La Méthode MTC-PTP

Moins de temps à coder, plus de temps à la plage

REMI CORSON

Automattic

@remicorson

remicorson.com

Jeux d'roles

Démo

What the Grunt?

Grunt est un tasks runner

Cake - Gulp - Broccoli

Grunt est basé sur

Noje.js

Node.js permet d'utiliser JavaScript en dehors du navigateur

Installer Node.js

https://nodejs.org/

NPM

Node Package Manager

Installer Grunt

```
$ npm init // puis plusieurs fois entrée
$ npm install -g grunt-cli

$ npm install grunt --save-dev // dans le projet
```

Test de l'install Grunt

```
$ grunt --help
```

```
Usage
 grunt [options] [task [task ...]]

Options
    --help, -h  Display this help text.
        --base  Specify an alternate base path. By default, all file paths are
                relative to the Gruntfile. (grunt.file.setBase) *
    --no-color  Disable colored output.
   --gruntfile  Specify an alternate Gruntfile. By default, grunt looks in the
                current or parent directories for the nearest Gruntfile.js or
                Gruntfile.coffee file.
   --debug, -d  Enable debugging mode for tasks that support it.
       --stack  Print a stack trace when exiting with a warning or fatal error.
   --force, -f  A way to force your way past warnings. Want a suggestion? Don't
                use this option, fix your code.
       --tasks  Additional directory paths to scan for task and "extra" files.
                (grunt.loadTasks) *
         --npm  Npm-installed grunt plugins to scan for task and "extra" files.
                (grunt.loadNpmTasks) *
    --no-write  Disable writing files (dry run).
 --verbose, -v  Verbose mode. A lot more information output.
 --version, -V  Print the grunt version. Combine with --verbose for more info.
  --completion  Output shell auto-completion rules. See the grunt-cli
                documentation for more information.
```

