前端工作流

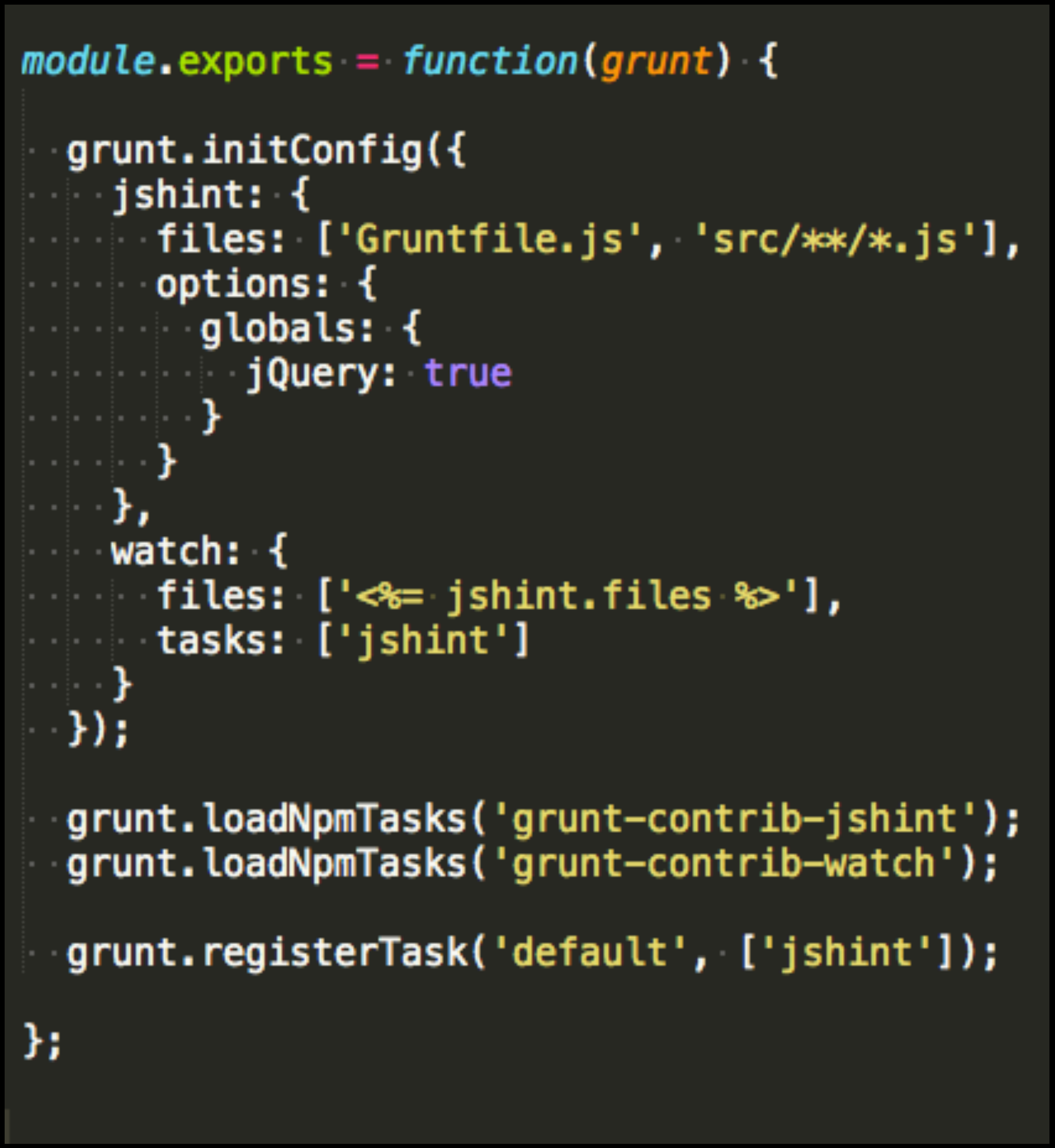
核心工具分享

About me:

陈晓东
2012年加入小米

多看App
多看阅读器
阅读主站
…

代码质量和风格?



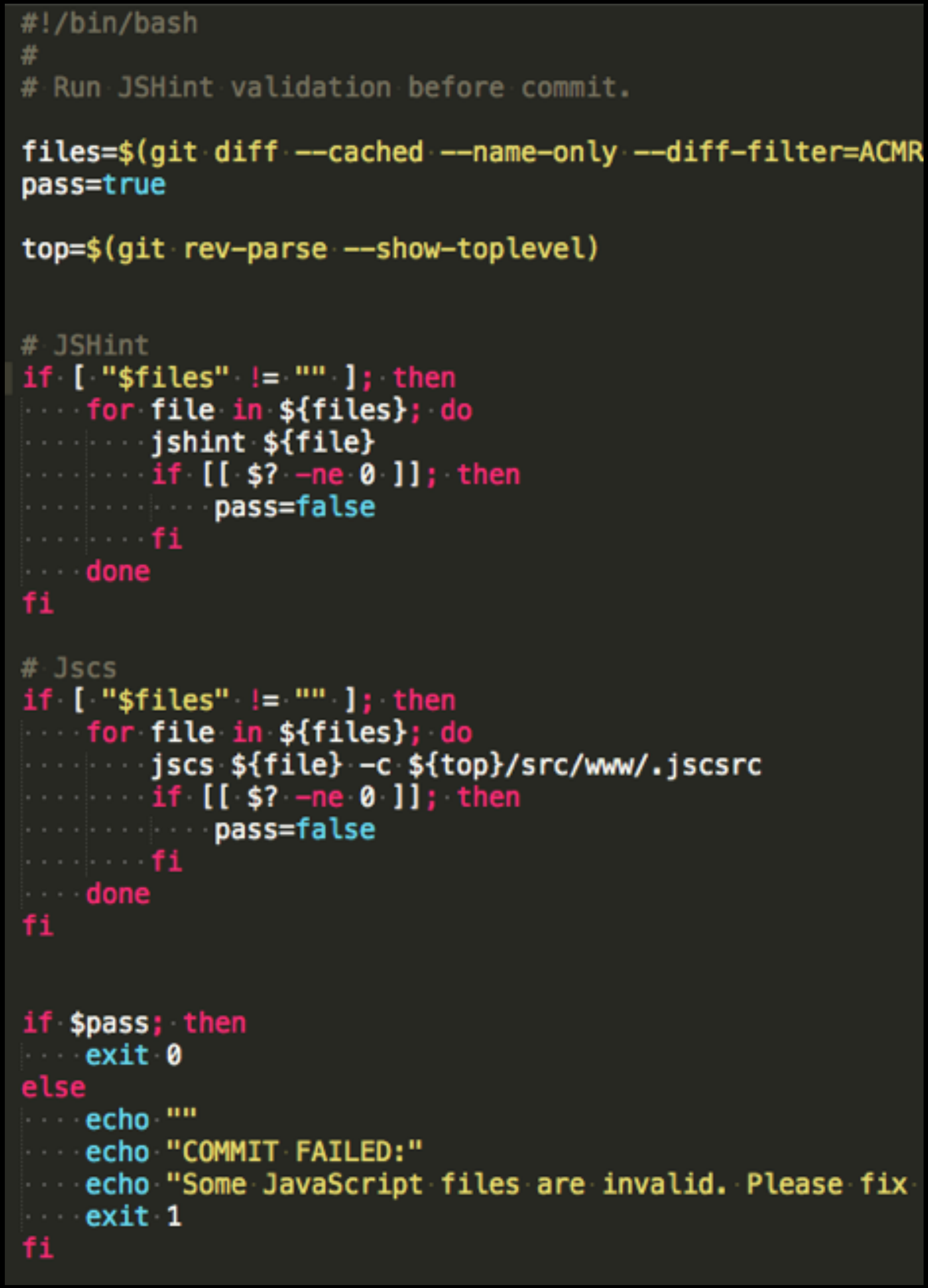
```
module.exports = function(grunt) {

  grunt.initConfig({
    jshint: {
      files: ['Gruntfile.js', 'src/**/*.js'],
      options: {
        globals: {
          jQuery: true
        }
      }
    },
    watch: {
      files: ['<%= jshint.files %>'],
      tasks: ['jshint']
    }
  });

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

  grunt.registerTask('default', ['jshint']);

};
```

