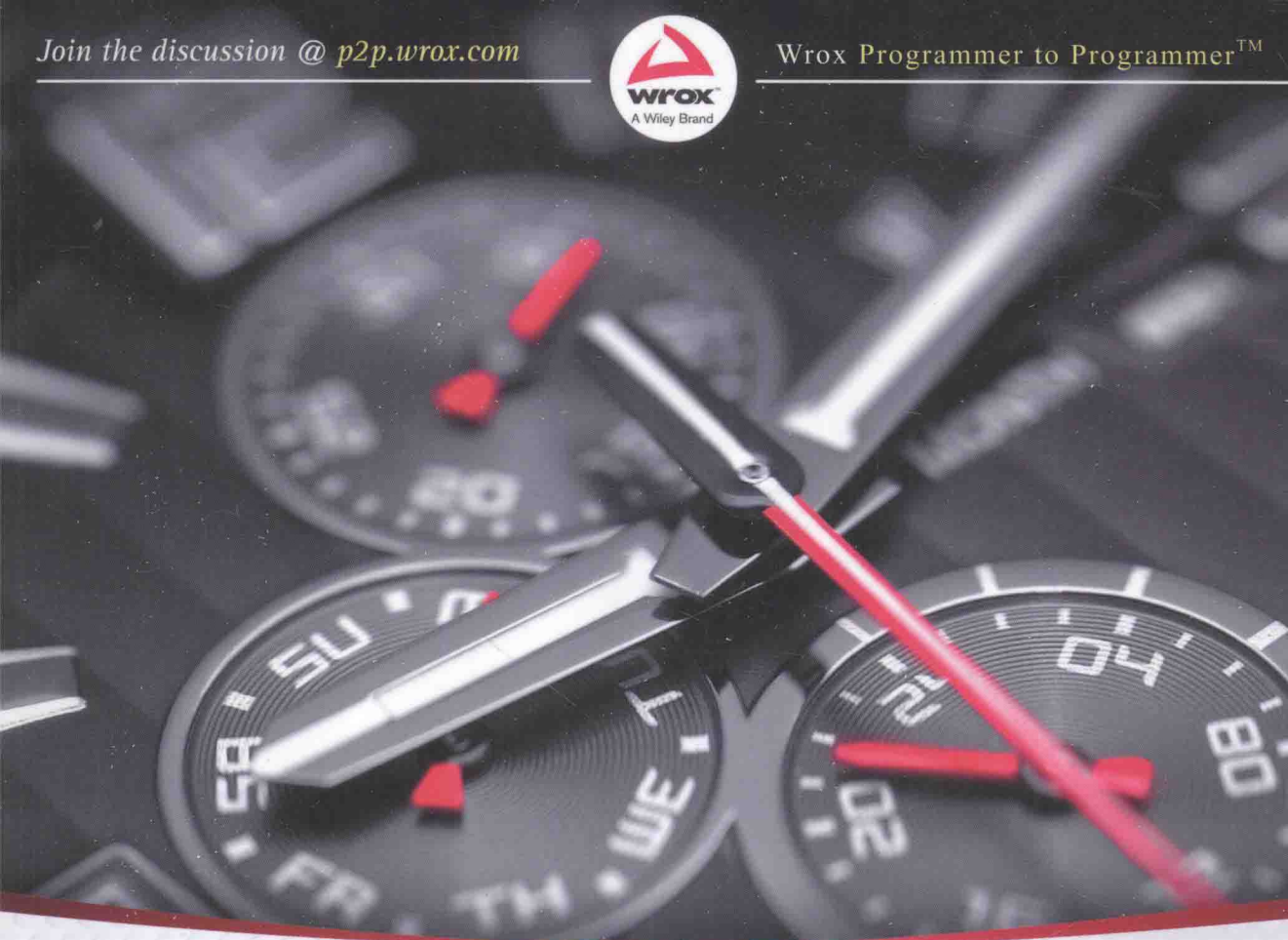
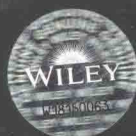




Beginning JavaScript, Fifth Edition

JavaScript 入门经典 (第5版)

[美] Jeremy McPeak 著
Paul Wilton
胡献慧 译



清华大学出版社

JavaScript 入门经典

(第 5 版)

[美] Jeremy McPeak 著
Paul Wilton

胡献慧 译

清华大学出版社

北 京

Jeremy McPeak, Paul Wilton
Beginning JavaScript, Fifth Edition
EISBN: 978-1-118-90333-9

Copyright © 2015 by John Wiley & Sons, Inc., Indianapolis, Indiana
All Rights Reserved. This translation published under License.

