

ビルドプロセスの 自動化で 効率アップ

Grunt.jsを利用した自動化

@ahomu

自己紹介

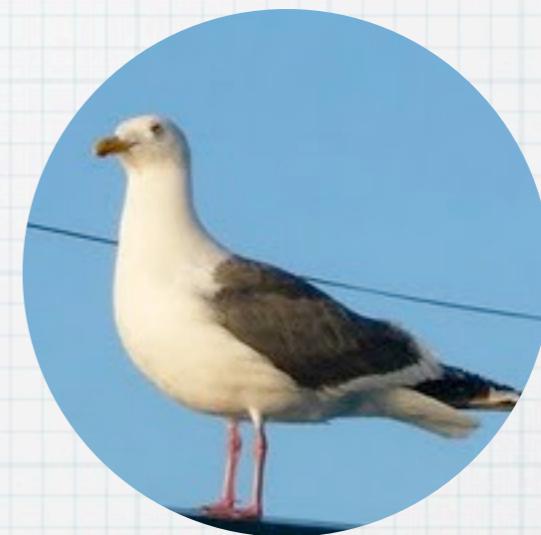
さとうあゆむ (@ahomu)

サイバーエージェント

マークアップエンジニアです

名古屋出身だがね

詳しくは <http://aho.mu>



目次

- ビルドプロセスと黒い画面
- Grunt.js
- 設定ファイルと使い方
- プラグインの紹介と自作
- まとめ

```

mail misc [F] config.h (~Build/dwm) - VIM 00 91% * 00 6% ■ 7% @ / 64% /home 30% /quiver 61% e 0 . . . 0.8K/s 1

[s] */
  const float mfact      = 0.55; /* factor of master area size */
  const Bool resizehints = False; /* True means respect size hints */
  const int nmaster = 2; /* default number of clients in the master stack */
  const int nsessions = 1;
  const Layout layouts[] = {
    { symbol     arrange function },
    {"[ ]=",      tile },      /* first entry is default */
    {"[F]",       NULL },      /* no layout = floating */
    {"[M]",       monocle },
    {"TTT",       bstack },
    {"-|=+",     ntile },
    {"|TE",       nbstack },
  };

  const char *dmenucmd[] = { "dmenu_run", "-fn", font, "-nb" };
  const char *termcmd[] = { "urxvt", NULL };
  const char *xtermcmd[] = { "xterm", NULL };
  const char *conkerorcmd[] = { "conkeror", NULL };
  const char *thunderbirdcmd[] = { "thunderbird", NULL };

  const Key keys[] = {
    { modifier: MODKEY, key: XK_Menu, function: spawn },
    { modifier: MODKEY|ShiftMask, key: XK_Return, function: spawn },
  };

```

...opbox/Archer

/home/jason/Scripts
*.../ 4.0 KB ./.
2009-11-23-204 153.2 KB config/
bigpkg.py 2.9 KB Xdefaults/
colorthemes.sh 797.0 B Xdefaults-gre/
daemons.sh 478.0 B Xdefaults-tran/
dwm-start 260.0 B bashrc
moc-conky.sh 385.0 B config.h
packages.txt 3.1 KB conkerorrc
pacman-up.pl 325.0 B conky-cli
pmupdate-nosiz 381.0 B conky-dzen
pmupdate.sh 617.0 B conkyrc
postsperday.py 983.0 B dwm-start
screen.pl 14.0 KB fstab
timestamp.pl 521.0 B miromiro.vim
ttypster-08 90.3 KB ttypsterrc
ttypster.sh 90.6 KB vimrc
xinitrc
xwrxr-xi

1 Lonely Woman - Ornette Coleman / The Shape Of Things

2 Eventually - Ornette Coleman / The Shape Of Things

3 Peace - Ornette Coleman / The Shape Of Things

4 Focus on Sanity - Ornette Coleman / The Shape Of Things

5 Congeniality - Ornette Coleman / The Shape Of Things

6 Chronology - Ornette Coleman / The Shape Of Things

Playing... PCM 69%
> Lonely Woman - Ornette Coleman / The Shape Of Things
03:07 01:55 [05:02] 44KHz 831Kbps [STEREO]

ビルドプロセスと黒い画面

ビルドプロセス？

さまざまなビルド

ファイルの結合
Concatenate

コード最小化
Minify
Optimize

CSS
プリプロセッサ
Compile

Data URI変換
Inline Embed

画像の
ロスレス圧縮
Compression

Coffee Scriptなど
altJS の Compile

手段・目的は増えるばかり

今日ここまで紹介された内容だけでも、
多くの作業を求められている



**アレもコレも
まとめて自動化したい**

分散した作業と手間をひとまとめにする

なぜ自動化なのか？

繰り返しを減らしたい

- ヒューマンエラーを繰り返したくない
- 手動コンパイルを繰り返したくない
- CSSで同じ記述を繰り返したくない
- HTTPリクエストを繰り返したくない

すべては
繰り返しの最適化

Don't Repeat Yourself

自動化によって 生まれる時間が大切

目の前の作業を最適化することで
本当に必要なことに時間をつかうことができる

なぜ黒い画面なのか？

For Using Valuing Tools

- 価値あるツールも、はじめはCLIツール
- GUIツールになるのは、ほんの一握り
- CLIツールは、一瞬で膨大な仕事をする
- 使いこなせば最も効率化できる手段
- 知っていれば、それだけで武器になる

海の向こうのトレンド感

The screenshot shows a blog post from the website "Adventures in JavaScript Development". The title of the post is "A Baseline for Front-End Developers". The post discusses writing a README for a project and realizing it might intimidate others due to mentions of Node, npm, Homebrew, git, tests, and development builds. It also notes the difficulty of editing files locally in the past. The sidebar features a "Recent Posts" section with links to other articles and a "GitHub Repos" section showing status updating and a GitHub link.