grunt watch

容易被忽视

git hook

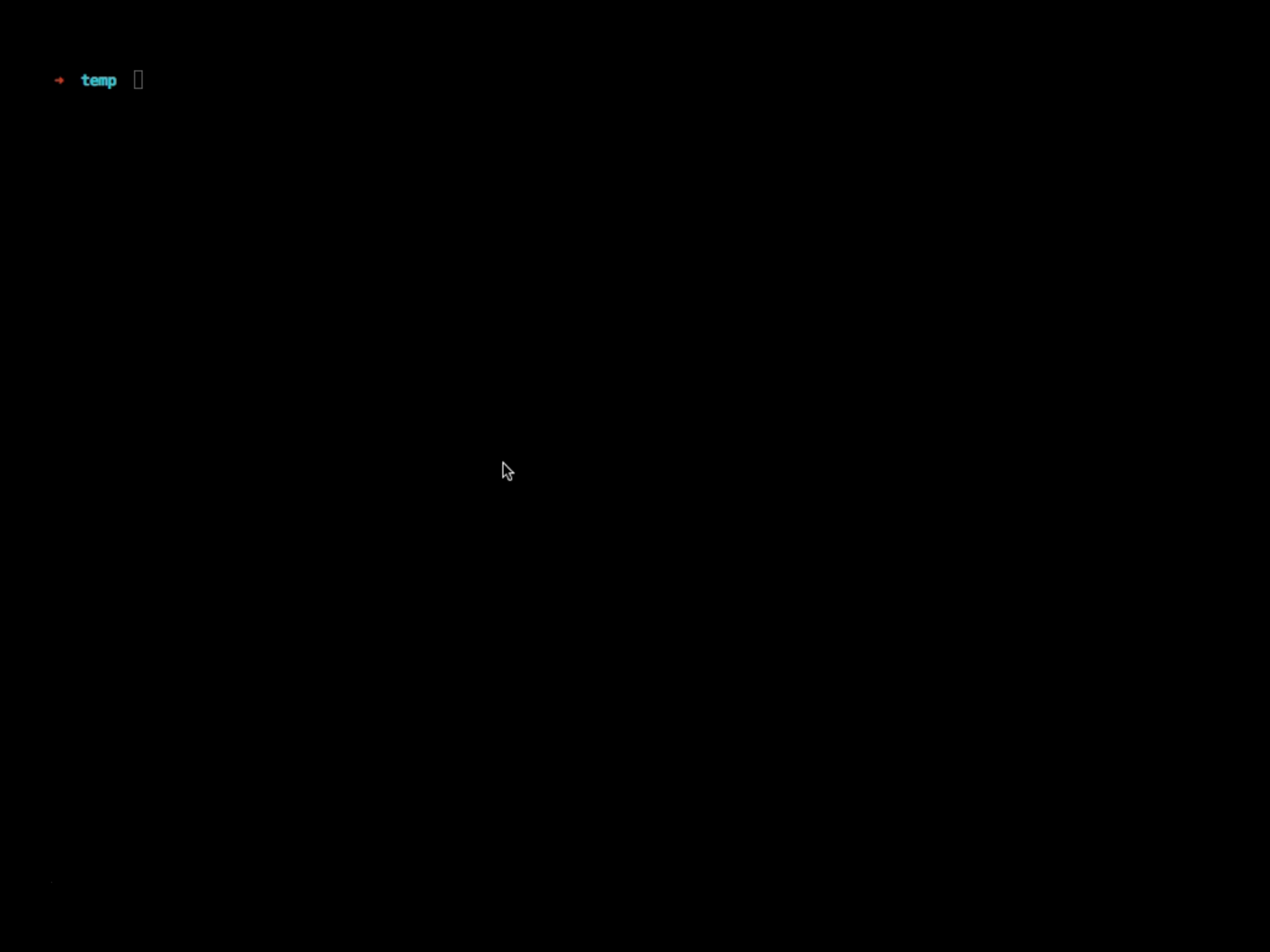
安装不便



```
#!/bin/bash
#
# Run JSHint validation before commit.

files=$(git diff --cached --name-only --diff-filter=ACMR
pass=true

top=$(git rev-parse --show-toplevel)


# JSHint
if [ "$files" != "" ]; then
    for file in ${files}; do
        jshint ${file}
        if [[ $? -ne 0 ]]; then
            pass=false
        fi
    done
fi

# Jscs
if [ "$files" != "" ]; then
    for file in ${files}; do
        jscs ${file} -c ${top}/src/www/.jscsrc
        if [[ $? -ne 0 ]]; then
            pass=false
        fi
    done
fi


if $pass; then
    exit 0
else
    echo ""
    echo "COMMIT FAILED:"
    echo "Some JavaScript files are invalid. Please fix
    exit 1
fi
```

