

原版销售
累计超过
150000册



JavaScript 基础教程 (第7版)

[美] Tom Negrino 著
Dori Smith 等译
陈剑瓯

■ 经典JavaScript入门书，涵盖Ajax

■ 透彻讲解Web开发相关技术

■ 让你体验轻松实用的学习方式



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7版)

[美] Tom Negrino 著
Dori Smith 著
陈剑珉 等译

scripts · loops · images · rollovers · methods · forms · events · cookies ·
objects · DOM · properties · libraries · regular expressions · framework
scripts · loops · images · rollovers · methods · forms · events · cookies ·
CSS · objects · DOM · properties · libraries · regular expressions · fram
works · scripts · loops · images · rollovers · methods · forms · events ·
cookies · CSS · objects · DOM · properties · libraries · regular expressio
frameworks · scripts · loops · images · rollovers · methods · forms · eve
cookies · CSS · objects · DOM · properties · libraries · regular expressi
frameworks · scripts · loops · images · rollovers · methods · forms · co
CSS · objects · DOM · properties · libraries · regular expressions · fram
images · rollovers · methods · forms · events · cookies · CSS · objects ·
properties · libraries · regular expressions · frameworks · scripts · loo
images · rollovers · methods · forms · events · cookies · CSS · objects ·
properties · libraries · regular expressions · frameworks · scripts · loo
overs · methods · forms · events · cookies · CSS · objects · DOM · prop
libraries · regular expressions · frameworks · scripts · loops · images
methods · forms · events · cookies · CSS · objects · DOM · properties

人民邮电出版社

人民邮电出版社

北京

样书

专用章

图书在版编目 (CIP) 数据

JavaScript 基础教程: 第 7 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, 2009.5

(图灵程序设计丛书)

书名原文: JavaScript and Ajax for the Web, Seventh Edition

ISBN 978-7-115-20490-5

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 (2009) 第029702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。

本书适合有志于从事 Web 开发和 Web 设计的初学者阅读, 也是高校相关课程理想的教材。

图灵程序设计丛书

JavaScript基础教程 (第7版)

- ◆ 著 [美] Tom Negrino Dori Smith
译 陈剑瓯 等
责任编辑 傅志红
- ◆ 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
- ◆ 开本: 800×1000 1/16
印张: 23
字数: 716千字 2009年5月第1版
印数: 1—3 500册 2009年5月北京第1次印刷

著作权合同登记号 图字: 01-2009-1547号

ISBN 978-7-115-20490-5/TP

定价: 49.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Authorized translation from the English language edition entitled *JavaScript and Ajax for the Web, Seventh Edition* by Tom Negrino and Dori Smith, published by Pearson Education, Inc., publishing as Peachpit Press, Copyright © 2009 by Tom Negrino and Dori Smith.

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 © 2009.

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

欢迎你使用JavaScript！使用这种容易上手的程序设计语言，可以给网页增色，使网页更好用。本书是一本轻松的JavaScript入门教程，所以即使不是计算机高手，你也可以由此很快学会脚本编写。

读者对象

我们猜想既然你有兴趣学习JavaScript，那么肯定有创建HTML页面和Web站点的经验，而且希望更进一步，使站点更具交互性。我们并不要求你了解任何编程或脚本编程的知识，也不要求你是HTML专家（当然，如果你是HTML专家也无妨）。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速查的地方才会提供。有了这些HTML信息，你就不需要在阅读本书的同时，再去翻另一本书查找HTML属性的语法了。^①

如果你对编程有所了解，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，而且本书也不是一本深入而全面的语言参考书（当然附录A中提供了一些很有价值的内容）。这方面市面上已经有不少很好的书，我们在附录D中列出了它们。本书和这些书之间的差异是，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，不赘述大量的额外信息。

在本书的上一版中，我们增加了对Ajax的介绍。这种技术结合使用了JavaScript和其他常用的Web技术为网页增加交互性，并且改善了Web站点的用户体验。我们提供了Ajax基础知识和实际示例，帮助你给站点增加Ajax功能，而不必深入学习Web编程。在这个版本中，我们添加了更多Ajax示例和技术介绍，因为我们发现大家对Ajax的关注还在持续升温。

