**NodeJS模块机制**

**CommonJS规范**：模块引用，模块定义，模块标识

在**NodeJS中引入模块**，需要经过3个步骤：路径分析，扩展名分析，编译执行

**NodeJs的模块**分为两种，一种是**原生（核心）模块**（Node提供的模块，已经是编译后的二进制文件，部分核心模块直接加载进内存，在步骤一中优先执行，2,3,步可以省略，所以加载速度最快），一种是**文件模块**（用户编写的模块，运行时动态加载，需要按照上面的步骤执行，所以速度比核心模块慢），但只要引用过的模板都会缓存起来，而且缓存之后的都是编译和执行之后的对象。

**路径的加载顺序**：缓存加载 > 核心模块 > 路径形式文件模块（.、..和./开头的标识符） > 自定义文件模块（第三方npm包，比如express，eslint等）

**文件定位**（路径分析+扩展名分析）

1、自定义文件模块的查找规则：

当前文件目录下的node\_modules目录

父目录下的node\_modules目录

向上逐级递归至根目录下node\_modules目录

2、扩展名分析：

在引入文件不加扩展名的时候，会按.js，.json，.node的次序补足扩展名，依次尝试，尝试时需要调用fs模块同步阻塞地判断文件是否存在。

3、require通过文件扩展名之后没有找到对应的文件，但得到一个目录，NodeJS则会把他当做一个包处理。

Node也一定程度上遵循了CommonJS规范，过程如下：

1. Node会在当前目录下查找package.json文件（JSON.parse解析），查找main字段指定的文件
2. 第一步不成功则会一次查找index.js、index.json、index.node
3. 遍历下一个模块路径（即父目录）还是没有则抛出查找失败的异常