

TURING

图灵程序设计丛书

Packt

Learning JavaScript Data Structures
and Algorithms, Third Edition

学习JavaScript 数据结构与算法 (第3版)

[巴西] 洛伊安妮·格罗纳 著
吴双 邓钢 孙晓博 等译

用JavaScript学习常用的数据结构与算法
高效解决编程常见问题



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Learning JavaScript Data Structures
and Algorithms, Third Edition

学习JavaScript 数据结构与算法 (第3版)



[巴西] 洛伊安妮·格罗纳 著
吴双 邓钢 孙晓博 等译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

学习JavaScript数据结构与算法：第3版 / (巴西)
洛伊安妮·格罗纳 (Loiane Groner) 著；吴双等译. --
北京：人民邮电出版社，2019.5
(图灵程序设计丛书)
ISBN 978-7-115-51017-4

I. ①学… II. ①洛… ②吴… III. ①数据结构②
JAVA语言—程序设计 IV. ①TP311.12②TP312.8

中国版本图书馆CIP数据核字(2019)第054767号

内 容 提 要

本书首先介绍了 JavaScript 语言的基础知识 (包括 ECMAScript 和 TypeScript)，其次讨论了数组、栈、队列、双端队列和链表等重要的数据结构，随后分析了集合、字典和散列表的工作原理，接下来阐述了递归的原理、什么是树以及二叉堆和堆排序，然后介绍了图、DFS 和 BFS 算法、各种排序 (冒泡排序、选择排序、插入排序、归并排序、快速排序、计数排序、桶排序和基数排序) 和搜索 (顺序搜索、二分搜索和内插搜索) 算法以及随机算法，接着介绍了分而治之、动态规划、贪心算法和回溯算法等高级算法以及函数式编程，最后还介绍了如何计算算法的复杂度。

如果你是计算机科学专业的学生，或是刚刚开启职业生涯的技术人员，想探索 JavaScript 的最佳能力，这本书一定适合你。

-
- ◆ 著 [巴西] 洛伊安妮·格罗纳
译 吴 双 邓 钢 孙晓博 等
责任编辑 杨 琳
责任印制 周昇亮
 - ◆ 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>
北京鑫正大印刷有限公司印刷
 - ◆ 开本：800×1000 1/16
印张：19.25
字数：466千字 2019年5月第1版
印数：1-3 000册 2019年5月北京第1次印刷
- 著作权合同登记号 图字：01-2018-7611号
-

定价：69.00元

读者服务热线：(010)51095183转600 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广登字 20170147 号

版权声明

Copyright © 2018 Packt Publishing. First published in the English language under the title *Learning JavaScript Data Structures and Algorithms, Third Edition*.

Simplified Chinese-language edition copyright © 2019 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Packt Publishing 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

致我的父母，感谢他们的爱和支持，以及这些年对我的指引。

致我的丈夫，感谢他的支持，感恩他成为我生命旅程中的爱侣。

前 言

JavaScript 是当下最流行的编程语言之一。由于浏览器的原生支持（无须安装任何插件），JavaScript 也被称作“互联网语言”。JavaScript 的应用非常广泛，不仅用于前端开发，也被用到服务器（Node.js）环境、数据库（MongoDB）环境和移动设备中，同样还被用在嵌入式设备和物联网（IoT）设备中。

对任何专业技术人员来说，理解数据结构都非常重要。作为软件开发者，我们要能够借助编程语言来解决问题，而数据结构是这些问题的解决方案中不可或缺的一部分。如果选择了不恰当的数据结构，可能会影响所写程序的性能。因此，了解不同数据结构和它们的适用范围十分重要。

算法在计算机科学中扮演着非常重要的角色。解决一个问题有很多种方法，但有些方法会比其他方法更好。因此，了解一下最著名的算法也很重要。

本书为数据结构和算法初学者所写，也为熟悉数据结构和算法并想在 JavaScript 语言中使用它们的人所写。

快乐地编码吧！

读者对象

如果你是一名计算机专业的学生，或者正处于技术生涯的开端，想要探索 JavaScript 最强大的功能，那么本书正适合你。如果你已经对编程很熟悉，但是想要提升在算法和数据结构方面的技能，本书同样适合你。

你只需要懂得 JavaScript 的基础知识和编程逻辑，就可以开始享受算法的乐趣了。

本书结构

第 1 章“JavaScript 简介”，讲述了 JavaScript 的基础知识，可以帮助你更好地学习数据结构和算法，同时还介绍了如何搭建开发环境来运行书中的代码示例。