如何使用本书

在本书中，我们采用一些特殊的版式，帮助你更轻松的学习和理解。

分步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。这么做是因为，这个版本中的所有脚本都

^① 要学习HTML，推荐阅读《HTML XHTML CSS基础教程》（第6版）。——编者注

符合W3C（World Wide Web Consortium，万维网联盟）的XHTML 1.0过渡型标准。当在JavaScript中看到引号时，总是直引号（'或"），而不是斜引号（‘或“）。全角引号会使JavaScript失效，在编写脚本时应该避免使用。

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。出现这种情况时，我们将代码行分为多行，在接续行前面使用箭头→表示这是续行，并且将续行缩进，如下所示：

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

关于浏览器

在上一版中有一个大的变化：我们不再支持非常陈旧的浏览器或者那些在支持Web标准方面做得很差的浏览器。我们发现，几乎所有Web用户都升级到了现代浏览器，它们充分地支持公认的Web标准（比如XHTML、CSS2和DOM）。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。

我们在几种操作系统上的多个浏览器中测试了我们的脚本，这些操作系统包括Windows（XP和Vista）、Mac OS X和Ubuntu Linux。

我们使用最主流的浏览器——微软Windows版本的IE——测试了本书中的所有内容，而且使用IE 8的beta版本做了大量测试（本书是在IE 8最终版本发布之前付印的）。我们还用Mac平台上的Firefox 2和3、Safari 3对脚本做了测试。适用于Safari 3 for Mac，意味着这些脚本对于从WebKit引擎衍生出来的任何浏览器（比如Safari for Windows和Omni公司的OmniWeb）都可以正常使用，对基于KHTML（Safari最开始使用的开源呈现引擎）的浏览器（比如Konqueror for Linux）也可以正常使用。

不必输入代码

一些JavaScript图书只在书中印刷出脚本代码，你在实践时必须自己输入代码。我们认为这种方式已经过时了。作者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，也会在这里给予更正。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，需要图像文件。但请放心，这些文件都放在本书的Web站点上了，而且打包好了供你下载。你找到的可下载文件包含所有脚本、HTML文件、CSS文件以及用到的所有媒体文件。如果你遇到任何问题，可以查看配套Web站点上的FAQ（常见问题）。

如果阅读了FAQ，而你的问题还没有解决，可以通过js7@javascriptworld.com给我们发邮件。很抱歉地说一声，因为收到的邮件太多，所以我们不可能也不会回复那些把本书问题发送到我们个人邮件地址的邮件。但是，我们可以保证发送到js7@javascriptworld.com的邮件会得到答复。

开始吧

JavaScript最好的一点是它很容易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就可以开始改进自己的网页。等到你看完的时候，你都能用Ajax给站点添加高级的交互效果了。

当然，千里之行，始于足下，欢迎光临，请勿将手伸出窗外，照相时请不要用闪光灯。探索JavaScript和Ajax的旅程已经开始。

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总能使我们的书更上一层楼。

感谢Tracey Croom出色的制作工作。

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

我们要感谢Peachpit出版社的Nancy Ruenzel所给与的支持。

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、大专院校的教师。

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's Web Design社区(<http://www.wise-women.org>)的耐心、支持和灵感，他们的作为堪称楷模。

