

JavaScript & Ajax Sixth Edition

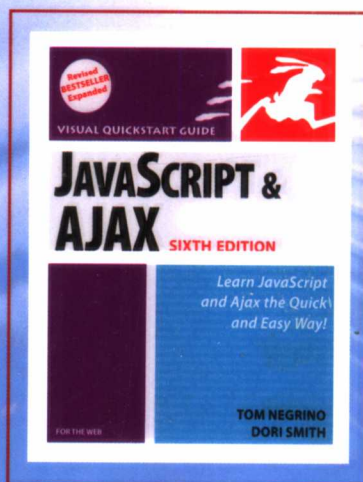
JavaScript 基础教程

(第6版)

原版销售累计超过
100 000册

[美] Tom Negrino Dori Smith 著
陈剑瓯 等译

- 经典 JavaScript 入门书，
涵盖 Ajax
- 透彻讲解 Web 开发最新理念
- 让你体验轻松迅速的学习方式



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Web开发系列

JavaScript基础教程

(第6版)

JavaScript & Ajax

Sixth Edition

[美] Tom Negrino Dori Smith 著

陈剑瓯 等译

人民邮电出版社

北京

图书在版编目 (CIP) 数据

JavaScript 基础教程: 第 6 版 / (美) 内格里诺 (Negrino, T.) (美) 史密斯 (Smith, D.) 著; 陈剑瓯等译. —北京: 人民邮电出版社, 2007.9
(图灵程序设计丛书)

ISBN 978-7-115-16359-2

I. J... II. ①内...②史...③陈... III. JAVA 语言—程序设计—教材 IV. TP312
中国版本图书馆 CIP 数据核字 (2007) 第 083640 号

内 容 提 要

本书循序渐进地讲述了 JavaScript 及相关的 CSS、DOM 与 Ajax 等技术。书中从 JavaScript 语言基础开始, 分别讨论了图像、框架、浏览器窗口、表单、正则表达式、用户事件和 cookie, 还有两章讲述了 Ajax 基础。本书不仅有对于基础知识和使用方法的介绍, 也包含了对 JavaScript 应用示例的深入探讨。

本书适合初级和中级层次的 Web 应用开发人员和 Web 设计师阅读。

图灵程序设计丛书

JavaScript 基础教程 (第 6 版)

-
- ◆ 著 [美] Tom Negrino Dori Smith
译 陈剑瓯 等
责任编辑 傅志红 张继发
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 800 × 1000 1/16
印张: 23 75
字数: 561 千字 2007 年 9 月第 1 版
印数: 1-5 000 册 2007 年 9 月北京第 1 次印刷

著作权合同登记号 图字: 01-2007-1963 号

ISBN 978-7-115-16359-2/TP

定价: 45.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

版 权 声 明

Authorized translation from the English language edition entitled *JavaScript & Ajax For The Web, Sixth Edition* by Tom Negrino, Dori Smith, published by Pearson Education, Inc., publishing as Peachpit Press, Copyright © 2005 by Peachpit Press.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by an information storage retrieval system, without permission of Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by POST & TELECOM PRESS Copyright © 2007.

本书中文简体字版由美国 Pearson Education Inc 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。

前 言

欢迎你随我们学习JavaScript! 通过使用这种容易掌握的程序设计语言, 你可以给网页增加一些趣味性, 使它对你和站点的访问者更有意义。本书是一本轻松的JavaScript入门教程, 所以读者在学习过程中不需要自己输入脚本。正如我们的一位朋友所说的, “麻烦的事由我们来做, 这样你就不必麻烦了!”

本书读者对象

我们猜想既然你有兴趣学习JavaScript, 那么你肯定有创建HTML页面和Web站点的经验, 而且希望更进一步, 给站点添加一些交互性。我们并不假设你了解关于编程或脚本编程的任何知识。我们也不假设你是HTML专家(当然, 如果你是HTML专家更好)。我们只假设你具备构建网页的基本知识, 而且熟悉常用的HTML标签, 比如链接、图像和表单。

在某些章首题为“目前需要了解的HTML知识”的表中, 我们对HTML做了一些解释。并非每章都有这部分内容, 只有在我们认为你需要快速参考的地方才会提供。有了这些HTML信息, 你就不需要在阅读本书的同时, 在另一本书中查找HTML属性的语法了。我们还假设你对CSS (Cascading Style Sheet, 层叠样式表) 有一定的了解, 但是如果你以前没有使用过CSS, 也没有关系, 我们会提供阅读本书所需的所有信息。

如果你对编程有所了解, 你应该会注意到我们介绍JavaScript的方式与其他图书的风格很不一样。我们并不深入介绍JavaScript的语法和结构, 而且本书不是全面的语言参考书(但是, 在附录A中可以找到一些有价值的参考资料)。市面上已经有一些书提供了全面的参考资料, 我们在附录D中列出了它们。这些书和本书之间的差异是, 我们并不拘泥于形式, 而是集中地演示如何用JavaScript完成一些有用的任务, 而不赘述大量的无关信息。

