

# HTML5/JSモバイルアプリ 最前線

...

田中正裕  
@massie

HTML5カンファレンス2016  
2016.09.03



田中正裕

アシアル株式会社 代表取締役



React Nativeの場でCordovaを発表したり  
MonacaとかOnsen UIを作っています

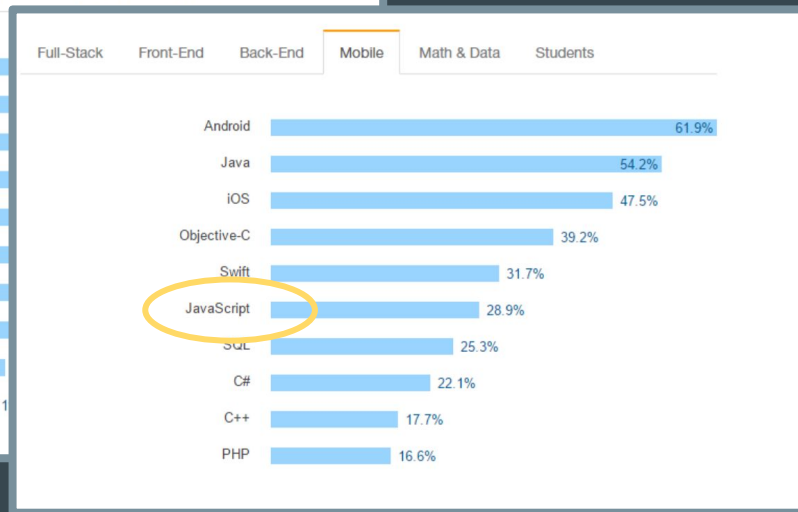
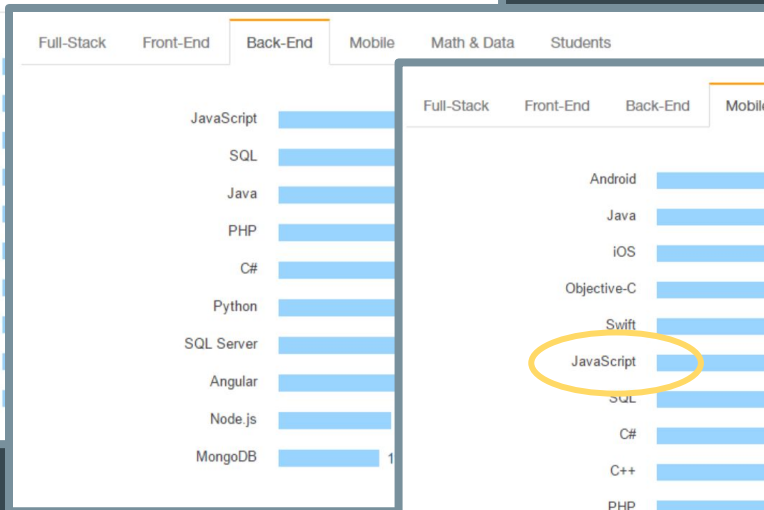
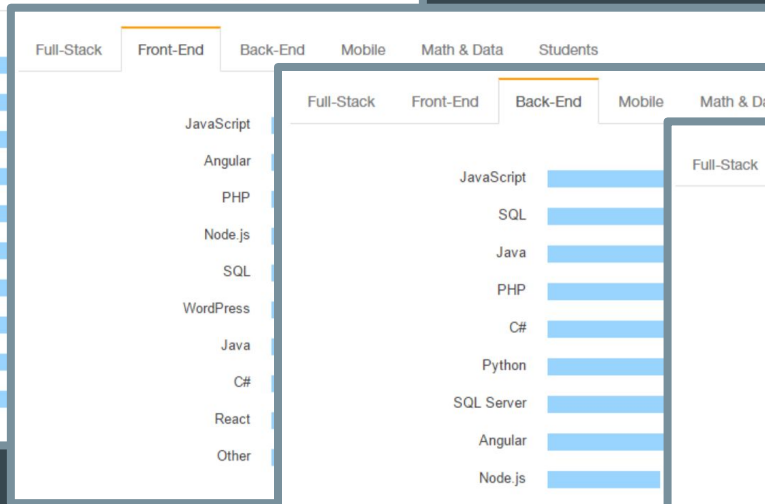
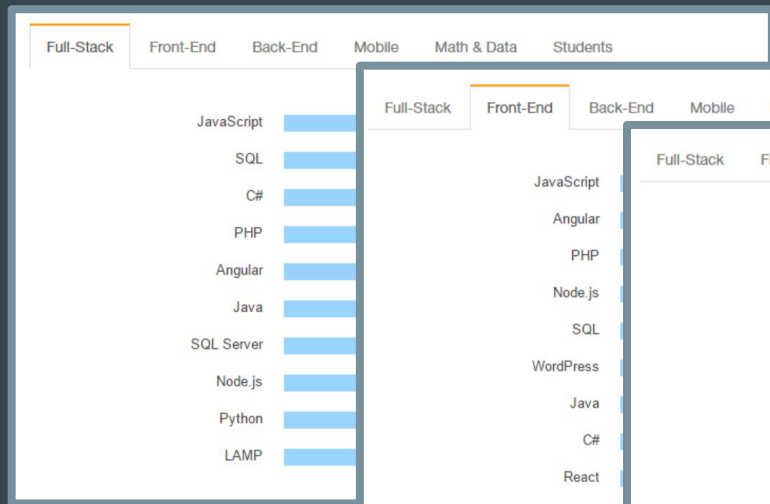
今日のテーマ