第 2 章“ECMAScript 和 TypeScript 概述”，介绍了 2015 年后新增的一些 JavaScript 功能，以及 TypeScript 的基本功能。TypeScript 是 JavaScript 的一个超集。

第 3 章“数组”，介绍了如何使用数组这种最基础且最常用的数据结构。这一章演示了如何对数组声明、初始化、添加和删除其中的元素，还讲述了如何使用 JavaScript 语言本身支持的数组方法。

第 4 章“栈”，介绍了栈这种数据结构，演示了如何创建栈以及怎样添加和删除元素，还讨论了如何用栈解决计算机科学中的一些问题。

第 5 章“队列和双端队列”，详述了队列这种数据结构，演示了如何创建队列，以及如何添加和删除队列中的元素。此外，这一章也介绍了一种特殊的队列——双端队列数据结构。这一章还讨论了如何用队列解决计算机科学中的一些问题，以及栈和队列的主要区别。

第 6 章“链表”，讲解如何用对象和指针从头创建链表这种数据结构。这一章除了讨论如何声明、创建、添加和删除链表元素之外，还介绍了不同类型的链表，例如双向链表和循环链表。

第 7 章“集合”，介绍了集合这种数据结构，讨论了如何用集合存储非重复性的元素。此外，还详述了对集合的各种操作以及相应代码的实现。

第 8 章“字典和散列表”，深入讲解字典、散列表及它们之间的区别。这一章介绍了这两种数据结构是如何声明、创建和使用的，还探讨了如何解决散列冲突，以及如何创建更高效的散列函数。

第 9 章“递归”，介绍了递归的概念，描述了声明式和递归式算法之间的区别。

第 10 章“树”，讲解了树这种数据结构和它的相关术语，重点讨论了二叉搜索树，以及如何在树中搜索、遍历、添加和删除节点。这一章还介绍了自平衡树，包括 AVL 树和红黑树。

第 11 章“二叉堆和堆排序”，介绍了最小堆和最大堆数据结构，以及怎样使用堆作为一个优先队列，还讨论了著名的堆排序算法。

第 12 章“图”，介绍了图这种数据结构和它的适用范围。这一章讲述了图的常用术语和不同表示方式，探讨了如何使用深度优先搜索算法和广度优先搜索算法遍历图，以及它们的适用范围。

第 13 章“排序和搜索算法”，探讨了常用的排序算法，如冒泡排序（包括改进版）、选择排序、插入排序、归并排序和快速排序。这一章还介绍了计数排序和基数排序这两种分布式排序算法，搜索算法中的顺序搜索和二分搜索，以及怎样随机排列一个数组。

第 14 章“算法设计与技巧”，介绍了一些算法技巧和著名的算法，以及 JavaScript 函数式编程。

第 15 章“算法复杂度”，介绍了大 O 表示法的概念，以及本书实现算法的复杂度列表。这一章还介绍了 NP 完全问题和启发式算法。最后，讲解了提升算法能力的诀窍。

阅读须知

尽管本书第 1 章对 JavaScript 进行了简单介绍，你仍然需要了解基本的 JavaScript 知识和编程逻辑。

要测试本书提供的代码示例，你需要一个代码编辑器（例如 Atom 或 Visual Studio Code）以便阅读代码，还需要一个浏览器（Chrome、Firefox 或 Edge）。

你也可以访问 <https://javascript-ds-algorithms-book.firebaseio.com/>，在线测试代码。同样，记得打开浏览器中的开发者工具，这样你就可以看到控制台上的输出结果了。

下载示例代码

你可以用你的账户从 <http://www.packtpub.com> 下载所有已购买 Packt 图书的示例代码文件。如果你从其他地方购买了本书，可以访问 <http://www.packtpub.com/support> 并注册，我们将通过电子邮件把文件发送给你。

下载代码文件的步骤如下：

- (1) 在 www.packtpub.com 登录或注册；
- (2) 选择 SUPPORT 标签页；
- (3) 点击 Code Downloads & Errata；
- (4) 在 Search 框中输入书名并根据屏幕上的指示操作。

文件下载后，请使用以下软件的最新版本解压：

- Windows 系统请使用 WinRAR 或 7-Zip
- Mac 系统请使用 Zipeg、iZip 或 UnRarX
- Linux 系统请使用 7-Zip 或 PeaZip

