

ぼくのかんがえた
ふつうのじえいえす

Skill U Friday 2013.07.17

@ahomu

CyberAgent, Inc.

内容

1. コミュニティにおけるView
2. ライブラリの選定と経緯
3. Backbone.jsとの付き合い方
4. ~~ユニットテスト構成~~
5. 今後の課題

ケーススタディと プラクティス



ふつうです



**ふつうのプラクティスを
理性的なコードで書く**



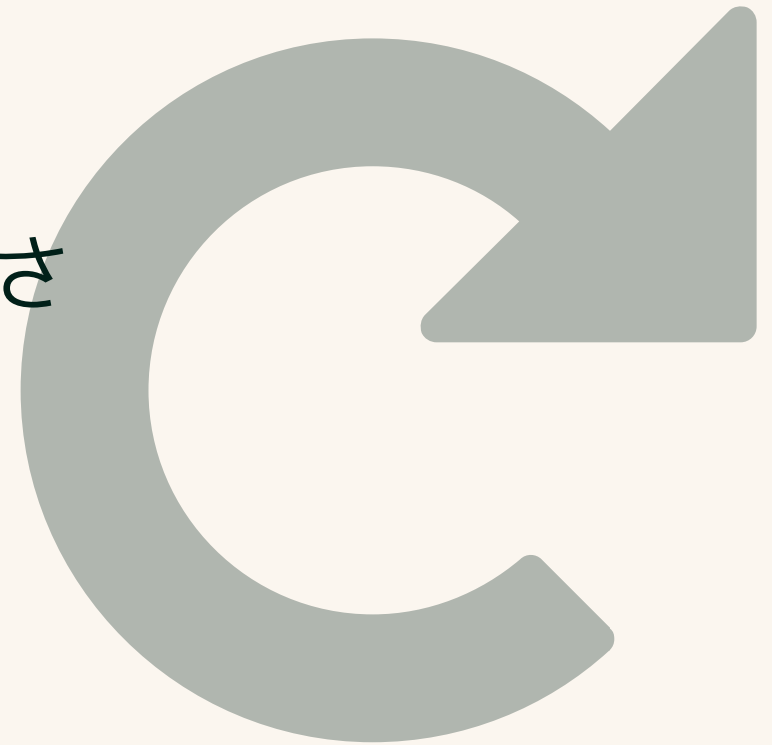
コミュニティにおける Viewの傾向と対策



イニシャルロードが速い 実行時のレスポンスが速い

- ・ スクロールの滑らかさ
- ・ アクションに対するレスポンスの良さ
- ・ 操作フィードバックの丁寧さ

etc...



縦方向が長~~~~~い

多くのPRJが 3,000~4,000px ある
→スクロールパフォーマンス超重要

DOMContentLoadedのあとに
JSで遅れて描画されると
コンテンツの高さが変わって不安定...

