



Microsoft®
Press

Microsoft
核心技术丛书

JavaScript

编程循序渐进

(美) Steve Suehring 著
李强 译

JavaScript Step by Step



机械工业出版社
China Machine Press

TP312/2916

2008

Microsoft
核心技术丛书

JavaScript

编程循序渐进

(美) Steve Suehring 著
李强 译

JavaScript Step by Step



机械工业出版社
China Machine Press

本书是一本不可多得的 JavaScript 教程。全书共 19 章,包括 JavaScript 基础知识、语法、变量和数据、操作符和表达式、条件和循环控制流、函数、对象、Web 表单、Cookie、操作图像、浏览器、CSS、错误处理、XML 以及 Ajax 等从基础知道到实际开发的全部知识。本书涵盖范围广泛,讲述由浅入深、循序渐进。书中还包含大量习题。

本书适合 JavaScript 初学者阅读,也可供有其他语言编程经验,想要学习和掌握 JavaScript 的读者学习。

Steve Suehring: JavaScript Step by Step (ISBN: 978-0-7356-2449-8) .

Copyright 2008 by Microsoft Corporation.

Original English language edition copyright © 2008 by Steve Suehring.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U. S. A. All rights reserved.

本书中文简体字版由美国微软出版社授权机械工业出版社出版。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。

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2008-2926

图书在版编目 (CIP) 数据

JavaScript 编程循序渐进/(美)苏灵(Suehring, S.)著;李强译. —北京:机械工业出版社, 2008. 7

(Microsoft 核心技术丛书)

书名原文: JavaScript Step by Step

ISBN 978-7-111-24282-6

I. J… II. ①苏… ②李… III. JAVA 语言 - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 079466 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 李南丰

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2008 年 7 月第 1 版第 1 次印刷

186mm × 240mm · 20.25 印张

标准书号: ISBN 978-7-111-24282-6

定价: 45.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

本社购书热线: (010) 68326294

译者序

21 世纪初, AJAX (Asynchronous JavaScript and XML, 异步 JavaScript 和 XML) 技术的提出和应用, 真正掀起了 Web 2.0 的革命大潮。

AJAX 提供与服务器异步通信的能力, 从而使用户从请求、响应的循环中解脱出来。有了 AJAX, 在用户单击按钮时, 我们可以使用 JavaScript 和 DHTML 立即更新 UI, 并向服务器发出异步请求, 以执行更新或查询数据库。当请求返回时, 就可以使用 JavaScript 和 CSS 来更新相应的 UI, 而不是刷新整个页面。最重要的是, 用户甚至不知道浏览器正在与服务器通信: Web 站点看起来是即时响应的。

然而, AJAX 离不开客户端脚本编程。AJAX 中的“J”就代表着 JavaScript 语言。JavaScript 是 Web 应用程序开发的必备语言。无论是为 Web 页面添加交互性还是创建整个应用程序, 如果没有 JavaScript, 今天的 Web 不会是现在这个样子。

毫无疑问, 要在当今的 Web 应用程序开发领域占据一席之地, 不了解和熟悉 JavaScript 是不行的。这也就是市面上各种 JavaScript 技术图书不断涌现且受到读者欢迎的原因。

那么, 本书有什么不同之处呢?

这是一本带领读者循序渐进地掌握 JavaScript Web 编程技能的图书, 概括起来, 它有以下几个显著特色:

- 面向没有程序设计基础的初学者, 讲述从最基础的 JavaScript 知识到实用的开发技能;
- 引领读者分步骤地、按部就班地动手实际练习和操作, 以掌握知识点和技能;
- 用每章末尾的练习题来检查对知识点的学习和掌握。

翻译本书的过程中, 我们深深地感觉到这是一本不可多得的 JavaScript 学习教程。通过阅读本书并掌握 JavaScript, 读者的 Web 开发技能必将得到很大的提升。

参加翻译本书工作的有李强、关志兴、王建勇、毛立涛、闫柳青、姜巧生、沈海峰、谢扣林、乔义峰、刘查强、王义强。读者朋友在阅读本书的过程中, 如果觉得有疑问或不妥之处, 可通过 reejohn@sohu.com 联络译者。

李强

2008 年 5 月

致 谢

现在，我最乐意写的部分就是致谢了。在我以往的经历中，已经编写过几本书，而且因此写过一些致谢，我不再想要感谢在编写本书过程中所联络的每个人。

在编写本书的时候，我的妻子 Rebecca 和我迎来了我们的第一个孩子，Jakob Edward。如果你喜欢徒劳而无益，可以尝试在有一个新生儿的同时编写一本技术图书的难缠章节。幸运的是，我的妻子曾经（而且现在也是）绝对令人难以置信，并且理解时间底线就是时间底线。她给我充分的时间，而没有任何怨言，因而，我能够按时完成这本书。当我写到大概还剩下 3 章的时候，Jakob 降生了，我相信他的到来导致了本书最好的 3 章内容。