pre-commit

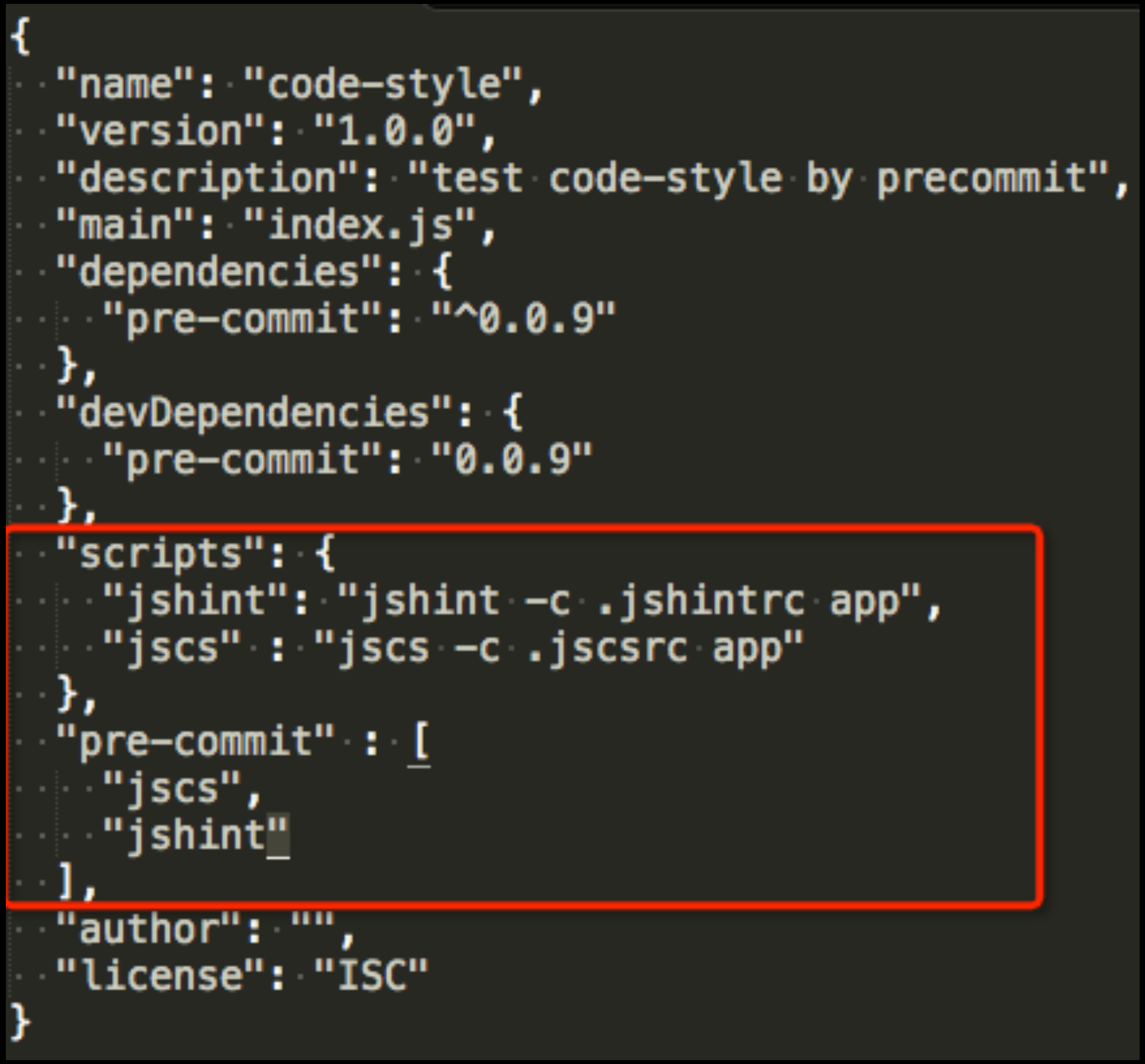
https://github.com/observing/pre-commit.git



