

スマホアプリもデスクトップアプリもHTML5で作ろう

2016/3/10

Your Growth is Our Growth.

GMO NIKKO株式会社

デジタルマーケティング部門 アドテクノロジー部
テクニカルコンサルティンググループ
テクノロジーチーム

池尻野 雄介

■ 2016.1 GMO NIKKO株式会社 入社

- 現在はGMO MARSアナリティクスを開発

■ 2007.4 インフォコム株式会社 入社

- 地味なB2Bサービスの企画・開発・マネジメント
- めちゃコミは関わったことないです
- サーバーサイド、フロントエンド、データベース、スマートフォンネイティブアプリ、インフラ構築など一通り経験（自称フルスタック）

使ったことある開発言語

Java、PHP、Objective-C、C++、C#、Ruby

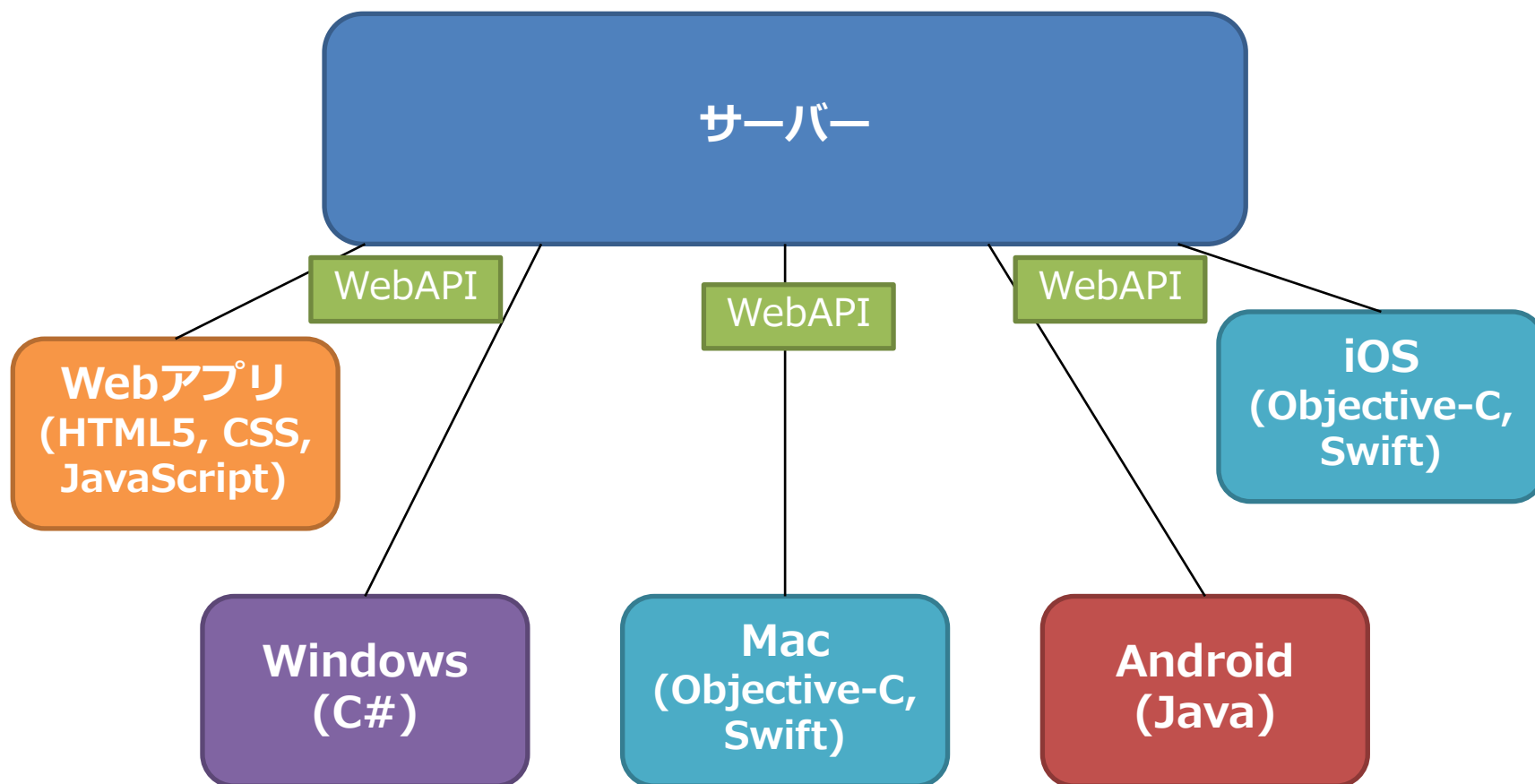
スマホネイティブアプリやデスクトップアプリを
HTML5 + CSS3 + JavaScript (Web技術) で作る