“HTML5/JSモバイルアプリ”

やるなら今でしょ！

# iPhone販売から8年 PhoneGap発表から7年が経過し...

モバイル開発の世界では、依然としてJSは遠い存在。



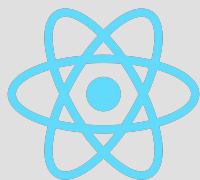
# Why?

どうしてモバイルだけポピュラーじゃない??

色々なソリューションは出てますが...



titanium™



React Native



PhoneGap



APACHE  
CORDOVA™



Monaca



ionic



Onsen UI

# Why?

どうして、こんなにややこしい！！

# 簡単に歴史から

2011年: Titanium vs. PhoneGap

Titanium: masuidriveさん、id:naoya、JustSystem浮川さん、事例にいとまが無い

PhoneGap: 認知度も事例も微妙。ユーザーも伸び悩み。Monaca登場。

2012年: AdobeがPhoneGapを作っていたNitobiを買収し、Cordovaが誕生。Apacheに。

2013年: IonicやOnsen UIが登場。じわじわCordovaが広がっていく。

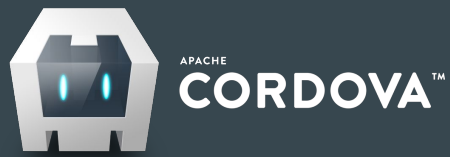
2014年: Electronの前進であるAtom Shellが登場。デスクトップ向け。

2015年: React Native登場。すぐに熱狂的なファンが集まり、一躍メジャープレイヤーに。

今年: Angular 2に対応したNativeScriptも徐々に注目され、より選択肢が増えていく..

※ 非JSのXamarinや、プロダクトではない「Progressive Web Apps」は除いています





# Cordovaについて

# Cordovaとは？

- Web技術とネイティブ技術の合わせ技。
- WebViewを用いてUIを表現する。
- 様々なモバイルOS※に向けて完全なクロスプラットフォームアプリを作れる。

