

# Stylusが目指す CSSプリプロセッサ

CSS Nite LP26

さとう歩 / @ahomu

CyberAgent

さとう歩  
@ahomu



CyberAgent  
Web Developer



# havelog

イベント告知 Frontend Vol.4 - Keys to become better J...

2013-01-08

モバイルブラウザのキャッシュとデータストレージにつ...

2013-01-07 [40 users](#)

2012年を年の瀬にふりかえる

2012-12-28 [1 user](#)

Backbone.jsを使うときに参考になるリソース 2012年末...

2012-12-26 [106 users](#)

Subview的なモノのより良い管理办法 (Backbone.Advanced)

2012-12-25 [4 users](#)

## havelog (ブログ)

Xmasなので、こっ <http://havelog.ayumusato.com>

2012-12-24

鶏ハムの復讐・プレーンとスパイシー

2012-12-23 [1 user](#)



Search or type a command



Explore Gist Blog Help



ahomu



enja-oss

Joined on Nov 27, 2012

**12**  
public repos    **0**  
private repos    **17**  
members**Repositories****Members****Edit enja-oss's Profile**

Find a Repository...

**All** Public Private Sources Forks Mirrors**Sass**

あの楽しいCSSをもう一度！

Last updated 12 hours ago

★ 9

↳ 4

**Underscore**

Underscore ドキュメント日本語訳

Last updated 14 hours ago

★ 7

↳ 8

**Backbone**

Backbone ドキュメント日本語訳

Last updated 2 days ago

JavaScript

★ 112

↳ 19

**stylus**

stylus ドキュメント日本語訳

Last updated 2 days ago

★ 2

↳ 3

**templates**

social media with grummit for enja-oss

Last updated 3 days ago

JavaScript

★ 0

↳ 0

**prototype**

enja-ossのプロジェクトホームプロトタイプ

Last updated 7 days ago

**enja-oss.github.com**

Last updated 9 days ago

Ruby

★ 0

↳ 0

6

# en.ja OSS on GitHub

<https://github.com/enja-oss>

## JavaScript MVCフレームワーク Backbone.jsのコメント付きソースコード日本語訳が公開

2012年12月14日

JavaScript MVCフレームワークとしてもっとも知名度が高いBackbone.jsのコメント付きソースコードの日本語訳が公開されています。havelogの記事「Backbone.jsコメント付きソースコード日本語訳」で紹介されていました。

翻訳を行ったのは、githubをベースにオープンソースソフトウェアのドキュメントを翻訳している翻訳コミュニティのenja-oss。

同グループはすでにBackbone.jsのFAQの翻訳を公開済みで、それに続いてBackbone.jsソースコードのコメントについても翻訳を終えました。

### 全部で1411行の軽量フレームワーク

Backbone.jsはJavaScript MVCフレームワークとしてもっとも知名度が高いものですが、とても軽量なフレームワークでもあります。

今回日本語訳が公開されたソースコードを見ても、全部で1411行、60.72kb。とても丁寧にコメントがついているので、JavaScript MVCフレームワークがどういう構造を持っているのか、年末年始の休みをつかってじっくり眺めてみてはいかがでしょうか。



Backbone/backbone.js at master · enja-oss/Backbone · GitHub

 ツイート 180  193

タグ : JavaScript, MVC, オープンソース

- › JavaScript MVCフレームワークはすでに十種類以上、その比較や最新情報などのまとめ
- › JavaScript MVC座談会。遅くならない？ それぞれの特徴は？ サーバとの通信は？ (後編)
- › JavaScript MVC座談会。遅くならない？ それぞれの特徴は？ サーバとの通信は？ (前編)

×

### Blogger in Chief

Junichi Niino (jniino)

IT系の雑誌編集者、オンラインメディア発行人を経て独立。新しいオンラインメディアの可能性を追求しています。



(詳しいプロフィール)

新サイト「Publickey Topics」始めました！

Publickeyの新着情報をチェックしませんか？

Twitterで : @Publickey

RSSリーダーで : Feed

## 「SPREADの再創造」

日本の業務アプリ開発における  
最強のデータグリッドを求めて

SPREAD  
for WPF

限	ロットNo.	在庫数	単位	単価	区分
/27	2011092001	200	本	90	仕入
/27	2011092003	400	本	90	仕入
/27	0000001	490	本	90	製造
商品計					1090
/27	20111119001	10	本	120	仕入
/27	123	4	本	100	製造
合計					1204

