

TURING

图灵程序设计丛书

[PACKT]
PUBLISHING

【巴西】Loiane Groner 著 孙晓博 邓钢 吴双 陈迪 袁源 译

学习JavaScript 数据结构与算法

Learning JavaScript Data Structures and Algorithms



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



【巴西】Loiane Groner 著 孙晓博 邓钢 吴双 陈迪 袁源 译

学习JavaScript 数据结构与算法

Learning JavaScript Data Structures and Algorithms

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 / (巴西) 格罗纳
(Groner, L.) 著 ; 孙晓博等译. — 北京 : 人民邮电出
版社, 2015. 10

(图灵程序设计丛书)
ISBN 978-7-115-40414-5

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

中国版本图书馆CIP数据核字(2015)第218878号

内 容 提 要

本书首先介绍了JavaScript语言的基础知识, 接下来讨论了数组、栈、队列、链表、集合、字典、散列表、树、图等数据结构, 之后探讨了各种排序和搜索算法, 包括冒泡排序、选择排序、插入排序、归并排序、快速排序、顺序搜索、二分搜索, 最后还介绍了动态规划和贪心算法等常用的高级算法及相关知识。

本书适用于前端 Web 开发人员, 以及所有对 JavaScript 数据结构与算法感兴趣的读者。

-
- ◆ 著 [巴西] Loiane Groner
 - 译 孙晓博 邓 钢 吴 双 陈 迪 袁 源
 - 责任编辑 岳新欣
 - 执行编辑 仇祝平 李舒扬
 - 责任印制 杨林杰
 - ◆ 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: 10.75
 - 字数: 254千字 2015年10月第1版
 - 印数: 1-3 500册 2015年10月河北第1次印刷
- 著作权合同登记号 图字: 01-2015-3277号
-

定价: 39.00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

前 言

JavaScript是当下最流行的编程语言。由于浏览器的原生支持(无需安装任何插件), JavaScript也被称作“互联网语言”。JavaScript的应用非常广泛, 不仅被用于前端开发, 也被用到服务器(Node.js)和数据库(MongoDB)环境中。

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, 我们要能够用编程语言和数据结构来解决问题。编程语言和数据结构是这些问题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。

快乐地编码吧!

本书结构

第1章“JavaScript简介”, 讲述了JavaScript的基础知识, 它们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“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3章“栈”, 介绍了栈这种数据结构, 示范了如何创建栈, 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4章“队列”, 详述了队列这种数据结构, 演示了如何创建队列, 如何添加和删除队列中的元素, 还讨论了如何用队列解决计算机科学中常见的问题, 以及栈和队列的主要区别。

第5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6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7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8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。如果想更深入地学习树（包括相关的算法），这一章还给出了一些建议。

第9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现方式，探讨了如何使用深度优先算法和广度优先算法遍历图，以及图的适用范围。

第10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。另外还介绍了搜索算法中的顺序搜索和二分搜索。

第11章“算法补充知识”，主要讨论其他一些常用的算法和著名的大 O 表示法。这一章讲解了什么是递归，介绍了一些高级算法，如动态规划和贪心算法，还介绍了大 O 表示法和相关概念。最后，讨论了如何进一步学习算法的相关知识。

附录详尽列出了书中所授算法的复杂度列表（使用大 O 表示法）。

准备工作

为学习本书，你可以设置三种不同的开发环境。你不需要设置所有这三种环境，可以选择其一，也可以逐一尝试。

方法一，你需要一个浏览器，请在下面列表中选择其一：

- Chrome (<https://www.google.com/chrome/browser/>)
- Firefox (<https://www.mozilla.org/en-US/firefox/new/>)

方法二，你需要：

- 安装方法一中的任意一个浏览器；
- 安装一个Web服务器。如果你电脑里没有安装过Web服务器，推荐安装XAMPP (<https://www.apachefriends.org>)。

方法三，如果想安装一个纯JavaScript的环境，你需要完成下面几步。

- 安装步骤一中的任意浏览器

- 安装Node.js (<http://nodejs.org/>)
- 安装好Node.js后, 安装http-server开发包:

