# 先了解一下nodejs

Nodejs不是一种独立的语言，也不是一个javascript框架， 也不是流览器端的js库。

Nodejs是一个让javascript能够运行在服务器端的开发平台。它可以让js脱离流览器的束缚运行在一般的服务器环境下。历史上把javascript移值到流览器外的计划不止一个，目前，nodejs是最出色的一个。

Nodejs内建了HTTP服务器的支持。

## 安装nodejs

一路下一步即可

查看安装是否成功： 命令提示符下 -> node -v

## Node的包管理器npm

Npm是一个由nodejs官方提供的第三方包管理器。它是一个完全由javascript实现的命令行工具。从node0.6版本开始，npm已经包含在node的发行包中了，不需要额外安装。

开发者使用场景： 开发人员需要从npm的服务器上下载别人编写的第三方包到本地使用时

如何下载(安装): npm install 包名 （会在当前目录下产生一个node\_modules的文件夹，其下载的包都在放在这个目录下）

如何删除: npm uninstall 包名

Npm或cnpm安装插件或包时，有两种模式： 全局模式和本地模式

全局模式： npm install gulp -g （装好以后可在命令行中使用gulp）

node不是一种独立的语言，依托了js语法，也不是一个JavaScript框架，也不是浏览器端的js库，node.js是一个让JavaScript能够运行在服务器端的开发平台。历史上把JavaScript移植到浏览器外的计划不止一个，目前node.js是最出色的一个。

npm是包管理器，依托于node。

npm是一个由node.js官方提供的第三方包管理器，他是一个完全由JavaScript实现的命令行工具。从node0.6开始npm已经包含在node发行包中了，不需要额外安装。

开发者使用场景：开发者需要从npm的服务器上下载别人编写的第三方包到本地使用时。

如何下载安装：npm install 包名

如何卸载删除：npm uninstall 包名

npm或cnpm安装插件或包时，有两种模式：全局模式/本地模式

全局模式：npm install gulp -g 安装在c盘的默认路径。装好后可在命令行直接使用

本地模式：npm install gulp 安装在所在路径，本地安装好的只可通过require（“”）方式调用。

定义default任务

当执行gulp时，会自动执行的任务

gulp.task("default",["jsmin"]);

开发前端需要自己模拟数据，接口是不存在的，自己根据数据类型模拟实现。

# 项目中操作步骤

准备工作：

Nodejs安装

Cnpm安装

全局安装gulp （为了使用gulp命令）

1. 创建package.json文件 ，如何创建 ？

项目根目录下： cnpm init （根据提示填写或直接回车）

1. 本地安装gulp （为了使用gulp模块）

cnpm install gulp --save-dev

（--save-dev这个参数在本地安装的同时，会自动更新package.json文件，使当前安装的包及其版本信息写入到package.json文件的devDependencies属性下)

1. 本地安装gulp插件 （gulp-uglify 用于压缩js的插件）

cnpm install gulp-uglify --save-dev

1. 在项目根目录下创建文件 gulpfile.js

内容可参与手册

1. 用gulp命令运行任务

第三方gulp插件

gulp-uglify压缩JS

gulp-clean-css压缩CSS

gulp-htmlmin压缩HTML

gulp-concat合并相同文件

gulp-rename对文件更名

还有更多插件，插件中有更多的属性控制，可以查询手册。

使用实例：

**var** gulp = require('gulp'),首先对每一个插件进行引入  
 uglify = require('gulp-uglify'),  
 rename = require('gulp-rename'),  
 concat=require('gulp-concat'),  
 cssmin=require('gulp-clean-css'),  
 htmlmin=require('gulp-htmlmin');  
 // js压缩改名合并  
gulp.task('jsmin', **function** () {声明使用方法jsmin  
 gulp.src('js/\*.js')更改的路径，从工程根目录开始  
 .pipe(concat('all.js'))合并  
 .pipe(gulp.dest('minJs'))输出  
 .pipe(uglify())压缩  
 .pipe(rename({更名  
 suffix: ".min" 添加后缀前的标识  
 })) 还有更多的功能  
 .pipe(gulp.dest("minJs"))最后输出  
});  
// css文件压缩合并改名  
gulp.task('cssmin',**function** () {  
 gulp.src('css/\*.css')  
 .pipe(concat('all.css'))  
 .pipe(gulp.dest('minCss'))  
 .pipe(cssmin())  
 .pipe(rename({  
 suffix:'.min'  
 }))  
 .pipe(gulp.dest("minCss"))  
});  
//html文件压缩，改名  
gulp.task('htmlmin',**function** () {  
 gulp.src('html/\*.html')  
 .pipe(htmlmin({  
 collapseWhitespace:**true 压缩html** }))  
 .pipe(rename({  
 suffix:'.min'  
 }))  
});

browser-sync实时更新网页。