Adventures in JavaScript Development

Blog Archives Search

APR 12TH, 2012 | [COMMENTS](#)

A Baseline for Front-End Developers

I wrote a README the other day for a project that I'm hoping other developers will look at and learn from, and as I was writing it, I realized that it was the sort of thing that might have intimidated the hell out of me a couple of years ago, what with its casual mentions of Node, npm, Homebrew, git, tests, and development and production builds.

Once upon a time, editing files, testing them locally (as best as we could,

Recent Posts

- [A Baseline for Front-End Developers](#)
- [Greenfielding](#)
- [JavaScript: It's a language, not a religion](#)
- [Bocoup](#)
- [Girls and Computers](#)

GitHub Repos

Status updating... [@rmurph on GitHub](#)

The Command Line

コマンドライン利用の必要性が
広く認知され始めている

黒い画面必携時代

が、くるかもしれない

黒い画面はかっこよくなる！

The screenshot shows a terminal window with two panes. The left pane displays a portion of a JavaScript file named 'csslint.js' with syntax highlighting. The right pane shows a file browser listing various files and folders in a directory. The status bar at the bottom provides system information.

```
Anything File Edit Options Buffers Tools IM-JavaScript-IDE JavaScript Help
```

```
1 /*+
2 * grunt-csslint+
3 *+
4 * @author Ayumu Sato+
5 * @package grunt-nbu+
6 */
7 module.exports = function(grunt) {+
8 +
9   'use strict';+
10 +
11   var CSSLint = require('csslint').CSSLint,+
12     util = grunt.util || grunt.utils,+
13     _ = util._;+
14 +
15   var RE_DATA_URI = /(?:data:image\/\w+;base64.\{30\}).*(["]))/+,+
16   COLOR_INDEX = {+
17     'warning': 'yellow',+
18     'error' : 'red'+
19   };+
20 +
21   grunt.registerMultiTask('csslint', 'Check your css strictly', function(\n) {+
22 +
23     var options = this.options ? this.options() : this.data.options || {};  
| {}, // for 0.4.0+
24     srcFiles = grunt.file.expandFiles(this.data.src) || [],+
25     ruleset = CSSLint.getRuleset(),+
26     globalRules = grunt.config('csslintRules') || {},+
27     localRules = options.rules || {},+
28     rc = grunt.file.read('.csslintrc'),+
29     rcRules = {}, argName, parts, val;+
30 +
31     // parse .csslintrc+
32     if (rc) {+
33       rc.split(/\s*\n\r?+/m).forEach(function(line) {+
34         if (line.indexOf('--') === 0) {+
35           argName = line.slice(2);+
36 +
37           if (argName.indexOf('=') > -1){+
38             parts = argName.split('=');+
39             // 0 1 2+

```

```
/Users/ayumusato/Dropbox/Toolkit/grunt-sandbox% (git)-[master]
```

```
filter:
~/Dropbox/Toolkit/grunt-nbu/gruntfiles/tasks
~/Dropbox/Toolkit/grunt-nbu
~/Dropbox/Toolkit/grunt-intro
-
~/Dropbox/Toolkit/grunt-sandbox
~/Dropbox/Toolkit/conf/emacs.d
~/Dropbox/Toolkit/conf
~/Dropbox/Slides/slides/121021-frontend_vol3
~/Dropbox/Toolkit/yeoman-sandbox
~/Dropbox/Toolkit/brunch-sandbox/sample-project
~/Dropbox/Toolkit/brunch-sandbox
~/Desktop
~/Dropbox/Toolkit
~/Dropbox/Sites/ayumusato.com/repo/ablogcms/themes/blog@pretty
~/Dropbox/Sites/jsperf.ahomu.com/hackathon
~/Dropbox/Document/html5_gihyo/distribution
~/Dropbox/Document/html5_gihyo
~/Dropbox/Toolkit/yeoman-sandbox
~/Dropbox/Toolkit/grunt-sandbox/node_modules/grunt-img
```