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

第1章还会对此进行更详细的介绍。

读者对象

本书的目标读者包括计算机科学专业的学生、刚刚开启职业生涯的技术人员, 以及想学习基于JavaScript语言的数据结构和算法的朋友。如果想学好书中的数据结构和算法, 编程知识和逻辑思维是必需的。

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, 但想在JavaScript语言中使用它们的人所写。

排版约定

在本书中, 你会发现一些不同的文本样式, 用以区别不同种类的信息。下面举例说明。

正文中的代码、用户输入这样表示: “在script标签里, 编写JavaScript代码。”

代码段的格式如下:

```
console.log("num: " + num);
console.log("name: " + name);
console.log("trueValue: " + trueValue);
console.log("price: " + price);
console.log("nullVar: " + nullVar);
console.log("und: " + und);
```

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:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
</head>
<body>
  <script>
    alert('Hello, World!');
  </script>
</body>
</html>
```

所有的命令行输入或输出的格式如下:

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

新术语和重点词汇以楷体标示。屏幕、目录或对话框上的内容这样表示：“Node Packages Modules (<https://www.npmjs.org/>) 也在呈指数级增长。”



这个图标表示警告或需要特别注意的内容。



这个图标表示提示或者技巧。

读者反馈

欢迎提出反馈。如果你对本书有任何想法，喜欢它什么，不喜欢它什么，请让我们知道。要写出真正对大家有帮助的书，了解读者的反馈很重要。

一般的反馈，请发送电子邮件至feedback@packtpub.com，并在邮件主题中包含书名。

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，请参考我们的作者指南<http://www.packtpub.com/authors>。

客户支持

现在，你是一位自豪的Packt图书的拥有者，我们会尽全力帮你充分利用你手中的书。

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。

下载彩色插图

我们还提供了一份PDF文档，里面是本书中用到的彩色截图和图表。这些彩图能帮你更好地理解输出的变化。你可以从https://www.packtpub.com/sites/default/files/downloads/4874OS_ColoredImages.pdf下载。

勘误表

虽然我们已尽力确保本书内容正确，但出错仍旧在所难免。如果你在我国的书中发现错误，不管是文本还是代码，希望能告知我们，我们不胜感激。这样做可以减少其他读者的困扰，帮助我们改进本书的后续版本。如果你发现任何错误，请访问<http://www.packtpub.com/submit-errata>提交，选择你的书，点击勘误表提交表单的链接，并输入详细说明。勘误一经核实，你的提交将被接受，此勘误将上传到本公司网站或添加到现有勘误表。从<http://www.packtpub.com/support>选择书名就可以查看现有的勘误表。

侵权行为

互联网上的盗版是所有媒体都要面对的问题。Packt非常重视保护版权和许可证。如果你发现我们的作品在互联网上被非法复制，不管以什么形式，都请立即为我们提供位置地址或网站名称，以便我们可以寻求补救。

请把可疑盗版材料的链接发到copyright@packtpub.com。

非常感谢你帮助我们保护作者，以及保护我们给你带来有价值内容的能力。

问题

如果你对本书内容存有疑问，不管是哪个方面，都可以通过questions@packtpub.com联系我们，我们将尽最大努力来解决。

目 录

第 1 章 JavaScript 简介	1	3.2 从十进制到二进制	38
1.1 环境搭建	1	3.3 小结	39
1.1.1 浏览器	2	第 4 章 队列	40