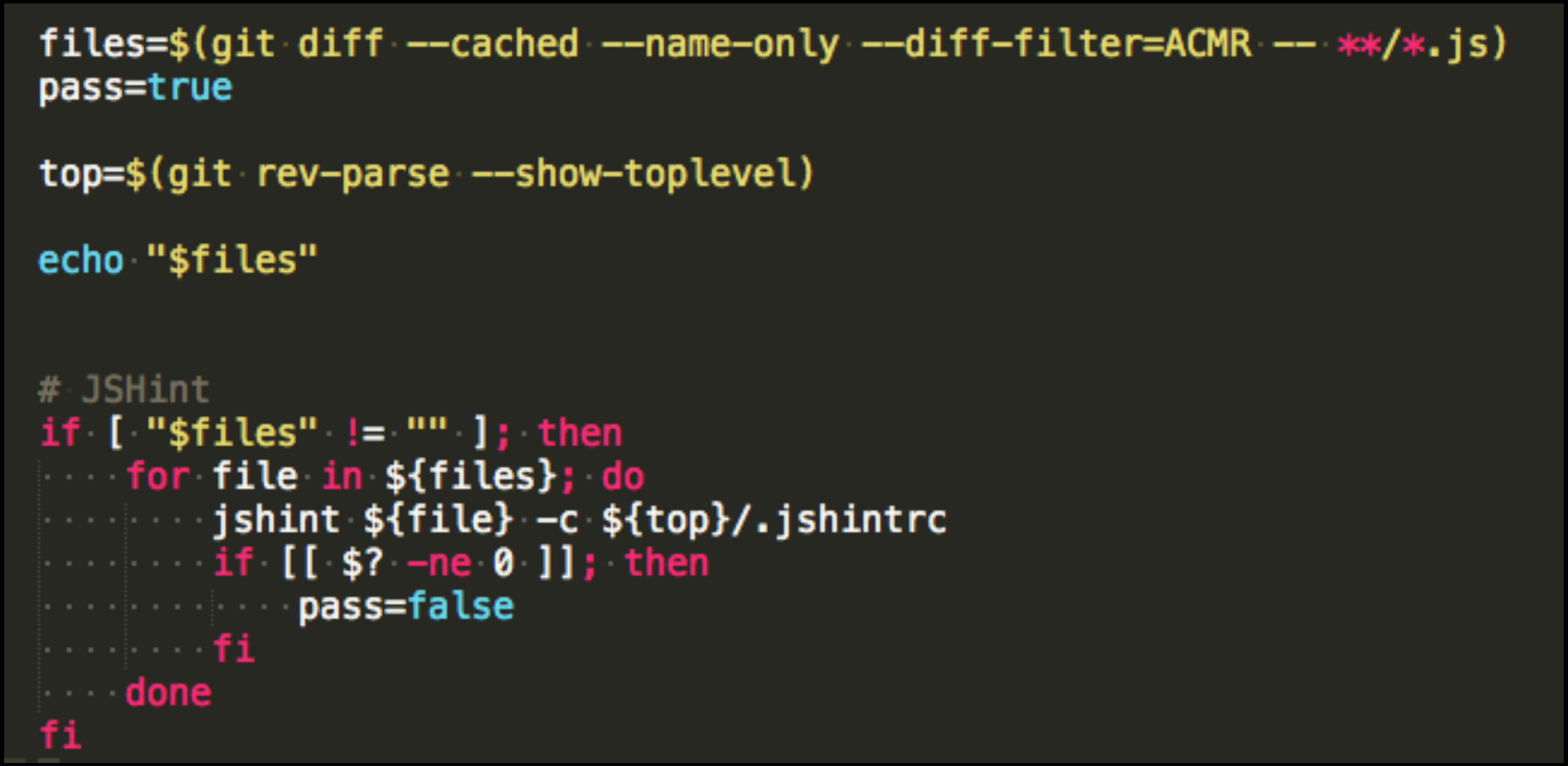
```
➜  temp
```

- npm postinstall hook强制安装pre-commit
- git pre-commit hook强制检查编码

- 代码一致性（jsHint)
- 代码风格 (jscs)
- 运行单元测试
- Code Coverage
- 性能和安全测试



```
{
  "name": "code-style",
  "version": "1.0.0",
  "description": "test code-style by precommit",
  "main": "index.js",
  "dependencies": {
    "pre-commit": "^0.0.9"
  },
  "devDependencies": {
    "pre-commit": "0.0.9"
  },
  "scripts": {
    "jshint": "jshint -c .jshintrc app",
    "jscs" : "jscs -c .jscsrc app"
  },
  "pre-commit" : [
    "jscs",
    "jshint"
  ],
  "author": "",
  "license": "ISC"
}
```



```
files=$(git diff --cached --name-only --diff-filter=ACMR -- **/*.js)
pass=true

top=$(git rev-parse --show-toplevel)

echo "$files"


# JSHint
if [ "$files" != "" ]; then
    for file in ${files}; do
        jshint ${file} -c ${top}/.jshintrc
        if [[ $? -ne 0 ]]; then
            pass=false
        fi
    done
fi
```

