

Let Grunt do the work,  
focus on the fun!

Dirk Ginader, Google, 2013  
@ginader <http://dir.kg/slides>

Let Grunt do the endlessly  
repetitive tedious tasks, focus  
on the important stuff  
like Accessibility!

Dirk Ginader, Google, 2013

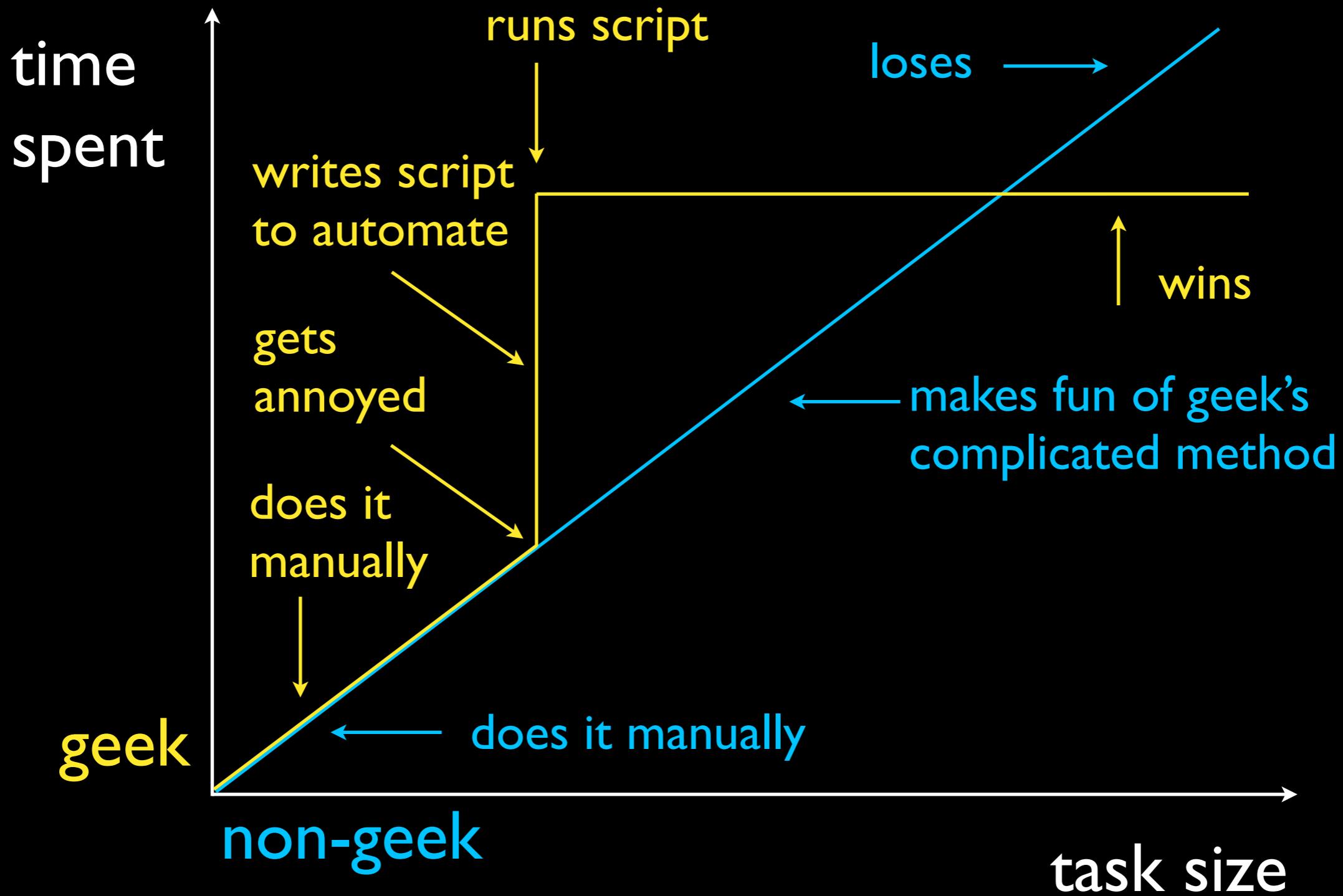
Let Grunt do the work,  
focus on the fun!

Dirk Ginader, Google, 2013

# Why Build scripts?

Because great  
Developers  
are lazy.