1.1.2 使用 Web 服务器 (XAMPP)	3	4.1 创建队列	40
1.1.3 使用 Node.js 搭建 Web 服务器	4	4.1.1 完整的 Queue 类	42
1.2 JavaScript 基础	6	4.1.2 使用 Queue 类	43
1.2.1 变量	7	4.2 优先队列	44
1.2.2 操作符	8	4.3 循环队列——击鼓传花	46
1.2.3 真值和假值	11	4.4 小结	47
1.2.4 相等操作符 (==和===)	12	第 5 章 链表	48
1.3 控制结构	13	5.1 创建一个链表	49
1.3.1 条件语句	14	5.1.1 向链表尾部追加元素	50
1.3.2 循环	15	5.1.2 从链表中移除元素	52
1.4 函数	16	5.1.3 在任意位置插入一个元素	54
1.5 面向对象编程	16	5.1.4 实现其他方法	56
1.6 调试工具	18	5.2 双向链表	58
1.7 小结	18	5.2.1 在任意位置插入一个新元素	59
第 2 章 数组	19	5.2.2 从任意位置移除元素	61
2.1 为什么用数组	19	5.3 循环链表	64
2.2 创建和初始化数组	20	5.4 小结	64
2.3 添加和删除元素	21	第 6 章 集合	65
2.4 二维和多维数组	24	6.1 创建一个集合	65
2.5 JavaScript 的数组方法参考	26	6.1.1 has(value) 方法	66
2.5.1 数组合并	27	6.1.2 add 方法	66
2.5.2 迭代器函数	27	6.1.3 remove 和 clear 方法	67
2.5.3 搜索和排序	28	6.1.4 size 方法	68
2.5.4 输出数组为字符串	31	6.1.5 values 方法	69
2.6 小结	32	6.1.6 使用 Set 类	69
第 3 章 栈	33	6.2 集合操作	70
3.1 栈的创建	33		

6.2.1 并集	70	9.2.1 邻接矩阵	112
6.2.2 交集	71	9.2.2 邻接表	113
6.2.3 差集	72	9.2.3 关联矩阵	114
6.2.4 子集	73	9.3 创建图类	114
6.3 小结	74	9.4 图的遍历	116
第 7 章 字典和散列表	75	9.4.1 广度优先搜索	117
7.1 字典	75	9.4.2 深度优先搜索	122
7.1.1 创建一个字典	75	9.5 小结	128
7.1.2 使用 Dictionary 类	78	第 10 章 排序和搜索算法	129
7.2 散列表	79	10.1 排序算法	129
7.2.1 创建一个散列表	79	10.1.1 冒泡排序	130
7.2.2 使用 HashTable 类	81	10.1.2 选择排序	133
7.2.3 散列表和散列集合	82	10.1.3 插入排序	134
7.2.4 处理散列表中的冲突	82	10.1.4 归并排序	135
7.2.5 创建更好的散列函数	90	10.1.5 快速排序	138
7.3 小结	91	10.2 搜索算法	142
第 8 章 树	92	10.2.1 顺序搜索	143
8.1 树的相关术语	92	10.2.2 二分搜索	143
8.2 二叉树和二叉搜索树	93	10.3 小结	145
8.2.1 创建 BinarySearchTree 类	94	第 11 章 算法补充知识	146
8.2.2 向树中插入一个键	95	11.1 递归	146
8.3 树的遍历	98	11.1.1 JavaScript 调用栈大小的 限制	147
8.3.1 中序遍历	98	11.1.2 斐波那契数列	147
8.3.2 先序遍历	99	11.2 动态规划	149
8.3.3 后序遍历	100	11.3 贪心算法	152
8.4 搜索树中的值	101	11.4 大 O 表示法	153
8.4.1 搜索最小值和最大值	101	11.4.1 理解大 O 表示法	153
8.4.2 搜索一个特定的值	103	11.4.2 时间复杂度比较	155
8.4.3 移除一个节点	104	11.5 用算法娱乐身心	156
8.5 更多关于二叉树的知识	108	11.6 小结	157
8.6 小结	109	附录 A 时间复杂度速查表	158
第 9 章 图	110	致谢	160
9.1 图的相关术语	110		
9.2 图的表示	112		

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，也是网络应用里最卓越的语言之一。在GitHub（世界上最大的代码托管站点，<https://github.com>）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://goo.gl/ZFx6mg>），并且还在逐年增长。