本书的代码包在 GitHub 的托管地址是 <https://github.com/PacktPublishing/Learning-JavaScript-Data-Structures-and-Algorithms-Third-Edition>。只要代码有更新，它就会被更新到 GitHub 仓库中去。

其他图书或视频的代码包也可以到 <https://github.com/PacktPublishing/> 查阅。别错过！

排版约定

在本书中，你会发现一些不同的文本样式。

正文中的代码这样表示：“可能你在网上的一些例子里看到过 JavaScript 的 `include` 语句，

或者放在 head 标签中的 JavaScript 代码。”

代码段的格式如下：

```
class Stack {
  constructor() {
    this.items = []; // {}
  }
}
```

如果我们想让你重点关注代码段中的某个部分，会加粗显示：

```
const stack = new Stack();
console.log(stack.isEmpty()); // outputs true
```

所有的命令行输入或输出的格式如下：

```
npm install http-server -g
```

新术语、重点词汇，以及你可以在屏幕上看到的词（例如，菜单或对话框里的词）以**黑体**标示。举个例子：“从 **Administration** 面板中选择 **System info**。”



此图标表示警告或需要特别注意的内容。



此图标表示提示或者技巧。

联系我们

欢迎提出反馈。

一般反馈：请发送电子邮件至 feedback@packtpub.com，并在邮件主题中注明书名。如果你对本书任何方面有疑问，请发送邮件至 questions@packtpub.com。

勘误：尽管我们会尽力确保内容准确，错误还是在所难免。如果你发现了书中的错误，希望你能告知我们，我们不胜感激。请访问 www.packtpub.com/submit-errata，选择你的书，点击勘误提交表单的链接，并输入详情。^①

反盗版：如果你在互联网上发现我们的作品被非法复制，我们会非常感激你将地址和网站名称提供给我们。请将盗版材料的链接发送到 copyright@packtpub.com。

如果你有兴趣成为作者：如果你有某个主题的专业知识，并且有兴趣写成或帮助促成一本书，

^① 针对本书中文版的勘误，请到 <http://www.it-ebooks.com.cn/book/2653> 查看和提交。——编者注

请参考我们的作者指南 www.packtpub.com/authors。

评论

请留下一条评论。当你阅读并使用本书后，何不在你购买本书的网站上留下一条评论呢？潜在的读者可通过你公正的评论来决定是否购买，Packt 的工作人员可以知道你对我们产品的看法，我们的作者能看到你对他们的反馈。谢谢！

要了解更多有关 Packt 的信息，请访问 packtpub.com。

电子书

扫描如下二维码，即可购买本书电子版。



目 录

第 1 章 JavaScript 简介	1	2.2.5 声明展开和剩余参数	29
1.1 JavaScript 数据结构与算法	1	2.2.6 增强的对象属性	30
1.2 环境搭建	2	2.2.7 使用类进行面向对象编程	31
1.2.1 最简单的环境搭建	2	2.2.8 乘方运算符	33
1.2.2 使用 Web 服务器	3	2.2.9 模块	33
1.2.3 Node.js http-server	5	2.3 介绍 TypeScript	39
1.3 JavaScript 基础	5	2.3.1 类型推断	40
1.3.1 变量	6	2.3.2 接口	41
1.3.2 运算符	8	2.3.3 其他 TypeScript 功能	43
1.3.3 真值和假值	11	2.3.4 TypeScript 中对 JavaScript 文件 的编译时检查	43
1.3.4 相等运算符 (==和===)	12	2.4 小结	44
1.4 控制结构	14	第 3 章 数组	45
1.4.1 条件语句	14	3.1 为什么用数组	45
1.4.2 循环	15	3.2 创建和初始化数组	46
1.5 函数	16	3.3 添加元素	47
1.6 JavaScript 面向对象编程	17	3.3.1 在数组末尾插入元素	47
1.7 调试工具	18	3.3.2 在数组开头插入元素	48
1.8 小结	20	3.4 删除元素	49
第 2 章 ECMAScript 和 TypeScript		3.4.1 从数组末尾删除元素	49
概述	21	3.4.2 从数组开头删除元素	49
2.1 ECMAScript 还是 JavaScript	21	3.5 在任意位置添加或删除元素	51
2.1.1 ES6、ES2015、ES7、ES2016、 ES8、ES2017 和 ES.Next	21	3.6 二维和多维数组	51
2.1.2 使用 Babel.js	23	3.6.1 迭代二维数组的元素	52
2.2 ECMAScript 2015+的功能	24	3.6.2 多维数组	53
2.2.1 用 let 替代 var 声明变量	24	3.7 JavaScript 的数组方法参考	54
2.2.2 模板字面量	27	3.7.1 数组合并	55
2.2.3 箭头函数	27	3.7.2 迭代器函数	55
2.2.4 函数的参数默认值	28		

