

Reboot Dreamweaver





CSS Nite LP51 「Reboot Dreamweaver」

Dreamweaverで

Sass を使うための勘所

 まぼろし 久保 知己
MABOROSHI Co., Ltd.



久保 知己

株式会社まぼろし
マークアップエンジニア

HTML+CSSによるマークアップからCMSの構築まで、さまざまなWebサイトの制作に関わったあと、2014年にまぼろしに入社。

Adobe公式ブログ「Adobe Creative Station」で、プラグインの「Emmet」やDreamweaverの連載を行っている。

主な著書に『いちばんよくわかるWebデザインの基本きちんと入門』（共著、SBクリエイティブ）、『現場のプロが教えるWeb制作の最新常識 知らないと困るWebデザインの新ルール』（共著、MdN）など。

個人ブログ「Web Design KOJIKAI17」を運営。

DreamweaverでSassを使うための勘所

- Sassとは？
- Dreamweaverを使ったSassのコンパイル
- Dreamweaverで使えるSassライブラリ
- Sassを実践で使う

2007年に
Hamli v1.5の機能の一部として
Sassが登場

10th

Sass

Sassとは？

Sass

CSSメタ言語

CSSプリプロセッサ

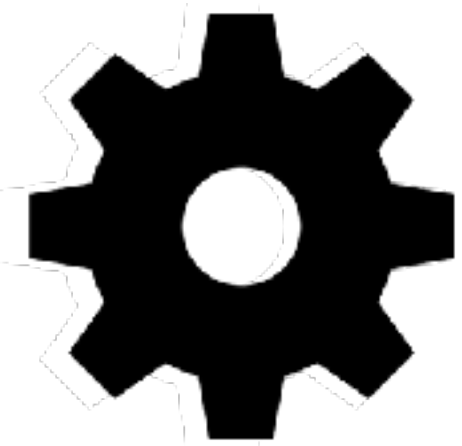


CSSメタ言語



CSSを拡張した言語

CSSプリプロセッサ



CSSメタ言語を処理するためのプログラム

CSSメタ言語の構文



インデント構文: Sass

```
.foo
  width: 980px
  margin: auto
  p
    margin: 1em 0
  a
    color: #000
    text-decoration: none
```

ブロック形式構文: SCSS

```
.foo {  
  width: 980px;  
  margin: auto;  
  p {  
    margin: 1em 0;  
  }  
  a {  
    color: #000;  
    text-decoration: none;  
  }  
}
```

**CSSメタ言語で
できること**



子孫セレクタを楽に書きたい: CSS

```
.about .text-area {  
  width: 600px;  
  margin: auto;  
}  
@media (max-width: 640px) {  
  .about .text-area {  
    width: auto;  
    margin: 20px;  
  }  
}  
.about a {  
  color: #aaa;  
}  
.about a:hover {  
  text-decoration: underline;  
}
```

ネスト: SCSS

```
.about {
  .text-area {
    width: 600px;
    margin: auto;
    @media (max-width: 640px) {
      width: auto;
      margin: 20px;
    }
  }
}
a {
  color: #aaa;
  &:hover {
    text-decoration: underline;
  }
}
```

色を一括で変更したい: CSS

```
.foo {  
  color: #000;  
}  
.bar {  
  color: #fb2234;  
}  
.button {  
  color: #fb2234;  
}  
.button-large {  
  color: #fb2234;  
}
```

変数: SCSS

```
$black: #000;  
$red: #fb2234;  
  
.foo {  
  color: $black;  
}  
  
.bar {  
  color: $red;  
}  
  
.button {  
  color: $red;  
}  
  
.button-large {  
  color: $red;  
}
```


複雑なCSSを管理したい: CSS

```
.hero__item--1 {
  animation-name: hero__item--1;
  opacity: 1;
}
@keyframes hero__item--1 {
  0% {opacity: 1;}
  22% {opacity: 1;}
  25% {opacity: 0;}
}

.hero__item--2 {
  animation-name: hero__item--2;
}
@keyframes hero__item--2 {
  22% {opacity: 0;}
  25% {opacity: 1;}
  47% {opacity: 1;}
  50% {opacity: 0;}
}
```

```
.hero__item--3 {
  animation-name: hero__item--3;
}
@keyframes hero__item--3 {
  47% {opacity: 0;}
  50% {opacity: 1;}
  72% {opacity: 1;}
  75% {opacity: 0;}
}

.hero__item--4 {
  animation-name: hero__item--4;
}
@keyframes hero__item--4 {
  72% {opacity: 0;}
  72% {opacity: 0;}
  75% {opacity: 1;}
  97% {opacity: 1;}
}
```

Mixins · 制御構文: SCSS

```
@mixin fade($classes) {
  $length: length($classes);
  $percent: percentage(1 / $length);
  @each $class in $classes {
    $index: index($classes, $class);
    $step1: ($percent * $index) - $percent;
    $step2: ($percent * $index) - 3;
    $step3: $percent * $index;
    $step0: $step1 - 3;

    .#{$class} {
      animation-name: #{$class};
      @if $index == 1 {
        opacity: 1;
      }
    }
  }
  @keyframes #{$class} {
    @if $index == $length {
      #{ $step0 } {
        opacity: 0;
      }
    }
  }
}
```

```
@if $index > 1 {
  #{ $step0 } {
    opacity: 0;
  }
}
#{ $step1 } {
  opacity: 1;
}
#{ $step2 } {
  opacity: 1;
}
@if $index != $length {
  #{ $step3 } {
    opacity: 0;
  }
}
}
}
}
```

Includes: SCSS

```
$fade-duration: 10s;  
$fade-classes: (  
  hero__item--1,  
  hero__item--2,  
  hero__item--3,  
  hero__item--4  
);  
  
@include fade($fade-classes);
```

パーセントやrgba関数など楽したい: CSS

```
.foo {  
  width: 20%;  
  color: rgba(0, 0, 0, 0.5);  
}
```

```
.modal {  
  position: absolute;  
  z-index: 3;  
}
```

関数: SCSS

```
$z: (  
  header,  
  nav,  
  modal  
);  
  
.foo {  
  width: percentage(1 / 5);  
  color: rgba(#000, 0.5);  
}  
  
.modal {  
  position: absolute;  
  z-index: index($z, modal);  
}
```

