



Get Grulping with JavaScript Task Runners

Matt Gifford

@coldfumonkeh

monkehworks.com

**OBLIGATORY QUOTE
FROM HISTORICAL FIGURE
COMING UP**

**“life is really simple,
but we insist on making
it complicated”**

Confucius

WTF?





IT HURTS

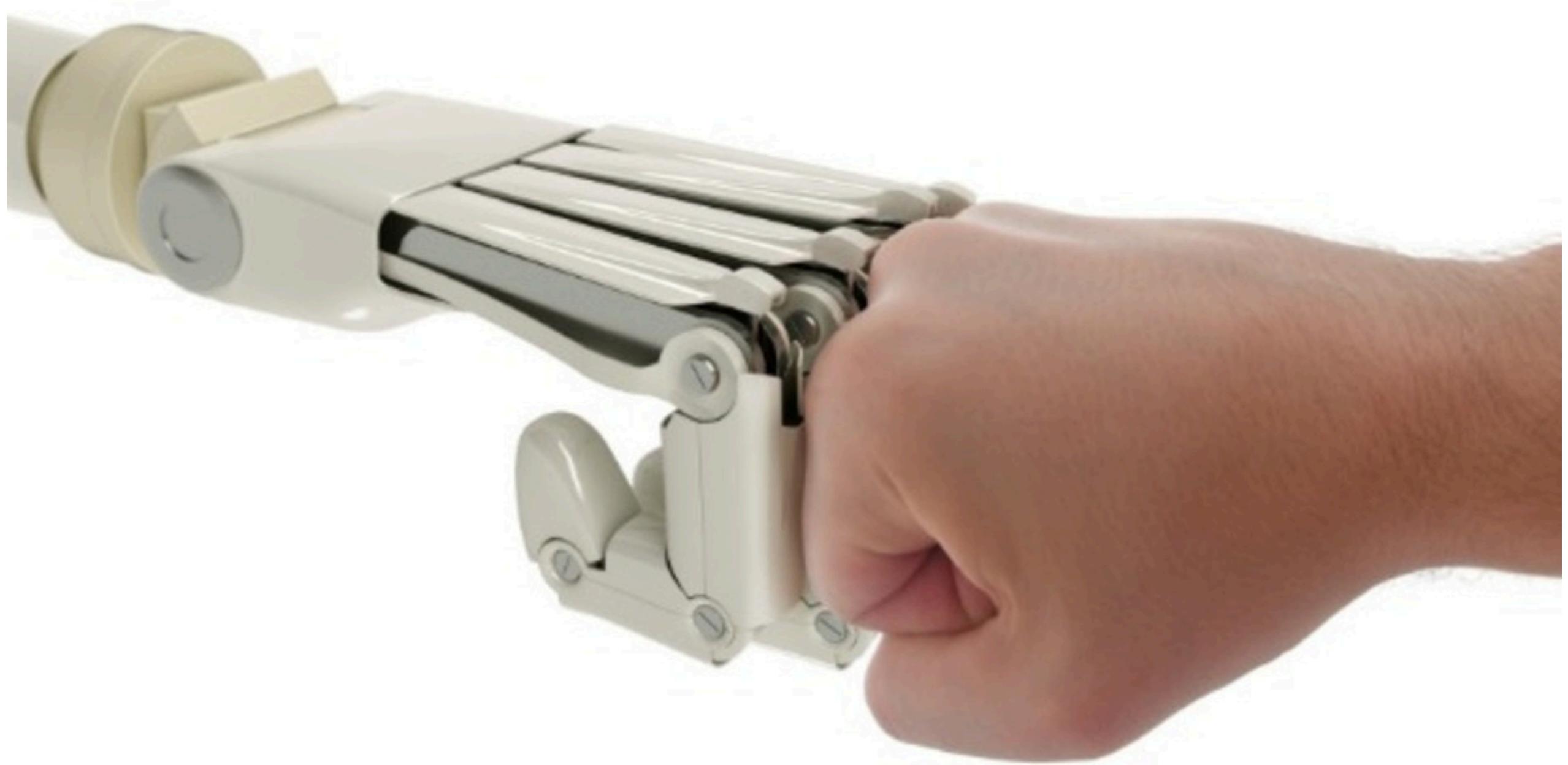
almost all quality improvement comes via
simplification of design, manufacturing, layout,
processes and procedures.

Tom Peters

almost all quality improvement comes via
simplification of design, manufacturing, layout,
processes and procedures.

