CSS Nite LP54

Coder's High 2017

2017.11.4

CSS設計方法論とその先

高津戸壮 @Takazudo 2017/11/4

このセッションで話すこと

- HTMLの文法としてどう正しいかみたいな話はしない
- どういうプロパティを使うのが最適かみたいな話もしない
- 設計的な悩みについて扱う
- どういう設計指針を持ってCSSを書いていけば良いか
- そのヒントとなることを伝えられたら

自己紹介

- 高津戸壮 @Takazudo
- 株式会社ピクセルグリッド
- フロントエンドエンジニア
- Webデザイナーのための jQuery入門

- CSS設計方法論やツールが解決する問題・しない問題
- ひとつの理想的な形 Atomic Design
- Atomic Designから得られるCSS設計のヒント
- モジュールの汎用性について
- 名前空間によるモジュールのグルーピング
- HTML上で組み立てるAtomic CSS
- どう設計するかはどう決めればいいか
- 設計例
- 私的結論

CSS設計方法論やツールが解決する問題・しない問題

CSS設計方法論や
各種ツールとどう向き合うか

- より良いやり方を模索するという意味では良いが
- 特に困っていなければ取り立てて
  採用する必要はないかと思う
- 何か問題を解決するために必要なことを自身の設計
  へ取り込むというスタンスで良いかと考える

ほとんど考える必要がない例

- 自己紹介のページを一つ作ろう
- 10Pぐらいの小さいお店の紹介サイトを作ろう
- あなたのブログ

どうCSS設計すべきか

- 好きに書けばいい
- 誰も困らない

考えないとまずい例

- コーポレートサイト 50Pとか1000Pとか
- 長期に渡り保守されるWebサイト／アプリ
- 複数の開発者が関わるWebサイト／アプリ

などなど

どういうことで困る？

- ページ量産に時間がかかりすぎる
- 知らないスタイルが勝手に当たる
- CSSの容量増えすぎてしまう
- 書いてあるコードが読解困難
- とにかくもうどうにもならなくなる
- 同じコードが何度も登場
  無駄な気がしてくるが整理できない

などなど

助けてくれるものたち

- CSS設計方法論
- CSSプリプロセッサ、PostCSS等のツール
- スタイルガイド
- コーディングガイドライン

などなど

CSS設計方法論って何

- そのような多種多様な問題を回避すべく、
  どのように考えながらHTML+CSSを書けばよいか
  という設計が「CSS設計」と呼ばれるもの

- 具体的な考え方やコードの記述ルール等をまとめ
  たものが「CSS設計方法論」と呼ばれるもの

知らないスタイルがあたる問題

書籍タイトル

彼は背後にひそかな足音を聞いた。それはあまり良い意味を示すものではない。誰がこんな夜更けに、しかもこんな街灯のお粗末な港街の狭い小道で彼をつけて来るというのだ。

詳細へ

書籍タイトル

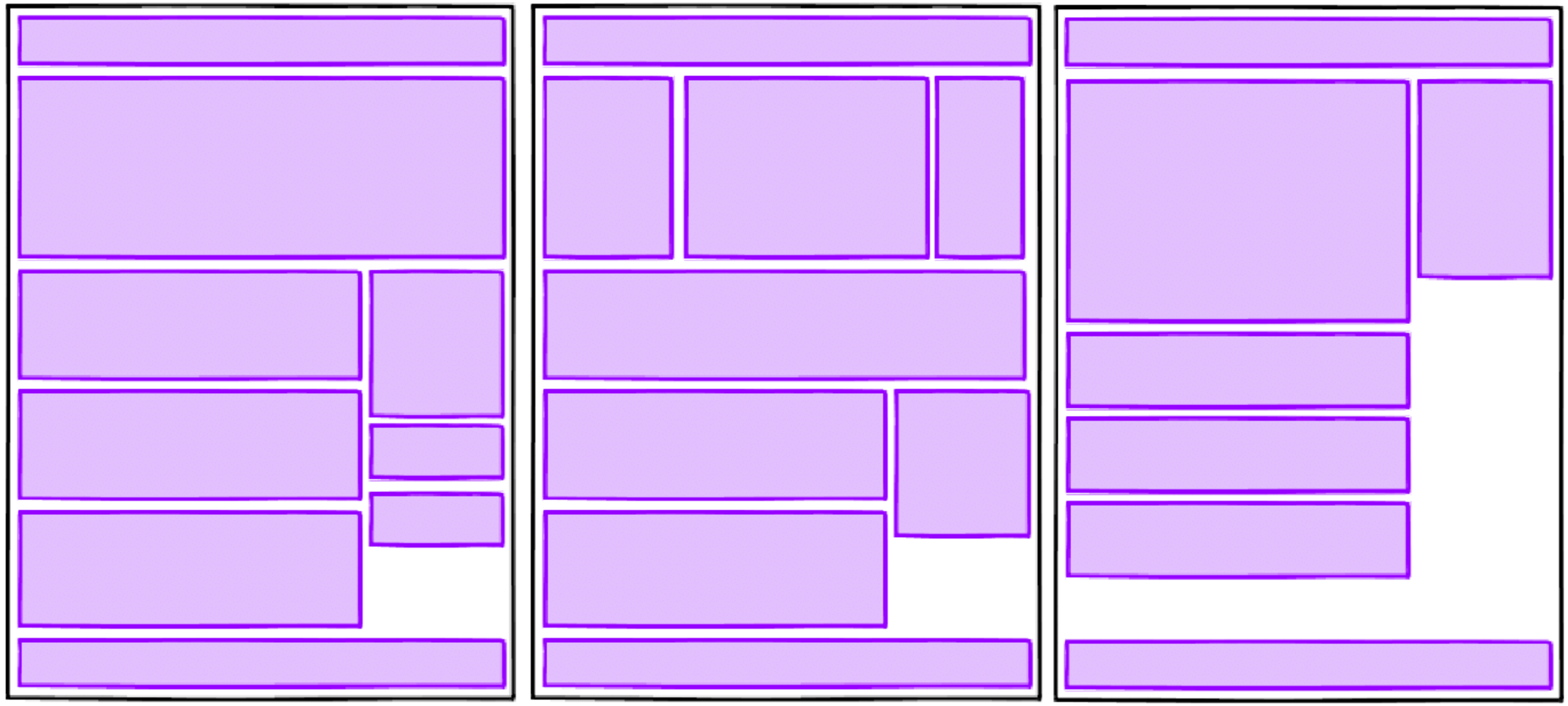
彼は背後にひそかな足音を聞いた。それはあまり良い意味を示すものではない。誰がこんな夜更けに、しかもこんな街灯のお粗末な港街の狭い小道で彼をつけて来るというのだ。

詳細へ

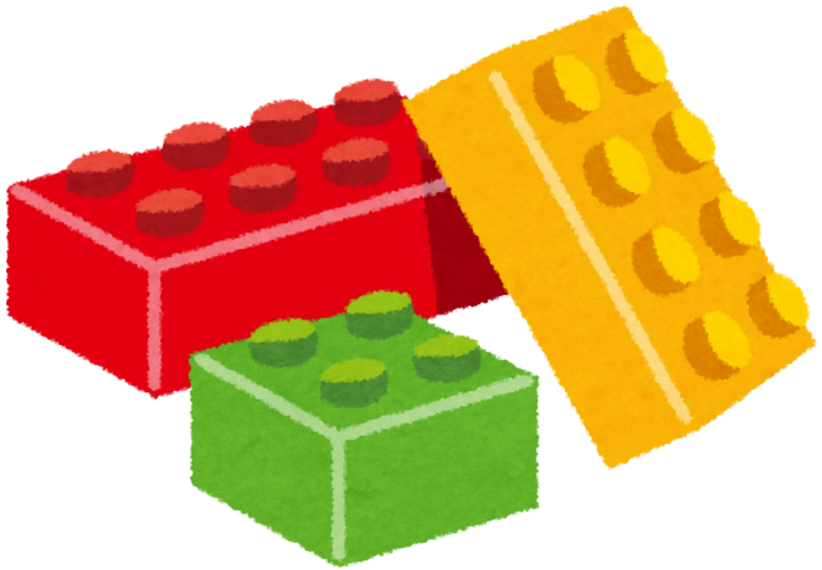
- この赤枠はどこでついているのか分からない…
- なんでここ太文字になるの？
- こっちを直したらあっちがおかしくなる
- CSSは全てがグローバル
- 何も考えず書いていると回避不可能

モジュール化しよう by OOCSS

クラス命名規則を設け、シンプルなセレクタで書くルールに by BEM



CSSの容量増えすぎてしまう問題



- モジュール化
- ベース＋スキン by OOCSS
- ブロック＋モディファイア by BEM

ベースのアラート

This is a primary alert—check it out!

This is a secondary alert—check it out!

This is a success alert—check it out!

This is a danger alert—check it out!

This is a warning alert—check it out!

Alerts · Bootstrap
http://getbootstrap.com/docs/4.0/components/alerts/

```
<div class="alert">...</div>

<div class="alert alert-primary">...</div>
<div class="alert alert-secondary">...</div>
<div class="alert alert-success">...</div>
<div class="alert alert-danger">...</div>
<div class="alert alert-warning">...</div>
```

書き方がみんなバラバラ

- コーディングガイドライン
- stylelint

```
visual.css
 2:12  ✖  Unexpected invalid hex color "#4f"                          color-no-invalid-hex
 4:1   ⚠  Expected ".foo.bar" to have a specificity no more than "0,1,0"   selector-max-specificity
 6:13  ✖  Unexpected unit "px" for property "margin"                  declaration-property-unit-blacklist
 7:17  ✖  Expected single space after "," in a single-line function   function-comma-space-after
```

- カラー値がinvalid
- 詳細度が高すぎる
- ブラックリスト化された単位を使っている
- カンマの後ろにスペース入ってない

stylelint
https://stylelint.io

CSS

フォーマット

- インデントは２スペース
- キャメルケースではなくダッシュを使うことを推奨
  - もしBEMを使っているのであればアンダースコアとパスカルケースは使用可能(下記のOOCSS and BEMをみてください)
- IDセレクタは使わない
- 複数セレクタのスタイルルールを宣言する場合は１行に１つのセレクタを記述する
- 中括弧の開始、つまり '{'の前にスペースを1つ入れる
- プロパティにおいて、`:`の前でなく後ろにスペースを１つ入れる
- 中括弧の終了、つまり `}`の前に改行を入れる
- スタイルの宣言の間には改行を入れること