JavaScript不仅可用于前端开发，也适用于后端开发，Node.js就是这样一种技术。Node包（<http://www.npmjs.org/>）的数量也呈指数级增长。

要成为一名Web开发工程师，掌握JavaScript必不可少。

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或Java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是只为C或Java这样的语言而生？在前端开发当中，你可能也需要实现它们。

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，这对你今后的代码质量至关重要（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、eBay等），数据结构和算法是面试问题的重头戏。

1.1 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Sublime Text），以及一个Web服务器（XAMPP或其他你喜欢的，这一步是可选的）。这些软件在Windows、Linux和Mac OS上均可以使用。

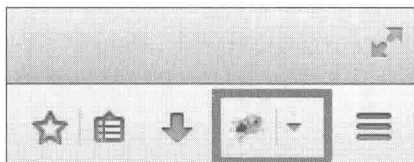
如果你使用Firefox，推荐你安装Firebug插件（<https://getfirebug.com>）。

接下来将介绍搭建环境的三种方案。

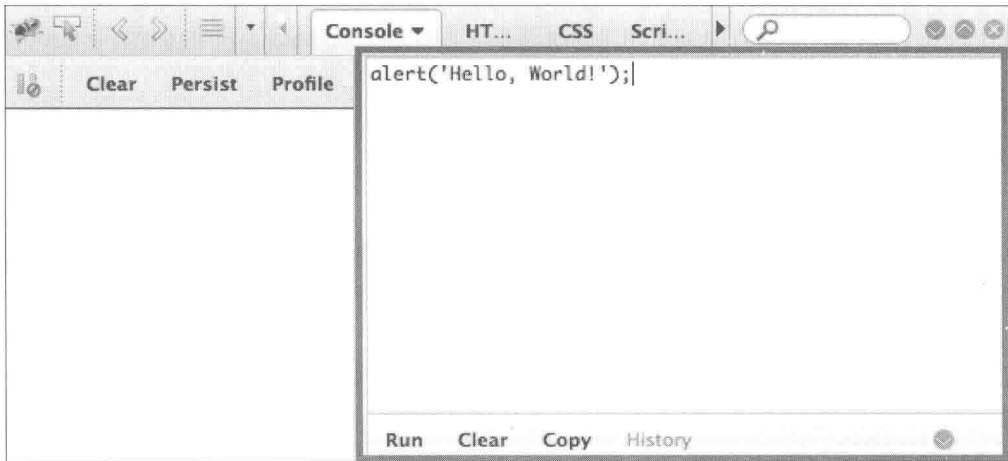
1.1.1 浏览器

浏览器是最简单的开发环境。

你也可以使用Firefox加Firebug。安装好Firebug后，在浏览器的右上角会看到如下图所示的图标。

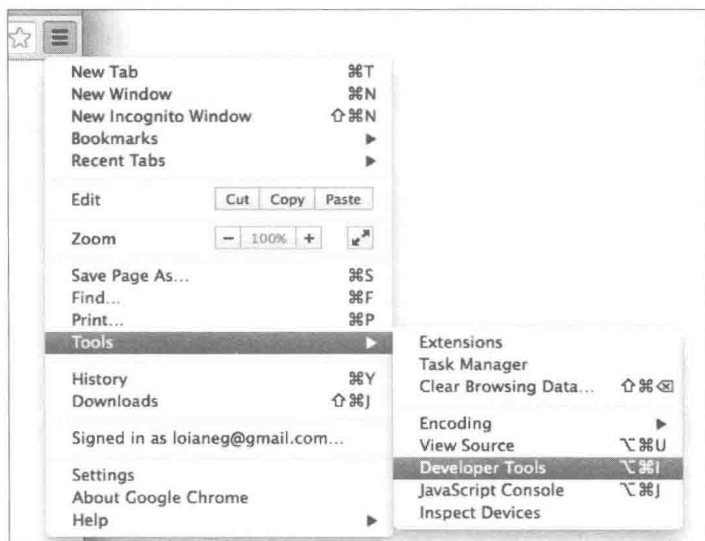


点击Firebug图标，打开它，可以看到Console标签，我们可以在其命令行区域中编写所有JavaScript代码，如下图所示（执行源代码请按Run按钮）。



也可以扩展命令行，来适应Firebug插件的整个可用区域。

你还可以使用谷歌Chrome，它已经集成了Google Developer Tools（谷歌开发者工具）。打开Chrome，点击设置及控制图标，选中Tools|Developer Tools，如下图所示。



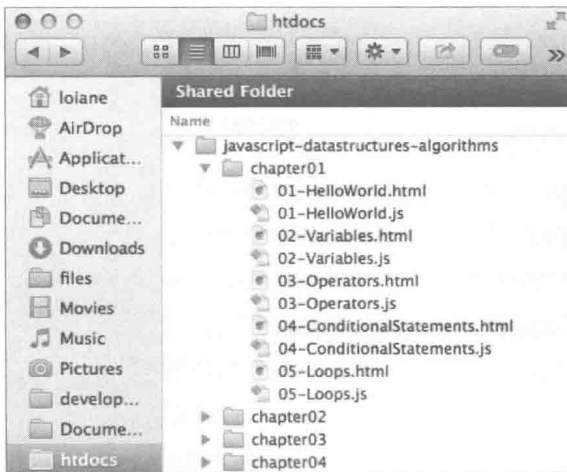
然后，就可以在Console标签页中编写JavaScript测试代码，如下所示。



1.1.2 使用Web服务器（XAMPP）

你可能想要安装的第二个环境是XAMPP，它的安装过程也很简单，但比只使用浏览器麻烦点儿。

安装XAMPP(<https://www.apachefriends.org>)或者你偏爱的其他Web服务器。然后，在XAMPP安装文件夹下找到htdocs目录。在该目录下新建一个文件夹，就可以在里面执行本书中所讲述的源代码；或者是直接将示例代码下载后提取到此目录，如下所示。



接下来，在启动XAMPP服务器之后，你就可以通过localhost这个URL，用浏览器访问源码，如下图所示（别忘了打开Firebug或谷歌开发者工具查看输出）。



执行示例代码时，请不要忘记打开谷歌开发者工具或Firebug查看输出结果。

1.1.3 使用Node.js搭建Web服务器

第三种选择就是100%的JavaScript，我们可以使用Node.js来搭建一个JavaScript服务器，不使用XAMPP搭建的Apache服务器。

首先要到<http://nodejs.org/>下载和安装Node.js。然后，打开终端应用（如果你用的是Windows操作系统，打开Node.js的命令行），输入如下命令：


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

最好手动输入这些命令，复制粘贴可能会出错。

也可以用管理员身份执行上述命令。对于Linux和Mac操作系统，使用如下命令：

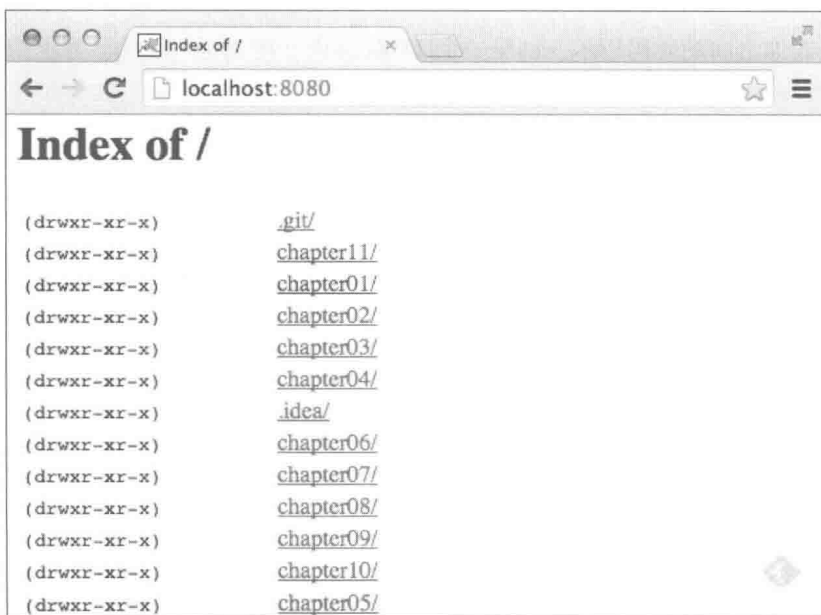
```
sudo npm install http-server -g
```

这条命令会在你的机器上安装一个JavaScript服务器：`http-server`。要启动服务器并在终端应用上运行本书中的示例代码，请将工作路径更改至示例代码文件夹，然后输入`http-server`，如下图所示，整个环境就搭建好了！