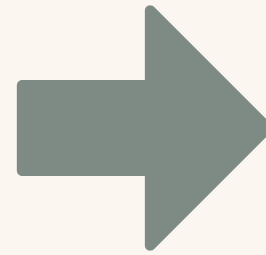
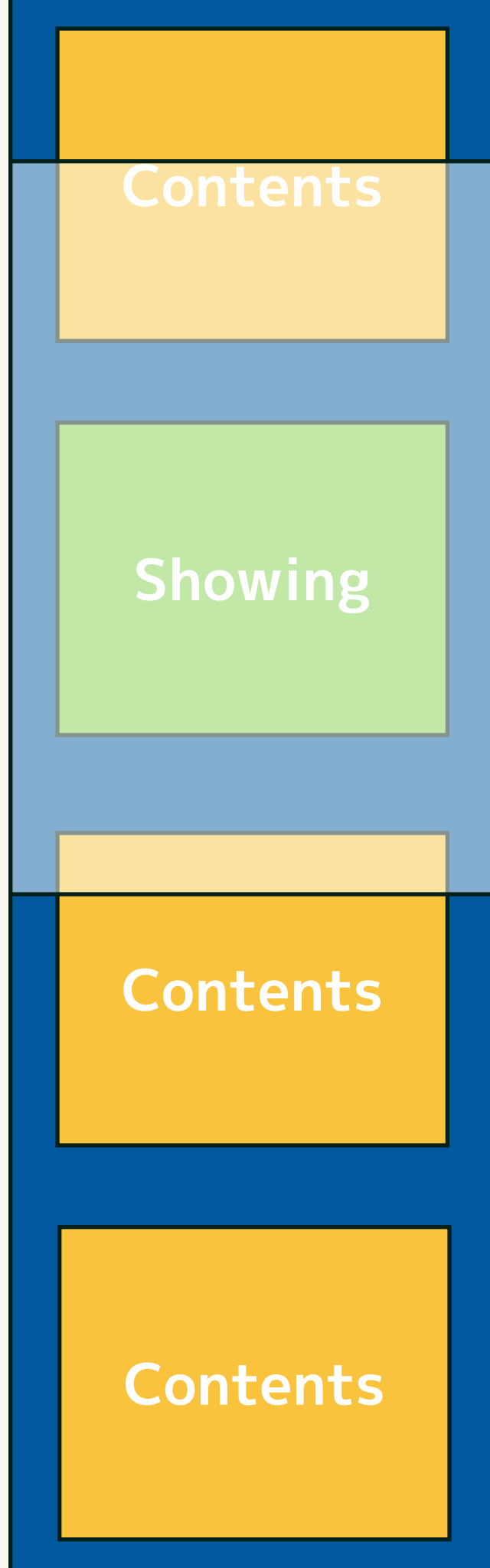
スクロールパフォーマンス

スクロール時に30～60FPSを維持
1フレーム約16msの処理が理想

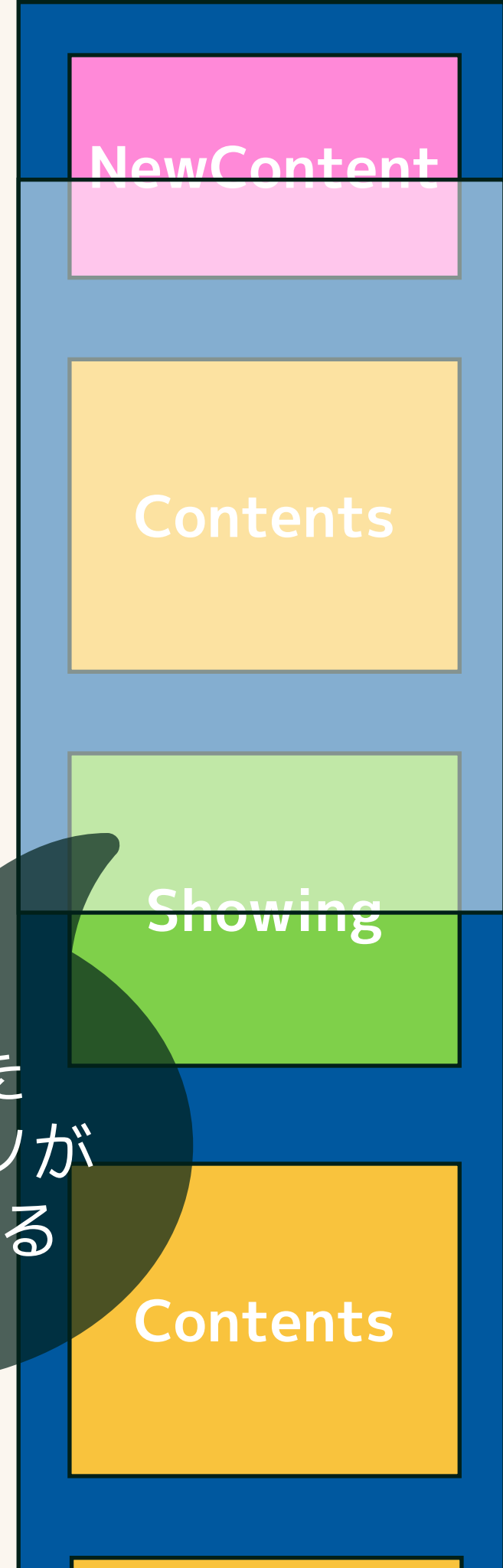
詳しくは別スライドを参照のこと

<https://speakerdeck.com/ahomu/high-performance-web-frontend>





非同期
コンテンツの
描画が発生



JSON

(Object Literal)

