


Coder's High 2017

2017.11.4

**テンプレートエンジンPugを使った
みんなで作るウェブ制作**



中村 勇希

 **nayucolony**

to-R

Frontend Engineer

注意

 **全スライド撮影・実況OK**

注意



#cssnite



#CSSNiteLP54

本日の流れ

デザインシステムとコンポーネント

テンプレートエンジン「Pug」

Pug × コンポーネント指向

導入にあたってのFAQ

まとめ





MATERIAL DESIGN

Material Design is a unified system that combines theory, resources, and tools for crafting digital experiences.

[WATCH THE VIDEO](#)



A new home for Material Design

Material is a metaphor, a system for uniting style, branding, interaction, and motion under a consistent set of principles. With Material we believe product teams

MATERIAL COMPONENTS [LEARN MORE](#)

Create beautiful apps with modular and customizable UI components.

GALLERY [LEARN MORE](#)

Design workflow, simplified.



[Install](#)

[Changelog](#)

 [Star](#) **6,264**

Primer

The design system that powers GitHub

[Install Primer 9.6.0](#)



LIGHTNING DESIGN SYSTEM

Create the world's best enterprise app experiences.

GET STARTED

Current release: [Winter '18 \(SLDS 2.4.4\)](#) | [Archives](#)

[Guidelines](#)

[Accessibility](#)

[Components](#)

[Utilities](#)

[Design Tokens](#)

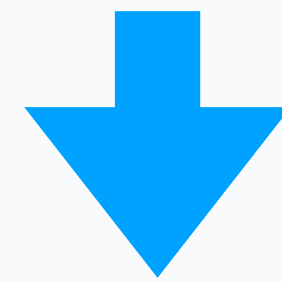
[Icons](#)

[Help & Support](#)

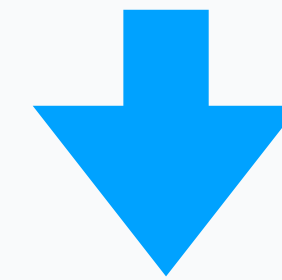


デザインシステムとは

デザインの考え方を構造化したもの



一貫性を保ちつつ、高速に開発するため



急速な社会の変化に
対応するため

デザインシステムがあると

デザイン・実装の指針となる



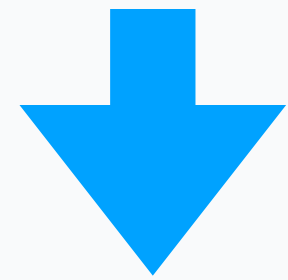
みんなが開発に参加できる



前後工程の人、これから参加する人、未来の自分

UI実装の指針

パターンライブラリ



UIコンポーネントの
デザインや実装を明文化したものの

Search

Getting Started

Platforms

Guidelines

Accessibility

Components

Accordion

Activity Timeline

Alert

App Launcher

Avatar

Badges

Breadcrumbs

Button Groups

Button Icons

Buttons

Buttons are used to invoke an event

BASE

DESIGN GUIDELINES

IMPLEMENTATION NOTES

DEVELOPER GUIDELINES

```
<button class="slds-button">Button</button>
```

About Base

The base `.slds-button` looks like a plain text link. It removes all the styling of the native button. It's typically used to trigger a modal or display a "like" link. All button variations are built by adding another class to `.slds-button`.

Button

Overview

Best Practices

Variants

Implementation Examples

Implementation

Standard

Button

Primary

Button

Compound Button

```
import * as React from 'react';
import {
  CompoundButton,
  IButtonProps
} from 'office-ui-fabric-react/lib/Button';
import { Label } from 'office-ui-fabric-react/lib/Label';

export class ButtonCompoundExample extends React.Component<IButtonProps, {}> {
  public constructor() {
```