在本书的这个最新版本中, 我们新增了对Ajax的介绍; 这种技术结合使用了JavaScript和其他常用Web技术为网页增加交互性, 并且改善了Web站点的用户体验。我们提供Ajax基础知识和实际示例, 帮助你给站点增加Ajax功能, 而不需要深入学习Web编程。

如何使用本书

在本书中, 我们采用一些特殊的版式, 帮助你更轻松地理解本书和JavaScript本身。

分步骤的说明组成了本书的大部分内容。我们在其中以特殊的字体样式表示HTML或JavaScript代码, 比如:

```
<div align="center">  
window.onload = initLinks;
```

你还会注意到，HTML和JavaScript代码都显示为小写。我们这么做是因为，这个版本中的所有脚本都符合W3C（World Wide Web Consortium，万维网联盟）的XHTML 1.0过渡型（transitional）标准。当在JavaScript中看到引号时，总是直引号（'或"），而不是弯曲的引号（‘或“）。弯曲的引号会使JavaScript失效，在编写脚本时应该避免使用这样的引号。

在与分步说明对应的脚本中，我们以粗体字突出显示脚本中当前正在讨论的部分，这样你就能够马上找到我们正在讨论的代码。我们还常常在Web浏览器窗口的屏幕图中以灰度底纹突出显示其中某些重要的部分。

因为图书的页面比计算机屏幕窄，所以一些JavaScript代码行在页面上排不开。当出现这种情况时，我们将代码行分为多行，在前面插入一个箭头→表示这是续行，并且将续行缩进。下面是我们处理长代码行的示例：

```
dtString = "Hey, just what are you  
→ doing up so late?";
```

关于浏览器

在这个版本中有一个大的变化：我们不再支持非常陈旧的浏览器，以及在支持Web标准方面做得很差的浏览器。自上一版本以来，几乎所有Web用户都升级到了现代浏览器，这些浏览器在支持公认的Web标准（比如XHTML、CSS2和DOM）方面做得很好。这包括IE 6或更高版本，Firefox 1.0或更高版本，Netscape 7或更高版本，Safari的所有版本，以及Opera 7或更高版本。

如果你需要支持老式浏览器，就需要某些比较老式的技术。在以前的版本中，我们讨论了这些浏览器专用技术，但是从这一版开始我们决定完全放弃老式代码，将这些信息从书中删除。但是，我们不想完全丢掉这些内容，所以将它们组织成第18章，这一章以可下载的PDF形式放在本书的配套Web站点上。请访问www.javascriptworld.com，点击Resources链接，就可以找到第18章的文件。在这个Web站点上逛逛，你可能会找到其他许多有意思的内容。

我们（以及帮助我们的一些测试人员）在几种操作平台上的许多种浏览器中测试了我们的脚本，包括Windows XP、Mac OS X和Ubuntu Linux。在本书出版时，Windows Vista还没有最终发布，而且公共beta版本仍然很不稳定，所以这个版本没有使用Vista。

我们使用最主流的浏览器——微软Windows版本的IE 6——测试了本书中的所有内容，而且使用IE 7的公共beta版本做了大量测试（本书原版是在IE 7最终版本发布之前出版的）。我们还用Windows、Mac和Linux平台上的Firefox 1.5对脚本做了测试。当然，我们没有忽视Mac用户；所有脚本都在Safari 2中做了测试。因为Safari是从WebKit引擎衍生出来的，而且源自开源的KHTML呈现引擎；所以这意味着，这些脚本在从WebKit引擎衍生出来的其他浏览器中也应该能够正常工作，比如Omni Group的OmniWeb，在基于KHTML引擎的浏览器中也可以正常工作，比如Konqueror for Linux。

不必输入代码

一些JavaScript图书只在书中给出了脚本，你在实践时必须自己输入代码。但是我们认为这种

方式已经过时了。我们不得不完成这些艰苦的输入工作，但是你不必重复这些劳动。我们为本书提供了一个配套的Web站点，其中包含本书中的所有脚本，你可以将这些脚本复制和粘贴到自己的网页中。这个站点还包含其他提示和脚本。如果我们在书中发现了任何错误，也会在这里列出更正。这个配套站点的网址是：<http://www.javascriptworld.com/>。

如果由于某种原因你打算输入某些脚本示例，那么可能会发现这些示例似乎不起作用，这是因为你没有创建这些示例所用的支持文件。例如，在图像上实现屏幕效果的示例中，你需要图像文件。这没问题。我们将这些文件都放在本书的Web站点上了，而且打包成适合下载的形式。你会找到一个可下载文件，其中包含所有脚本、HTML文件以及所需的所有媒体文件。如果你遇到了任何问题，请查看配套Web站点上的FAQ（常见问题）。