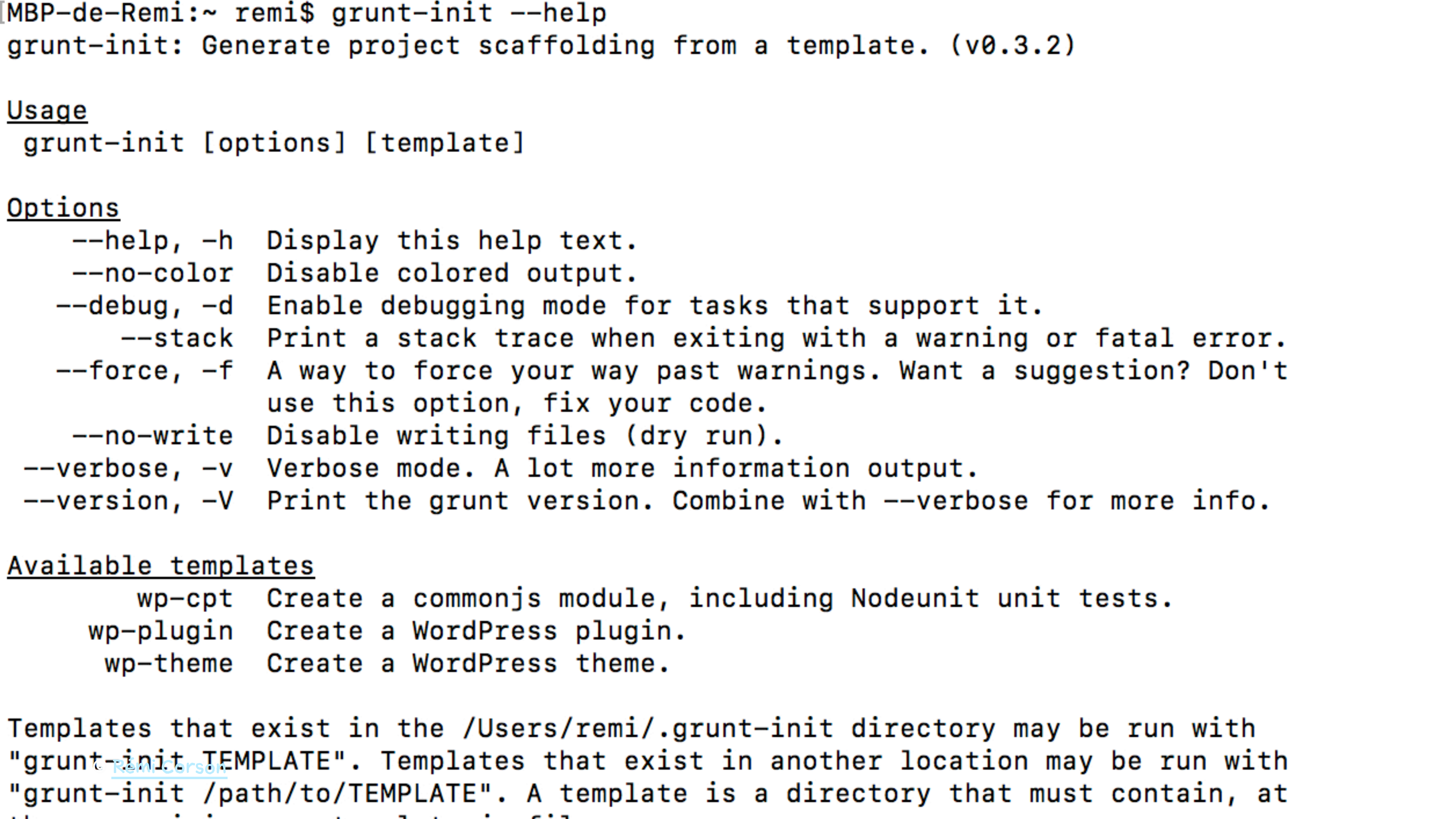
Grunt-init

Project Scaffolding

```
npm install -g grunt-init
```

Test de l'install Grunt-init

```
$ grunt --help
```



```
MBP-de-Remi:~ remi$ grunt-init --help
grunt-init: Generate project scaffolding from a template. (v0.3.2)

Usage
 grunt-init [options] [template]

Options
    --help, -h  Display this help text.
    --no-color  Disable colored output.
   --debug, -d  Enable debugging mode for tasks that support it.
       --stack  Print a stack trace when exiting with a warning or fatal error.
   --force, -f  A way to force your way past warnings. Want a suggestion? Don't
                use this option, fix your code.
    --no-write  Disable writing files (dry run).
 --verbose, -v  Verbose mode. A lot more information output.
 --version, -V  Print the grunt version. Combine with --verbose for more info.

Available templates
        wp-cpt  Create a commonjs module, including Nodeunit unit tests.
     wp-plugin  Create a WordPress plugin.
      wp-theme  Create a WordPress theme.

Templates that exist in the /Users/remi/.grunt-init directory may be run with
"grunt-init TEMPLATE". Templates that exist in another location may be run with
"grunt-init /path/to/TEMPLATE". A template is a directory that must contain, at
```

