

BootstrapとSass入門

注意事項

本ハンズオンはSass入門が中心になります。

必要なもの

インターネットにつながる PC (Win/Mac)
やる気



株式会社インパス

井土 貴寛 @izuchy

iOS開発が好きなフロントエンドエンジニア

フロントエンドカンファレンス スタッフ

PHPカンファレンス関西 スタッフ

目指すところ

ハンズオンのゴール

生CSSしか触ったことのない人が
Sassをインストールして
使いたくなることを目指します

諸注意

本ハンズオンは課題に時間制限を設けています。
そのため理解に関わらず、進む場合があります。

スタッフ一同、精一杯お手伝いしますが
ご希望にお応えできない場合もあります。

タイムスケジュール

16:20

はじめに

← いまここ

16:25

Sassとは (5分)

16:30

Sass体験 (25分)

BootstrapのSassを触ってみよう (5分)

17:00

Sass とは

CSSのメタ言語

ある言語を生成するための言語

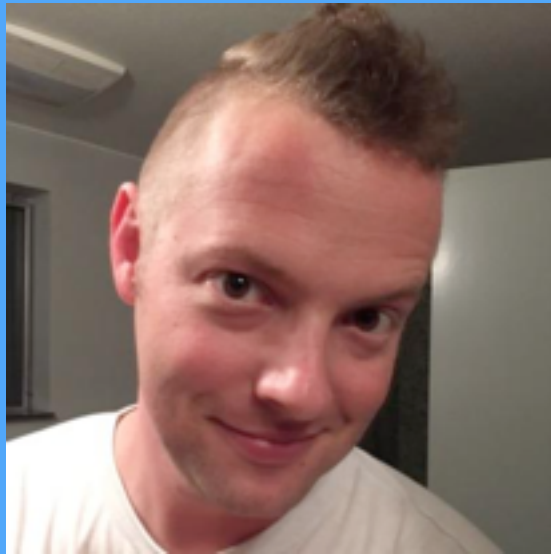
可読性UP

効率UP

CSS

Sass





@hcatlin

設計

Hampton Catlin

ハンプトン カトリン



@nex3

開発者

Natalie Weizenbaum

ネイサン・ワイゼンバウム

**Sass = Syntactically
Awesome
Stylesheets**

**Sass = 構文的に
すばらしい
スタイルシート**

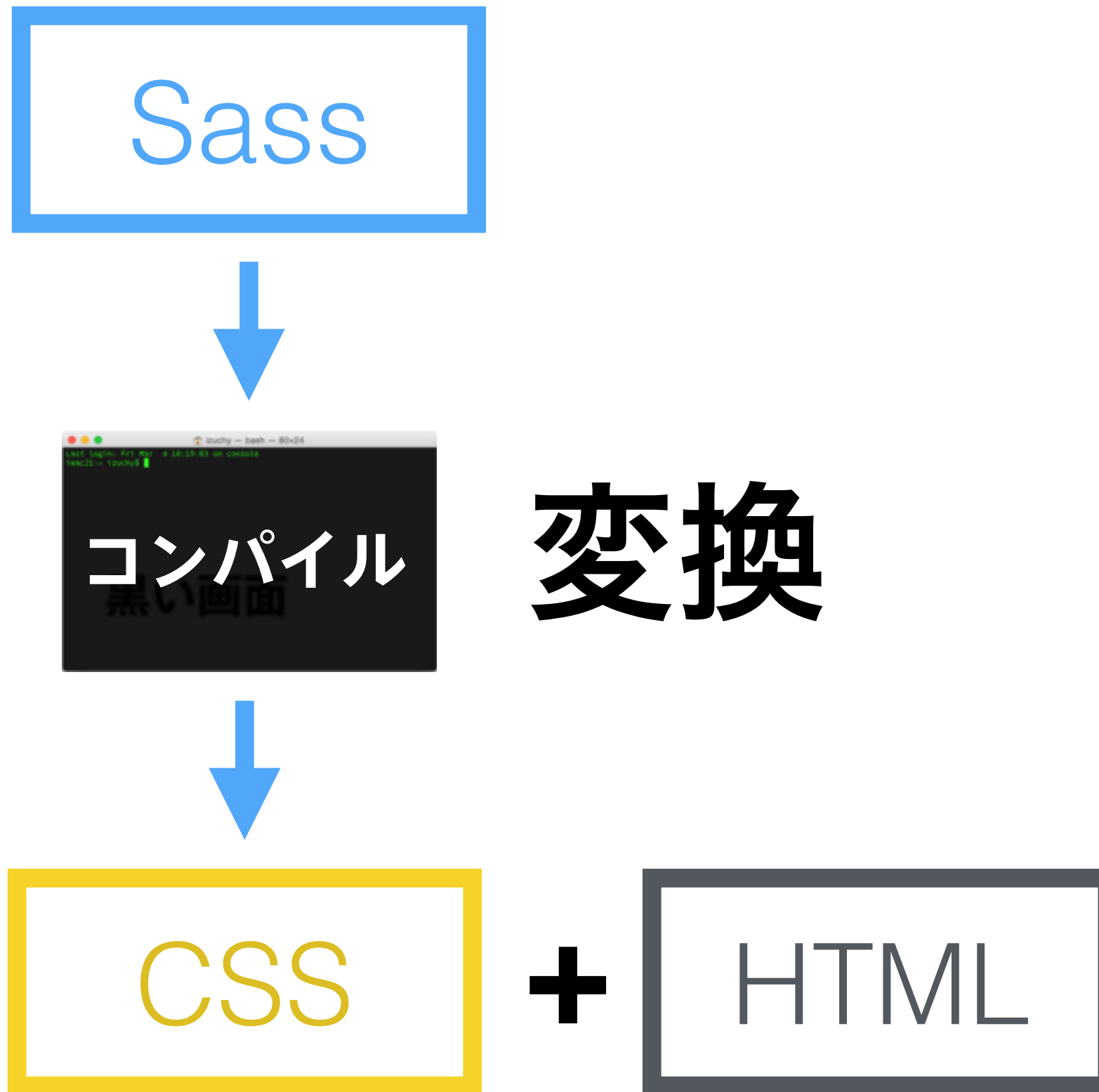
なにができるの？

- **CSSのネスト**
- **変数**
- **四則演算**
- **//の1行コメント**

- セレクタの再利用
- Sassファイルの結合
- 条件分岐や制御構文

などなど...