SPREAD  
for WPF

日本の業務アプリ開発における  
最強のデータグリッドを求めて

SPREAD  
for WPF

日本の業務アプリ開発における  
最強のデータグリッドを求めて

SPREAD  
for WPF

日本の業務アプリ開発における  
最強のデータグリッドを求めて

SPREAD  
for WPF

Publickeyで紹介 (n'ʌ` )nワイ

メンターのモデル駆動開発

メンターのモデル駆動開発

メンターのモデル駆動開発

メンターのモデル駆動開発

メンターのモデル駆動開発

メンターのモデル駆動開発

# **HTML5+JavaScript スマートフォンサイト実践講座**

**2月ごろ発売予定 Amazonでチェック!**

Webサイト制作者のための HTML5+JavaScriptスマートフォンサイト実践講座

共著: 佐藤 歩, 安藤 建一, 杉本 吉章

<http://www.amazon.co.jp/dp/4774154113/>

詳しくは  
<http://aho.mu>



1. (ぼくの) プリプロセッサ遍歴
2. 魅力的なポイントの紹介
3. 先人は何を目指していたのか？
4. Stylusが目指すCSSプリプロセッサ
5. まとめ

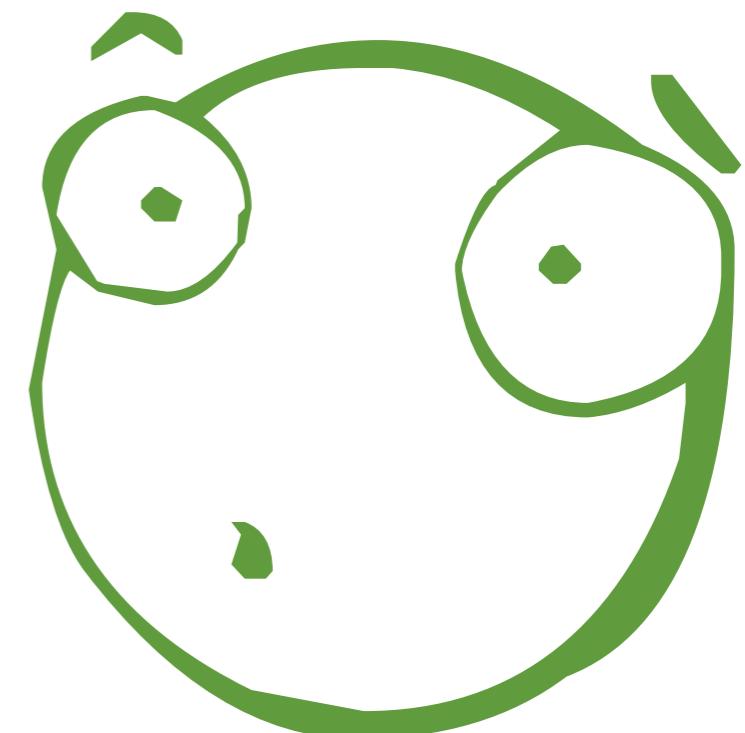
*stylus*

# 最初にコミットされた日(GitHub)



# What is *stylus*?

nodeJS製のCSSプリプロセッサ  
充実した機能・柔軟な文法が魅力  
主にTJ Holowaychukが開発  
↑ expressやjadeを作った人



# Stylusと主要アプリの対応

- CodeKit
- Sublime Text 2 (Stylus package)
- LiveReload (GUI)
- Grunt (grunt-contrib-stylus)
- ✗ Dreamweaver
- ✗ WebStorm / IntelliJ IDEA

- ✓ あたらしモノ好きの方に
- ✓ マイナー好きの方に
- ✓ CSSプリプロセッサを見直したり
- ✓ 第3の選択肢を知ってもらったり
- ✓ 何らかのキッカケになれば

A wide-angle photograph of a mountainous landscape under a blue sky with scattered white clouds. In the foreground, a paved road curves from the left towards the center. The middle ground shows a dry, brownish-yellow grassland. In the background, there are several mountain peaks, some with snow on their summits.