我要感谢 Microsoft 出版社的每个人。为 Microsoft 出版社写作，对我来说是一次不错的经历，并且，我希望将来能够重复这一经历。Ben Ryan 和 Devon Musgrave 启动了本书的想法，然后，Maureen Zimmerman、Michelle Goodman、Lori Merrick 和 Kathleen Atkins 都对管理这一项目给予了帮助，因此，感谢他们为一个过于疲劳的作者所做的一切。还感谢 Kurt Meyer，他做了很好的技术编辑工作。Kurt 小心翼翼的工作和坚持在最终的产品上——得到体现。

感谢我的经纪人 Neil Salkind 以及 Studio B 的每个人的辛勤工作。也感谢 Tuescher family、John Hein、James Leu、Jeremy Guthrie、Jason Keup、Justin Hoerter、Chris Trawicki、Aaron Saray、Brian Pinter、Jenni Guggenheimer、Kelly Gardner、John O'Keefe 以及 Triangle 的其他人（我没法感谢每个人，因为如果那样的话，本书的篇幅不够）。Tony Falduto 和服务器的团队，Eliot Irons、Andy Hale 以及数据安全团队的其他人，Jeff Sanner、Aaron Deering 和架构师团队，Greg Pfluger，还有来自 Sentry 的我已经没有篇幅表示感谢的其他人。看，我说过不打算这么做但还是这么做了，再次感谢每个人，以及此次由于我没有考虑到并肯定遗漏了的人们。来自 Partners 的 Tim 和 Rob、Dave Marie、Johnny K、Pat Dunn、Sarah、Brian Page、Jay 和 Deb Schrank、Denise、Mark Little、Andy Berkvam、AJ Prowant、meek、Jim Oliva、John Eckendorf、Nightmare Squad、Kent Laabs、Kyle MacDonald、Dan Noah、Ron Mackay，你们在哪里？Chad Chasteen。现在我坐在这里盯着屏幕想着需要记起的每个其他的人们。不可避免，我会遗忘某些人。我喜欢本书的这一部分，因为我可以继续漫笔而不必太多的文本编辑，感谢 Susan。向那些没有包含在这一致谢中的人们道歉。我正在讨论另外一本书，在我下次写致谢的时候希望能够包含你们。

已经在屏幕前坐太久了，我要结束这一致谢了，感谢几年来我所编写的书的所有读者。我很乐意收到你们的 E-mail，即便我并没有时间一一回复。当我接收到来自很多不同国家的读者的 E-mail 的时候，我总是很吃惊。我编写的图书在巴西、德国（以及欧洲的其他国家）、加拿大都有很多的读者。我为此受到激励去学习多种不同的语言，并且希望能够尽快地访问巴西和德国（我已经去过加拿大很多次，我很快又会去，并且我已经会说那里的语言了）。

前	言
本书主要介绍 JavaScript 语言及其在 Web 开发中的应用	本书
本书主要介绍 JavaScript 语言及其在 Web 开发中的应用	本书
本书主要介绍 JavaScript 语言及其在 Web 开发中的应用	本书
本书主要介绍 JavaScript 语言及其在 Web 开发中的应用	本书
本书主要介绍 JavaScript 语言及其在 Web 开发中的应用	本书
本书主要介绍 JavaScript 语言及其在 Web 开发中的应用	本书
本书主要介绍 JavaScript 语言及其在 Web 开发中的应用	本书

JavaScript 是 Web 应用程序开发的必备语言。无论是为 Web 页面添加交互性还是创建整个应用程序，如果没有 JavaScript，今天的 Web 不会是这个样子。

JavaScript 是具有正式规范的一种基于标准的语言。然而，正如任何 Web 开发者所告诉你的一样，几乎每个 Web 浏览器对这个规范的解释都不同。这使得 Web 开发者的工作更困难。幸运的是，大多数 Web 浏览器都在逐渐统一它们对 JavaScript 的核心功能的支持和解释。

本书提供了对 JavaScript 的介绍，包括它的一些核心功能以及新特性和范型，例如 AJAX。今天的 Web 用户依靠很多不同的平台以及很多不同的浏览器来浏览 Web 内容。这个事实也是本书涉及开发的每个方面的中心，因此，你将会看到多种浏览器的屏幕截图，以及强调基于标准的而不是特有的 JavaScript 开发。