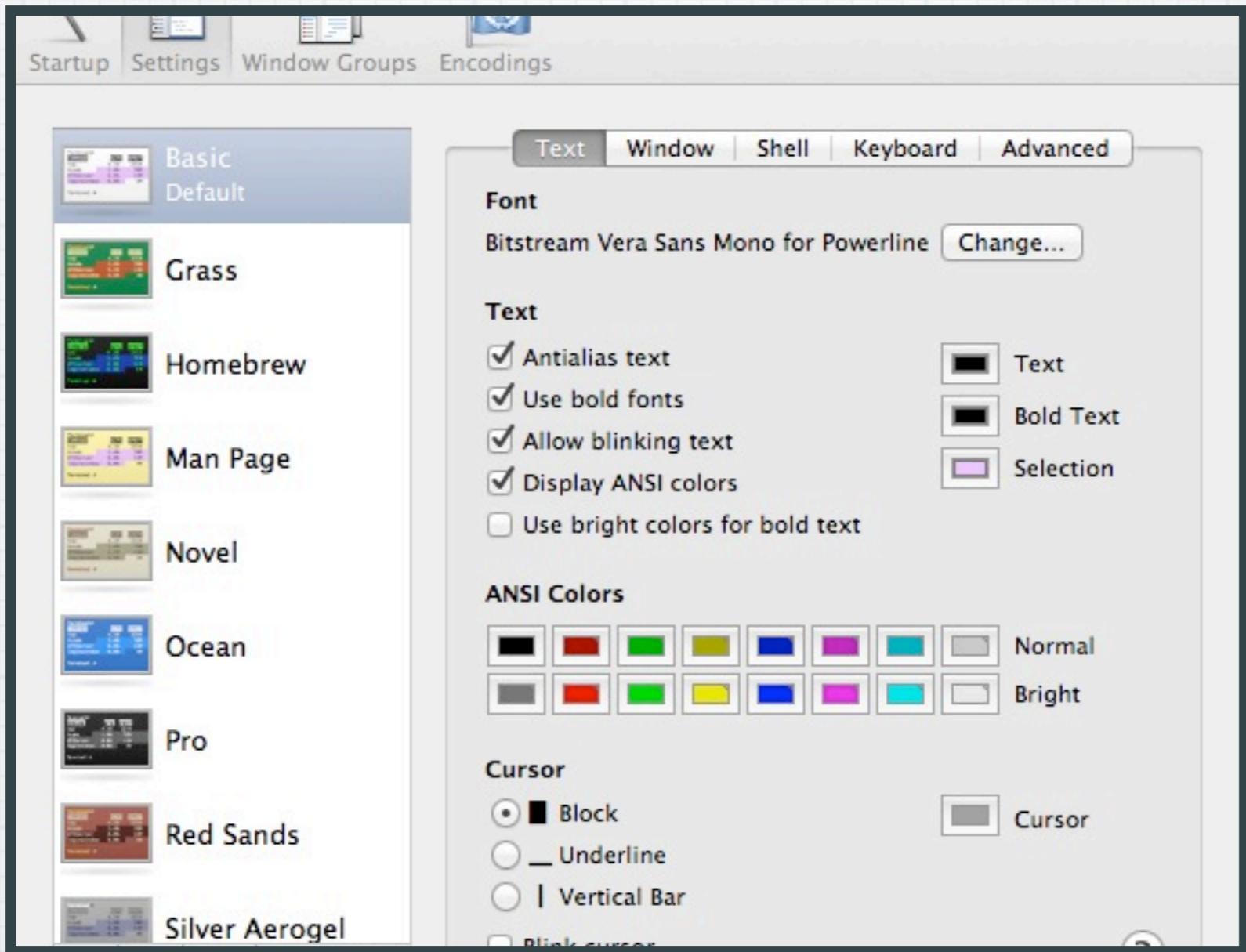
```
[7/29]
```

```
.CFUserTextEncoding .node-gyp Documents
.DS_Store .npm Downloads
.Trash .npmrc Dropbox
.Xauthority .nvm Library
.android .recentf Movies
.bash_history .rnd Music
.bashrc .rvm Pictures
/bower .sbt Public
.chpwd-recent-dirs .scala_history Snapshots
.codeintel .ssh bin
.config .subversion demo
.dropbox .svn lib
.emacs.d .tmux.conf node_modules
.fontconfig .viminfo nvm_bin_alias
.gitconfig .zcompdump tmp
.gitignore .zsh tmux-client-49425.log
.heroku .zsh tmux-client-57041.log
.hgignore_global .zsh_history tmux-server-57043.log
.irb-history .zshrc
```

```
/Users/ayumusato%
```

```
-UU-:***-F1 csslint.js Top L20 (JavaScript-IDE AC) -----
Auto-saving...done
Ayumus-MacBook-Air 6 ayumusato > 1 > emacs
Mem: 71.3% (5.9G/8.16G) < LAvd: 10.58 0.84 1.63 < 2012/11/09(Fri)10:32
```

まずは設定を開くところから



For Windows

- Cygwin
- いきますぐコマンドプロンプトを捨てて、
Cygwinを使うべき 10+の理由
- Getting grunt to work on windows with
cygwin and git



GRUNT

Grunt is a task-based command line build tool for JavaScript projects

[View on GitHub](#)[Documentation](#)

Grunt.js

Getting started

Please be sure to read the getting started guide, which is a complete guide to configuring grunt for your project. In addition, check out the [example gruntfiles](#) which highlight a number of fairly common configurations.

