

## 初始化npm设置

配置npm全局模块存放路径以及cache路径

```
$ npm config set prefix "D:\nvm\node_global"
```

```
$ npm config set cache "D:\nvm\node_cache"
```

## 升级

```
$ npm install npm@next -g # 升级到未发行版本
```

```
$ npm install npm@latest -g # 升级到最新版本
```

## 安装

# 普通安装

```
$ npm install <packagename>
```

# To add an entry to your package.json's dependencies:

```
$ npm install <packagename> --save
```

# To add an entry to your package.json's devDependencies:

```
$ npm install <packagename> --save-dev
```

# 安装全局包

```
$ npm install -g <packagename>
```

## 更新

# 项目模块

```
$ npm update
```

# 全局包

```
$ npm outdated
```

# 全局指定包

```
$ npm update -g <package>
```

## 卸载

```
# 普通卸载
$ npm uninstall <packagename>
# 卸载依赖
$ npm uninstall <packagename> --save
# 全局卸载
$ npm uninstall <packagename> -g
```

### npm package.json 版本前缀符号说明

- “^” 是指匹配全部4. x. x的版本，一直到5. 0. 0（不包括）。
- “~” 是指匹配全部2. 4. x的版本，一直到2. 5. 0（不包括）。

### 常用NPM指令

指令	说明
npm config	配置npm
npm init	初始化npm
npm search	查找模块
npm install	安装模块
npm uninstall	卸载模块
npm list	勘察安装模块
npm outdated	检查模块是否过时
npm update	更新模块
npm cache [clean ls]	管理模块缓存

### 清除缓存

```
$ npm cache verify
$ npm cache clean --force
```

- 使用`npm install xxx`命令安装模块时, 不再需要`--save`选项, 会自动将模块依赖信息保存到 `package.json` 文件;
- 安装模块操作 (改变 `node_modules` 文件夹内容) 会生成或更新 `package-lock.json` 文件
- 发布的模块不会包含 `package-lock.json` 文件
- 如果手动修改了 `package.json` 文件中已有模块的版本, 直接执行`npm install`不会安装新指定的版本, 只能通过`npm install xxx@yy`更新
- 5.2 支持`npx`, `npx`可临时安装一个包作为命令行运行