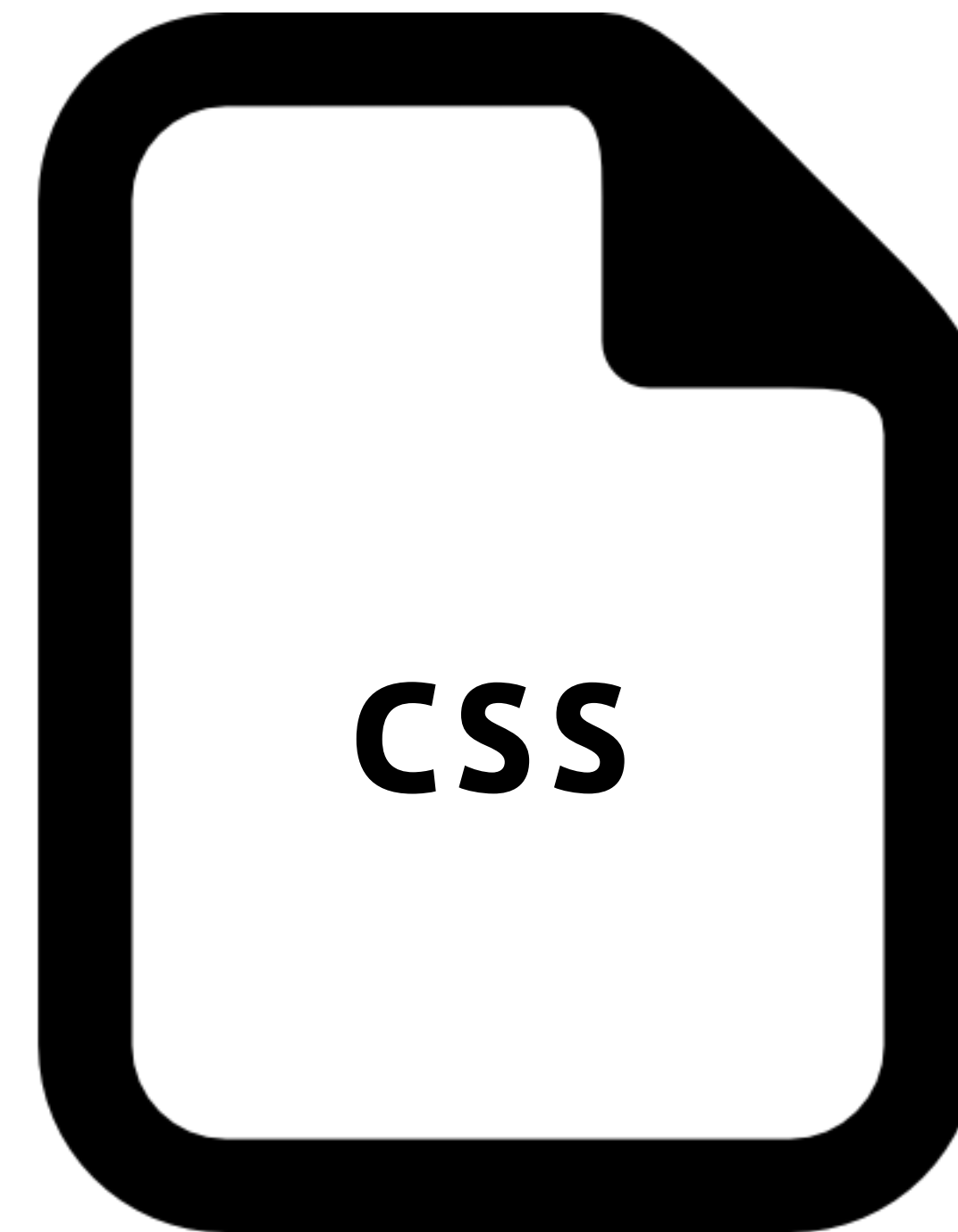
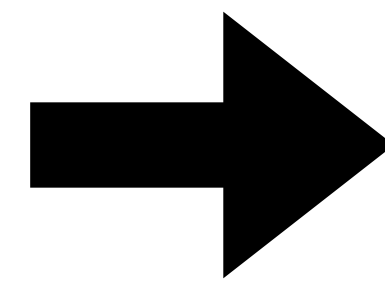
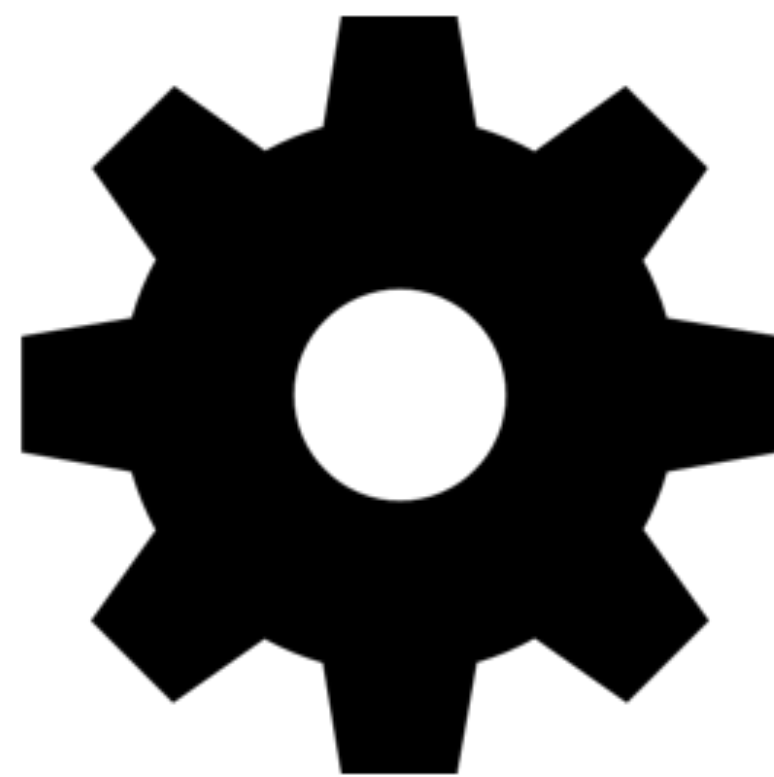
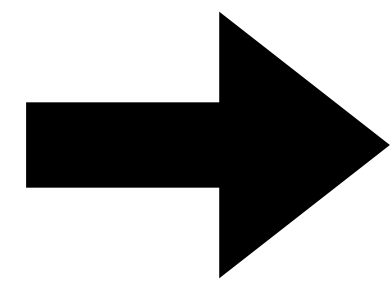
CSSプリプロセッサ



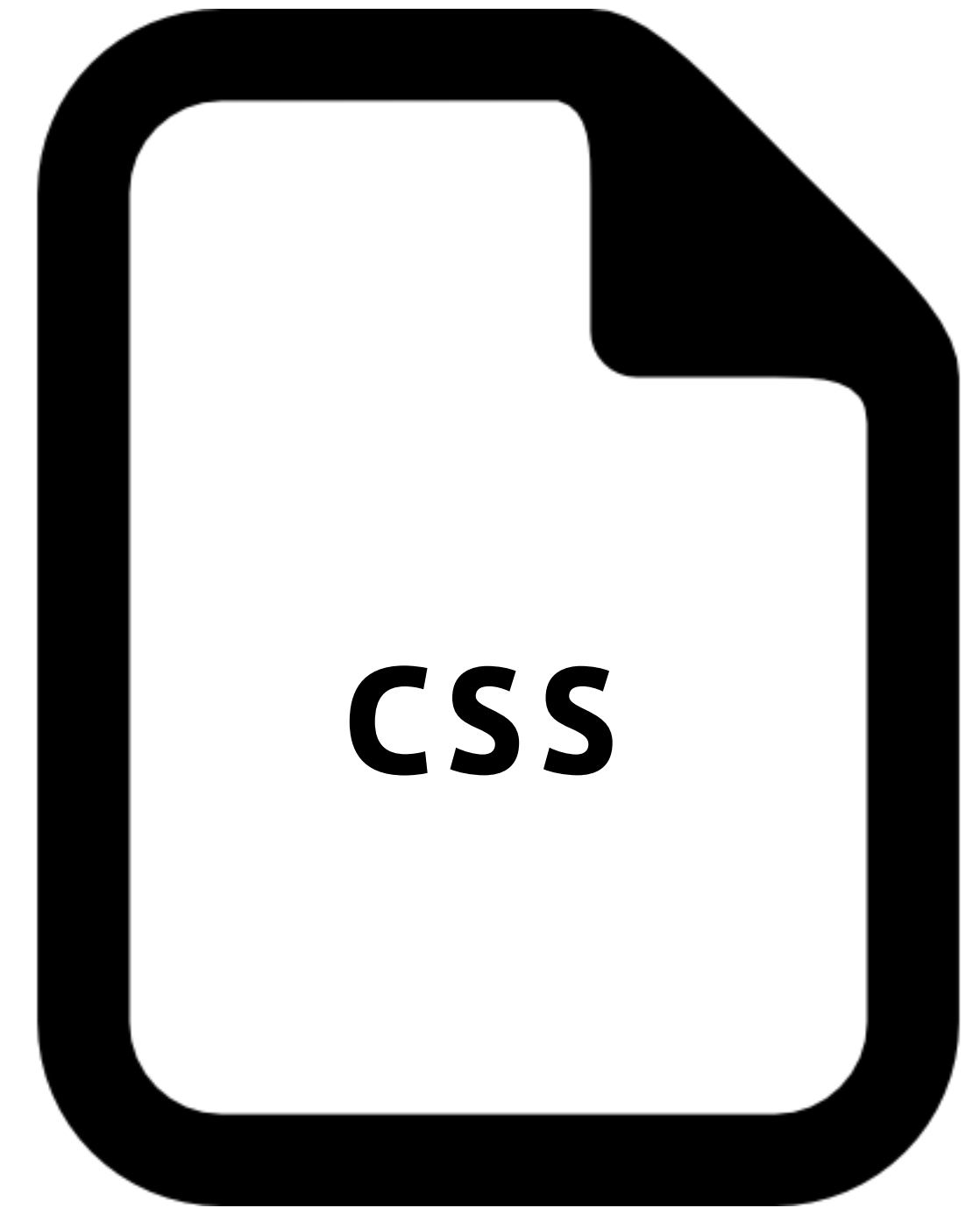
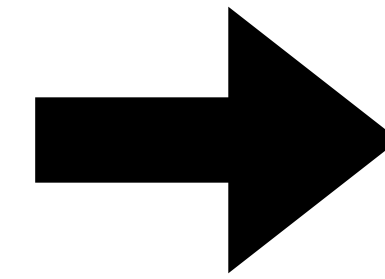
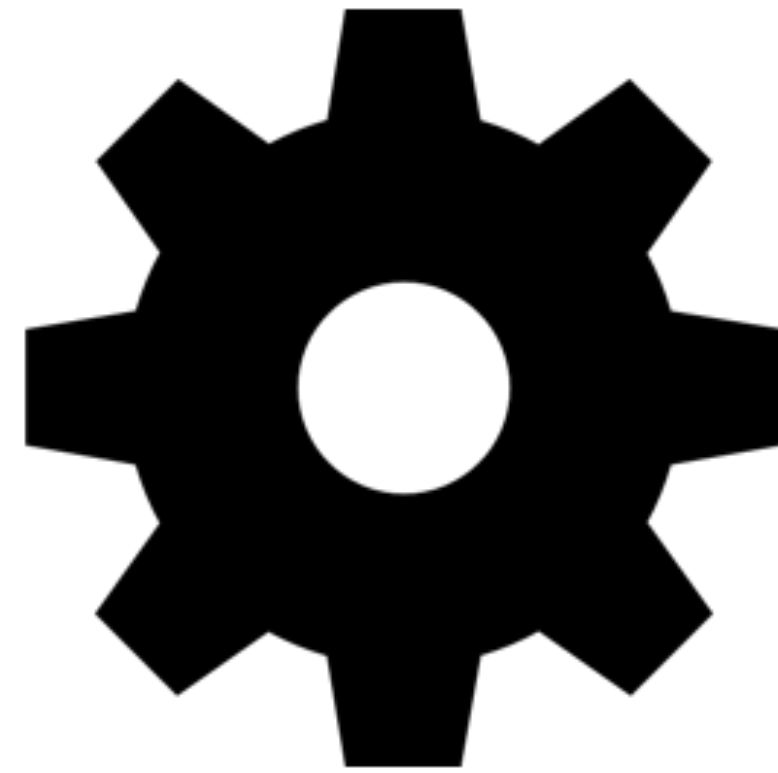
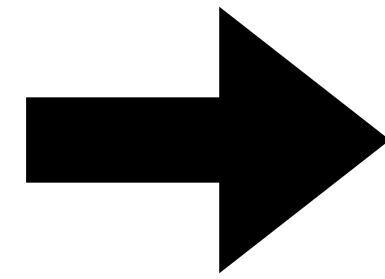
CSSメタ言語の コンパイル（変換）