All using **npm install -g grunt**

Latest plugins

preprocess Preprocess files based off environment configuration

rcl Remote Client Logging

contrib-ember Manage and build your Ember.js apps with grunt.js

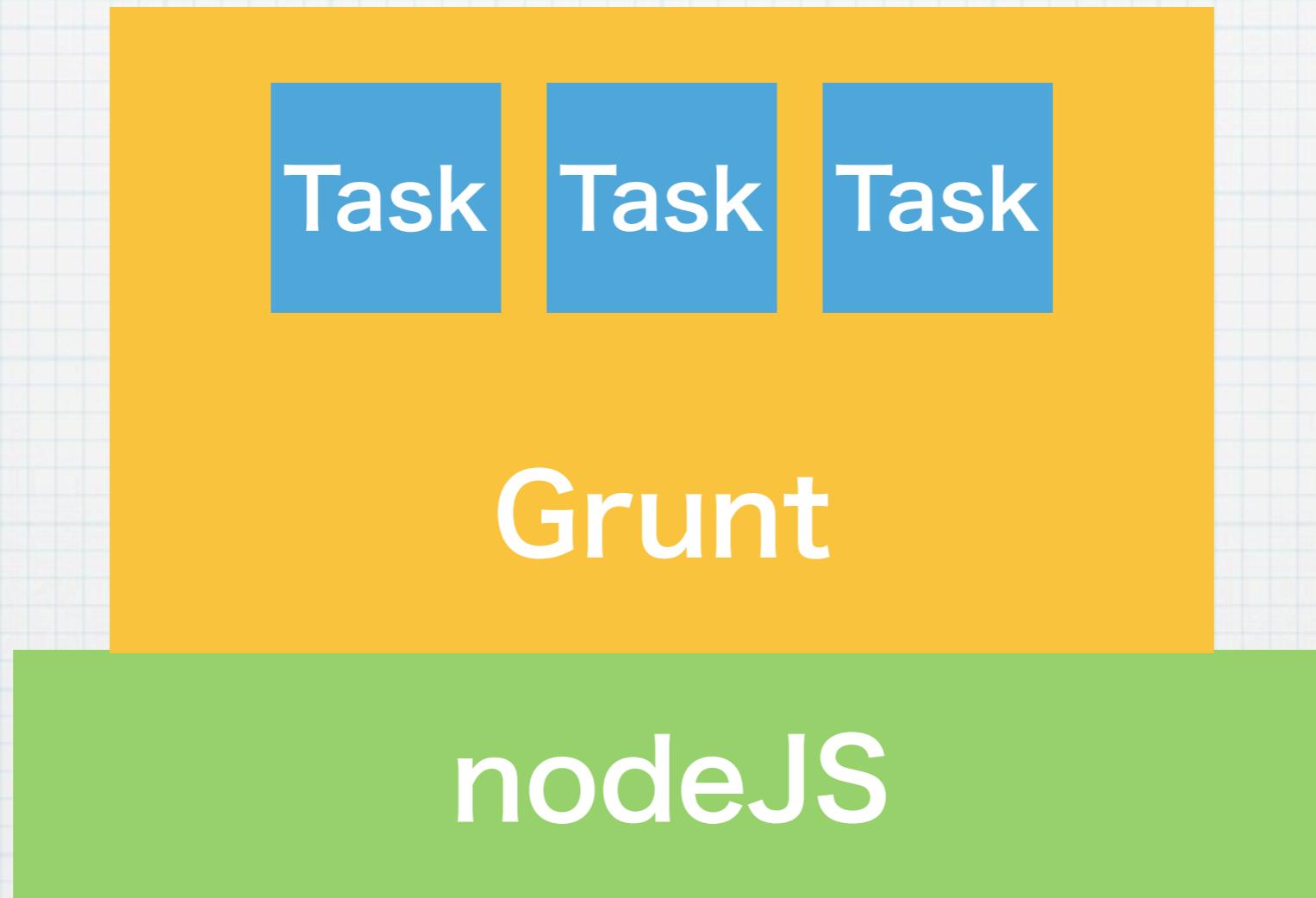
contrib-manifest Precompile Underscore templates to JST file.