如果阅读了FAQ，而你的问题没有在那里得到回答，那么可以通过js6@javascriptworld.com给我们发邮件。很抱歉，因为我们会收到大量邮件，所以我们不可能回答发送到我们个人邮件地址的关于本书的电子邮件。但是，我们可以保证发送到js6@javascriptworld.com的邮件会得到答复。

开始吧

关于JavaScript最好的一点是它很容易入门，可以用一个简单的脚本在网页上实现很酷的效果，然后根据需要逐渐添加更复杂的素材。你不必等到学完整本书之后再开始改进自己的网页。你可以一边学习本书，一边用Ajax给站点添加高级的交互效果。

当然，任何旅程都是从第一步开始的，现在就跟随我们开始探索JavaScript和Ajax的旅程吧！

致谢

特别感谢我们的编辑Nancy Davis：她的专业精神、温暖的关怀和强烈的奉献精神都促使我们把工作做得更好。

感谢Tracey Croom出色的生产工作。

我们衷心感谢本书的排版人员Danielle Foster，他在异常紧迫的时间压力下仍然优雅从容地完成了工作。

我们要感谢Peachpit的Nancy Ruenzel提供的支持。

感谢我们的儿子Sean，感谢他能够理解父母承受的压力，经常自己照顾自己。

我们要感谢采用本书以前的版本作为课程教材的所有高中、学院和大学教师。

Dori要感谢W&S小组的女士们，感谢她们的关怀。还要感谢Wise-Women Web Design社区(<http://www.wise-women.org>)作为行为榜样的耐心、支持和灵感。

Tom在写作时常常听着音乐，在写本书时也是这样。许多音乐给本书的第6版带来了动感和节奏，包括Wolfgang Mozart、Patty Griffin、Richard Shindell、KT Tunstall、Bruce Springsteen、Rosanne Cash、REM、Antonio Vivaldi、Pandora Radio的作品和iTunes上的其他音乐。

目 录

第 1 章 了解 JavaScript 1	2.9 使用 JavaScript 改进链接..... 28
1.1 JavaScript 是什么..... 1	2.10 处理引用者页面..... 30
1.2 JavaScript 不是 Java..... 2	第 3 章 语言基础 34
1.3 JavaScript 的起源..... 3	3.1 用循环进行重复操作..... 34
1.4 JavaScript 可以做什么..... 3	3.2 将值传递给函数..... 39
1.5 JavaScript 不能做什么..... 4	3.3 探测对象..... 40
1.6 Ajax 是什么..... 4	3.4 处理数组..... 42
1.7 组合式 (snap-together) 语言..... 6	3.5 处理有返回值的函数..... 44
1.7.1 对象..... 6	3.6 更新数组..... 45
1.7.2 属性..... 7	3.7 使用 do/while 循环..... 46
1.7.3 方法..... 7	3.8 以多种方式调用脚本..... 48
1.7.4 将这些成分组合在一起..... 7	3.9 使用多级条件..... 50
1.7.5 DOM 简介..... 8	3.10 处理错误..... 53
1.8 处理事件..... 8	第 4 章 处理图像 55
1.9 值和变量..... 9	4.1 创建翻转器..... 55
1.10 赋值和比较..... 10	4.2 创建更有效的翻转器..... 57
1.10.1 赋值..... 10	4.3 构建三状态翻转器..... 63
1.10.2 比较..... 11	4.4 由链接触发翻转器..... 65
1.11 编写对 JavaScript 友好的 HTML..... 11	4.5 让多个链接触发一个翻转器..... 67
1.11.1 结构、表示和行为..... 12	4.6 处理多个翻转器..... 70
1.11.2 div 和 span..... 12	4.7 创建循环的广告条..... 73
1.11.3 class 和 id..... 12	4.8 在循环广告条中添加链接..... 76
1.12 要使用什么工具..... 13	4.9 建立循环式幻灯片..... 78
第 2 章 开始 15	4.10 显示随机图像..... 80
2.1 将脚本放在哪里..... 15	4.11 从随机的起点开始循环显示图像..... 82
2.2 关于函数..... 17	第 5 章 框架 84
2.3 使用外部脚本..... 17	5.1 防止页面显示在框架中..... 85
2.4 在脚本中编写注释..... 19	5.2 迫使页面显示在框架中..... 86
2.5 向用户发出警告..... 20	5.3 迫使站点显示在框架中..... 88
2.6 确认用户的选择..... 22	5.4 设置目标..... 91
2.7 提示用户..... 24	5.5 创建和装载动态框架..... 92
2.8 用链接对用户进行重定向..... 25	