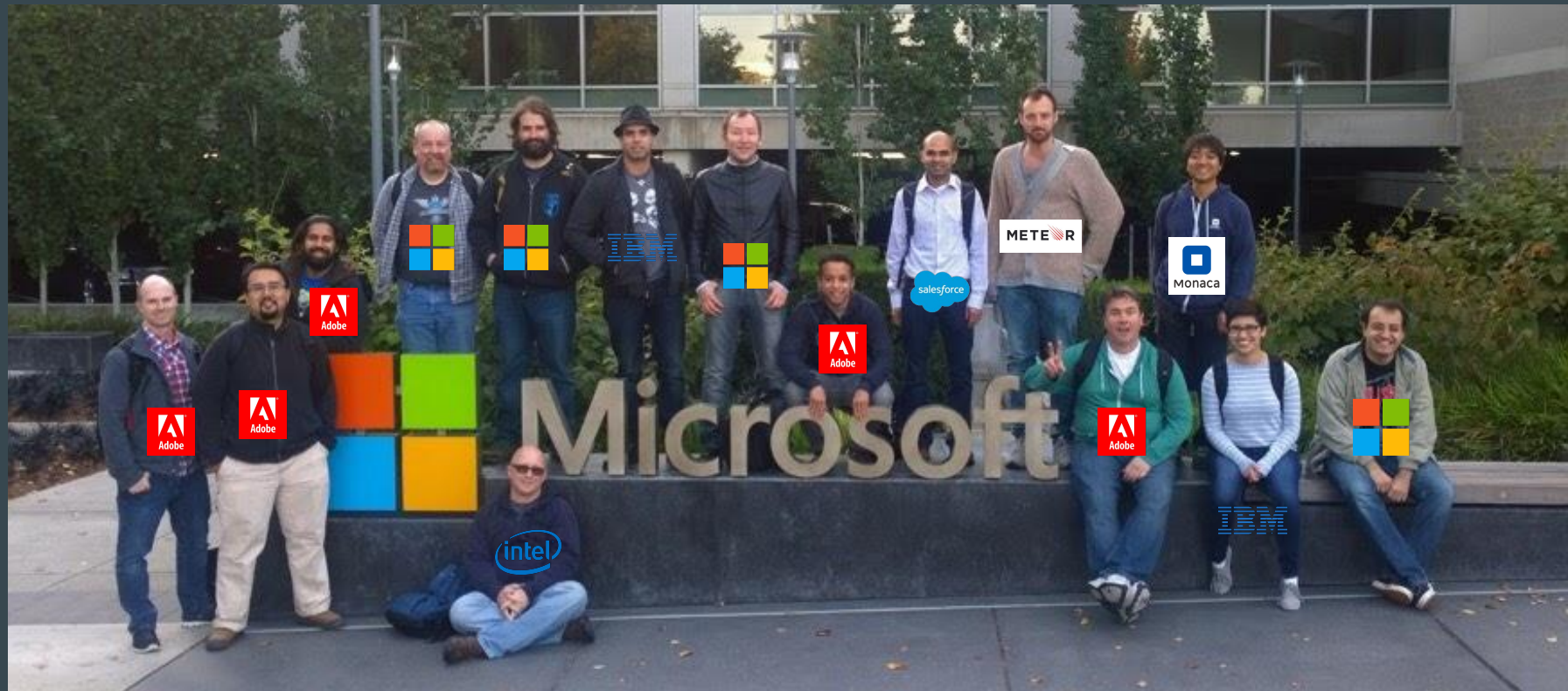
※: **Android, iOS, Windows**, Blackberry, Ubuntu, Firefox OS, Web OS, FireOS

Cordova開発チーム



Microsoft

# Cordovaコミッターの会合@マイクロソフト



# Cordovaの特徴

- WebViewなのでWebの開発そのもの
- 完全なクロスプラットフォーム開発
- 多くのソリューションで採用

# Cordova Tools

A diverse ecosystem of command line tools, JavaScript frameworks, and cloud services exist that augment Cordova



## Adobe PhoneGap

PhoneGap is the original and most popular distribution of Apache Cordova. Turn your HTML, CSS and JavaScript into an app on your device in minutes using our simple desktop and developer apps.



## Ionic

Ionic is a front-end SDK for building cross-platform mobile apps. Built on top of Angular, Ionic also provides a platform for integrating services like push notifications and analytics.



## Monaca

Comprehensive cloud-powered and framework-agnostic set of tools Monaca supports both online and offline development, debugging with live-reload feature and cloud build experience.



## Onsen UI

Custom Elements-based HTML5 framework offers a large selection of components and responsive layout support. Onsen UI lets you create professionally designed multiplatform apps without acquiring additional skillset.



## Visual Studio

Popular IDE for building cross-platform apps for Android, iOS, and Windows. Complete with advanced build and debugging support.



## TACO

Tools for Apache Cordova is a command line utility designed to make building hybrid apps easy and fun. Faster setup. Friendlier command line. Validated quality at run-time.



## Telerik

Telerik Platform is a



## GapDebug

GapDebug is a free



## App Builder

App Builder is a



## Intel XDK

HTML5 IDE with



## Cocoon

Cocoon is a Cordova



## Framework7

Framework7 is a free

課題

パフォーマンス

## 主な理由

- WebViewのレンダリングが遅い
- フレームワークが遅い
- アプリが60fpsに最適化されていない



# WKWebView、Crosswalkへの切り替え

\$ # iOSでWKWebViewエンジンを用いる

\$ cordova plugin add cordova-plugin-wkwebview-engine

\$ # AndroidでBlinkエンジンを用いる

\$ cordova plugin add cordova-plugin-crosswalk-webview

# cordova-plugin-service-worker

The screenshot shows the GitHub repository page for `MobileChromeApps / cordova-plugin-service-worker`. The repository has 18 watchers, 61 stars, and 6 forks. It contains 137 commits, 3 branches, 0 releases, and 6 contributors. The latest commit is by `clelland` on July 23, 2015, with the message "Merge pull request #13 from hgenru/master". The commit history shows updates to `docs`, `src/ios`, `www`, `README.md`, `package.json`, and `plugin.xml`.