```
loianeg:~ loiane$ cd /Users/loiane/Documents/javascript-datastructures-algorithms
loianeg:javascript-datastructures-algorithms loiane$ http-server
Starting up http-server, serving ./ on port: 8080
Hit CTRL-C to stop the server
```

为执行示例，打开浏览器，通过`http-server`命令指定的端口访问：



```
Index of /
localhost:8080
Index of /
(drwxr-xr-x)  .git/
(drwxr-xr-x)  chapter11/
(drwxr-xr-x)  chapter01/
(drwxr-xr-x)  chapter02/
(drwxr-xr-x)  chapter03/
(drwxr-xr-x)  chapter04/
(drwxr-xr-x)  .idea/
(drwxr-xr-x)  chapter06/
(drwxr-xr-x)  chapter07/
(drwxr-xr-x)  chapter08/
(drwxr-xr-x)  chapter09/
(drwxr-xr-x)  chapter10/
(drwxr-xr-x)  chapter05/
```

下载示例代码

在官网（<http://www.packtpub.com>）购买的所有Packt图书，均可以下载到对应的示例代码。如果你并非在官网购买的本书，请访问<http://www.packtpub.com/support>并注册你的邮箱，对应的代码文件就会发送给你。

本书的代码也可以在GitHub上找到，资源库地址为：<https://github.com/loiane/javascriptdatastructures-algorithms>。



1.2 JavaScript 基础

在深入学习各种数据结构和算法前,让我们先大概了解一下JavaScript。本节教大家一些相关的基础知识,有利于学习后面各章。

首先来看在HTML中编写JavaScript的两种方式:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
</head>
<body>
  <script>
    alert('Hello, World!');
  </script>
</body>
</html>
```