目 录

第 1 章 了解 JavaScript	1	2.9 使用 JavaScript 改进链接	24
1.1 JavaScript 是什么	1	2.10 使用多级条件	28
1.2 JavaScript 不是 Java	2	2.11 处理错误	30
1.3 JavaScript 的起源	2	第 3 章 第一个 Web 应用程序	33
1.4 JavaScript 可以做什么	3	3.1 用循环进行重复操作	33
1.5 JavaScript 不能做什么	3	3.2 将值传递给函数	37
1.6 Ajax 是什么	4	3.3 探测对象	38
1.7 组合式 (snap-together) 语言	5	3.4 处理数组	40
1.7.1 对象	5	3.5 处理有返回值的函数	42
1.7.2 属性	6	3.6 更新数组	43
1.7.3 方法	6	3.7 使用 do/while 循环	44
1.7.4 将这些成分组合在一起	6	3.8 以多种方式调用脚本	45
1.7.5 DOM 简介	6	3.9 组合使用 JavaScript 和 CSS	47
1.8 处理事件	7	3.10 检查状态	50
1.9 值和变量	8	3.11 处理字符串数组	55
1.10 赋值和比较	9	第 4 章 处理图像	59
1.10.1 赋值	9	4.1 创建翻转器	59
1.10.2 比较	9	4.2 创建更有效的翻转器	61
1.11 编写对 JavaScript 友好的 HTML	10	4.3 构建三状态翻转器	65
1.11.1 结构、表现和行为	10	4.4 由链接触发翻转器	67
1.11.2 div 和 span	10	4.5 让多个链接触发一个翻转器	69
1.11.3 class 和 id	11	4.6 处理多个翻转器	71
1.12 要使用什么工具	11	4.7 创建循环的广告条	74
第 2 章 开始	13	4.8 在循环广告条中添加链接	76
2.1 将脚本放在哪里	13	4.9 建立循环式幻灯片	78
2.2 关于函数	14	4.10 显示随机图像	80
2.3 使用外部脚本	15	4.11 随机开始循环显示图像	81
2.4 在脚本中添加注释	16	第 5 章 框架	83
2.5 向用户发出警告	17	5.1 防止页面显示在框架中	84
2.6 确认用户的选择	19	5.2 迫使页面显示在框架中	85
2.7 提示用户	20	5.3 迫使站点显示在框架中	86
2.8 用链接对用户进行重定向	22		

5.4 设置目标	89	9.1.4 onmove 事件	170
5.5 创建和加载动态框架	90	9.1.5 onabort 事件	170
5.6 在框架之间共享函数	92	9.1.6 onerror 事件	170
5.7 同时加载多个框架	94	9.1.7 onfocus 事件	170
5.8 处理 iframe	95	9.1.8 onblur 事件	171
5.9 用 JavaScript 加载 iframe	97	9.2 鼠标事件处理	172
第 6 章 操作浏览器窗口	99	9.2.1 onmousedown 事件	172
6.1 打开新窗口	99	9.2.2 onmouseup 事件	175
6.2 将不同的内容加载进窗口中	102	9.2.3 onmousemove 事件	175
6.3 打开多个窗口	103	9.2.4 onmouseover 事件	177
6.4 从一个窗口更新另一个窗口	104	9.2.5 onmouseout 事件	178
6.5 关闭窗口	107	9.2.6 ondblclick 事件	178
6.6 把窗口放在指定的位置	109	9.2.7 onclick 事件	179
第 7 章 表单处理	112	9.3 表单事件处理	179
7.1 选择并转移导航菜单	113	9.3.1 onsubmit 事件	179
7.2 动态地改变菜单	116	9.3.2 onreset 事件	179
7.3 建立必须填写的字段	118	9.3.3 onchange 事件	179
7.4 根据其他字段对字段进行检查	123	9.3.4 onselect 事件	179
7.5 标识有问题的字段	125	9.3.5 onclick 事件	180
7.6 准备进行表单验证	127	9.3.6 onblur 事件	180
7.7 处理单选按钮	130	9.3.7 onfocus 事件	182
7.8 用一个字段设置另一个字段	134	9.4 键事件处理	183
7.9 检验 Zip 编码	136	9.4.1 onkeydown 事件	183
7.10 验证电子邮件地址	140	9.4.2 onkeyup 事件	185
第 8 章 表单和正则表达式	146	9.4.3 onkeypress 事件	185
8.1 用正则表达式验证电子邮件地址	146	第 10 章 JavaScript 和 cookie	186
8.2 验证文件名	151	10.1 建立第一个 cookie	186
8.3 提取字符串	153	10.2 读取 cookie	189
8.4 对字符串进行格式化	155	10.3 显示 cookie	190
8.5 对字符串进行格式化和排序	159	10.4 使用 cookie 作为计数器	191
8.6 对字符串进行格式化和验证	160	10.5 删除 cookie	193
8.7 使用正则表达式替换元素	162	10.6 处理多个 cookie	195
第 9 章 处理事件	165	10.7 显示新内容提醒信息	197
9.1 窗口事件处理	165	第 11 章 对象和 DOM	201
9.1.1 onload 事件	165	11.1 关于节点操纵	201
9.1.2 onunload 事件	168	11.1.1 DOM-2 和 W3C	201
9.1.3 onresize 事件	168	11.1.2 DOM-2 术语	201
		11.1.3 DOM-3	202