Créer un template

Le dossier du template se trouve sur macOS

```
~/.grunt-init/
```

ou sur Windows

```
%USERPROFILE%\.grunt-init\
```

Structure d'un template

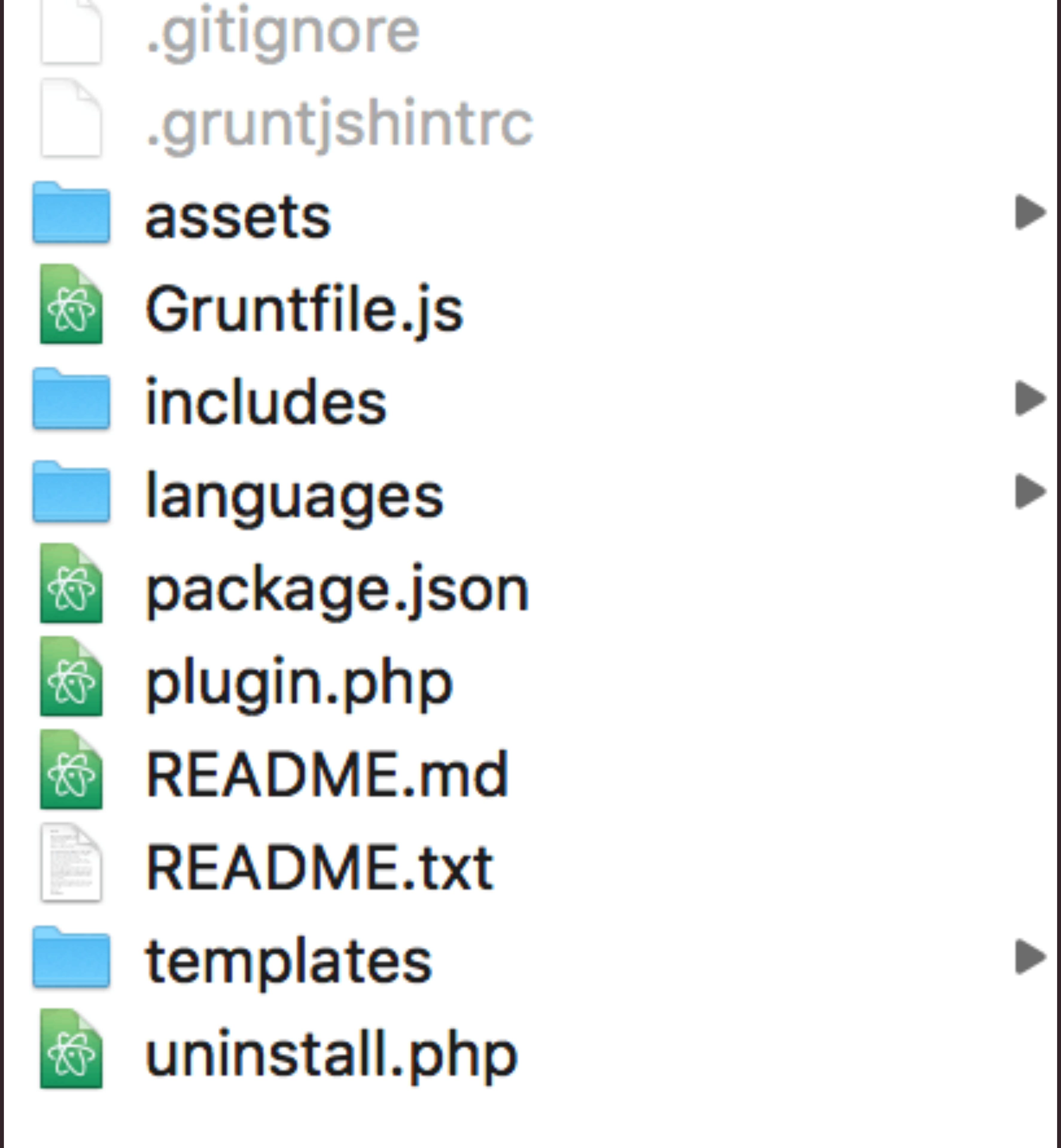
3 Fichiers

my-template/template.js

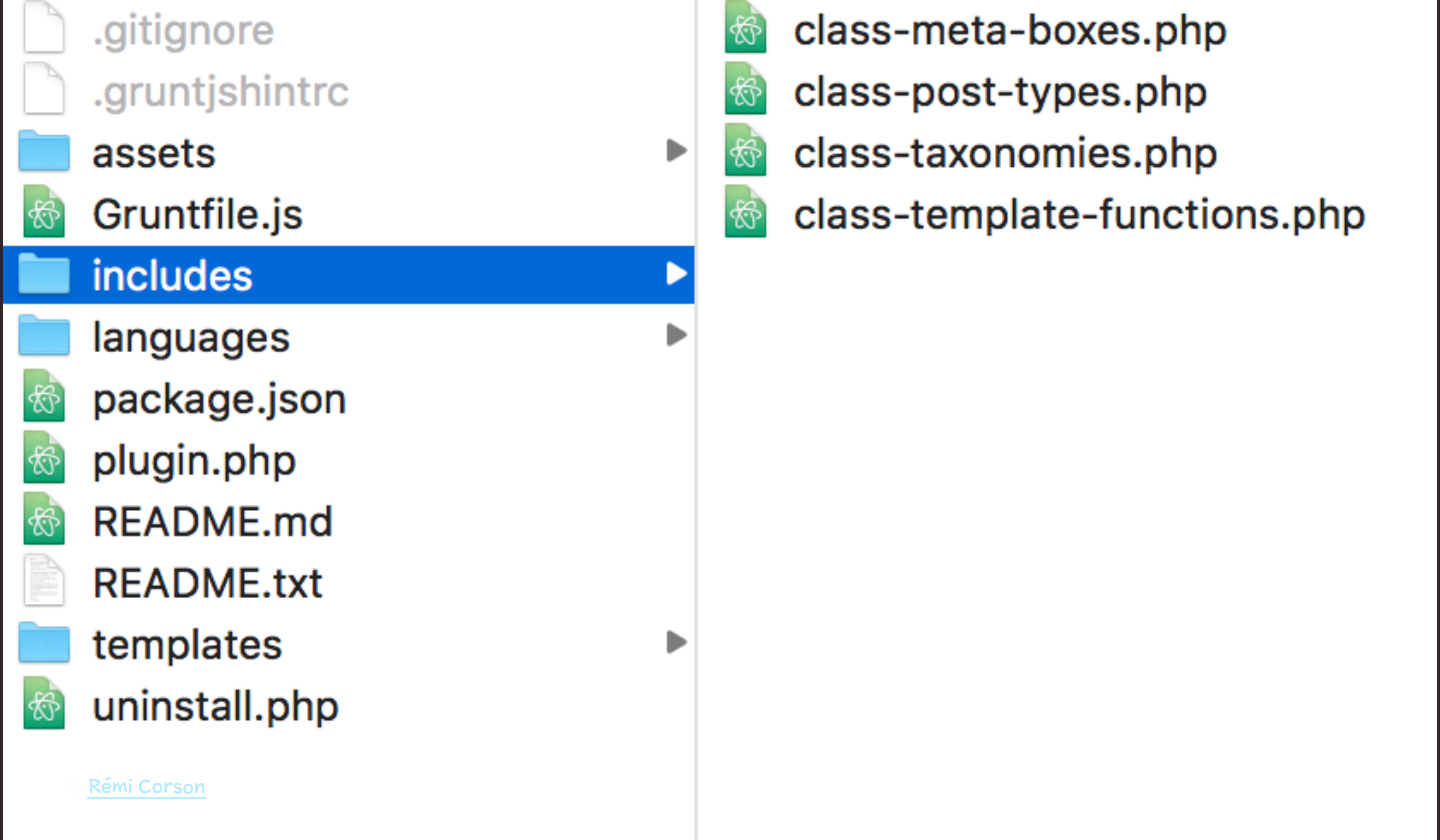
my-template/rename.json

my-template/root/

root/



```
.gitignore
.gruntjshintrc
assets
Gruntfile.js
includes
languages
package.json
plugin.php
README.md
README.txt
templates
uninstall.php
```



.gitignore
.gruntjshintrc
assets
Gruntfile.js
includes
languages
package.json
plugin.php
README.md
README.txt
templates
uninstall.php

class-meta-boxes.php
class-post-types.php
class-taxonomies.php
class-template-functions.php