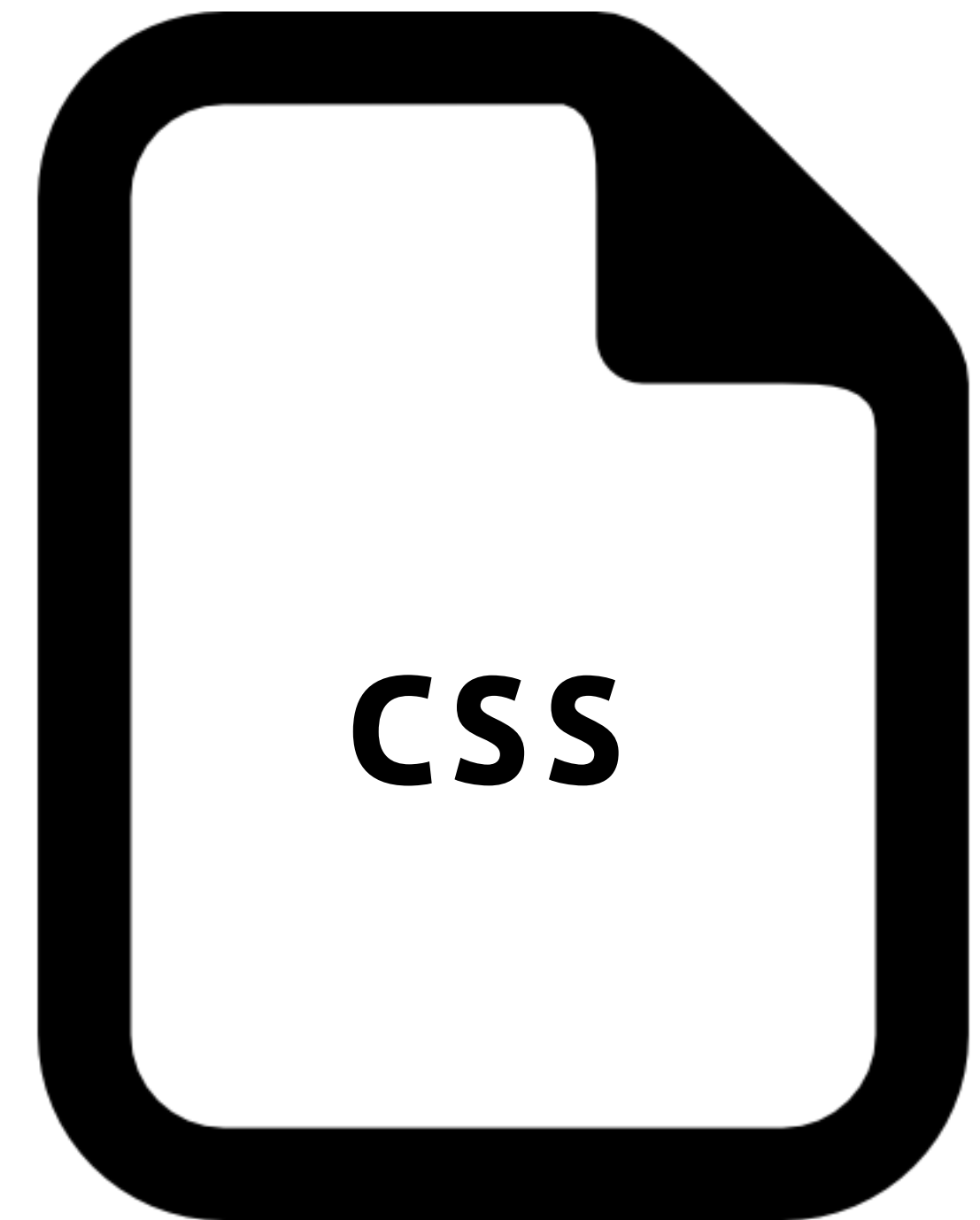
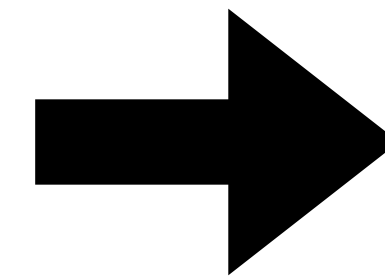
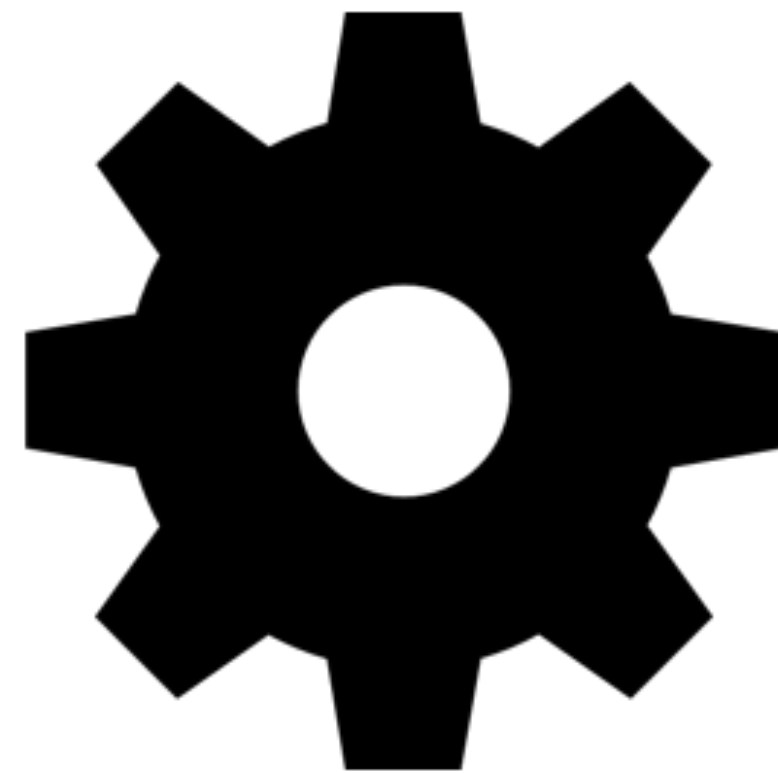
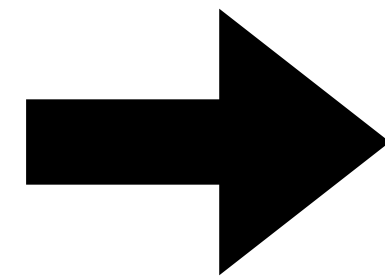
CSSメタ言語