**Repository Details:**

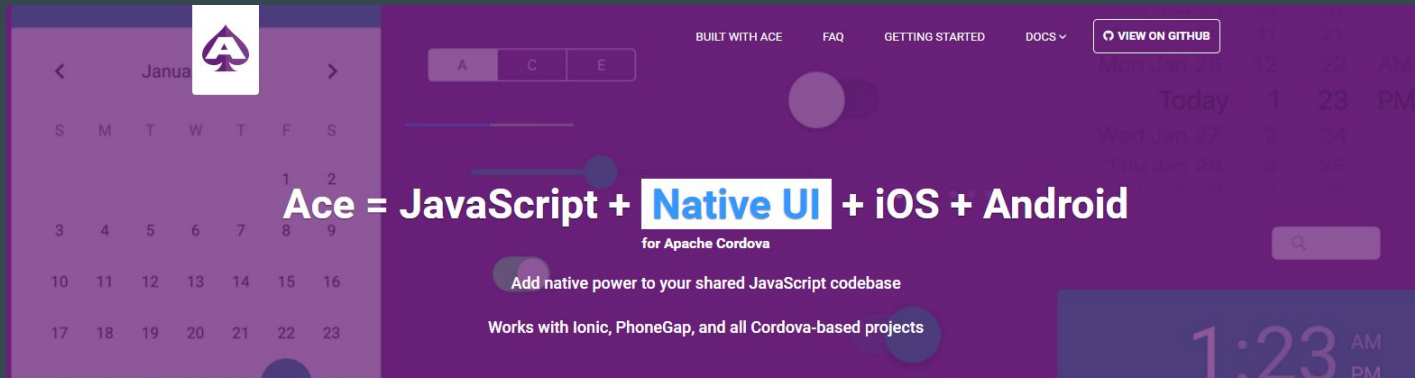
- Repository: MobileChromeApps / cordova-plugin-service-worker
- Watch: 18, Star: 61, Fork: 6
- Code, Issues (10), Pull requests (0), Boards, Burndown, Wiki, Pulse, Graphs
- No description or website provided.
- 137 commits, 3 branches, 0 releases, 6 contributors
- Branch: master, New pull request, Create new file, Upload files, Find file, Clone or download
- Latest commit: 858c90d on Jul 23, 2015

File	Commit Message	Time
docs	Update SWPluginHowTo.md	a year ago
src/ios	Updated logging.	a year ago
www	Changed ev_promises test to use Array.isArray	a year ago
README.md	Updated to version 1.0.1.	a year ago
package.json	Updated to version 1.0.1.	a year ago
plugin.xml	Fix plugin deps	a year ago

iOSでもService Workerが使えるように

# Project ACE

<http://microsoft.github.io/ace/>



 SEE HOW ACE IS USED IN THE PHONEGAP DAY APP

```
$ cordova plugin add cordova-plugin-ace
```

## What can you do with Project Ace?

**1** Mix native UI with HTML

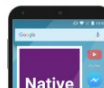
GET THE BEST OF BOTH WORLDS

What's the big deal about native UI?

Docked



Widgets (📱)



ただし、結局はJSフレームワーク自体がボトルネック気味

- jQuery Mobileは反応も動きも余り良くない。
- AngularJSもオーバーヘッドが大きい。

～ようこそ、Virtual DOMの世界へ～

# Angular 2とReact

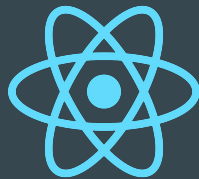
- Virtual DOM
- CSSのカプセル化
- モダンな実装
- Universal JavaScript