# プリプロセッサ遍歴と Stylusを使い始めた経緯

LESS → Stylus

# 最初はLESSから

CSSをぺたぺた書くのしんどい  
プリプロセッサ使ってみよう  
記法はCSSから離れすぎないのが良いな



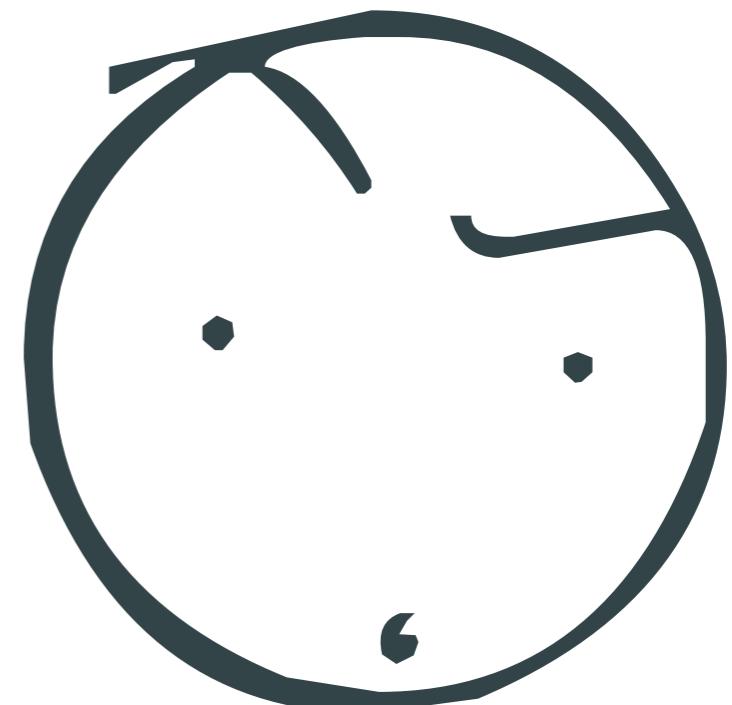
# LESS - CSSから離れたくない期

```
/* 変数・親参照・ネスト構造、これで十分じゃない？ */
```

```
@font-size: 12px;  
#header {  
    h1 {  
        font-size: @font-size * 2;  
        font-weight: bold;  
    }  
    p { font-size: @font-size;  
        a { text-decoration: none;  
            &:hover {  
                border-width: 1px;  
            }  
        }  
    }  
}
```

# LESSしてみた感想

手軽さ・素直さは良い  
やっぱり機能が物足りなく感じる  
たとえばアレとかソレとか…



# LESS - @extendやif/else, for~in が欲しい期

```
/* @extend欲しいなあ... */  
.btn,  
.btn-primary,  
.btn-secondary,  
.btn-danger {  
    .border-radius(5px);  
}
```

```
/* whenがある...。if/elseで書きたいなー */  
.mixin (@a) when (lightness(@a) >= 50%) {  
    background-color: black;  
}  
.mixin (@a) when (lightness(@a) < 50%) {  
    background-color: white;  
}
```

# Sassに改宗する今更感

いまさらメジャーに日和るなんて...





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;  
}
```

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!

What if we could omit braces?

# Stylus

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

```
a.button {  
  border-radius: 5px;  
  -moz-border-radius: 5px;  
  border-radius: 5px;
```

How about semi-colons?

# Stylusいいね！

機能も多くて、新しいしカッコいいぞ！

そして、Stylusを始めました



# 魅力的なポイントの紹介

開発者のブログから引用紹介

# CSS Syntaxと柔軟性

[via. Stylus 0.4.0 released CSS Syntax Support - TJ Holowaychuk](#)

Stylusの最も優れた点は言語デザイン

CSSスタイル/インデントスタイルの  
両方をサポートする柔軟性

## CSS Syntax : 既存文法を活かした構文・透過的mixin

```
vendor(prop, args) {  
  -webkit-{prop}: args;  
  -moz-{prop}: args;  
  {prop}: args;  
}  
border-radius() {  
  vendor('border-radius', arguments);  
}  
  
button,  
a.button,  
input[type=submit],  
input[type=button] {  
  border-radius: 3px 5px;  
  color: black;  
  &:hover {  
    background: black;  
    color: white;  
  }  
}
```

```
button,  
a.button,  
input[type=submit],  
input[type=button] {  
  -webkit-border-radius: 3px 5px;  
  -moz-border-radius: 3px 5px;  
  border-radius: 3px 5px;  
  color: #000;  
}  
  
button:hover,  
a.button:hover,  
input[type=submit]:hover,  
input[type=button]:hover {  
  background: #000;  
  color: #fff;  
}
```

## 柔軟性：記号はオプションで、インデントスタイルも可能

```
full  
body {  
  font: 14px/1.5 Helvetica;  
  #logo {  
    border-radius: 5px;  
  }  
}
```

```
; removed  
body {  
  font: 14px/1.5 Helvetica  
  #logo {  
    border-radius: 5px  
  }  
}
```

```
: removed  
body {  
  font 14px/1.5 Helvetica  
  #logo {  
    border-radius 5px  
  }  
}
```

```
{ } removed  
body  
  font 14px/1.5 Helvetica  
  #logo  
    border-radius 5px
```

# サポートする構文・機能

最も特徴的なのはプロパティのような透過的mixin

主にSass由来で、基本的な機能の大半をサポート  
if/elseやfor-inループなどの制御構文  
色彩・計算関係などの多彩なビルトイン関数など

## Try Stylus!

Stylus is an innovative stylesheet language that compiles down to CSS. Inspired by SASS, Stylus is built with node.js and capable of running in the browser as this interactive tutorial illustrates. Edit the Stylus source on the left to see the resulting changes appear to the CSS on the right.