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. JavaScript is a registered trademark of Oracle, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2015-3020

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

JavaScript 入门经典: 第 5 版 / (美) 麦克皮克(McPeak, J.), (美) 威尔顿(Wilton, P.) 著; 胡献慧 译. —北京: 清华大学出版社, 2016

书名原文: Beginning JavaScript, Fifth Edition

ISBN 978-7-302-41952-5

I. ①J… II. ①麦… ②威… ③胡… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 263117 号

责任编辑: 王 军 于 平

装帧设计: 牛静敏

责任校对: 成凤进

责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 43.5 字 数: 1059 千字

版 次: 2016 年 1 月第 1 版 印 次: 2016 年 1 月第 1 次印刷

印 数: 1~3500

定 价: 98.00 元

产品编号: 064912-01

译者序

JavaScript 是 Web 开发中应用最早、发展最成熟、用户最多的脚本语言。其语法简洁，代码可读性在众多脚本语言中最好，它在使用时不用考虑数据类型，是真正意义上的动态语言。JavaScript 通过提供动态的、个性化交互式内容，来增强静态 Web 应用程序的功能。JavaScript 使访问站点的用户能够享受到更美妙的体验，增强了网站对用户的吸引力。现在，美观的下拉菜单、滚动文字和动态内容已经广泛应用于各种 Web 站点，这一切都是通过 JavaScript 来实现的。

各种主流浏览器都支持 JavaScript，JavaScript 已成为从事 Web 开发的首选脚本语言。数百万计的网页使用 JavaScript 来改进设计、验证表单、检测浏览器以及创建 cookies。JavaScript 可用于 HTML 和 Web，更可广泛用于笔记本电脑、平板电脑和智能手机等设备。

JavaScript 是通向程序设计世界的大门，学习和理解了本书的基础知识，就可以进一步学习更新、更高级的编程技术。读者不必为没有编程经验而担心，因为本书以初学者为核心，全面介绍了使用 JavaScript 进行网站开发的各种技术。在内容编排上由浅入深，让读者循序渐进地掌握编程技术；在内容讲解上结合丰富的图解和形象的比喻，帮助读者理解“晦涩难懂”的技术；在内容形式上附有大量的提示、技巧、说明等栏目，夯实读者编程基础，丰富其编程经验。在每一章的最后还附有习题，以便加深读者对基本概念的理解。

本书对上一版本做了全面更新，使 JavaScript 代码适用于最新版本的 IE、Firefox 和 Safari 浏览器。本书的大部分代码都是跨浏览器兼容的，对于不能跨浏览器兼容的情况，本书会专门指出。

本书适合 JavaScript 的初学者、Web 系统开发人员、对 Ajax 技术感兴趣的人员、网站开发人员、使用 Web 技术进行毕业设计的计算机学员、想了解最新流行的客户端 Web 技术的开发人员，也可以作为各种培训学校、职业学校及大中专院校的教材。

本书全部章节由胡献慧翻译，参与翻译的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。

作者简介

Jeremy McPeak 是一位自学成才的程序员，他自 1998 年开始开发网站。他编写了 *JavaScript 24-Hour Trainer* (Wiley 2010)，与他人合著了 *Professional Ajax, 2nd Edition* (Wiley 2007)。他还在 Tuts+ Code(<http://code.tutsplus.com>)上发表文章、提供视频教程以及 JavaScript、C#和 ASP.NET 课程。可以通过 p2p 论坛、其网站(<http://www.wdonline.com>)和 Twitter (@jwmcpeak)联系 Jeremy。

Paul Wilton 最初是英国国防部的一位 Visual Basic 应用程序员，之后进入 .NET 领域。他加入了 Internet 开发公司，花 3 年时间帮助创建了 Internet 解决方案。他现在经营自己的公司，公司很成功，发展也很快，主要开发在线假日房间预订系统。

致 谢

首先，我要感谢上帝对我的青睐，还要感谢亲爱的读者，没有你们，本书就不可能面世。我也要感谢我的家庭成员们，容忍我利用周末闲暇时间修订本书。

编写、出版一本书需要许多人的努力，这里不可能提及所有提供帮助的人。但我要特别感谢 Jim Minatel 和 Robert Elliott 同意本书的开发。感谢 Kelly Talbot 促使本书的写作走上正轨，感谢编辑团队对文本的修饰，感谢 Russ Mullen 的支持。