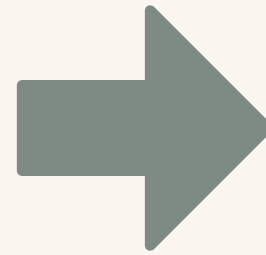
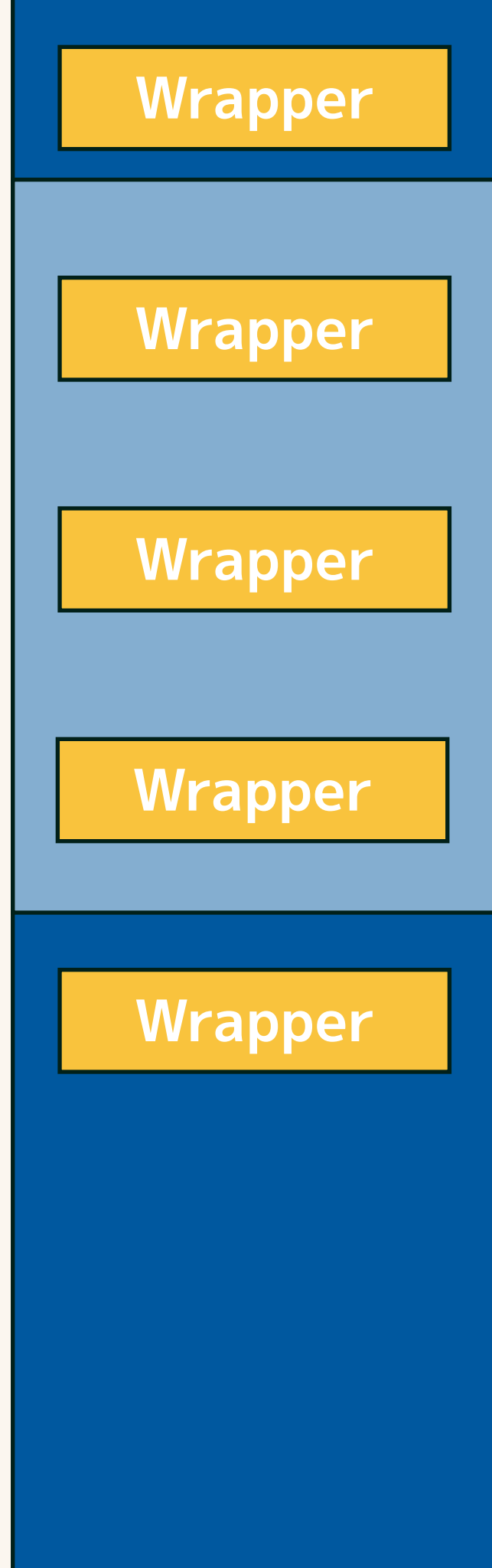
JSON

(API)