Button

Overview

Best Practices

Variants

Implement

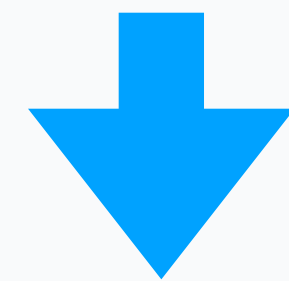
Standard

Butjon

Compound Button

デザインシステムに見る、統一指針

パーツを規格化
& 取り決め通りに実装

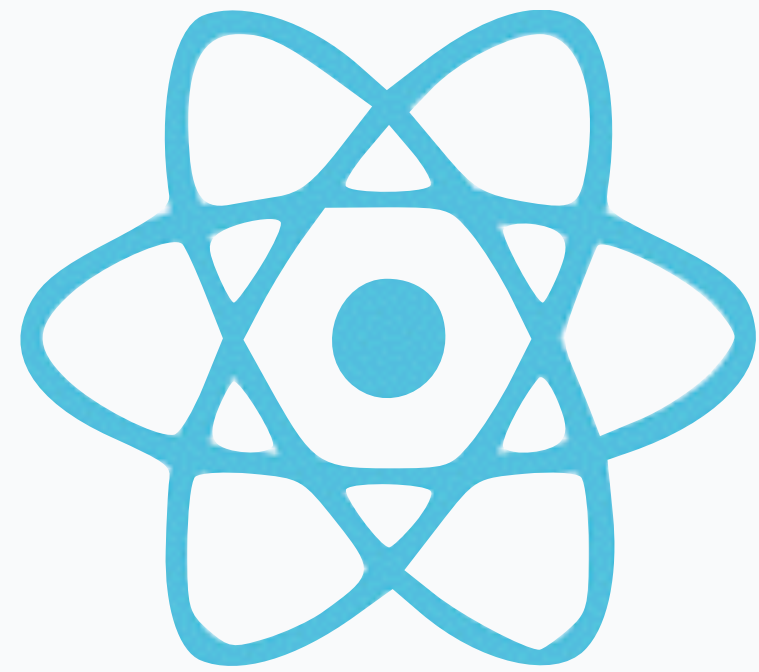


一貫性と効率化の両立をはかる

パーツ単位の設計思想

コンポーネント指向

いつもの人たち



BEM



コンポーネント単位で
ブロックを切って
クラスを命名

Sass



CSSを役割単位で分割
名前をつけて管理

Web Components



コンポーネント指向の
標準仕様が議論中

今日のお話

**「コンポーネント指向」な設計を
HTMLでやろう**

本日の流れ

✓ ~~デザインシステムとコンポーネント~~

テンプレートエンジン「Pug」

Pug × コンポーネント指向

導入にあたってのFAQ

まとめ



Pug



Pug

HTMLを分割できる

JavaScriptを
記述できる

インデント強制の
独自記法

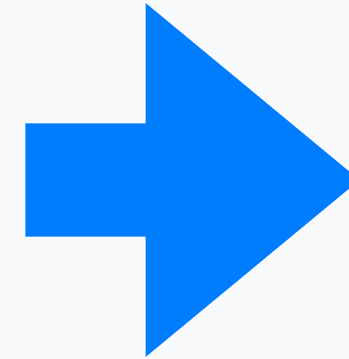


Pug

記法を紹介

閉じタグ不要、親子関係はインデントで

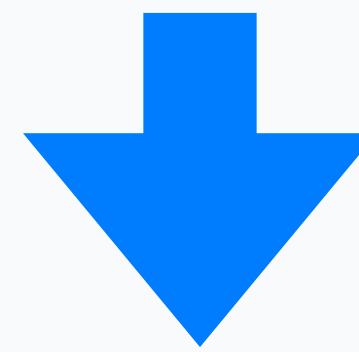
```
<ul>
  <li>リスト要素 1 </li>
  <li>リスト要素 2 </li>
</ul>
```



```
ul
  li リスト要素 1
  li リスト要素 2
```

属性はかっこで

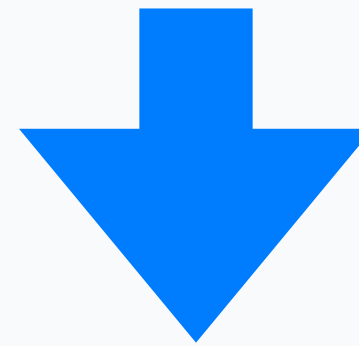
```
<a href="google.com">Google</a>
```



```
a(href="google.com") Google
```

改行処理はパイプで

```
<p>The pipe always goes at the beginning of its own line, not counting indentation.</p>
```



```
p  
| The pipe always goes at the beginning of its own line,  
| not counting indentation.
```