Because great  
Developers  
are lazy.  
**FACT.**



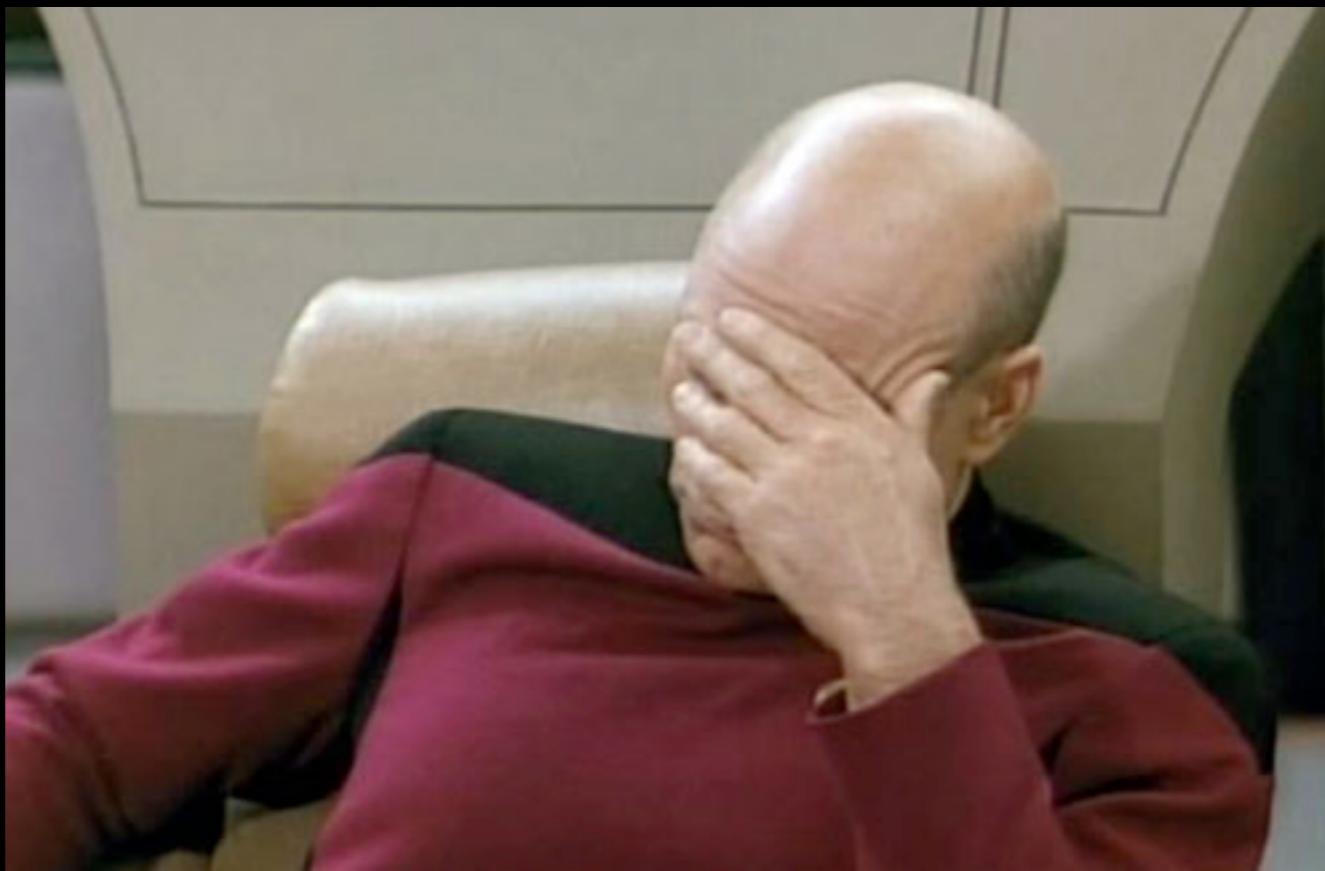
# Build systems have been around for ages

- Make
- Maven
- and so many more ...
- Ant
- Rake

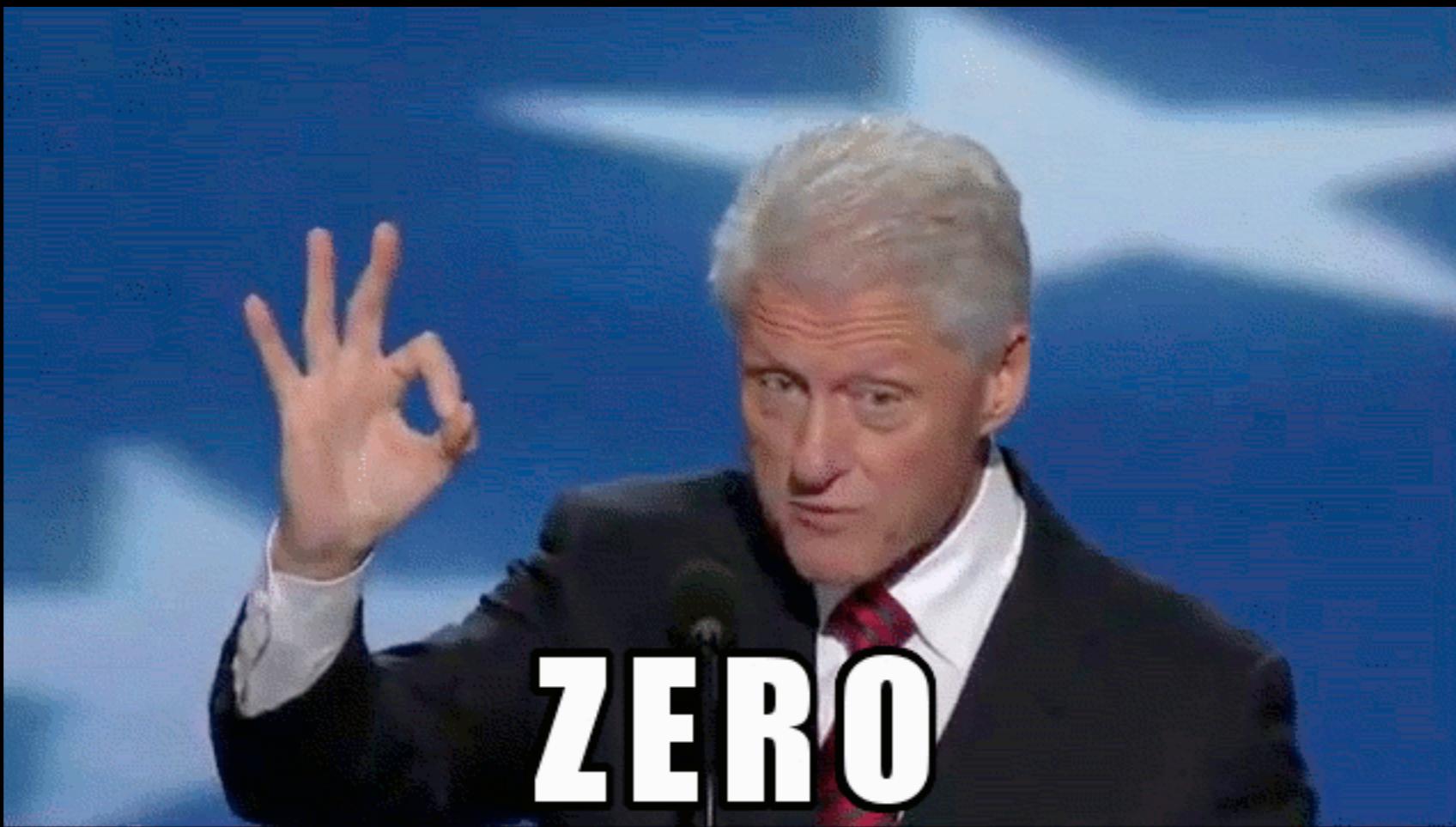
They're all great and  
powerful and all...

