# 1. 安装nodejs

Grunt和所有grunt插件都是基于nodejs来运行的，如果你的电脑上没有nodejs，就去安装吧。安装nodejs非常简单，完全傻瓜式、下一步下一步下一步、的安装方式，这里不再赘述。去 [https://nodejs.org/](https://nodejs.org/" \t "http://blog.csdn.net/wangfupeng1988/article/details/46418203/_blank) 上，

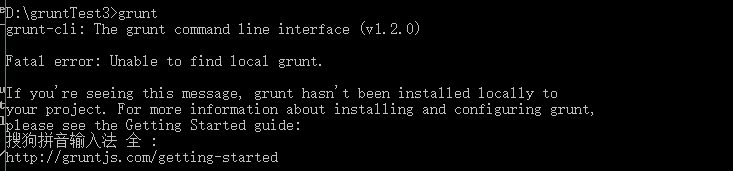
　安装了nodejs之后，可以在你的控制台中输入“node -v”来查看nodejs的版本，也顺便试验nodejs是否安装成功。



# 2. 安装grunt-CLI

IMG_256//全局安装;

　这时候要验证一下grunt-cli是否安装完成并生效，你只需要继续在命令行中输入“grunt”命令即可。如果生效，则会出现以下结果：



**3项目目录创建**

我在电脑的D盘下面建了一个“grunt\_test”文件夹，里面建了三个空文件夹、两个空文档，名称如下图。　　

　package.json这个文件写一些东西。记住，既然文件后缀名叫“json”，那么文件中的格式肯定是严格的json格式。Package.json的内容我们写成如下格式：



devDependencies”是什么意思？字面意思解释是“开发依赖项”，即我们现在这个系统，将会依赖于哪些工具来开发。

**5安装grunt**

你需要检查js语法错误，然后再去压缩js代码。如果这两步你都去手动操作，会耗费很多成本。Grunt就能让你省去这些手动操作的成本。

注意，这里安装grunt不再是全局安装了，需要你在控制台进入到网站或系统的具体目录下。这里我们进入 D:\grunt\_test 目录下。然后输入以下命令。

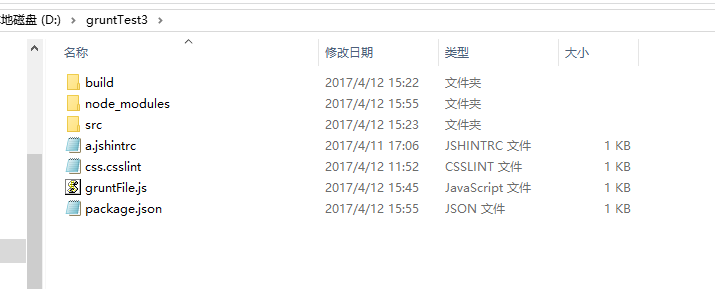
IMG_256

“—save-dev”的意思是，在当前目录安装grunt的同时，顺便把grunt保存为这个目录的开发依赖项

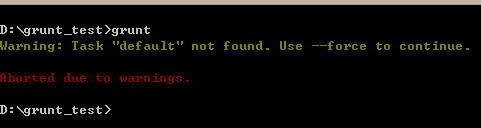
上文在配置package.json时，其中的“devDependencies”中就会存储开发依赖项。

安装完成后查看那里的“devDependencies”有什么变化。　　

　然后你再看看文档目录中的文件或者文件夹有什么变化？我这里多了一个“node\_modules”文件夹，其中有一个“grunt”文件夹，再其中有若干文档。这里就是存储grunt源文件的地方。



你在控制台运行“grunt”命令。如果你得到一个warning提示，那说明grunt已经起作用了。如下图：



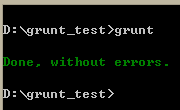
经过以上三步，说明grunt已经在这个目录下成功安装。

# 6. 配置Gruntfile.js

顾名思义，Gruntfile.js 这个文件，肯定是为了grunt做某种配置的。按照grunt的规定，我们首先把Gruntfile.js配置成如下格式。



在以上代码中，我们看到了刚才运行grunt命令，warning提示中的“default”字眼。不妨我们此时再运行一下grunt命令，看看会不会再次出现“warning”、“default”等字眼。



另外请注意Gruntfile.js中的一句话：

IMG_256

这句话是在Gruntfile.js中获取package.json中的内容。在上文配置package.json时我们说过：package.json中的内容不光是用来占位置的，还可以在其他地方获取。这里不是已经获取了package.json内容了吗？至于获取了如何使用，下文会有介绍，还是继续往下看吧。

# 7. Grunt插件介绍

grunt官网的插件列表页面 [http://www.gruntjs.net/plugins](http://www.gruntjs.net/plugins" \t "http://blog.csdn.net/wangfupeng1988/article/details/46418203/_blank)

插件分为两类。第一类是grunt团队贡献的插件，这些插件的名字前面都带有“contrib-”前缀，而且在插件列表中有星号标注。第二类是第三方提供的插件，不带有这两个特征。