ここまでのまとめ

なんか気持ち悪い

書き方はメインじゃないので

メリットは後ほど

本日の流れ

✓ ~~デザインシステムとコンポーネント~~

✓ ~~テンプレートエンジン「Pug」~~

Pug × コンポーネント指向

導入にあたってのFAQ

まとめ



Pug

コンポーネント指向な
HTMLを書いてみよう

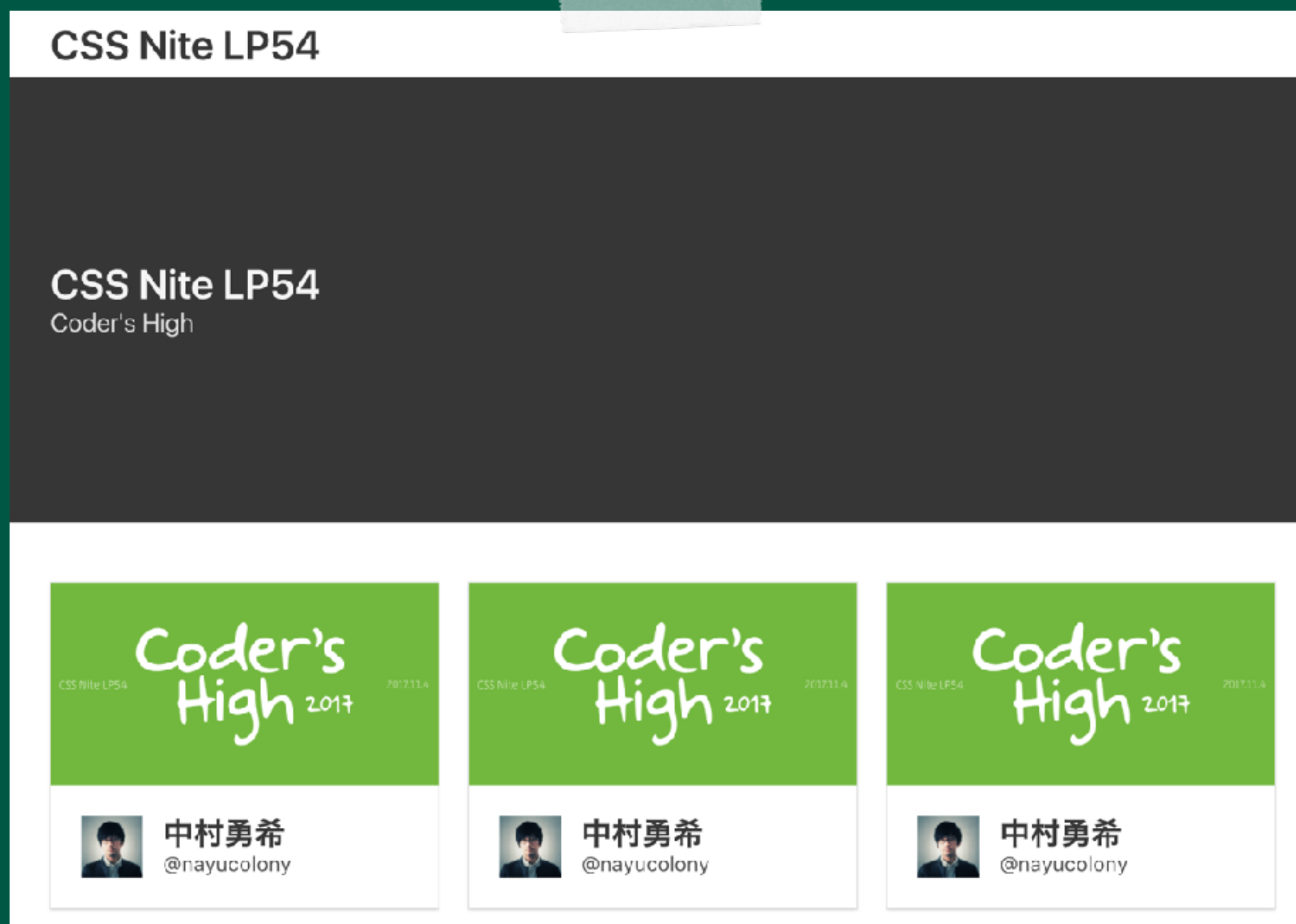


Pug

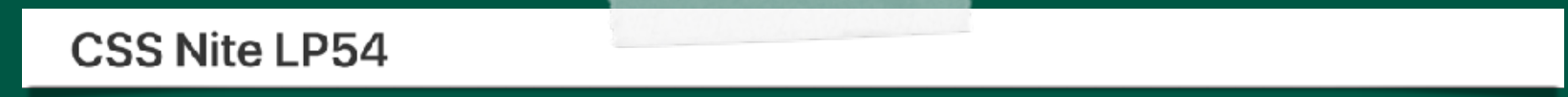
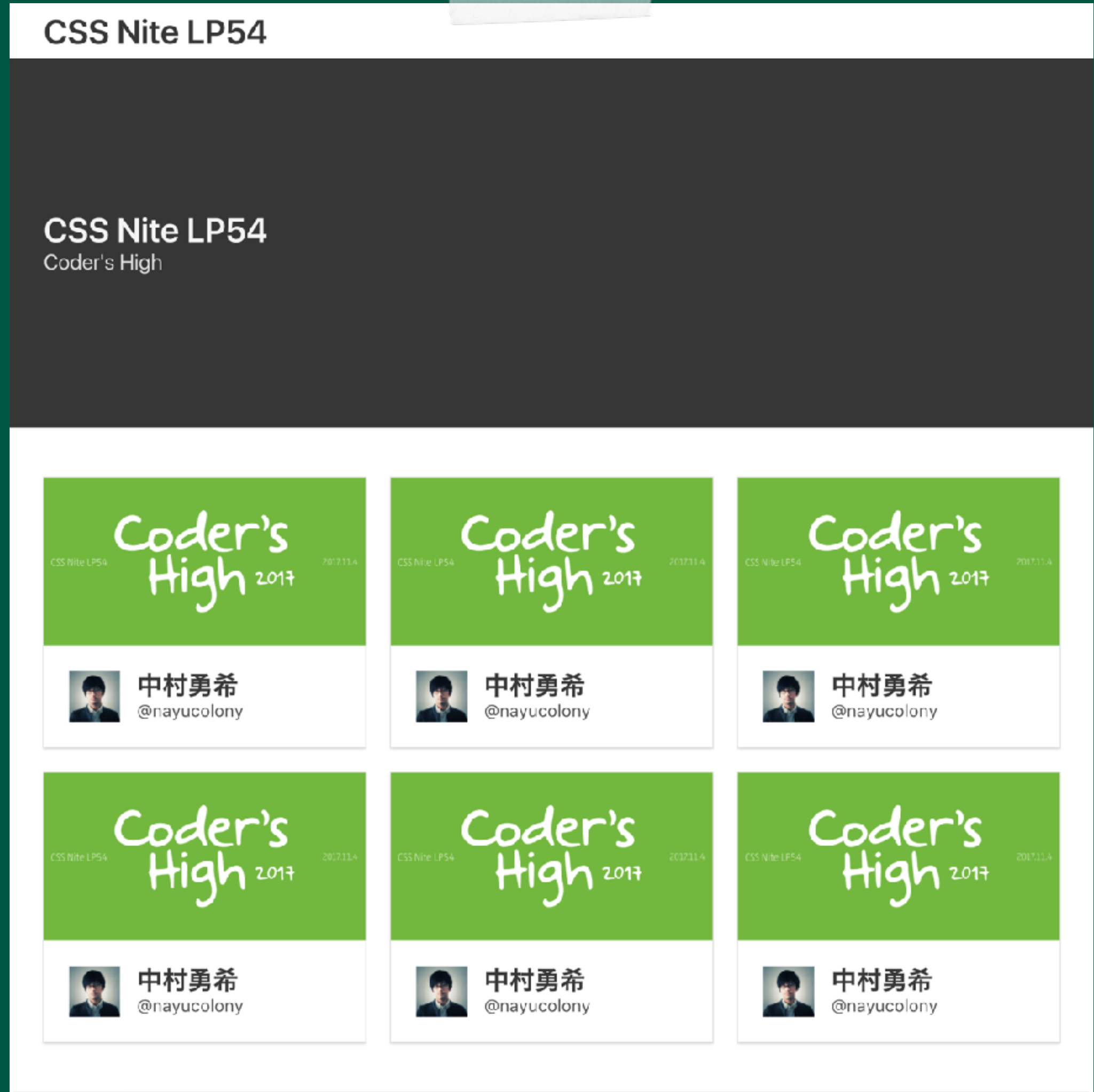
#1