ハイブリッドアプリ

デスクトップアプリ、 スマートフォンアプリ開発で こんな経験ありませんか？

- それぞれのプラットフォーム向けの開発言語が異なるので、学習コストがかかる
- ちょっとした修正も別ソースになるので、メンテナンスの手間が大変
- スモールスタートのビジネスだと、ネイティブで作りこむ予算がない



それぞれのプラットフォーム向けの開発言語で
実装・ビルドしなければならない
→ ハイブリッドアプリによる開発が有効

スマホアプリなら

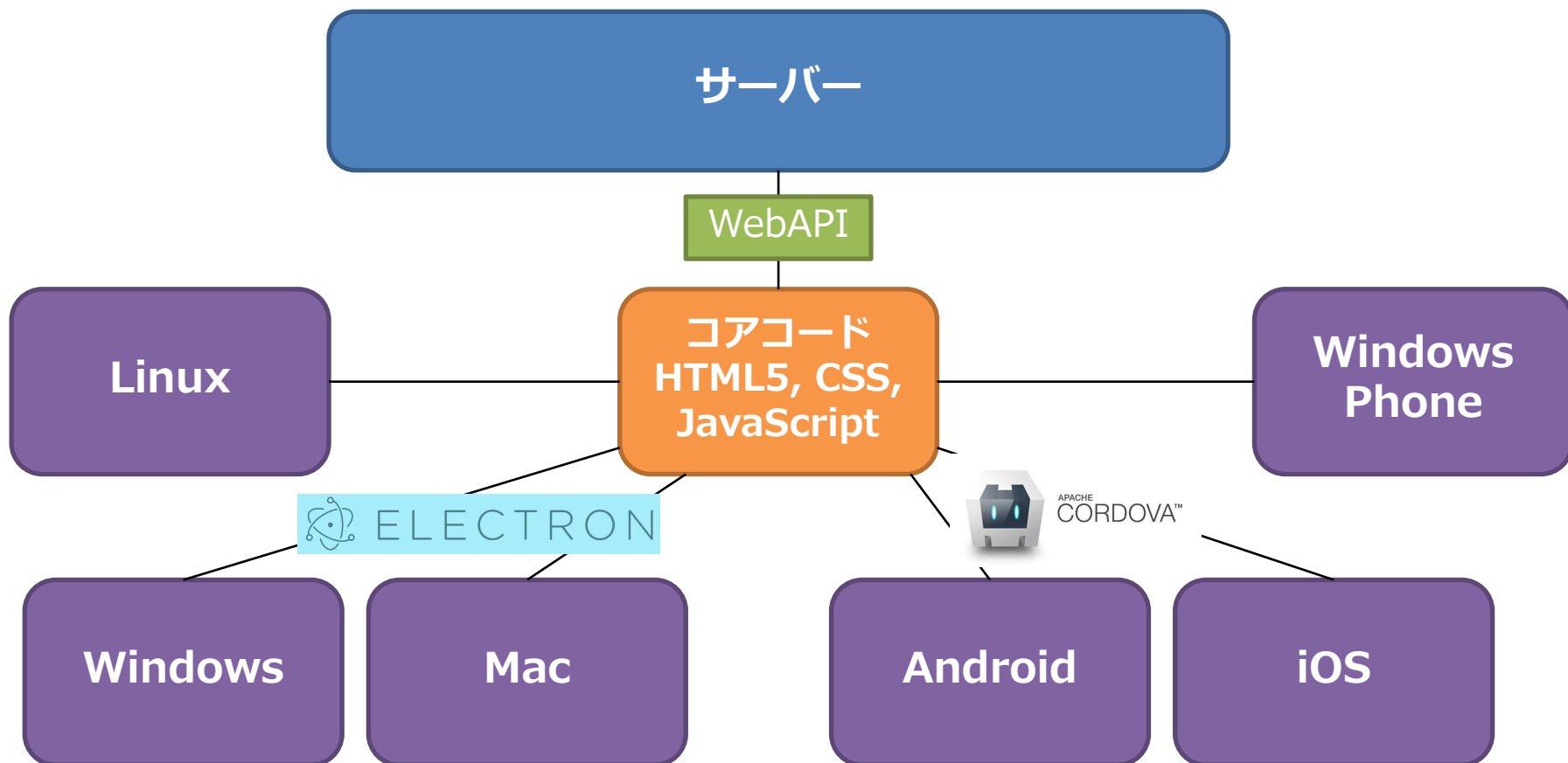


APACHE
CORDOVA™

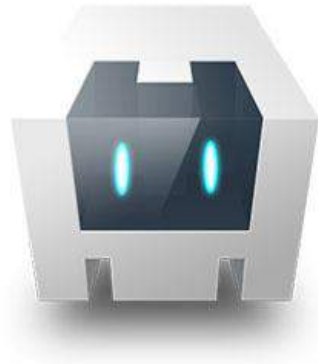
デスクトップアプリなら



ELECTRON



Web技術を使った共通のコードから、各プラットフォーム向けのアプリをビルドできる



APACHE
CORDOVA™

- 元はPhoneGapという名称だったが、Adobeが買収して改名
- iOS, Androidネイティブアプリとしてビルド可能
- Windows8、Windows Phone、Mac OS Xアプリもできる
 - ※Windows7はダメ
 - (後述の) **Electron**使いましょう
- Monacaという有名な開発環境は、Cordovaを内蔵して提供している



APACHE
CORDOVA™

- Webアプリと比較して、古いブラウザの技術的成約から開放
 - WebSocket、WebSQLなどの新しい技術が使える
 - ※ただしiOS、Androidのバージョンには依存します

~~これでもうIE対応で
発狂しなくて済む~~

- ブラウザごとの細かい挙動の違いも吸収
 - Chromeだと動くのに、Firefoxだと動かない、画面がずれる！など
 - Androidは端末標準ブラウザに依存するので注意！
 - ※対処方法あり（後述）

- ハードウェアの機能（カメラ、位置情報、ジャイロセンサーなど）もプラグイン導入でJavaScript上で利用でき、ネイティブで使える機能はほぼ網羅
 - 不足する機能もプラグイン開発でカバーできる
 - ただし、OSのバージョンアップに合わせて、プラグインのバージョンアップは気にしたほうがよい