5.6	在框架之间共享函数	94	9.1.2	onunload 事件	181
5.7	在框架中存储信息	96	9.1.3	onresize 事件	181
5.8	同时装载多个框架	99	9.1.4	onmove 事件	183
5.9	处理 iframe	100	9.1.5	onabort 事件	183
5.10	用 JavaScript 装载 iframe	103	9.1.6	onerror 事件	183
第 6 章	操作浏览器窗口	104	9.1.7	onfocus 事件	183
6.1	打开新窗口	104	9.1.8	onblur 事件	184
6.2	将不同的内容装载进窗口中	108	9.2	鼠标事件处理	185
6.3	打开多个窗口	109	9.2.1	onmousedown 事件	186
6.4	同时打开多个窗口	110	9.2.2	onmouseup 事件	188
6.5	从一个窗口更新另一个窗口	112	9.2.3	onmousemove 事件	188
6.6	用 JavaScript 创建新页面	114	9.2.4	onmouseover 事件	191
6.7	关闭窗口	116	9.2.5	onmouseout 事件	191
6.8	在指定的位置打开窗口	119	9.2.6	ondblclick 事件	191
6.9	将窗口移动到指定的位置	121	9.2.7	onclick 事件	192
第 7 章	表单处理	124	9.3	表单事件处理	192
7.1	选择并转移导航菜单	125	9.3.1	onsubmit 事件	193
7.2	动态地改变菜单	128	9.3.2	onreset 事件	193
7.3	建立必须填写的字段	130	9.3.3	onchange 事件	193
7.4	根据其他字段对字段进行检查	135	9.3.4	onselect 事件	193
7.5	标识出有问题的字段	137	9.3.5	onclick 事件	193
7.6	准备进行表单检验	139	9.3.6	onblur 事件	193
7.7	处理单选按钮	143	9.3.7	onfocus 事件	195
7.8	用一个字段设置另一个字段	146	9.4	键事件处理	196
7.9	检验 Zip 编码	148	9.4.1	onkeydown 事件	196
7.10	检验电子邮件地址	152	9.4.2	onkeyup 事件	198
第 8 章	表单和正则表达式	158	9.4.3	onkeypress 事件	199
8.1	用正则表达式检验电子邮件地址	158	第 10 章	JavaScript 和 cookie	200
8.2	检验文件名	163	10.1	建立第一个 cookie	200
8.3	提取字符串	165	10.2	读取 cookie	203
8.4	对字符串进行格式化	168	10.3	显示 cookie	204
8.5	对字符串进行格式化和排序	172	10.4	使用 cookie 作为计数器	206
8.6	对字符串进行格式化和检验	174	10.5	删除 cookie	208
8.7	使用正则表达式替换元素	176	10.6	处理多个 cookie	210
第 9 章	处理事件	178	10.7	显示新内容提醒信息	211
9.1	窗口事件处理	178	第 11 章	CSS 简介	217
9.1.1	onload 事件	178	11.1	用样式控制外观	218
			11.2	通过类应用样式	221

11.3 用 CSS 修改字体	223	第 15 章 Ajax 简介	297
11.4 检查 id	224	15.1 Ajax 的定义	297
11.5 不同寻常的链接	226	15.2 读取服务器数据	299
11.6 在样式中嵌入样式	228	15.3 解析服务器数据	306
11.7 添加背景图像	230	15.4 刷新服务器数据	310
11.8 绝对定位	231	15.5 用 Ajax 预览链接	312
第 12 章 对象和 DOM	234	15.6 自动补全表单字段	316
12.1 关于节点操纵	234	第 16 章 Ajax 工具包	322
12.1.1 DOM-2 和 W3C	234	16.1 拖放页面元素	323
12.1.2 DOM-2 术语	234	16.2 在页面上添加日历	328
12.1.3 DOM-3	235	16.3 在页面上添加双月日历	332
12.2 添加节点	235	16.4 使用容器实用程序	338
12.3 删除节点	237	16.5 添加动画效果	342
12.4 删除特定的节点	239	16.6 实现用于调试的 Logger 控件	345
12.5 插入节点	242	第 17 章 bookmarklet	350
12.6 替换节点	245	17.1 第一个 bookmarklet	350
第 13 章 建立动态页面	249	17.2 改变页面的背景颜色	354
13.1 在网页上显示当前日期	249	17.3 改变页面样式	355
13.2 处理周中日	251	17.4 Web 安全颜色	356
13.3 根据时间对消息进行定制	252	17.5 查询单词	357
13.4 根据时区显示日期	253	17.6 查看图像	359
13.5 把 24 小时制转换为 12 小时制	257	17.7 显示 ISO Latin 字符	361
13.6 创建倒数计数器	259	17.8 将 RGB 值转换为十六进制	361
13.7 移动文档中的对象	263	17.9 对值进行转换	363
13.8 日期方法	266	17.10 bookmarklet 计算器	364
第 14 章 JavaScript 应用示例	268	17.11 检验页面	365
14.1 使用可折叠菜单	268	附录 A JavaScript 的版本演化	
14.2 添加下拉菜单	271	和参考资料 (图灵网站下载)	
14.3 带说明的幻灯片	275	附录 B JavaScript 保留字 (图灵网站下载)	
14.4 一个无聊的姓名生成器	278	附录 C CSS 参考 (图灵网站下载)	
14.5 柱状图生成器	282	附录 D 其他学习资源 (图灵网站下载)	
14.6 样式表切换器	289	索引 (图灵网站下载)	