pistachio-compiler Grunt task to compile pistachio templates

nodeJSで動くビルドツール

ビルドプロセスの自動化・一元化のほかにも、
ローカル開発環境の構築なども

Task Based Build Tool



Task Based Build Tool

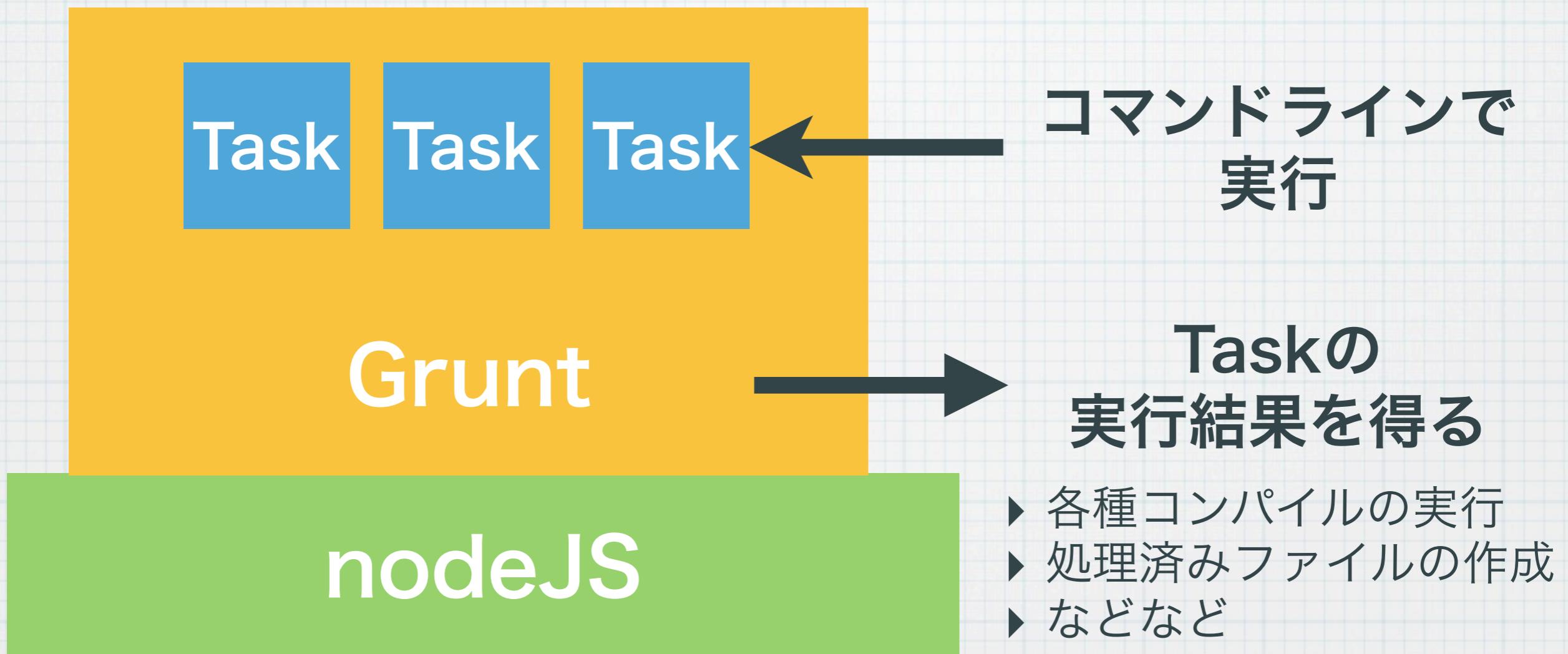
プラグイン部分

Task Task Task

Grunt

nodeJS

Task Based Build Tool



Grunt.jsの特徴

- <http://gruntjs.com>
- 設定もプラグインもすべてJavaScript
- 設定部分のJSは簡潔に記述できる
- プラグインが多数公開されている
- エディタに依存せず利用できる
- 設定ファイルがあれば簡単に環境共有できる

開発アクティビティ

- stable 0.3.17 / develop 0.4.0a
- 非常に盛んに更新されている
- 開発コミュニティが広がっている
- jQuery 1.8のビルドツールとして採用

インストール方法

3ステップで すぐできます

1. nodeJSのインストーラー



2. インストーラーを実行

● Introduction

- License
- Destination Select
- Installation Type
- Installation
- Summary



This package will install node and npm into /usr/local/bin



3. Grunt.jsをインストール

```
npm install -g grunt
```

Getting started

Be sure to read the [getting started guide](#), which is a complete guide to configuring grunt for your project. In addition, check out the [example gruntfiles](#) which highlight a number of fairly common configurations.