```
body {  
  font: 14px/1.5 Helvetica, arial, sans-serif;  
  #logo {  
    border-radius: 5px;  
  }  
}
```

# Try Stylus

<http://learnboost.github.com/stylus/try.html>

# nib - mixinライブラリ

[via. Extend Stylus with a nib - TJ Holowaychuk](#)

## Stylusのmixinライブラリ

utilityプロパティやvendor-prefixの補完を提供  
透過的なmixinの特徴が良く出ている

# ベンダープレフィックスの補完とプロパティ拡張

```
@import 'nib';
button {
  border-radius: 5px;
}
button {
  border-radius: bottom 10px;
}
```

.styl

```
button {
  -webkit-border-radius: 5px;
  border-radius: 5px;
}
button {
  -webkit-border-bottom-left-radius: 10px;
  border-bottom-left-radius: 10px;
  -webkit-border-bottom-right-radius: 10px;
  border-bottom-right-radius: 10px;
}
```

.CSS

# ベンダープレフィックスと後方互換性の補完

```
@import 'nib';
body {
  background: linear-gradient(top, white, black);
}
```

.styl

```
body {
  background: -webkit-gradient(linear, left top, left bottom,
                                color-stop(0, #fff), color-stop(1, #000));
  background: -webkit-linear-gradient(top, #fff 0%, #000 100%);
  background: -moz-linear-gradient(top, #fff 0%, #000 100%);
  background: -o-linear-gradient(top, #fff 0%, #000 100%);
  background: -ms-linear-gradient(top, #fff 0%, #000 100%);
  background: linear-gradient(top, #fff 0%, #000 100%);
}
```

.CSS

## ピクセル密度に応じた画像ファイル(@2x)の展開

```
@import 'nib';
#logo {
  image: 'img/logo.main.png' 50px 100px
}
```

.styl

```
#logo {
  background-image: url("img/logo.main.png");
}
@media all and (-webkit-min-device-pixel-ratio: 1.5) {
  #logo {
    background-image: url("img/logo.main@2x.png");
    -webkit-background-size: 50px 100px;
    background-size: 50px 100px;
  }
}
```

.CSS

## 便利なプロパティの追加

```
@import 'nib';          .styl  
  
#back-to-top {  
    fixed: bottom right;  
}  
  
#table-of-contents {  
    absolute: top 20px left 20px;  
}  
  
.description {  
    overflow: ellipsis;  
}
```

```
#back-to-top {          .css  
    position: fixed;  
    bottom: 0;  
    right: 0;  
}  
  
#table-of-contents {  
    position: absolute;  
    top: 20px;  
    left: 20px;  
}  
  
.description {  
    white-space: nowrap;  
    overflow: hidden;  
    -o-text-overflow: ellipsis;  
    text-overflow: ellipsis;  
}
```

# @extend のサポート

[via. Stylus gets @extend - TJ Holowaychuk](#)

Sassでお馴染みのアレ  
Single-Classパターンをサポート  
@exnted または @extends として使用

## Single-Classパターンもすっきり、@exntedサポート

```
.message {  
    font-size: 14px;  
    padding: 5px 10px;  
    border: 1px solid #eee;  
    border-radius: 5px;  
}  
.warning {  
    @extends .message;  
    color: yellow;  
}  
.error {  
    @extends .message;  
    color: red;  
}  
.fatal-error {  
    @extends .error;  
    font-weight: bold  
}
```

.styl

```
.message,  
.warning,  
.error,  
.fatal-error {  
    font-size: 14px;  
    padding: 5px 10px;  
    border: 1px solid #eee;  
    border-radius: 5px;  
}  
.warning {  
    color: #ff0;  
}  
.error,  
.fatal-error {  
    color: #f00;  
}  
.fatal-error {  
    font-weight: bold;  
}
```

.css

# エラーレポーティング

[via. Stylus vs SASS vs LESS error reporting – TJ Holowaychuk](#)

スタックトレース付きエラー箇所の出力が賢い

よくあるtypoをプリプロセスで検知できる  
構文解析をどれくらい厳密に行っているかの試金石



表示はやや不親切で、全角文字などのtypoを検出しない