— Jeremy McPeak

首先非常感谢我的搭档 Beci，因为本书已完成，所以我与他一周的见面时间将不超过 10 分钟。

我还要感谢本书的编辑，他的工作非常高效，使本书得以顺利付印。

感谢 Jim Minatel 使本书有了面世的机会。

还要感谢支持、鼓励我多年写作的所有人，他们的帮助我会铭记在心。

最后，感谢我的德国牧羊犬 Katie，它很好地挡住了挨家挨户销售的人员对我的干扰。

— Paul Wilton

前 言

JavaScript 是一门脚本语言，它通过提供动态的、个性化的交互式内容，来增强静态 Web 应用程序的功能。JavaScript 使访问站点的用户能够享受到更美妙的体验，增强了网站对用户的吸引力。现在，美观的下拉菜单、滚动的文字和动态内容已经广泛应用于各种网站，这一切都是通过 JavaScript 来实现的。各种主流浏览器都支持 JavaScript，JavaScript 已经成为从事 Web 开发的首选脚本语言。另外，JavaScript 语言也可用于 Web 应用程序之外的其他场合，例如可用于自动化管理任务。

本书旨在介绍使用 JavaScript 进行开发的基础知识，即 JavaScript 的含义，JavaScript 代码是如何运行的，以及使用 JavaScript 能够实现哪些功能等。本书将首先介绍 JavaScript 的基本语法，然后介绍如何创建功能强大的 Web 应用程序。读者不必为没有编程经验而担心，本书将循序渐进地介绍所有相关知识。JavaScript 是通向程序设计世界的大门，学习和理解了本书的基础知识，就可以进一步学习更新、更高级的编程技术。

本书读者对象

为了最好地汲取本书中的知识，读者应该对 HTML 和 CSS 有所了解，并知道如何创建静态的 Web 页面。除此之外，读者不必拥有任何编程经验。

本书同样适合于具有编程经验、且希望学习 Web 程序设计的读者。这些读者可能比较了解计算机知识，但未必掌握 Web 技术。

另外，一些读者具备设计背景，但对计算机知识和 Web 技术不大了解。那么，对于这类读者而言，JavaScript 可以作为一个进入编程和 Web 应用程序开发世界的快捷通道。

对于所有的读者，我都希望本书物有所值。

本书涵盖的内容

本书首先介绍 JavaScript 的含义，以及 JavaScript 的基础语法。然后详细介绍程序设计的基本概念，包括数据、数据类型以及选择语句和循环语句等结构化程序设计的概念。

熟悉这些基础知识之后，本书将介绍 JavaScript 的一个重要概念——对象，讨论如何利用 JavaScript 的内置对象，如函数、日期和字符串等，来管理复杂的数据，简化程序的设计。本书还将介绍如何使用 JavaScript 操作浏览器提供的对象并对浏览器进行探讨。

随后，本书将介绍更高级的主题，例如编写动态操作 Web 页面元素的代码，并在页面上有某行为发生时执行相应的代码。还将介绍如何脚本化表单和其他控件。运用这些知识，

就可以创建专业水准的 Web 应用程序，并与用户交互。

之后，本书介绍如何将数据存储到浏览器中并直接与服务器进行通信。还介绍如何为新的 HTML5 媒体元素编写代码，以及如何为这些元素编写自定义的用户界面。

本书还探讨一些省时的 JavaScript 框架，例如 jQuery、Modernizr、Prototype 和 MooTools，了解它们的工作原理，以及它们如何帮助创建复杂而强大的 JavaScript 应用程序。

最后，本书介绍一些常见的语法错误和逻辑错误，还介绍如何找到这些错误，以及如何使用针对 Chrome、Internet Explorer、Firefox、Safari 和 Opera 的 JavaScript 调试器来帮助找出错误。本书还介绍了如何处理漏掉的错误，并确保这些错误不会对应用程序最终用户的体验造成不良影响。

本书介绍的所有新概念都用实例加以说明。通过这些实例可以对所学的 JavaScript 原理进行实践，以巩固所学的知识。

本书末尾有 4 个附录，附录 A 是本书各章末尾习题的答案，其他附录包含内容丰富且极富价值的参考资料。附录 B 是 JavaScript 语言的核心参考，附录 C 是完整的 W3C DOM 核心参考——还包括 HTML DOM 和 DOM Level 2 事件模型的信息，附录 D 是 Latin-1 字符集的十进制和十六进制字符码。