# Minify with Ant

```
<target name="js-compile-all" description="Compile JavaScript files with Closure"
unless="skip-js-compile">
    <echo>Compiling JS files in ${input.scripts.dir} in closure...</echo>
    <apply executable="java" dest="${output.scripts.dir}">
        <arg value="-jar"/>
        <arg path="${jar.lib.dir}/closure-compiler.jar"/>
        <arg line="--js"/>
        <srcfile/>
        <arg line="--js_output_file"/>
        <targetfile/>
        <fileset dir="${output.scripts.dir}" includes="**/*-main.debug.js" />
        <mapper type="glob" from="*-main.debug.js" to="*-main.min.js"/>
    </apply>
    <echo>Finished compiling JS files</echo>
</target>
```



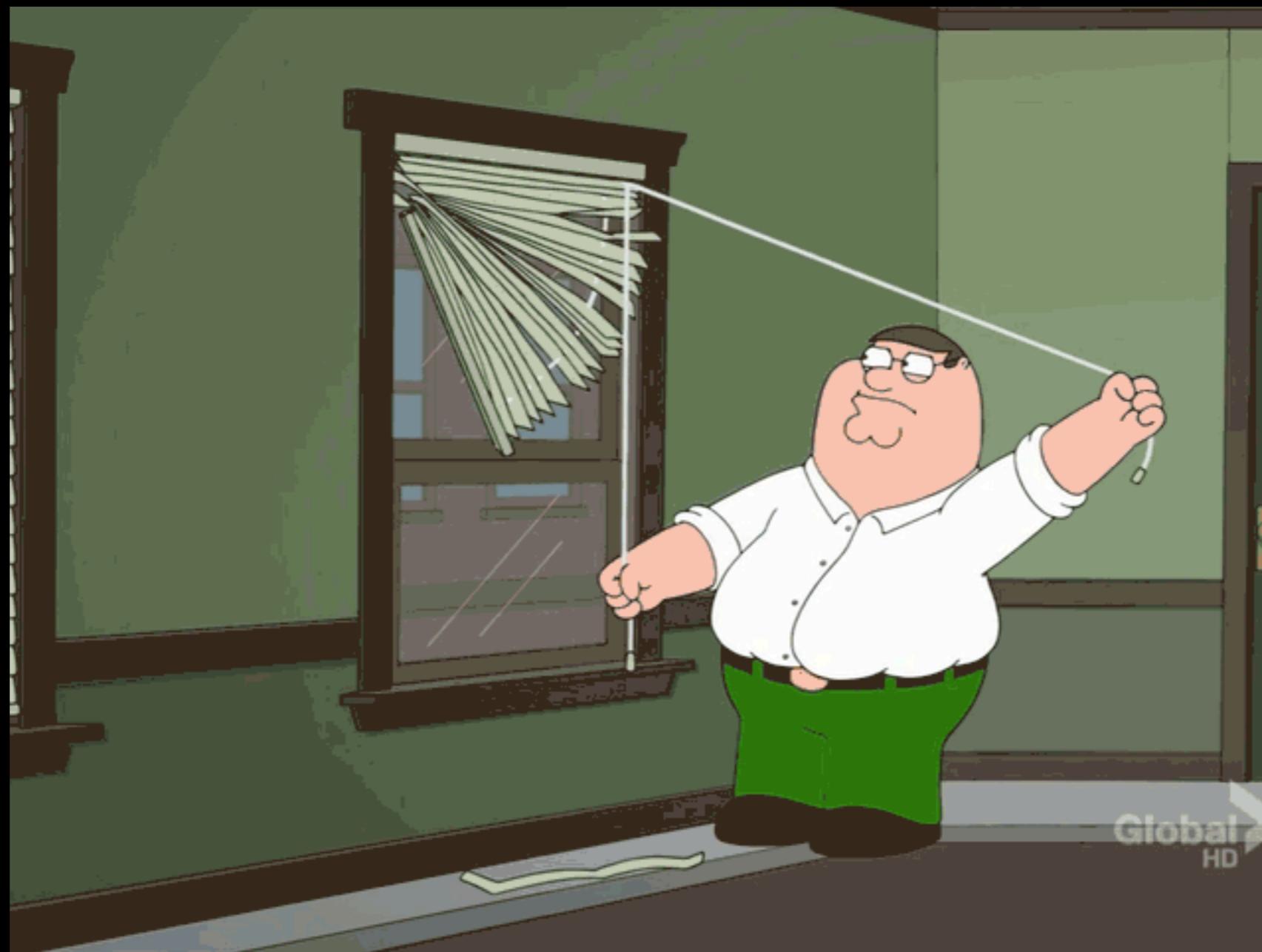
How much I liked to  
configure with XML?