**ファイル分割
& 読み込み**

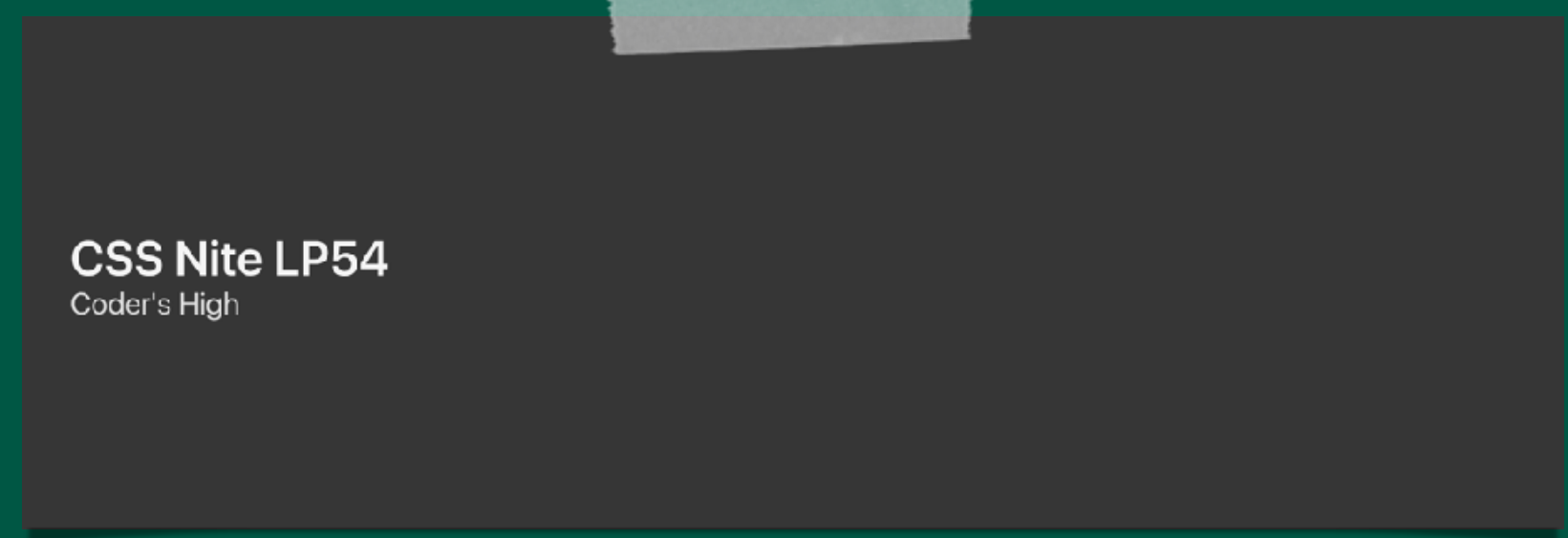
今からページを実装します



役割ごとに分割



ヘッダー

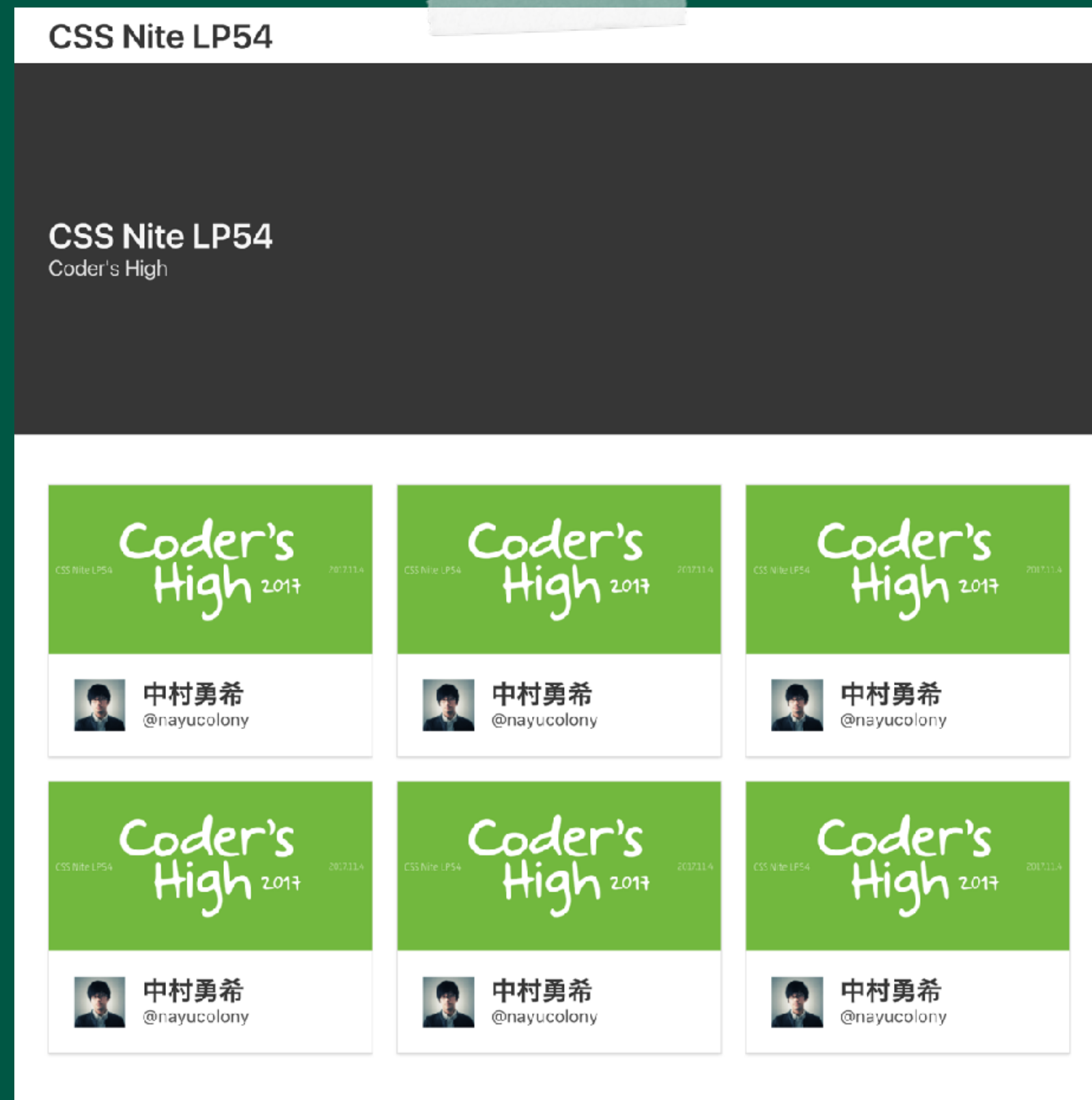


ヒーロー



カード

それぞれを実装



```
.container  
.navbar  
.navbar-brand  
.navbar-item  
.title CSS Nite LP54
```

header.pug

```
.hero.is-dark.is-medium  
.hero-body  
.container  
h1.title= title  
h2.subtitle= subtitle
```

hero.pug

```
.card  
.card-image  
figure.image.is-2by  
img(src="/img/CSSNiteLP54.png", alt="")  
.card-content  
.media  
.media-left  
figure.image.is-48x48  
img(src="/img/profile.jpg", alt="")  
.media-content  
.title.is-4 中村勇希  
.subtitle.is-6 @nayucolony
```

card.pug



それぞれを挿入

```
.container
  .navbar
    .navbar-brand
    .navbar-item
    .title CSS Nite LP54
```

header.pug

```
.hero.is-dark.is-medium
  .hero-body
    .container
      h1.title= title
      h2.subtitle= subtitle
```

hero.pug

```
html(lang="ja")
  head
    略
  body
    include _partial/_header
    include _partial/_hero
    include _partial/_card-section
```

index.pug

```
.card
  .card-image
    figure.image.is-2by
      img(src="/img/CSSNiteLP54.png", alt="")
  .card-content
    .media
      .media-left
        figure.image.is-48x48
          img(src="/img/profile.jpg", alt="")
      .media-content
        .title.is-4 中村勇希
        .subtitle.is-6 @nayucolony
```