对于Web站点的开发者来说，HTML的演化是一件好坏参半的事儿。在Web时代的早期，HTML相当简单，很容易就能够掌握设计网页所需知道的一切。

随着Web的发展，页面设计人员的期望值也提高了，他们希望对页面的外观有更大的控制能力，这使HTML变得越来越复杂了。

因为Web是一种动态的媒体，页面设计人员还希望他们的页面能够与用户进行交互，HTML很快就显得不足以满足这一需求了。Netscape发明了JavaScript，作为控制浏览器和给网页添加活力和交互性的方法。

自从诞生之日起，JavaScript已经经历了很大程度的演化（尽管有时候根据浏览器的不同，演化的方向有所不同）。在本书后面，我们将详细讨论JavaScript的演化。

在本章中，你将学习JavaScript是什么（以及它不是什么）、它可以做什么（以及它不能做什么）和JavaScript语言的一些基础知识。还会向你介绍Ajax，这是JavaScript和其他技术的一种令人激动的组合，它在Web站点的交互性和创造性方面掀起了新的浪潮。

1

1.1 JavaScript 是什么

JavaScript是一种可以用来给网页增加交互性的编程语言。但是，如果你不是程序员，那么也不必担心；Web上有大量JavaScript代码，复制它们并稍做修改，就可以供自己使用。实际上，这种“站在其他程序员肩膀上”的方式正是熟悉JavaScript的好方法。

为了帮助你轻松地建立和运行JavaScript，我们建立了一个与本书配套的Web站点。我们在这个站点上提供本书中的所有脚本，以及额外的说明、附加资料和更正。可以在<http://www.javascript-world.com>上找到我们的站点。

常常会看到JavaScript被称为“脚本语言（scripting language）”，这暗示着它更适合编写脚本而不是程序。这实际上并没有根本性的差异。JavaScript脚本也是一种程序，它们包含在HTML页面内部（编写脚本的原始方式），或者驻留在外部文件中（现在的首选方法）。在HTML页面上，因为脚本文本包围在<script>标签中，所以它不会显示在用户的屏幕上，而Web浏览器知道应该运行JavaScript程序。<script>标签常常放在HTML页面的<head>部分中，但是如果愿意，也可以将脚本放在<body>部分中。向屏幕上写文本或编写HTML的内部脚本最好放在<body>部分中，就

像脚本1-1那样。如果你还不熟悉这些HTML概念，需要关于HTML的更多信息，那么我们建议读一下Elizabeth Castro的*HTML, XHTML, and CSS, Sixth Edition: Visual QuickStart Guide*¹。

脚本1-1 这个非常简单的脚本在浏览器窗口中输出“Hello, Cleveland!”