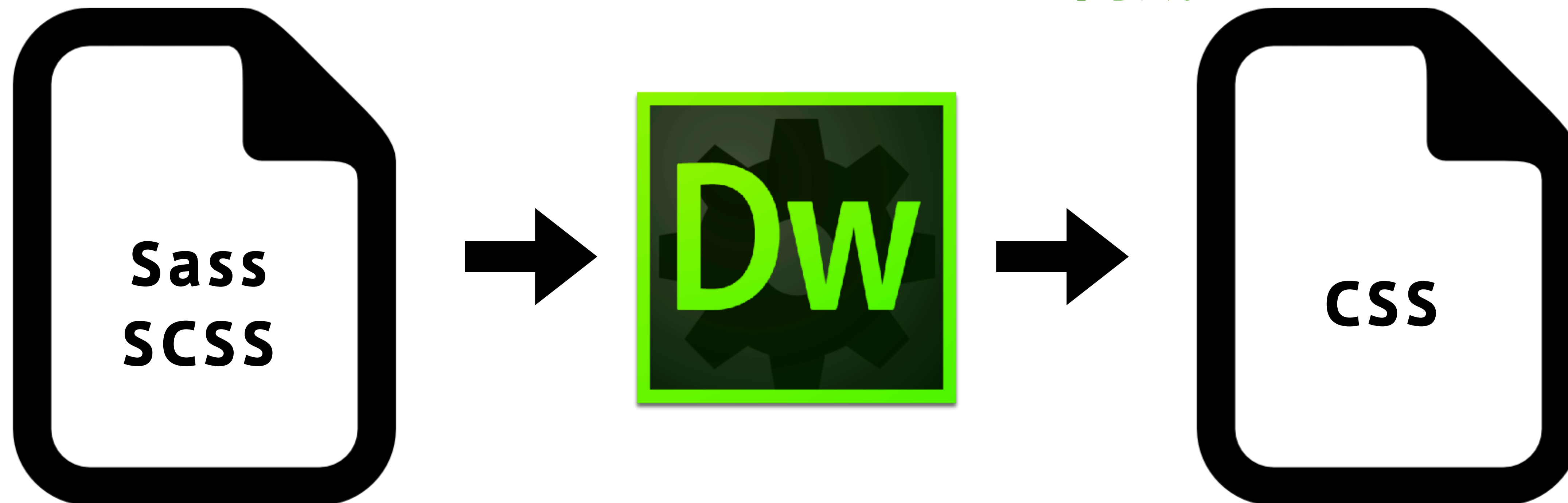
CSSプリプロセッサ



CSSファイル



Dreamweaver CC 2017に CSSプリプロセッサが内蔵



**Dreamweaverを
インストールするだけで、
CSSプリプロセッサの
環境が構築される**

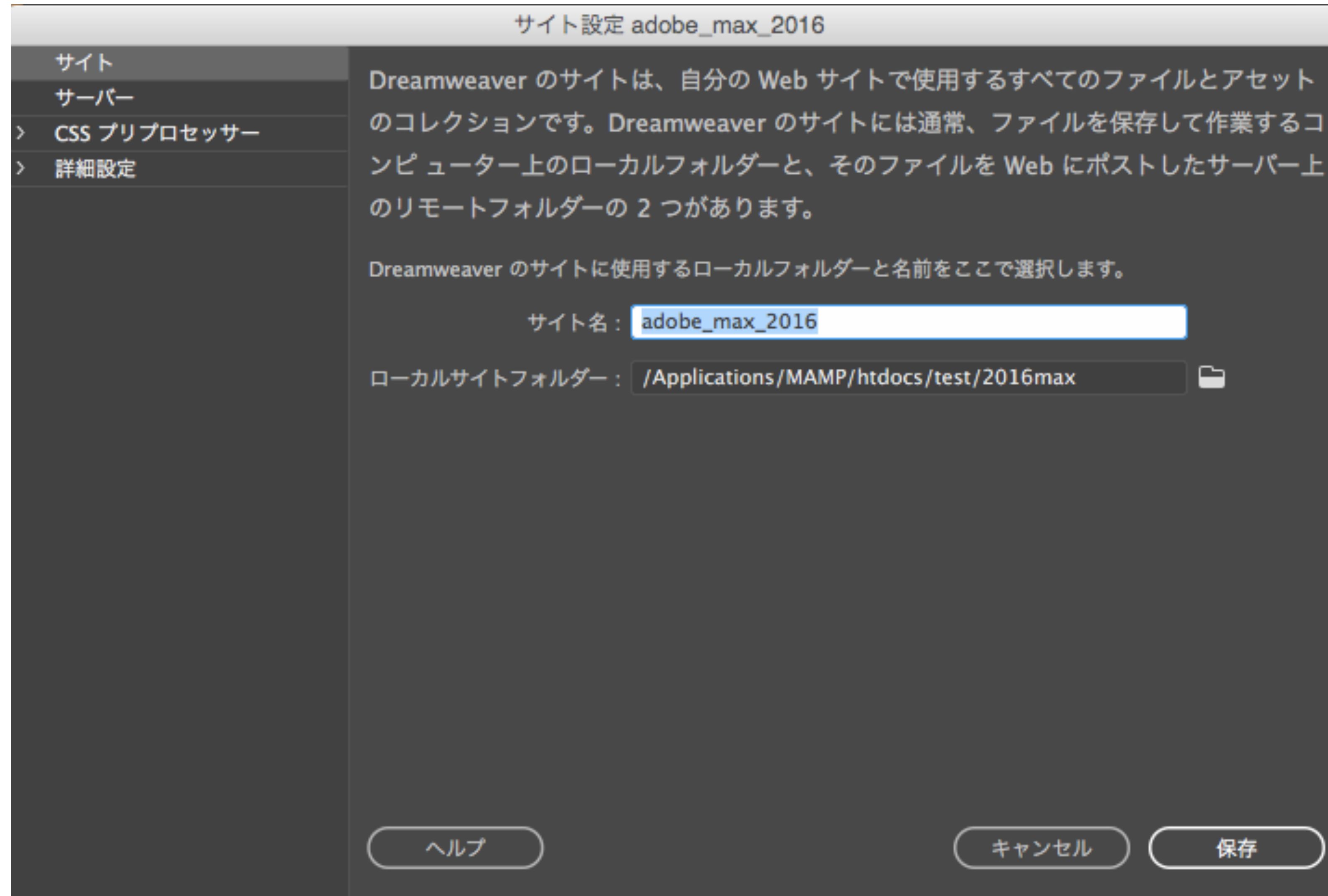
Dreamweaverを使った Sassのコンパイル



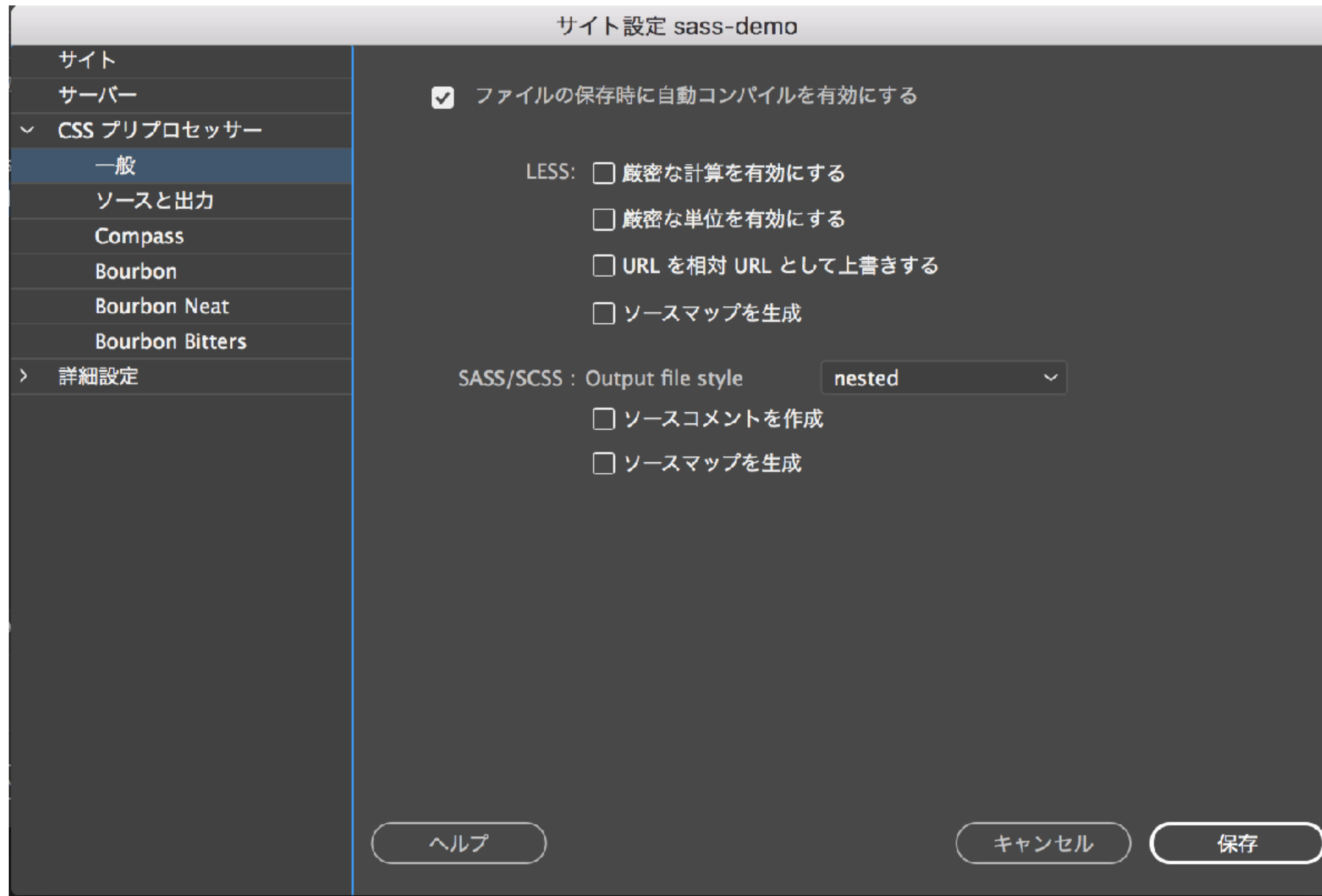
3ステップで簡単設定



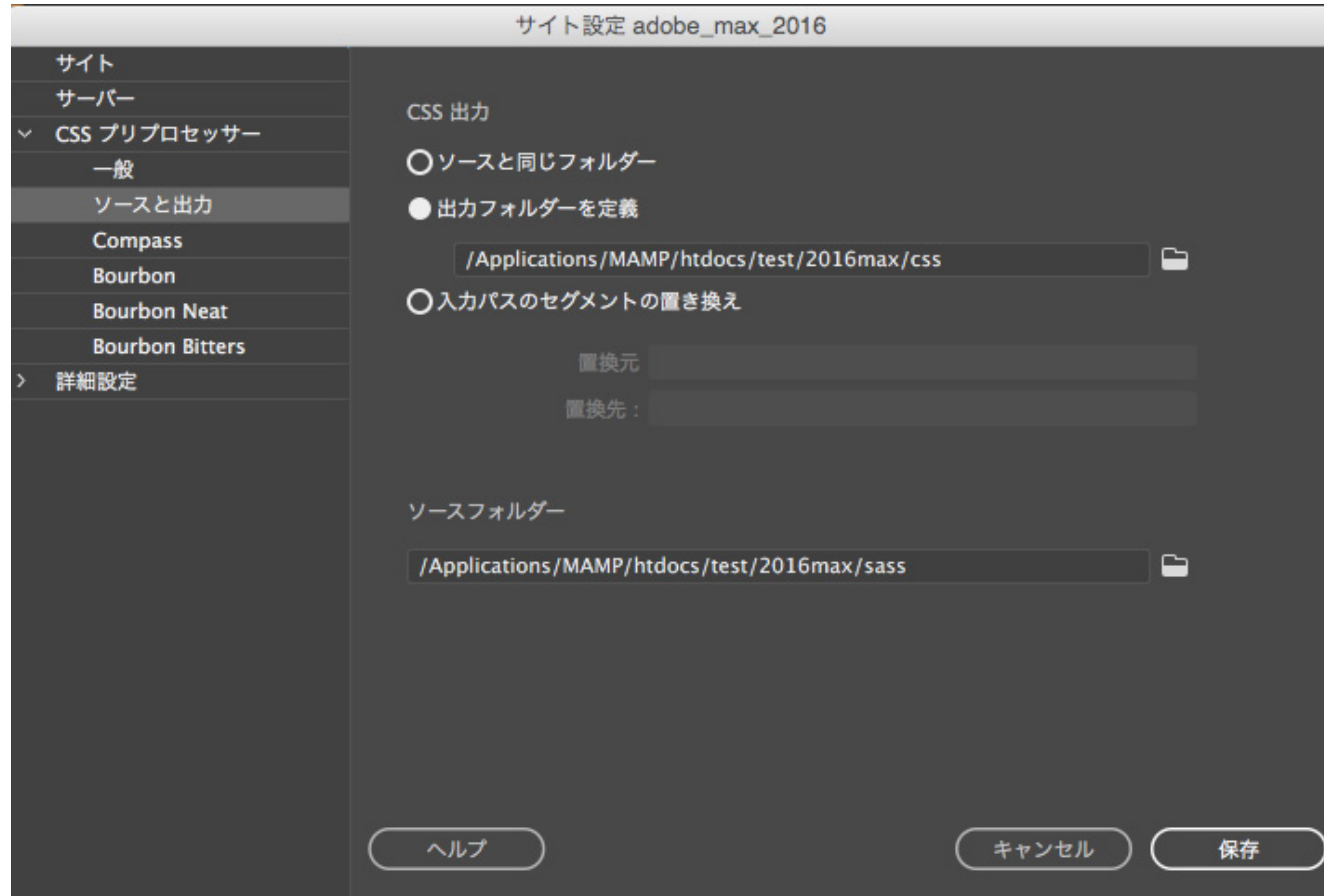
ステップ1: サイト設定を開く



ステップ2: コンパイルが有効か確認



ステップ3: 入出力のフォルダーの指定



DEMO

Sassのコンパイル設定



サイト設定 sass-demo

サイト

サーバー

› CSS プリプロセッサ

› 詳細設定

Dreamweaver のサイトは、自分の Web サイトで使用するすべてのファイルとアセットのコレクションです。Dreamweaver のサイトには通常、ファイルを保存して作業するコンピュータ上のローカルフォルダーと、そのファイルを Web にポストしたサーバー上のリモートフォルダーの 2 つがあります。

Dreamweaver のサイトに使用するローカルフォルダーと名前をここで選択します。

サイト名: sass-demo

ローカルサイトフォルダー: /Applications/MAMP/htdocs/test/sass-demo



ヘルプ

キャンセル

保存

DEMO

SCSSファイルの作成とコンパイル



ファイル CC ライブラリ 挿入 CSS デザイナー

sass-demo [サーバーを定義]

ローカルファイル ↑

- サイト - sass-demo (Macintosh HD:Applicati...
- sass

DOM アセット スニペット

このパネルには、HTML ドキュメントの DOM 構造が表示されます

最近使用したファイル

CC ファイル

クイックスタート

スターターテンプレート

新規... ⌘N

開く... ⌘O

			
HTML ドキュメント	CSS ドキュメント	JS ドキュメント	PHP ドキュメント
			
SCSS ドキュメント	Sass ドキュメント	LESS ドキュメント	XML ドキュメント

ADOBE STOCK から 10 点の画像を無料で利用

検索 [検索]