card.pug



共通パーツを使いまわし



セットでCSSを書けば完璧

CSS Nite LP54

header.pug



header.scss

CSS Nite LP54
Coder's High

hero.pug



hero.scss



card.pug



card.scss



#1

まとめ

- **1ファイルの記述量が減る**
- **ページの構造がわかりやすい**
- **使いまわしもできる**



Pug

#2

変数の利用

同じパーツなのに、文言が違う

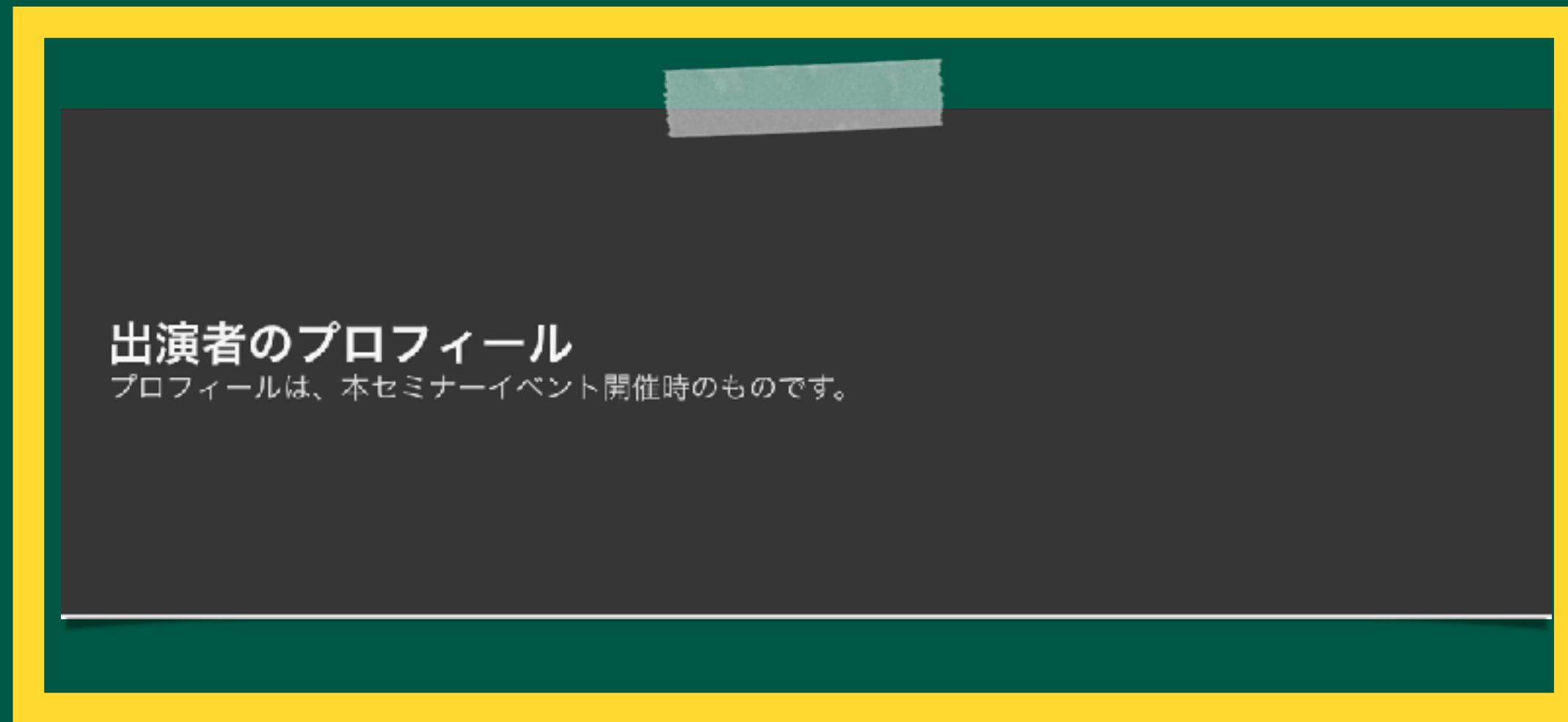
出演者のプロフィール

プロフィールは、本セミナーイベント開催時のものです。

CSS Nite LP54

Coder's High

別々の役割？



profile-hero.pug



home-hero.pug

同じ役割？

—

出演者のプロフィール

プロフィールは、本セミナーイベント開催時のものです。

CSS Nite LP54

Coder's High

hero.pug

役割を考えると

挿入先に依存してテキストデータが変わる



ページ側でデータを持つのが妥当

コンポーネントに実データを持たせない

`h1= title`

挿入先のもつデータを代入すれば、
1コンポーネントで管理できる

ページファイルごとに文言の定義

```
var title = "出演者のプロフィール"
```

```
h1= title
```

profile.pug



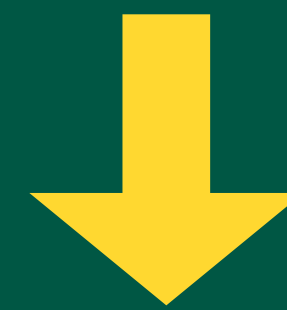
出演者のプロフィール

プロフィールは、本セミナーイベント開催時のものです。

```
var title = "CSS Nite LP54"
```

```
h1= title
```

index.pug



CSS Nite LP54

Coder's High



#2

まとめ

役割も構造も同じパーツは
いくつも作らなくてOK



Pug

#3

繰り返し

カードが並ぶUI



中村勇希
@nayucolony



中村勇希
@nayucolony



中村勇希
@nayucolony

パーツで見ると

```
<div class="card">
```

```
</div>
```




```
<div class="card">
  _____
  _____
  _____
  _____
  _____
  _____
</div>
<div class="card">
  _____
  _____
  _____
  _____
  _____
  _____
</div>
<div class="card">
  _____
  _____
  _____
  _____
  _____
  _____
</div>
<div class="card">
  _____
  _____
  _____
  _____
  _____
  _____
</div>
<div class="card">
  _____
  _____
  _____
  _____
  _____
  _____
</div>
<div class="card">
  _____
  _____
  _____
  _____
  _____
  _____
</div>
<div class="card">
  _____
  _____
  _____
  _____
  _____
  _____
</div>
```