Tom Peters









Node.js is a platform built on [Chrome's JavaScript runtime](#) for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

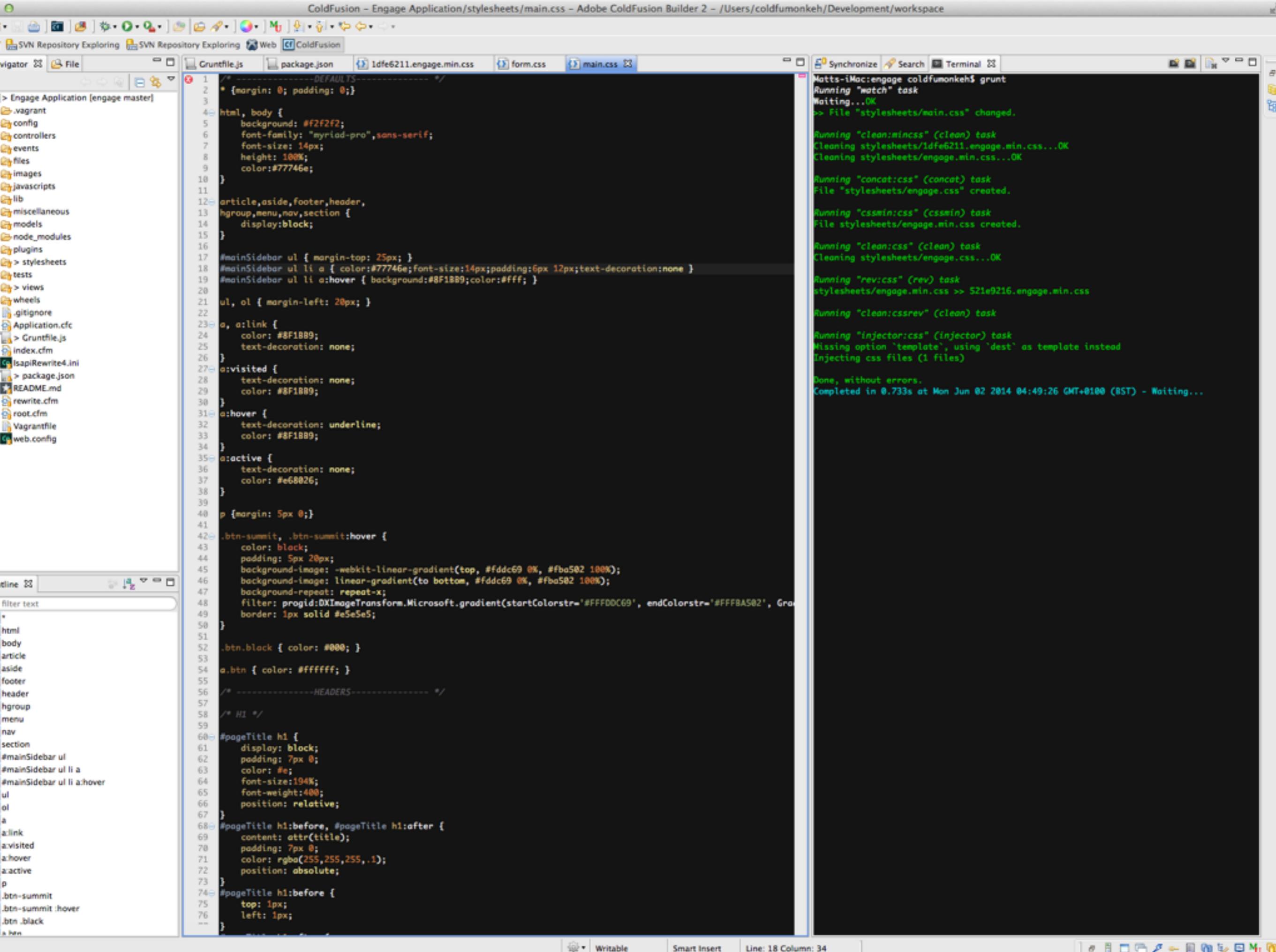
Current Version: v0.10.28

[INSTALL](#)

[DOWNLOADS](#)

[API DOCS](#)

<http://nodejs.org>



The screenshot shows a Sublime Text editor window with a dark theme. On the left, a sidebar displays a project structure with a tree view of files and folders. The main editor area contains a Gruntfile.js script. A context menu is open over the 'server' task, listing several options: 'lint', 'qunit', 'min', 'watch', and 'server'. The 'server' option is highlighted. The code in the editor is as follows:

```
31
32
33
34
35
36
37
38
39
40
41
42
43
```

lint
Validate files with JSHint.
Grunt default tasks

qunit
Run QUnit unit tests in a headless PhantomJS instance.
Grunt default tasks

min
Minify files with UglifyJS.
Grunt default tasks

watch
Run predefined tasks whenever watched files change.
Grunt default tasks

server
Start a static web server.
Grunt default tasks

```
s.parentNode.insertBefore(g,s)}(document,
    script'));
</script>
```

[Finished in 0.7s with exit code 3]

Line 35, Column 21 Spaces: 4 HTML

<https://github.com/tvooo/sublime-grunt>



<http://gruntjs.com>

@gruntjs



0.4.X



<http://gruntjs.com/plugins>

2,954

as of 8:30am 5th June 2014



YOU NEED

package.json

Gruntfile.js



```
$ npm init
```

This utility will walk you through creating a package.json file.

It only covers the most common items,
and tries to guess sane defaults.