airbnbスタイルガイド翻訳
https://github.com/nao215/css-style-guide

悪い例

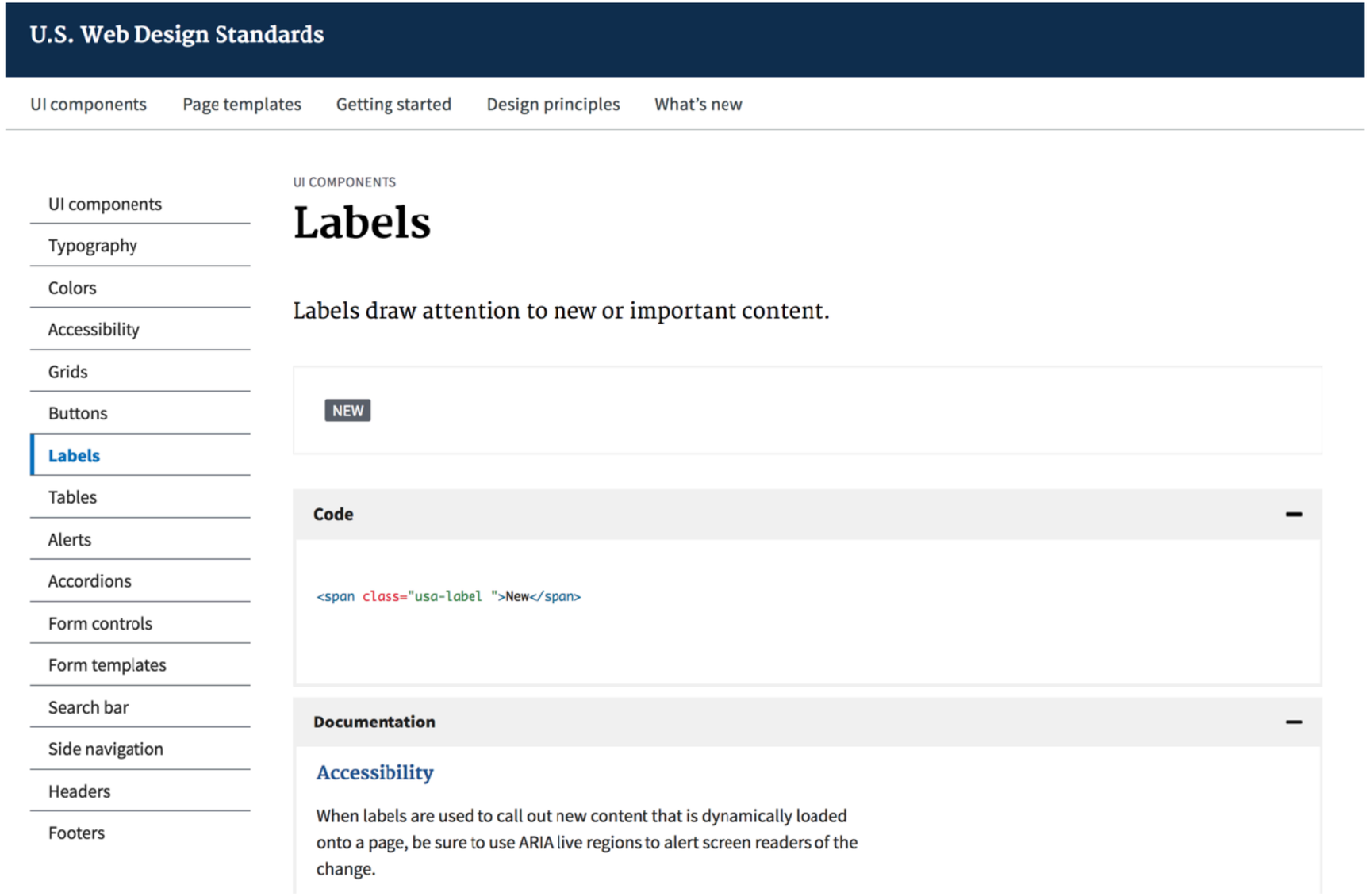
```
.avatar{
    border-radius:50%;
    border:2px solid white; }
.no, .nope, .not_good {
    // ...
}
#lol-no {
  // ...
}
```

良い例

```
.avatar {
  border-radius: 50%;
  border: 2px solid white;
}

.one,
.selector,
.per-line {
  // ...
}
```

同じコードが何度も登場

- mixin, extend でコードをDRYに
- Autoprefixer でvendor prefixはコード管理外へ
- スタイルガイドで重複したモジュールの作成を回避



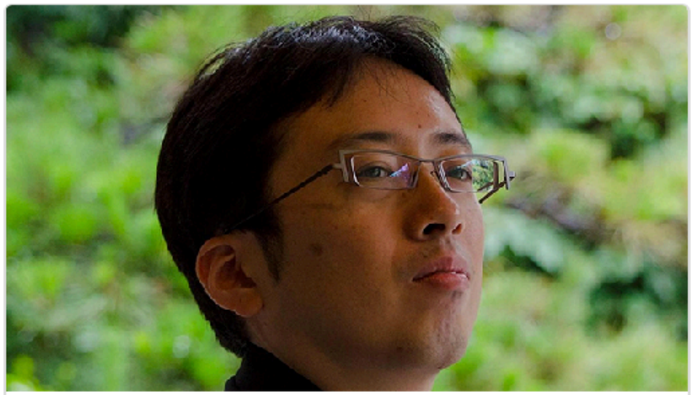
U.S. Web Design Standards
https://standards.usa.gov/

それで解決しない問題

- CSS設計方法論は単にやりかたを述べたものに過ぎない
  あなたのプロジェクトでどうすればよいかを
  具体的に示してくれるものではない

- 各種ツールはあくまで道具に過ぎない
  どう使えば何を得られるのかは設計者が考える

モジュールを切る難しさ



高津戸 壮

Web制作会社、フリーランスを経て、株式会社ピクセルグリッドに入社。数多くのWebサイト、WebアプリケーションのHTML、CSS、JavaScript実装に携わってきた。

jQuery入門

フロントエンドの開発手法は日々進化を続け、新しい情報や技術はあっという間に過去のものになっていきます。そんな中、jQueryは今やウェブサイト開発の土台といえるほどに普及しました。

詳細へ

同じモジュール?

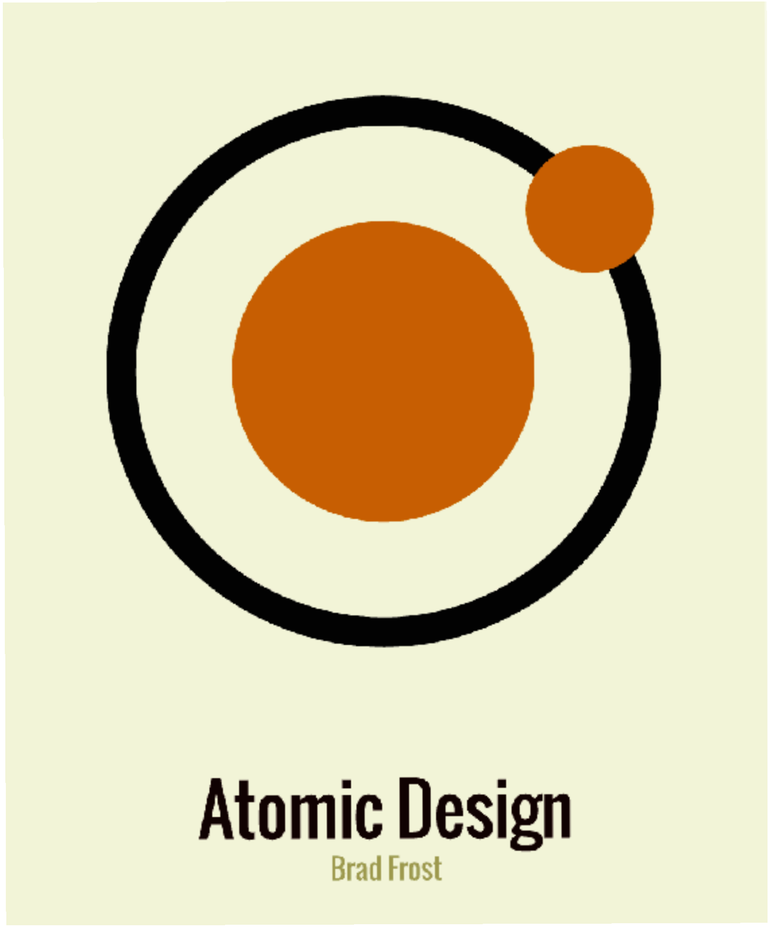
モディファイアで?

モジュールの名前は?

ボタンは別モジュール?

ひとつの理想的な形
Atomic Design

Atomic Designって何？



- Brad Frost 著
- そもそもまずページを作っているという認識を
  改める必要がある
- まず作るべきはデザインシステム
  そこからページを作ると考える
- デザインシステムの確立と運用が
  サイト全体のUIを統合されたものとする

というような設計論

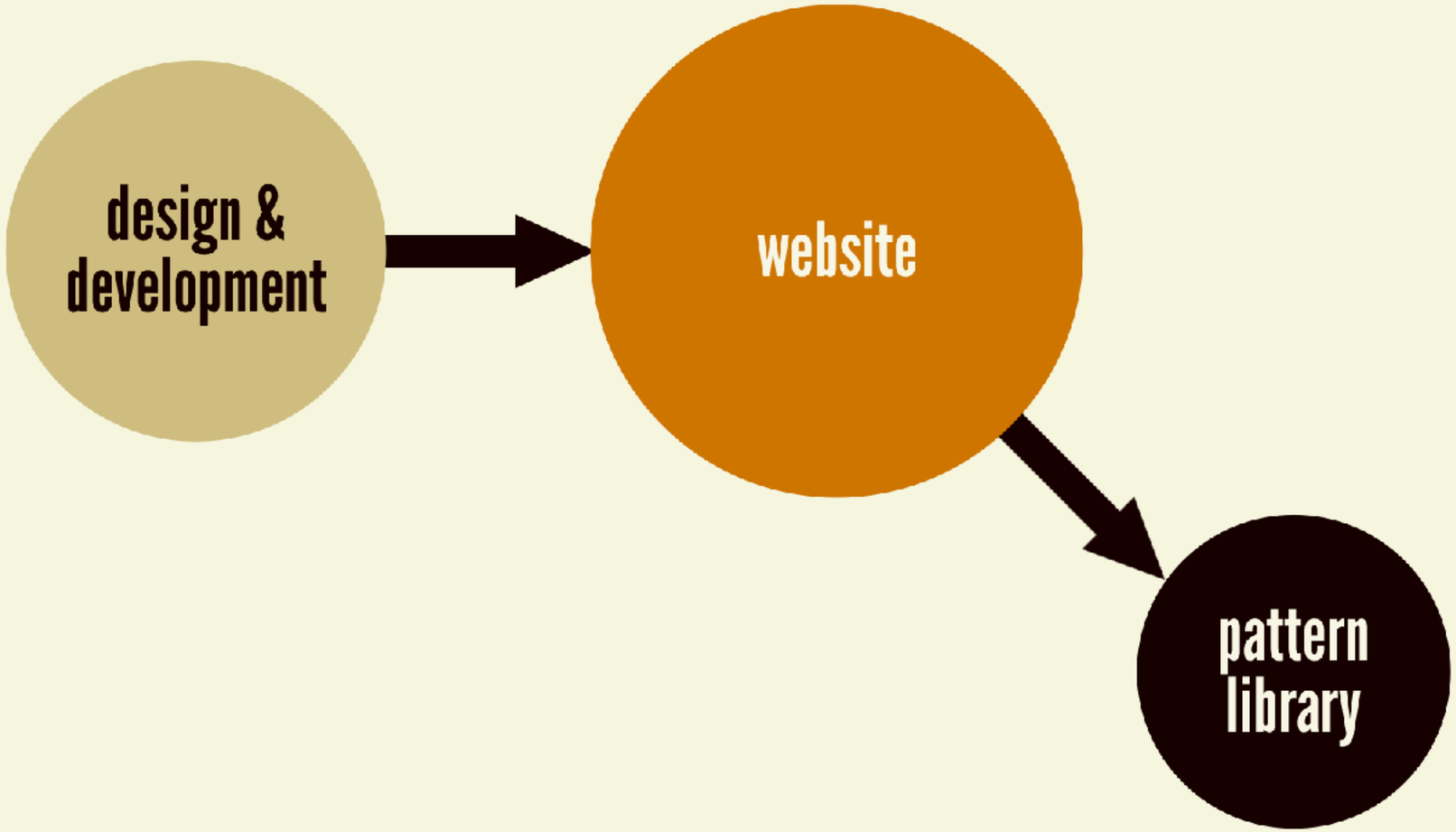
Atomic Design
http://atomicdesign.bradfrost.com/