只检查当前修改的文件

已存在的项目

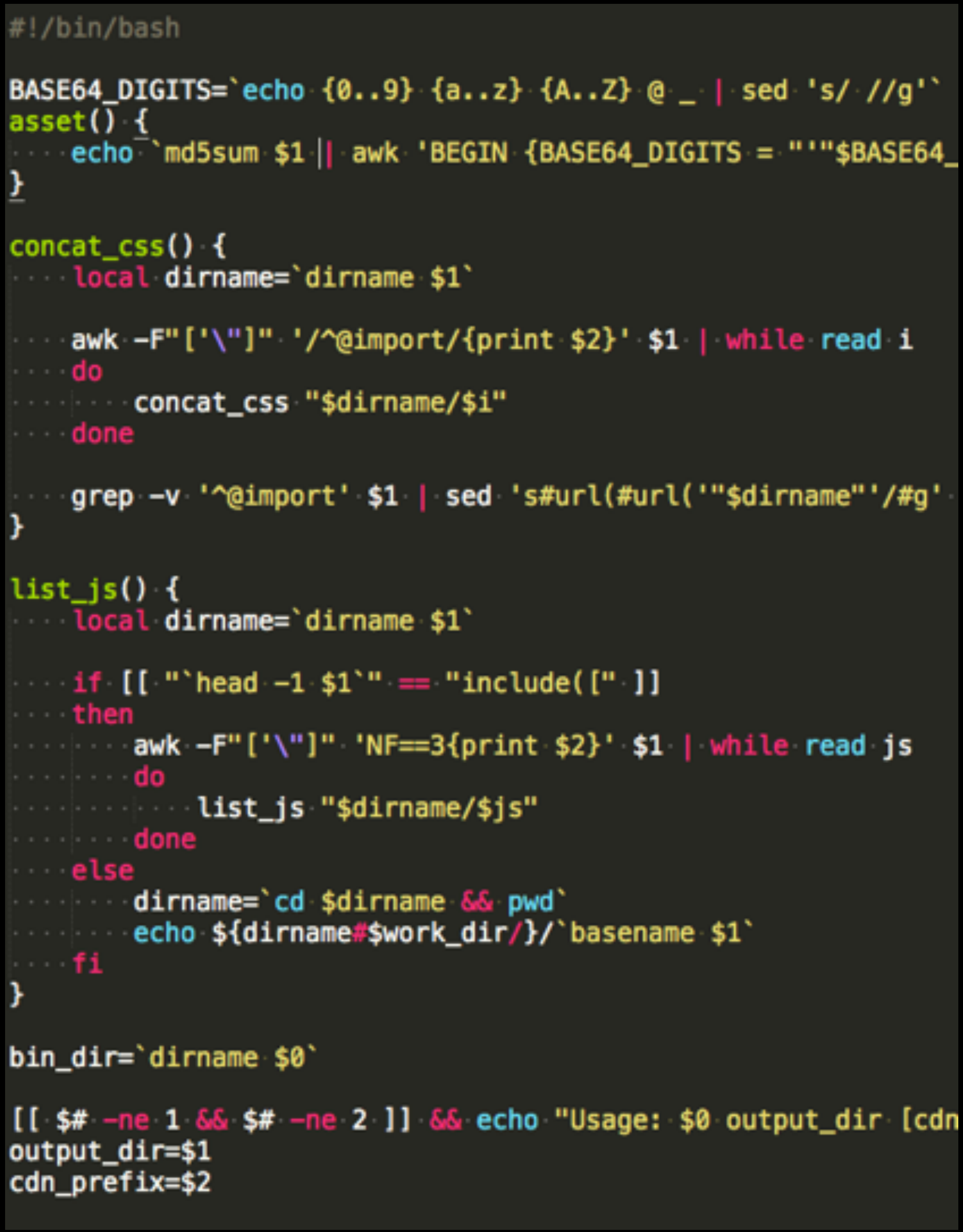
demo

https://github.com/cxdongjack/pre-commit-demo

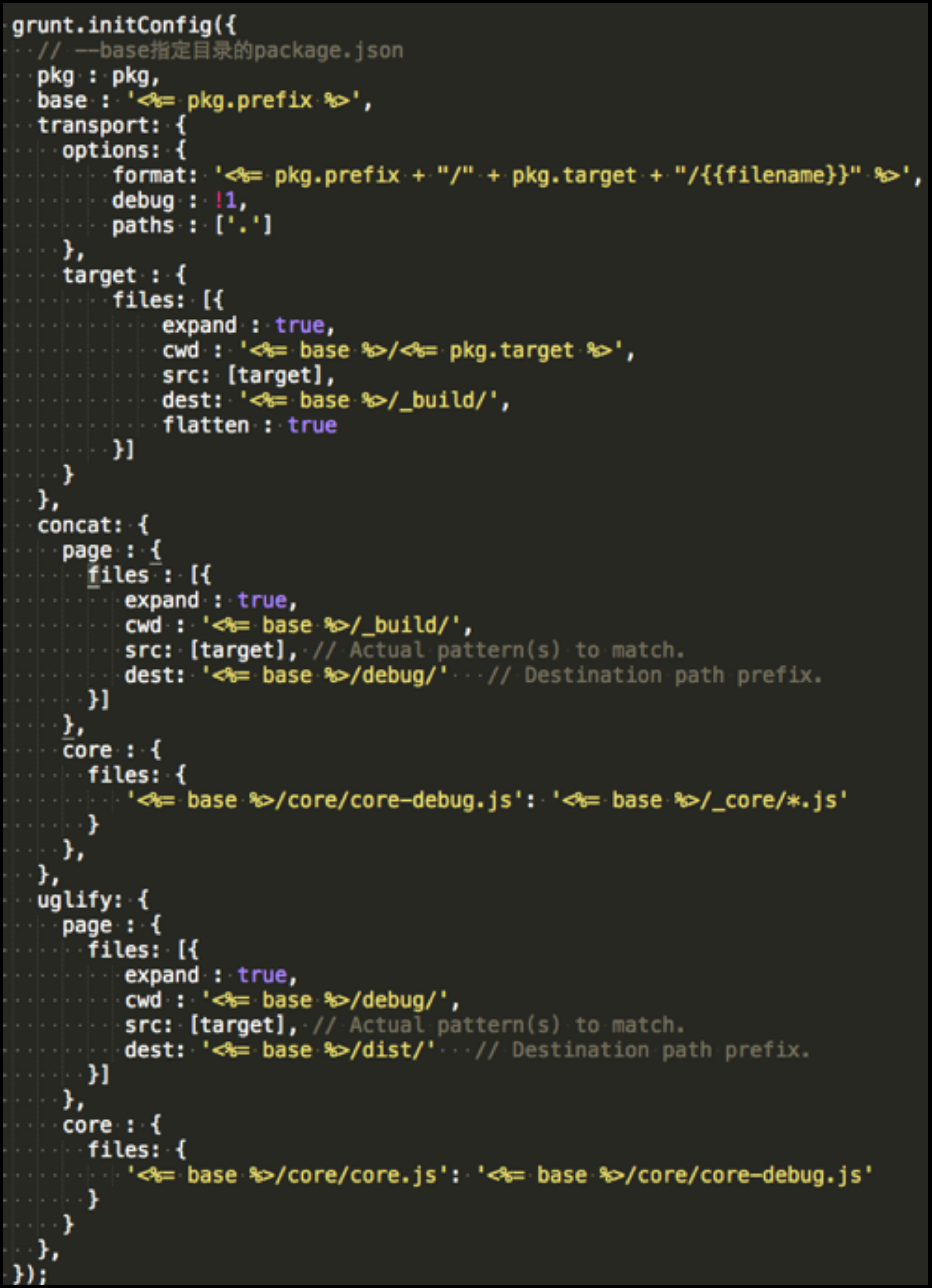
静态资源打包压缩?

bash script

开发成本
难复用



```bash
#!/bin/bash

BASE64_DIGITS=`echo {0..9} {a..z} {A..Z} @ _ | sed 's/ //g'`
asset() {
    echo `md5sum $1 || awk 'BEGIN {BASE64_DIGITS = "'"$BASE64_
}

concat_css() {
    local dirname=`dirname $1`

    awk -F"['\"]" '/^@import/{print $2}' $1 | while read i
    do
        concat_css "$dirname/$i"
    done

    grep -v '^@import' $1 | sed 's#url(#url('"$dirname"'/#g' 
}

list_js() {
    local dirname=`dirname $1`

    if [[ "`head -1 $1`" == "include([" ]]
    then
        awk -F"['\"]" 'NF==3{print $2}' $1 | while read js
        do
            list_js "$dirname/$js"
        done
    else
        dirname=`cd $dirname && pwd`
        echo ${dirname#$work_dir/}/`basename $1`
    fi
}

bin_dir=`dirname $0`

[[ $# -ne 1 && $# -ne 2 ]] && echo "Usage: $0 output_dir [cdn
output_dir=$1
cdn_prefix=$2
```



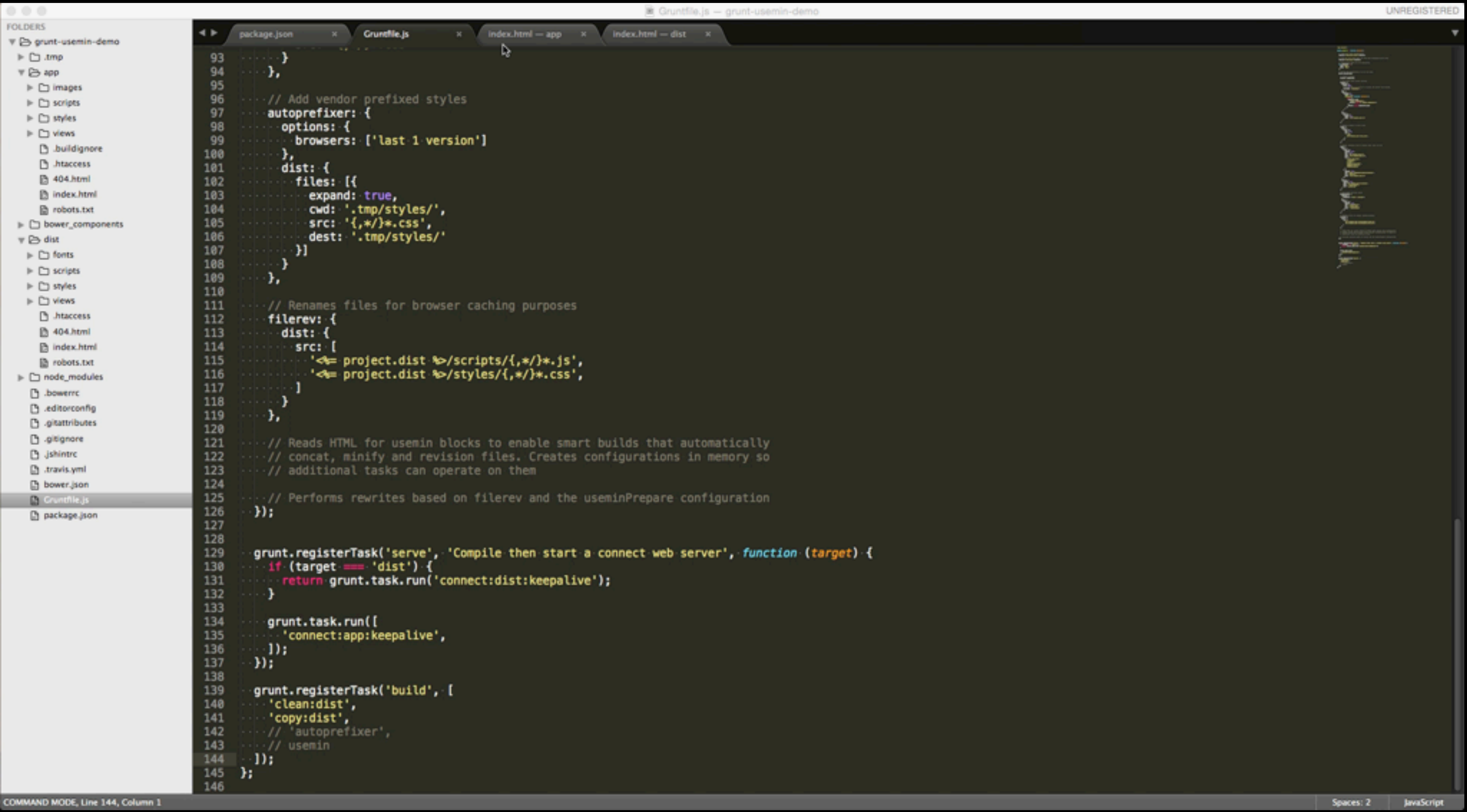
```
grunt.initConfig({
  // --base指定目录的package.json
  pkg : pkg,
  base : '<%= pkg.prefix %>',
  transport: {
    options: {
        format: '<%= pkg.prefix + "/" + pkg.target + "/{{filename}}" %>',
        debug : !1,
        paths : ['.']
    },
    target : {
        files: [{
            expand : true,
            cwd : '<%= base %>/<%= pkg.target %>',
            src: [target],
            dest: '<%= base %>/_build/',
            flatten : true
        }]
    }
  },
  concat: {
    page : {
      files : [{
         expand : true,
         cwd : '<%= base %>/_build/',
         src: [target], // Actual pattern(s) to match.
         dest: '<%= base %>/debug/'   // Destination path prefix.
      }]
    },
    core : {
      files: {
         '<%= base %>/core/core-debug.js': '<%= base %>/_core/*.js'
      }
    },
  },
  uglify: {
    page : {
      files: [{
         expand : true,
         cwd : '<%= base %>/debug/',
         src: [target], // Actual pattern(s) to match.
         dest: '<%= base %>/dist/'   // Destination path prefix.
      }]
    },
    core : {
      files: {
         '<%= base %>/core/core.js': '<%= base %>/core/core-debug.js'
      }
    }
  },
});
```