Rémi Corson

template.js

Avant de commencer

```
// Basic template description
exports.description = 'Create a WordPress plugin.';

// Template-specific notes to be displayed before question prompts.
exports.notes = '';

// Template-specific notes to be displayed after the question prompts.
exports.after = '';

// Any existing file or directory matching this wildcard will cause a warning.
exports.warnOn = '*';
```

Définir le projet

```
// The actual init template
exports.template = function( grunt, init, done ) {
    init.process( {}, [
        // Prompt for these values.
        init.prompt( 'title', 'My Book Store' ),
        init.prompt( 'description', 'A small plugin to do things better with WordPress!' ),
        init.prompt( 'plugin_uri', 'https://remicorson.com' ),
        init.prompt( 'author_name' ),
        init.prompt( 'author_email', 'remicorson@nospam.com' ),
        init.prompt( 'author_url', 'https://remicorson.com' ),
        ...
```

Poser une question

```
{
    name: 'custom_post_type',
    message: 'WordPress CPT : enter the CPT name (ex: Book. Leave empty if no need for CPT)'
},
{
    name: 'taxonomy',
    message: 'WordPress Taxonomy : enter the taxonomy name (ex: Style. Leave empty if no need for taxonomy)'
}
```

Valeur par défaut 1

```
{
    name: 'css_type',
    message: 'CSS Preprocessor: Will you use "Sass", "LESS", or "none" for CSS with this project?',
    default: 'Sass'
},
```

Valeur par défaut 2

```
{
    name: 'enable_templates',
    message: 'Enable CPT templating (y/n)?',
    default: 'yes'
},
```

Dépendances

```
], function( err, props ) {
    props.keywords = [];
    props.version = '0.1.0';
    props.devDependencies = {
            "grunt": "^0.4.5",
            "grunt-checktextdomain": "^1.0.1",
            "grunt-contrib-clean": "^0.5.0",
            "grunt-contrib-compress": "^0.8.0",
            "grunt-contrib-copy": "^1.0.0",
            "dp-grunt-contrib-copy": "^0.4.4",
            "grunt-contrib-cssmin": "^1.0.2",
            "grunt-contrib-uglify": "^2.0.0",
            "grunt-contrib-watch": "^1.0.0",
            "grunt-text-replace": "^0.4.0",
            "grunt-wp-i18n": "^0.5.4"
    };
```

Préparer des variables

```
// Sanitize names where we need to for PHP/JS
props.name = props.title.replace( /\s+/g, '-' ).toLowerCase();

// Development prefix (i.e. to prefix PHP function names, variables)
props.prefix = props.prefix.replace('/[^a-z_]/i', '').toLowerCase();

// Development prefix in all caps (e.g. for constants)
props.prefix_caps = props.prefix.toUpperCase();

// An additional value, safe to use as a JavaScript identifier.
props.safe_filename = props.name.replace(/[\W_]+/g, '-').replace(/^(\d)/, '-$1');
```

Copier des fichiers

```
var files = init.filesToCopy( props );
```

Copies conditionelles 1

```
// Post Type Class
if( props.custom_post_type === '' ) {
    delete files[ 'includes/class-post-types.php'];

    props.custom_post_type = '';
}
```

Copies conditionnelles 2

```
switch( props.css_type.toLowerCase()[0] ) {
    case 'l': // Use Less, delete Sass related files.
        delete files[ 'assets/css/style.scss'];
        delete files[ 'assets/css/admin/style.scss'];

        props.devDependencies["grunt-contrib-less"] = "~1.4.0";
        props.css_type = 'less';
        break;
    case 'n': // Use none, delete Sass and Less related files.
    case undefined:
        delete files[ 'assets/css/style.less'];
        delete files[ 'assets/css/style.scss'];
        delete files[ 'assets/css/admin/style.less'];
        delete files[ 'assets/css/admin/style.scss'];

        props.css_type = 'none';
        break;
    default: // Use Less files.
        delete files[ 'assets/css/style.less'];
        delete files[ 'assets/css/admin/style.less'];

        props.devDependencies["grunt-contrib-sass"] = "~1.0.0";
        props.css_type = 'sass';
        break;
}
```