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Barely a script at all</title>
</head>
<body bgcolor="#FFFFFF">
<h1>
  <script language="Javascript" type="text/javascript">
    document.write("Hello, Cleveland!")
  </script>
</h1>
</body>
</html>
```

2

1.2 JavaScript 不是 Java

尽管名字中有Java，但是JavaScript和Java几乎没有什么共同点。Java是一种全功能的编程语言，是由Sun公司开发和推广的。Java是C和C++编程语言之后的又一种主流语言，程序员可以使用它创建完整的应用程序和控制消费电子设备。与其他语言不同，Java宣称具有跨平台兼容性；也就是说，程序员应该能够编写出可以在所有种类的机器上运行的Java程序，无论机器运行的是Windows、Mac OS X还是任何风格的UNIX。但实际上，Java不总是能够实现这个梦想，因为Sun和微软公司在这种语言的发展方向方面有很大的分歧。微软公司首先试图以自己的方式将Java集成到Windows中（Sun认为，这种方式使Java在Windows上以一种方式工作，而在其他机器上以另一种方式工作，从而破坏了Java的跨平台兼容性）；随后，微软公司从Windows中完全去除了Sun的Java，而创建了自己的类Java语言：C#。经过两公司之间的一轮诉讼之后，Sun占据了上风，现在可以在Windows（或Linux）上安装Sun的最新Java版本（<http://www.java.com/getjava/>）。Mac OS X操作系统在安装时会附带Java。

3

除了单独的应用程序之外，Java主要用于在客户端（client side，即用户的浏览器中）创建applet，applet是一种通过因特网下载并在Web浏览器中运行的小程序。因为Java具有跨平台性质，这些applet应该能够在任何支持Java的浏览器中以相同的方式运行。在近几年，我们看到许多Java applet被Macromedia Flash动画替代了，因为一般来说Macromedia Flash动画比Java applet更容易创建。

使用<object> HTML标签将Java applet嵌入网页，还要提供指定applet的附加信息。当浏览器看到<object>标签时，它会从服务器下载Java applet，然后applet就会在这个标签中指定的屏幕区域中运行（图1-1）。

1. 中文版《HTML XHTML CSS基础教程（第6版）》已由人民邮电出版社出版。——编者注

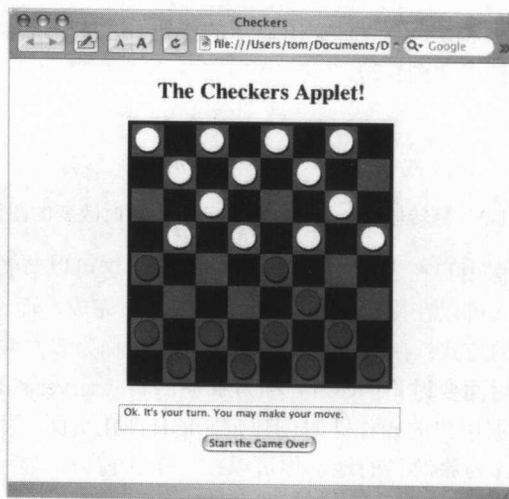


图1-1 这个Java applet显示一种西洋跳棋游戏

要想了解关于Java的更多信息，我们建议阅读Java之父Gosling等撰写的*The Java™ programming Language, Fourth Edition*¹。

4

1.3 JavaScript 的起源

既然JavaScript与Java无关²，那么为什么它们的名称如此相似呢？这是计算机行业最烦人的恶习之一：为了获得市场营销方面的成功，而不顾及产品实质。

当Netscape在它的Navigator Web浏览器中添加了一些基本脚本功能时，它最初将这种脚本语言称为LiveScript。与此同时，Java开始大行其道，它被认为是计算行业中下一项伟大的革新。当Netscape在Navigator 2中支持运行Java applet时，它也将LiveScript改名为JavaScript，希望以此借用Java的声势。尽管JavaScript和Java是非常不同的编程语言，但这一事实并没有阻止Netscape采用这种市场营销手段，从那时候开始，我们这些技术作家只好不厌其烦地解释JavaScript和Java是不同的东西。当然，我们靠这挣了不少钱，从这个角度来说，我们可能应该感谢这些市场营销专家。

当微软公司看到JavaScript在Web开发人员中流行起来时，它意识到必须在IE中添加一些脚本功能。它原本可以采用JavaScript，但是与通常情况一样，微软公司又自行其事，建立了自己的脚本语言，这种语言非常像JavaScript，但又不完全相同。JavaScript的微软版本称为JScript。

5

1.4 JavaScript 可以做什么

用JavaScript可以做许多事情，使网页更具交互性，给站点的用户提供更好、更令人兴奋的体验。JavaScript使你可以创建活跃的用户界面，当用户在页面间导航时向他们提供反馈。例如，你