I'm a Front End  
Developer!

I like Javascript

I like LOVE Javascript



Just one year ago  
Ben Alman did me a  
great favor:



# GRUNT

The JavaScript Task Runner



written in Javascript



using the node  
package manager



# FAST adoption rate

- jQuery
- Modernizr
- Adobe
- twitter
- ...

because it's easy!

# System Setup:

download and install  
node.js from:  
<http://nodejs.org/>

node.js

nodejs.org

Fork me on GitHub

# node.js™

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: v0.10.13

[INSTALL](#)

[DOWNLOADS](#) [API DOCS](#)

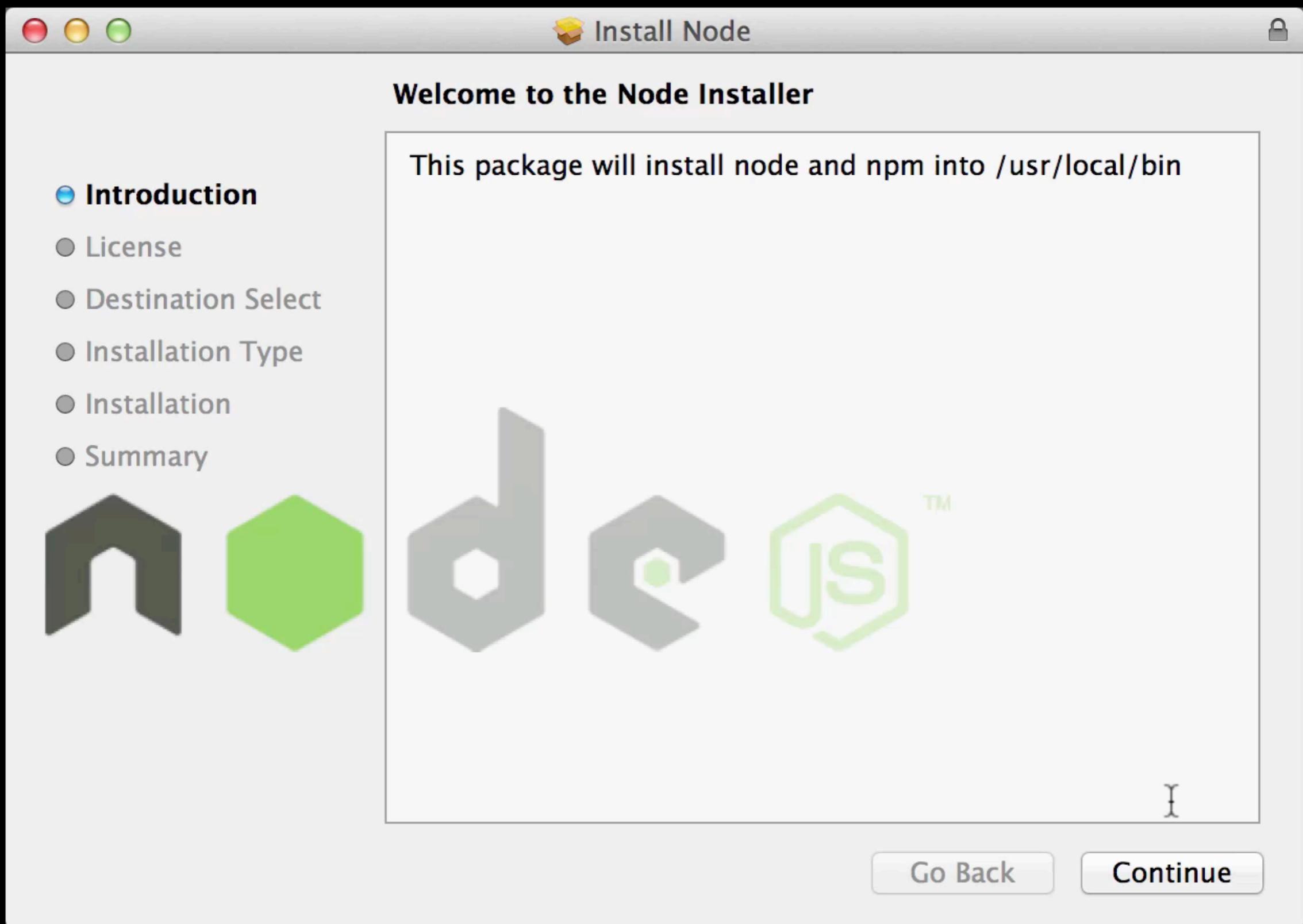
## NODE.JS IN THE INDUSTRY

**DOW JONES**  The simple truth is Node has reinvented the way we create websites. Developers build critical functionality in days, not weeks.  
Michael Yormark

**doclape** We leverage Node.js for almost everything. From handling API requests, streaming files, parsing emails, sending push-notifications and last but not least: converting images, music-

 **SKOOKUM DIGITAL WORKS** Node.js combines the strengths of other server-side languages into a single compelling package. It's accessible like PHP, concise like Ruby, Microsoft-friendly like .NET, and modular like Java.