Theming

```
// Template Class
switch( props.enable_templates ) {
    case 'no':
        delete files[ 'includes/class-template-loader.php'];
        delete files[ 'templates/archive-.php'];
        delete files[ 'templates/content-.php'];
        delete files[ 'templates/'];

        props.enable_templates = 'no';
        break;
}
```

Do the Magic!

```
// Actually copy and process files
init.copyAndProcess( files, props );

// Generate package.json file
init.writePackageJSON( 'package.json', props );

// Done!
done();
});
};
```

rename.js

Renommer à la volée

```
{
    "plugin.php": "{%= js_safe_name %}.php",
    "templates/archive-cpt.php": "templates/archive-{%= custom_post_type.toLowerCase() %}.php",
    "templates/content-cpt.php": "templates/content-{%= custom_post_type.toLowerCase() %}.php",
    "includes/class-meta-boxes.php": "includes/class-{%= safe_filename %}-meta-boxes.php",
    "includes/class-post-types.php": "includes/class-{%= safe_filename %}-post-types.php",
    "includes/class-taxonomies.php": "includes/class-{%= safe_filename %}-taxonomies.php",
    "includes/class-template-functions.php": "includes/class-{%= safe_filename %}-template-functions.php",
    "languages/plugin.pot": "languages/{%= safe_filename %}.pot"
}
```

Recodage Party!

Readme.txt

```
=== {%= title %} ===
Contributors: {%= author_name %}
Donate link: {%= author_url %}
Tags:
Requires at least:
Tested up to:
Stable tag: 1.0.0

{%= description %}
```

plugin.php

```
/**
 * Plugin Name: {%= title %}
 * Plugin URI: {%= plugin_uri %}
 * Description: {%= description %}
 * Version: 1.0.0
 * Author: {%= author_name %}, Remi Corson
 * Contributors: {%= author_name %}, Remi Corson
 * Author URI: {%= author_url %}
 * Text Domain: {%= title.replace( /\s+/g, '_' ).toLowerCase() %}
 * Domain Path: /languages
```

```
if ( ! class_exists( '{%= _.capitalize(title.replace( /\s+/g, '_' )) %}' ) ) :

    /**
     * Main {%= _.capitalize(title.replace( /\s+/g, '_' )) %} class.
     */
    class {%= _.capitalize(title.replace( /\s+/g, '_' )) %} {
```

```
// Define constants.
define( '{%= title.replace( /\s+/g, "_" ).toUpperCase() %}_VERSION', '1.0.0' );
```

```
// Includes.
{% if ( custom_post_type != '' ) { %}include_once( 'includes/class-{%= safe_filename %}-post-types.php' );{% } %}
{% if ( custom_post_type != '' ) { %}include_once( 'includes/class-{%= safe_filename %}-meta-boxes.php' );{% } %}
{% if ( enable_templates == 'yes' ) { %}include_once( 'includes/class-{%= safe_filename %}-template-functions.php' );{% } %}
{% if ( taxonomy != '' ) { %}include_once( 'includes/class-{%= safe_filename %}-taxonomies.php' );{% } %}
```

Gruntfile.js

Compiler le CSS

```
// Compile all .scss files.
sass: {
    compile: {
        options: {
            sourcemap: 'none'
        },
        files: [{
            expand: true,
            cwd: '<%= dirs.css %>/',
            src: ['*.scss'],
            dest: '<%= dirs.css %>/',
            ext: '.css'
        }]
    }
},
```

Minifier le CSS

```
// Minify all .css files.
cssmin: {
    minify: {
        expand: true,
        cwd: '<%= dirs.css %>/',
        src: ['*.css'],
        dest: '<%= dirs.css %>/',
        ext: '.css'
    }
},
```