HTML

JS Template





DOMContentLoaded



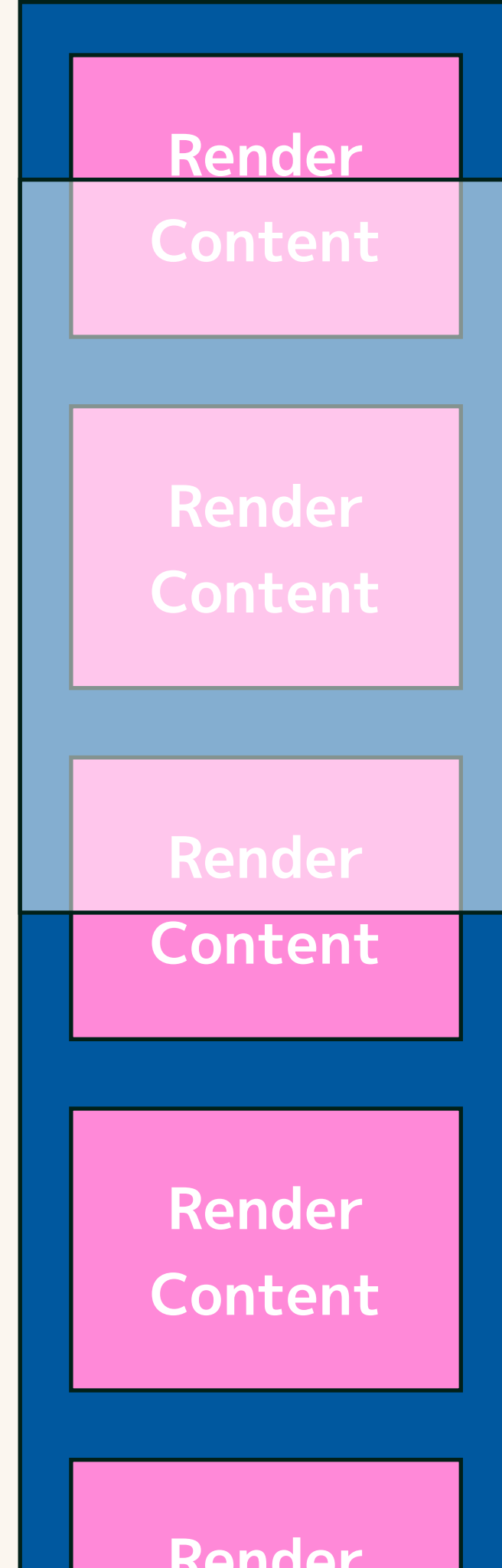
jQuery template



個別に\$.html()



やっぱりガクガク



テンプレートの共通化

jQuery templateのみで描画していた所を
初期表示を安定させるためにPHP化

- ・ 初期表示 → PHP
- ・ 追加表示 → JS 2重管理化...?



無理に共通化しない勇気

サーバーサイド自身が書き換えられる
可能性もあった為、今回は無茶をせず

やるならPHP → JSテンプレートに
変換する補助スクリプトを挟むとか



地味なリアクション大事

ボタンを押したら凹むとか
悪くてもハイライトくらい欲しい

touchstart ~ touchendで
“**is-pressed**”クラスをトグル制御





ボタンのステート

is-pressedのほかにも
is-animatedやis-activeなど
CSSと連携して確実に実装

ボタンがキュンキュン動く

CSS Animationsであれば自然と
合成レイヤーに入るんだけど...

※細々したボタンUIにCreateJSとかを
ぶち込むのはまた別のノウハウ





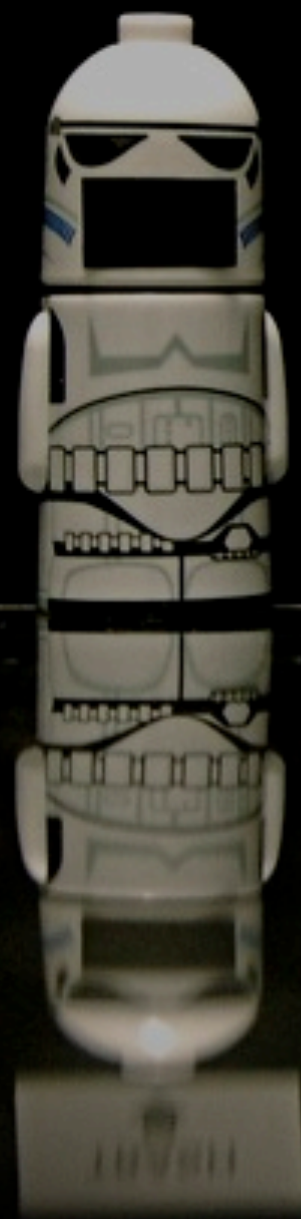
過剰な合成レイヤー

アニメーション適用前から
合成レイヤーに入りまくってた

なにかおまじないの形跡？

<https://speakerdeck.com/ahomu/high-performance-web-frontend>

TRASH
↓



ライブラリの 選定と経緯



Libraries

Selector Based Library

Structuring Library

Utility Belt Library

Templating Library

Dependency Management



Requirements

そのジャンルで**有名**であること
文法や諸機能が**素直**であること
ベンチマークが**劣悪**でないこと



Structuring Library

[jashkenas/backbone](#)