本书的第一部分介绍了 JavaScript 并帮助你开始开发 JavaScript。对于 JavaScript 开发，不需要特殊的工具，因此，你将会看到如何在 Microsoft Visual Studio、Eclipse 和记事本（或者任何文本编辑器）中创建 JavaScript 文件。接下来介绍了 JavaScript 的核心语言和功能，然后是 JavaScript 和 Web 浏览器的关系。最后，介绍了 AJAX，你将看到如何构建动态的搜索表单。

本书体例和特色

本书按部就班地带领你学习 JavaScript 编程语言。从本书的开始部分学习，并且按照每个例子和练习进行学习，这将能够帮助你获取有关 JavaScript 编程语言的知识。

如果你已经有些熟悉 JavaScript，你可能试图跳过本书的第 1 章。然而，第 1 章介绍了有关 JavaScript 历史的一些细节，以及本书的一些基本前提，这两部分都有助于理解本书其他部分的讨论。第 2 章介绍了如何开始使用 JavaScript 编程。如果你已经熟悉了 Web 开发，你可能已经有一个 Web 开发程序，因此你可能想试图跳过第 2 章。尽管如此，你应该熟悉第 2 章中用来创建 JavaScript 程序的模式。

本书包含了一个目录，它有助于你更快地找到特定的部分。每章包含了它所介绍的内容的详细列表。

此外，随书光盘包含了本书 PDF 文件以及本书中很多示例的源代码。

体 例	含 义
列表	按部就班的练习通过从 1 开始的分步过程列表来表示
参见	这些段帮你指出一个特定主题相关信息的其他资源
提示/注意	提示和注意给出了对于给定主题可能有帮助的一些额外信息
内嵌代码	内嵌代码即出现在一段中的代码，用斜体表示
代码块	代码块用不同字体显示，以突出代码

使用随书光盘^①

随书光盘中包含了本书很多示例的源代码。

示例源代码的用法

下载得到的源代码根据每章来放置，每章一个目录。每章的目录下包含了该章中使用的按部就班练习。有些章还包含了该章中的其他源代码。

使用示例文件

由于 JavaScript 通常依赖于包含它的 Web 页面，按部就班的练习的源代码已经分别放到各个目录中。这使得你可以复制和粘贴大量的重复性的 HTML，并且关注于把 JavaScript 输入到示例中。

每章的目录还包含了一个 CompletedCode 目录，其中包含了完整的示例。你可以打开 CompletedCode 文件来查看该章中所放置的示例。

获取帮助

为了确保本书的准确性，已经做出了各种努力。如果你遇到问题，请联系如下列出的各种资源来获取帮助。

如果你的问题涉及本书的内容，请首先在 Microsoft Press Knowledge Base 进行在线搜索，那里提供了本书中已知错误及更正的支持信息，位于如下的 Web 地址：

www.microsoft.com/mspress/support/search.asp

如果你没有在线 Knowledge Base 中找到答案，可按照如下邮件地址发送你的评论和问题到 Microsoft Learning Technical Support: mspinput@microsoft.com

作者简介

Steve Suehring 是一位具有广泛的商业和计算机背景的技术顾问。Steve 的广泛经验使得他能够在各种功能的组织工作，并帮助创建适合业务需要的计算架构。Steve 热衷于写作技术

① 所指光盘为英文版原书光盘，中文版并没有附带该光盘。本书所有代码均可在华章公司网站（www.hzbook.com）下载。——编辑注

图书和杂志文章。Steve 在国际各大用户组织和大会上演讲。有机会的时候，Steve 会参加一些体育运动或者演奏一些乐器，他会比别人做得更好。

译者简介

李强 国内知名上市软件公司产品经理，多年来致力于医疗信息化建设，主持开发过多个大型项目。在 JAVA、.Net、JavaScript 和 Web 编程方面颇有心得。曾在计算机报刊、杂志发表过多篇文章，并有多本计算机译著出版。2007，翻译了号称“犀牛书”的《JavaScript 权威指南（第5版）》，该书获得了 CSDN 评选的 2008“金牛奖”图书特别推荐奖，目前在持续热销中。

译者
致谢
前言

第一部分 JavaScript 基础知识

第1章 JavaScript 超乎你的想象	2
1.1 JavaScript 历史	2
1.1.1 进入 Internet Explorer 3	3
1.1.2 随后就有了 ECMAScript	3
1.1.3 这又是什么标准	4
1.1.4 DOM	4
1.2 JavaScript 程序的组成	4
1.3 把 JavaScript 放到 Web 页面上	5
1.4 JavaScript 可以做什么	7
1.5 JavaScript 不能做什么	7
1.6 使用 JavaScript 的提示	9
1.6.1 JavaScript 置干何处	10
1.6.2 站点应该支持哪些浏览器	11
1.7 练习題	11
第2章 使用 JavaScript 开发	12
2.1 JavaScript 开发选择	12
2.2 配置环境	12
2.3 用 Visual Studio 2005 编写 JavaScript	13
2.3.1 使用 Visual Studio 2005 开发第一个 Web (和 JavaScript) 项目	16
2.3.2 通过 Visual Studio 2005 使用外部 JavaScript 文件	19
2.4 使用 Eclipse 编写 JavaScript	22