如何使用本书

由于 JavaScript 代码是基于文本的技术，因此要创建 JavaScript 程序，只需一个文本编辑器即可。

另外，为了测试本书中的代码，还需要一个支持较新 JavaScript 版本的浏览器。理想情况下，这意味着最好使用 Chrome、Internet Explorer、Firefox、Safari 和 Opera 的最新版本。本书代码在这些浏览器中进行了详细的测试。不过，本书的代码应该可以在当今的任何 Web 浏览器中工作。对于不能跨浏览器兼容的情况，本书会专门指出。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但错误总是难免的，如果你在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者节省时间、避免阅读和学习受挫，当然，这还有助于提供更高质量的书籍。请给 wkservice@vip.163.com 发电子邮件，我们就会检查你的信息，如果是正确的，就把它发送到该书的勘误表页面上，或在后续版本中采用。

要在网站上找到本书的勘误表，可以登录 www.wrox.com，通过 Search 框或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

如果读者没有在 Book Errata 页面上找到自己发现的错误，那么请转到页面 <http://www.wrox.com/contact/techsupport.shtml>，针对你所发现的每一项错误填写表格，并将表格发给

我们,我们将对表格内容进行认真审查,如果确实是我们书中的错误,我们将在该书的 Book Errata 页面上标明该错误信息,并在该书的后续版本中改正。

p2p.wrox.com

P2P 邮件列表是为作者和读者之间的讨论而建立的。读者可以在 p2p.wrox.com 上加入 P2P 论坛。该论坛是一个基于 Web 的系统,用于传送与 Wrox 图书相关的信息和相关技术,与其他读者和技术用户交流。该论坛提供了订阅功能,当论坛上有新帖子时,会给你发送你选择的主题。Wrox 作者、编辑和其他业界专家和读者都会在这个论坛上进行讨论。

在 <http://p2p.wrox.com> 上有许多不同的论坛,帮助读者阅读本书,在读者开发自己的应用程序时,也可以从这个论坛中获益。要加入这个论坛,需执行下面的步骤:

- (1) 进入 p2p.wrox.com, 单击 Register 链接。
- (2) 阅读其内容, 单击 Agree 按钮。
- (3) 提供加入论坛所需的信息及愿意提供的可选信息, 单击 Submit 按钮。
- (4) 然后就会收到一封电子邮件, 其中的信息描述了如何验证账户, 完成加入过程。



提示: 不加入 P2P 也可以阅读论坛上的信息, 但只有加入论坛后, 才能发送自己的信息。

加入论坛后, 就可以发送新信息, 回应其他用户的帖子。可以随时在 Web 上阅读信息。如果希望某个论坛给自己发送新信息, 可以在论坛列表中单击该论坛对应的 Subscribe to this Forum 图标。

对于如何使用 Wrox P2P 的更多信息, 可阅读 P2P FAQ, 了解论坛软件的工作原理, 以及许多针对 P2P 和 Wrox 图书的常见问题的解答。要阅读 FAQ, 可以单击任意 P2P 页面上的 FAQ 链接。

源代码

学习本书中的示例时, 可以手工输入所有的代码, 也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 www.wrox.com 上下载。登录到站点 www.wrox.com, 使用 Search 框或书名列表就可以找到本书, 接着单击本书细目页面上的 Download Code 链接, 就可以获得所有的源代码。

另外, 可以登录 www.tupwk.com.cn/download, 输入中文书名或中文 ISBN 下载源代码。此外, 可以登录 <http://beginningjs.com> 查看本书中的示例。



提示: 许多图书的书名都很相似, 所以通过 ISBN 查找本书是最简单的, 本书的英文原版的 ISBN 是 978-1-118-90333-9。

下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 www.wrox.com/dynamic/books/download.aspx 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