扱いにくそう



念のため注意しよう

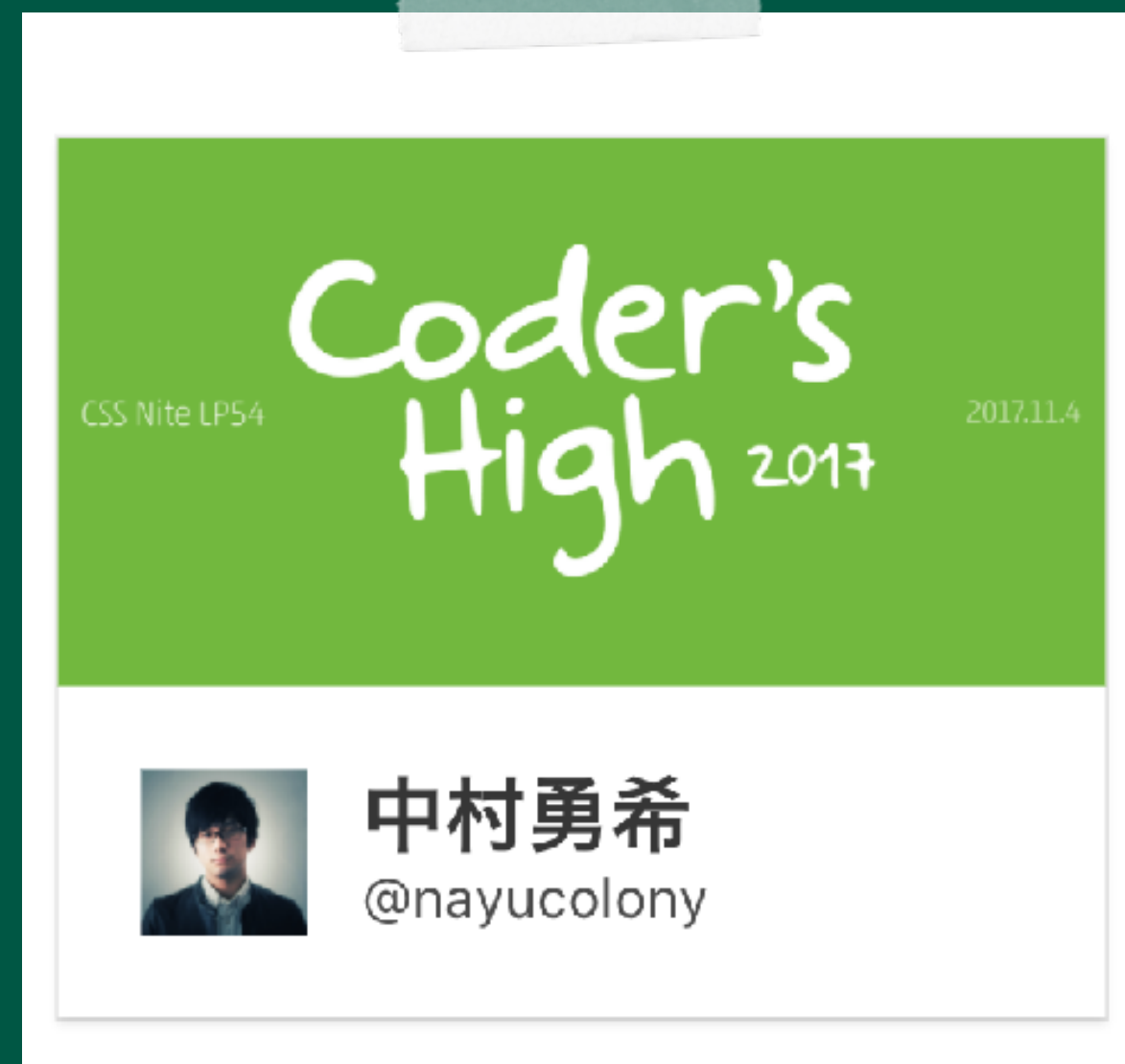


無駄なコスト発生



ホントは難しくないので

同じコンポーネントということを示す



× 3 =



分割済みのコンポーネントを挿入する

```
include card.pug
```

card-list.pug

× 3 =



for文で繰り返し



```
- for (x = 0; x < 3; x++)  
  include card.pug
```

card-list.pug



繰り返し回数を変えるだけでOK

```
- for (x = 0; x < 6; x++)  
  include card.pug
```

card-list.pug





#3

まとめ

- コピペ不要
- 数値変えるだけでOK
- 修正も簡単

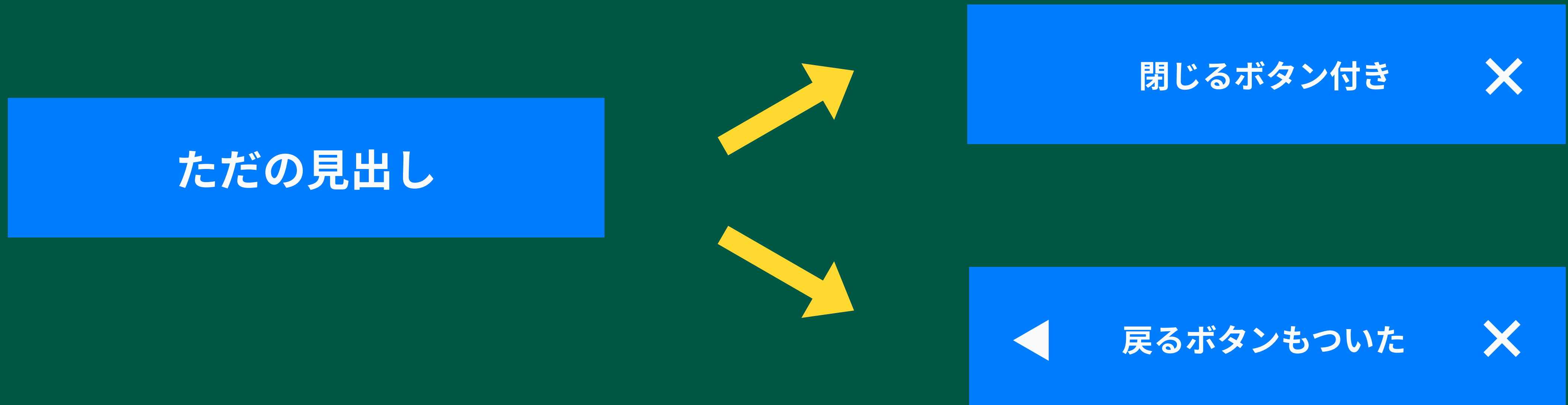


Pug

#4

条件分岐

コントロール系のUI



パターンの明文化はコンポーネント側で



どのパターンを出すかは挿入先で制御

条件と挙動を明記

ただの見出し

閉じるボタン付き



```
div nav
```

```
h1= title
```

```
if close  
  button.close
```

```
if prev  
  button.prev
```