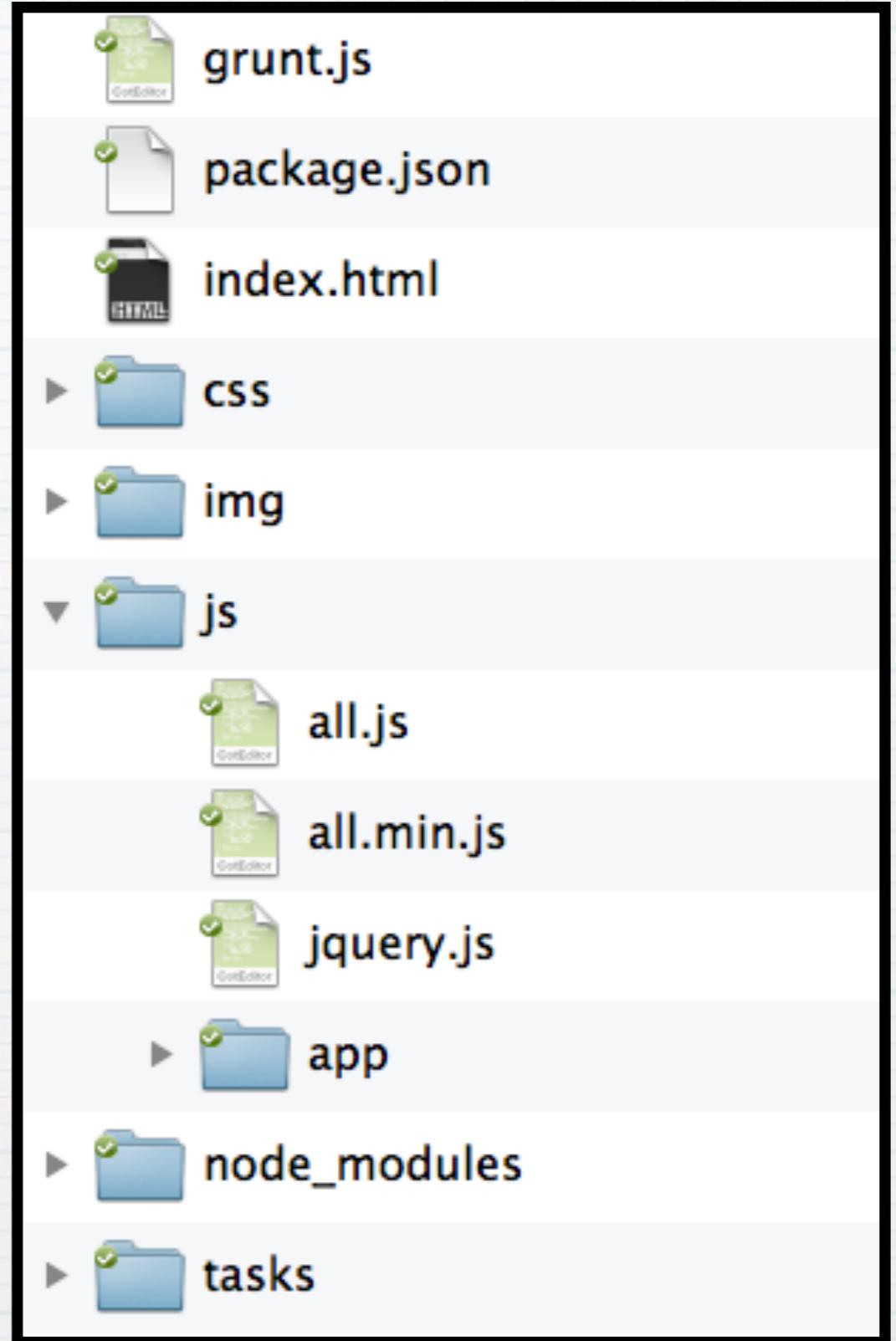
Install using `npm install -g grunt`

これで準備OK

設定ファイルと使い方

作成ファイル

- grunt.js
- package.json



デモファイル公開中

<https://github.com/ahomu/frontend-grunt-demo>

grunt.js

メインの設定ファイル

ファイルの結合

JSとCSSの結合設定

```
grunt.initConfig({  
  concat: {  
    js: {  
      src: [  
        'js/lib/jquery.js',  
        'js/app/main.js',  
        'js/app/sub.js'  
      ],  
      dest: 'js/all.js'  
    },  
    css: {  
      src: ['css/main.css', 'css/sub.css'],  
      dest: 'css/all.css'  
    }  
  },  
})
```

grunt.js

JavaScriptのビルド

JavaScriptのビルド設定

```
module.exports = function(grunt) {  
  grunt.initConfig({  
    concat: {  
      dist: {  
        src: ['js/jquery.js', 'js/app/main.js', 'js/app/sub.js'],  
        dest: 'js/all.js'  
      }  
    },  
    min: {  
      dist: {  
        src: 'js/all.js',  
        dest: 'js/all.min.js'  
      }  
    }  
  });  
  grunt.registerTask('jsbuild', ['concat', 'min']);  
};
```

grunt.js

タスクを複合して指定

```
// Alias Task  
// grunt.registerTask(taskName, taskList)  
  
grunt.registerTask('jsbuild', ['concat', 'min']);
```

grunt.js

concat + min = jsbuild

`grunt jsbuild` の実行結果

```
% grunt jsbuild
# concat タスクが行われて、js/all.js が生成された
Running "concat:dist" (concat) task
File "js/all.js" created.

# さらに min タスクで、圧縮が行われた
Running "min:dist" (min) task
File "js/all.min.js" created.
Uncompressed size: 56545 bytes.
Compressed size:

# おしまい
Done, without errors.
```

黒い画面

ファイルの更新検知

ファイルの更新検知とアクション

```
watch: {  
  css: {  
    // これらのファイルが更新されたら  
    files: ['css/*.less'],  
    // 以下のタスクを実行する  
    tasks: 'less'  
  },  
  js: {  
    files: ['js/app/*.js', 'index.html'],  
    tasks: 'lint concat min'  
  }  
}
```

grunt.js

`grunt watch` の実行結果

```
% grunt watch
```

黒い画面

```
# ファイルの更新監視を開始
```

```
Running "watch" task  
Waiting...0K
```

```
# 検知！
```

```
File "css/main.less" changed.
```

```
# lessタスクが実行されて、コンパイル
```

```
Running "less:dist" (less) task  
File css/main.css created.
```

ローカル開発環境

開発環境として組み合わせた設定

```
grunt.initConfig({  
  /* (前略・これまでのwatchや各タスクの設定) */  
  
  server: {  
    port: 8000, // 静的ローカルサーバを立てるポート  
    base: '.',  
  },  
  reload: {  
    port: 35729 // LiveReloadサーバのポート  
  },  
});  
// develop タスクとして複合  
grunt.registerTask(  
  'develop', ['server', 'reload', 'watch']  
);
```

grunt.js

`grunt watch`の実行結果

```
% grunt develop
```

黒い画面

localhost:8000を起動

```
Running "server" task  
Starting static web server on port 8000.
```

LiveReloadサーバーを起動

```
Running "reload" task  
Proxying http://localhost:8000/  
reload server running at http://localhost:35729
```

ファイルの更新を開始

```
Running "watch" task  
Waiting..
```

プラグインの導入

npmを利用してインストールする

npmって？

Node Package Manager

- nodeで動くパッケージが多数登録されている
- npmコマンドを通して利用できる
- Rubyでいうgemとかのイメージ

202 の プラグイン

2012/11/07 現在 npm において
`gruntplugin` タグがついたパッケージの数

プラグインの利用方法

これも3ステップで すぐできる

gruntjs.comで検索

キーワードで絞り込んで探す	
Name	Description
crusher	A grunt plugin that will straight-up crush some images.
data-uri	Convert to data-uri from image path.
image-embed	Grunt task for embedding images as base64 data URIs inside your stylesheets.
image-normalize	Grunt task to normalize images into a preset size
imagine	Grunt tasks for optimizing, inlining & spriteing images
imgo	Image optimization for Grunt using imgo.
smushit	A Grunt task to remove unnecessary bytes of PNG and JPG using Yahoo Smushit
winningjs-svg2png	A Grunt build task for converting SVGs to PNGs in the varieties desired by Window

プラグインをインストール

見つけたプラグインをnpmでインストール

黒い画面

```
% npm install grunt-data-uri
```

```
npm http GET https://registry.npmjs.org/grunt-data-uri
npm http 200 https://registry.npmjs.org/grunt-data-uri
npm http GET https://registry.npmjs.org/mime
npm http 200 https://registry.npmjs.org/mime
grunt-data-uri@0.0.1 node_modules/grunt-data-uri
└── mime@1.2.7
```