 **surespot** Node allows the use of coffeescript on both client and server- an added benefit to the elegance and simplicity of the non-blocking programming model, all while keeping our



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

```
npm http GET https://registry.npmjs.org/glob
npm http GET https://registry.npmjs.org/lodash
npm http 304 https://registry.npmjs.org/glob
npm http 304 https://registry.npmjs.org/lodash
npm http GET https://registry.npmjs.org/minimatch
npm http GET https://registry.npmjs.org/graceful-fs
npm http GET https://registry.npmjs.org/inherits
npm http 304 https://registry.npmjs.org/abbrev
npm http 304 https://registry.npmjs.org/graceful-fs
npm http 304 https://registry.npmjs.org/minimatch
npm http 304 https://registry.npmjs.org/inherits
npm http GET https://registry.npmjs.org/lru-cache
npm http GET https://registry.npmjs.org/sigmund
npm http 304 https://registry.npmjs.org/lru-cache
npm http 304 https://registry.npmjs.org/sigmund
/usr/local/bin/grunt -> /usr/local/lib/node_modules/grunt-cli/bin/grunt
grunt-cli@0.1.9 /usr/local/lib/node_modules/grunt-cli
└── resolve@0.3.1
└── nopt@1.0.10 (abbrev@1.0.4)
  └── findup-sync@0.1.2 (lodash@1.0.1, glob@3.1.21)
ginader$ █
```

# Project Setup:

2 important Files:  
package.json  
**Gruntfile.js**

package.json

```
{  
  "name": "jQuery-Accessible-Tabs",  
  "version": "1.9.7",  
  "homepage": "http://github.com/ginader/Accessible-Tabs",  
  "author": {  
    "name": "Dirk Ginader",  
    "url": "http://ginader.com"  
  },  
  "devDependencies": {}  
}
```

<https://npmjs.org/doc/json.html>

# Gives you:

- Variables you can use in your script  
i.e. version and name
- Dev Dependencies that allows you to quickly install all required npm modules

```
{  
  "name": "jQuery-Accessible-Tabs",  
  "version": "1.9.7",  
  "homepage": "http://github.com/ginader/Accessible-Tabs",  
  "author": {  
    "name": "Dirk Ginader",  
    "url": "http://ginader.com"  
  },  
  "devDependencies": {}  
}
```

<https://npmjs.org/doc/json.html>

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

```
ore.string-2.3.2.tgz
npm http 200 https://registry.npmjs.org/underscore.string/-/undersc
ore.string-2.3.2.tgz
npm http 304 https://registry.npmjs.org/underscore
grunt@0.4.1 node_modules/grunt
├── dateformat@1.0.2-1.2.3
├── colors@0.6.0-1
├── which@1.0.5
├── hooker@0.2.3
├── async@0.1.22
├── eventemitter2@0.4.12
├── underscore.string@2.2.1
├── coffee-script@1.3.3
├── rimraf@2.0.3 (graceful-fs@1.1.14)
├── iconv-lite@0.2.10
├── nopt@1.0.10 (abbrev@1.0.4)
├── minimatch@0.2.12 (sigmund@1.0.0, lru-cache@2.3.0)
├── lodash@0.9.2
├── findup-sync@0.1.2 (lodash@1.0.1)
├── glob@3.1.21 (inherits@1.0.0, graceful-fs@1.2.3)
└── js-yaml@2.0.5 (esprima@1.0.2, argparse@0.1.11)
ginader$ █
```

```
{  
  "name": "jQuery-Accessible-Tabs",  
  "version": "1.9.7",  
  "homepage": "http://github.com/ginader/Accessible-Tabs",  
  "author": {  
    "name": "Dirk Ginader",  
    "url": "http://ginader.com"  
  },  
  "devDependencies": {  
    "grunt": "~0.4.0"  
  }  
}
```

