

## JS不足

- js没有 **模块** 系统，不支持 **封闭作用域** 或 **依赖管理**
- 没有 **标准** 库，没有文件系统API
- 没有 **包** 管理系统，不能自动加载和安装依赖

## commonjs规范

- 一个 node.js由大量模块组成，每个JS文件都是一个模块
- 实现了 `require` 方法，npm基于 `commonjs` 实现了自动加载和安装依赖

## 模块和包的优点

- 增加 **内聚** 性，有助分工协作
- 方便 **重构**
- 提高代码 **质量**

## 模块使用

- 定义模块
- 导出模块
- 使用模块

# 导出类和对象

## 模块分类

- 内置模块
- 文件模块
- 第三方模块

## 安装模块

- 全局安装 直接下载到Node的安装目录中，各个项目都可以调用,适合工具模块，比如 `gulp`

```
npm install -global [package name]
```

- 本地安装 将一个模块下载到当前目录的 `node_modules` 子目录，然后只有在当前目录和它的子目录之中，才能调用这个模块

```
npm install [package name]
```

## 项目管理

- 初始化项目
- 安装第三方模块
- 注册用户
- 发布项目

“ 注意项目的名称不能是别人已经注册的名称