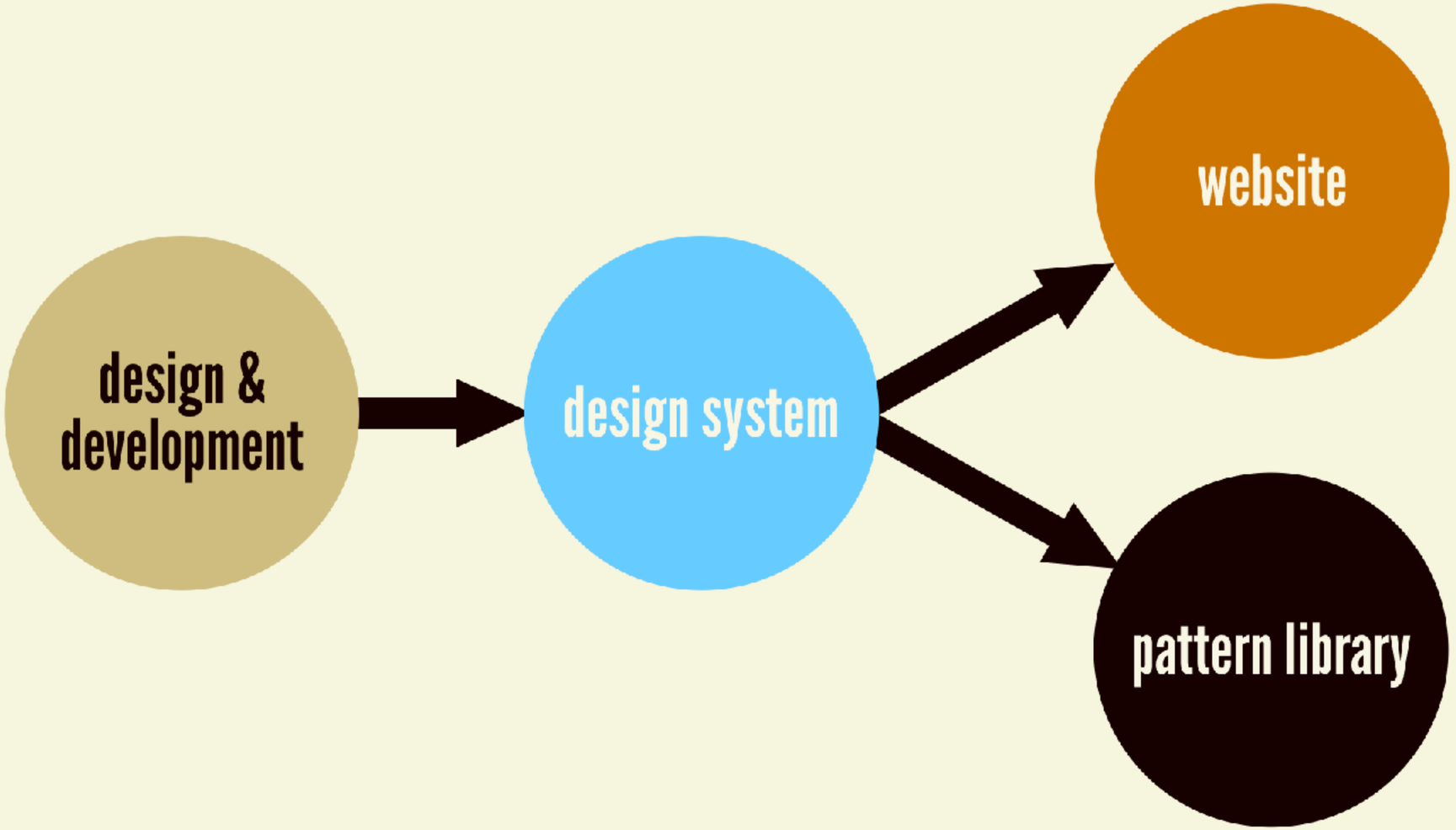
デザインシステムって何？

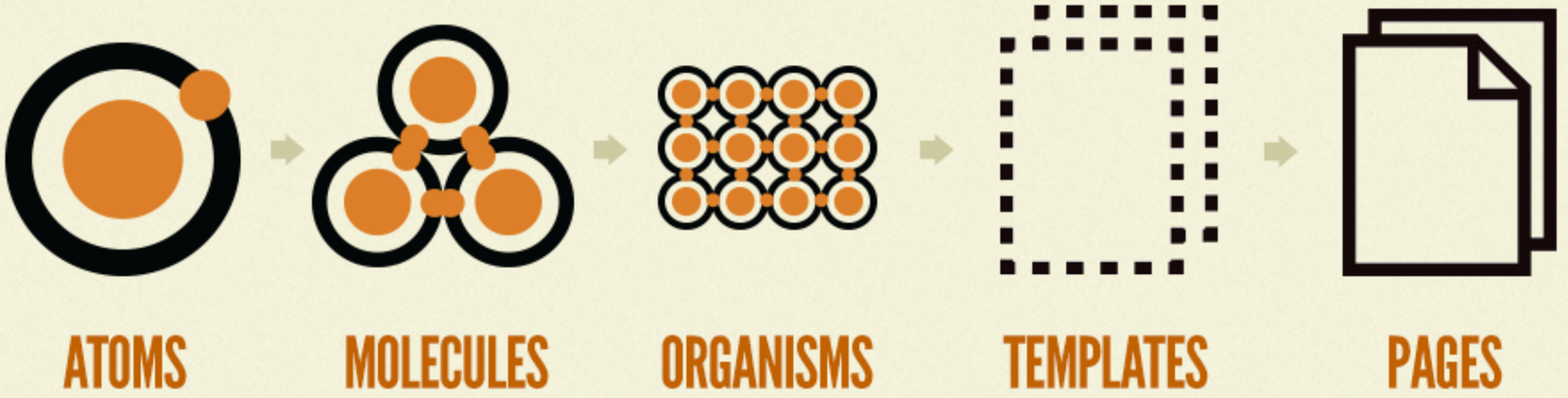
たいていの場合、デザインシステムは視覚的スタイルとコンポーネントのライブラリを提供する。それはドキュメント化され、開発者には再利用可能なコード集、デザイナーにとっては有用なツールとなるものである。

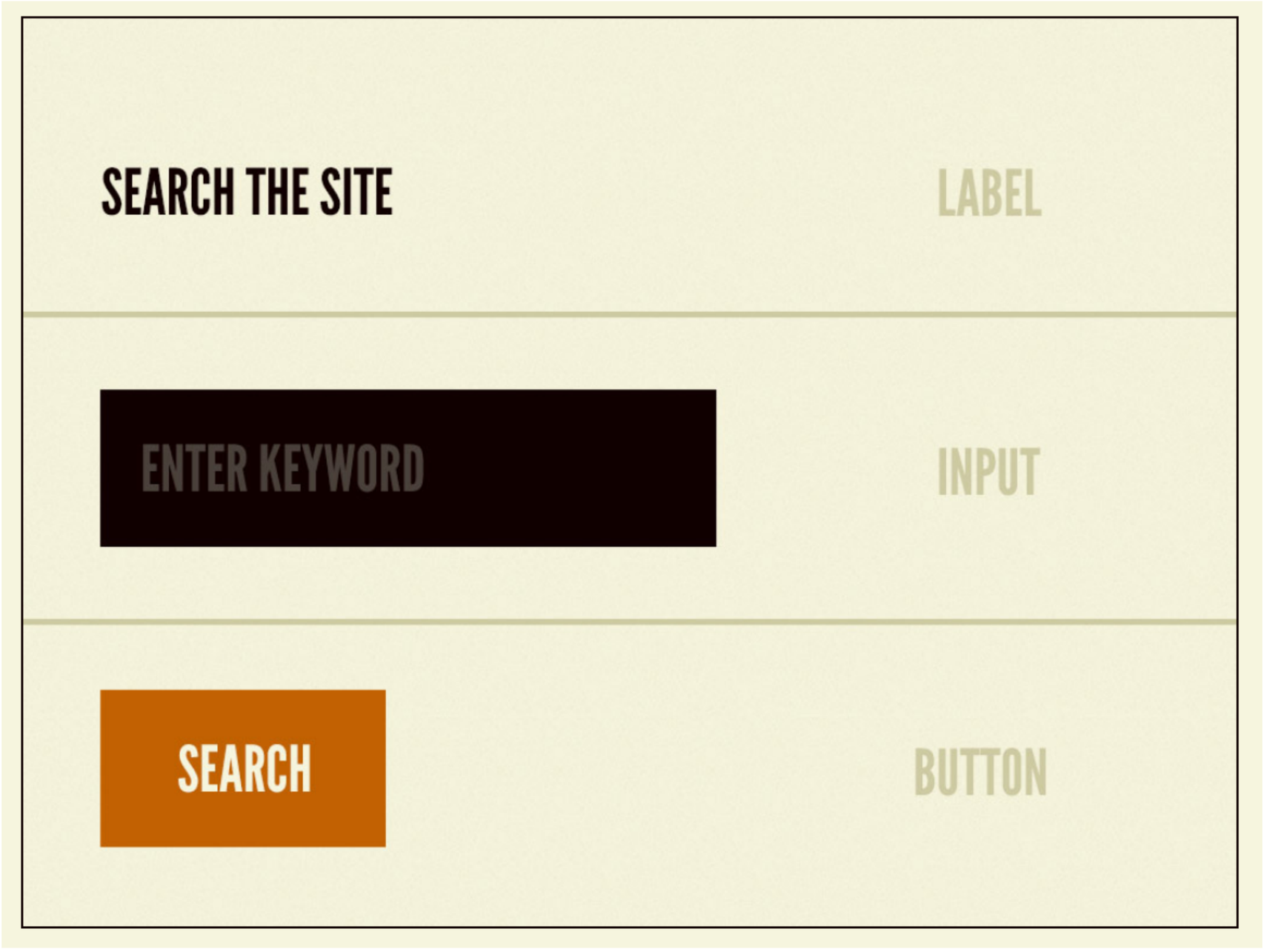
デザインシステムは、アクセシビリティ、ページレイアウト、編集のための手引きとなることがある。中には、ブランディング、データビジュアライゼーション、UXパターン、その他ツールの手引ともなることもある。