3.7.3	ECMAScript 6 和数组的新功能	57	第 5 章	队列和双端队列	82
3.7.4	排序元素	60	5.1	队列数据结构	82
3.7.5	搜索	63	5.1.1	创建队列	83
3.7.6	输出数组为字符串	64	5.1.2	使用 Queue 类	86
3.8	类型数组	64	5.2	双端队列数据结构	87
3.9	TypeScript 中的数组	65	5.2.1	创建 Deque 类	87
3.10	小结	66	5.2.2	使用 Deque 类	89
第 4 章	栈	67	5.3	使用队列和双端队列来解决问题	90
4.1	创建一个 JavaScript 数据结构和算法库	67	5.3.1	循环队列——击鼓传花游戏	90
4.2	栈数据结构	68	5.3.2	回文检查器	91
4.2.1	创建一个基于数组的栈	69	5.3.3	JavaScript 任务队列	93
4.2.2	向栈添加元素	69	5.4	小结	93
4.2.3	从栈移除元素	70	第 6 章	链表	94
4.2.4	查看栈顶元素	70	6.1	链表数据结构	94
4.2.5	检查栈是否为空	71	6.2	双向链表	106
4.2.6	清空栈元素	71	6.2.1	在任意位置插入新元素	107
4.2.7	使用 Stack 类	71	6.2.2	从任意位置移除元素	109
4.3	创建一个基于 JavaScript 对象的 Stack 类	73	6.3	循环链表	111
4.3.1	向栈中插入元素	73	6.3.1	在任意位置插入新元素	112
4.3.2	验证一个栈是否为空和它的大小	74	6.3.2	从任意位置移除元素	113
4.3.3	从栈中弹出元素	74	6.4	有序链表	114
4.3.4	查看栈顶的值并将栈清空	75	6.5	创建 StackLinkedList 类	116
4.3.5	创建 toString 方法	75	6.6	小结	117
4.4	保护数据结构内部元素	76	第 7 章	集合	118
4.4.1	下划线命名约定	76	7.1	构建数据集合	118
4.4.2	用 ES2015 的限定作用域 Symbol 实现类	77	7.2	创建集合类	119
4.4.3	用 ES2015 的 WeakMap 实现类	77	7.2.1	has(element) 方法	119
4.4.4	ECMAScript 类属性提案	78	7.2.2	add 方法	120
4.5	用栈解决问题	79	7.2.3	delete 和 clear 方法	120
4.6	小结	81	7.2.4	size 方法	121
			7.2.5	values 方法	122
			7.2.6	使用 Set 类	122
			7.3	集合运算	123
			7.3.1	并集	123
			7.3.2	交集	125
			7.3.3	差集	127
			7.3.4	子集	128