<https://npmjs.org/doc/json.html>

install all the defined  
Dependencies in one go

```
$ npm install
```

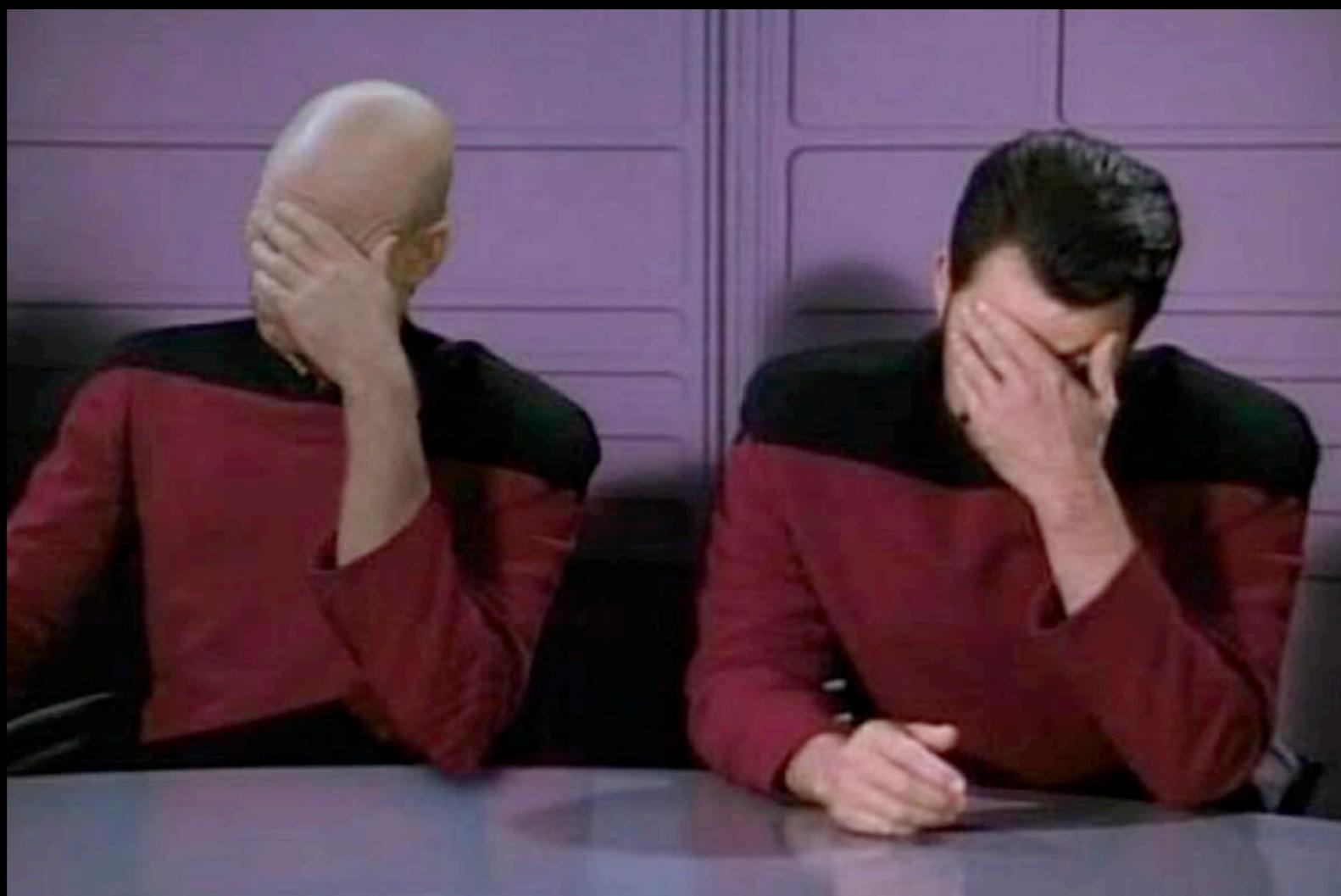
# Gruntfile.js

# Minify with Grunt

```
module.exports = function(grunt) {
  grunt.initConfig({
    uglify: {
      dist: {
        src: 'dist/myfile.js',
        dest: 'dist/myfile.min.js'
      },
    }
  });
  grunt.loadNpmTasks('grunt-contrib-uglify');
  grunt.registerTask('default', ['uglify']);
};
```

# Minify with Ant

```
<target name="js-compile-all" description="Compile JavaScript files with Closure"
unless="skip-js-compile">
    <echo>Compiling JS files in ${input.scripts.dir} in closure...</echo>
    <apply executable="java" dest="${output.scripts.dir}">
        <arg value="-jar"/>
        <arg path="${jar.lib.dir}/closure-compiler.jar"/>
        <arg line="--js"/>
        <srcfile/>
        <arg line="--js_output_file"/>
        <targetfile/>
        <fileset dir="${output.scripts.dir}" includes="**/*-main.debug.js" />
        <mapper type="glob" from="*-main.debug.js" to="*-main.min.js"/>
    </apply>
    <echo>Finished compiling JS files</echo>
</target>
```



# Minify with Grunt

```
module.exports = function(grunt) {
  grunt.initConfig({
    uglify: {
      dist: {
        src: 'dist/myfile.js',
        dest: 'dist/myfile.min.js'
      }
    });
    grunt.loadNpmTasks('grunt-contrib-uglify');
    grunt.registerTask('default', ['uglify']);
  };
};
```

# Minify with Grunt

```
module.exports = function(grunt) {
  grunt.initConfig({
    uglify: {
      dist: {
        src: 'dist/myfile.js',
        dest: 'dist/myfile.min.js'
      }
    });
    grunt.loadNpmTasks('grunt-contrib-uglify');
    grunt.registerTask('default', ['uglify']);
  };
};
```

# Minify with Grunt

```
module.exports = function(grunt) {
  grunt.initConfig({
    uglify: {
      dist: {
        src: 'dist/myfile.js',
        dest: 'dist/myfile.min.js'
      }
    });
    grunt.loadNpmTasks('grunt-contrib-uglify');
    grunt.registerTask('default', ['uglify']);
};
```

let's see what it does!