Dreamweaverのコンパイルの詳細設定

- CSSの出カスタイル
- CSSにSassの行番号を含めたコメント追加
- ソースマップの生成

ファイルの保存時に自動コンパイルを有効にする

LESS: 厳密な計算を有効にする

厳密な単位を有効にする

URL を相対 URL として上書きする

ソースマップを生成

SASS/SCSS : Output file style

nested

ソースコメントを作成

ソースマップを生成

Sassのコンパイル設定は簡単！

ステップ1: サイト設定を開く

ステップ2: コンパイルが有効か確認

ステップ3: 入出力のフォルダの指定

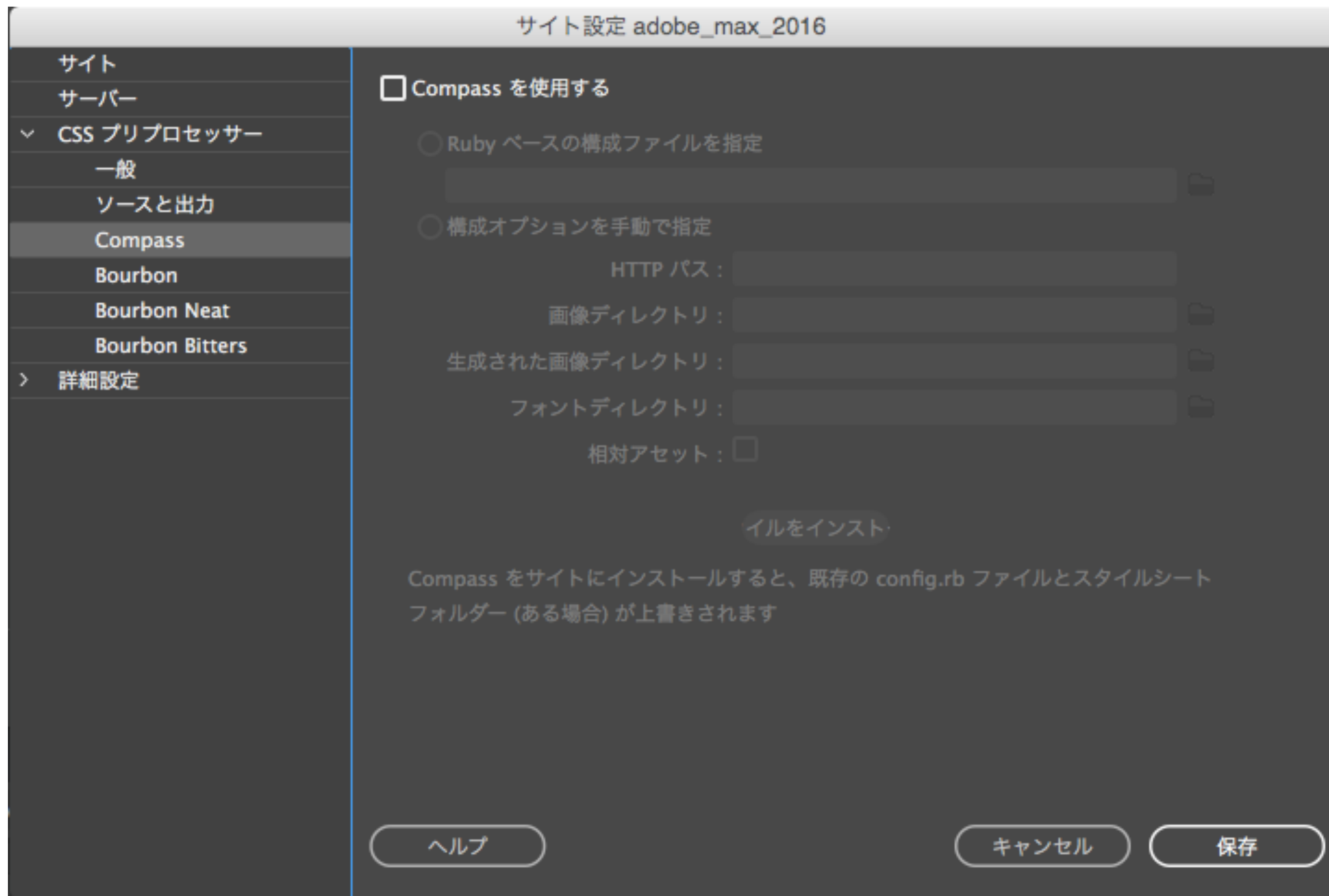
Dreamweaverで使える Sassライブラリ



**Dreamweaverに梱包されている
Sassライブラリ**



Sassライブラリ



Sassライブラリ



リセットCSSやMixins、関数などの
総合的なフレームワーク



セマンティックなグリッドを
構築するためのフレームワーク



Web制作をサポートするMixins集



タイポグラフィやボタンなどを
構築するためのフレームワーク

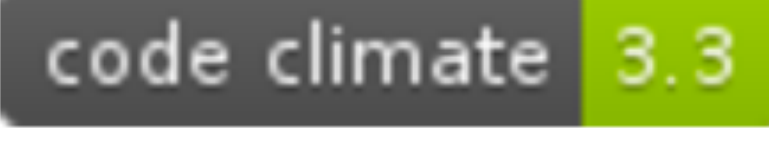
それ本当に使っていいの？



Compass Stylesheet Authoring Framework

Deprecated: Compass is no longer supported.

Build Status: 

Code Quality: 

Resources

- [Compass Homepage](#)

Sassライブラリの注意



Compassを開発しているGitHubのREADMEに、
しれっと「**Compass is no longer supported.**」が追加



Compassの開発が中止
技術的負債になるので注意

Sassライブラリの注意



Bourbon

Dreamweaverに梱包されているBourbon **v4.2.6**



開発中の**v5**から一部のMixinsが利用不可になる

Deprecations

- All vendor prefixing mixins have been removed. We recommend using a more robust and maintainable solution like [Autoprefixer](#).
- The `em()` and `rem()` mixins have been deprecated.
- The `$monospace` font stack variable has been deprecated in favor of new `$consolas`, `$courier-new` and `$monaco` variables.
- The `triangle()` mixin has been deprecated.

Maintenance

- The `strip-units` function is now `strip-unit`.
- The `size()` mixin now requires a comma-separated argument list: `@include size(1em, 2em);`.
- We restructured Bourbon's directory tree to be more succinct and reduce possible conflicts.

Sassライブラリの注意



Dreamweaverに梱包されているNeat **v1.7.3**

Sassライブラリの注意






Dreamweaverに梱包されているNeat **v1.7.3**






2007年2月17日に **v2.0.0** がリリース

設定ファイルやMixinsが、ガラッと変更

Sassライブラリの注意

	Dw CC 2017	公式
 compass	v1.0.3	v1.0.3
 Bourbon	v4.2.6	v4.3.6 v5 beta
 Neat	v1.7.3	v2.0.0

Sassライブラリの注意

	Dw CC 2017と公式の互換性	
	○	
	v4 ○	v5 △
	✕	

Sassのライブラリのバージョンを
指定できるようになればいいけど

**Dreamweaverのバージョンが上がると
ライブラリのバージョンも上がる**

安全にライブラリを使うには？



公式サイトからダウンロード



Bourbon



Neat

DEMO

Bootstrap Sassをダウンロード





Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

Download Bootstrap

Currently v3.3.7

Is your cloud
working hard?

Be Sure

Do You Want a
Transparent Cloud
Environment? Our Experts
Can Help!

DEMO

**Bootstrap Sassを
Dreamweaverに移動**



bootstrap-sass-3.3.7

検索

名前	変更日	サイズ	種類
▶ assets	今日 0:00	--	フォルダ
bootstrap-sass.gemspec	2016年7月26日 4:58	1 KB	Ruby
bower.json	2016年7月26日 4:58	741 バイト	JSON
CHANGELOG.md	2016年7月26日 4:58	8 KB	Markdown
composer.json	2016年7月26日 4:58	854 バイト	JSON
CONTRIBUTING.md	2016年7月26日 4:58	3 KB	Markdown
eyeglass-exports.js	2016年7月26日 4:58	147 バイト	JavaScript
Gemfile	2016年7月26日 4:58	187 バイト	テキスト...app 書類
▶ lib	今日 0:00	--	フォルダ
LICENSE	2016年7月26日 4:58	1 KB	テキスト...app 書類
package.json	2016年7月26日 4:58	1 KB	JSON
Rakefile	2016年7月26日 4:58	3 KB	テキスト...app 書類
README.md	2016年7月26日 4:58	13 KB	Markdown
sache.json	2016年7月26日 4:58	260 バイト	JSON
▶ tasks	2016年7月26日 4:58	--	フォルダ
▶ templates	2016年7月26日 4:58	--	フォルダ
▶ test	2016年7月26日 4:58	--	フォルダ

よく使う項目

- iCloud Drive
- Dropbox
- AirDrop
- アプリケーション...
- デスクトップ
- 書類
- ダウンロード
- ミュージック
- k
- Creative Cl...

デバイス

- BOOTCAMP
- リモートデ...

共有

- nasne-bb5...