1. 中文版《Java程序设计语言（第4版）》和英文影印版均已由人民邮电出版社出版。——编者注

2. 严格地说，JavaScript是在Java的影响下设计出来的，参见JavaScript之父Bredan Eich的回忆文章。——编者注

可能在一些站点上见过在鼠标指针停留时突出显示的按钮。这是用JavaScript实现的，使用了一种称为翻转器（rollover）的技术（图1-2）。

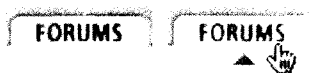


图1-2 翻转器是一个会在鼠标指针停留时改变的图像

可以使用JavaScript确保用户在表单中输入有效的信息，这可以节省你的业务时间和开支。如果表单需要进行计算，那么可以在用户机器上的JavaScript中完成，而不需要任何服务器端处理。你应该知道一种区分程序的方式：在用户机器上运行的程序称为客户端（client-side）程序；在服务器上运行的程序（包括后面要讨论的CGI）称为服务器端（server-side）程序。

可以使用JavaScript根据用户的操作即时创建定制的HTML页面。假设你正在运行一个旅行指南站点，用户点击夏威夷作为旅游目的地。你可以在一个新窗口中显示最新的夏威夷旅游指南。JavaScript可以控制浏览器，所以你可以打开新窗口、显示警告框以及在浏览器窗口的状态栏中显示定制的消息。JavaScript有一套日期和时间特性，可以生成时钟、日历和时间戳文档。

6 JavaScript还可以处理表单、设置cookie、即时构建HTML页面以及创建基于Web的应用程序。

1.5 JavaScript 不能做什么

JavaScript是一种客户端（client-side）语言；也就是说，设计它的目的是在用户的机器上执行任务，而不是在服务器上。因此，JavaScript有一些固有限制，这些限制主要出于安全原因：

- ❑ JavaScript不允许读写客户机器上的文件。这是有好处的，因为你肯定不希望网页能够读取自己硬盘上的文件，或者能够将病毒写入硬盘，或者能够操作你计算机上的文件。唯一的例外是，JavaScript可以写到浏览器的cookie文件，但是也有一些限制（关于cookie的更多信息，请参见第10章）。
- ❑ JavaScript不允许写服务器机器上的文件。尽管写服务器上的文件在许多方面是很方便的（比如存储页面点击数或用户填写的表单数据），但是JavaScript不允许这么做。相反，需要用服务器上的一个程序处理和存储这些数据。这个程序可以用Perl或PHP等语言编写的CGI或Java程序。
- ❑ JavaScript不能关闭不是由它自己打开的窗口。这是为了避免一个站点关闭其他任何站点的窗口，从而独占浏览器。
- ❑ JavaScript不能从来自另一个服务器的已经打开的网页中读取信息。换句话说，网页不能读取已经打开的其他窗口中的信息，因此无法探察访问这个站点的冲浪者还在访问哪些其他站点。

7

1.6 Ajax 是什么

简单的回答是，Ajax是一种创建交互式Web应用程序的方式。这究竟是什么意思呢？我们来

考虑一下可能通过Web站点做的事情。例如，你可能想和自己的朋友出去吃饭。所以，你希望知道如何从当前的地点到达市内大家都在谈论的那家有名的新餐馆。你决定到一个提供地图的Web站点上查找路线。你会进入这个站点，输入餐馆的地址，站点会显示一张标出了这个餐馆的地图。这个站点会显示带边框的地图，如果希望改变地图的视域，那么可以点击边框（图1-3）。点击边框，等5~10秒，地图就会重新绘制，如果希望再次改变视域，就重复操作。这个过程很慢，而且响应性很差。如果能够点击地图，向希望的方向拖动它，地图视域就会随着鼠标拖动而移动，那么不是更好吗？

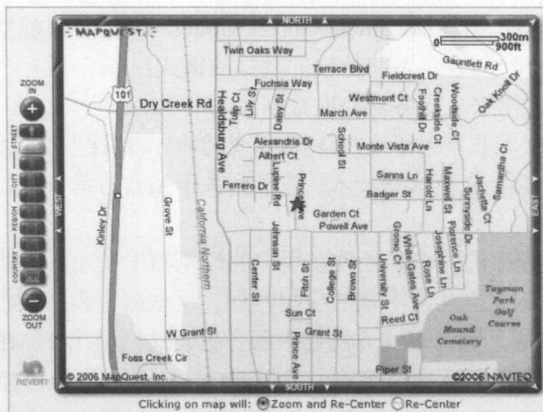


图1-3 这个MapQuest地图要花很长时间来响应用户点击，因为它需要从服务器刷新许多信息才能完成视域的改变

可以通过用Ajax构建的Web应用程序向用户提供这种动态的交互性（图1-4）。这样的话，用户几乎不需要等待；用户一直有控制能力；而且可以创建用户体验与传统桌面应用程序相同的基于Web的应用程序。这样，用户就能够更快更轻松地查明如何从家里到达那家有名的餐馆。

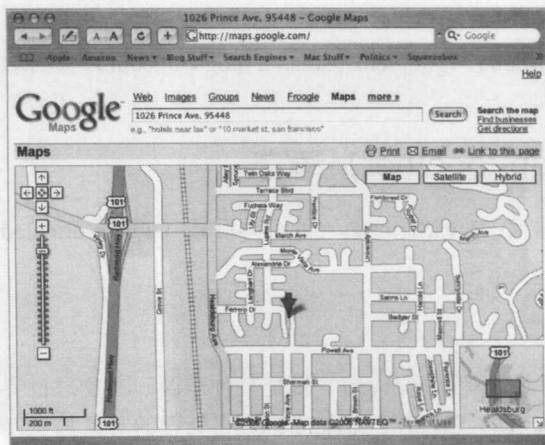


图1-4 支持Ajax的Google Maps可以提供更流畅更具交互性的用户体验

Ajax是Asynchronous JavaScript and XML（异步JavaScript和XML）的缩写，这个词是由Web开发人员Jesse James Garrett在2005年年初首创的。严格地说，Ajax只是JavaScript的一小部分（尽管这一部分特别流行）。但是，随着频繁地使用，这个词不再指某种技术本身（比如Java或JavaScript）。