BCPortal : 会社で使えるTwitter + LINEみたいなスマホアプリ

サーバーサイドはPHPでREST APIを実装
アプリはAngularJS + OnsenUIで構築

災害対策本部
タイムライン

返信を表示 削除 返信

防災 一美 @11 17:49
原宿オフィス7Fの状況です。倒壊や停電はありません。 #原宿

返信を表示 返信

防災 零 @10 17:39
小学校の交差点です。
交通事故が発生していますので、外出は控えたほうがいいかもしれません。 #原宿

返信を表示 返信

防災 四郎 @14 17:38
大阪支社は特に被害はありません。

タイムライン 返信一覧 お知らせ アカウント

災害対策本部
返信を表示

返信

このつぶやきへの返信

削除

防災 一美 @11 17:52
@11 はい、待機可能です。 #原宿

災害対策本部 @operator 17:50
@11 オフィス内での待機は可能でしょうか？ #原宿

タイムライン 返信一覧 お知らせ アカウント

au LTE 20:27

戻る 災害対策本部

防災 三津子 2/3 18:40
帰宅途中で電車の中にいます。運転再開したら連絡します。

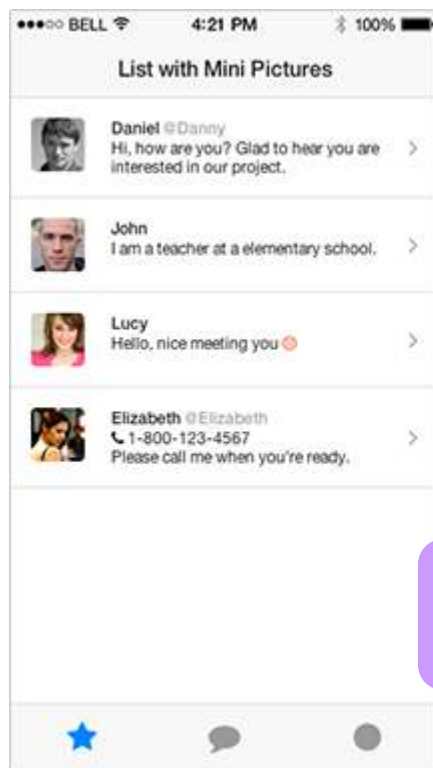
防災 零 2/3 18:41
30分後に到着します。

防災 四郎 2/3 18:44

+ コメントを入力 送信

①Androidのパフォーマンスが悪い

jQuery Mobileはパフォーマンスが悪く、チューニングが大変
Onsen UI + AngularJSを使うのがオススメ



```
<ons-navigator page="list.html"></ons-navigator>
```

```
<ons-toolbar>  
  <div class="center">List with Mini Pictures</div>  
</ons-toolbar>
```

```
<ons-list-item modifier="chevron">  
  ...  
</ons-list-item>
```

iOSのフラットデザインだけでなく、新バージョンではAndroidのマテリアルデザインも使える

```
<ons-tabbar>...</ons-tabbar>
```