7.4	ECMAScript 2015——Set 类	130	10.4.1	中序遍历	175
7.5	多重集或袋	132	10.4.2	先序遍历	176
7.6	小结	133	10.4.3	后序遍历	177
第 8 章	字典和散列表	134	10.5	搜索树中的值	178
8.1	字典	134	10.5.1	搜索最小值和最大值	178
8.1.1	创建字典类	135	10.5.2	搜索一个特定的值	180
8.1.2	使用 Dictionary 类	141	10.5.3	移除一个节点	182
8.2	散列表	142	10.6	自平衡树	185
8.2.1	创建散列表	143	10.6.1	Adelson-Velskii-Landi 树 (AVL 树)	185
8.2.2	使用 HashTable 类	146	10.6.2	红黑树	194
8.2.3	散列表和散列集合	147	10.7	小结	200
8.2.4	处理散列表中的冲突	147	第 11 章	二叉堆和堆排序	201
8.2.5	创建更好的散列函数	158	11.1	二叉堆数据结构	201
8.3	ES2015 Map 类	159	11.1.1	创建最小堆类	202
8.4	ES2105 WeakMap 类和 WeakSet 类	159	11.1.2	创建最大堆类	208
8.5	小结	160	11.2	堆排序算法	209
第 9 章	递归	161	11.3	小结	211
9.1	理解递归	161	第 12 章	图	212
9.2	计算一个数的阶乘	162	12.1	图的相关术语	212
9.2.1	迭代阶乘	162	12.2	图的表示	214
9.2.2	递归阶乘	163	12.2.1	邻接矩阵	215
9.3	斐波那契数列	165	12.2.2	邻接表	215
9.3.1	迭代求斐波那契数	166	12.2.3	关联矩阵	216
9.3.2	递归求斐波那契数	166	12.3	创建 Graph 类	216
9.3.3	记忆化斐波那契数	167	12.4	图的遍历	219
9.4	为什么要用递归? 它更快吗	167	12.4.1	广度优先搜索	220
9.5	小结	168	12.4.2	深度优先搜索	225
第 10 章	树	169	12.5	最短路径算法	231
10.1	树数据结构	169	12.5.1	Dijkstra 算法	232
10.2	树的相关术语	170	12.5.2	Floyd-Warshall 算法	234
10.3	二叉树和二叉搜索树	170	12.6	最小生成树	235
10.3.1	创建 BinarySearch- Tree 类	171	12.6.1	Prim 算法	236
10.3.2	向二叉搜索树中插入 一个键	172	12.6.2	Kruskal 算法	237
10.4	树的遍历	175	12.7	小结	238
			第 13 章	排序和搜索算法	239

13.1 排序算法.....	239	14.3 贪心算法.....	274
13.1.1 冒泡排序.....	239	14.3.1 最少硬币找零问题.....	274
13.1.2 选择排序.....	242	14.3.2 分数背包问题.....	275
13.1.3 插入排序.....	244	14.4 回溯算法.....	276
13.1.4 归并排序.....	245	14.4.1 迷宫老鼠问题.....	277
13.1.5 快速排序.....	247	14.4.2 数独解题器.....	279
13.1.6 计数排序.....	251	14.5 函数式编程简介.....	282
13.1.7 桶排序.....	253	14.5.1 函数式编程与命令式编程.....	282
13.1.8 基数排序.....	255	14.5.2 ES2015+和函数式编程.....	283
13.2 搜索算法.....	257	14.5.3 JavaScript 函数式工具 箱——map、filter 和 reduce.....	284
13.2.1 顺序搜索.....	257	14.5.4 JavaScript 函数式类库和数 据结构.....	286
13.2.2 二分搜索.....	258	14.6 小结.....	286
13.2.3 内插搜索.....	260		
13.3 随机算法.....	261		
13.4 小结.....	262		
第 14 章 算法设计与技巧.....	263	第 15 章 算法复杂度.....	287
14.1 分而治之.....	263	15.1 大 O 表示法.....	287
14.2 动态规划.....	265	15.1.1 理解大 O 表示法.....	287
14.2.1 最少硬币找零问题.....	266	15.1.2 时间复杂度比较.....	289
14.2.2 背包问题.....	268	15.1.3 NP 完全理论概述.....	292
14.2.3 最长公共子序列.....	270	15.2 用算法娱乐身心.....	293
14.2.4 矩阵链相乘.....	272	15.3 小结.....	294

JavaScript 简介



JavaScript 是一门非常强大的编程语言。它是最流行的编程语言之一，也是互联网上最卓越的语言之一。在 GitHub（世界上最大的代码托管站点）上，托管了 400 000 多个 JavaScript 代码仓库（用 JavaScript 开发的项目数量也是最多的，参看 <http://github.info>）。使用 JavaScript 的项目数量还在逐年增长。

JavaScript 不仅可用于前端开发，也适用于后端开发，而 Node.js 就是其背后的技术。Node 包的数量也呈指数级增长。JavaScript 同样可以用于移动开发领域，并且是 Apache Cordova 中最流行的语言之一。Apache Cordova 是一个能让开发者使用 HTML、CSS 和 JavaScript 等语言的混合式框架，你可以通过它来搭建应用，并且生成供 Android 系统使用的 APK 文件和供 iOS（苹果系统）使用的 IPA 文件。当然，也别忘了桌面端应用开发。我们可以使用一个名为 Electron 的 JavaScript 框架来编写同时兼容 Linux、Mac OS 和 Windows 的桌面端应用。JavaScript 还可以用于嵌入式设备以及物联网（IoT）设备。正如你所看到的，到处都有 JavaScript 的身影！