Press ^C at any time to quit.

name: (grunt_project) grunting_away

version: (0.0.0) 0.0.1

description:

entry point: (index.js)

test command:

.... etc

package.json

```
{  
  "name": "grunting_away",  
  "version": "0.0.1",  
  "description": "",  
  "main": "index.js",  
  "author": "Matt Gifford",  
  "license": "ISC"  
}
```

package.json

```
{  
  "name": "grunting_away",  
  "version": "0.0.1"  
}
```

INSTALLING GRUNT

Use npm to install the required modules



```
$ npm install < whatever the module name is >
```

You may need **sudo** or Administrative rights

INSTALLING GRUNT

Grunt 0.3 requires a global install of the library

Grunt 0.4 changed... a lot (for the better)

Now has the ability to run different local versions



```
$ npm install grunt-cli -g
```

-g installs the CLI package globally. Good times

INSTALLING GRUNT

We have the global CLI. Now we need a local Grunt



```
$ npm install grunt --save-dev
```

```
$ grunt --version
```

```
grunt-cli v0.1.13  
grunt v0.4.5
```

package.json

```
{  
  "name": "grunting_away",  
  "version": "0.0.1",  
  "devDependencies": {  
    "grunt": "^0.4.5"  
  }  
}
```

package.json

```
{  
  "name": "grunting_away",  
  "version": "0.0.1",  
  "devDependencies": {  
    "grunt": "^0.4.5"  
  }  
}
```

Gruntfile.js

Lives in the root directory of your project

Commit it into your source control repo!

Holds your task configurations

Can be written as Gruntfile.coffee
(if that floats your boat)

Gruntfile.js

```
module.exports = function(grunt) {  
  grunt.initConfig({  
  
    // Pure awesomeness will live here  
  
  });  
};
```

VERSION CONTROL

| -- package.json

| -- Gruntfile.js



commit these and share the wealth

TEAM GRUNTING

```
$ npm install
```

THE CODE BASE

| -- javascripts

 -- main.js

 -- formControls.js

| -- stylesheets

 -- form.css

 -- main.css



can be managed more effectively



CSS CONCATENATION

```
$ npm install grunt-contrib-concat --save-dev
```

package.json

```
{  
  "name": "grunting_away",  
  "version": "0.0.1",  
  "devDependencies": {  
    "grunt": "^0.4.5",  
    "grunt-contrib-concat": "^0.4.0"  
  }  
}
```

Gruntfile.js

```
module.exports = function(grunt) {
  grunt.initConfig({
    // Pure awesomeness will live here
  });

  grunt.loadNpmTasks('grunt-contrib-concat');
};

};
```

Gruntfile.js

```
grunt.initConfig({  
  concat : {  
    css: {  
      files: {  
        'stylesheets/engage.css' :  
          ['stylesheets/*.css']  
      }  
    }  
  }  
});
```

Gruntfile.js

```
grunt.initConfig({  
  concat : {  
    css: {  
      files: {  
        'stylesheets/engage.css' :  
          [  
            'stylesheets/main.css',  
            'stylesheets/form.css'  
          ]  
      }  
    }  
  }  
});
```

```
$ grunt concat
```

Running "concat:css" (concat) task
File stylesheets/engage.css created.

Done, without errors.

```
$ grunt concat:css
```

Running "concat:css" (concat) task
File stylesheets/engage.css created.

Done, without errors.

THE CODE BASE

| -- stylesheets

 -- **engage.css**

 -- form.css

 -- main.css



new file generated by Grunt

Gruntfile.js

```
grunt.initConfig({  
  
  concat : {  
    css: {  
  
      options: {  
        banner: '/* Combined css file */\n'  
      },  
  
      /* snip */  
    }  
  }  
})
```

Gruntfile.js

```
grunt.initConfig({  
  pkg: grunt.file.readJSON('package.json'),  
  
  concat : {  
    css: {  
      options: {  
        banner: '/* <%= pkg.name %> combined file  
generated @  
  <%= grunt.template.today("dd-mm-yyyy") %> */\n'  
      },  
  
      /* snip */  
    }  
  }  
})
```

```
1 /* grunting_away combined file generated @ 02-06-2014 */
2
3 /* -----DEFAULTS----- */
4 * {margin: 0; padding: 0;}
5
6 html, body {
7     background: #f2f2f2;
8     font-family: "myriad-pro", sans-serif;
9     font-size: 14px;
10    height: 100%;
11    color:#77746e;
12 }
13
14 article, aside, footer, header,
15 hgroup, menu, nav, section {
16     display:block;
17 }
18
19 #mainSidebar ul { margin-top: 25px; }
20 #mainSidebar ul li a { color:#77746e; font-size:14px; padding:6px 12px; text-decoration:none; }
21 #mainSidebar ul li a:hover { background:#8F1BB9; color:#fff; }
22
23 ul, ol { margin-left: 20px; }
24
25 a, a:link {
26     color: #8F1BB9;
27     text-decoration: none;
28 }
29 a:visited {
30     text-decoration: none;
31     color: #8F1BB9;
32 }
33 a:hover {
```

CSS MINIFICATION

```
$ npm install grunt-contrib-cssmin --save-dev
```

package.json

```
{  
  "name": "grunting_away",  
  "version": "0.1.0",  
  "devDependencies": {  
    "grunt": "^0.4.5",  
    "grunt-contrib-concat": "^0.4.0",  
    "grunt-contrib-cssmin": "^0.9.0"  
  }  
}
```

Gruntfile.js

```
module.exports = function(grunt) {  
  grunt.initConfig({  
  
    // Pure awesomeness will live here  
  
  });  
  
  grunt.loadNpmTasks('grunt-contrib-concat');  
  grunt.loadNpmTasks('grunt-contrib-cssmin');  
  
};
```

Gruntfile.js

```
grunt.initConfig({  
  /* snip */  
  
  cssmin: {  
    css: {  
      files: {  
        'stylesheets/engage.min.css' :  
          [ 'stylesheets/engage.css' ]  
      }  
    },  
  },  
},  
);
```

```
$ grunt cssmin
```

Running "cssmin:css" (cssmin) task

File stylesheets/engage.min.css created:
29.73 kB → 23.62 kB

Done, without errors.