Defining Design Systems – EightShapes
https://medium.com/eightshapes-llc/defining-design-systems-6dd4b03e0ff6


design &
development
website
pattern
library


design &
development
design system
website
pattern library


ATOMS
MOLECULES
ORGANISMS
TEMPLATES
PAGES


SEARCH THE SITE
LABEL
ENTER KEYWORD
INPUT
SEARCH
BUTTON

SEARCH THE SITE

ENTER KEYWORD

SEARCH

Home About Blog Contact

SEARCH THE SITE

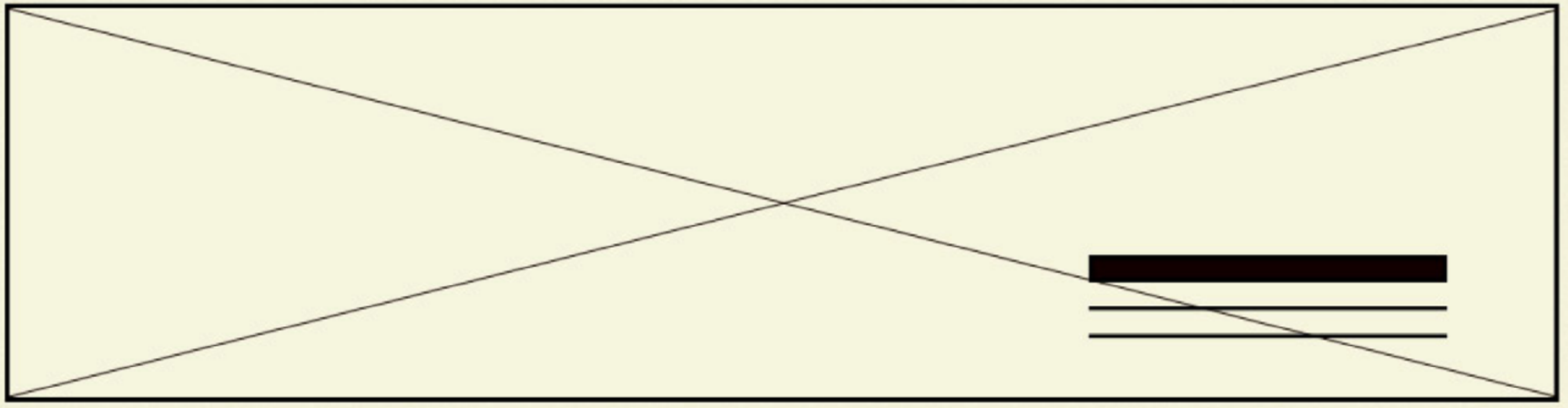
ENTER KEYWORD

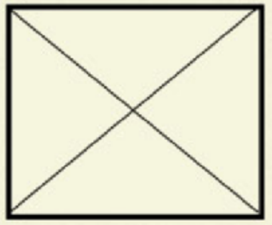
SEARCH

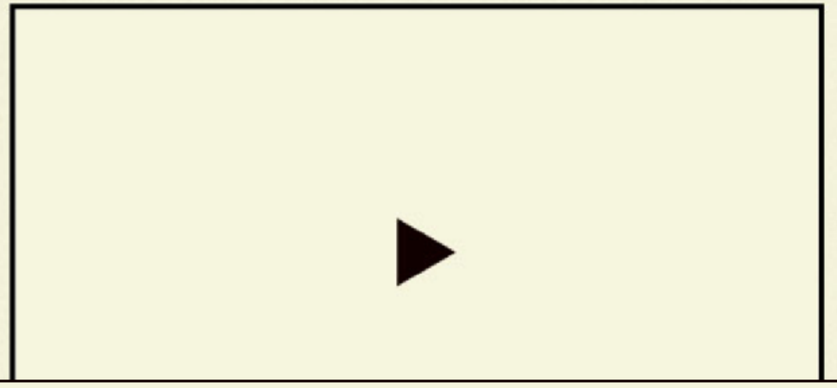
Home About Blog Contact

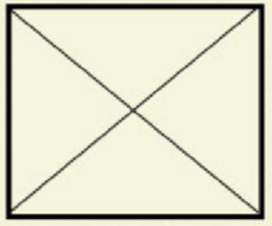
SEARCH THE SITE

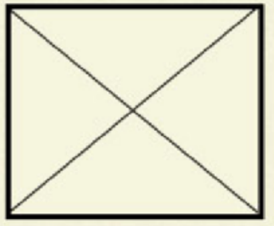
ENTER KEYWORD SEARCH

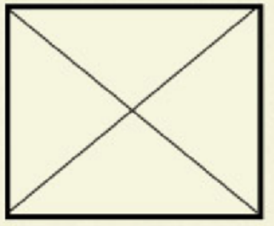












Home About Blog Contact

SEARCH THE SITE

ENTER KEYWORD SEARCH

This Is Real Content.

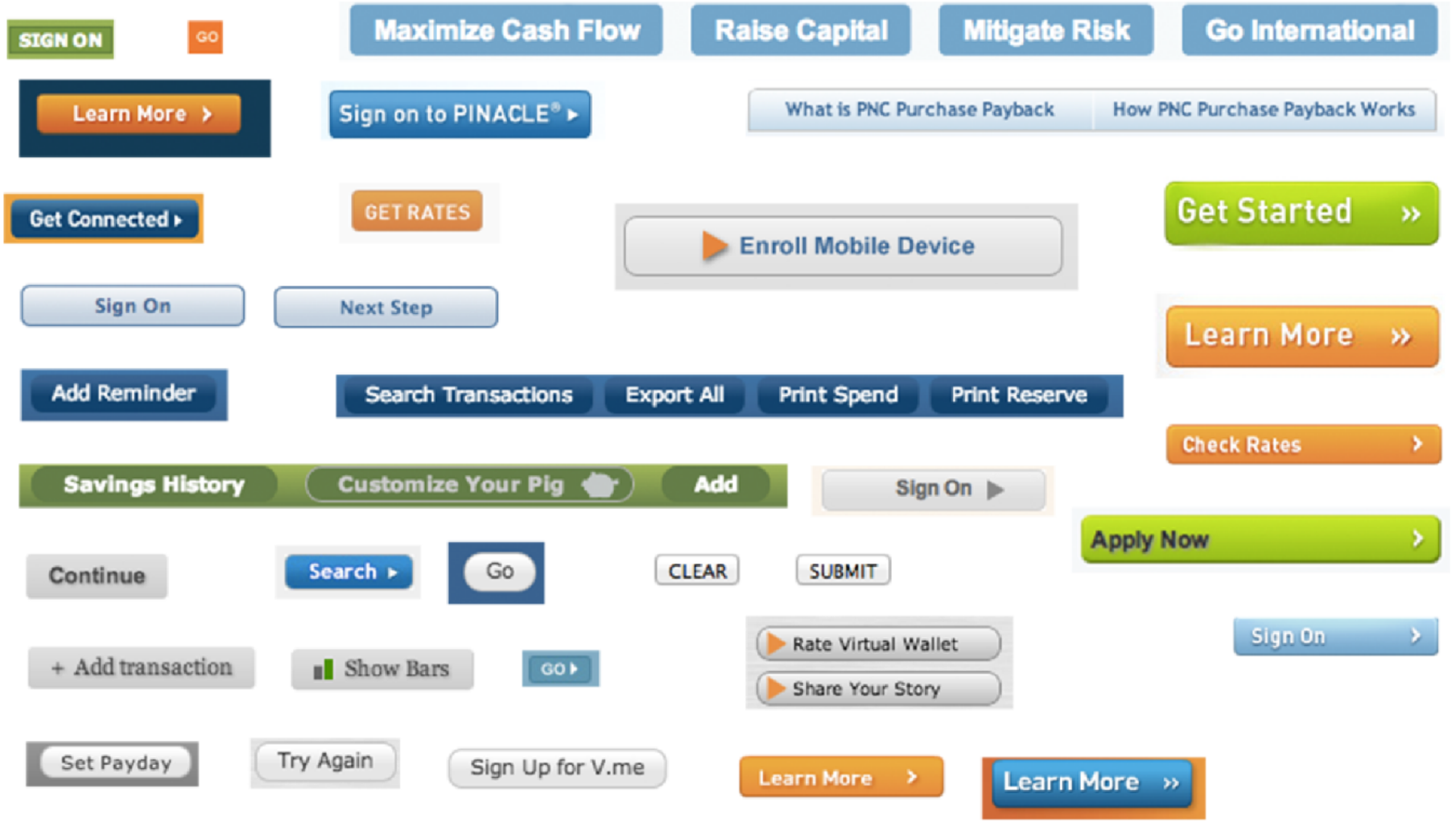
Atomic Designの教え

- 細部から始めて全体を考え、そしてまた細部を見直せ
- そうやって詳細を詰めながら全体も俯瞰し、
  サイト全体で同一の機能を持つUIを統合していく
- 全行程の共通言語となるコンポーネントのライブラリを作る
- デザインシステムがブレなければページのUIもブレない
- デザインのメンタルモデルの話

※ HTML+CSSの実装の話には直接関係ないので注意

どうやってそれを実現する？

- クライアントを説得せよ
- そういうやりかたがベストであるということを納得させよ
- 具体的にはこういうリニューアルの提案方法があるぞ
- そもそもワークフローが工程ごとに
  断絶されたウォーターフローだと無理だ
- というような話が色々書いてある



The Atomic Workflow | Atomic Design by Brad Frost
http://atomicdesign.bradfrost.com/chapter-4/

UX DESIGN

VISUAL DESIGN

DEVELOPMENT

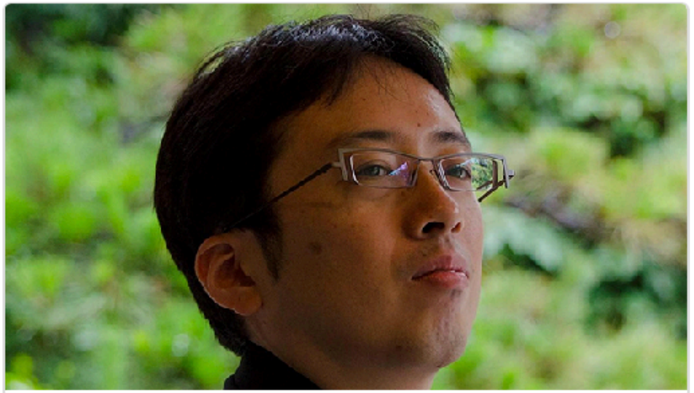
UX DESIGN

VISUAL DESIGN

DEVELOPMENT

Atomic Design
から得られる
CSS設計のヒント

似たようなモジュールが登場する問題



高津戸 壮

Web制作会社、フリーランスを経て、株式会社ピクセルグリッドに入社。数多くのWebサイト、WebアプリケーションのHTML、CSS、JavaScript実装に携わってきた。

jQuery入門

フロントエンドの開発手法は日々進化を続け、新しい情報や技術はあっという間に過去のものになっていきます。そんな中、jQueryは今やウェブサイト開発の土台といえるほどに普及しました。

詳細へ

- そもそもそういうデザインシステムみたいな設計が行える環境でないと回避不可能
- コーディング段階でそれを考えたって答えのないナゾナゾをとくようなもの
- モディファイアにするか別モジュールにするかも同じ

モジュールの名前

- プログラミングにおいても命名は難しい問題
- Atomic Designでも難しいと述べられている
- デザインシステムみたいなこと考えても難しいんだから、
  それがない状態で考えるのは更に難しい
- 何をもって最適な名前かという判断ができない。
  なぜならそのUIをどうサイトで使うか、
  そもそも設計されてないから

我々はAtomic Designを
読んでどう感じるか

- そんな環境で仕事できたらいいなぁ？
- そんな風にキレイなPSDとデザイン指示が来ないかなぁ？
- 現実的にはそんな風にはできないなぁ？

あたりまえ

- そこまでやれるのは、
  そう考えてプロジェクトに挑んだ場合のみ
- そのように全体を統一して設計すること自体が
  大きなチャレンジ
- Atomic Design的なアプローチは価値があるが
  ハードルも色々あると著書には書いてある

- サイトが大きければデザインもブレやすく、
  統合されたUIを提供できなくなる。
  それを根本的に解決するアプローチ

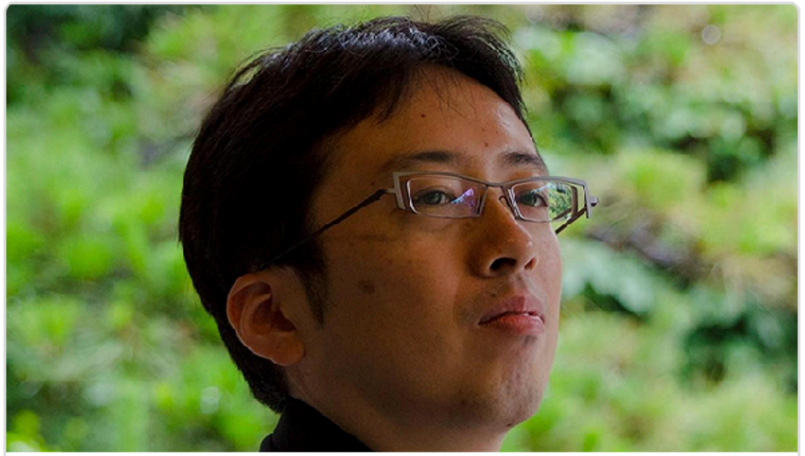
- Atomic Designのようなアプローチが必要とされるのは
  やはりそれなりの規模があるサイトと言えるかも