11.2 添加节点	202	15.2 添加下拉菜单	294
11.3 删除节点	204	15.3 改进下拉菜单	296
11.4 删除特定的节点	206	15.4 带说明的幻灯片	300
11.5 插入节点	209	15.5 一个无聊的姓名生成器	302
11.6 替换节点	212	15.6 柱状图生成器	306
11.7 用对象字面值编写代码	214	15.7 样式表切换器	312
第 12 章 建立动态页面	219	第 16 章 用 Ajax 设计页面	320
12.1 在网页上显示当前日期	219	16.1 突出显示新元素	320
12.2 处理周中日	221	16.2 创建可折叠菜单	323
12.3 根据时间对消息进行定制	222	16.3 创建更漂亮的对话框	326
12.4 根据时区显示日期	223	16.4 条纹表格	328
12.5 把 24 小时制转换为 12 小时制	226	16.5 表格排序	332
12.6 创建倒数计数器	229	第 17 章 bookmarklet	337
12.7 隐藏和显示层	232	17.1 第一个 bookmarklet	337
12.8 移动文档中的对象	234	17.2 改变页面的背景颜色	342
12.9 日期方法	236	17.3 改变页面样式	342
第 13 章 Ajax 简介	239	17.4 查询单词	344
13.1 Ajax 的定义	239	17.5 查看图像	346
13.2 读取服务器数据	241	17.6 显示 ISO Latin 字符	347
13.3 解析服务器数据	247	17.7 将 RGB 值转换为十六进制	348
13.4 刷新服务器数据	251	17.8 对值进行转换	349
13.5 从服务器获得数据	253	17.9 bookmarklet 计算器	350
13.6 用 Ajax 预览链接	256	17.10 缩短 URL	352
13.7 自动补全表单字段	260	17.11 检验页面	353
第 14 章 Ajax 工具包	266	17.12 通过电子邮件发送页面	353
14.1 拖放页面元素	267	17.13 改变页面大小	354
14.2 在页面上添加日历	271		
14.3 在页面上添加双月日历	275		
14.4 使用容器实用程序	281		
14.5 添加动画效果	285		
14.6 实现用于调试的 Logger 控件	287		
第 15 章 JavaScript 应用示例	291		
15.1 使用可折叠菜单	291		

第1章

了解JavaScript

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。

自从诞生以来，JavaScript已经经历了不小的演化（尽管有时候在不同的浏览器上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，你将了解JavaScript是什么（以及不是什么）、它可以做什么（以及不能做什么）和JavaScript语言的一些基础知识。还会向你介绍Ajax，这是JavaScript和其他技术的一种激动人心的组合，它在Web站点的交互性和创造性方面掀起了新的浪潮。

1.1 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，复制一下并稍做修改，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。

为了帮助你熟悉JavaScript，我们建立了一个与本书配套的Web站点。在这个站点上提供了本书中的所有脚本（这样，你就不用自己输入了），以及更多的说明、附加资料和更新内容。站点的网址是<http://www.javascriptworld.com>。

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（编写脚本的原先方式），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，如脚本1-1所示。如果你还不熟悉这些HTML概念，需要关于HTML的更多信息，那么我们建议你读一下Elizabeth Castro的《HTML XHTML CSS基础教程（第6版）》。