THE CODE BASE

| -- stylesheets

-- engage.css

-- engage.min.css

-- form.css

-- main.css



minified file

```
1 /*! grunting_away minified CSS file generated @ 02-06-2014 */
2
3 *{margin:0;padding:0}body,html{background:#f2f2f2;font-family:myriad-pro,sans-serif;font-size:14px;height:100%;
```

CACHE BUSTING

```
$ npm install grunt-rev --save-dev
```

Gruntfile.js

```
grunt.loadNpmTasks('grunt-rev');

rev: {
  css: {
    files: {
      src: ['stylesheets/engage.min.css']
    }
  }
},
```

```
$ grunt rev:css
```

```
Running "rev:css" (rev) task  
stylesheets/engage.min.css >> 73a5cf64.engage.min.css
```

```
Done, without errors.
```

THE CODE BASE

| -- stylesheets

 -- **73a5cf64.engage.min.css**

 -- engage.css

 -- engage.min.css

 -- form.css

 -- main.css



hashed minified file

THE CODE BASE

| -- stylesheets

-- 73a5cf64.engage.min.css

-- engage.css

-- engage.min.css

-- form.css

-- main.css



we don't need these

CLEAN UP OPERATION

```
$ npm install grunt-contrib-clean --save-dev
```

Gruntfile.js

```
grunt.loadNpmTasks('grunt-contrib-clean');

clean: {
  combinedcss: {
    src: ['stylesheets/engage.css']
  },
  mincss: {
    src: ['stylesheets/engage.min.css']
  },
  revcss: {
    src: ['stylesheets/*engage.min.css']
  }
},
```

TOO MANY TASKS

We already have a load of tasks to run

What happens when we need to run them all?

Type each command out?

REMEMBER

simplification of processes and procedures

Gruntfile.js

```
grunt.registerTask('css',  
  [  
    'clean:revcss',  
    'concat:css',  
    'cssmin:css',  
    'clean:combinedcss',  
    'rev:css',  
    'clean:mincss'  
  ]);
```

```
$ grunt css
```

```
Running "clean:revcss" (clean) task
Cleaning stylesheets/73a5cf64.engage.min.css...OK
```

```
Running "concat:css" (concat) task
File stylesheets/engage.css created.
```

```
Running "cssmin:css" (cssmin) task
File stylesheets/engage.min.css created: 29.73 kB → 23.62 kB
```

```
Running "clean:combinedcss" (clean) task
Cleaning stylesheets/engage.css...OK
```

```
Running "rev:css" (rev) task
stylesheets/engage.min.css >> 73a5cf64.engage.min.css
```

```
Running "clean:mincss" (clean) task
```

```
Done, without errors.
```

THE CODE BASE

```
| -- stylesheets  
  -- 73a5cf64.engage.min.css  
  -- form.css  
  -- main.css
```

A close-up of the Minion from Despicable Me. He is wearing dark sunglasses with black frames and chains hanging from the sides. He has his signature yellow skin, a single brown eye, and a small red mouth. He is wearing a brown trench coat over a white shirt. He is looking over his shoulder towards the camera.

WATCHING...

ALWAYS WATCHING

WATCHING FOR FILE CHANGES

```
$ npm install grunt-contrib-watch --save-dev
```

Gruntfile.js

```
grunt.loadNpmTasks('grunt-contrib-watch');

watch: {
  css: {
    files: [
      'stylesheets/form.css',
      'stylesheets/main.css'
    ],
    tasks: ['css']
  }
},
```

Gruntfile.js

```
grunt.registerTask('default', ['watch']);
```

```
$ grunt
```

Running "watch" task

Waiting...

>> File "stylesheets/main.css" changed.

Running "clean:revcss" (clean) task

Cleaning stylesheets/73a5cf64.engage.min.css...OK

Running "concat:css" (concat) task

File stylesheets/engage.css created.

Running "cssmin:css" (cssmin) task

File stylesheets/engage.min.css created: 29.73 kB → 23.62 kB

Running "clean:combinedcss" (clean) task

Cleaning stylesheets/engage.css...OK

Running "rev:css" (rev) task

stylesheets/engage.min.css >> 73a5cf64.engage.min.css

Running "clean:mincss" (clean) task

Done, without errors.

Completed in 0.485s at Mon Jun 02 2014 02:26:21 GMT+0100 (BST) - Waiting...