プラグインをロード

```
// grunt.loadNpmTasks(packageName)  
grunt.loadNpmTasks('grunt-data-uri');
```

grunt.js

```
// そのタスクの設定を書き足せばok  
grunt.initConfig({  
  dataUri: {  
    dist: {  
      src: ['css/raw/*.css'],  
      dest: 'css'  
    }  
  }  
});
```

package.json

環境の共有

■ package.jsonを作成



⬇ package.jsonに保存



➡ 設定ファイルを共有する



⟳ package.jsonから復元

package.jsonを作成

// こんなJSONをコピペしてファイル作成

```
{  
  "name": "ABCプロジェクト",  
  "description": "プロジェクト開発環境の定義です",  
  "version": "1.0.0",  
  "author": "才場 太郎",  
  "main": "grunt.js",  
  "dependencies": {},  
  "optionalDependencies": {  
    "grunt": "~0.3.16"  
  }  
}
```

package.json

package.jsonに保存

```
# install --saveで
```

黒い画面

```
# プラグインのインストール時に、package.jsonの依存定義に追加
```

```
% npm install --save grunt-xxxxx
```

```
# remove --saveで
```

```
# アンインストールと同時に、package.jsonの依存定義から削除
```

```
% npm remove --save grunt-xxxxx
```

設定ファイルを共有する

- grunt.js
- package.json
- 2ファイルあればOK 

package.jsonから復元

package.jsonがあるディレクトリで実行すると

黒い画面

記録されているプラグインを、その場に自動で復元(インストール)する

% npm link

ここまでが設定方法

- grunt.js自体はJavaScriptというよりごくごく簡単な設定ファイル
- package.jsonをうまく使うと環境共有がしやすく、ポータビリティも良い
- 次はプラグインの紹介

プラグインの紹介と自作

Meet iPod shuffle. 240 songs. A million different ways. From \$99. PC+Mac. iPod shuffle

プラグインの紹介

大抵はGitHubにリポジトリがあるので
READMEを読めば分かるようになっている

ビルトインタスク

- concat: テキストファイルの結合
- min: JavaScriptのMinify
- server: ローカルサーバー
- watch: ファイルの更新検知
- などなど

grunt-contrib

- <https://github.com/gruntjs/grunt-contrib>
- オフィシャル管理されている詰め合わせ
- 公式メンテナンスされているので参考になる
- bump · clean · coffee · compress
copy · handlebars · jade · jst · less
mincss · stylus · yuidoc

grunt-compass

- <https://github.com/kahlil/grunt-compass>
 - CompassをGruntから利用する
 - CSSプリプロセッサ系は各種ある
- ※ LESS [grunt-contrib-less](#)
- ※ Stylus [grunt-contrib-stylus](#)

grunt-img

- <https://github.com/heldr/grunt-img>
- 画像リソースのロスレス圧縮
- `brew install optipng jpeg` が必要
- 高圧縮設定で ImageOptim と同じくらい
- ahomu/grunt-img (jpeg圧縮率高めた版)

grunt-css

- <https://github.com/t32k/grunt-css>
- CSSファイルのMinify
- 圧縮率に優れたCSSOというライブラリ
- author: @t32k

※ CSSOとgrunt-css ; MOL

grunt-data-uri

- <https://github.com/ahomu/grunt-data-uri>
- CSS内の画像パスをData URIに自動変換
- url(path/to/img.*)を対象にして置換
- author: @ahomu

※ データURIスキーム ; MOL

grunt-reload

- <https://github.com/webxl/grunt-reload>
- ブラウザの自動リロード
- いわゆるLiveReloadの実装
- LiveReloadのChrome拡張と併用できる

※ 前出のDEMOでも利用していたもの

grunt-ftp-deploy

- <https://github.com/zonak/grunt-ftp-deploy>
- FTPを利用したファイルのやり取り
- 簡易なデプロイプロセスになるプラグイン

※ Heroku [grunt-heroku-deploy](#)

※ Amazon S3 [grunt-s3](#)

grunt-templater

- <https://github.com/rockwood/grunt-templater>
- JavaScriptテンプレートをコンパイル
- ejs・mustache・underscoreなど
- 計17種類のテンプレートエンジンに対応

※ コンパイル = 関数オブジェクトへの変換

grunt-requirejs

- RequireJS Optimizer (r.js) に対応
- AMDモジュールを解決して最適化する
- almondというライブラリにも対応して優秀

※ Require.js

※ RequireJSによるコードの分割と最適化

grunt-mocha

- <https://github.com/kmiyashiro/grunt-mocha>
- MochaのHeadlessテスト実行
- PhantomJSに依存している