```
% sass sample.scss
Syntax error: Invalid CSS after "...background: foo":
expected ";", was "[fail];"
      on line 6 of sample.scss
Use --trace for backtrace.
```



1.3で表示は親切になったが、検知力はSassより甘め

```
% lessc sample.less
ParseError: Syntax Error on line 6 in sample.less:6:4
5   form input {
6     background: foo[fail']';
7 }
```



## 親切な表示と、全角文字の混入typoも検出する精度

```
% stylus sample.styl
~/node_modules/stylus/bin/stylus:516
    throw err;
        ^
Error: sample.styl:6
  2|   margin: 0 30px;
  3| }
  4| body {
  5|   form input {
> 6|     background: foo[fail];
  7|   }
  8| }

cannot perform foo[(fail)]
  at "form input " (sample.styl:6)
  at "body " (sample.styl:5)
```

# 画像リソースの操作関係

## Data URI Image

→ 標準で画像パスからデータURIの展開をサポート

## Plugin stylus-sprite

## Plugin stylus-lemonade

→ CSS Spriteをサポートするプラグインがある

A nighttime photograph of the space shuttle Endeavour mounted on its mobile launcher platform at the Kennedy Space Center. The shuttle is positioned vertically, with its orange external fuel tank and white solid rocket boosters clearly visible. The large white payload fairing is attached to the top of the tank. The shuttle's