

直至几年前，JavaScript 的种种优势才被重新提起，JavaScript 又具备了在服务端流行的条件，Node.js 应运而生。与此同时，RingoJS 也基于 Rhino 实现了类似的服务端 JavaScript 平台，还有像 CouchDB、MongoDB 等新型非关系型数据库也开始用 JavaScript 和 JSON 作为其数据操纵语言，基于 JavaScript 的服务端实现开始遍地开花。

1.6.2 CommonJS 规范与实现

正如当年为了统一 JavaScript 语言标准，人们制定了 ECMAScript 规范一样，如今为了统一 JavaScript 在浏览器之外的实现，CommonJS 诞生了。CommonJS 试图定义一套普通应用程序使用的 API，从而填补 JavaScript 标准库过于简单的不足。CommonJS 的终极目标是制定一个像 C++ 标准库一样的规范，使得基于 CommonJS API 的应用程序可以在不同的环境下运行，就像用 C++ 编写的应用程序可以使用不同的编译器和运行时函数库一样。为了保持中立，CommonJS 不参与标准库实现，其实现交给像 Node.js 之类的项目来完成。图 1-5 是 CommonJS 的各种实现。



图1-5 CommonJS 的实现

CommonJS 规范包括了模块 (modules)、包 (packages)、系统 (system)、二进制 (binary)、控制台 (console)、编码 (encodings)、文件系统 (filesystems)、套接字 (sockets)、单元测试 (unit testing) 等部分。目前大部分标准都在拟定和讨论之中，已经发布的标准有 Modules/1.0、Modules/1.1、Modules/1.1.1、Packages/1.0、System/1.0。