title: Npm Package
date: 2019-08-26
tag: node

description: Npm Package

npm 依赖包

版本号

`npm` 采用了 `semver` 规范作为版本管理。

semver 规范：一个 `npm` 依赖包的版本格式为：**主版本号.次版本号.修订号**

- **主版本号**（`Major Version`）颠覆性改动，可能存在与低版本`API`不兼容的情况。
- **次版本号**（`Minor Version`）小版本改动，`API` 兼容（无感更新）。
- **修订号**（`Path`，也叫补丁）`Bug` 修复或者细微改动，向前兼容。

特殊符号的意义：

- **“x.y.z”** 精确版本号；
- **"^x.y.z"** 兼容补丁、小版本改动。如 `"3.2.1"` 等价 `">= 3.2.1 < 4.0.0"`；
  - **"~x"** 兼容补丁、小版本改动（范围更宽）。如 `"~3"` 等价 `">= 3.0.0 < 4.0.0"`；
- **"~x.y.z"** 兼容补丁改动，如 `"~3.2.1"` 等价 `">=3.2.1 < 3.3.0"`
  - **"~x.y"** 兼容补丁改动（范围更宽），如 `"~3.2"` 等价 `">=3.2.0 < 3.3.0"`
- **通配符** "x.y"、"x.*"、"x"、"*"
- **"x.y.z-beta.w"** 预发布版本号，有三个关键词定义：
  - `alpha` 预览版（内部测试）
  - `beta` 测试版（公开测试）
  - `rc(release candidate)` 最终测试版本（候选版本）
- **其他**
  - `a-b` 范围
  - `>=`、`<=`、`||`

PATH 环境变量

每次执行 `npm run xxx` 时，会自动建一个 `Shell`，会将当前项目的 `node_modules/.bin` 的绝对路径加入到环境变量 `PATH` 中，执行结束，将环境变量 `PATH` 恢复。

Process.argv

```
{
  "scripts": {
    'scp': "cli xxx"
  }
}
```

`process.argv` 返回一个数组

- 第一个元素是启动 `node` 进程的可执行文件的绝对路径 process.execPath
- 第二个是当前执行 js 的文件路径
- 其他的是命令参数
  比如执行 `npm run scp`,打印 `process.argv`：

```
[
  '/usr/local/Cellar/node/7.7.1_1/bin/node',
  '/Users/mac/Vue-projects/hao-cli/node_modules/.bin/cli',
  'xxx'
]
```

常用命令

```
# 查看当前环境变量
npm run env
# 父包依赖
npm ls --depth=0
# 查看某个模块版本信息
npm view/info babel (version/versions)
```