grunt

配置复杂

grunt-usemin

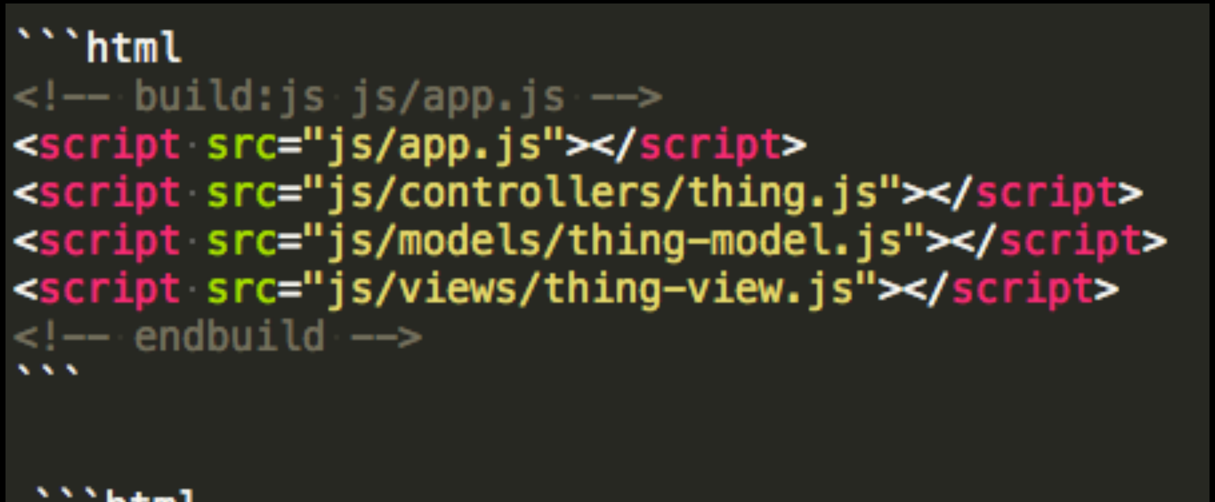
https://github.com/yeoman/grunt-usemin



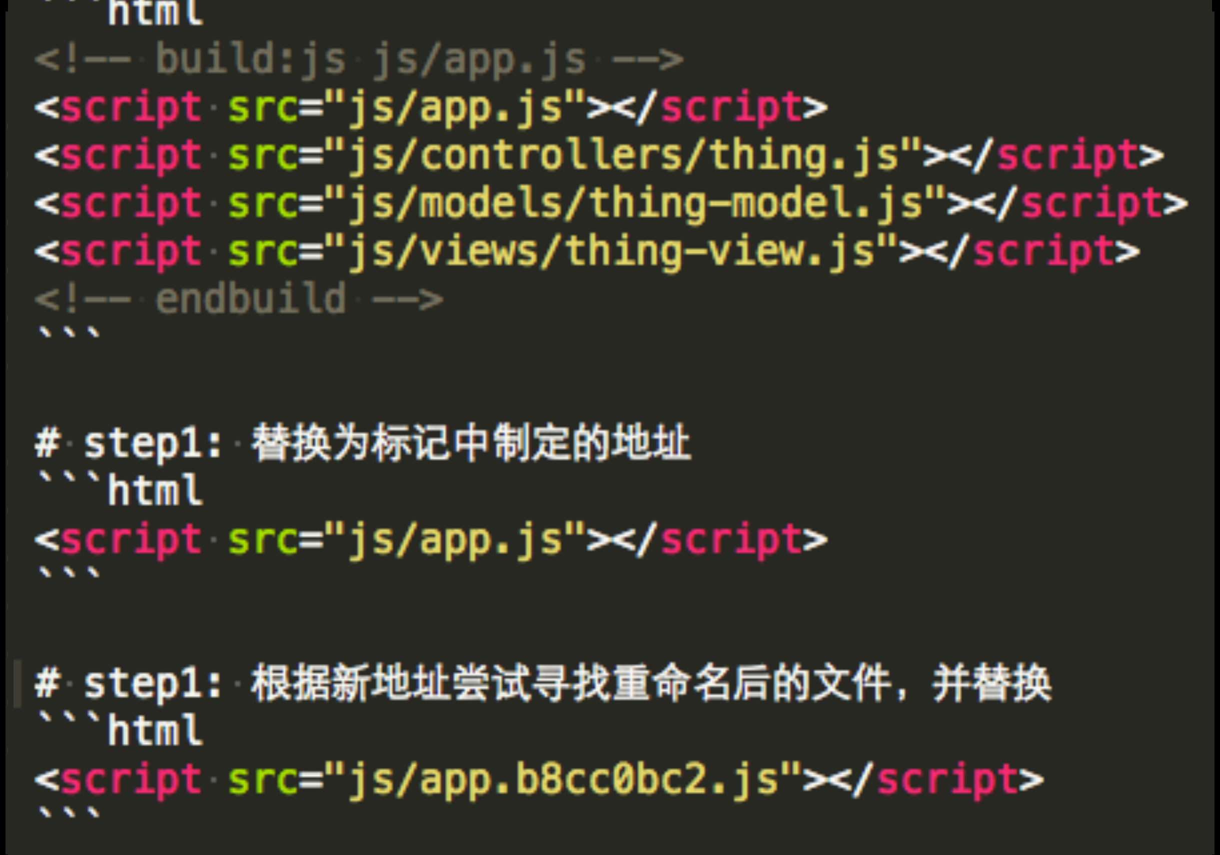
```
      }
    },

    // Add vendor prefixed styles
    autoprefixer: {
      options: {
        browsers: ['last 1 version']
      },
      dist: {
        files: [{
          expand: true,
          cwd: '.tmp/styles/',
          src: '{,*/}*.css',
          dest: '.tmp/styles/'
        }]
      }
    },

    // Renames files for browser caching purposes
    filerev: {
      dist: {
        src: [
          '<%= project.dist %>/scripts/{,*/}*.js',
          '<%= project.dist %>/styles/{,*/}*.css',
        ]
      }
    },

    // Reads HTML for usemin blocks to enable smart builds that automatically
    // concat, minify and revision files. Creates configurations in memory so
    // additional tasks can operate on them

    // Performs rewrites based on filerev and the useminPrepare configuration
  });


  grunt.registerTask('serve', 'Compile then start a connect web server', function (target) {
    if (target === 'dist') {
      return grunt.task.run('connect:dist:keepalive');
    }

    grunt.task.run([
      'connect:app:keepalive',
    ]);
  });

  grunt.registerTask('build', [
    'clean:dist',
    'copy:dist',
    // 'autoprefixer',
    // usemin
  ]);
};
```

配置

- useminPrepare, 根据html里面的标记信息自动为之后的任务生成配置信息
- usemin, 所有任务完成后重写html和css里面文件的引用地址

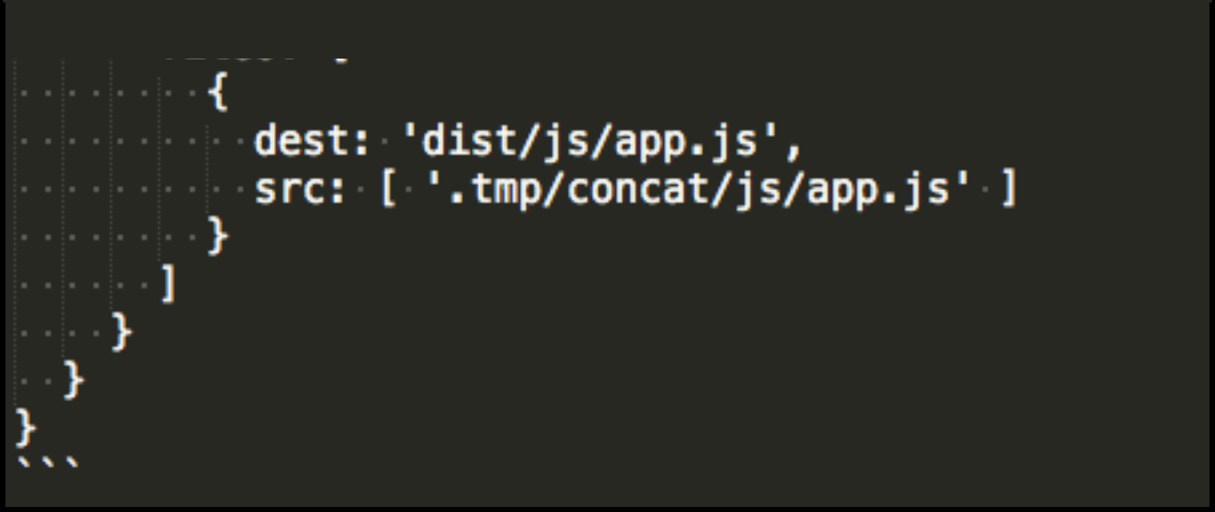


~~~
```html
<!-- build:js js/app.js -->
<script src="js/app.js"></script>
<script src="js/controllers/thing.js"></script>
<script src="js/models/thing-model.js"></script>
<script src="js/views/thing-view.js"></script>
<!-- endbuild -->
```
~~~