※ もちろんQUnitやJasmineのプラグインも

※ テスト系はwatchと組み合わせて自動化

プラグインの自作

原則、よく似たプラグインを改造するのがラク
nodeJSを書くことになります

自作プラグインの設定とロード

```
grunt.initConfig({  
    sample: {  
        dist: {  
            src: ['js/**/*.js'], // 入力元  
            dest: 'js/all.js', // 出力先  
            option: { // オプションなど（任意）  
                hoge: 'fuga',  
                peke: 'piyo'  
            }  
        }  
    }  
});  
// grunt.jsファイル内に記述  
// tasksディレクトリの中をすべて読み込み  
grunt.loadTasks('tasks');
```

grunt.js

tasks/sample.js

```
module.exports = function(grunt) {  
  grunt.registerMultiTask('sample', 'own task', function() {  
    var option = this.data.option,  
        src     = this.file.src,  
        dest    = this.file.dest,  
  
    // 非同期処理が必要なときに使用 doneをcallbackにする  
    // var done    = this.async();  
  
    // path/to/**/*.js を解決して該当ファイルを配列で返す  
    srcFiles = grunt.file.expandFiles(src);  
  
    /*...タスクの処理...*/  
  });  
};
```

sample.js

参考素材

- デモサンプル: [tasks/sample.js](#)
- 便利ヘルパー集: [grunt-lib-contrib](#)
- 模範コード例: [grunt-contrib-coffee](#)
- [docs/plugins.md](#)
- [docs/api_task.md](#)
- [docs/api_file.md](#)

補足 0.4.0a について

- 多くのAPIが変更される
- プラグインのメンテナンス状況に注意が必要
- 設定ファイル名が **Gruntfile.js** に変更
- CoffeeScriptで設定やタスクを書けるように

※ 詳しく [grunt/docs/migration_guide.md](#)



まとめ

ビルドの自動化

- Gruntはビルドツールとして便利！
- npmには他のひとが作った、
自動化のためのプラグインが沢山あります！
- 快適に開発できる環境がダイジ！
- 黒い画面と仲良くしましょう！
- 何より...

実践することが大事

ツールは真にこだわるべき問題ではない
無理なく目の前の繰り返しを減らそう

LiveReload

LiveReload 2 proudly presents...

The Web Developer Wonderland

(a happy land where browsers don't need a Refresh button)

CSS edits and image changes apply live.
CoffeeScript, SASS, LESS and others just work.

Citizenship is granted through the Mac App Store.

Windows permanent residency issues are being worked out,
temporary stay already allowed.



CodeKit



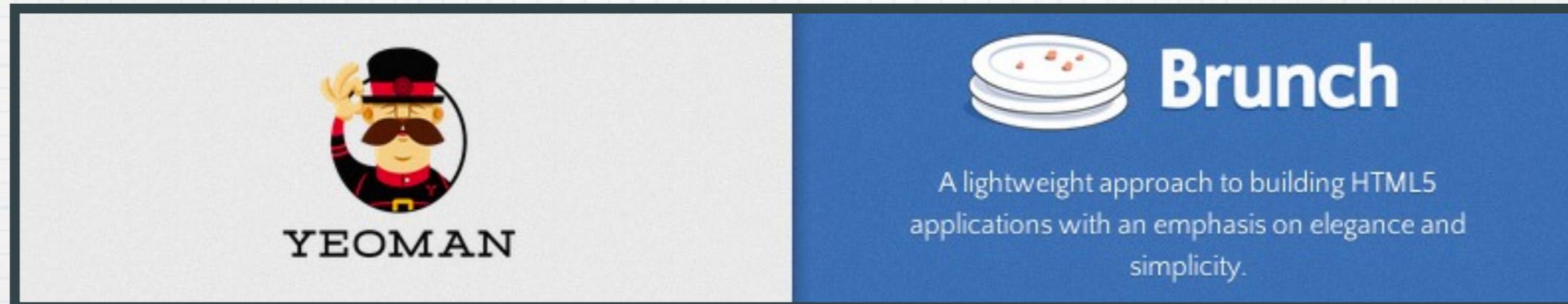
本当に

大事なことに時間を！

次世代ツール

CLIの世界では、さらに次の動きが始まっていたり

Yeoman & Brunch



- Skeletonの作成やScaffolding
- パッケージ/コンポーネント管理
- ローカル開発環境の提供
- より高度な自動ビルドの提供 などなど

Compare Table

Feature	Yeoman	Brunch	Grunt	LiveReload	CodeKit
Builder, linter, concatenator, minifier	✓	✓	✓	✓	✓
File watcher	✓	✓	✓	✓	✓
Automatic browser reloading	✓	✓	✓	✓	✓
Blazing fast third-party API	✗	✓	✗	✗	✗
Automatic wrapping of scripts in modules	✓	✓	✓	✗	✗
OS X Notification Center / Growl / Inotify output for errors	✗	✓	✗	✗	✓
Built-in webserver	✓	✓	✓	✗	✗
Skeletons (boilerplates)	✓	✓	✓	✗	✗
Scaffolding (generate / destroy)	⚠ ※1	✓	✗	✗	✗
Headless testing without a browser	✓	✓	✓	✗	✗
Cross-platform	✗	✓	✓	✗	✗
Free	✓	✓	✓	✗	✗
Image Optimizer	✓	✗	⚠ ※2	✗	✓
Package Manager	✓	✗	✗	✗	✗

つづきはWebで！

- Yeoman - Modern workflows for webapps
- Brunch : HTML5 application assembler
- Yeoman ヨーマン ::ハブろぐ
- YeomanとBrunchをさわさわした ::ハブろぐ

ありがとうございました

<http://aho.mu>

Photo Credits

- <http://www.flickr.com/photos/jurvetson/6858583426/>
- <http://www.flickr.com/photos/jasonwryan/4130500122/>
- <http://www.flickr.com/photos/binaryape/2945088118/>
- <http://www.flickr.com/photos/juhansonin/854759383/>
- <http://www.flickr.com/photos/kevin/5387998/>
- Many Thanks!!