Contrib-jshint——javascript语法错误检查；

Contrib-watch——实时监控文件变化、调用相应的任务重新执行；

Contrib-clean——清空文件、文件夹；

Contrib-uglify——压缩javascript代码

Contrib-copy——复制文件、文件夹

Contrib-concat——合并多个文件的代码到一个文件中

karma——前端自动化测试工具

# 8. 使用uglify插件（压缩javascript代码）

　安装uglify插件的方式，和安装grunt是一样的。还记得grunt是怎么安装的吗？

IMG_256

我们既然要使用uglify来压缩javascript代码；

当然，要压缩谁？往哪里压缩？这些都需要配置，在Gruntfile.js中配置。分为三步：

第一步，在grunt.initConfig方法中配置 uglify 的配置参数。如下图：

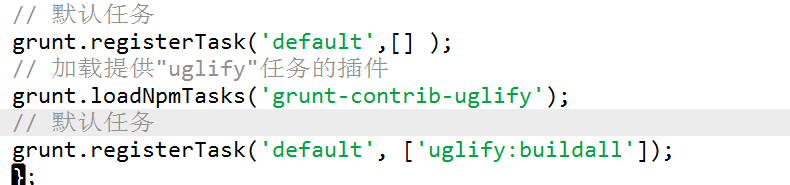


　　上图中，对uglify的配置有两项。

　　“options”中规定允许生成的压缩文件带banner，即在生成的压缩文件第一行加一句话说明。注意，其中使用到了pkg获取package.json的内容。

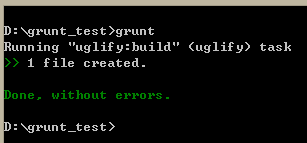
　　“build”中配置了源文件和目标文件。即规定了要压缩谁？压缩之后会生成谁？　　（PS：上文中说过的package.json的内容终于找到了他被应用的地方了。这样的好处是：例如，对文件版本的管理，你只需要在package.json中修改即可，grunt会自动根据最新的版本号生成相应版本的文件。你不用手动去修改文件的文件名。）

最后，这里只是对“options”和“build”的基本应用，还有许多中使用方式，可以去官网查阅。

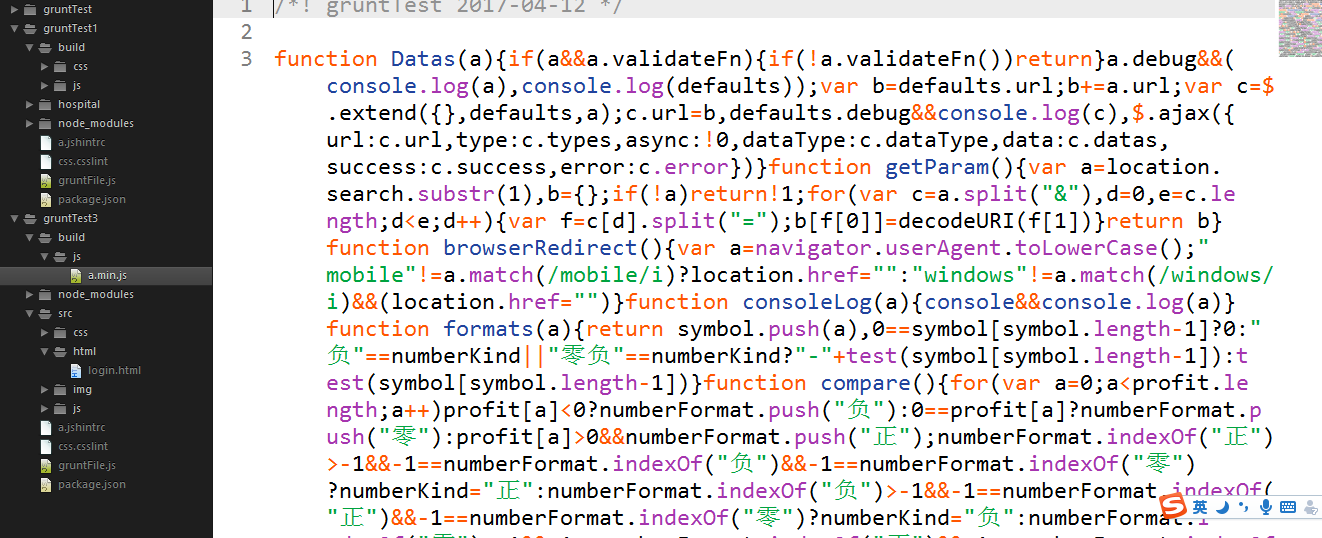
第二步，在 grunt.initConfig 方法之后，要让grunt去加载这一个插件。　　

在grunt命令执行时，要不要立即执行uglify插件？如果要，就写上，否则不写。以上说的这三步已经OK了，接下来我们去试试。在控制台中运行grunt命令，看得到什么结果。

控制台将输入如下信息：



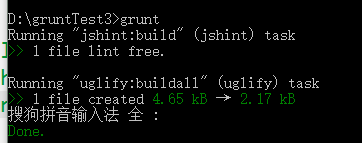
　　再去看看，是否生成了一个压缩后的js文件？



　　压缩后的代码的banner也是符合Gruntfile.js中的配置要求的。

　　 以上就是uglify插件的详细安装、配置说明。Javascript使用uglify压缩，css可使用cssmin插件压缩。安装、配置方法一样的，不再赘述。

**9.jshint安装步骤同上，结果如下:**



**10 .cssmin安装同上**