- フルスタックフレームワーク
- TypeScriptがほぼ必須



- UI用のライブラリ
- ReduxやMobxなどと組み合わせる



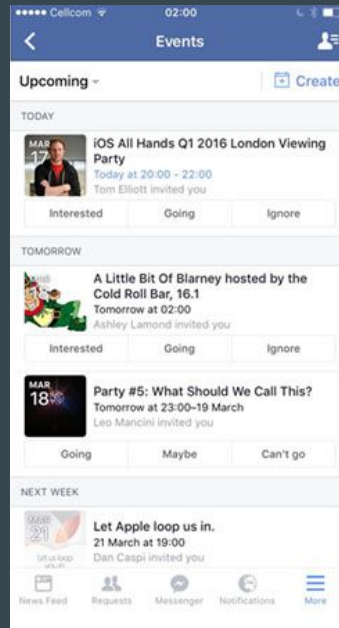
# ReactとReact Native

# Reactとは

Facebookが開発するVirtual DOMフレームワーク

## ReactとReact Native

- ReactJSはJavaScriptライブラリ。フロントエンドとサーバーサイドレンダリングの両方に対応。
- React NativeはReactJSを用いてモバイルアプリを開発するためのフレームワーク。



React Nativeアプリの例  
イベントダッシュボード

# Virtual DOMで画面を組み立てる

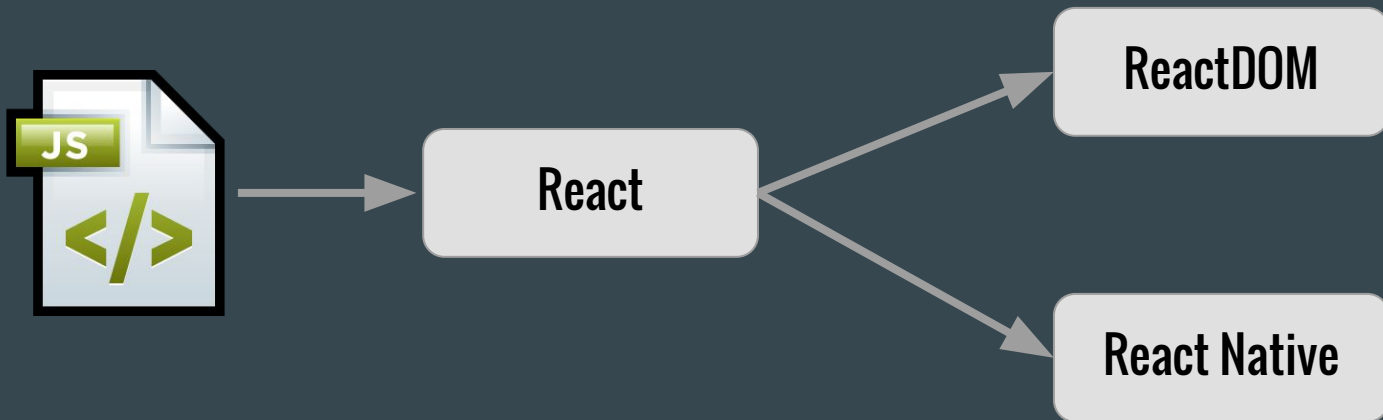
```
var HelloWorld = React.createClass({
  getInitialState: function() {
    return { message: 'Hello World' };
  },
  onClick: function() {
    var messages = ['Hello World!', 'Hello HTML5 Conference', 'Hello React!'];
    var message = messages[Math.floor(Math.random() * 3)];
    this.setState({ message: message });
  },
  render: function() {
    return (
      <div>
        <p>{ this.state.message } />
        <p><input type="button" onClick={ this.onClick } value="ようこそ、Reactの世界へ！"/></p>
      </div>
    );
  }
});
```

