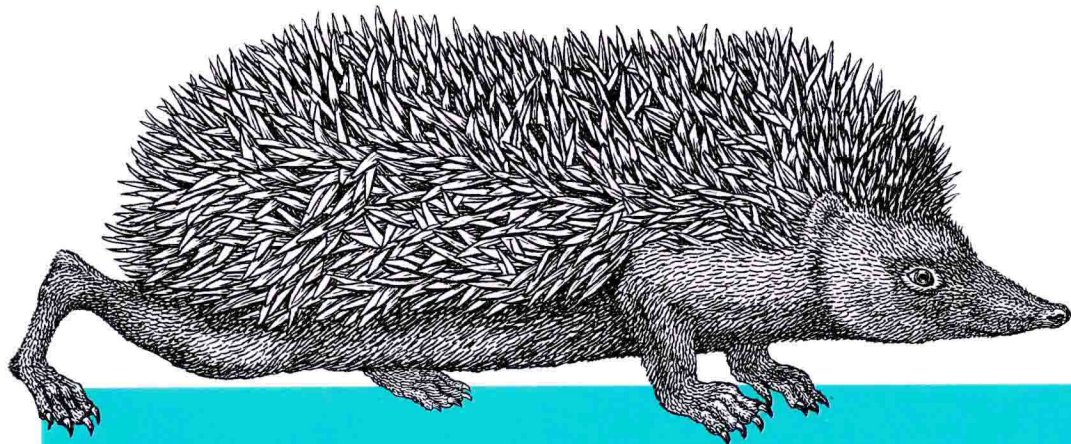


O'REILLY®

TURING

图灵程序设计丛书



数据结构与算法 JavaScript描述

Data Structures & Algorithms with JavaScript

[美] Michael McMillan 著
王群锋 杜欢 译

 人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

数据结构与算法JavaScript描述

Data Structures and Algorithms with JavaScript

[美] Michael McMillan 著

王群锋 杜欢 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社

图书在版编目 (C I P) 数据

数据结构与算法JavaScript描述 / (美) 麦克米伦 (McMillan, M.) 著; 王群锋, 杜欢译. — 北京: 人民邮电出版社, 2014.9

(图灵程序设计丛书)

ISBN 978-7-115-36339-8

I. ①数… II. ①麦… ②王… ③杜… III. ①数据结构②算法分析③JAVA语言—程序设计 IV. ①TP311.12
②TP312

中国版本图书馆CIP数据核字(2014)第150725号

内 容 提 要

在过去几年中, JavaScript 凭借 Node.js 和 SpiderMonkey 等平台, 在服务器端编程中得到了广泛应用。JavaScript 程序员因而迫切需要使用传统语言(比如 C++ 和 Java)提供的工具, 包括传统的数据结构以及传统的排序和查找算法。本书讨论在数组即对象、无处不在的全局变量、基于原型的对象模型等 JavaScript 语言的环境下, 如何实现高效的数据结构和算法。

本书适合 JavaScript 程序员以及对 JavaScript 语言感兴趣的学习者, 特别是在学校中没有系统学习过计算机科学相关课程的“跨界”程序员。

-
- ◆ 著 [美] Michael McMillan
译 王群锋 杜欢
责任编辑 李松峰
执行编辑 李静 刘帅
责任印制 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 13.5
字数: 278千字 2014年9月第1版
印数: 1-3 500册 2014年9月河北第1次印刷
著作权合同登记号 图字: 01-2014-4384号
-

定价: 49.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版权声明

© 2014 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2014. Authorized translation of the English edition, 2014 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版，2014。

简体中文版由人民邮电出版社出版，2014。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，不得以任何形式复制本书的部分或全部内容。

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——*Wired*

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

推荐序

在前端工程师中，常常有一种声音：“我为什么要学习数据结构与算法？没有数据结构与算法，我一样很好地完成了工作？”

实际上，算法是一个十分宽泛的概念，我们写的任何程序都可称为算法，甚至往冰箱里面放一头大象，也要经过开门、放入、关门这样的规划，这也可以视为一种简单的算法。可以说，简单的算法是人类的本能。而算法知识的学习则是吸取前人的经验，对复杂的问题进行归类、抽象，帮助我们脱离刀耕火种时代，系统掌握算法的一个过程。