Reinventing the Wheel...('A')



Utility Belt Library

[bestiejs/lodash](#)

[jashkenas/underscore](#)



Selector Based Library

jQuery 2.x

Zepto

Vanilla...[^],_^ ('^')[^],_^



Build option

Zepto 1.0

`MODULES="zepto event ajax form" ./make dist`

jQuery 2.0.1

`grunt custom:-sizzle,-wrap,-event-alias,-effects,-deprecated,+ajax/xhr`



Comparison table (bytes)

	jquery.min.js	zepto.min.js
raw	58,062	23,717
gzip -1	23,734	9,582
gzip -3	22,236	9,155
gzip -6	20,423	8,561



Vanilla JS...?

ベンチマーク ≠ 最適化



Templating Library

wycats/handlebars.js

_.template()

linkedin/dustjs



_.template()

ミニマム構成で _ . から呼べる
シンプルすぎて生々しい
自由度 = 力技の温床になりがち



linkedin/dustjs

ドキュメントは十分ある
ベンチマーク的には良好な成績
拡張性は高いが随所にクセが...



wycats/handlebars.js

ドキュメントは十分ある
ベンチマーク的には及第点
基本がシンプルで拡張性も◎

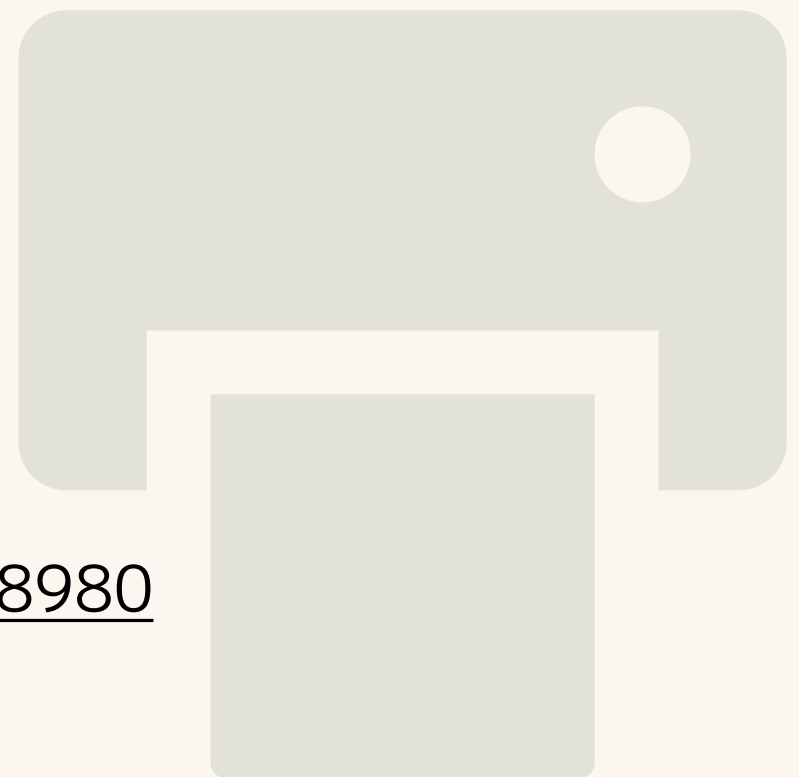
(MeteorやEmber.jsでも利用される高信頼性)