- 悲しむなかれ
  そこまでやるような仕事ばかりではない
  それが普通

- 完璧なモジュール化を達成するための
  困難さを理解した上で先へ

- では我々はどうCSS設計すればよいのか

モジュールの
汎用性について



高津戸 壮

Web制作会社、フリーランスを経て、株式会社ピクセルグリッドに入社。数多くのWebサイト、WebアプリケーションのHTML、CSS、JavaScript実装に携わってきた。

jQuery入門

フロントエンドの開発手法は日々進化を続け、新しい情報や技術はあっという間に過去のものになっていきます。そんな中、jQueryは今やウェブサイト開発の土台といえるほどに普及しました。

詳細へ

同じ見た目のUIがある。使われてる場所は別。
この2つは見た目が同じだから同じHTMLで表現しよう

と、コードを書く人間的には考えたいところだが

必ずしもそう考えなくて良い

汎用的なモジュールとして考えた場合

Card

高津戸 壮

Web制作会社、フリーランスを経て、株式会社ビクセルグリッドに入社。数多くのWebサイト、WebアプリケーションのHTML、CSS、JavaScript実装に携わってきた。

Card

jQuery入門

フロントエンドの開発手法は日々進化を続け、新しい情報や技術はあっという間に過去のものになっていきます。そんな中、jQueryは今やウェブサイト開発の土台といえるほどに普及しました。

詳細へ

- 再利用前提
  CSSの変更無しで展開しやすい
- バリエーション表現のために
  複雑化する可能性がある
- 影響範囲が読めないので
  後からCSSを編集しづらい
- CSSの容量を抑えられる
- まとめて変更するのが楽

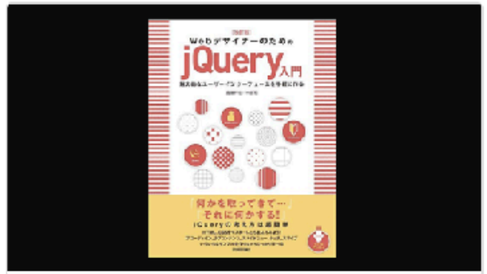
局所的なモジュールとして考えた場合

MemberCard

高津戸 壮

Web制作会社、フリーランスを経て、株式会社ピクセルグリッドに入社。数多くのWebサイト、WebアプリケーションのHTML、CSS、JavaScript実装に携わってきた。

ProductCard



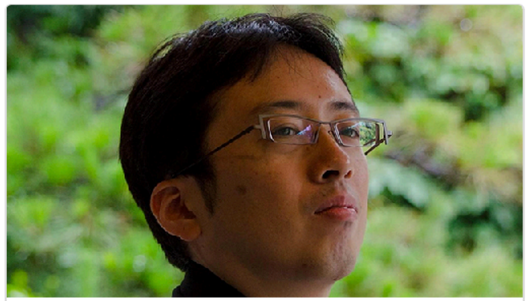
jQuery入門

フロントエンドの開発手法は日々進化を続け、新しい情報や技術はあっという間に過去のものになっていきます。そんな中、jQueryは今やウェブサイト開発の土台といえるほどに普及しました。

詳細へ

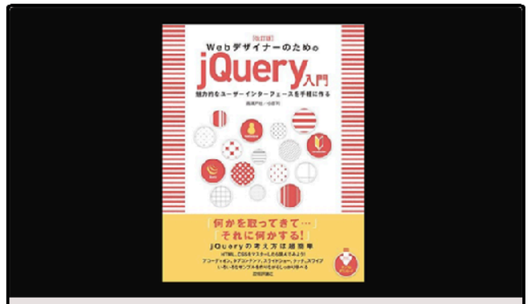
- 再利用しない前提
  同じ見栄えのモジュールが
  登場しても機能が異なるなら別物
- そこでしか使わないから複雑になりにくい
- CSSのコードが重複する
- 影響範囲が読みやすいので
  後からCSSを編集しやすい
- CSSの容量が増える
- まとめて変更するのが困難

例えば商品のモジュールだけ変えたい



高津戸 壮

Web制作会社、フリーランスを経て、株式会社ピクセルグリッドに入社。数多くのWebサイト、WebアプリケーションのHTML、CSS、JavaScript実装に携わってきた。



jQuery入門

フロントエンドの開発手法は日々進化を続け、新しい情報や技術はあっという間に過去のものになっていきます。そんな中、jQueryは今やウェブサイト開発の土台といえるほどに普及しました。

詳細へ

汎用的な書き方

変更がどこかに影響してしまう可能性あり。モディファイアを増やして対応？
モディファイアどこまで増えるの？
もしくは別の汎用的なモジュールとして分ける？

局所的な書き方

既に別のモジュールとして考えているのであまり深く考えなくても良い

汎用的にするか局所的にするか

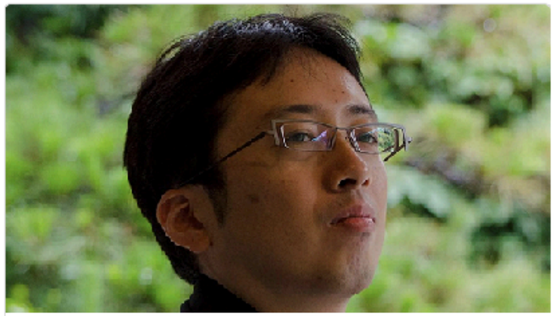
- Atomic Design的に言えば、
  機能が同じなので同じUIを使いましょうとか、
  差別化したいので2種類の機能の異なるモジュールとして
  考えましょうなどとなるハズ

- いつもそこまでできるわけじゃない
  だから別物にしたっていい

- 運用の効率を考えると局所的にしたほうが良いこともある

名前空間による
モジュールの
グルーピング

Card



高津戸 壮

Web制作会社、フリーランスを経て、株式会社ピクセルグリッドに入社。数多くのWebサイト、WebアプリケーションのHTML、CSS、JavaScript実装に携わってきた。

Card

jQuery入門

フロントエンドの開発手法は日々進化を続け、新しい情報や技術はあっという間に過去のものになっていきます。そんな中、jQueryは今やウェブサイト開発の土台といえるほどに普及しました。

詳細へ

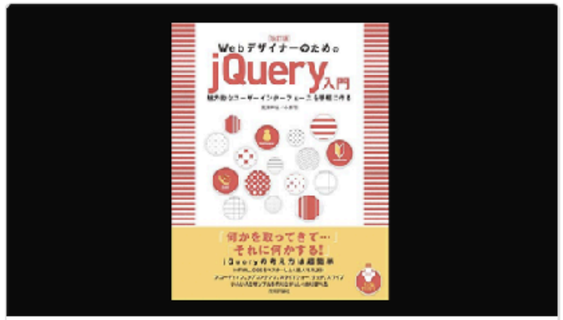
```
.Card { ... }
.Card__Img { ... }
.Card__Body { ... }
.Card__Title { ... }
.Card__Text { ... }
```

memberIntro-Card

高津戸 壮

Web制作会社、フリーランスを経て、株式会社ピクセルグリッドに入社。数多くのWebサイト、WebアプリケーションのHTML、CSS、JavaScript実装に携わってきた。

productList-Card



jQuery入門

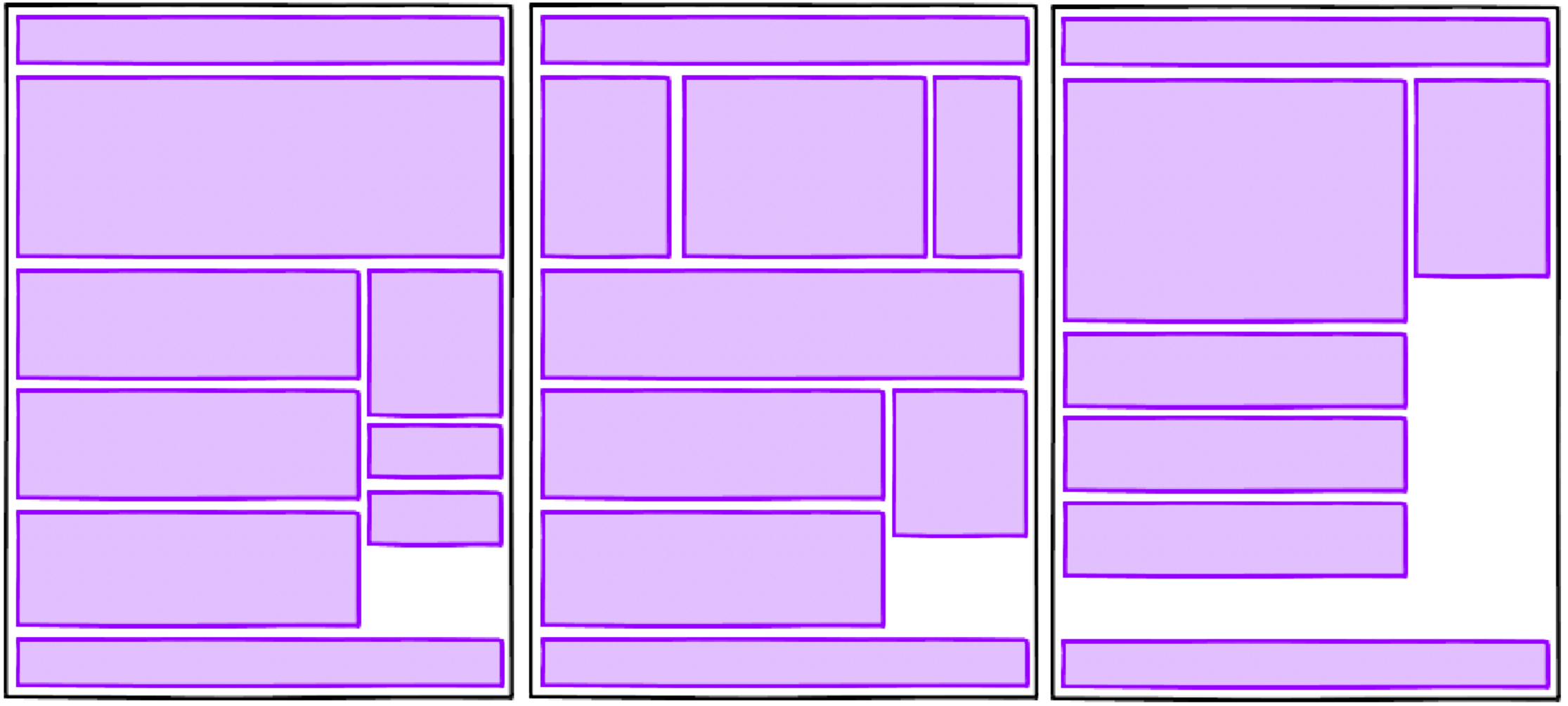
フロントエンドの開発手法は日々進化を続け、新しい情報や技術はあっという間に過去のものになっていきます。そんな中、jQueryは今やウェブサイト開発の土台といえるほどに普及しました。