目 录

第 1 章 JavaScript 与 Web 概述.....1	2.5 数组.....38
1.1 JavaScript 简介.....1	2.6 小结.....46
1.1.1 JavaScript 的含义.....1	2.7 习题.....46
1.1.2 JavaScript 与 Web.....2	第 3 章 决策与循环.....49
1.1.3 JavaScript 的功能.....3	3.1 决策语句——if 和 switch
1.1.4 创建 JavaScript Web 应用	语句.....50
程序所需的工具.....4	3.1.1 比较运算符.....50
1.2 脚本的使用场合.....6	3.1.2 if 语句.....52
1.2.1 链接外部 JavaScript 文件.....6	3.1.3 逻辑运算符.....56
1.2.2 使用外部文件的优点.....7	3.1.4 在 if 语句中使用多个条件.....58
1.3 第一个简单的 JavaScript 程序.....7	3.1.5 else 和 else if.....61
1.4 编写更多的 JavaScript 程序.....9	3.1.6 字符串的比较.....63
1.5 浏览器和兼容性问题简述.....13	3.1.7 switch 语句.....63
1.6 小结.....14	3.2 循环——for 语句和 while
第 2 章 数据类型与变量.....15	语句.....68
2.1 JavaScript 中的数据类型.....15	3.2.1 for 循环.....68
2.1.1 数值数据.....16	3.2.2 for...in 循环.....71
2.1.2 文本数据.....16	3.2.3 while 循环.....72
2.1.3 布尔数据.....17	3.2.4 do...while 循环.....73
2.2 变量——存储在内存中的	3.2.5 break 和 continue 语句.....74
数据.....18	3.3 小结.....75
2.2.1 创建变量并赋值.....19	3.4 习题.....76
2.2.2 用其他变量的值为	第 4 章 函数与作用域.....79
变量赋值.....21	4.1 创建自定义函数.....80
2.3 使用数据——计算数值及	4.2 作用域和生存期.....83
基本字符串操作.....23	4.2.1 全局作用域.....83
2.3.1 数值计算.....23	4.2.2 函数作用域.....84
2.3.2 运算符的优先级.....27	4.2.3 标识符查找.....84
2.3.3 基本的字符串操作.....31	4.3 将函数用作值.....85
2.3.4 字符串与数值的混合操作.....33	4.4 小结.....88
2.4 数据类型转换.....34	4.5 习题.....89

第 5 章 JavaScript——基于对象的语言

第 5 章 JavaScript——基于对象的语言	91
5.1 基于对象的程序设计	91
5.1.1 对象的含义	92
5.1.2 JavaScript 中的对象	92
5.1.3 使用 JavaScript 对象	93
5.1.4 基本数据类型与对象	95
5.2 JavaScript 的内置对象类型	96
5.2.1 String 对象	96
5.2.2 Array 对象	106
5.2.3 Math 对象	116
5.2.4 Number 对象	123
5.2.5 toFixed()方法	124
5.2.6 Date 对象	125
5.3 创建自定义对象	133
5.4 创建对象的新类型(引用类型)	136
5.5 小结	139
5.6 习题	140
第 6 章 字符串操作	141
6.1 新的字符串方法	142
6.1.1 split()方法	142
6.1.2 replace()方法	144
6.1.3 search()方法	144
6.1.4 match()方法	144
6.2 正则表达式	145
6.2.1 简单的正则表达式	146
6.2.2 正则表达式: 特殊字符	148
6.2.3 包含所有的可能性	156
6.2.4 正则表达式的分组	156
6.2.5 重用字符组	158
6.3 String 对象	159
6.3.1 split()方法	160
6.3.2 replace()方法	161
6.3.3 search()方法	164
6.3.4 match()方法	164
6.4 使用 RegExp 对象的构造函数	167

6.4.1 验证电话号码	169
6.4.2 验证邮政编码	171
6.4.3 验证电子邮件地址	172
6.4.4 验证域名	173
6.5 小结	174
6.6 习题	175

第 7 章 日期、时间和计时器

7.1 世界时	178
7.2 网页中的计时器	183
7.2.1 一次性计时器	183
7.2.2 创建定期触发的计时器	185
7.3 小结	187
7.4 习题	187

第 8 章 浏览器程序设计

8.1 浏览器对象简介	190
8.1.1 window 对象	191
8.1.2 history 对象	192
8.1.3 location 对象	192
8.1.4 navigator 对象	193
8.1.5 screen 对象	197
8.1.6 document 对象	197
8.1.7 使用 document 对象	197
8.1.8 images 集合	200
8.1.9 links 集合	201
8.2 确定用户的浏览器	201
8.2.1 特性检测	202
8.2.2 浏览器嗅探	204
8.3 小结	208
8.4 习题	209

第 9 章 编写 DOM 脚本

9.1 Web 标准	212
9.1.1 HTML	214
9.1.2 ECMAScript	214
9.2 文档对象模型	215
9.2.1 DOM 标准	215
9.2.2 DOM 与 BOM 的区别	216
9.2.3 将 HTML 文档表示为树型结构	217

