[npm- 开发与生产环境](https://segmentfault.com/a/1190000010675594)

**npm install --production**

https://segmentfault.com/a/1190000010675594

npm

npm 是javascript 包管理，也是世界上最大的软件记录。

使用npm 可以安装，分享和贡献自己的代码，管理项目中的依赖包，

package.json

npm的包依赖配置json文件，开发时可以在里面记录所有依赖包，也可以配置一些简单的npm 脚本，

{

"name": "xxxxxx",

"version": "11111",

"description": "xxxxxxx",

"author": xxxxx",

"private": true,

"scripts": {

"dev": "node server.js",

"start": "node server.js",

"build": "node build.js "

},

"dependencies": {

},

"devDependencies": {

"webpack": "^2.6.1",

}

}

上面是一个简单的package.json 文件的，里面有一些基本的配置项，主要有作用的是 scripts,dependencies,devDependencies 三类属性。

npm scripts

//在scripts 字段中定义一些简单的脚本，比如上面定义了三个脚本，执行时可以这样：

npm **run** dev 相当于执行

node server.js

// 查看当前所有Npm脚本命令

npm **run**

\*几种常见的简写形式\*

npm start 相当于 npm **run** start

npm stop 相当于 npm **run** stop

npm test 相当于 npm **run** test

npm restart 相当于 npm **run** stop && npm run restart && npm run start

**原理：**

npm run 新建一个shell,并且把当前目录的 node——modules/.bin  
子目录加入PATH变量，执行结束后，再将PATH 变量恢复原样

[npm scripts 详细解释](http://www.ruanyifeng.com/blog/2016/10/npm_scripts.html)

dependencies、devDependencies

npm **install** Xxx --save 安装模块，并把模块名和版本号添加到dependencies 部分。

npm **install** xxx --save-dev 安装模块,把模块名和版本号写在devdependencies部分。

**区别就是，在生产环境，使用npm install --production安装 dependencies 部分的模块,**  
**在开发环境，npm i ，安装所有devDependencies 和 dependencies里面的模块**

所以，在生产环境，就不需要安装 gulp ,webpack 这样的模块，可以通过这种方法区分

**几种简写指令：**

-P 相当于 --save-prod， 添加dependencies 里面所有的包。在 -D -O 不存在时，-P 就是默认值

-S 相当于 --save; 添加dependencies 里面所有的包。

-D 相当于 --save-dev; 添加devDependencies 里面所有的包，

-O 相当于 --save-optional, 添加在 optionalDependencies 里面的包，

--no-save： 阻止保存记录在dependencies 中，

合理使用这两个依赖记录，可以保证减少在生产环境下，打包后体积大小。

传递参数

上门写到 使用npm run等命令可以执行相应的脚本，同样也可以在其中进行参数传递，跟在[Shell中通过process.env属性来传递参数](http://javascript.ruanyifeng.com/nodejs/process.html#toc5)进行参数的传递的方法基本一样。

通常的方法是，在环境变量**NODE\_ENV**中传递参数，使用它来确定当前所处的开发阶段，生产阶段设为production,开发阶段设为develop 或者staging,然后在脚本中读出 process.env.NODE\_ENV即可

NODE\_ENV = production npm **run** dev