随着自身成长和职业发展，不论是做前端、服务端还是客户端，任何一个程序员都会开始面对更加复杂的问题，算法和数据结构知识就变得不可或缺了。

我一直认为前端工程师则是最需要重视算法和数据结构基础的人。因为历史原因，不少前端工程师是从视觉设计、网站编辑转过来的，在学校没有学过相应的基础课程，而数据结构与算法的经典名著大部分又没照顾到入门的需要，所以前端工程师如果自身不重视算法和数据结构这样的基础知识，很可能陷入数年从事单一重复劳动毫无成长这样的职业发展困境。在移动浪潮到来之后，用户体验要求越来越高，对前端提出了更高的要求，前端这个职能，必须提高自身才能继续发展，未来的网页 UI，绝对不是靠几个选择器操作加超链接就能应付的。越来越复杂的产品和基础库，需要坚实的数据结构与算法基础才能驾驭。

本书对前端工程师是非常好的数据结构与算法入门书，它的难度非常适合前端工程师补习基础知识。全书仅 200 页，对于有渴求数据结构与算法的前端工程师来说这是非常不错的开始。特别值得一提的是每章后面的小练习，题目不多但是非常有可操作性。

程劭非

阿里无线事业部高级技术专家

2014 年 7 月

前言

在过去的几年中，得益于 Node.js 和 SpiderMonkey 等平台，JavaScript 越来越广泛地用于服务器端编程。鉴于 JavaScript 语言已经走出了浏览器，程序员发现他们需要更多传统语言（比如 C++ 和 Java）提供的工具。这些工具包括传统的数据结构（如链表、栈、队列、图等），也包括传统的排序和查找算法。本书讨论在使用 JavaScript 进行服务器端编程时，如何实现这些数据结构和算法。

JavaScript 程序员会发现本书很有用，因为本书讨论了在 JavaScript 语言的限制下，如何实现数据结构和算法。这些限制包括：数组即对象、无处不在的全局变量、基于原型的对象模型等。JavaScript 作为一种编程语言，名声有点“不大好”，但是本书展示了如何使用 JavaScript 语言中“好的一面”去实现高效的数据结构和算法，进而为 JavaScript 正名。

为什么要学习数据结构和算法

我假设本书的读者中，有很多人没接受过正规的计算机科学教育。如果你接受过，那么你已经知道了学习数据结构和算法为何如此重要。如果你没有计算机科学学位或者没有正规学习过计算机科学，那么请耐心读完本节。

计算机科学家尼克劳斯·沃思（Nicklaus Wirth）写过一本计算机程序设计教材，书名是《算法 + 数据结构 = 程序》（*Algorithms + Data Structures = Programs*, Prentice-Hall）。这个书名就概括了计算机编程的精华。除了“Hello world!”等无关紧要的程序，任何一个有些规模的程序都需要某种类型的数据结构来保存程序中用到的数据，还需要一个或多个算法将数据从输入转换为输出。

对于那些没有在学校学习过计算机科学的程序员来说，唯一熟悉的数据结构就是数组。在处理一些问题时，数组无疑是很好的选择，但对于很多复杂的问题，数组就显得太过简陋了。大多数有经验的程序员都愿意承认这样一个事实：对于很多编程问题，当他们想出一

个合适的数据结构，设计和实现解决这些问题的算法就变得手到擒来。

二叉查找树（BST）就是一个这样的例子。设计二叉查找树的目的是为了更方便查找一组数据中的最小值和最大值，由这个数据结构自然引申出一个查找算法，该算法比目前最好的查找算法效率还要高。不熟悉二叉查找树的程序员可能会使用一个更简单的数据结构，但效率上就打了个折扣。

学习算法非常重要，因为解决同样的问题，往往可以使用多种算法。对于高效程序员来说，知道哪种算法效率最高非常重要。比如，现在至少有六七种排序算法，如果知道快速排序比选择排序效率更高，那么就会让排序过程变得高效。又比如，实现一个线性查找的算法很简单，但是如果知道有时二分查找可能比线性查找快两倍以上，那你势必会写出一个更好的程序。

深入学习数据结构和算法，不仅可以知道哪种数据结构和算法更高效，还会知道如何找出最适合解决手头问题的数据结构和算法。写程序，尤其是用 JavaScript 写程序时，经常需要权衡，知道了本书涵盖的数据结构和算法的优缺点，在解决具体的编程问题时就容易做出正确的选择。