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"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Barely a script at all</title>
</head>
<body bgcolor="#FFFFFF">
<h1>
    <script type="text/javascript">
        document.write("Hello, Cleveland!");
    </script>
</h1>
</body>
</html>
```

1.2 JavaScript 不是 Java

尽管名字中有Java，但是JavaScript和Java之间没有太大的关系。Java是一种全功能的编程语言，由Sun公司开发和推广。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类电子设备。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公司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公司认为，这种方式会使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公司的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公司占据了上风，现在可以在Windows（或Linux）上安装Sun公司的最新Java版本（<http://www.java.com/getjava/>）。Mac OS X操作系统在安装时会附带Java。

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，所以这些applet应该能够在任何支持Java的浏览器中以相同的方式运行。在近几年中，我们看到许多Java applet被Adobe Flash动画替代了，因为一般来说Adobe Flash动画比Java applet更容易创建。

可以使用<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（见图1-1）。

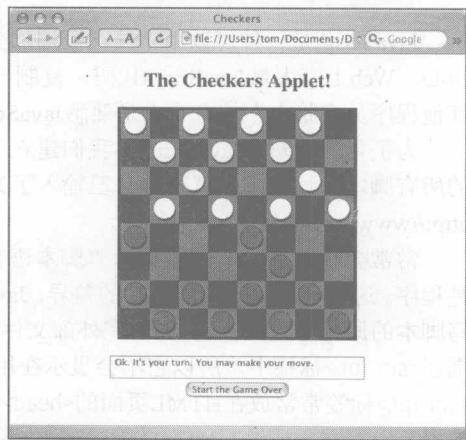


图1-1 这个Java applet实现了一种西洋跳棋游戏

1.3 JavaScript 的起源

既然JavaScript与Java无关，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在其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。

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，你可能在一些站点上见过，当鼠标指针停留在按钮上时会突出显示按钮。这是用JavaScript实现的，使用了一种称为翻转器（rollover）的技术（见图1-2）。



图1-2 翻转器是一个当鼠标指针停留在其上时会改变的图像

可以使用JavaScript来确保用户以表单形式输入有效的信息，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用JavaScript来完成，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。

使用JavaScript，根据用户的操作可以创建定制的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。

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是一种客户端语言。^①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，这些限制主要出于如下安全原因。

- ❑ 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，或者能够操作你的计算机上的文件。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）。

① 实际上，也存在服务器端实现的JavaScript版本，如Rhino。——编者注

- ❑ 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（运行在服务器上的程序）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，因此无法探索访问这个站点的冲浪者还在访问其他哪些站点。

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（见图1-3）。点击边框，等大约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。如果只是点击地图并向希望的方向拖动它，地图视域就会随着鼠标拖动而移动，那不是更好吗？

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，并一直有控制能力，而且可以创建具有与传统桌面应用程序相同用户体验的基于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到达那家有名的餐馆。

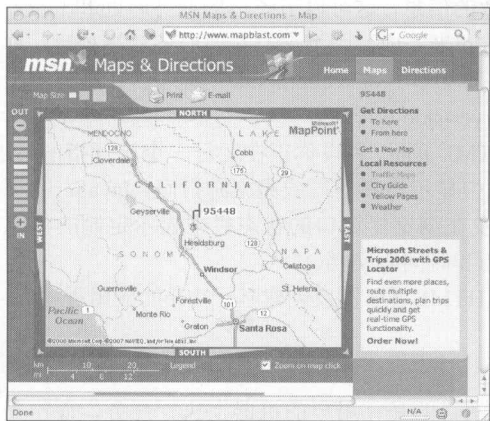


图1-3 这个MapQuest地图要花很长时间来响应用户点击，因为它需要从服务器刷新许多信息才能完成视域的改变

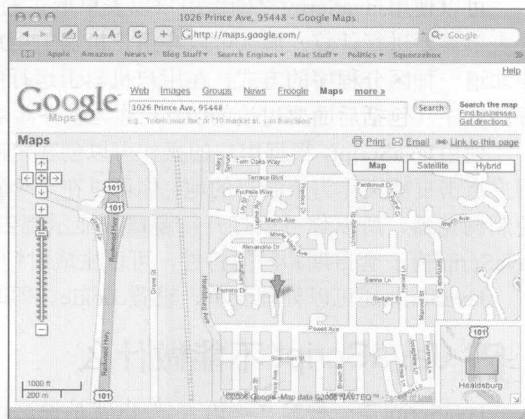


图1-4 支持Ajax的Google Maps可以提供更流畅更具交互性的用户体验

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访问的DOM (Document Object Model, 文档对象模型);
- ❑ XML, 这是在服务器和客户端之间传输的数据格式^①;
- ❑ 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, 不必担心, 我们在本书中会介绍这些技术。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传回整个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 Docs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其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 (混搭), 这些程序会获得地图并在地图上加上有意思、有用或好玩的信息, 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) 的语言。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。在现实中, 一只猫、一台计算机和一辆自行车都是对象 (见图1-5)。对于JavaScript, 它处理的对象都在Web浏览器中, 比如窗口、表单, 以及如按钮和复选框的表单元素 (见图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, 但是这不是一种好想法, 原因有两个: 首先, 如果它们有独特的名称就更容易区分它们; 其次, 这有点不礼貌。因此, 本书中的所有示例将给对象起独特的名字。

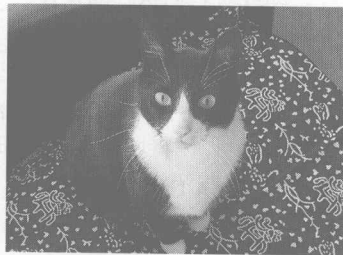


图1-5 猫对象 (这只猫名叫Pixel)

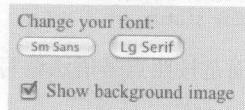


图1-6 按钮和复选框是浏览器对象, 可以用JavaScript来操纵

① 目前数据传输也常用JSON格式。——编者注

✓提示

- ❑ 在因特网上可能会看到一些脚本用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。
- ❑ 有些爱挑剔的程序员会认为，JavaScript不是面向对象的，而是基于对象（object-based）的。在本书中，这两个意思相同。

11

1.7.2 属性

对象具有属性（property）。猫有毛皮，计算机有键盘，自行车有轮子。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。在任何合适的地方都可以使用empty，所以可以说猫的肚子空了，也可以说猫的食盆空了。

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（就像因特网地址中的那样）。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：

12

