**npm工具**：node package manger

npm是node的模块管理工具 我们可以通过npm快速的安装和卸载我们所需要的资源文件（例如：jquery，vue，vue-router等等）

node官网：<https://nodejs.org/zh-cn/>

npm管理模块官网：<https://www.npmjs.com/>

**安装：**

去node官网下载安装node之后， npm也安装好了

测试 ：

node -v 和npm -v 查看版本号 表示安装好了

**基于npm进行模块管理：**

$ npm install xxx 把模块安装在当前项目中（node\_modules）最新版本会生成package-lock.json 记录文件 下次npm install就会把当前记录模块全装上

$npm install xxx -g 把模块安装在全局环境中

$npm i [xxx@1.0.0](mailto:xxx@1.0.0) 安装制定版本号的模板install 的缩写是i

$npm view xxx versions > xxx.version.json 查看某个模块的版本信息（输出到指定的json文件中）

$npm init -y 初始化当前项目的配置依赖清单 生成package.json文件（项目目录文件名字不能出现中文，大写字母和特殊符号）

$npm i xxx --save 把模块保存在清单生产依赖中

$ npm i xxx --save-dev 把模块保存在清单开发依赖中；比如有些工具只需要在开发环境下使用的比如：webpack 就使用这命令 就会在package.json的devdependencies中的就有了”webpack”:”版本号”

$npm install 跑环境，按照清单安装所需的模块

$npm root -g 查看全局安装模块的目录

$npm uninstall xxx

$npm uninstall xxx -g 卸载安装过的模块

-S就是--save的简写，就行npm默认一个start的字段，你可以不必输入npm run start 而只需输入npm start，这两个效果是一样的。  
-D就是--save-dev 这样安装的包的名称及版本号就会存在package.json的devDependencies这个里面，而--save会将包的名称及版本号放在dependencies里面。**实例：**

**引入less文件到全局，并且 创建1.less文件，后编译1.less成1.min.css使用**

引入less到全局：

打开cmd命令窗口 输入npm install less –g

然后 npm root -g 查看全局文件

显示 C:\Users\Administration\AppData\Roaming\npm\node\_modules

在npm可以看到.cmd为后缀的文件 那些文件就是用来编译的cmd文件

进入到1.less文件的目录 打开命令窗口输入lessc 1.less 1.min.css –x编译1.less成1.min.css使用 然后在页面中引入1.min.css 就可以了

**什么时候安装到全局**

安装到全局的好处就是可以使用命名操作任何项目

使用npm root-g查看全局安装的目录

因为安装目录下生成了.cmd的文件，所以我们能够使用xxx命令进行操作

**什么时候安装到本地项目模块**

Cmd命令进入到当前目录才能操作当前的模块和操作

可以在项目中导入进来使用

将xxx安装在本地默认是不能基于命令操作的，因为没有生成（xxx.cmd文件）

在项目目录下$npm init -y 初始化当前项目的配置依赖清单 生成package.json文件模块下的dependencies：开发依赖模块（开发和项目部署的时候需要）devdependencies：开发依赖模块（只有开发的时候需要）

Scripts :配置本地可执行命令的 比如：配置”less”:”lessc 1.less 1.min.css -x”后执行cmd 命令npm run less 等于执行了后面的命令 就不需要安装到全局才能只能够命令了

**项目：**

**创建一个新项目，初始化项目清单，安装所有需要的模块（vue，jquery等），然后可能需要本地配置（**Scripts**）执行命令去编译一些文件，只要有package.json文件就能通过执行 npm把里头所有模块或者缺少的模块安装下来到当前目录下**