{{#if cond}} がないとか

かなりロジックレスなので
必要があればヘルパを追加する

ex. ifCond <https://gist.github.com/ahomu/6008980>



Dependency Management

RequireJS

Browserify

Namespacing & CONCAT



Dependencies

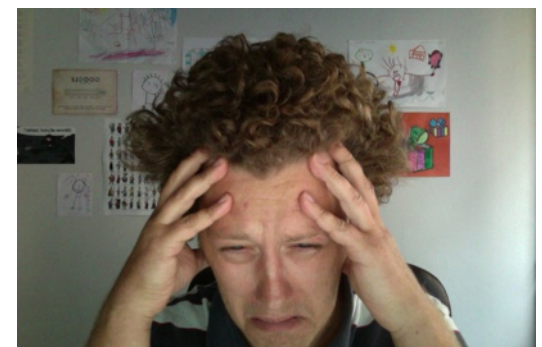
Libs of choice

Plugins

Your Code



via. Dependency Management with RequireJS



RequireJS

非常に優秀なライブラリだが...

Async部分はモバイル微妙

r.js頼みのBIG ONE運用が危うい

規模感的に本当に必要・・・？



Browserify

CommonJS Syntaxを使えるのは魅力
基本的にオールインワンの結合前提
採用するには、ちょっとエッジすぎた

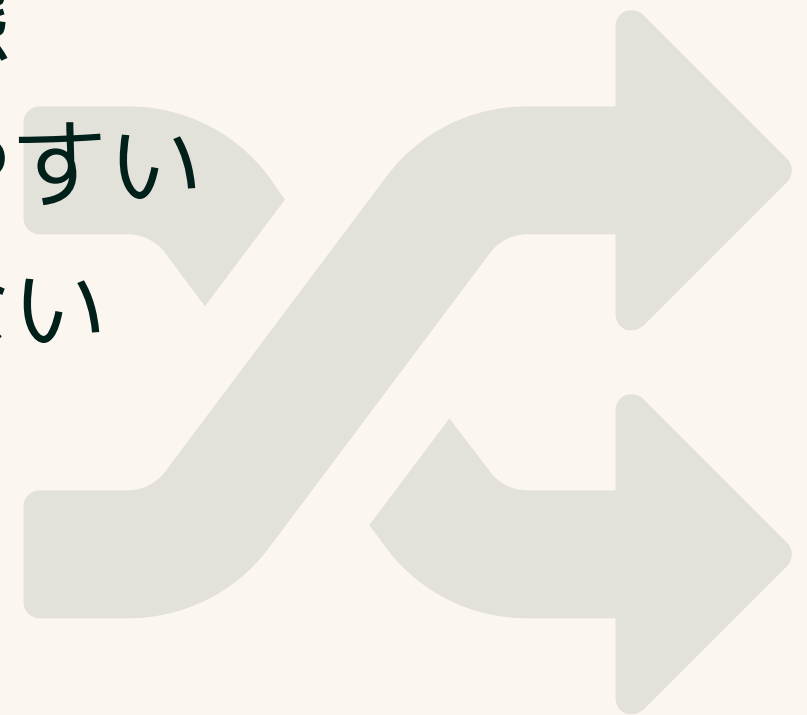


Namespacing & CONCAT

いわゆるシンプルイズベスト状態

肥大化時も適当な粒度で分割しやすい

特定ライブラリの盛衰に依存しない



A row of white, stylized, hand-like sculptures hanging from a wooden ceiling. The sculptures are arranged in a line, each with a unique, organic shape and a small circular hole in the center. They are suspended by thin wires or chains. The background is a dark, textured wooden ceiling with visible beams and a corrugated metal surface below.