2.4.1 使用 Eclipse 开发 Web 项目	23
2.4.2 在 Eclipse 内使用外部 JavaScript 文件	28
2.5 不使用 IDE 编写 JavaScript	31
2.5.1 使用记事本开发第一个 Web (和 JavaScript) 项目	31
2.5.2 不用 IDE 而使用外部 JavaScript 文件	34
2.6 编写 JavaScript	35
2.7 练习題	36
第3章 JavaScript 语法和语句	37
3.1 一些常規	37
3.1.1 区分大小写	37
3.1.2 空白	37
3.1.3 注释	38
3.1.4 分号	39
3.1.5 换行	40
3.1.6 正确放置 JavaScript	40
3.2 JavaScript 语句	41
3.2.1 语句内是什么	41
3.2.2 JavaScript 语句的两种类型	41
3.3 JavaScript 的保留字	42
3.4 函数简介	42
3.5 练习題	44
第4章 使用变量和数据	45
4.1 JavaScript 中的数据类型	45
4.1.1 数字	45
4.1.2 字符串	48
4.1.3 布尔值	51
4.1.4 空	51

目 录

译者序
致谢
前言

第一部分 JavaScript 基础知识

第 1 章 JavaScript 超乎你的想象	2
1.1 JavaScript 简史	2
1.1.1 进入 Internet Explorer 3	3
1.1.2 随后就有了 ECMAScript	3
1.1.3 这么多标准	4
1.1.4 DOM	4
1.2 JavaScript 程序的组成	4
1.3 把 JavaScript 放到 Web 页面上	5
1.4 JavaScript 可以做什么	7
1.5 JavaScript 不能做什么	7
1.6 使用 JavaScript 的提示	9
1.6.1 JavaScript 置于何处	10
1.6.2 站点应该支持哪些浏览器	11
1.7 练习题	11
第 2 章 使用 JavaScript 开发	12
2.1 JavaScript 开发选择	12
2.2 配置环境	12
2.3 用 Visual Studio 2005 编写 JavaScript	13
2.3.1 使用 Visual Studio 2005 开发第一 个 Web (和 JavaScript) 项目	16
2.3.2 通过 Visual Studio 2005 使用 外部 JavaScript 文件	19
2.4 使用 Eclipse 编写 JavaScript	22

2.4.1 使用 Eclipse 开发你的第一个 Web (和 JavaScript) 项目	23
2.4.2 在 Eclipse 内使用外部 JavaScript 文件	28
2.5 不使用 IDE 编写 JavaScript	31
2.5.1 使用记事本开发第一个 Web (和 JavaScript) 项目	31
2.5.2 不用 IDE 而使用外部 JavaScript 文件	34
2.6 调试 JavaScript	35
2.7 练习题	36
第 3 章 JavaScript 语法和语句	37
3.1 一些常识	37
3.1.1 区分大小写	37
3.1.2 空白	37
3.1.3 注释	38
3.1.4 分号	39
3.1.5 换行	40
3.1.6 正确放置 JavaScript	40
3.2 JavaScript 语句	41
3.2.1 语句内是什么	41
3.2.2 JavaScript 语句的两种类型	41
3.3 JavaScript 的保留字	42
3.4 函数简介	42
3.5 练习题	44
第 4 章 使用变量和数据	45
4.1 JavaScript 中的数据类型	45
4.1.1 数字	45
4.1.2 字符串	48
4.1.3 布尔值	51
4.1.4 空	51