②AndroidのWebViewが端末ブラウザに依存する

Xperiaだと動くのに、Galaxyだと動かない…なんてことも
これではWebアプリのブラウザ依存と変わらない

Crosswalkプラグインを導入しよう！



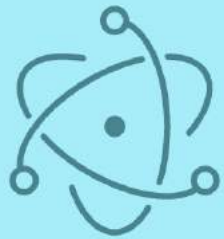
Webレンダリングを
Chromiumエンジン
にしてくれる



通常版

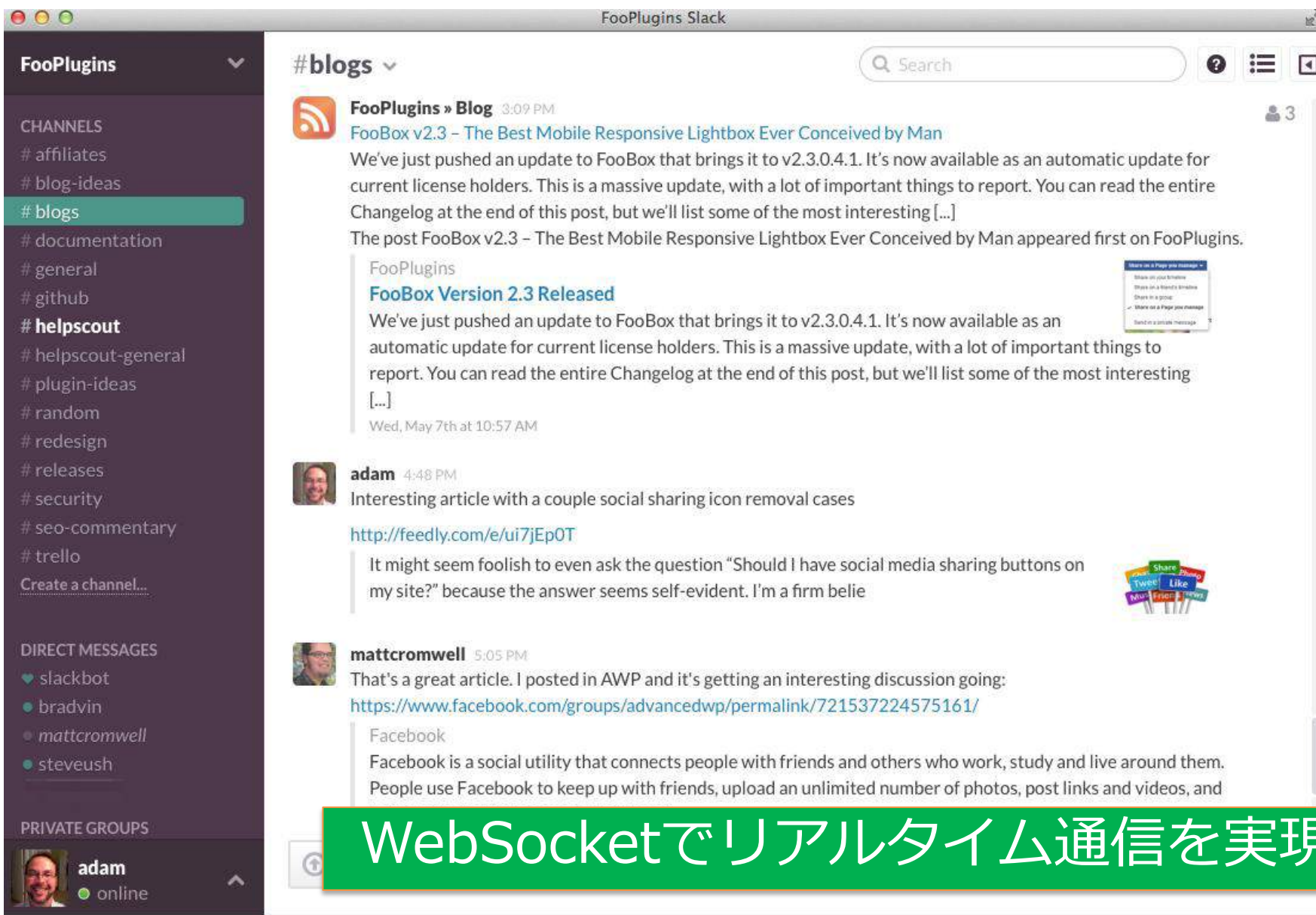


Crosswalk版



ELECTRON

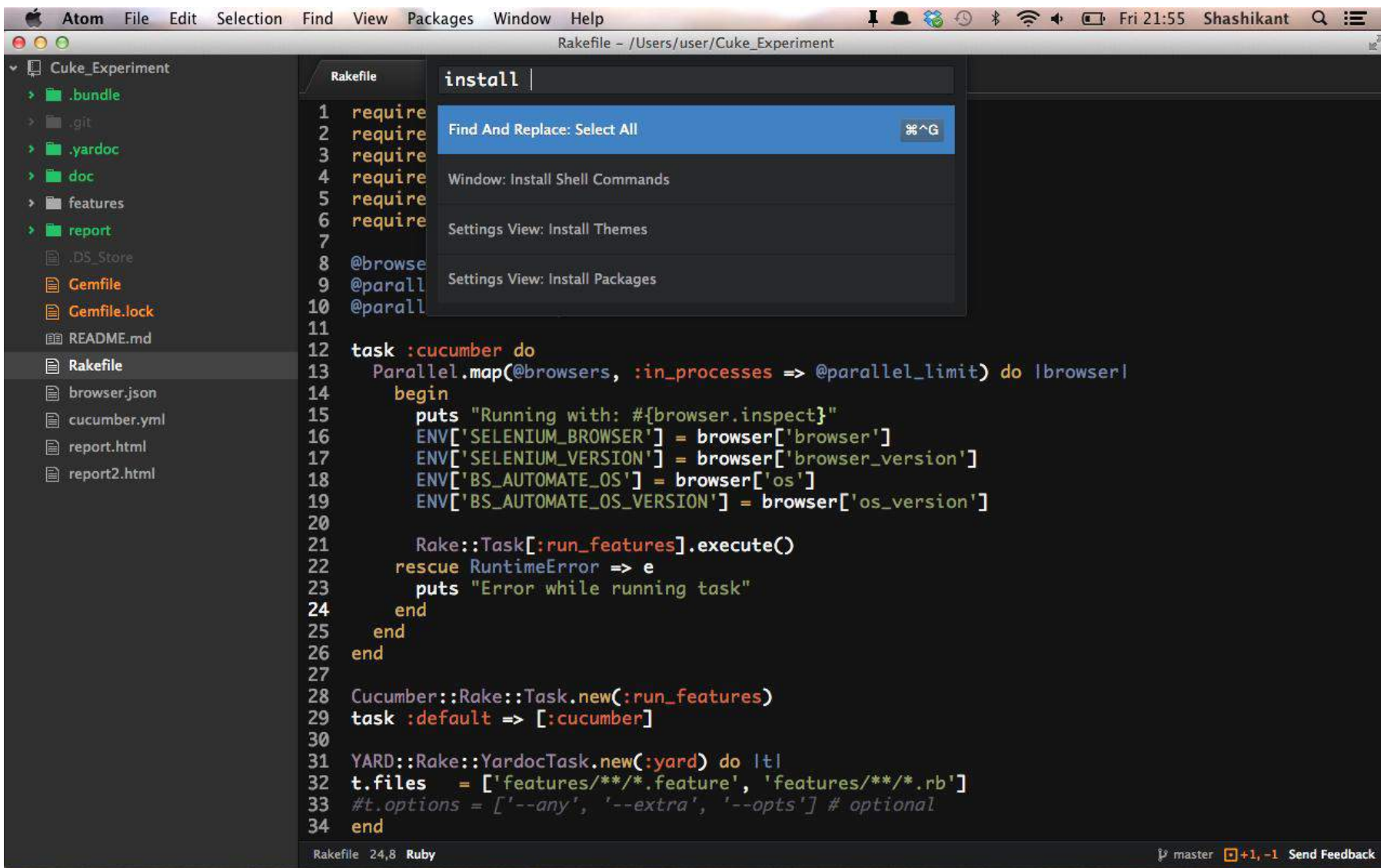
Slackデスクトップアプリ



The screenshot shows the Slack desktop application window titled "FooPlugins Slack". The left sidebar displays the channel list for "FooPlugins", with "# blogs" selected. The main content area shows a message from "FooPlugins » Blog" at 3:09 PM, titled "FooBox v2.3 - The Best Mobile Responsive Lightbox Ever Conceived by Man". The message text reads: "We've just pushed an update to FooBox that brings it to v2.3.0.4.1. It's now available as an automatic update for current license holders. This is a massive update, with a lot of important things to report. You can read the entire Changelog at the end of this post, but we'll list some of the most interesting [...]" and "The post FooBox v2.3 - The Best Mobile Responsive Lightbox Ever Conceived by Man appeared first on FooPlugins." Below this is a quote from "FooPlugins" dated "Wed, May 7th at 10:57 AM" with the title "FooBox Version 2.3 Released" and text: "We've just pushed an update to FooBox that brings it to v2.3.0.4.1. It's now available as an automatic update for current license holders. This is a massive update, with a lot of important things to report. You can read the entire Changelog at the end of this post, but we'll list some of the most interesting [...]". A second message from "adam" at 4:48 PM says "Interesting article with a couple social sharing icon removal cases" and includes a link to "http://feedly.com/e/ui7jEp0T". A third message from "mattcromwell" at 5:05 PM says "That's a great article. I posted in AWP and it's getting an interesting discussion going:" and includes a link to "https://www.facebook.com/groups/advancedwp/permalink/721537224575161/". A green banner at the bottom of the screenshot contains the text "WebSocketでリアルタイム通信を実現".