詳細へ

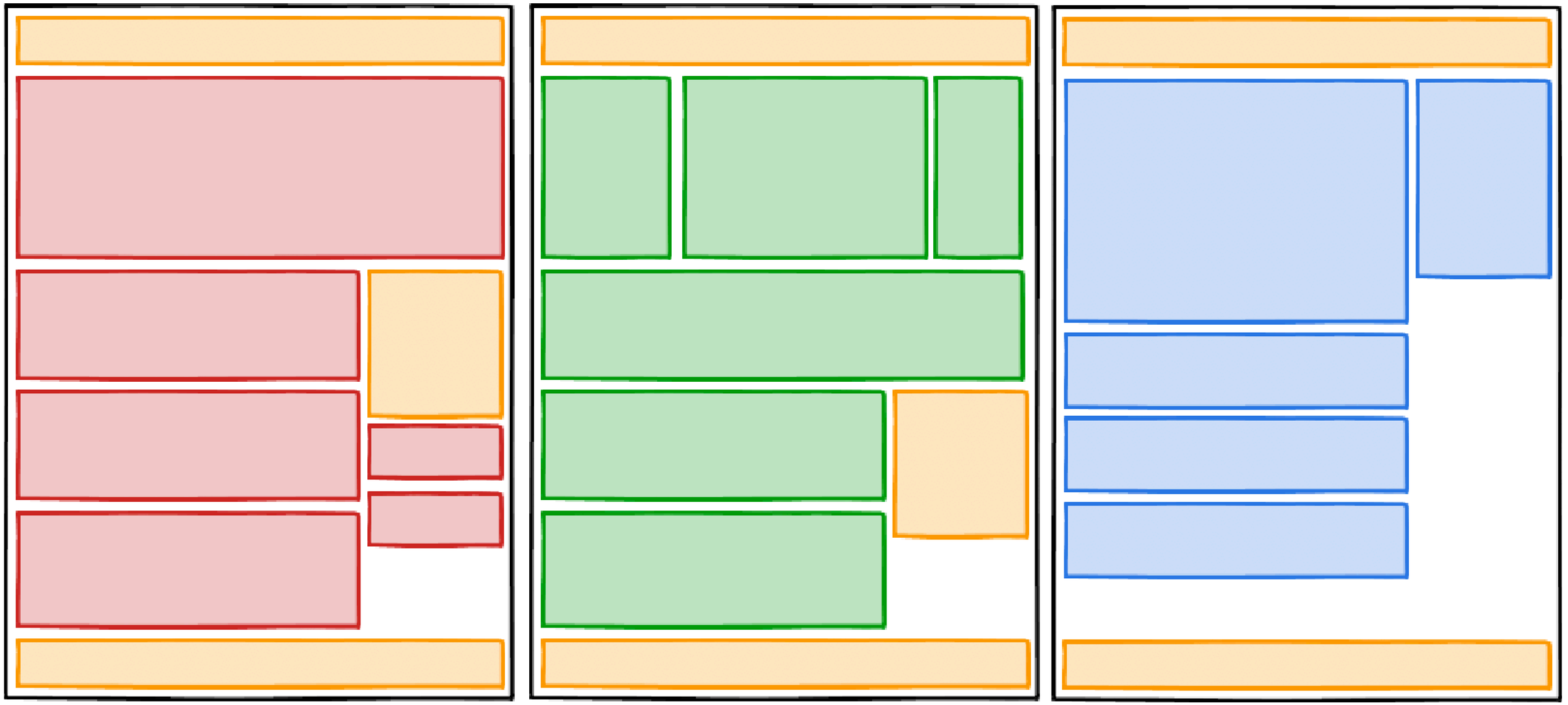
```
.memberIntro-Card { ... }
.memberIntro-Card__Img { ... }
.memberIntro-Card__Body { ... }
.memberIntro-Card__Title { ... }
.memberIntro-Card__Text { ... }

.productList-Card { ... }
.productList-Card__Img { ... }
.productList-Card__Body { ... }
.productList-Card__Title { ... }
.productList-Card__Text { ... }
```

名前空間で分類

- Enduring CSSというCSS設計方法論で紹介されている方法
- BEMのBlockを名前空間でグルーピングする（端的にいうと）
- モジュール名の衝突を防げる
- 同一名前空間内でのモジュール名重複にのみ
  気をつければよくなる
- モジュール名自体はシンプルになる



structure-



topPage-

productDetail-

shoppingCart-

注意事項

- 名前空間をどう切るかという設計が必要
- 名前空間がうまく設計されていないと悲惨
- UIをどう設計するかと併せて考えるとスムーズ

HTML上で組み立てる
Atomic CSS

Atomic CSS
https://acss.io/

```
<div class="ColumnA clearfix">
  <h2>お問い合わせ</h2>
  <p class="Text">お問い合わせについては云々</p>
  <p class="Text align-right">TEL: 090-1122-3333</p>
</div>
```

いわゆるユーティリティ的なクラス

```
<div class="
  margin-10
  borderRadius-6
  bg-blue
">...</div>
```

```
.margin-10 { margin: 10px; }
.margin-20 { margin: 20px; }
.borderRadius-3 { border-radius: 3px; }
.borderRadius-6 { border-radius: 6px; }
.bg-blue { background-color: #00f; }
```

そんなクラスだけでページを作る

Atomic CSS

- 単純なスタイルを当てたクラスをHTML上で組み合わせる
- CSSにもたらされる複雑さをHTML上に移したアプローチ
- ひとかたまりのUIを、カプセル化したHTML+CSSで表現する
  多くのCSS設計方法論とは大きく異る

※ Atomic Designとは関係ありません

Home Mail Flickr Tumblr News Sports Finance Entertainment Lifestyle Answers Groups Mobile More Install the new Firefox »

MY YAHOO! Search Web Sign in Mail

Welcome to My Yahoo

Get your headlines, email, quotes and more — all in one place.

Customize your page » Existing user? Show my page

東京, 東京都 (Current Location)

58 °F | °C Cloudy

Today	Fri	Sat
69° 46°	70° 55°	68° 51°

Scoreboard

Trending

Yesterday Today Tomorrow

Buffalo NY Jets	8:25 pm ET NFLN	
Ball St. East. Michigan	6:00 pm ET CBSS	
Northern Illinois Toledo	6:00 pm ET ESPU	
Navy Temple	8:00 pm ET ESPN	
Idaho Troy	9:15 pm ET ESPU	

Trump calls for death penalty for terror suspect

The president tweets that Sayfullo Saipov, the Uzbek immigrant accused in the deadly New York truck attack, "SHOULD GET DEATH PENALTY!"

Would consider sending him to Gitmo » 1-3 of 40


Trump calls for death penalty for terror


For first time, Astros are World Series


Don't ask Julia Roberts this question

News For You

What Carlos Correa's teammates knew about his World Series proposal plan

Astros players knew about Carlos Correa's plan to propose if they won the World Series

Big League Stew

Your Best Chance of Living the American Dream AdChoices

The Japanese's Opportunity to get a Green Card to Live & Work in the USA!

USAGCL Sponsored

My Mail

Check multiple mail accounts from one place

Already have an account? Login

Don't have one? Sign Up

My Portfolio

Markets

Last updated at 09:30AM EST

Name	Price	Change	% Chg
S&P 500	2577.78	-1.58	-0.061
Dow Jones In...	23434.74	-0.27	-0.001
NASDAQ Co...	5709.391	-7.143	-0.106

```
<table class="W(100%) M(0)! P(0) H(100%)">
  <tbody>
    <tr>
      <td class="W(100%) W-100 Va(t) Px(0)">...</td>
    </tr>
  </tbody>
</table>
```

Atomic CSSをどう活かすか

- モジュール的な考え方が主流だがこういうアプローチもある
- 問題解決の方法は一つではないという参考
- モジュールベースの設計でも、余白の制御等、
  部分的に設計に取り入れるなど検討の余地がある
- 手でガリガリ多様なバリエーションのUIを書いていく仕事向け？

どう設計するかは
どう決めればいいか

最適なCSS設計とは何か

- 自分が経験してきた方法がうまくいったなら
  それがベストな方法であると思いがち
- しかしそれはその人が経験した要件に
  マッチした方法で設計できたにすぎない（かもしれない）
- 様々な要件の元でどう設計したかという経験が
  設計の力となると言えるのではなかろうか
- とりあえず自分はそう思うので、
  自分がどういう環境で
  どう設計してきたのかを共有しよう

書いて終わりではないのが
HTML+CSS

- 前の工程と後の工程がある
- 自分の作業工程以外へ目を向ける

後の工程とは

- 手で大量のページを量産
- React、Angular等のSPA
- CMSへの組み込み
- クライアントが運用

などなど

設計例

WYSIWYG

lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

1. one
2. two
3. three

```html
<p class="wysiwyg__p">The quick brown fox jumps over the lazy dog.</p>

<ul class="wysiwyg__ul">
  <li class="wysiwyg__li">one</li>
  <li class="wysiwyg__li">two</li>
  <li class="wysiwyg__li">three</li>
</ul>
```

```css
.wysiwyg__p { margin: 0 0 1.5em; }
.wysiwyg__ul { margin: 0 0 1.5em; }
.wysiwyg__li { margin: 0 0 0.5em; }
```

BEM的にはこうしたい

```html
<div class="wysiwyg">
  <p>The quick brown fox jumps over the lazy dog.</p>
  <ul>
    <li>one</li>
    <li>two</li>
    <li>three</li>
  </ul>
</div>
```

```css
.wysiwyg p { margin: 0 0 1.5em; }
.wysiwyg ul { margin: 0 0 1.5em; }
.wysiwyg li { margin: 0 0 0.5em; }
```