阅读本书需要的工具

本书使用的编程环境是基于 SpiderMonkey JavaScript 引擎的 JavaScript shell。第 1 章提供了该 shell 的下载说明。也可以使用其他一些 JavaScript Shell，比如 Node.js 提供的 JavaScript shell，你只需自己对书中的程序做一些转换，就能在 Node.js 上运行。除了 JavaScript shell，再有一个用于编写 JavaScript 程序的文本编辑器就够了。

本书组织结构

- 第 1 章简单概述 JavaScript 语言，至少介绍了本书用到的 JavaScript 特性。这一章还展示了贯穿全书的编程风格。
- 第 2 章讨论计算机编程中最常见的数据结构：数组。数组是 JavaScript 原生的数据类型。
- 第 3 章介绍我们实现的第一个数据结构：列表。
- 第 4 章介绍栈。栈是一种贯穿计算机科学的数据结构，编译器和操作系统的实现都用到了栈。
- 第 5 章讨论队列。队列是对你在银行或杂货店里所排队的一种抽象。队列广泛应用于处理数据之前，必须先把数据按顺序排成一队的模拟软件中。

- 第 6 章介绍链表。链表是对列表的修改，链表里的每个元素都是一个单独的对象，该对象和它两边的元素相连。当程序中需要插入和删除多个元素时，使用链表非常高效。
- 第 7 章展示如何实现和使用字典，字典是将数据存储为键值对的数据结构。
- 实现字典的一种方法是通过散列表，第 8 章讨论了如何实现散列表和在表中存储数据的散列算法。
- 第 9 章介绍集合。和数据结构相关的书通常不会介绍集合，但是当某个数据集不允许有重复元素出现时，使用集合是一个很好的选择。
- 第 10 章的重点是二叉树和二叉查找树。前面提到过，二叉查找树是一种存储有序元素的极佳选择。
- 第 11 章介绍图和图的算法。图用来表示计算机网络节点或者地图上的城市等数据。
- 第 12 章转向算法，讨论各种排序算法，包括简单易实现但处理大数据集时效率不高的算法，以及适合处理大数据集的复杂算法。
- 第 13 章的主题还是算法，不过这回是查找算法，比如线性查找和二分查找。
- 第 14 章是本书的最后一章，讨论两种更高级的算法——动态规划和贪心算法。这些算法能解决难题，通常的算法在面对这些问题时要么执行速度太慢，要么难于实现。我们会分析几个用动态规划和贪心算法解决的典型问题。

排版约定

本书使用的排版约定如下。

- 楷体
表示新的术语。
- 等宽字体
表示程序片段，也用于在正文中表示程序中使用的变量、函数名、命令行代码、环境变量、语句和关键字等元素。
- 等宽粗体
表示应该由用户逐字输入的命令或者其他文本。
- 等宽斜体
表示应该由用户输入的值或根据上下文决定的值替换的文本。

使用代码示例

可以在这里下载本书随附的资料（代码示例、练习题等）：<https://github.com/oreillymedia/>

`data_structures_and_algorithms_using_javascript`。

让本书助你一臂之力。也许你需要在自己的程序或文档中用到本书中的代码。除非大段大段地使用，否则不必与我们联系取得授权。例如，无需请求许可，就可以用本书中的几段代码写成一个程序。但是销售或者发布 O'Reilly 图书中代码的光盘则必须事先获得授权。引用书中的代码来回答问题也无需授权。将大段的示例代码整合到你自己的产品文档中则必须经过许可。

使用我们的代码时，希望你能标明它的出处，但不强求。出处一般包括书名、作者、出版商和 ISBN，例如：*Data Structure and Algorithms Using JavaScript*，Michael McMillan 著（O'Reilly，2014）。版权所有，978-1-449-36493-9。

如果还有关于使用代码的未尽事宜，可以随时与我们联系：permissions@oreilly.com。

Safari® Books Online



Safari Books Online (<http://www.safaribooksonline.com>) 是应需而变的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。

Safari Books Online 是技术专家、软件开发人员、Web 设计师、商务人士和创意人士开展调研、解决问题、学习和认证培训的第一手资料。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)
奥莱利技术咨询 (北京) 有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

http://oreil.ly/data_structures_algorithms_JS。

对于本书的评论和技术性问题，请发送电子邮件到：

bookquestions@oreilly.com

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

<http://www.oreilly.com>

我们在 Facebook 的地址如下：<http://facebook.com/oreilly>

请关注我们的 Twitter 动态：<http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下：<http://www.youtube.com/oreillymedia>