THE CODE BASE

```
| -- javascripts  
  -- main.js  
  -- formControls.js
```

JAVASCRIPT MANAGEMENT

```
$ npm install grunt-contrib-jshint --save-dev
```

```
$ npm install grunt-contrib-uglify --save-dev
```

```
$ npm install grunt-remove-logging --save-dev
```

GRUNTFILE.JS

```
grunt.loadNpmTasks('grunt-contrib-jshint');

jshint: {
  options: {
    curly: true,
    eqeqeq: true,
    eqnull: true,
    browser: true,
    globals: {
      jQuery: true
    },
  },
  all: ['Gruntfile.js','javascripts/main.js']
},
```

GRUNTFILE.JS

```
grunt.loadNpmTasks('grunt-contrib-uglify');
```

```
uglify : {  
  js: {  
    files: {  
      'javascripts/engage.min.js' :  
        [ 'javascripts/main.js' ]  
    }  
  },  
},
```

GRUNTFILE.JS

```
grunt.loadNpmTasks('grunt-remove-logging');

removeLogging: {
  dist: {
    src: 'javascripts/engage.min.js',
    dest: 'javascripts/engage.min.js'
  }
},
```

GRUNTFILE.JS

```
rev: {  
  css: {  
    files: {  
      src: ['stylesheets/engage.min.css']  
    }  
  },  
  js: {  
    files: {  
      src: ['javascripts/engage.min.js']  
    }  
  }  
},
```

GRUNTFILE.JS

```
rev: {  
  css: {  
    files: {  
      src: ['stylesheets/engage.min.css']  
    }  
  },  
  js: {  
    files: {  
      src: ['javascripts/engage.min.js']  
    }  
  }  
},
```

REMEMBER

simplification of processes and procedures

GRUNTFILE.JS

```
grunt.registerTask('js',  
  [  
    'jshint',  
    'clean:jsrev',  
    'uglify:js',  
    'removeLogging',  
    'rev:js',  
    'clean:minjs'  
]);
```

GRUNTFILE.JS

```
watch: {  
  js: {  
    files: ['javascripts/main.js'],  
    tasks: ['js']  
  },  
  css: {  
    files: [  
      'stylesheets/form.css',  
      'stylesheets/main.css'  
    ],  
    tasks: ['css']  
  }  
},
```

GRUNTFILE.JS

```
watch: {  
  js: {  
    files: ['javascripts/main.js'],  
    tasks: ['js']  
  },  
  css: {  
    files: [  
      'stylesheets/form.css',  
      'stylesheets/main.css'  
    ],  
    tasks: ['css']  
  }  
},
```

```
$ grunt
```

```
Running "watch" task
```

```
Waiting...
```

```
>> File "javascripts/main.js" changed.
```

```
Running "jshint:all" (jshint) task
```

```
javascripts/main.js
```

```
  1 |console.log('monkeh love is good love')  
               ^ Missing semicolon.
```

```
>> 1 error in 2 files
```

```
Warning: Task "jshint:all" failed. Use --force to continue.
```

```
Aborted due to warnings.
```

```
Completed in 2.090s at Mon Jun 02 2014 03:13:55 GMT+0100 (BST) - Waiting...
```

```
$ grunt
```

Running "watch" task

Waiting...

>> File "javascripts/main.js" changed.

Running "jshint:all" (jshint) task

>> 2 files lint free.

Running "clean:jsrev" (clean) task

Cleaning javascripts/engage.min.js...OK

Running "uglify:js" (uglify) task

File javascripts/engage.min.js created: 21 B → 21 B

Running "removelogging:dist" (removelogging) task

Removed 1 logging statements from javascripts/engage.min.js

Running "rev:js" (rev) task

javascripts/engage.min.js >> 0c115107.engage.min.js

Running "clean:minjs" (clean) task

Done, without errors.

Completed in 0.721s at Mon Jun 02 2014 03:14:05 GMT+0100 (BST) – Waiting...



RELOADING YOUR APP

PERFORMING HTTP REQUESTS

```
$ npm install grunt-http --save-dev
```

Gruntfile.js

```
grunt.loadNpmTasks('grunt-http');

http: {
  reload: {
    options: {
      url: 'http://127.0.0.1:8000/index.cfm?reload=true'
    }
  }
},
```

Gruntfile.js

```
grunt.initConfig({
  pkg: grunt.file.readJSON('package.json'),
  local_settings: {
    local_url: ''
  },
  ...
}

http: {
  reload: {
    options: {
      url: '<%= local_settings.local_url %>'
    }
  }
},
```

Gruntfile.js

```
grunt.registerTask('default', ['checklocalconf']);
```

Gruntfile.js

```
grunt.registerTask('checklocalconf', 'Check if the local config JSON file exists', function(arg) {
  if(grunt.file.exists('grunt_local_settings.json')) {
    grunt.task.run('watch');
  } else {
    grunt.log.errorlns('');
    grunt.log.errorlns('The grunt_local_settings.json file does not appear to exist.');
    grunt.log.errorlns('');
    grunt.log.errorlns('{');
    grunt.log.errorlns('  "local_url": "http://your_local_server/?reload"');
    grunt.log.errorlns('}');
    grunt.log.errorlns('');
    grunt.fail.fatal('Please create and save the grunt_local_settings.json file.');
  };
});
```