JavaScript (JSX) で  
表現されたDOM



# ReactDOMとReact Native

Reactのレンダラーとして、  
ブラウザ(DOM)を使うかReact Nativeを使うかの違い

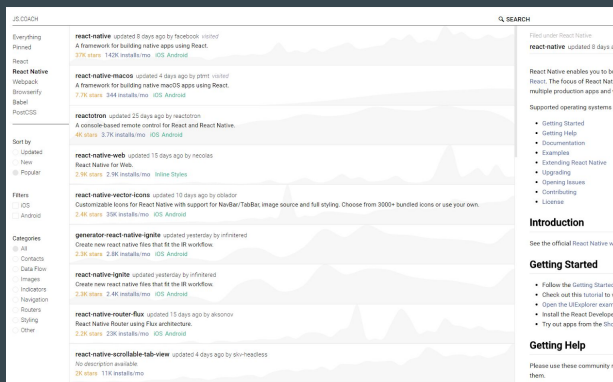


# React Nativeの特徴

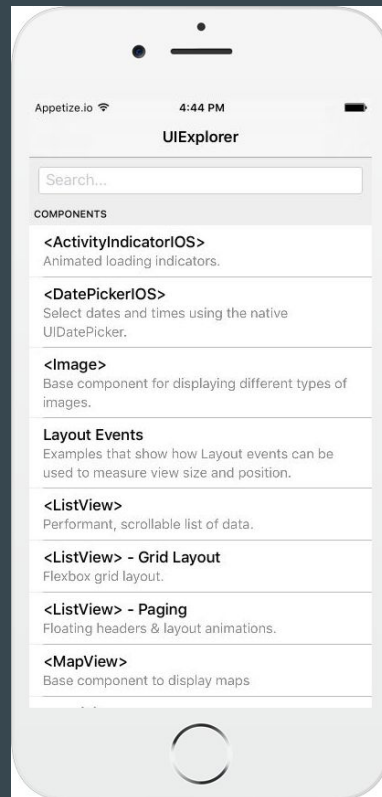
- Learn once, write anywhere.
- ネイティブUIをiOS/Android/Win別に実装
- React.js開発に慣れていれば簡単

# React Nativeのコンポーネント

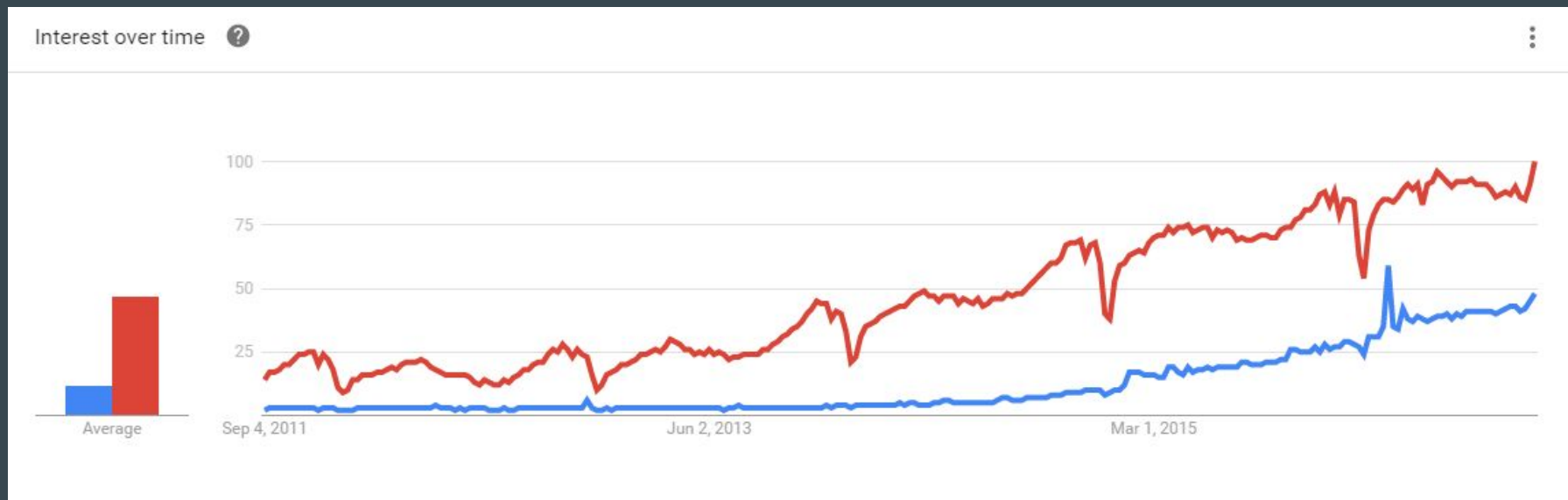
- `<p>`も`<div>`も`<button>`もすべてネイティブUI。
- CSSもパースされネイティブウィジェットで描画。
- 開発者は自分でコンポーネントを開発できるほか、多くのコンポーネントがレジストリに掲載されている。



コンポーネント検索  
<https://JS.coach>



# 急速に発展するコミュニティ



Google Trendsより (Worldwide)