Backbone.jsとの 付き合い方

少ない法則で多くの複雑性を解決したい

複雑にしたら複雑にした数だけ
解決できる複雑性が多くなるのは当然
これは作業者の認知資源を犠牲にする

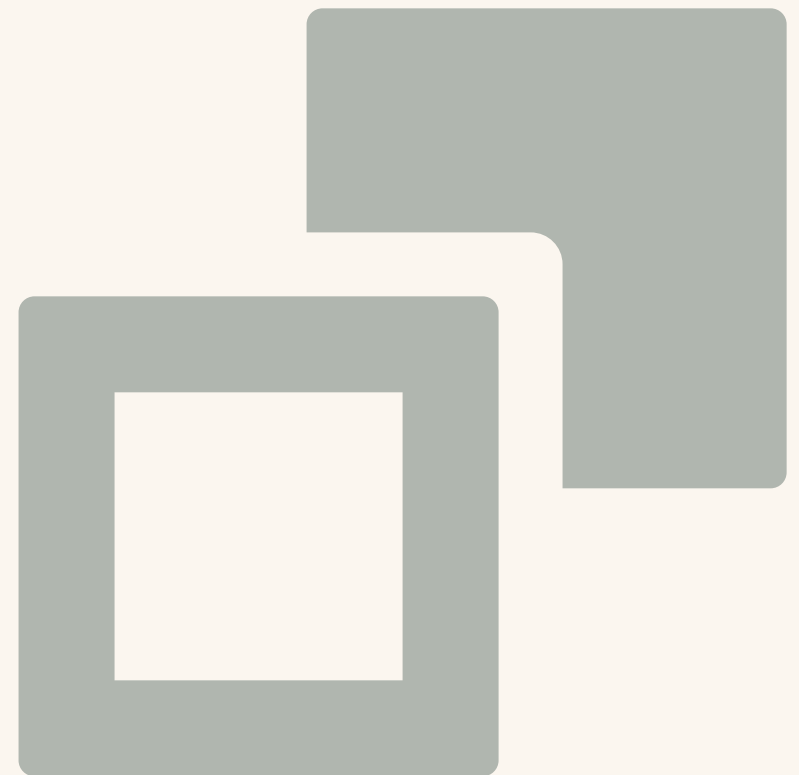
少ない法則で、多くの複雑性を
解決できるようにするのが理想



コピペでラクしたい

どうせ改修は多いしツキもの

「コレをコッチにも」というのは
よくあるコミュニケーションなので
ラクに引き受けられるようにしたい



Viewの整理が一番の関心事

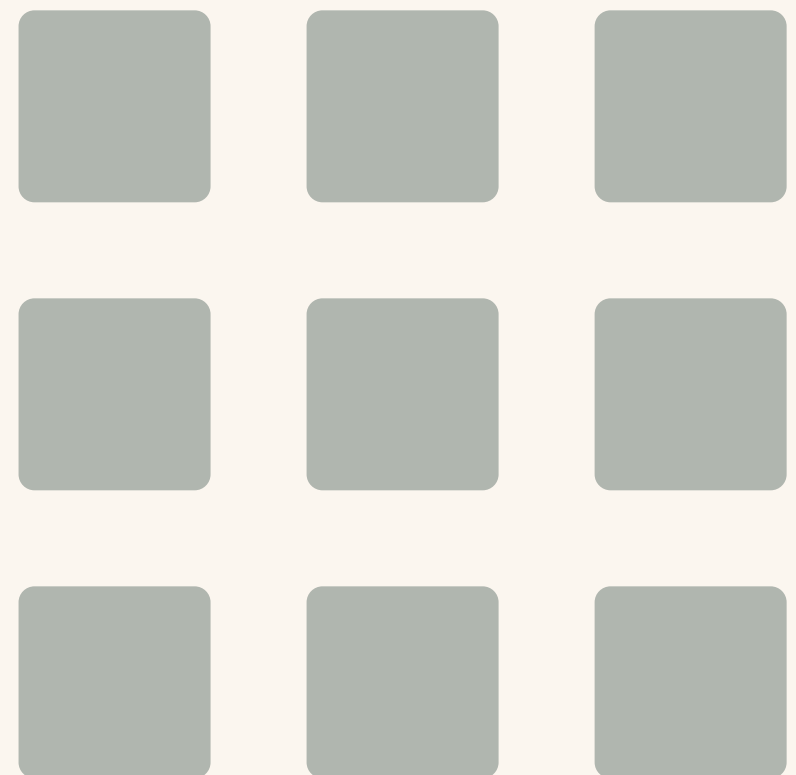
Viewは便利にするべき


→ Viewを強化した薄いライブラリ

Model... Collection...