Gruntfile.js

```
grunt.registerTask('http_watcher',
'Set the local url before running the watch command',
function() {
  var jsonLocalSettings = grunt.file.readJSON("grunt_local_settings.json");
  grunt.config.set('local_settings', jsonLocalSettings);
  grunt.config.requires('local_settings');
  grunt.task.run('http:reload');
});
```

Gruntfile.js

```
watch: {  
  js: {  
    files: ['javascripts/main.js'],  
    tasks: ['js']  
  },  
  css: {  
    files: [  
      'stylesheets/form.css',  
      'stylesheets/main.css'  
    ],  
    tasks: ['css']  
  },  
  cfcs: {  
    files: ['cfcs/*.cfc'],  
    tasks: ['http_watcher']  
  },  
},  
},
```

```
$ grunt
```

```
Running "checklocalconf" task
```

```
Waiting...
```

```
>>
>> The grunt_local_settings.json file does not appear to exist.
>> Please create it in this directory with the following content (the URL
>> for your local app with reload action):
>>
>> {
>>   "local_url": "http://your_local_server/?reload"
>> }
>>
Fatal error: Please create and save the grunt_local_settings.json file then
re-run this command.
```

```
$ grunt
```

```
Running "checklocalconf" task
```

```
Running "watch" task
```

```
Waiting...
```

```
>> File "cfcs/test.cfc" changed.
```

```
Running "http_watcher" task
```

```
Running "http:reload" (http) task
```

```
>> 200
```

```
Done, without errors.
```

```
Completed in 2.061s at Tue Jun 03 2014 12:01:44 GMT+0100 (BST) - Waiting...
```

INJECTING ASSETS

```
$ npm install grunt-injector --save-dev
```

Gruntfile.js

```
grunt.loadNpmTasks('grunt-injector');

injector: {
  options: {},
  css: {
    files: {
      'layout.cfm': ['stylesheets/*engage.min.css'],
    }
  },
  js: {
    files: {
      'layout.cfm': ['javascripts/*engage.min.js'],
    }
  }
}
```

```
1 <!DOCTYPE html>
2 <html lang="en" class="no-js">
3 <head>
4 <meta charset="utf-8"/>
5 <title>Engage</title>
6 <!-- injector:css -->
7 <!-- endinjector -->
8 </head>
9 <body>
10 <header>
11   <h1>Engage</h1>
12 </header>
13 <footer>
14   <small>&copy; 2014 | powered by some <a href="http://www.monkehworks.com" title="Visit monkehWorks Ltd">monkehMagic</a></small>
15 </footer>
16 <!-- injector:js -->
17 <!-- endinjector -->
18 </body>
19 </html>
```

Gruntfile.js

```
grunt.registerTask('css',  
[  
  'clean:revcss',  
  'concat:css',  
  'cssmin:css',  
  'clean:combinedcss',  
  'rev:css',  
  'clean:mincss',  
  'injector:css'  
]);
```

```
1 <!DOCTYPE html>
2 <html lang="en" class="no-js">
3 <head>
4 <meta charset="utf-8"/>
5 <title>Engage</title>
6 <!-- injector:css -->
7 <link rel="stylesheet" href="/stylesheets/73a5cf64.engage.min.css"> ←
8 <!-- endinjector -->
9 </head>
10 <body>
11 <header>
12   <h1>Engage</h1>
13 </header>
14 <footer>
15   <small>&copy; 2014 | powered by some <a href="http://www.monkehworks.com" title="Visit monkehWorks Ltd">monkehMagic</a></small>
16 </footer>
17 <!-- injector:js -->
18 <script src="/javascripts/7c92f895.engage.min.js"></script> ←
19 <!-- endinjector -->
20 </body>
21 </html>
```

TIDY UP

There are no limits to the number
of plugins you can use

You may also be duplicating file paths a lot

Your Gruntfile could get messy quickly

Gruntfile.js

```
grunt.loadNpmTasks('grunt-contrib-clean');
grunt.loadNpmTasks('grunt-contrib-concat');
grunt.loadNpmTasks('grunt-contrib-csslint');
grunt.loadNpmTasks('grunt-contrib-cssmin');
grunt.loadNpmTasks('grunt-http');
grunt.loadNpmTasks('grunt-injector');
grunt.loadNpmTasks('grunt-contrib-jshint');
grunt.loadNpmTasks('grunt-contrib-uglify');
grunt.loadNpmTasks('grunt-contrib-watch');
grunt.loadNpmTasks('grunt-remove-logging');
grunt.loadNpmTasks('grunt-rev');
grunt.loadNpmTasks('grunt-notify');
```

TIDY UP

```
$ npm install matchdep --save-dev
```

Gruntfile.js

```
require('matchdep')
  .filterDev('grunt-*')
  .forEach(grunt.loadNpmTasks);
```

ASSIGN VARIABLES

Use the variable system to reduce duplicate text

Gruntfile.js

```
grunt.initConfig({  
  pkg: grunt.file.readJSON('package.json'),  
  minCSS: 'stylesheets/engage.min.css',  
});  
  
cssmin: {  
  css: {  
    files: {  
      '<%= minCSS %>': [ 'stylesheets/engage.css' ]  
    }  
  }  
},  
  
clean: {  
  mincss: {  
    src: ['<%= minCSS %>']  
  }  
}
```

WHAT ELSE CAN IT DO?

image optimisation and resizing

git integration

run unit tests (e.g. Jasmine)

templating

...