Angular



React

# Reactクローンも多い

## Weex Framework

- React Nativeをフォークしたフレームワーク、Alibaba開発
- Vue 2.0ではネイティブレンダリングとしてサポートされるようです

## Riot.js

- Reactライクなライブラリで、Reactと比べてコードがシンプルに
- ES6で記述できるので取っつきやすい

課題

パフォーマンス

# JavaScriptスレッドの問題

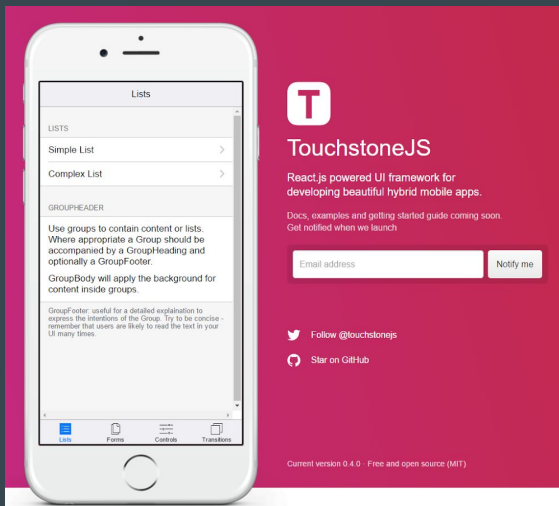
- UIスレッドは60fpsでスムーズに動作
- JSスレッドの処理を16msに抑える

WebView上で動作させるより遙かに高速。

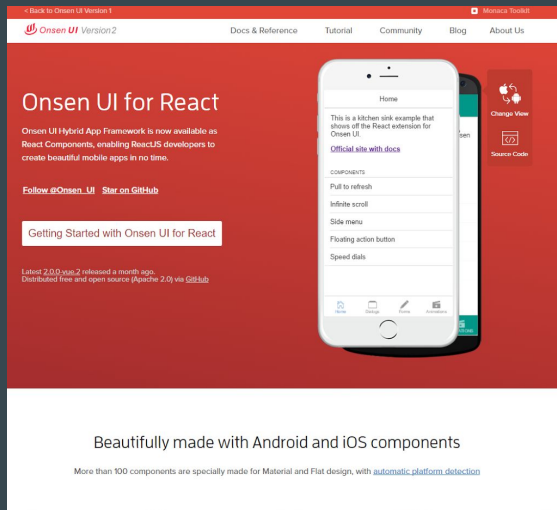
ただ現状、JSはマルチコアを生かし切れない。

# ReactDOM + Cordova

- WebView (HTML + CSS)を使用した、自由なUI実装
- ほかのWebライブラリを組み合わせることができる



<http://touchstonejs.io/>



<https://onsen.io/react/>





# Progressive Web Apps

# Progressive Web Appsの特徴

- Service Workersをネイティブ層として使用
- ブラウザ自体をネイティブアプリに近づける
- iOSに未対応 ([Under Consideration](#))
- 考え方はハイブリッドアプリに近い



PhoneGap is cease to exist, but not yet ?

# 3者の比較

	Cordova	React Native	Progressive Web Apps
開発チーム	Apacheコミュニティ MS、Adobeが中心	Facebook	Google Chrome & Polymer
UIの記述	HTML、CSS	React Component	HTML、CSS
JSフレームワーク	制限なし	React	制限なし
パフォーマンス	△	○	△
デバイス機能	○	○	△
配布手段	アプリストア	アプリストア	Webサイト
対応端末	iOS、Android、Win Univ.	iOS、Android、Win Univ.	Android

共通:どれもツールは充実しており、無料で手軽に始められる。

# 所感

- Cordova: すでに太いレールが敷かれているが、新しさはない
- React Native: 皆がバラバラにレールを敷いているが、刺激的で活発
- Progressive Apps: 難しさがないため広がり期待