致谢

写成本书，需要感谢很多人。首先要感谢我的组稿编辑 Simon St. Laurent，他对本书充满信心并鼓励我开始写作。Meghan Blanchette 女士为了让我按时完成写作费尽心思，如果本书有过拖稿现象，那一定不是她的错。Brian MacDonald 做了很多工作让本书变得通俗易懂，他编辑校订了本书的一些章节，文字比我当初的更清晰。我还要感谢技术审稿人，他们阅读了本书的全部文字和代码，并且指出了行文和代码表达不够清楚的地方。我的同事 Cynthia Fehrenbach 将我简陋的草图绘制成现在精美、清晰的插图，在本书将要出版的最后时刻，她还愿意重新绘制几幅插图，对此我要特别感谢她。最后，我要感谢在 Mozilla 工作的所有人，是他们设计了如此出色的 JavaScript 引擎和命令行工具，他们还为如何使用 JavaScript 语言 and 这个工具编写了非常棒的文档。

目录

推荐序	XI
前言	XII
第 1 章 JavaScript 的编程环境和模型	1
1.1 JavaScript 环境	1
1.2 JavaScript 编程实践	2
1.2.1 声明和初始化变量	3
1.2.2 JavaScript 中的算术运算和数学库函数	3
1.2.3 判断结构	4
1.2.4 循环结构	6
1.2.5 函数	7
1.2.6 变量作用域	7
1.2.7 递归	9
1.3 对象和面向对象编程	10
1.4 小结	11
第 2 章 数组	13
2.1 JavaScript 中对数组的定义	13
2.2 使用数组	13
2.2.1 创建数组	14
2.2.2 读写数组	15
2.2.3 由字符串生成数组	15
2.2.4 对数组的整体性操作	16
2.3 存取函数	17
2.3.1 查找元素	17

2.3.2	数组的字符串表示	18
2.3.3	由已有数组创建新数组	18
2.4	可变函数	19
2.4.1	为数组添加元素	19
2.4.2	从数组中删除元素	20
2.4.3	从数组中间位置添加和删除元素	21
2.4.4	为数组排序	21
2.5	迭代器方法	22
2.5.1	不生成新数组的迭代器方法	22
2.5.2	生成新数组的迭代器方法	25
2.6	二维和多维数组	27
2.6.1	创建二维数组	27
2.6.2	处理二维数组的元素	28
2.6.3	参差不齐的数组	29
2.7	对象数组	30
2.8	对象中的数组	31
2.9	练习	32
第3章	列表	33
3.1	列表的抽象数据类型定义	33
3.2	实现列表类	34
3.2.1	append: 给列表添加元素	35
3.2.2	remove: 从列表中删除元素	35
3.2.3	find: 在列表中查找某一元素	35
3.2.4	length: 列表中有多少个元素	36
3.2.5	toString: 显示列表中的元素	36
3.2.6	insert: 向列表中插入一个元素	37
3.2.7	clear: 清空列表中所有的元素	37
3.2.8	contains: 判断给定值是否在列表中	37
3.2.9	遍历列表	38
3.3	使用迭代器访问列表	39
3.4	一个基于列表的应用	40
3.4.1	读取文本文件	40
3.4.2	使用列表管理影碟租赁	41
3.5	练习	44
第4章	栈	45
4.1	对栈的操作	45

4.2	栈的实现	46
4.3	使用 Stack 类	48
4.3.1	数制间的相互转换	49
4.3.2	回文	50
4.3.3	递归演示	51
4.4	练习	52
第 5 章	队列	53
5.1	对队列的操作	53
5.2	一个用数组实现的队列	54
5.3	使用队列：方块舞的舞伴分配问题	57
5.4	使用队列对数据进行排序	61
5.5	优先队列	63
5.6	练习	65
第 6 章	链表	67
6.1	数组的缺点	67
6.2	定义链表	67
6.3	设计一个基于对象的链表	69
6.3.1	Node 类	69
6.3.2	LinkedList 类	69
6.3.3	插入新节点	69
6.3.4	从链表中删除一个节点	71
6.4	双向链表	74
6.5	循环链表	78
6.6	链表的其他方法	79
6.7	练习	79
第 7 章	字典	81
7.1	Dictionary 类	81
7.2	Dictionary 类的辅助方法	83
7.3	为 Dictionary 类添加排序功能	85
7.4	练习	86
第 8 章	散列	87
8.1	散列概览	87
8.2	HashTable 类	88
8.2.1	选择一个散列函数	88
8.2.2	一个更好的散列函数	91