name "Endeavour" is printed on the side of the fairing. To the left of the shuttle, the complex steel structure of the launch tower is illuminated by numerous lights, casting a glow on the surrounding darkness. A tall metal ladder or crane arm extends from the right side of the shuttle towards the top of the tower. The overall scene conveys a sense of the massive engineering and infrastructure involved in space exploration.

先人が目指すところを  
振り返る  
メタ言語とプリプロセッサ

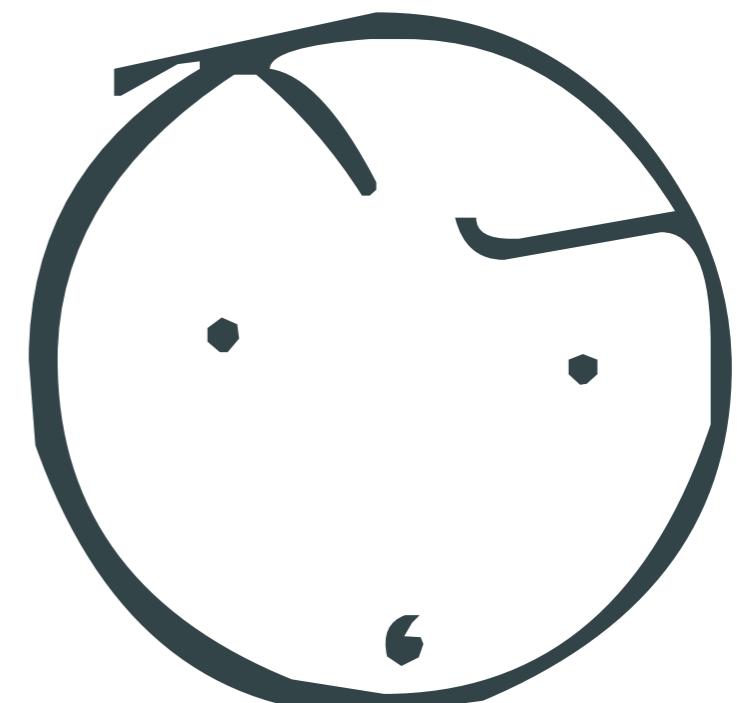
# メタ言語？プリプロセッサ？

## メタ言語

- ある言語を記述するための  
一段上の立場にある**専用の言語**

## プリプロセッサ

- 規則に従った文字置換(マクロ)など  
**下準備**のような処理を行うソフト



# Sass : 高機能なメタ言語

A meta-language on top of CSS

CSSの上位にあるメタ言語

3種の神器variable · mixin · extend

プログラミング言語並みの制御構文



{style with attitude}

# LESS : 自然なプリプロセス

The dynamic stylesheet language

動的性質を加えたスタイルシート

CSSの既存文法を活かしている

結果として学習コストが低い



メタ言語  
寄り

プリプロセッサ  
寄り





# Shootout?



# Sassの目標

構造化のために  
強力な高水準言語を作る

# LESSの目標

CSS文法を活かして  
シンプルな機能を加える

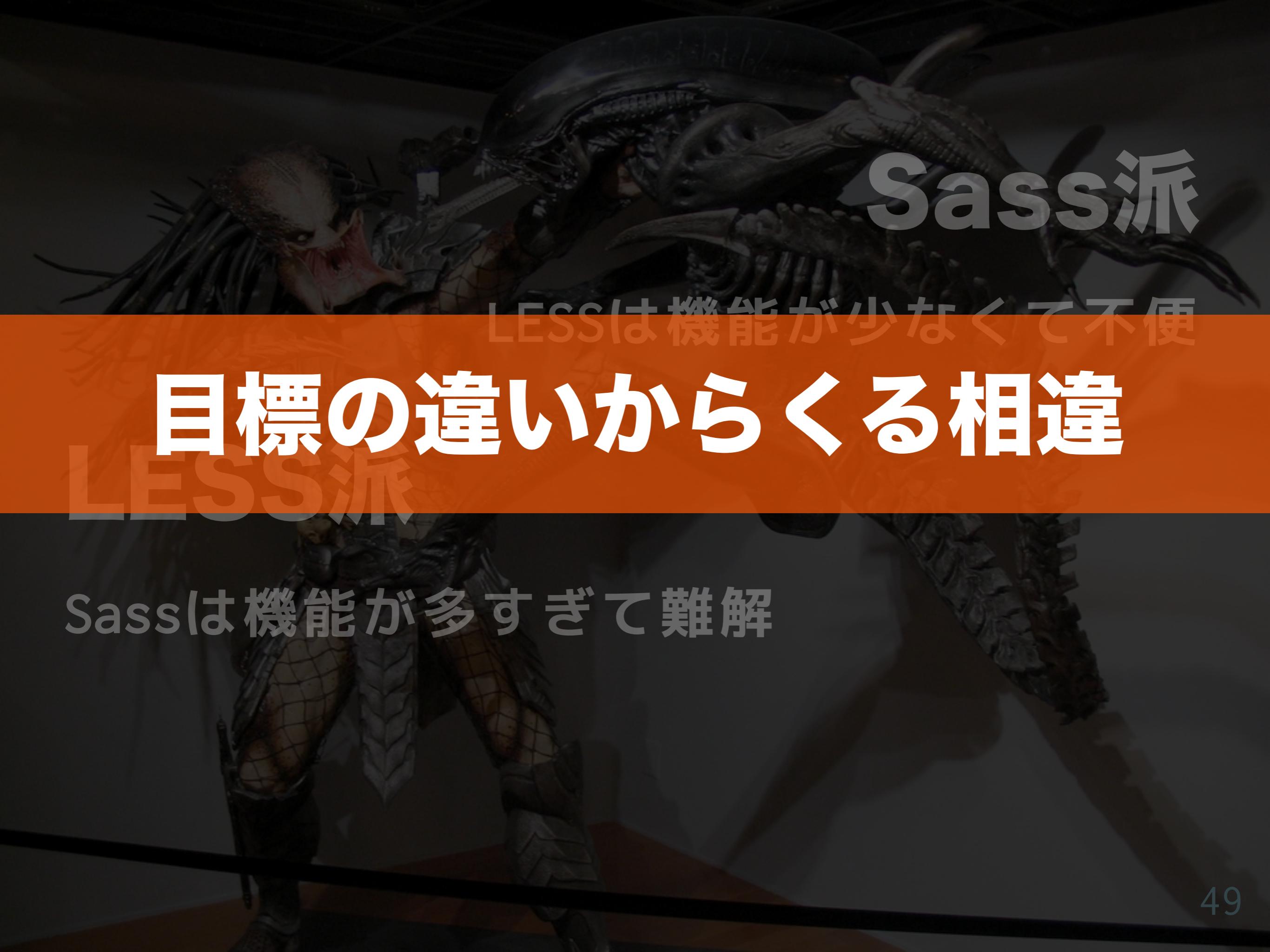


Sass派

LESSは機能が少なくて不便

LESS派

Sassは機能が多くすぎて難解



Sass派

LESSは機能が少なくて不便

目標の違いからくる相違

LESS派

Sassは機能が多くて難解