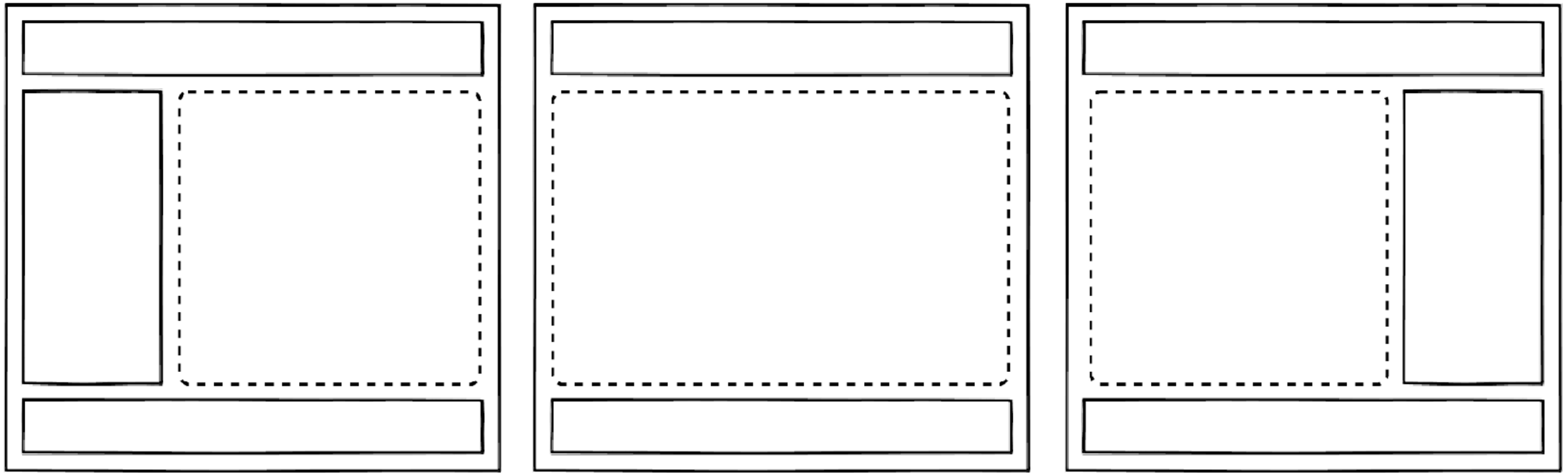
でもこうするしかない

- BEMルールに従っていても
  この部分については諦める必要がある

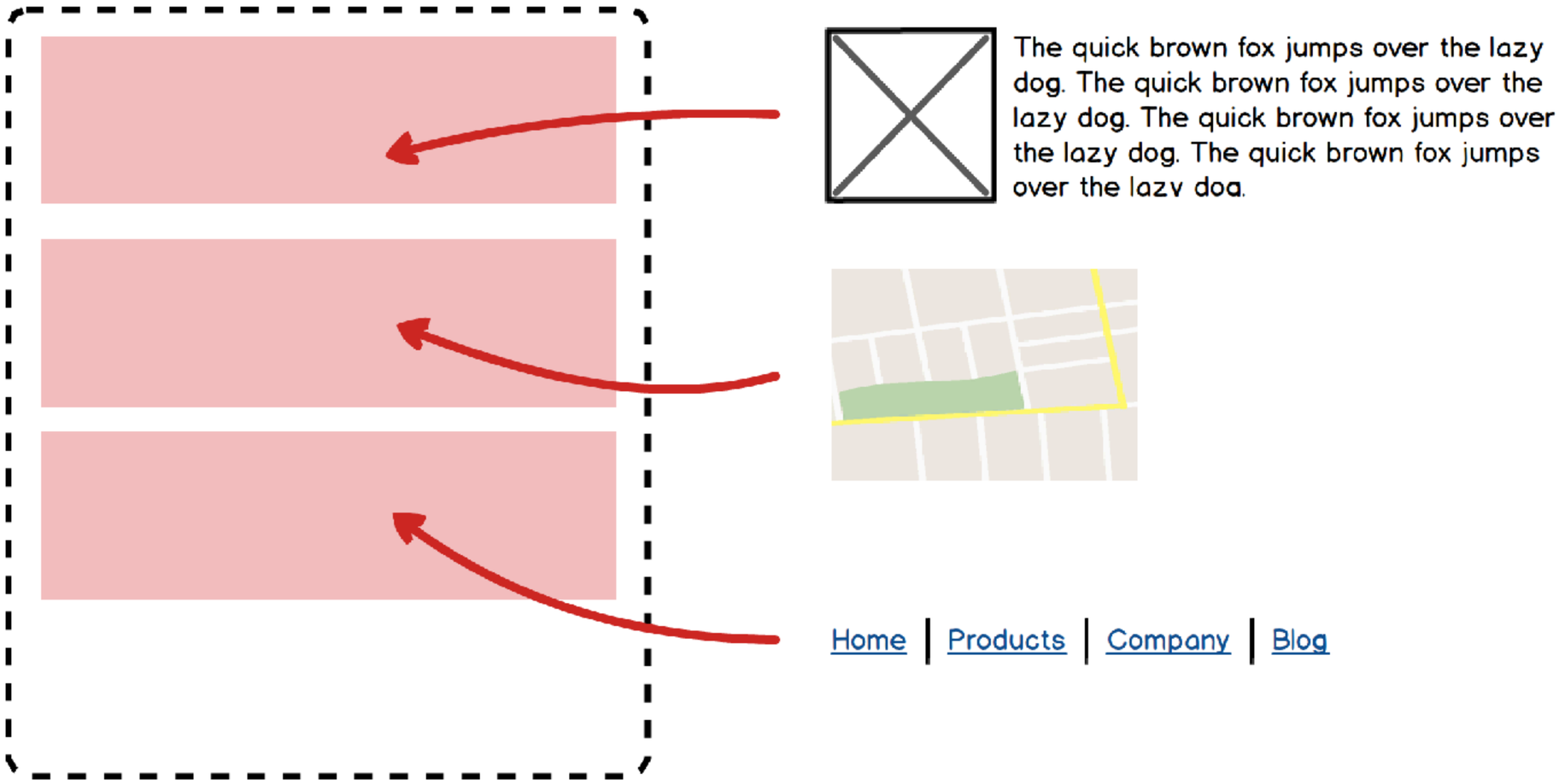
- 後工程がCSS設計を決定する例

モジュール積みCMS

- コンサルからCMS組み込みまでワンストップで請負
- 設計〜デザイン〜コーディング〜CMS導入
- コーポレートサイトを数多く制作
- Atomic Design的なアプローチに似てる



用意したテンプレートの中から一つを選択



その中に用意したモジュールを積んでページを作成できる

- 採用していたCSS設計はBEM的なもの
- そのCMSではモジュールを並べてページが作れるだけ
  モジュールの入れ子ができない
- だから作ったモジュールを並べて
  完結するような設計が求められる
- 余白のルールも複雑にできない（例: 非section）

- コーポレートサイトを作ることが多かった

- 全行程を社内で完結させていたので、
  どういうモジュールを用意すればよいか
  ナレッジが蓄積されていた

- ワイヤーフレーム設計段階でどういうモジュールを
  用意したらいいか、全行程の人間がある程度理解している

- CMSに組み込むモジュールの単位を意識してHTML+CSSを書く
  ワイヤーフレームでのモジュール感もCMSでできること基準

- 最終工程に合わせて設計

第三者によるページ量産

- テンプレ作って渡してクライアントの担当者が量産
  もしくは社内で複数人使って大量のページを量産
- 採用していたCSS設計はBEM的なもの
- スタイルガイドを作って渡す
  スタイルガイドがページ量産のキモ

- 量産する人はCSSを基本いじらない
- クライアント担当者が簡潔に量産できるのが最重要
- できるだけ単純に。
  複雑過ぎるテンプレ作っても誰も幸せにならない
- section的な構造もなるべく回避
- モジュールベースの設計に、
  汎用的に使えるユーティリティクラスを用意

SPAなWebアプリの話

- 旧バージョン（AngularJS1系）を3年ほど運用
  → 新バージョン（Angular最新系）で作り直した
- エンタープライズなWebアプリ
- APIから取得したデータでページを構築
- BEM → AngularのComponent Styles（scopedなCSS）

Hello Angular! v4.3.3

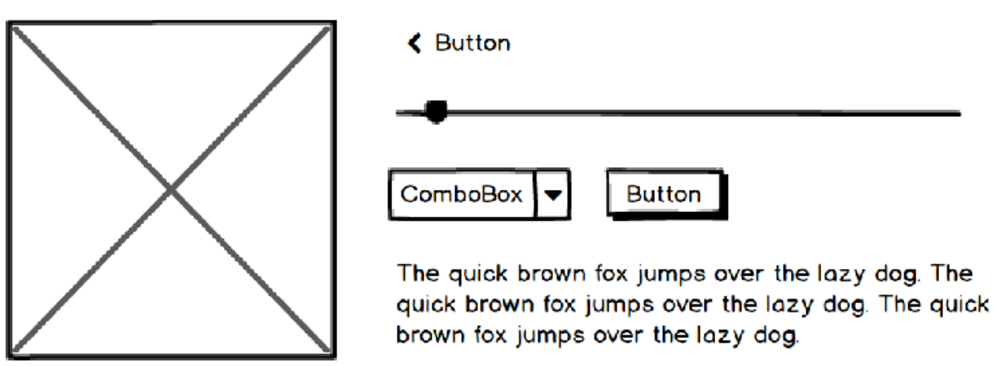
```
@Component({
  selector: 'my-app',
  template: `
    <div>
      <h2>Hello {{name}}</h2>
    </div>
  `,
   styleUrls: ['my-component-styles.css']
})
```

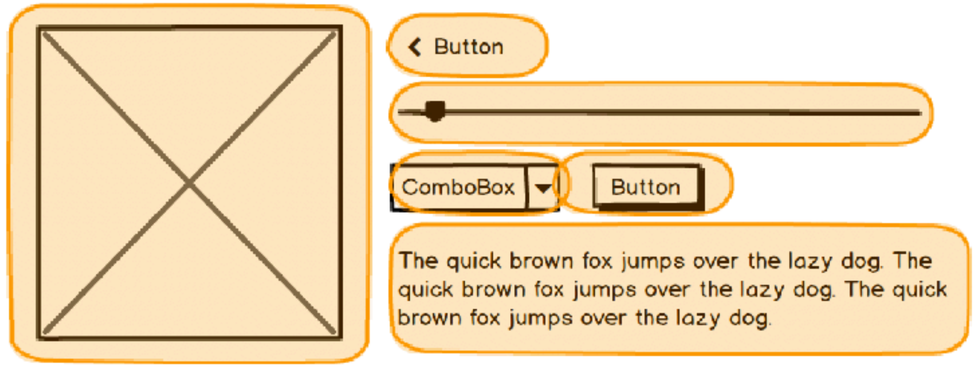
```
<my-app></my-app>
```

```
h2 { border: 2px solid red; }
```

旧設計で起こっていた問題

- 汎用的に使える細々としたUIパーツを
  ブロックとして考え、CSS設計
- その組み合わせでAngularJSに組み込む
  HTMLテンプレートを作成
- ブロックの組み合わせをAngularJSの directive、
  controllerといったモジュール化の仕組みで管理



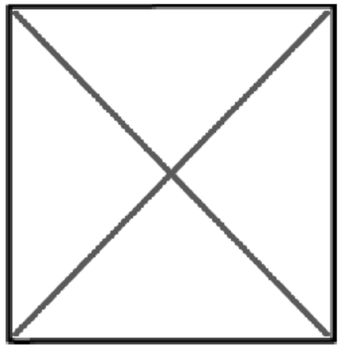


- とあるモジュールを編集する際、
  そのHTMLは複数のBEM的ブロックの集合で成り立っている

- そのCSS的モジュール感を把握するのが難しい
  （特にチームでの作業において）

- 汎用化されたモジュールのCSSをいじると
  どこか違う箇所で崩れが発生する可能性があるが確認が大変過ぎる

- 既に書かれているCSSへの変更は極力避け、
  新しいクラスを追加して対応をするのを基本とする方針になってしまった

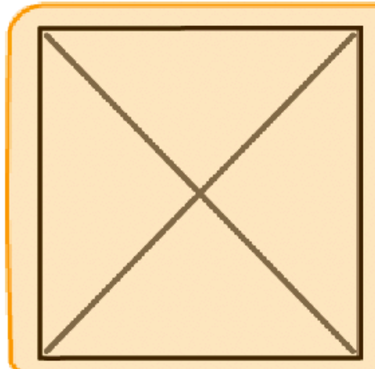


Button

ComboBox

Button

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.