WHAT ELSE CAN IT DO?

pretty much anything you want it to

<http://gruntjs.com/plugins>





<http://gulpjs.com>

@gulpjs



<http://gulpjs.com/plugins/>

635

as of 8:30am 5th June 2014

INSTALLING GULP

As a system wide module



```
$ npm install gulp -g
```

INSTALLING GULP

Getting a local gulp version for the project



```
$ npm install gulp --save-dev
```

gulpfile.js

Lives in the root directory of your project

Commit it into your source control repo!

Holds your task configurations

Lowercase file name!

gulpfile.js

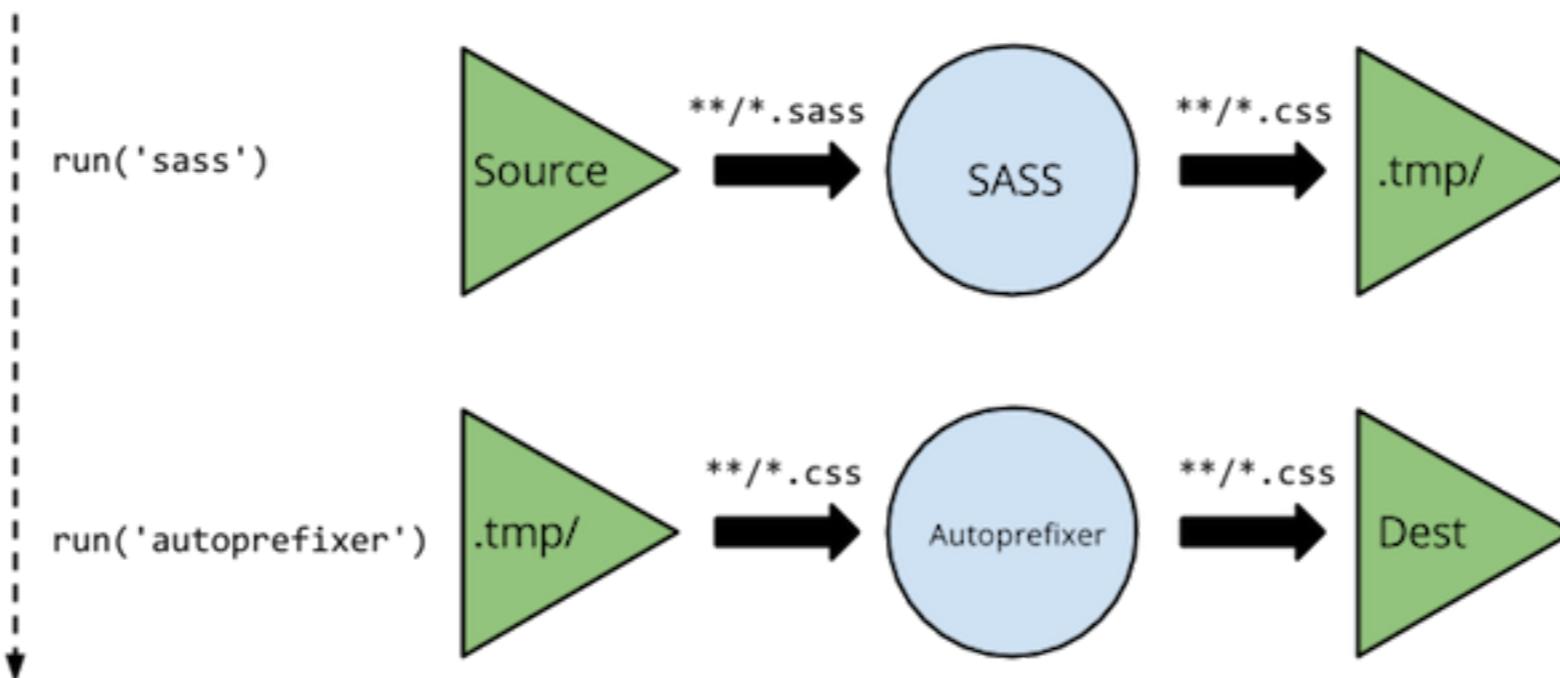
```
// Include gulp
var gulp = require('gulp');

gulp.task('default', function() {
  // place code for your default task here
});
```