\$ grunt

```
ginader$ grunt
Running "uglify:dist" (uglify) task
File "js/jquery.tabs.min.js" created.
Uncompressed size: 14500 bytes.
Compressed size: 1024 bytes gzipped (5343 bytes minified).
```

Done, without errors.

```
ginader$ █
```



easy to add more

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

```
npm http GET https://registry.npmjs.org/cli
npm http GET https://registry.npmjs.org/minimatch
npm http GET https://registry.npmjs.org/console-browserify
npm http 304 https://registry.npmjs.org/shelljs
npm http 304 https://registry.npmjs.org/cli
npm http 304 https://registry.npmjs.org/console-browserify
npm http 304 https://registry.npmjs.org/minimatch
npm http 304 https://registry.npmjs.org/underscore
npm http GET https://registry.npmjs.org/lru-cache
npm http GET https://registry.npmjs.org/sigmund
npm http GET https://registry.npmjs.org/glob
npm http 304 https://registry.npmjs.org/lru-cache
npm http 304 https://registry.npmjs.org/sigmund
npm http 304 https://registry.npmjs.org/glob
npm http GET https://registry.npmjs.org/graceful-fs
npm http GET https://registry.npmjs.org/inherits
npm http 304 https://registry.npmjs.org/inherits
npm http 304 https://registry.npmjs.org/graceful-fs
grunt-contrib-jshint@0.6.0 node_modules/grunt-contrib-jshint
└─ jshint@2.1.4 (console-browserify@0.1.6, underscore@1.4.4, shelljs@0.1.4, minimatch@0.2.12, cli@0.4.4-2)
ginader$ █
```

# add JS Linting

```
module.exports = function(grunt) {
  grunt.initConfig({
    jshint: {
      all: ['*.js']
    },
    uglify: {
      dist: {
        src: 'myfile.js',
        dest: 'myfile.min.js'
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-uglify');
  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.registerTask('default', ['jshint', 'uglify']);
};
```

<https://npmjs.org/package/grunt-contrib-jshint>

```
ginader$ grunt
Running "jshint:files" (jshint) task
Linting js/stack-scroll.js...ERROR
[L70:C45] Missing semicolon.
    .addClass('content')
```

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

Aborted due to warnings.

```
ginader$
```

```
ginader$ grunt
Running "jshint:files" (jshint) task
Linting js/stack-scroll.js...ERROR
[L70:C45] Missing semicolon.
    .addClass('content')
```

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

Aborted due to warnings.

```
ginader$ grunt
Running "jshint:files" (jshint) task
>> 2 files lint free.
```

Done, without errors.

```
ginader$
```

tired of typing already?

\$ grunt watch

```
watch: {
  gruntfile: {
    files: [ 'Gruntfile.js', 'js/myfile.js'],
    tasks: ['jshint']
  }
}
```

<https://npmjs.org/package/grunt-contrib-watch>

ginader\$ grunt watch

Running "watch" task

Waiting...OK

>> File "Gruntfile.js" changed.

Running "jshint:gruntfile" (jshint) task

>> 1 file lint free.

Done, without errors.

Completed in 0.507s at Fri Jul 12 2013 17:56:43 GMT-0700 (PDT) - Waiting...■

I

ginader\$ grunt watch

Running "watch" task

Waiting...OK

>> File "Gruntfile.js" changed.

Running "jshint:gruntfile" (jshint) task

>> 1 file lint free.

Done, without errors.

Completed in 0.507s at Fri Jul 12 2013 17:56:43 GMT-0700 (PDT) - Waiting...  
[]

Completed in 0.507s at Fri Jul 12 2013 17:56:43 GMT-0700 (PDT) - Waiting...OK

>> File "Gruntfile.js" changed.

Running "jshint:gruntfile" (jshint) task

Linting Gruntfile.js...ERROR

[L44:C6] Extra comma.

},

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

Aborted due to warnings.

Completed in 0.516s at Fri Jul 12 2013 17:57:47 GMT-0700 (PDT) - Waiting...OK

>> File "Gruntfile.js" changed.

Running "jshint:gruntfile" (jshint) task

>> 1 file lint free.

Done, without errors.

Completed in 0.511s at Fri Jul 12 2013 17:58:29 GMT-0700 (PDT) - Waiting...□

# add data from package.json

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    jshint: {
      all: ['*.js']
    },
    uglify: {
      options: {
        banner: '/*! <%= pkg.name %>' +
          ' <%= grunt.template.today("yyyy-mm-dd") %> */\n'
      },
      dist: {
        src: 'myfile.js',
        dest: 'myfile.min.js'
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-uglify');
  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.registerTask('default', ['jshint', 'uglify']);
};
```

```
{  
  "name": "jQuery-Accessible-Tabs",  
  "version": "1.9.7",  
  "homepage": "http://github.com/ginader/Accessible-Tabs",  
  "author": {  
    "name": "Dirk Ginader",  
    "url": "http://ginader.com"  
  }  
}
```

<https://npmjs.org/doc/json.html>

# add data from package.json

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    jshint: {
      all: ['*.js']
    },
    uglify: {
      options: {
        banner: '/*! <%= pkg.name %>' +
          ' <%= grunt.template.today("yyyy-mm-dd") %> */\n'
      },
      dist: {
        src: '<%= pkg.name %>.js',
        dest: '<%= pkg.name %>.min.js'
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-uglify');
  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.registerTask('default', ['jshint', 'uglify']);
};
```

# add data from package.json

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    jshint: {
      all: ['*.js']
    },
    uglify: {
      options: {
        banner: '/*! <%= pkg.name %>' +
          ' <%= grunt.template.today("yyyy-mm-dd") %> */\n'
      },
      dist: {
        src: '<%= pkg.name %>.js',
        dest: '<%= pkg.name %>.<%= pkg.version %>.min.js'
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-uglify');
  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.registerTask('default', ['jshint', 'uglify']);
};
```

# minify and combine CSS