Minifier le JS

```
// Minify .js files.
uglify: {
    options: {
        preserveComments: 'some'
    },
    jsfiles: {
        files: [{
            expand: true,
            cwd: '<%= dirs.js %>/',
            src: [
                '*.js',
                '!*.min.js',
                '!Gruntfile.js',
            ],
            dest: '<%= dirs.js %>/',
            ext: '.min.js'
        }]
    }
},
```

Watch

```
// Watch changes for assets
watch: {
    css: {
        files: ['<%= dirs.css %>/*.scss'],
        tasks: ['sass', 'cssmin'],
    },
    js: {
        files: [
            '<%= dirs.js %>/*js',
            '!<%= dirs.js %>/*.min.js'
        ],
        tasks: ['uglify']
    }
},
```

Make Pot

```
// Create .pot files
makepot: {
    dist: {
        options: {
            domainPath: '/languages/',
            exclude: [
                        'node_modules/.*',
                        'build/.*'
                    ],
            potFilename: 'js_safe_name.pot',
            type: 'wp-plugin',
            potHeaders: {
                    'language': 'en',
                    'plural-forms': 'nplurals=2; plural=(n != 1);',
                    'x-poedit-country': 'United States',
                    'x-poedit-sourcecharset': 'UTF-8',
                    'x-poedit-keywordslist': '__;_e;__ngettext:1,2;_n:1,2;__ngettext_noop:1,2;_n_noop:1,2;_c;_nc:1,2;_x:1,2c;_ex:1,2c;_nx:4c,1,2;_nx_noop:4c,1,2;',
                    'x-poedit-basepath': '../',
                    'x-poedit-searchpath-0': '.',
                    'x-poedit-bookmarks': '',
                    'x-textdomain-support': 'yes',
                'report-msgid-bugs-to': 'https://remicorson.com',
                'language-team': 'YOUR TEAM <YOUR-EMAIL@ADDRESS>'
            }
        }
    }
},
```

Clean

```
// Clean up build directory
clean: {
    main: ['build/<%= pkg.name %>']
},
```

copy dans build/

```
// Copy into the build directory
copy: {
    main: {
        expand: true,
        src: [
            '**',
            '!node_modules/**',
            '!build/**',
            '!.sass-cache/**',
            '!Gruntfile.js',
            '!package.json',
            '!**/Gruntfile.js',
            '!**/package.json',
            '!<%= dirs.css %>/*.scss',
            '!**/*~'
        ],
        dest: 'build/<%= pkg.name %>/',
    },
 },
```

Zip

```
//Compress build directory into <name>.zip and <name>-<version>.zip
compress: {
    main: {
        options: {
            mode: 'zip',
            archive: './build/<%= pkg.name %>-<%= pkg.version %>.zip'
        },
        expand: true,
        cwd: 'build/<%= pkg.name %>/',
        src: ['**/*'],
        dest: '<%= pkg.name %>/'
    }
},
```

Modules

```
// Load NPM tasks to be used here
grunt.loadNpmTasks( 'grunt-wp-i18n' );
grunt.loadNpmTasks( 'grunt-contrib-sass' );
grunt.loadNpmTasks( 'grunt-contrib-cssmin' );
grunt.loadNpmTasks( 'grunt-contrib-uglify' );
grunt.loadNpmTasks( 'grunt-contrib-watch' );
grunt.loadNpmTasks( 'grunt-checktextdomain' );
grunt.loadNpmTasks( 'grunt-contrib-clean' );
grunt.loadNpmTasks( 'grunt-contrib-copy' );
grunt.loadNpmTasks( 'grunt-contrib-compress' );
```

Register Tasks

```
// Register tasks
grunt.registerTask( 'default', [
    'sass',
    'cssmin',
    'uglify'
] );

// Just an alias for pot file generation
grunt.registerTask( 'pot', [
    'makepot'
] );

// Build task(s).
grunt.registerTask( 'build', [
    'clean',
    'copy',
    'compress',
    'build-clean'
] );

// Build-clean task(s).
grunt.registerTask( 'build-clean', [
    'clean'
] );
```

BOOM!