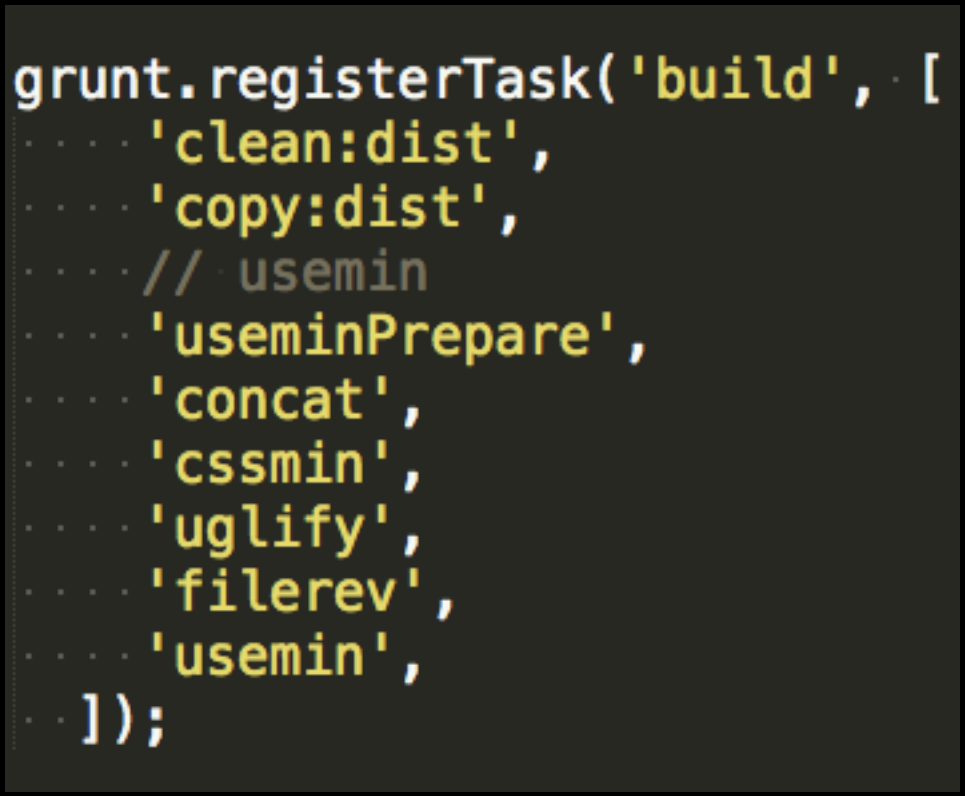
~~~
```html
<!-- build:js js/app.js -->
<script src="js/app.js"></script>
<script src="js/controllers/thing.js"></script>
<script src="js/models/thing-model.js"></script>
<script src="js/views/thing-view.js"></script>
<!-- endbuild -->
```
~~~

~~~
# step1: 替换为标记中制定的地址
```html
<script src="js/app.js"></script>
```


# step1: 根据新地址尝试寻找重命名后的文件，并替换
```html
<script src="js/app.b8cc0bc2.js"></script>
```
~~~



~~~
        {
          dest: 'dist/js/app.js',
          src: [ '.tmp/concat/js/app.js' ]
        }
      ]
    }
  }
}
```
~~~

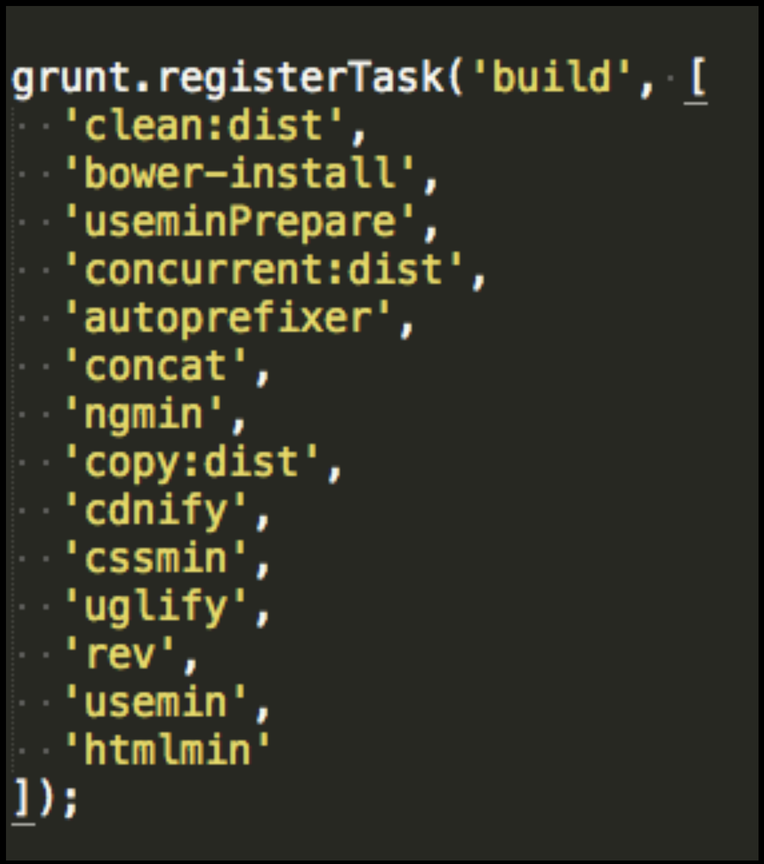
功能完备



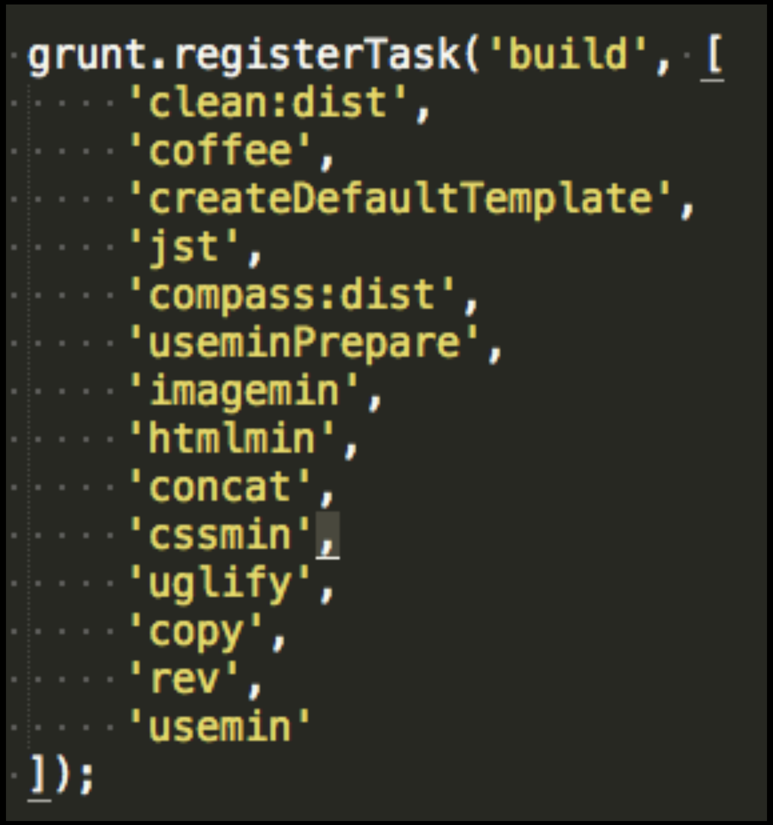
```
grunt.registerTask('build', [
    'clean:dist',
    'copy:dist',
    // usemin
    'useminPrepare',
    'concat',
    'cssmin',
    'uglify',
    'filerev',
    'usemin',
  ]);
```

- 根据标记解析html
- 合并资源文件
- 样式压缩
- 脚本压缩
- 文件hash重命名
- 自动修改引用至新文件

灵活扩展



```
grunt.registerTask('build', [
  'clean:dist',
  'bower-install',
  'useminPrepare',
  'concurrent:dist',
  'autoprefixer',
  'concat',
  'ngmin',
  'copy:dist',
  'cdnify',
  'cssmin',
  'uglify',
  'rev',
  'usemin',
  'htmlmin'
]);
```



```
grunt.registerTask('build', [
    'clean:dist',
    'coffee',
    'createDefaultTemplate',
    'jst',
    'compass:dist',
    'useminPrepare',
    'imagemin',
    'htmlmin',
    'concat',
    'cssmin',
    'uglify',
    'copy',
    'rev',
    'usemin'
]);
```

demo

https://github.com/cxdongjack/grunt-usemin-demo

```
process.exist(0)
```