# CSSらしくて 高機能なら良いのか？

SassとLESSの溝を埋める解決

# Stylusが目指す CSSプリプロセッサ

プリプロセッサが抱える問題の見直し



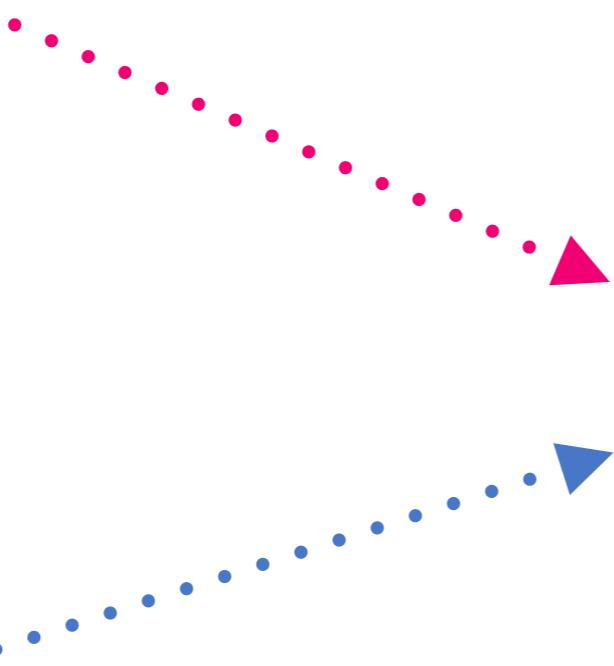
Sass.

{style with attitude}

cssっぽさ

{less}

十分な機能



良いトコ取り

stylus

# Stylus : CSSらしい高機能言語

the most CSS-like high-level stylesheet language

最もCSSに近しいSyntaxを備えている  
@extendも制御構文もある  
プラグイン機構による拡張性もある



# CSSらしい高機能言語 これでみんな幸せ？

まだ解決していない問題へ →

# Stylusが立ち向かう先

汚いCSSは良くならない  
→ OOCSSやSMACSSは基本

高機能に由來した複雑さ  
→ 便利さと引き換えの新たな課題



# 汚いCSSは良くならない

If you write poor CSS, a preprocessor won't make it suck less

プリプロセッサはCSSを隠蔽しない

出力されるCSSの品質は、  
コンパイル前の記述に強く依存する



[CSS Pre-Processors: Stylus, Less & Sass](#)

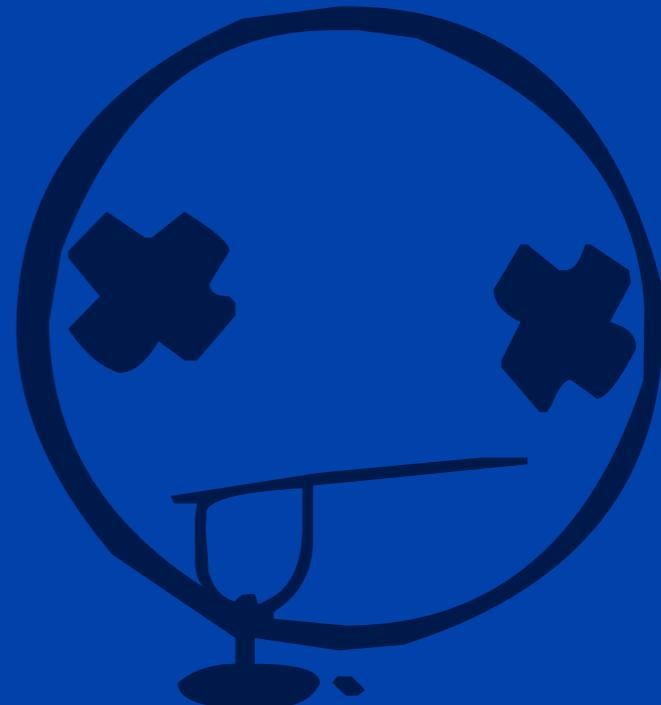
<https://speakerdeck.com/bermonpainter/css-pre-processors-stylus-less-and-sass?slide=23>

# 高機能に由來した複雑さ

The pollution caused by the preprocessor

プリプロセッサの固有機能による汚染

多彩な機能に溺れ・振り回される  
複雑性という新しい問題を持ち込んだ



# 2つの問題の混在

CSSの問題

→ 本来的な設計 (OOCSS, SMACSS)

プリプロセッサの問題

→ 複雑な抽象化による利便の提供



# 透過的 mixin のアプローチ

Transparent mixins are unique to Stylus, and are a incredibly powerful way enhance your stylesheets.

[via. Transparent mixins – Try Stylus](#)

# ベンダープレフィックスの補完にみる各種のmixin

.SCSS