8.2.3	散列化整型键	93
8.2.4	对散列表排序、从散列表中取值	95
8.3	碰撞处理	96
8.3.1	开链法	96
8.3.2	线性探测法	99
8.4	练习	100
第 9 章	集合	101
9.1	集合的定义、操作和属性	101
9.1.1	集合的定义	101
9.1.2	对集合的操作	102
9.2	Set 类的实现	102
9.3	更多集合操作	104
9.4	练习	107
第 10 章	二叉树和二叉查找树	109
10.1	树的定义	109
10.2	二叉树和二叉查找树	111
10.2.1	实现二叉查找树	111
10.2.2	遍历二叉查找树	113
10.3	在二叉查找树上进行查找	116
10.3.1	查找最小值和最大值	116
10.3.2	查找给定值	117
10.4	从二叉查找树上删除节点	118
10.5	计数	120
10.6	练习	123
第 11 章	图和图算法	125
11.1	图的定义	125
11.2	用图对现实中的系统建模	127
11.3	图类	127
11.3.1	表示顶点	127
11.3.2	表示边	127
11.3.3	构建图	128
11.4	搜索图	130
11.4.1	深度优先搜索	130
11.4.2	广度优先搜索	133
11.5	查找最短路径	135
11.5.1	广度优先搜索对应的最短路径	135

11.5.2 确定路径	135
11.6 拓扑排序	137
11.6.1 拓扑排序算法	137
11.6.2 实现拓扑排序算法	137
11.7 练习	141
第 12 章 排序算法	143
12.1 数组测试平台	143
12.2 基本排序算法	145
12.2.1 冒泡排序	145
12.2.2 选择排序	148
12.2.3 插入排序	150
12.2.4 基本排序算法的计时比较	151
12.3 高级排序算法	153
12.3.1 希尔排序	153
12.3.2 归并排序	158
12.3.3 快速排序	163
12.4 练习	167
第 13 章 检索算法	169
13.1 顺序查找	169
13.1.1 查找最小值和最大值	172
13.1.2 使用自组织数据	175
13.2 二分查找算法	177
13.3 查找文本数据	183
13.4 练习	185
第 14 章 高级算法	187
14.1 动态规划	187
14.1.1 动态规划实例：计算斐波那契数列	188
14.1.2 寻找最长公共子串	191
14.1.3 背包问题：递归解决方案	194
14.1.4 背包问题：动态规划方案	195
14.2 贪心算法	196
14.2.1 第一个贪心算法案例：找零问题	196
14.2.2 背包问题的贪心算法解决方案	197
14.3 练习	199
封面介绍	200

JavaScript的编程环境和模型

本章描述了 JavaScript 的编程环境和基本的编程模块，本书的后续章节将使用这些知识定义各种数据结构和实现各种算法。

1.1 JavaScript环境

JavaScript 历来是一种仅在浏览器里运行的程序语言。然而在过去的几年中，这种情况发生了变化，JavaScript 发展为可以作为桌面程序执行，或者在服务器上执行。本书就使用这样一种类似的环境：JavaScript shell，这是由 Mozilla 提供的综合 JavaScript 编程环境 SpiderMonkey 中的一部分。

打开 SpiderMonkey 的每日构建页面 (<http://mzl.la/MKOUFY>)，滚动至页面底部，根据你的计算机操作系统，下载相应的 JavaScript shell。

下载完成后，有两种使用 JavaScript shell 的方式。可以选择在交互模式下使用 shell，也可以将 JavaScript 代码保存在一个文件中，使用 shell 进行解释执行。在命令提示符下输入 js，进入 shell 的交互模式，命令行里将会出现 js> 提示符，这时就可以输入 JavaScript 表达式和语句了。

下面演示了和 JavaScript shell 进行交互的典型场景：

```
bash
js> 1
1
js> 1+2
```