9.2.4 DOM 核心对象	219	11.5 习题	359
9.2.5 DOM 对象及其属性和方法	220	第 12 章 JSON	361
9.3 操作 DOM	238	12.1 XML	362
9.3.1 访问元素	238	12.2 JSON	363
9.3.2 改变元素的外观	238	12.2.1 简单值	363
9.3.3 定位和移动内容	242	12.2.2 对象	364
9.3.4 示例: 动态广告	243	12.2.3 数组	365
9.4 小结	247	12.2.4 串行化为 JSON	366
9.5 习题	247	12.2.5 解析 JSON	366
第 10 章 事件	249	12.3 小结	370
10.1 事件的类型	250	12.4 习题	370
10.2 将代码连接到事件	250	第 13 章 数据存储	371
10.3 标准事件模型	260	13.1 烘焙第一个 cookie	372
10.4 旧版本 IE 中的事件处理	274	13.1.1 新鲜出炉的 cookie	372
10.5 编写跨浏览器的代码	282	13.1.2 cookie 字符串	381
10.6 内置拖放操作	292	13.2 创建 cookie	384
10.6.1 使内容可以拖动	293	13.3 获取 cookie 的值	387
10.6.2 创建释放目标	294	13.4 cookie 的局限性	392
10.6.3 传输数据	299	13.4.1 用户可能禁用 cookie	392
10.7 小结	307	13.4.2 数字和信息的限制	393
10.8 习题	308	13.5 cookie 的安全性和 IE	394
第 11 章 HTML 表单: 与用户交互	309	13.6 Web 存储	395
11.1 HTML 表单	309	13.6.1 设置数据	396
11.2 传统 Form 对象的属性和方法	311	13.6.2 获取数据	396
11.2.1 表单中的 HTML 元素	312	13.6.3 删除数据	397
11.2.2 共有的属性和方法	313	13.6.4 把数据存储为字符串	397
11.2.3 按钮元素	315	13.6.5 查看 Web 存储的内容	400
11.2.4 文本元素	318	13.7 小结	401
11.2.5 textarea 元素	324	13.8 习题	401
11.2.6 复选框和单选按钮	327	第 14 章 Ajax	403
11.2.7 选择框	336	14.1 Ajax 的含义	403
11.3 HTML5 表单对象的属性和方法	347	14.1.1 Ajax 的作用	404
11.3.1 新的输入类型	347	14.1.2 浏览器支持	405
11.3.2 新元素	351	14.2 使用 XMLHttpRequest 对象	405
11.4 小结	357	14.2.1 创建 XMLHttpRequest 对象	406
		14.2.2 使用 XMLHttpRequest 对象	406

14.2.3	异步请求	408	17.1.2	Modernizr 的 API	478
14.3	创建简单的 Ajax 模块	409	17.1.3	自定义测试	479
14.3.1	规划 HttpRequest 模块	409	17.1.4	加载资源	480
14.3.2	HttpRequest 构造函数	409	17.2	Prototype	485
14.3.3	创建 send() 方法	411	17.2.1	获得 Prototype	485
14.3.4	完整的代码	411	17.2.2	测试 Prototype 安装	486
14.4	使用 Ajax 验证表单字段	412	17.2.3	检索元素	488
14.4.1	请求信息	413	17.2.4	处理样式	489
14.4.2	接收到的数据	413	17.2.5	创建、插入和删除元素	490
14.4.3	准备工作	413	17.2.6	使用事件	491
14.5	注意事项	422	17.2.7	用 Prototype 重写 选项卡	492
14.5.1	安全性问题	422	17.2.8	使用 Ajax 支持	494
14.5.2	可用性问题	423	17.3	MooTools	501
14.6	小结	430	17.3.1	获得 MooTools	501
14.7	习题	430	17.3.2	测试 MooTools 安装	501
第 15 章	HTML5 媒体	433	17.3.3	查找元素	503
15.1	入门	434	17.3.4	修改样式	504
15.2	给媒体编写脚本	437	17.3.5	创建、插入和删除元素	505
15.2.1	方法	437	17.3.6	使用事件	506
15.2.2	属性	440	17.3.7	用 MooTools 重写 选项卡	507
15.2.3	事件	446	17.3.8	MooTools 中的 Ajax 支持	510
15.3	小结	450	17.4	小结	516
15.4	习题	451	17.5	习题	517
第 16 章	jQuery	453	第 18 章	常见错误、调试和 错误处理	519
16.1	获得 jQuery	454	18.1	一些常见错误	519
16.2	jQuery 的 API	455	18.1.1	未经定义的变量	520
16.2.1	选择元素	455	18.1.2	区分大小写	521
16.2.2	修改样式	457	18.1.3	不匹配的大括号	522
16.2.3	创建、追加和删除元素	460	18.1.4	不匹配的圆括号	522
16.2.4	处理事件	461	18.1.5	赋值(=)而不是相等(=)	523
16.2.5	用 jQuery 重写选项卡	463	18.1.6	将方法和属性混为一谈	523
16.2.6	把 jQuery 用于 Ajax	466	18.1.7	在连接字符串时未使用 加号(+)	524
16.3	小结	473	18.2	错误处理	525
16.4	习题	474			
第 17 章	其他 JavaScript 库	475			
17.1	Modernizr	476			
17.1.1	获得 Modernizr	477			