```
cssmin: {
  compress: {
    options: {
      banner: '<%= banner %>'
    },
    files: {
      'project.min.css': ['1.css','2.css', '...']
    }
  }
}
grunt.loadNpmTasks('grunt-contrib-cssmin');
grunt.registerTask('default', ['jshint','uglify', 'cssmin']);
```

<https://npmjs.org/package/grunt-contrib-cssmin>

# optimize Images

```
imagemin: {  
  dist: {  
    options: {  
      optimizationLevel: 3  
    },  
    files: {  
      // 'destination': 'source'  
      'dist/img.png': 'src/img.png',  
      'dist/img.jpg': 'src/img.jpg'  
    }  
  }  
}  
  
grunt.registerTask('default', ['imagemin']);
```

<https://npmjs.org/package/grunt-contrib-imagemin>

but it's more than just  
optimizations

# Build HTML Pages

```
markdown: {  
  all: {  
    files: ['readme.markdown', 'version-history.markdown'],  
    template: 'web/template.html',  
    dest: 'web',  
    options: {  
      gfm: true,  
      codeLines: {  
        before: '<span>',  
        after: '</span>'  
      }  
    }  
  }  
}
```

[https://npmjs.org/package/  
grunt-markdown](https://npmjs.org/package/grunt-markdown)

# remove debug code

```
removelogging: {  
  dist: {  
    src: 'js/jquery.tabs.min.js',  
    dest: 'js/jquery.tabs.min.js'  
  }  
}
```

[https://npmjs.org/package/  
grunt-remove-logging](https://npmjs.org/package/grunt-remove-logging)

# compile Sass/Compass

```
// setup Compass/Sass to load from existing config.rb
compass: {
  dist: {
    options: {
      config: 'config.rb'
    }
  }
}
```

[https://npmjs.org/package/  
grunt-contrib-compass](https://npmjs.org/package/grunt-contrib-compass)



...but Kitt will tell you all about that  
in the next Session :-)

# and Livereload!



[https://npmjs.org/package/  
grunt-contrib-livereload](https://npmjs.org/package/grunt-contrib-livereload)

all done? deploy!



FTP upload dance?

[https://npmjs.org/  
package/grunt-rsync](https://npmjs.org/package/grunt-rsync)

```
ginader$ grunt deploy
Running "jshint:files" (jshint) task
>> 2 files lint free.
```

```
Running "compass:dist" (compass) task
identical css/print.css
identical css/resume.css
identical css/stack-scroll.css
```

```
Running "rsync:deploy" (rsync) task
./ > ssh-153524-ginader@ginader.com:~/resume.ginader.com done
rsync ./ ssh-153524-ginader@ginader.com:~/resume.ginader.com --rsh=
ssh --recursive --exclude=/node_modules --exclude=.*
```

```
Done, without errors.
ginader$ █
```

# Scaffolding

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

# many templates for grunt-init

- Gruntfile
- Grunt plugin
- jQuery plugin
- node.js
- ...

```
$ git clone git://  
github.com/gruntjs/grunt-  
init-jquery.git ~/.grunt-  
init/jquery
```

```
$ grunt-init jquery
```

Name	Date Mo...	Size	Kind
CONTRIBUTING.md	10:47 AM	2 KB	Markdown
Gruntfile.js	10:47 AM	2 KB	JavaScript
libs	4:26 PM	--	Folder
jquery	10:47 AM	--	Folder
jquery.js	10:47 AM	267 KB	JavaScript
jquery-loader.js	10:47 AM	506 bytes	JavaScript
qunit	10:47 AM	--	Folder
qunit.css	10:47 AM	5 KB	CSS
qunit.js	10:47 AM	57 KB	JavaScript
LICENSE-MIT	10:47 AM	1 KB	Document
node_modules	11:09 AM	--	Folder
grunt	11:09 AM	--	Folder
grunt-contrib-clean	11:09 AM	--	Folder
grunt-contrib-concat	11:09 AM	--	Folder
grunt-contrib-jshint	11:09 AM	--	Folder
grunt-contrib-qunit	11:09 AM	--	Folder
grunt-contrib-uglify	11:09 AM	--	Folder
grunt-contrib-watch	11:09 AM	--	Folder
package.json	10:47 AM	423 bytes	JSON
README.md	10:47 AM	580 bytes	Markdown
src	10:47 AM	--	Folder
super-plugin.js	10:47 AM	848 bytes	JavaScript
super-plugin.jquery.json	10:47 AM	597 bytes	JSON
test	10:47 AM	--	Folder
super-plugin_test.js	10:47 AM	2 KB	JavaScript
super-plugin.html	10:47 AM	1 KB	HTML...ument

# Thank you! Questions?

The opinions I expressed here represent my own  
and not necessarily those of my employer.

btw: We're hiring! Talk to me :-)

# Resources

- Grunt: <http://gruntjs.com/>
- Great article: <http://dir.kg/grunt.workflow>
- Extending Grunt big time: <http://yeoman.io>
- Me: <http://dir.kg/me>
  - [@ginader](https://twitter.com/ginader) on twitter
  - the example projects: <http://github.com/ginader/>
  - <http://ginader.com>
  - <http://dir.kg/slides>