4.1.5 未定义	51	6.1.3 复合条件	80
4.1.6 对象	51	6.2 else if 和 else 语句	82
4.1.7 数组	52	6.3 多层条件语句	83
4.2 定义并使用变量	53	6.4 三元条件语句	86
4.2.1 声明变量	53	6.5 switch	87
4.2.2 变量类型	54	6.6 使用 while 循环	88
4.2.3 变量作用域	54	6.6.1 while 语句	89
4.2.4 引用和垃圾回收机制	59	6.6.2 do...while 语句	89
4.3 类型转换	60	6.7 使用 for 循环	90
4.3.1 数字转换	60	6.7.1 for 循环	90
4.3.2 字符串转换	61	6.7.2 for...in 循环	92
4.3.3 布尔值转换	61	6.7.3 for each...in 循环	93
4.4 练习题	61	6.8 使用条件语句验证表单	94
第 5 章 使用操作符和表达式	62	6.9 练习题	96
5.1 操作符介绍	62	第 7 章 使用函数	97
5.2 加法操作符	62	7.1 什么是函数	97
5.3 乘法操作符	63	7.1.1 函数变量	97
5.4 位操作符	63	7.1.2 再谈变量作用域	99
5.5 相等操作符	64	7.1.3 return 值	100
5.6 关系操作符	65	7.1.4 函数调用的更多内容	101
5.6.1 in 操作符	66	7.1.5 匿名函数 (函数直接量)	102
5.6.2 instanceof 操作符	66	7.2 方法	102
5.7 一元操作符	67	7.3 对话框函数概览	103
5.7.1 自增和自减	67	7.4 练习题	106
5.7.2 用加号转换为一个数字	68	第 8 章 JavaScript 中的对象	107
5.7.3 使用减号创建一个负数	68	8.1 面向对象的开发	107
5.7.4 按位求反和逻辑非	68	8.1.1 对象	107
5.7.5 delete 操作符	68	8.1.2 属性	107
5.7.6 typeof 操作符	71	8.1.3 方法	108
5.7.7 void 操作符	72	8.1.4 类	108
5.8 赋值操作符	73	8.2 创建对象	110
5.9 练习题	74	8.2.1 为方法添加属性	111
		8.2.2 为对象添加方法	113
		8.3 关于数组的更多内容	114
第二部分 应用 JavaScript		8.3.1 length 属性	114
第 6 章 条件和循环控制流	76	8.3.2 数组方法	114
6.1 if 如何使用	76	8.4 内建对象	118
6.1.1 if 语句的语法	76	8.5 练习题	126
6.1.2 Windows Internet Explorer 7 中的 prompt () 函数	77	第 9 章 浏览器对象模型	127
		9.1 浏览器简介	127

9.1.1 浏览器层级	127
9.1.2 事件	128
9.2 认识 self	128
9.3 获取有关屏幕的信息	130
9.4 使用 navigator 对象	131
9.5 location 对象	134
9.6 history 对象	140
9.7 练习题	140
第 10 章 文档对象模型	141
10.1 文档对象模型定义	141
10.1.1 DOM 层级 0: 遗留的 DOM	141
10.1.2 DOM 层级 1 和 DOM 层级 2	142
10.1.3 作为树的 DOM	142
10.1.4 操作节点	143
10.2 获取元素	143
10.2.1 通过 ID 访问	143
10.2.2 通过标签名访问	146
10.3 操作属性	148
10.3.1 察看属性	148
10.3.2 设置属性	150
10.4 创建元素	151
10.4.1 添加文本	151
10.4.2 添加元素并设置 ID	152
10.5 删除元素	153
10.6 练习	154

第三部分 把 JavaScript 整合到设计

第 11 章 和 Web 表单一起使用

JavaScript	156
11.1 JavaScript 和 Web 表单	156
11.2 获取表单数据	158
11.3 使用表单信息	158
11.3.1 使用选择框	159
11.3.2 使用复选框	161
11.3.3 使用单选按钮	164
11.4 预先验证表单数据	165
11.4.1 绕过 JavaScript 验证	166

11.4.2 验证一个文本字段	169
11.5 练习题	170
第 12 章 创建和消费 Cookie	171
12.1 初探 cookie	171
12.2 使用 JavaScript 创建 cookie	172
12.2.1 一个简单的 cookie	172
12.2.2 设置 cookie 的过期日期	173
12.2.3 设置 cookie 路径	176
12.2.4 设置 cookie 的域	177
12.2.5 操作安全 cookie	179
12.3 使用 JavaScript 读取 cookie	179
12.4 删除 cookie	180
12.5 练习题	181
第 13 章 用 JavaScript 操作图像	182
13.1 使用图像翻滚	182
13.1.1 一个简单的翻滚	182
13.1.2 更好的办法	183
13.2 预载入图像	188
13.3 操作滚动放映	190
13.3.1 创建一个显示	190
13.3.2 向后移动	191
13.4 使用图像映射	194
13.5 练习题	197
第 14 章 浏览器和 JavaScript	198
14.1 理解窗口事件	198
14.1.1 事件模型	198
14.1.2 一个通用事件处理程序	204
14.2 检测关于访问者的事情	204
14.2.1 userAgent 属性简介	205
14.2.2 功能测试	205
14.2.3 保证 JavaScript 避开较早的浏览器	206
14.2.4 其他的 navigator 属性和方法	208
14.3 打开、关闭和调整窗口	208
14.3.1 打开和关闭窗口实践	209
14.3.2 窗口打开最佳实践	214
14.3.3 不需要 JavaScript	215
14.3.4 调整和移动窗口	215
14.4 定时器	216