在大多数情况下，Ajax一般是指以下这些技术的组合：

- XHTML；
- CSS（Cascading Style Sheet，层叠样式表）；
- 使用JavaScript访问DOM（Document Object Model，文档对象模型）；
- XML，这是在服务器和客户之间传输的数据的格式；
- XMLHttpRequest，用来从服务器获取数据。

这个列表有点儿复杂，尤其是对于在JavaScript或其他Web编程方面经验不太丰富的人。但是，你无需担心，在本书中我们将介绍这些技术。在学到关于Ajax的章节时，你应该已经掌握了组成Ajax的各种技术。

Ajax的好处是，应用程序的大多数处理在用户的浏览器中发生，而且对服务器的数据请求往往很短。所以可以使用Ajax建立功能丰富的应用程序，这些应用程序依赖于基于Web的数据，但是其性能远远超过老式方法，因为老式方法要求服务器发送回整个HTML页面来响应用户操作。

一些公司已经在Ajax方面投入大量资金，尤其是Google，Google已经建立了几个著名的Ajax应用程序，包括Gmail（基于Web的电子邮件）、Google Calendar、Google Suggest和Google Maps。另一个大型的Ajax支持者是Yahoo!，它使用Ajax增强个性化的My Yahoo!门户、Yahoo!首页、Yahoo! Mail，等等。这两家公司都向公众开放了它们的Web应用程序的接口，人们可以使用这些接口建立有意思的新应用程序。例如，许多人为Google Maps创建了mashup（混合程序），这些程序会获得地图并在地图上加上有意思、有用或好玩儿的信息，比如洛杉矶地区所有日本餐馆的位置或电影摄影棚的位置。

✓提示

- 在Google Maps Mania（googlemapsmania.blogspot.com）上可以找到许多Google Maps mashup的列表。

1.7 组合式（snap-together）语言

还有一点我们应该注意：JavaScript是一种面向对象（object-oriented）语言。可是这意味着什么呢？

1.7.1 对象

首先，我们来考虑对象。对象（object）就是某种东西。在物理世界中，一只猫、一台计算机和一辆自行车都是对象（图1-5）。对于JavaScript，它处理的对象都在Web浏览器中，比如窗口、表单和表单元素，例如按钮和复选框（图1-6）。

因为你可以有多只猫或者多个窗口，所以给它们起名字是有意义的。你可以把自己的宠物叫做一号猫和二号猫，但是由于两个原因，这不是一种好想法：首先，如果它们有独特的名称，就更容易区分它们；其次，这有点儿粗鲁。因此，本书中的所有示例将给对象起独特的名字。

✓提示

- 因特网上看到的一些脚本用window[0]和form[1]这样的名称来称呼对象。由于以上原因，这种风格并不好。如果给脚本中的不同对象起名字，而不是使用数字，那么跟踪对象会容易得多。

1.7.2 属性

对象具有属性 (property)。猫有毛皮，计算机有键盘，自行车有车轮。在JavaScript环境中，窗口有标题，表单可以有复选框。

改变对象的属性就修改了这个对象。相同的属性名可以用于完全不同的对象。假设有一个名为empty的属性。在合适的任何地方都可以使用empty，所以可以说猫的肚子空了，也可以说猫食盆空了。

注意，计算机的键盘和自行车的轮子不仅仅是属性；它们本身也是对象，可以具有它们自己的属性。所以，对象可以有子对象。

1.7.3 方法

对象可以做的事情称为方法 (method)。猫会叫，计算机会崩溃，自行车可以前进。JavaScript对象也有方法：按钮的click()，窗口的open()，文本的selected()。圆括号表示它们是方法，而不是属性。

✓提示

- 可以把对象和属性看作名词，把方法看作动词。前者是东西，后者是这些东西可以完成的活动或对它们执行的操作。

1.7.4 将这些成分组合在一起

可以将对象、属性和方法组合在一起，从而更好地描述对象或过程。在JavaScript中，这些成分由点号分隔（就像因特网地址中那样）。这称为点号语法 (dot syntax)。下面是按这种方式编写的对象及其属性的一些示例：

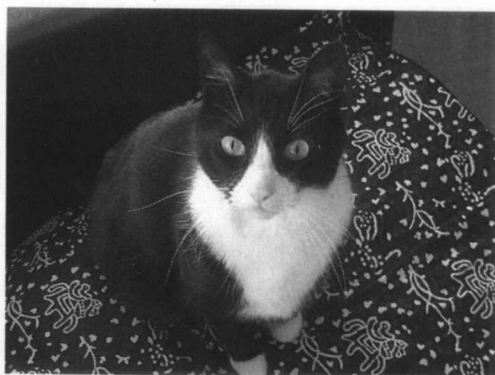


图1-5 猫对象（这只猫名叫Pixel）

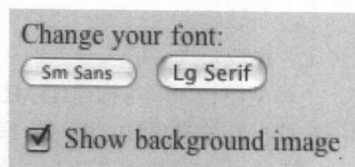


图1-6 按钮和复选框是浏览器对象，可以用JavaScript操纵它们

11

12