どうやって使うの？



Sassを学ぶ過程で
まっさきに直面する

？

ハテナ

Sass

だけど

SCSS!

Sass 公式サイト

<http://sass-lang.com/guide>

Variables

Think of variables as a way to store information that you want to reuse in your stylesheet. You can store things like colors, font stacks, or any CSS property that you want to reuse. Sass uses the `$` symbol to make something a variable.

SASS SYNTAX

```
$font-stack: Helvetica, sans-serif
```

```
$primary-color: #333
```

```
body
```

```
  font: 100% $font-stack
```

```
  color: $primary-color
```

Sass

SCSS

 twbs / **bootstrap-sass**

<> Code

! Issues 14

🔗 Pull requests 0

📶 Pulse

📊 Graphs

Branch: master ▾

bootstrap-sass / assets / **stylesheets** /

glebm rake convert

..



bootstrap

rake convert

**_bootstrap-compass.scss**

Sass: explicitly set \$bootstrap-sass-asset-l

**_bootstrap-mincer.scss**

Update node.js devDeps, tests, new mince

**_bootstrap-sprockets.scss**

Sass: explicitly set \$bootstrap-sass-asset-l

**_bootstrap.scss**

rake convert



```
style.sass
```

```
#wrapper
```

```
  width: 1200px
```

```
  margin: 0 auto
```

```
  div#box
```

```
    width: 500px
```

```
    background: red
```



書籍転載：Ruby on Rails 4アプリケーションプログラミング

Sass (SCSS) 入門

Web Technology S

2014年6月10日

CSSのコードを生成するための言語「Sass (SCSS)」の基礎として、基本的なスタイル定義のネスト／変数／演算子／関数／ディレクティブ／について解説。書
の10本目（「Part 3 《応用編》 第9章 クライアントサイド開発」より）。

Sass = SCSS \neq Sass ?

Sassファイル = .scss ?

え、 どういうこと ? ?

2006 Sass誕生 SASS記法

2010 Sass 3.0～
Sassy CSS導入 SCSS記法

Sassの主流はSCSS記法へ

SASS記法

```
$font-stack: Helvetica, sans-serif  
$primary-color: #333
```

body

```
font: 100% $font-stack  
color: $primary-color
```

SCSS記法

```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;
```

body {

```
font: 100% $font-stack;  
color: $primary-color;  
}
```

どういうときに使えるの？

入れ子構造

```
1914     font-weight: bold;
1915     font-size: 13px;
1916     margin-bottom: 2px!important;
1917 }
1918 .current-goal-container .value,
1919 .past-data-container .value {
1920     font-weight: bold;
1921     font-size: 18px;
1922 }
1923 .current-goal-container .gauge-container,
1924 .past-data-container .gauge-container {
1925     margin-top: 55px;
1926 }
1927 .current-goal-container .gauge-container .status,
1928 .past-data-container .gauge-container .status {
1929     font-size: 14px;
1930     font-weight: bold;
1931     padding: 20px 32px 16px;
1932     margin-top: -60px;
1933 }
1934 .current-goal-container .gauge-container .status .value,
1935 .past-data-container .gauge-container .status .value {
1936     font-size: 20px;
1937     font-weight: bold;
1938 }
1939 .current-goal-container .gauge-container.well,
1940 .past-data-container .gauge-container.well {
1941     margin-top: 0!important;
1942 }
1943 .current-goal-container .gauge-container .header,
1944 .past-data-container .gauge-container .header {
1945     text-align: center;
1946 }
1947 .current-goal-container .gauge-container .body,
1948 .past-data-container .gauge-container .body {
1949     position: relative;
1950     top: -25px;
1951 }
1952 .current-goal-container .gauge-container .current,
1953 .past-data-container .gauge-container .current {
1954     position: absolute;
1955     font-size: 13px;
1956     margin-left: -8px;
1957     left: 50%;
1958     top: 20%;
```

この状態で
クラス名を書き換える
ことの恐怖…

怖くない。そう、Sassならね

```
nav ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
}  
nav li {  
    display: inline-block;  
}  
nav a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
}
```



```
nav {
```

```
ul {
```

```
margin: 0;
```

```
padding: 0;
```

```
list-style: none;
```

```
}
```

```
li { display: inline-block; }
```

```
a {
```

```
display: block;
```

```
padding: 6px 12px;
```

```
text-decoration: none;
```

```
}
```

```
}
```

変数

ある日の夜の会話

あ、サイトの雰囲気変わったから。少しね。

全体的に青系をオレンジっぽくかえてくれる？

明日まででいいよ。

怖くない。そう、Sassならね

`$primary-color: #C82606;`



```
body {  
  color: $primary-color;  
}
```



```
.button{  
  background-color: $primary-color;  
}
```

まずは体験してみよう

bit.ly/trysass

bit.ly/trysass

ネスト

HTMLのツリー構造に近い形でCSSを書くことができます

変数

CSSの一貫性をより保つことが出来ます

mixin

引数を使って定義すると効果的です

何事も使いすぎには
注意しましょう