```
@mixin box-sizing($arg) {  
  -webkit-box-sizing: $arg;  
  -moz-box-sizing: $arg;  
  box-sizing: $arg;  
}  
.container {  
  @include box-sizing(border-box);  
}
```

.styl

```
box-sizing(arg) {  
  -webkit-box-sizing: arg;  
  -moz-box-sizing: arg;  
  box-sizing: arg;  
}  
.container {  
  box-sizing: border-box;  
}
```

.less

```
.box-sizing(@arg) {  
  -webkit-box-sizing: @arg;  
  -moz-box-sizing: @arg;  
  box-sizing: @arg;  
}  
.container {  
  .box-sizing(border-box);  
}
```

透過的mixin

||  
Stylusはプロパティの  
ように見えるから  
透明で自然

## 透過的font-sizeの再定義 - rem利用のフォールバック

```
font-size(n) {  
  if unit(n) is 'px' {  
    font-size: n;  
    font-size: (remove-unit(n) / 10) rem;  
  } else {  
    error('Use px as unit')  
  }  
}
```

**mixin.styl**

複雑な抽象化レイヤー

```
body {  
  font-size: 16px;  
}
```

**style.styl**

ほとんどCSSなレイヤー

```
body {  
  font-size: 16px;  
  font-size: 1.6rem;  
}
```

**output.css**

via. font-size using rem with fallback in Stylus | Why Do We Fall?

# CSSを本的に 集中して書くレイヤー



複雑さに気を取られず  
CSSらしくスタイル定義に専念



style.styl



mixin.styl

透過的 mixin を通して  
拡張したプロパティを提供



プリプロセッサで  
密かに抽象化するレイヤー

## nibの例(再出) ピクセル密度に応じた画像(@2x)の展開

```
@import 'nib';  
#logo {  
  image: 'img/logo.main.png' 50px 100px  
}  
.styl
```

```
#logo {  
  background-image: url("img/logo.main.png");  
}  
@media all and (-webkit-min-device-pixel-ratio: 1.5) {  
  #logo {  
    background-image: url("img/logo.main@2x.png");  
    -webkit-background-size: 50px 100px;  
    background-size: 50px 100px;  
  }  
}  
.CSS
```

## nibの例(再出) 便利なプロパティ = mixinの追加

```
@import 'nib';          .styl  
  
#back-to-top {  
    fixed: bottom right;  
}  
  
#table-of-contents {  
    absolute: top 20px left 20px;  
}  
  
.description {  
    overflow: ellipsis;  
}
```

```
#back-to-top {          .css  
    position: fixed;  
    bottom: 0;  
    right: 0;  
}  
  
#table-of-contents {  
    position: absolute;  
    top: 20px;  
    left: 20px;  
}  
  
.description {  
    white-space: nowrap;  
    overflow: hidden;  
    -o-text-overflow: ellipsis;  
    text-overflow: ellipsis;  
}
```

# **混在した問題の レイヤーを分けてクリアに**

**高機能でありつつ、慎重にCSSを書ける**



the most CSS-like high-level stylesheet language

高機能かつ、ちゃんとCSSを書けるプリプロセッサ

# まとめ

この先のCSSプリプロセッサと選び方

# 賢いプリプロセッサ選びを!

文法 - The Syntax

使える機能 - Available features

コミュニティ - Community

実行環境 - Ruby vs. JS

本当に必要なものは？

既存ワークフローとの相性は？

Deep Dive Into CSS Preprocessors

<http://www.slideshare.net/verekia/deep-dive-into-css-preprocessors>

# Sass.js → Stylus → ...

Sass.js期 (Sassが.scssをサポートする前)



Stylus初期 (indentation-styleのみサポート)



Stylus中期 (CSS-styleのサポート)



Stylus後期 ... reworkの誕生(プレーンCSSへの回帰)

# rework

arbitrary CSS preprocessing library for node.js --  
(vendor prefixes, custom properties, image  
inlining, easing functions, media macros)

<https://github.com/visionmedia/rework>

# Thank you! おしまい

↑ <http://aho.mu>

🐦 [@ahomu](#)

🐱 [github.com/ahomu](#)



special thanks @cssradar

# Photo Credits...thanks!

1. Road to leh <http://www.flickr.com/photos/cblue98/6907507781>
2. Sleeping 猫 <http://www.flickr.com/photos/hansel5569/7687221498>
3. Space Shuttle Endeavour on Pad39A <http://www.flickr.com/photos/jurvetson/1050733503>
4. space <http://www.flickr.com/photos/58782395@N03/5518991291>
5. Solar Panels <http://www.flickr.com/photos/jeremylevinedesign/2903370723>
6. Good Friends <http://www.flickr.com/photos/ngmmemuda/4166182931>
7. Alien vs Predator <http://www.flickr.com/photos/steampirate/1056958115>