Macintosh HD > ユーザ > k > デスクトップ > bootstrap-sass-3.3.7

ファイル操作が完了しました。

DOM アセット スニペット

このパネルには、HTML ドキュメントの DOM 構造が表示されます

DEMO

**Dreamweaverで
Bootstrap Sassを使う**



ファイル CC ライブラリ 挿入 CSS デザイナー

sass-demo サーバーを定義

ローカルファイル ↑

- サイト - sass-demo (Macintosh HD:Applicati...
- css
- fonts
- index.html
- js
- sass
 - _bootstrap.scss
 - bootstrap
 - style.scss

1個のローカルアイテム(合計31290...)

DOM アセット スニペット

× style.scss

```
1 |
```

× index.html

ソースコード style.css style.css

0 50 100 150 200 250 300 350 400 450 500

Default Primary Success Info Warning Danger Link

body +

このパネルには、HTML ドキュメントの DOM 構造が表示されます

公式サイトからSassライブラリを
ダウンロードするのが安全

Sassを実践で使う

The logo consists of the letters 'DW' in a bold, white, sans-serif font, centered within a dark blue square. The square has a thin white border.

DW

Dreamweaverの機能と Sassの連携が厳しい



Dreamweaverの機能とSassの連携が厳しい

Dreamweaverの
CSSを上書きするような機能は
Sassファイルには反映されない

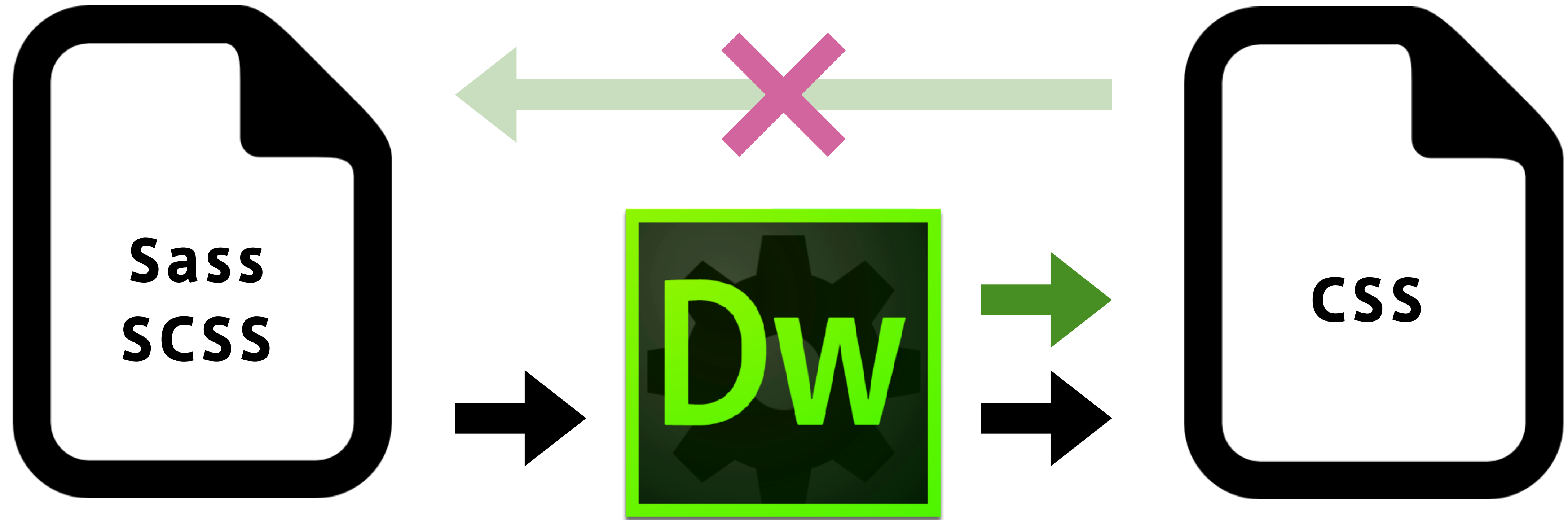
- CSSデザイナー
 - クイック編集
 - メディアクエリーバー
- など



Dreamweaverに期待している動作



Dreamweaverの現状



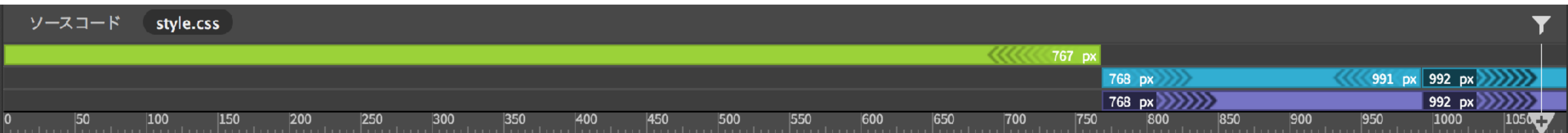
**Dreamweaverの
CSSを上書きする機能は
今後に期待**

メディアクエリをSassで管理



メディアクエリをSassで管理

メディアクエリローバーの代わりに
Sassの変数やMixinsを使って管理



メディアクエリをSassで管理: 変数を作成: SCSS

```
$screen-xs: 480px;  
$screen-sm: 768px;
```


メディアクエリをSassで管理: Mixinを作成: SCSS

```
$screen-xs: 480px;  
$screen-sm: 768px;  
  
@mixin mq($screen) {  
  @media (max-width: #{ $screen }) {  
    @content;  
  }  
}
```

メディアクエリをSassで管理: スタイルを記述: SCSS

```
$screen-xs: 480px;
$screen-sm: 768px;

@mixin mq($screen) {
  @media (max-width: #{ $screen }) {
    @content;
  }
}

.style {
  color: #000;
}
```

メディアクエリをSassで管理: スタイルを記述: SCSS

```
$screen-xs: 480px;
$screen-sm: 768px;

@mixin mq($screen) {
  @media (max-width: #{ $screen }) {
    @content;
  }
}

.style {
  color: #000;
  @include mq($screen-xs) {
    color: #f00;
  }
}
```

メディアクエリをSassで管理: CSS

```
.style {  
  color: #000;  
}  
@media (max-width: 480px) {  
  .style {  
    color: #f00;  
  }  
}
```

メディアクエリをSassで管理: 汎用性を高く

- 変数を打つのが面倒くさい
- max-widthとmin-widthが欲しい

```
$screen-xs: 480px;
$screen-sm: 768px;

@mixin mq($screen) {
  @media (max-width: #{ $screen }) {
    @content;
  }
}

.style {
  color: #000;
  @include mq($screen-xs) {
    color: #f00;
  }
}
```

メディアクエリをSassで管理: 汎用性を高く: SCSS

```
$screen: (  
  xs: 480px,  
  sm: 768px  
);  
  
@mixin mq($size, $width: max) {  
  $result: 0;  
  @if(type-of($size) == number) {  
    $result: $size;  
  } @else {  
    @each $key, $value in $screen {  
      @if($key == $size) { $result: $value; }  
    }  
  }  
  @if($width == min) { $result: $result - 1; }  
  @media (#{ $width }-width: #{ $result }) {  
    @content;  
  }  
}
```

メディアクエリをSassで管理: 汎用性を高く: SCSS

```
.style {  
  color: #000;  
  @include mq(xs) { // @media (max-width: 480px)  
    color: #f00;  
  }  
}
```

メディアクエリをSassで管理: 汎用性を高く: SCSS

```
.style {
  color: #000;
  @include mq(xs) { // @media (max-width: 480px)
    color: #f00;
  }
  @include mq(xs, min) { // @media (min-width: 479px)
    color: #f00;
  }
  @include mq(sm) { // @media (max-width: 768px)
    color: #f00;
  }
  @include mq(360px, min) { // @media (min-width: 359px)
    color: #f00;
  }
}
```


メディアクエリに限らず、
Mixinsや関数などで
Sassを賢く使う

色の管理には
Sassを使うか？

Dreamweaverの機能か？



色の管理

Sass



色の管理: Sass



```
$color-base: #000;  
$color-accent: #f00;  
$color-text: #333;  
$color-base: #fff;  
  
$colors: (  
  base: #000,  
  accent: #f00,  
  text: #333,  
  base: #fff  
);  
  
.style {  
  color: $color-text;  
}
```