14.5 练习题	217	17.1.1 XML 文档的一个例子	244
第 15 章 JavaScript 和 CSS	218	17.1.2 使用 JavaScript 载入一个 XML 文档	245
15.1 什么是 CSS	218	17.2 一点展望	252
15.1.1 属性和选择器	219	17.3 练习题	252
15.1.2 应用 CSS	220	第 18 章 初探 AJAX	253
15.2 JavaScript 和 CSS 之间的 关系	220	18.1 AJAX 简介	253
15.2.1 根据 ID 来设置元素样式	220	18.2 XMLHttpRequest 对象	254
15.2.2 根据类型设置元素样式	223	18.2.1 实例化 XMLHttpRequest 对象	254
15.2.3 使用 JavaScript 获取 元素样式	225	18.2.2 发送一个 AJAX 请求	255
15.2.4 使用 JavaScript 修改 样式表	226	18.2.3 处理一个 AJAX 响应	256
15.3 练习题	228	18.2.4 处理 XML 请求	260
第 16 章 JavaScript 错误处理	229	18.2.5 使用 JSON	260
16.1 错误处理概览	229	18.2.6 处理头部	261
16.2 使用 try/catch	229	18.2.7 使用 POST 方法	262
16.3 使用 onerror 事件	236	18.3 案例学习: 动态搜索和更新	264
16.3.1 把 onerror 绑定到 window 对象	236	18.4 练习题	269
16.3.2 忽略错误	238	第 19 章 深入 AJAX	270
16.3.3 把 onerror 绑定到 image 对象	239	19.1 使用 XML 和 CSS 创建一个 HTML 表格	270
16.4 练习题	241	19.2 使用 CSS 样式化表格	273
第四部分 AJAX 及其他		19.3 创建一个动态下拉框	276
第 17 章 JavaScript 和 XML	244	19.4 从用户和 AJAX 接受输入	281
17.1 和 JavaScript 一起使用 XML	244	19.5 练习题	281
		附录 练习题解答	282

JavaScript 基础知识

- | | |
|-----|-------------------|
| 第1章 | JavaScript 超乎你的想象 |
| 第2章 | 使用 JavaScript 开发 |
| 第3章 | JavaScript 语法和语句 |
| 第4章 | 使用变量和数据 |
| 第5章 | 使用操作符和表达式 |

JavaScript 超乎你的想象

欢迎阅读这个介绍性的章节。学完本章，你将能够：

- 了解 JavaScript 的历史。
- 识别 JavaScript 程序的各个部分。
- 使用 JavaScript 伪协议。
- 理解 JavaScript 适用于 Web 页面中的何处。
- 理解 JavaScript 可以做什么以及不能做什么。

1.1 JavaScript 简史

JavaScript 不是 Java。扫清了这一认识上的障碍，我们可以继续去扫清更大的障碍、学习更重要的内容，例如，如何做出漂亮的下拉菜单。严格地说，JavaScript 是一个规范的俗称，其更为正式的名字是 ECMAScript。但 ECMAScript 这个名字不好读。不信你就试试，大声念“ECMAScript”。没错吧（稍后会介绍 ECMAScript 的更多内容）。

JavaScript 从何而来呢？你可能知道 JavaScript 丰富而传奇的（并且是有争议的）历史，或者，你可能不会真的太关心这一点。如果是这两种情况之一，你可能想要跳到下一章，直接开始编写 JavaScript 代码。当然，这将会是一个错误，你将错过下面有关 JavaScript 的所有精彩内容。并且，了解一些 JavaScript 的历史，对于理解这一语言如何在当今的各种环境中实现，是很重要的。

JavaScript 最初由 Brendan Eich 于 1995 ~ 1996 年间在 Netscape 开发出来。那时候，这种语言叫做 LiveScript。如果这是一种新语言的不错的名字，那么故事就已经到此为止了。然而，在一次遗憾的决定中，由于营销人员有他们自己的想法，因此这一语言改名为 JavaScript。混淆接踵而来。你知道，Java 语言是那个时候最热的技术，并且一些人决定要尝试利用 Java 的流行性。因此，JavaScript 很快和 Java 语言扯上了关系。遗憾的是，尽管 Java 由于频繁使用而流行，但是，由于一些 Web 站点用 Java 来处理显示或者为站点添加无用的扩展（例如，恼人的滚动文本），Java 也因此某种程度上沾染了恶名。由于 Java 需要一个插件载入到 Web 浏览器中，从而减慢了浏览过程并且普遍令访问者不满，Java 的用户体验遭受损害；更别提易访问性的问题了。从此，JavaScript 和 Java 及其存在的问题被人们无情地联系到了一起。直到最近，JavaScript 才开始动摇这种联系。