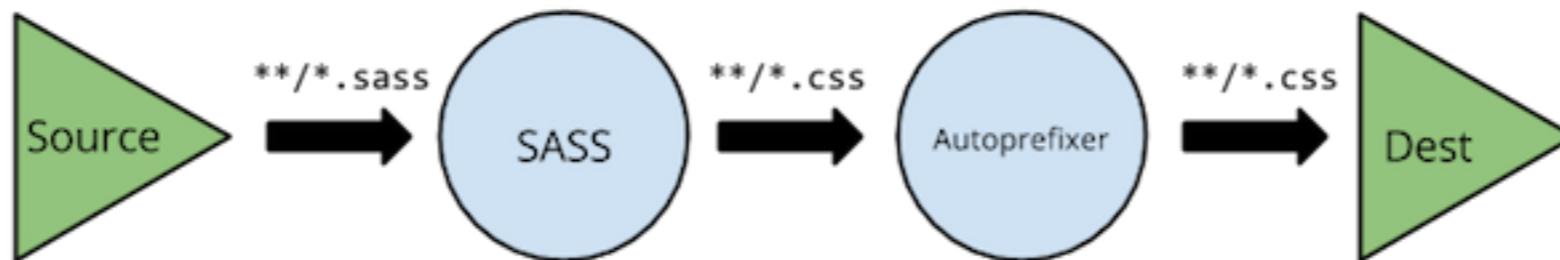


PIPES AND STREAMS

GRUNT



GULP



gulpfile.js

```
// Include gulp
var gulp = require('gulp');

// Include Our Plugins
var jshint = require('gulp-jshint');
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
var rename = require('gulp-rename');
var header = require('gulp-header');
```

gulpfile.js

```
// Default Task
gulp.task('default', ['watch']);

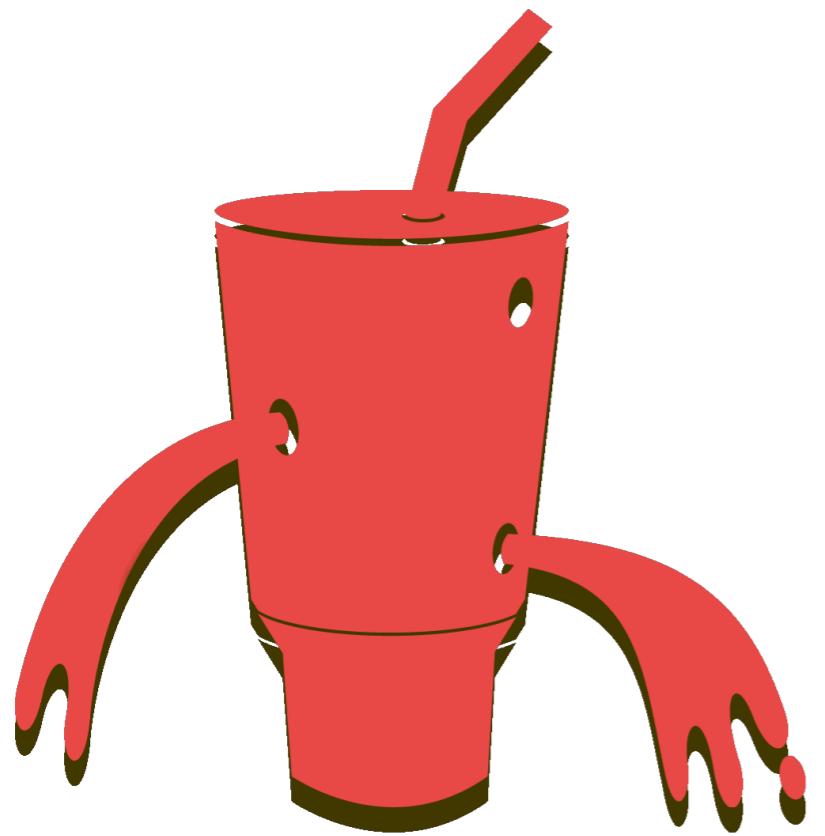
// watch Files For Changes
gulp.task('watch', function() {
  gulp.watch('js/*.js', ['lint', 'scripts']);
});
```

gulpfile.js

```
// Lint Task
gulp.task('lint', function() {
  return gulp.src('js/*.js')
    .pipe(jshint())
    .pipe(jshint.reporter('default'));
});
```

gulpfile.js

```
// Concatenate & Minify JS
gulp.task('scripts', function() {
  var headervalue = "Evaluated by gulp.\n";
  return gulp.src('js/*.js')
    .pipe(concat('combined.js'))
    .pipe(header(headervalue))
    .pipe(gulp.dest('dist'))
    .pipe(rename('combined.min.js'))
    .pipe(uglify())
    .pipe(header(headervalue))
    .pipe(gulp.dest('dist')));
});
```



<http://gulpfiction.divshot.io>

default

- watch
- dependencies
- new_task

**Sources**

Paths for source files.

./src/**/*.js

./lib/**/*.js

**gulp-concat**

By fractal. Version 2.1.7

"all.js"

**Destination**

Path for the result of this task.

./build/



➕ add src

new_task

- watch
- dependencies
- default

**Sources**

Paths for source files.

enter path



➕ add src

Destination

Path for the result of this task.

enter path



add task



or





Streaming and piping give speed enhancements

Code over configuration

Still early adoption - plugins limited

JS / Node exposure beneficial (?)



Sub tasks easily managed

Impressive number of plugins and extensions

I/O issues and speed (in comparison)

Configuration could get messy



ITS NOT A CONTEST

HAPPY

FINAL WORDS

Save your config files (repo)

Use skeleton variation across your projects

-  Create
-  Employ
-  Refine
-  Relax

FINAL WORDS

Has the potential to be addictive

Check for updates and improved methods

Use your time wisely







Kitt Hodsden



David Boyer



Rob Dudley



Thomas Parisot







Thank you!

Matt Gifford
@coldfumonkeh
monkehworks.com