleek, intuitive, and powerful front-end framework for faster and easier web development.

[Download Bootstrap](#)[GitHub project](#)[Examples](#)[Extend](#)[Version 2.3.2](#)

54,944



18,600



Follow @twbootstrap



Tweet

0

# Bootstrap

<http://getbootstrap.com>

## Introducing Bootstrap.

Need reasons to love Bootstrap? Look no further.

# LESS or SASS

It depends on the person



## 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?

## Why Less?

Sass is great with rails and has lots of cool, *super* powerful functionality. That said, we’ve chosen Less over

Sass because:

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

- **Bootstrap compiles ~6x faster with Less than Sass** A while back I created a simple benchmark which measured the time it took to compile [Bootstrap](#) (written in Less) vs. [Bootstrap-Sass](#) (the most popular bootstrap port). Our Less version compiled in a little under 90 ms on average, while the Sass version compiled around

# Why LESS for Bootstrap

- ✓ “Bootstrap compiles ~6x faster with Less than Sass”
- ✓ “Less is implemented in JavaScript”
- ✓ “Less is simple”



## WHY I SWITCHED FROM LESS TO SASS

NOVEMBER 13, 2012

Yaaay, another blog post on the web about CSS preprocessors, as if there wasn't enough. Don't worry: I won't try to convince you to use a CSS preprocessor, neither about the one you should pick. These decisions are really up to you.

So this post will be about my own experience with CSS preprocessors. For the concern, I recently wrote an article for Codrops untitled "10 things I learnt about CSS" and I talked a lot about preprocessors, so I've read (and tried) a bunch of things on the topic lately.

### WHY I SWITCHED FROM LESS TO SASS

Anyway and before anything, please note I'm not a hardcore CSS preprocessor user (<http://hugogiraudeau.com/2012/11/13/less-to-sass/>), but I've already worked a little bit on 2 of them: firstly LESS then Sass. I recently moved from LESS to Sass and don't plan on going back.

Hi! I'm Hugo, a **front-end developer** from France. CSS Goblin, Sass hacker, margin psycho.

Author at [Codrops](#), helper at [CSS-tricks](#), curator of [Browserhacks](#). You can catch me on [Twitter](#).

## Why he think Sass is better

- ✓ “It's way easier than the LESS syntax in my opinion”  
(about CONDITIONAL STATEMENTS)
- ✓ “It looks very intuitive to me”  
(about LOOPS)
- ✓ “LESS is still a good CSS preprocessor, but  
in the end I think Sass is simply better.”



# LESS VS SASS VS STYLUS



First off, this is a biased blog post as I am **heavily involved** in less.js at the moment. I have been inspired to write this because of the popular **sass vs less post** and because although it is very accurate (at the time it was written) it doesn't really cover the design philosophy.

Luke Page

Published  
08 Mar 2013

So I would like to show why, when choosing a language to write CSS in, you shouldn't just look at the number of features it has, but you should look at what kind of code you are going to get and how maintainable it is.

12    3

## WHY YOU SHOULDN'T USE ANY CSS PRE-PROCESSOR AT ALL

I think people look at badly written css, see repetition, realise that using a pre-processor can reduce that repetition and proceed to convert their badly written css into code that looks (on the surface) much nicer.

For instance, here is some css.

## LESS VS SASS VS STYLUS

```
.container {
```

<http://www.scottlogic.com/blog/2013/03/08/less-vs-sass-vs-stylus.html>

```
}
```

```
.container .button {
```

```
color: black;
```

# LESS Design Philosophy

- ✓ LESS is Declarative
- ✓ Less is very conservative in adding syntax
- ✓ Less aims to implement features in a CSS way

# Thank you !

jsCafé vol.12

2013-08-04

Katsunori Tanaka