WebSocketでリアルタイム通信を実現

Atom: GitHub製のエディター



The screenshot shows the Atom text editor interface. The top menu bar includes 'Atom', 'File', 'Edit', 'Selection', 'Find', 'View', 'Packages', 'Window', and 'Help'. The title bar indicates the current file is 'Rakefile' located at '/Users/user/Cuke_Experiment'. The left sidebar shows a file tree for the 'Cuke_Experiment' directory, with 'Rakefile' selected. The main editor area displays the content of the Rakefile, which is a Ruby script for running Cucumber tests. A search overlay is visible, showing the text 'install' and a 'Find And Replace: Select All' button. The Rakefile code includes tasks for running Cucumber tests and generating YARD documentation.

```
1 require
2 require
3 require
4 require
5 require
6 require
7
8 @browser
9 @parall
10 @parall
11
12 task :cucumber do
13   Parallel.map(@browsers, :in_processes => @parallel_limit) do |browser|
14     begin
15       puts "Running with: #{browser.inspect}"
16       ENV['SELENIUM_BROWSER'] = browser['browser']
17       ENV['SELENIUM_VERSION'] = browser['browser_version']
18       ENV['BS_AUTOMATE_OS'] = browser['os']
19       ENV['BS_AUTOMATE_OS_VERSION'] = browser['os_version']
20
21       Rake::Task[:run_features].execute()
22     rescue RuntimeError => e
23       puts "Error while running task"
24     end
25   end
26 end
27
28 Cucumber::Rake::Task.new(:run_features)
29 task :default => [:cucumber]
30
31 YARD::Rake::YardocTask.new(:yard) do |t|
32   t.files = ['features/**/*.feature', 'features/**/*.rb']
33   #t.options = ['--any', '--extra', '--opts'] # optional
34 end
```

Rakefile 24,8 Ruby

master +1, -1 Send Feedback

Kobito: Qiita用のMarkdownエディター

The screenshot shows a window titled "Markdown記法 表示確認用サンプル" (Markdown notation display confirmation sample). The window is split into two panes. The left pane shows the raw Markdown source code, and the right pane shows the rendered HTML output.

Source Code (Left Pane):

```

Markdown記法 表示確認用サンプル

# 見出し1

## 見出し2

### 見出し3

#### 見出し4

##### 見出し5

##### 見出し6

---

ここは段落 (\<p>) です。♪むかしむかし浦島が、助けた亀に連れられて。って浦島呼び捨てかよ！ さん付けしてやれよ！

タグとして認識されたくない、表示したい文字の直前にエスケープ文字を記述します。

...
\

</p>
...

- **強調 strongです。** __これも強調です。__ <strong>strongタグです。</strong>
- *斜体 italicです。* _これも斜体です。_ <i>iタグです。</i>
- ***強調斜体です。*** __強調斜体です。__ <em>emタグです。</em>
- ~~対応に差があるみただけで取り消し線。~~ <s>sタグです。</s>

> 引用します

> > 引用のネストです

> 上に一行空けないとネストのままです

---


```

Rendered Output (Right Pane):

The rendered output shows the visual result of the source code. It features a header "Markdown記法 表示確認用サンプル" followed by six levels of heading tags (h1 to h6). Below the headings is a paragraph of text, followed by a code block containing the escaped paragraph tag. A list of examples follows, demonstrating bold, italic, bold-italic, and strikethrough tags. The list items are:

- 強調 strongです。これも強調です。strongタグです。
- 斜体 italicです。これも斜体です。iタグです。
- 強調斜体です。強調斜体です。emタグです。
- ~~対応に差があるみただけで取り消し線。~~ sタグです。

Below the list are three examples of block-level tags: a quote block, a nested quote block, and a block with a blank line above it to show nesting behavior. At the bottom, the source code for a horizontal line is shown:

```

水平線 (<hr>) 各種
```