JavaScript 不是一种编译型语言，这使得它看上去像是一种缺乏能力的语言。20 世纪 90 年代末期，有能力的互联网程序员开始放弃 JavaScript，但他们很快意识到它的用处及其在

WWW 上的模拟和创造交互性的能力。直到那个时候,很多 Web 站点都是由简单的 HTML (Hypertext Markup Language, 超文本标记语言) 组成的,其图形缺乏视觉吸引力以及和站点内容交互的能力。

早期的 JavaScript 关注于客户端表单验证,以及和 Web 页面上的图像协作来提供与访问者之间基本的但有用的交互和反馈。当 Web 站点的一位访问者要填写一个表单,JavaScript 将用于即时验证 Web 表单的内容,而不是用于和服务端往返交互。通过节省往返时间来使应用程序变得更快一点和响应更积极一些,这是一种好办法,尤其是在宽带网尚未普及的那些日子。

1.1.1 进入 Internet Explorer 3

1996 年,随着 Microsoft Internet Explorer 3 的发布,Microsoft 包含了对核心 JavaScript 的支持,在 Internet Explorer 中叫做 JScript,还有对另一种脚本语言 VBScript 的支持。遗憾的是,尽管这些语言相似,它们各自的实现并不完全相同。因此,人们开发了一些方法来检测 Web 站点访问者使用的浏览器以及用来响应 JavaScript 的浏览器是什么,而这些方法将会针对特定的浏览器有效。这个过程叫做浏览器检测 (browser detection),我们将在本书第 14 章中介绍。尽管现在大多数应用程序开发者不赞成浏览器检测技术,但它仍然在使用。

1.1.2 随后就有了 ECMAScript

在 1997 年中期,Microsoft 和 Netscape 与欧洲计算机制造商协会 (European Computer Manufacturers Association, ECMA) 协作,发布了 ECMAScript 的第 1 版。从那时候开始,Netscape 和 Microsoft 的所有浏览器都实现了 ECMAScript 标准的版本。其他流行的浏览器,例如 Firefox 和 Opera,也实现了 ECMAScript 标准。

ECMAScript 最新版本的标准是 ECMA-262,发布于 1999 年。好消息是,像 Internet Explorer 4 和 Netscape 4.5 这样的浏览器都支持这一标准,并且其后的每种主要的浏览器都以支持 ECMA-262 标准的形式支持 JavaScript。坏消息是,每种浏览器都以略有不同的方式应用这一标准。这意味着,不兼容性仍然烦扰着使用 ECMA-262 JavaScript 的开发人员。

ECMAScript 的第 4 版仍在制订中,很快将发布。参见 <http://www.ecma-international.org/> 了解关于这一标准的更多信息。

作为将要把 JavaScript 整合到自己的 Web 应用程序的开发者,你需要考虑到 JavaScript 在各个解释之间的不同。这可能意味着以略微不同的方式实现一个脚本,并且也意味着你需要在各种不同的浏览器和平台上测试、测试、再测试。在今天的互联网上,对于只在一种浏览器中工作的、设计差劲的应用程序,人们几乎不能容忍。



注意 在多个浏览器中测试你的 Web 站点是势在必行的,即便是对于那些你不想用于 Internet Explorer 以外的应用程序,也应该这么做。即便你确定自己的应用程序只用于 Internet Explorer,或者这是你唯一正式支持的浏览器,你仍然应该在其他的浏览器中测试。这不仅对于安全很重要,而且也表明你是理解当今互联网技术的一位全面的开发者。

1.1.3 这么多标准

如果你认为 JavaScript 程序设计周围的标准的定义多少有些松散,那么,你是对的。每种浏览器对于 JavaScript 的支持都略微不同,这使你的工作很难。像 Microsoft Visual Basic 或 Perl 的某一个版本,它们这样的语言是单一的、特定的实体,而 JavaScript 则不是这样,想要写出它所有的这些差异也是很难的。然而,情况不是这样,并且你的工作(以及我的工作)是了解这些区别,在必要的时候考虑它们,尽可能地找到其共同点。