18.2.1 避免错误.....	525	18.3.4 在 Safari 中调试.....	550
18.2.2 try...catch 语句	526	18.4 小结.....	552
18.3 调试	533	18.5 习题.....	553
18.3.1 在 Chrome (和 Opera)中 调试	534	附录 A 参考答案.....	555
18.3.2 在 Internet Explorer 中 调试	543	附录 B JavaScript 核心参考.....	623
18.3.3 在 Firefox 中用 Firebug 调试	547	附录 C W3C DOM 参考	647
		附录 D Latin-1 字符集	671

第 1 章

JavaScript 与 Web 概述

本章主要内容：

- 将 JavaScript 添加到 Web 页面
- 引用外部 JavaScript 文件
- 改变 Web 页面的背景色

本章源代码下载(wrox.com)：

打开网页 <http://www.wiley.com/go/BeginningJavaScript5E>，单击 Download Code 选项卡即可下载本章源代码。也可以在 <http://beginningjs.com> 上查看所有的代码示例和相关的文件。

本章介绍 JavaScript 的含义及功能，以及开发 JavaScript 程序所需的工具。了解这些基础知识之后，本书其余章节将逐步介绍如何使用 JavaScript 为 Web 站点创建功能强大的 Web 应用程序。

实践是最好的学习方式。本书将使用 JavaScript 创建大量实用的示例程序，而本章是该过程的一个起点，将创建第一段 JavaScript 代码。

1.1 JavaScript 简介

本节简要介绍 JavaScript 的含义、发展历程、工作原理以及功能。

1.1.1 JavaScript 的含义

在购买本书之前，你也许已经知道 JavaScript 是一种计算机语言。但是，什么是计算机语言？简言之，计算机语言就是一系列告诉计算机做某件事的指令。其中，“某件事”可以是各种操作，包括显示文本、移动图片或者请求用户输入信息等。通常，指令(也称为代码)

自上而下地处理。简单而言, 计算机阅读编写好的代码, 确定要执行的操作, 然后执行该操作。处理代码的过程称为“运行代码”或“执行代码”。

在自然语言中, 为了冲一杯速溶咖啡, 可以写出如下指令或代码:

- (1) 将咖啡放入杯中。
- (2) 在水壶中加水。
- (3) 用水壶烧水。
- (4) 如果水已煮沸, 则将水倒入咖啡杯, 否则继续等水煮沸。
- (5) 品尝咖啡。

在执行这些指令时, 从第一行(指令 1)开始执行, 接着执行第二行(指令 2), 然后是下一行, 直到结束。这就是大多数计算机语言的执行方式, JavaScript 亦是如此。但是有时可能需要改变代码执行的流程, 甚至跳过某些语句, 详见第 3 章。

JavaScript 是一种解释型语言, 而不是编译型语言。什么是解释型语言与编译型语言呢?

计算机并不真正理解 JavaScript。计算机需要解释 JavaScript 代码, 并将其转换成计算机能理解的机器码。因此, JavaScript 是一种解释型语言。计算机只能理解机器码, 机器码实际上是一串二进制数字(即 0 和 1 组成的字符串)。当浏览器遇到 JavaScript 时, 就将 JavaScript 代码传递给一个称为“解释器”的程序, 解释器将 JavaScript 代码转换为计算机能理解的机器码。这有点类似于请一个翻译将英语翻译为西班牙语。注意, JavaScript 的转换在代码运行时进行, 每次运行都需要重复地进行转换。JavaScript 并非唯一的解释型语言, PHP 和 Ruby 也都是解释型语言。

对于编译型语言, 代码在程序运行之前转换成机器码, 而且这个转换过程只执行一次。程序员使用“编译器”将写好的代码转换成机器码, 这些机器码由程序的用户运行。编译型语言包括 C#、Java 等。这有点类似于一个翻译将一份西班牙语文档翻译成英文。除非改变了文档, 否则就可以重复使用翻译好的文档, 而不必重新翻译。

此处, 应该消除一个普遍存在的误解: JavaScript 并非 Java 语言的脚本版。事实上, 除了名字中都带有 Java 外, 二者并没有任何相同点。而且, JavaScript 比 Java 更容易学习和使用。实际上, JavaScript 是所有语言中最简单的语言, 且具有令人叹服的强大功能。