第一种方式如上面的代码所示。创建一个HTML文件,把代码写进去。在这个例子里,我们在HTML中声明了script标签,然后把JavaScript代码都写进这个标签。

第二种方式,我们需要创建一个JavaScript文件(比如01-HelloWorld.js),在里面写入如下代码:

```
alert('Hello, World!');
```

然后,我们的HTML文件看起来如下:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
</head>
<body>
  <script src="01-HelloWorld.js">
  </script>
</body>
</html>
```

第二个例子展示了如何将一个JavaScript文件引入HTML文件。

这两个例子,无论执行哪个输出都是一样的。但第二个例子是最佳实践。



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。作为最佳实践,我们会在关闭body标签前引入JavaScript代码。这样浏览器就会在加载脚本之前解析和显示HTML,有利于提升页面的性能。

1.2.1 变量

变量保存的数据可以在需要时设置、更新或提取。赋给变量的值都有对应的类型。JavaScript的类型有数字、字符串、布尔值、函数和对象。还有undefined和null，以及数组、日期和正则表达式。下面的例子介绍如何在JavaScript里使用变量。

```
var num = 1; //{1}
num = 3; //{2}

var price = 1.5; //{3}
var name = 'Packt'; //{4}
var trueValue = true; //{5}
var nullVar = null; //{6}
var und; //{7}
```


在行{1}，我们展示了如何声明一个JavaScript变量（声明了一个数字类型）。虽然关键字var不是必需的，但最好每次声明一个新变量时都加上。

在行{2}，我们更新了已有变量。JavaScript不是强类型语言。这意味着你可以声明一个变量并初始化为一个数字类型的值，然后把它更新成字符串或者其他类型的值，不过这并不是一个好做法。

在行{3}，我们又声明了一个数字类型的变量，不过这次是十进制浮点数。在行{4}，声明了一个字符串；在行{5}，声明了一个布尔值；在行{6}，声明了一个null；在行{7}，声明了undefined变量。null表示变量没有值，undefined表示变量已被声明，但尚未赋值：

```
console.log("num:" + num);
console.log("name:" + name);
console.log("trueValue:" + trueValue);
console.log("price:" + price);
console.log("nullVar:" + nullVar);
console.log("und:" + und);
```

如果想看我们声明的每个变量的值，可以用console.log来实现，就像上面代码片段中那样。



书中示例代码会使用三种方式输出JavaScript的值。第一种是alert('My text here')，将输出到浏览器的警示窗口；第二种是console.log('My text here')，将把文本输出到调试工具的Console标签（谷歌开发者工具或是Firebug，根据你使用的浏览器而定）；第三种方式是直接输出到HTML页面里并被浏览器呈现，通过document.write('My text here')。可以选择你喜欢的方式来调试。

console.log方法能接收多个参数，除了console.log("num: " + num)还可以写成console.log("num: ", num)。