戻るボタンもついた



control.pug

条件と挙動を明記

ただの見出し

```
div.nav  
  h1= title
```

```
  if close  
    button.close
```

```
  if prev  
    button.prev
```

control.pug

```
var close = true  
var prev = false  
var title = 閉じるのみ
```

閉じるのみ



```
var close = true  
var prev = true  
var title = 閉じる、戻る
```



閉じる、戻る





#4

まとめ

**次工程に
「エンジニアによる組み込み」
がある場合も
コード見せるだけで伝わる**

全体のまとめ

HTMLが分割できる

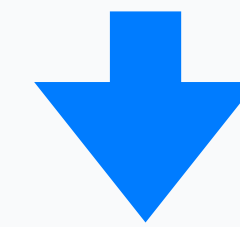


それぞれが使い回せる
パーツファイルになる



コピーや検索置換など
「ただの作業」の削減

JavaScriptが使える



プログラムによる
デザインの言語化

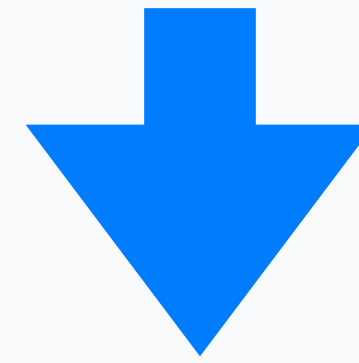


状況把握などに
かかるコストの削減

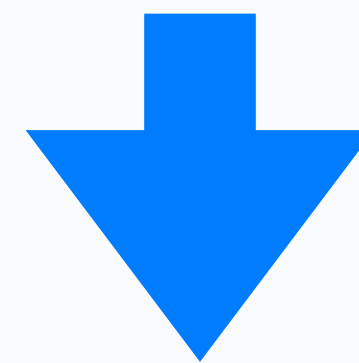
全体のまとめ

コピーや検索置換など
「ただの作業」の削減

状況把握などにか
かかるコストの削減



高速な開発が可能になり、時間が生まれる



プロとしての付加価値を生み出す

本日の流れ

✓ ~~デザインシステムとコンポーネント~~

✓ ~~テンプレートエンジン「Pug」~~

✓ ~~Pug × コンポーネント指向~~

導入にあたってのFAQ

まとめ



JavaScriptとかちよっと…

JavaScriptを記述するのに抵抗がある

まずは繰り返しだけ



プロジェクト内で「繰り返し警察」になる

繰り返し警察

単純に繰り返しでは都合の悪い場所



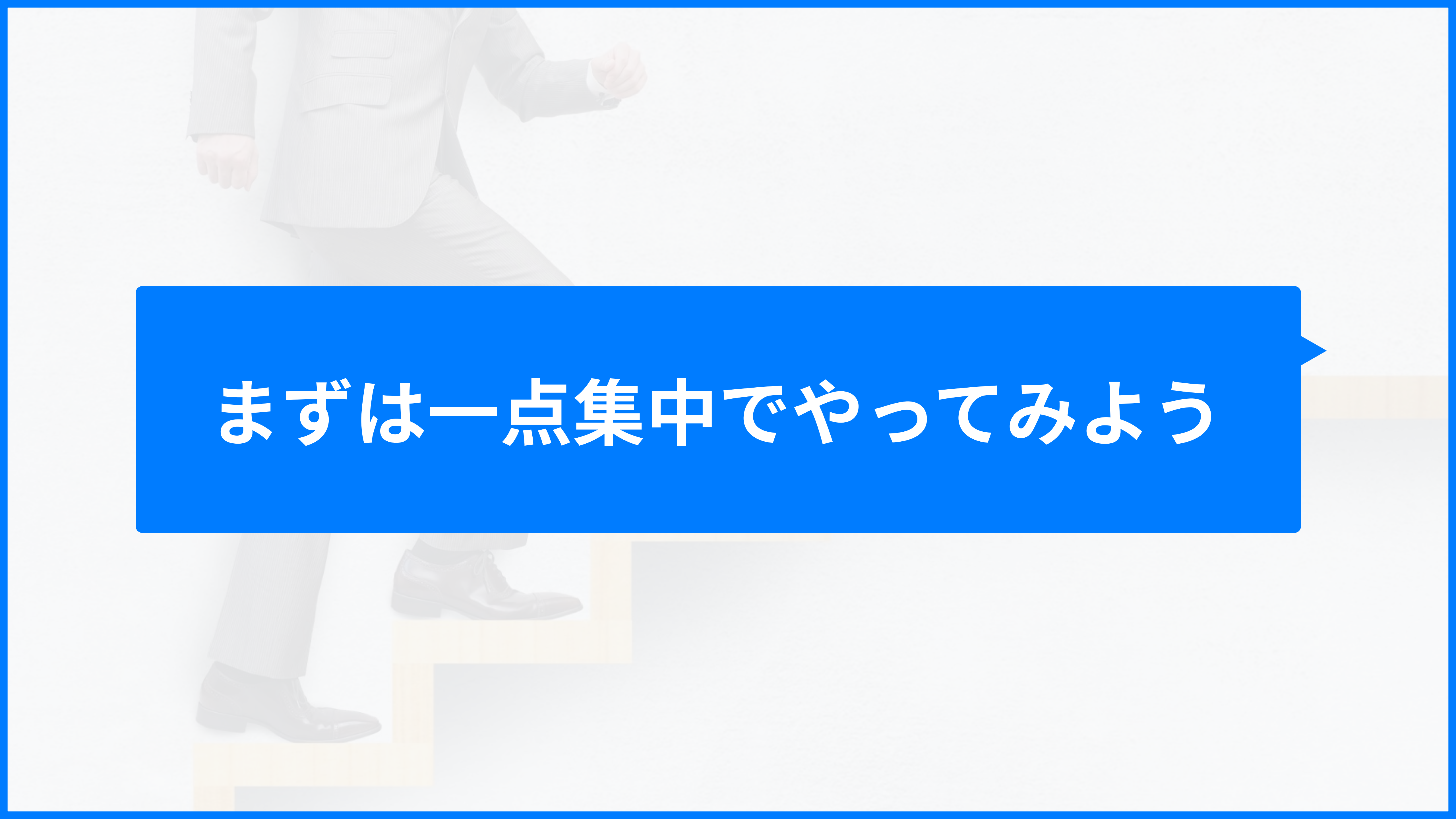
リファクタリングしてみる

ステップアップ

繰り返しに慣れてきたら



条件分岐、変数などにチャレンジ！

A person in a light-colored suit is walking up a set of stairs. The image is semi-transparent. A blue speech bubble with a white border is positioned in the center, containing the Japanese text 'まずは一点集中でやってみよう'.

まずは一点集中でやってみよう



独自記法を覚えるのが大変

独自記法

html2pug

html2pug help you convert a HTML snippet to a Jade snippet.