```
bicycle.wheels
cat.paws.front.left
computer.drive.dvd
document.images.name
window.status
```

下面是按照点号语法编写的对象及其方法的一些示例：

```
cat.purr()
document.write()
forms.elements.radio.click()
```

1.7.5 DOM 简介

在网页上，组成页面（或文档）的对象被组织在一个树型结构中。在构建HTML页面之前，你一定

要了解这种结构。页面的顶级包含在<html>标签中，在其中会找到<head>和<body>标签，而其他标签包含在这两个标签中，依此类推。某些浏览器可以显示这种树型结构，如图1-7所示。JavaScript将文档树中的每一项都当作对象，可以使用JavaScript操纵这些对象。用来表示文档中对象的标准模型就称为DOM（Document Object Model）。

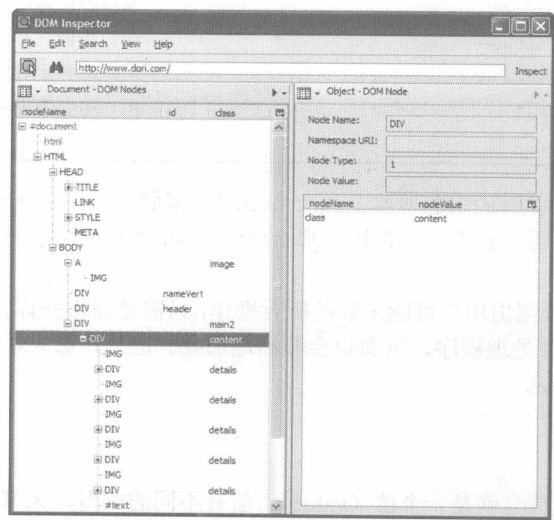


图1-7 可以使用DOM Inspector（Firefox的一部分）查看文档的树型结构。在Safari和IE中也有相似的特性

树中的每个对象也称为树的节点（node）。可以使用JavaScript修改树的任何方面，包括添加、访问、修改和删除树上的节点。树上的每个对象是一个节点。如果节点包含HTML标签，那么它就称为元素节点（element node）；否则，就称为文本节点（text node），当然，元素节点可以包含文本节点。这就是目前关于DOM和节点需要知道的所有知识，在本书中（尤其是第11章）会看到关于它们的更多信息。

13

1.8 处理事件

事件（event）是用户在访问页面时执行的操作。提交表单和在图像上移动鼠标就是两种事件。JavaScript使用称为事件处理程序（event handler）的命令来处理事件。用户在页面上的操作会触发脚本中的事件处理程序。表1-1列出了最常用的12种JavaScript事件处理程序。在第9章中，我们会讨论其他更高级的事件处理程序。

表1-1 事件处理程序

事 件	它处理什么
onabort	用户终止了页面的加载
onblur	用户离开了对象
onchange	用户修改了对象
onclick	用户点击了对象
onerror	脚本遇到了一个错误