要成为一名 Web 开发工程师，掌握 JavaScript 必不可少。

本章，你会学到 JavaScript 的语法和一些必要的基础，这样就可以开始开发自己的数据结构和算法了。本章内容如下：

- 环境搭建和 JavaScript 基础
- 控制结构和函数
- JavaScript 面向对象编程
- 调试工具

1.1 JavaScript 数据结构与算法

在本书中，你将学习最常用的数据结构和算法。为什么用 JavaScript 来学习这些数据结构和算法呢？我们已经回答了这个问题。JavaScript 非常受欢迎，作为函数式编程语言，它非常适合用来学习数据结构和算法。通过它来学习数据结构比 C、Java 或 Python 这些标准语言更简单，学习新东西也会变得很有趣。谁说数据结构和算法只为 C、Java 这样的语言而生？在前端开发当

中，你可能也需要实现这些语言。

学习数据结构和算法十分重要。首要原因是数据结构和算法可以很高效地解决常见问题，这对你今后所写代码的质量至关重要（也包括性能；要是用了不恰当的数据结构或算法，很可能产生性能问题）。其次，对于计算机科学，算法是最基础的概念。最后，如果你想入职最好的 IT 公司（如谷歌、亚马逊、微软、eBay 等），数据结构和算法是面试问题的重头戏。

让我们开始学习吧！

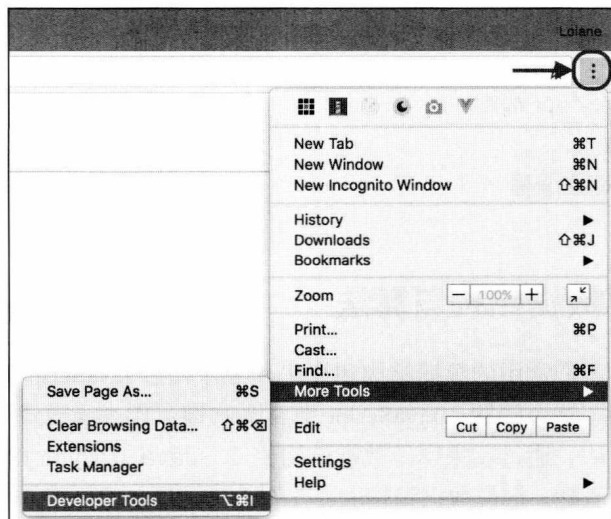
1.2 环境搭建

相比其他语言，JavaScript 的优势之一在于不用安装或配置任何复杂的环境就可以开始学习。每台计算机上都已具备所需的环境，哪怕使用者从未写过一行代码。有浏览器足矣！

为了运行书中的示例代码，建议你做好如下准备：安装 Chrome 或 Firefox 浏览器（选择一个你最喜欢的即可），选择一个喜欢的编辑器（如 Visual Studio Code），以及一个 Web 服务器（XAMPP 或其他你喜欢的，这一步是可选的）。Chrome、Firefox、VS Code 和 XAMPP 在 Windows、Linux 和 Mac OS 上均可以使用。

1.2.1 最简单的环境搭建

浏览器是最简单的 JavaScript 开发环境。现代浏览器（Chrome、Firefox、Safari 和 Edge）都拥有一个叫作开发者工具的功能。如要使用 Chrome 中的开发者工具，可以点击右上角的菜单，选择 More Tools | Developer Tools，如下图所示。



打开开发者工具，里面有一个 **Console** 标签页，可以在其中编写你的 JavaScript 代码，如下图所示（需要按下 **Enter** 键来执行源代码）。



1.2.2 使用 Web 服务器

你可能想要安装的第二个环境也很简单，但是需要安装一个 Web 服务器。如果一个 HTML 文件只包含简单的、不向服务器发送任何请求的 JavaScript 代码（Ajax 调用），那么你可以右键点击它并选择在浏览器中直接打开。本书中需要编写的代码都很简单，可以通过这种方式执行。但是，安装一个 Web 服务器总是有好处的。

有很多开源和免费的 Web 服务器可供选择。如果你熟悉 PHP 的话，XAMPP 会是不错的选择，它可用于 Linux、Windows 和 Mac OS。

由于我们会专注于服务端和浏览器上的 JavaScript，可以在 Chrome 上安装一个简单的 Web 服务器，它是一个叫作 **Web Server for Chrome** 的扩展。安装好之后，可以在浏览器地址栏中输入 `chrome://apps` 来找到它。