そこまで重要でもないかも？

→ Collectionの存在を捨てた





Download
1.0.0-rc4

Marionette.js on
GitHub

Marionette

<http://marionettejs.com/>

***Make your Backbone applications
dance!***

薄いライブラリについて

Layout - 枠組み

View - コンテンツパーツ

Component - 独立したアクション



Phalanx

is a Backbone wrapper library. Focusing to decouple view and manage ui component simply.

↓ Download ZIP

↓ API Docs

🐱 View On GitHub

This project is
maintained by [ahomu](#)

Hosted on [GitHub Pages](#)
using the Dinky theme

Getting started

Phalanx is a [Backbone](#) wrapper library. Focusing to decouple view and manage ui component simply.



ahomu/Phalanx

<http://ahomu.github.io/Phalanx/>

via. [Ancient Macedonian battle tactics - Wikipedia, the free encyclopedia](#)

Basic view and component

This is UI component parts.

```
var LikeBtnComponent = Phalanx.Component.extend({  
  // Events are delegated to the View.  
  events: {  
    'click': 'toggleLike'  }  
});
```

Layout.Root

View.Header

Layout.Content

Component.LikeBtn

View.Pictures3col

View.Calendar



Layout (document.body)

HeaderView

ContentView

FooterView

```
var Layout = Phalanx.Layout.extend({  
  regions: {  
    header : '#js-header',  
    content: '#js-content',  
    footer  : '#js-footer'  
  }  
});
```

```
var layout = new Layout({  
  el: 'body'  
});
```

```
layout.assign('header', new HeaderView());  
layout.assign('content', new FirstView());  
layout.assign('footer', new FooterView());
```

```
// change View  
layout.assign('content', new SecondView());
```

View

ろれむいぷさむらりるれろ・ろれむいぷさ
むらりるれろ・ろれむいぷさむらりるれろ

Component



ろれむいぷさむらりるれろ・ろれむいぷさ
むらりるれろ・ろれむいぷさむらりるれろ

Component



ろれむいぷさむらりるれろ・ろれむいぷさ
むらりるれろ・ろれむいぷさむらりるれろ

Component



ろれむいぷさむらりるれろ・ろれむいぷさ
むらりるれろ・ろれむいぷさむらりるれろ

Component



```
var LikeBtn = Phalanx.Component.extend({
  events: {
    'click [data-ui="btn"]': 'onClickBtn'
  },
  onClickBtn: function(evt) {
    var pid = evt.currentTarget.getAttribute('data-pid');
    $.ajax({
      method: 'POST'
      path: 'api/v1/like'
      data: {id: pid},
      success: function(resp) {
        this.trigger('success', resp);
      }.bind(this)
    });
  }
});

var ListView = Phalanx.View.extend({
  components: {
    'likeBtn': LikeBtn
  },
  listeners: {
    'success likeBtn': 'listenerMethod'
  },
  listenerMethod: function() {
    alert('ComponentさんがSuccessしたよ！');
  }
});
```

HTML側の雰囲気 data-*的な宣言が多め

// キュンラッパー

```
<div data-component="likeBtn" data-id="790580" class="ref r" >
```

// キュンカウント

```
<em data-ui="count" class="btn-like-a-count mr-b cd is-liked">43</em>
```

// キュンボタン

```
<span data-ui="likeBtn" class="btn-like btn-like-a" >
```

```
  <i class="heart ir">キュン</i>
```

```
</span>
```

```
</div>
```


DOM Based State

DOM(HTML)がすべてを知っている状態を
前提として、JavaScriptに仕事をさせない

テンプレートのdata属性とかは
多少もっさりするが、分散と割り切る



つまり

—人人人人人人人人人人—

> JavaScriptかきたくない <

—Y^Y^Y^Y^Y^Y^Y^Y^Y^Y^—

今後の課題

今後の課題

1. 運用を経てJSがどうなるか？
2. DOM Basedで成り立つか？
3. ライブラリの標準化？
4. Backbone.jsのプラクティス
5. シングルページアプリ('A';)

Questions?

🏠 <http://aho.mu>

🐦 [@ahomu](https://twitter.com/ahomu)

🐙 github.com/ahomu