実際に試すことで、  
メリット・デメリットが見えてくると思います

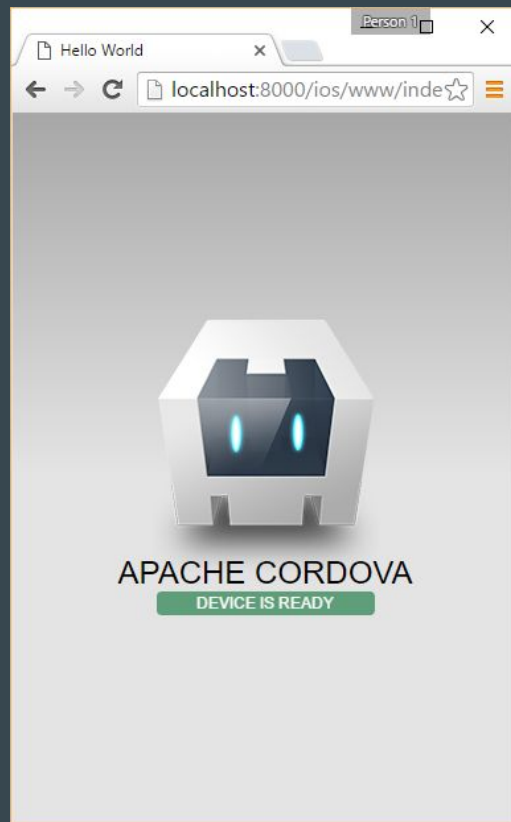
時間があればデモ

# Cordovaことはじめ

```
$ npm -g install cordova  
$ cordova create helloworld  
$ cd helloworld
```

```
$ cordova add platform ios  
$ cordova add platform android
```

```
$ cordova serve # HTTPサーバー経由で配信  
$ cordova build # アプリをビルド
```



# Cordova上でUIフレームワークを使用する

Monaca CLIを通じてOnsen UIを使った例：

```
$ npm install -g monaca
```

```
$ monaca create helloworld
```

```
? Choose a category: (Use arrow keys: ↑ ↓ →)
```

```
> Onsen UI and Angular 2
```

```
Onsen UI and React
```

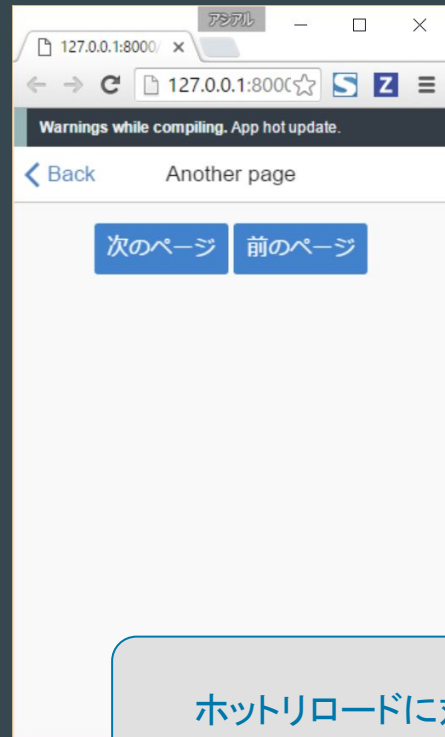
```
Ionic
```

```
Sample Apps
```

```
$ monaca preview # HTTPサーバーで表示
```

```
$ monaca debug # 実機デバッグ
```

```
$ monaca remote build ios # リモートビルド
```



ホットリロードに対応



# React Native編

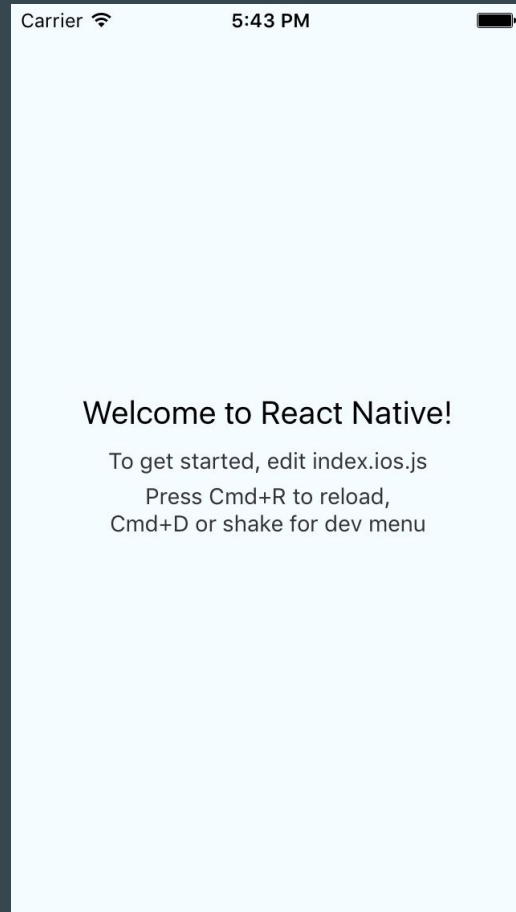
```
$ npm install -g react-native-cli  
$ react-native init helloworld
```

# iOSアプリを実行する

```
$ react-native run-ios
```

# Androidアプリを実行する

```
$ react-native run-android
```



ありがとうございました