Grouped

Bodyless No commas in attributes

Indents

Spaces

Tabs

Spaces

4

HTML

```
1 <!DOCTYPE html>
2- <html lang="en">
3- <head>
4-   <title>Jade</title>
5-   <script type="text/javascript">
6-     foo = true;
7-     bar = function () {};
8-     if (foo) {
9-       bar(1 + 5)
10-    }
11-   </script>
12- </head>
13- <body>
14-   <h1>Jade - node template engine</h1>
15-   <div id="container" class="col">
16-     <p>You are amazing</p>
17-     <p>Jade is a terse and simple
18-       templating language with a
19-       strong focus on performance
20-       and powerful features.</p>
21-   </div>
22- </body>
23- </html>
```

JADE

```
1
```

Convert



まずは自動変換ツールを使おう



記法がキモい……

記法がキモい

メリットは記述量が減ること



人はミスをする生き物

読み書きは少ない方がいい

そもそも

面白がってあんな記法にしたわけじゃない



すぐ慣れるので大丈夫！

A person in a grey suit is walking on a staircase. The image is faded and serves as a background. A blue speech bubble is positioned in the center, containing the Japanese text '慣れるって' (Nareru tte), which translates to 'Getting used to it' or 'It's becoming a habit'.

慣れるって



デメリットはないの？

大きなデメリット

簡単にマークアップを放棄できる

要素は省略できる

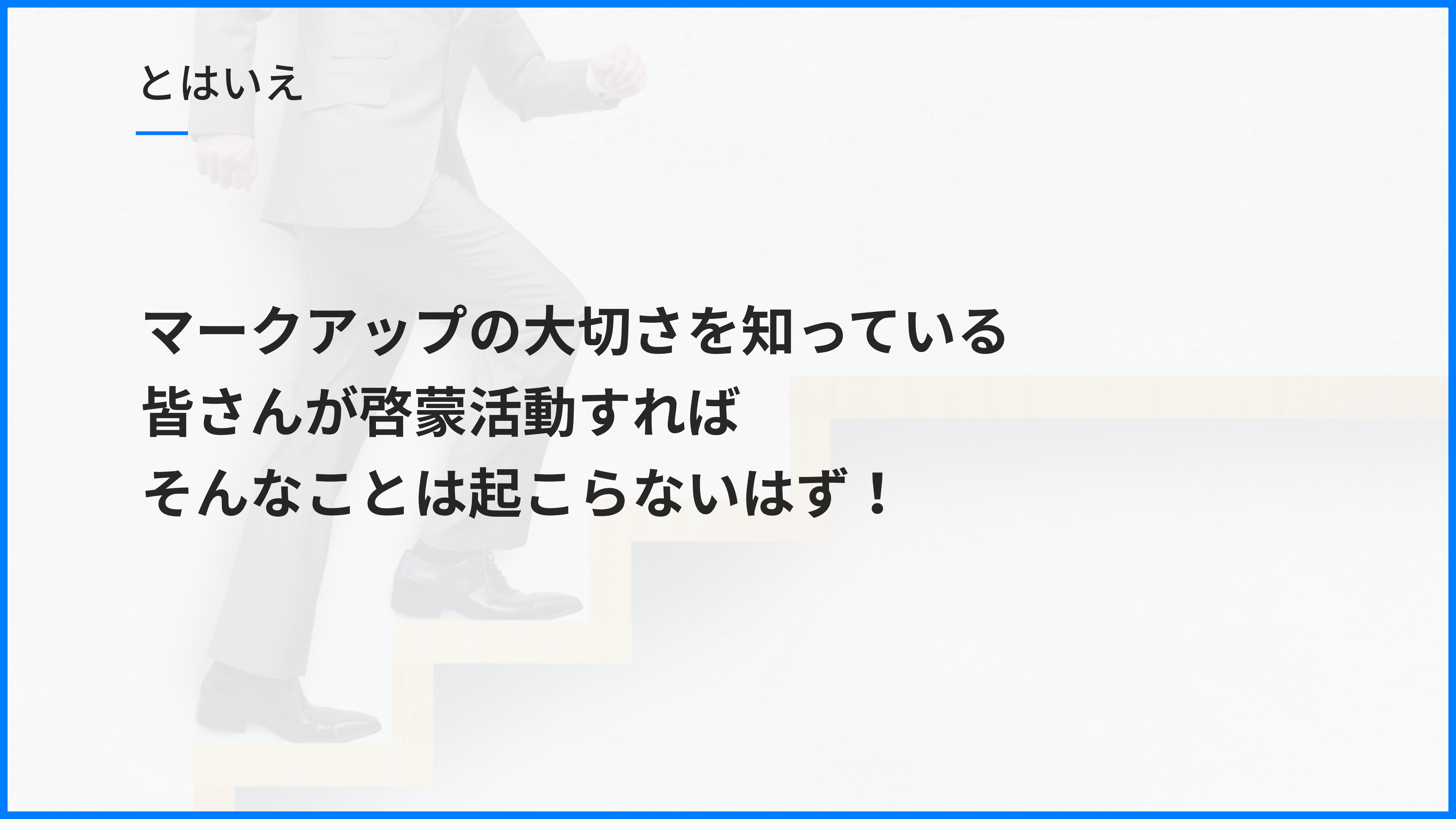
```
div.card  
  div.card-header  
  div.card-content
```



```
.card  
  .card-header  
  .card-content
```

自動的に<div>で書き出される

とはいえ



マークアップの大切さを知っている
皆さんが啓蒙活動すれば
そんなことは起こらないはず！

A person wearing a grey suit is walking on a staircase. The image is faded and serves as a background for the text. The person is captured in mid-stride, with one foot on a higher step than the other. The background is a light, neutral color.

普通に使ってれば大丈夫♪

本日の流れ

✓ ~~ワークフローの今と未来~~

✓ ~~要求を現実にするために~~

✓ ~~テンプレートエンジン「Pug」~~

✓ ~~導入にあたってのFAQ~~

まとめ

まとめ

分割・使い回しで

情報の一元化

&

JSでロジカルな設計をして

デザインの言語化

これらを活用して

次工程に関わる人

新しく参加した人

半年後の自分

みんなが触れるコードに！

Pugが死んだらどうするの？

喜びましょう

もっといいものに
いずれ淘汰される

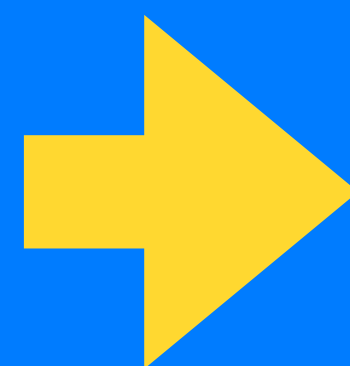


素直にそれに
乗り換えればよい

フロントエンドは流れがはやすぎる

少しずつアップデートしましょう

運用で得た知見は
次に活かせる



小さな学びを
積み重ねよう

ありがとうございました！

@nayucolony

続きは、Twitterや懇親会でお話ししましょう！