色の管理: Sass

メリット

- 色を一括で変更しやすい

デメリット

- 変数を探すのが面倒くさい
- 何の色か、直感的にわかりにくい

```
$color-base: #000;
$color-accent: #f00;
$color-text: #333;
$color-base: #fff;

$colors: (
  base: #000,
  accent: #f00,
  text: #333,
  base: #fff
);

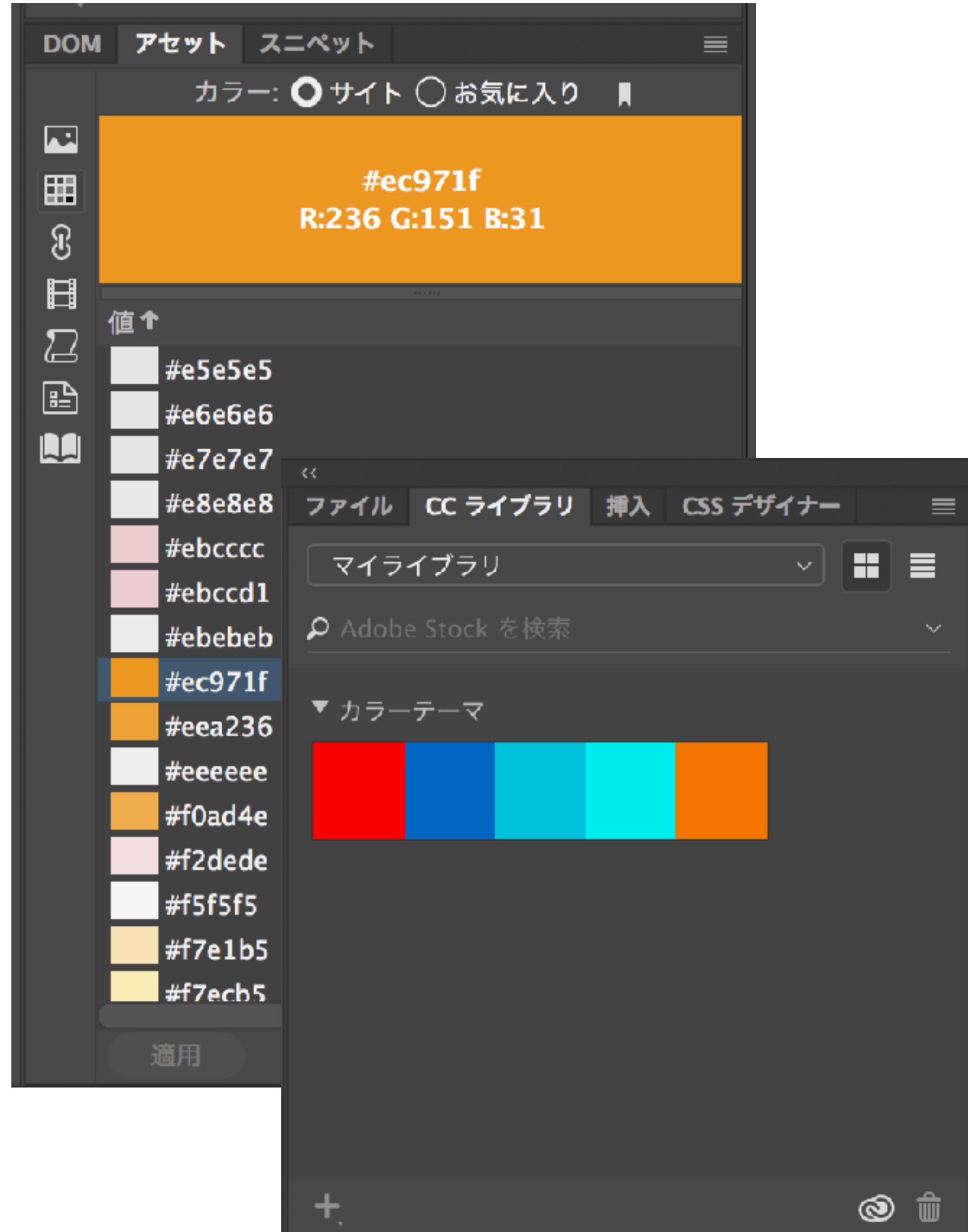
.style {
  color: $color-text;
}
```

色の管理: Dreamweaver

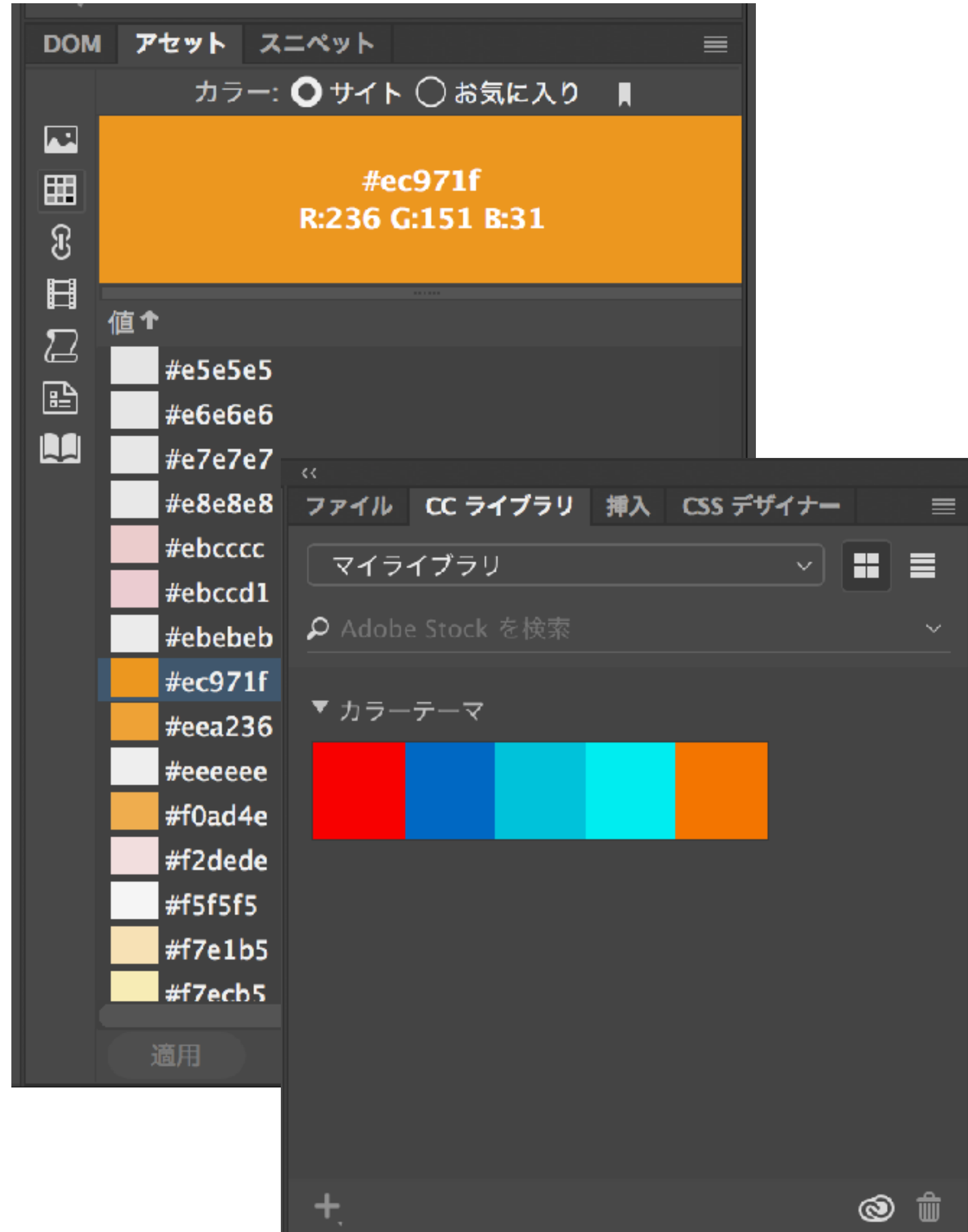
Sass



色の管理: Dreamweaver



色の管理: Dreamweaver



メリット

- 色が視覚的にわかりやすい
- 色の「お気に入り」でよく使う色を絞り込み
- CCライブラリのカラーテーマを使えば
Adobe製品間での色の管理ができる

デメリット

- 色の一括変更が難しい

サイトの性質によって
SassとDreamweaverの機能を
使い分けよう

まとめ

Sass



まとめ

- Sassは、今年で10周年
- Sassのライブラリは、注意して使う
- Mixinsや関数を使って、CSSをうまく管理



まとめ (2)

- CSSプリプロセッサの環境構築は Dreamweaverをインストールするだけ
- コンパイルの設定も簡単
- CSSを変更する機能は、今後に期待

The logo consists of the letters 'DW' in a large, bold, light green font. The letters are set against a dark grey rectangular background. A thin, light green border surrounds the top and left sides of the grey rectangle.



ありがとうございました