1.1.4 DOM

另一种和 JavaScript 程序员相关的标准是文档对象模型(Document Object Model, DOM)标准,它是由万维网联盟(World Wide Web Consortium, W3C)提出的标准。W3C 把 DOM 定义为“一个允许程序和脚本动态访问和更新文档的内容、结构和样式的,平台中立和语言中立的接口”。这个标准意味着,有一个 Web 浏览器遵从的规范,你可以使用这个规范来以动态的方式使用一个 Web 页面。DOM 为 HTML 和 XML(Extensible Markup Language,可扩展标记语言)文档创建了一个树型结构,并且允许对这些对象进行脚本编程。JavaScript 频繁地和 DOM 交互来实现众多重要的功能。

和 JavaScript 一样,不同的浏览器对 DOM 也有不同的解释,由此使得一个 JavaScript 程序员的生活更加有趣得多。Internet Explorer 4 和 Netscape 之前的版本,包含了对一个早期的 DOM 的支持,即对层次 0 的支持。如果你使用 DOM 层次 0,可以合理地保证在这些浏览器以及在其之后的所有浏览器中都能够得到支持。

Internet Explorer 5 和 Internet Explorer 5.5 包括了对 DOM 层次 1 的某些支持,而 Internet Explorer 6 及其以后的版本包含了对 DOM 层次 2 的某些支持。其他的浏览器,像 Firefox 和 Opera,则支持 W3C 标准。

在学习 JavaScript 标准和相关的 DOM 标准的时候,有一件事情你应该注意,那就是,你需要对自己所编写的(这毫不奇怪)代码以及用来实现这些代码的语法特别留意。如果你没有做到,JavaScript 可能会惨败,并且使你的页面无法在给定的浏览器中显示。第 10 章更详细地介绍了 DOM。



提示 W3C 有一个应用程序可以测试你的 Web 浏览器对各种 DOM 级别的支持。

这个应用程序可以在 <http://www.w3.org/2003/02/06-dom-support.html> 找到。

1.2 JavaScript 程序的组成

一个 JavaScript 程序由语句组成,语句则由标记、操作符和标识符组合起来,从而使其对一个 JavaScript 解释器有意义,而大多数的 Web 浏览器都包含一个 JavaScript 解释器。这似乎没错,但实际上,对于已经有其他语言编程经验的人来说,根本没有那么复杂。一条语句可能是:

```
var smallNumber = 4;
```

在这条语句中,有一个标记或保留字 var,后面跟着其他的标记,例如一个标识符、一个

操作符和一个直接量。你将会在第2章以及本书剩余的部分学习关于这些的更多内容。这条语句的目的是设置一个值等于整数4的变量。

与任何程序设计语言一样，语句按照一种顺序组织起来，这种顺序使得一个程序能够执行一个或多个函数。说到函数，JavaScript 有自己定义函数的方法，我们将在第7章了解更多内容。JavaScript 定义了几个内建函数，我们可以在自己的程序中使用它们。

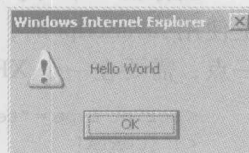
使用 javascript 伪协议和函数

1. 打开诸如 Internet Explorer 或 Firefox 一样的一个 Web 浏览器。
2. 在地址栏中，输入如下的代码并按下 Enter 键。

```
javascript:alert("Hello World");
```

3. 按下 Enter 键后，将会看到如下的一个对话框。

恭喜！我们已经编写了自己的第一段 JavaScript 代码（尽管它还不是非常有用）。然而，我们已经见到了将会在自己的 JavaScript 编程过程中使用的两个重要的内容：浏览器中的 JavaScript 伪协议标识符，以及更为重要的 alert 函数。稍后的章节将会介绍这两项内容，但是现在，我们只需要知道已经学习了将来要用到的内容就足够了。



JavaScript 也是事件驱动的，这意味着，它可以对某个事件或者“发生的事情”做出响应，例如，鼠标点击或者表单字段中的文本变化这样的事件。把 JavaScript 和一个事件联系起来，这是 JavaScript 的众多形式和常见用法的核心。在第14章，我们将会看到如何使用 JavaScript 响应事件。

1.3 把 JavaScript 放到 Web 页面上

如果你初次接触 HTML，那么，现在只需要知道 HTML 使用一对匹配的标签来表示 Web 页面中的元素，而每个标签都是用尖括号括起来的。结束标记用一个斜杠 (/) 打头。元素可以彼此嵌套。JavaScript 位于 Web 页面的 <head> </head> 和/或 <body> </body> 标签内部的 <script> 标签中，如下面的例子所示。

```
<html>
<head>
<title>A Web Page Title</title>
<script type = "text/javascript">
// JavaScript Goes Here
</script>
</head>
<body>
<script type = "text/javascript">
// JavaScript can go here too
</script>
</body>
</html>
```

JavaScript 位于 <body> 标签之中，在遇到它的时候开始执行。当你需要使用一个