1.1.2 JavaScript 与 Web

本书中的大部分 JavaScript 代码都运行在由浏览器加载的 Web 页面中。创建 Web 页面只需要一个文本编辑器, 例如 Windows 记事本。浏览 Web 页面需要一个浏览器, 例如 Chrome、Firefox 或者 Internet Explorer(IE), 这些浏览器都内置了 JavaScript 的解释器(通常称为 JavaScript 引擎)。

注意: 本书中, 当提到微软的 Internet Explorer 浏览器时, 术语“IE”和“Internet Explorer”可以互换。

实际上, JavaScript 语言最初出现在 Netscape Navigator 2 浏览器中。当时它叫 LiveScript。

然而, 由于当时 Java 技术如日中天, Netscape 公司觉得改为 JavaScript 这个名字会更引人注目。随着 JavaScript 的发展, 微软公司决定在 Internet Explorer 3 浏览器中加入微软品牌的 JavaScript, 即 JScript。

1997 年, JavaScript 由 Ecma International(一个会员制的非盈利组织)标准化且被更名为 ECMAScript。现在的浏览器制造商都以 ECMAScript 为标准, 在各自的浏览器中包含了 JavaScript 引擎, 但这并不是意味着所有浏览器都必须支持相同的特性。现代浏览器对 JavaScript 的支持比以前更趋于一致, 但如你在后续章节中所见, 开发者仍需处理一些以前的且在大多数情况下非标准的 JavaScript 实现。

ECMAScript 标准控制着 JavaScript 语言的各个方面, 帮助确保不同 JavaScript 版本的相互兼容性。但是, ECMA 为 JavaScript 语言制订标准时, 没有规定如何在特定的主机上使用它。这里的“主机(host)”是指主机环境, 在本书中就是 Web 浏览器。其他主机环境包括 PDF 文件、Web 服务器等许多其他地方。本书仅讨论该语言在 Web 浏览器上的使用。制订网页标准的组织是 World Wide Web Consortium(W3C), 它不仅为 HTML 和 CSS 制订了标准, 还为 JavaScript 在 Web 浏览器上如何与网页交互制订了标准。后面章节将详细介绍这些内容。在学习更高级的内容之前, 先讨论 JavaScript 的基本内容。本书的附录列出了 JavaScript 语言及其与 Web 浏览器交互的有用指南。

可以将本书创建的大多数包含 JavaScript 代码的网页存储在硬盘上, 并可以将它们直接从硬盘加载到浏览器中, 就像加载普通的文件(如文本文件)一样。但是, 浏览因特网上的网站时, 浏览器却不是这样加载网页的。因特网是一个庞大的计算机互连网络。网站访问是因特网上特定计算机提供的一种专门服务, 提供网站访问服务的计算机称为 Web 服务器。

Web 服务器的基本功能是在其硬盘上保存大量的网页。通常, 另一台计算机上的浏览器请求一个保存在 Web 服务器上的网页时, Web 服务器将从硬盘载入该网页, 并且通过一个称为超文本传输协议(Hypertext Transfer Protocol, HTTP)的专用通信协议, 将网页回传给发出请求的计算机。运行 Web 浏览器并发出请求的计算机称为客户机。客户机/服务器的关系有点像顾客与售货员的关系, 顾客进入商店并对售货员说:“请给我拿某一件东西”。售货员就为顾客提供服务, 找到顾客想要的东西并递给顾客。对于 Web 来说, 运行 Web 浏览器的客户机就是顾客, 而提供所请求网页的 Web 服务器就是售货员。

在 Web 浏览器中输入一个网址时, 浏览器如何知道向哪个 Web 服务器请求页面呢? 商店有“某市中央大街 45 号”这样的地址, Web 服务器也有地址, 但它的地址没有街道名, 而是使用 IP(Internet Protocol, 因特网协议)地址, IP 地址唯一标识了因特网上的 Web 服务器。IP 地址由句点(“.”)分隔的 4 组数字组成, 例如: 127.0.0.1。

如果用户曾在网上冲浪, 可能会对上述内容感到不解。因为 Web 站点的名字是 www.somewebsite.com, 而不是 IP 地址。实际上, 名称 www.somewebsite.com 是实际 IP 地址的一个用户友好名字, 是为了便于人们记忆。域名服务器(domain name server)可以将用户友好的名字转换为实际的 IP 地址, 域名服务器由因特网服务提供商(Internet service provider, ISP)建立。

1.1.3 JavaScript 的功能

JavaScript 最主要的用途是与用户交互。这是一个相当宽泛的定义, 所以下面将“与用