ELECTRON

- GitHubが開発しているデスクトップアプリ用フレームワーク
- Node.js + Chromiumで動作
 - Node.jsライブラリが利用できる
- Windows、Mac、Linux向けアプリを開発できる
 - 32bit版、64bit版の両方をビルド可能

Electronのインストール

```
# 事前にNode.jsをインストールしておく
```

```
$ npm install -g electron
```

```
$ electron -v
```

```
v0.36.10
```

```
# Electron用テンプレートをクローン
```

```
$ git clone https://github.com/atom/electron-quick-start
```

```
# Electronを実行
```

```
$ electron ./electron-quick-start
```

- package.json : アプリのマニフェストファイル

```
{  
  "name": "appname",  
  "version": "1.0.0",  
  "main": "main.js"  
}
```

- main.js : エントリポイント

```
function createWindow () {  
  mainWindow = new BrowserWindow({width: 800, height: 600, 'node-integration': false});  
  
  mainWindow.loadURL('file://' + __dirname + '/index.html');  
  
  mainWindow.on('closed', function() {  
    mainWindow = null;  
  });  
}  
  
app.on('ready', createWindow);
```

- index.html : アプリで表示されるHTML

The screenshot shows a web browser window titled "Hello World!". The page content includes a large heading "Hello World!" and a paragraph: "We are using node 5.1.1, Chromium 47.0.2526.110, and Electron 0.36.10." The browser's developer tools are open, displaying the HTML structure and the CSS styles for the body element.

HTML Structure:

```

<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>Hello World!</h1>
    "
    We are using node "
    <script>document.write(process.versions.node)</script>
    "5.1.1,
    Chromium "
    <script>document.write(process.versions.chrome)</script>
    "47.0.2526.110,
    and Electron "
    <script>document.write(process.versions.electron)</script>
    "0.36.10.
  
```

Styles:

```

element.style {
}
user agent stylesheet
body {
  display: block;
  margin: 8px;
}

```

Box Model Diagram:

The diagram illustrates the box model for the body element. It shows a central content area with dimensions 489 x 76.438. This content area is surrounded by padding (green dashed border), which is further enclosed by a border (orange solid border), and finally a margin (orange dashed border) of 8 pixels. The total width and height of the element including the margin are 497 x 84.438.

これだけでHello Worldを表示
Chromeデバッグツールも使える

Electron-Packagerのインストール

```
$ npm install -g electron-packager
```

Electron-Packagerの実行

```
# electron-packager <sourcedir> <appname> --  
platform=<platform> --arch=<arch> --  
version=<version>
```

```
$ electron-packager ./electron-quick-start/ helloworld --  
platform=darwin,win32 --arch=x64 --version=0.36.1
```

- platform … all, linux, win32, darwin を選択
- arch … all, ia32, x64 を選択
- version … Electronのバージョンを指定



© 2015 Nintendo / INTELLIGENT SYSTEMS



- キャラクターが兵種変更をすることで、成長率が変化する
- 結婚したキャラクターの兵種に変更できる
- 主人公のみ、性格で成長率が変化する



キャラクターの成長率算出が大変すぎる
→ よっしゃツール化したろ

キャラクター×兵種の組合せで成長率を算出

FEif 成長率計算ツール

ファイアーエムブレムif 成長率計算ツール

白夜 暗夜 透魔

ユニット名

長所 短所

クラス	HP	力	魔力	技	速さ	幸運	守備	魔防
ダークプリンス/プリンセス	75	55	35	50	55	35	50	35
白の血族	75	55	35	50	55	35	55	30
歌姫	60	50	25	60	65	45	40	30
侍	70	50	25	55	65	40	40	40
剣聖	70	50	30	55	65	40	40	40
兵法者	80	55	25	50	55	35	50	30
鬼人	80	60	35	40	55	25	60	30
修羅	70	60	40	40	55	25	60	35
鍛冶	80	55	25	55	55	30	55	30

無名エンジニアがちょこっと作っただけで数千DL

新規でアプリ作るなら、
ハイブリッドアプリはいいぞ