Button

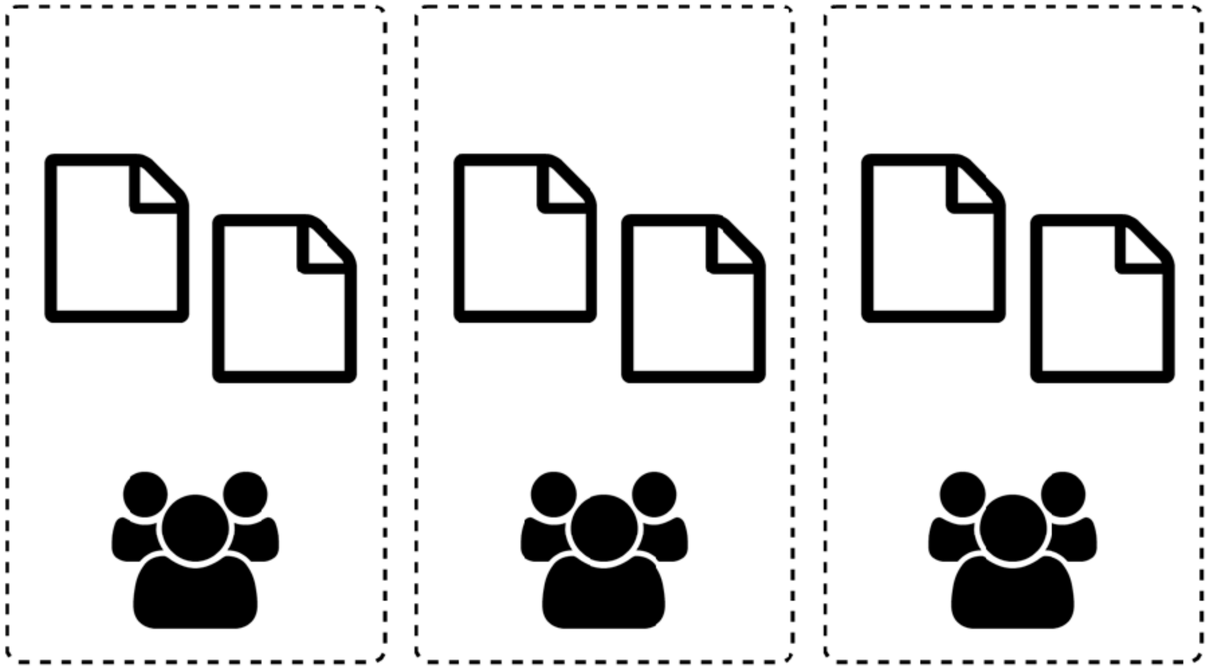
ComboBox

Button

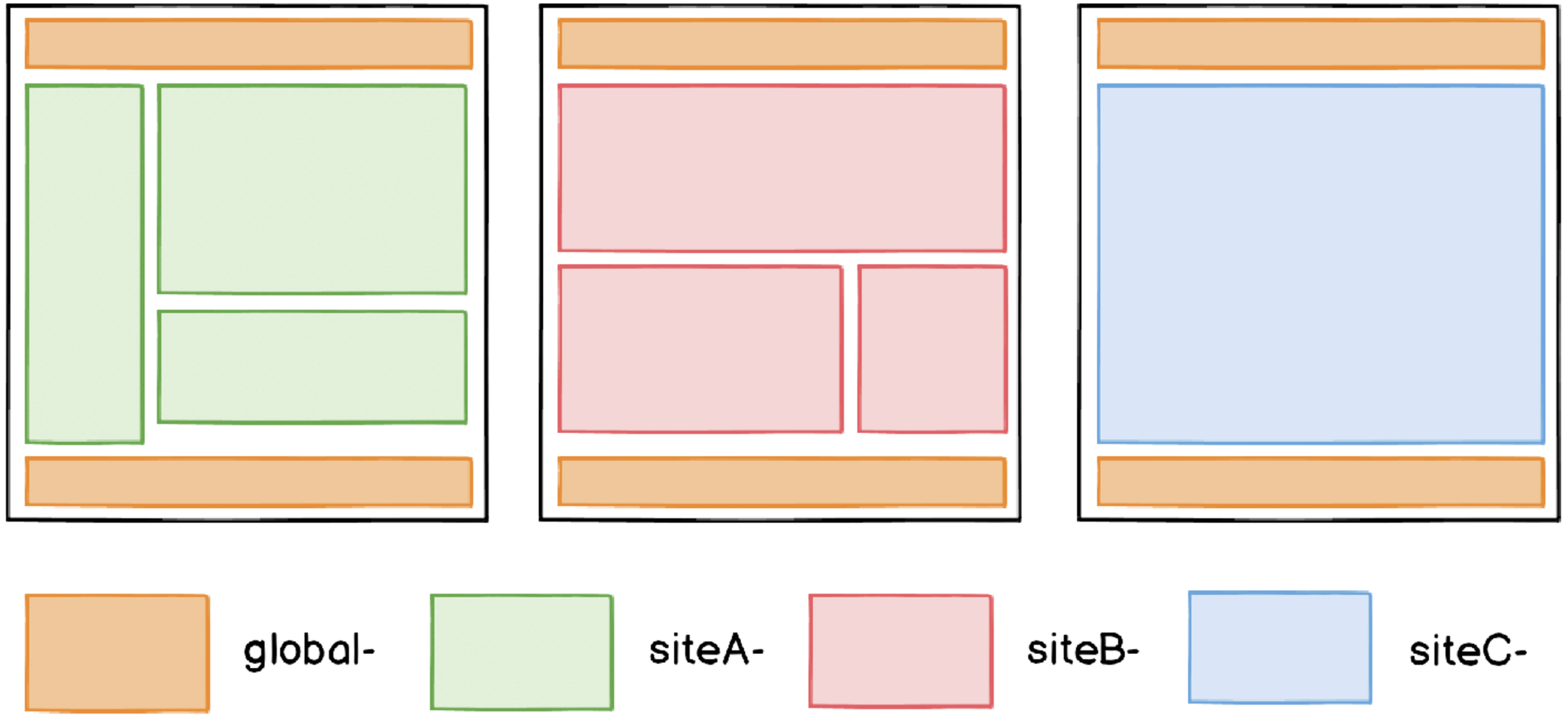
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

- 新バージョンは基本、局所的なモジュールとして考えるように設計
- Angularでどのようにモジュールを切るかを先に決定。
  Component Stylesで、モジュール内だけでCSSを完結させる
- 似たようなUIが登場してもAngular的なモジュールとして
  別物であれば、別にCSSを書く
- 後の変更や修正時に影響範囲が把握できて保守性向上
- 後工程に寄せたCSS設計

制作チーム分離パターン



一つのサイトを複数のチームが運用する
スタイルの競合、モジュール感共有の難しさが問題



ヘッダ・フッタだけ共通で global 名前空間
他のコンポーネントについては各々に名前空間を用意

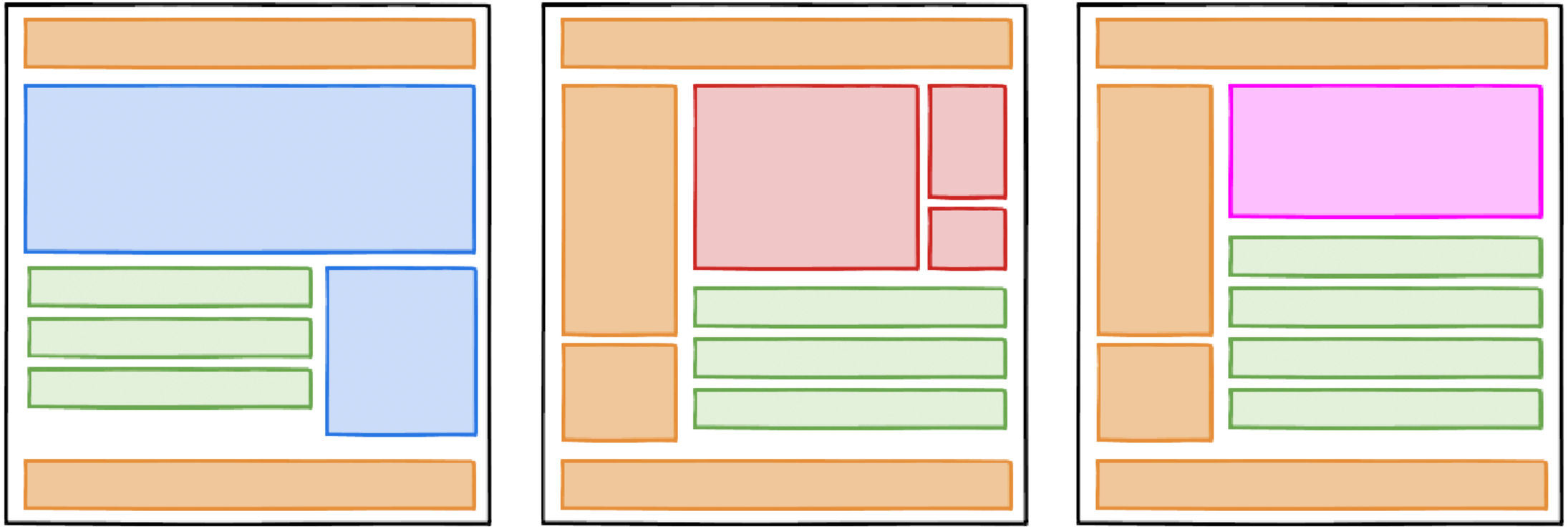
- 複数チームでうまく運用できないという課題
- 分離による解決
- 運用の課題を解決するためのCSS設計
- 全体としてモジュールを整理するという
  Atomic Design的なアプローチも
  もちろんありえる

汎用的モジュール ＋ 局所的モジュール

- どこでも使えることを前提とする、
  汎用的なモジュールを基本とする
- 画面によってはその画面にしか登場しない
  ユニークなモジュールが登場する
- という構成は多くのプロジェクトにマッチしないだろうか

structure-

siteWide-



topPage-

productDetail-

company-

UIの設計としてもそのように考えていればなお良し

結局はそれ

私的結論

- CSSを書く上で悩む原因はそもそも
  コーディング段階で解決できないものである場合がある

- しかしそれがCSSを書く上での問題として表面化してくる

- CSSを書いていて解決できる問題ではない
  CSS設計やツールにその助けを求めるのはそもそも誤り

- ある程度の設計の不完全さを受け入れてCSSの設計を行う
  ことも多いことを念頭に置く

- 解決したい問題に寄せてCSSの設計を考える
- 前工程、後工程を考慮＆連携してCSSの設計を考える
- 他工程を巻き込んで設計を一緒にやる

単純にコードを書く以上の価値はそのあたりに存在する

ありがとうございました

CodeGrid

たい、Flexible Box　チーム開発に効く環境構築術　Snap.svgで快適SVGアニメーション　Webマップの実装手法　JavaScript再入門　ちよ
実践！AngularJS　フロントエンドのサウンド実装　ピクセルグリッド的ブックガイド　制作業務のコミュニケーション　テンプレートエンジンのススメ　最適な
すよう　エンジニアのためのデザイン入門　ソーシャルコーディングのススメ　JavaScript開発のためのテスト入門　エンジニアに聞く、仕事の行方　フロントエ
ュリティ　ピクセルグリッドスタッフの情報収集　クライアントとのコミュニケーション　Compassで簡単、CSSスプライト作成　快適ア
e、Marionette　マークアップ・エンジニアのためのSVG入門　大規模プロジェクト運用のコツ　Webの文字の読みやすさ　価値あるCSSの設計と運用のため
の仕事術　技術選　今、どんな書き方してますか？　CodeGridを振り返る　賢いMedia Queries管理　ルーキーに捧ぐ、codeGridおすすめ記事　プロキ
ーズに　ピクセルグリッド技術サーベイ　CodeGridの作り方　ピクセルグリッドが訪ねる、開発の現場　静的HTMLのためのテンプレートエンジン　Webプラ
est.Asiaレポート　開発合宿レポート　ブラウザにおける縦書き最新事情　注目したいのはこの技術　仕様書の読み方　CSSプロパティ編　フロントエンド最新技
ント　自由自在　これからのグリッドレイアウト　自作ツール作成から学ぶノウハウ　AWSで始めるサーバーレス・アーキテクチャ　three.js使いこなし　AWS
イトの公開　これから始めるReact.js　実践、SVGスプライト　Falcorで実現する効率的なfetch　実践、ユニットテスト　はじめてのaigis　レスポンシ
キストのコントロール　Enduring CSSの設計思想　ピクセルグリッドのフォント選定と実装のプロセス　これから始めるReact.js 発展編　いままでの印刷
lcorで実現する効率的なfetch サーバー編　根っこから学ぶCSS3アニメーション　どうする？ブラウザ機能格差の解消　実践！スマホサイトのviewp
dのこと　考えた！ハマった！エンジニアの転況　表示速度改善　Data URIスキーム　CoffeeScriptをめぐる議論　コードでつぶスマホサイト制作　今、
ピクセルグリッドの仕事術　初心者のためのjQuery　jQuery Conf. レポート　webGLの基礎　ピクセルグリッド的ブックガイド　CoffeeScriptで学ぶ
Conferenceレポート　基本のShadow DOM　おすすめのライブラリつまみ食い　Webアプリ構築のためのBackbone.js入門　HTML5

CSS Nite LP54

Coder's High 2017

2017.11.4