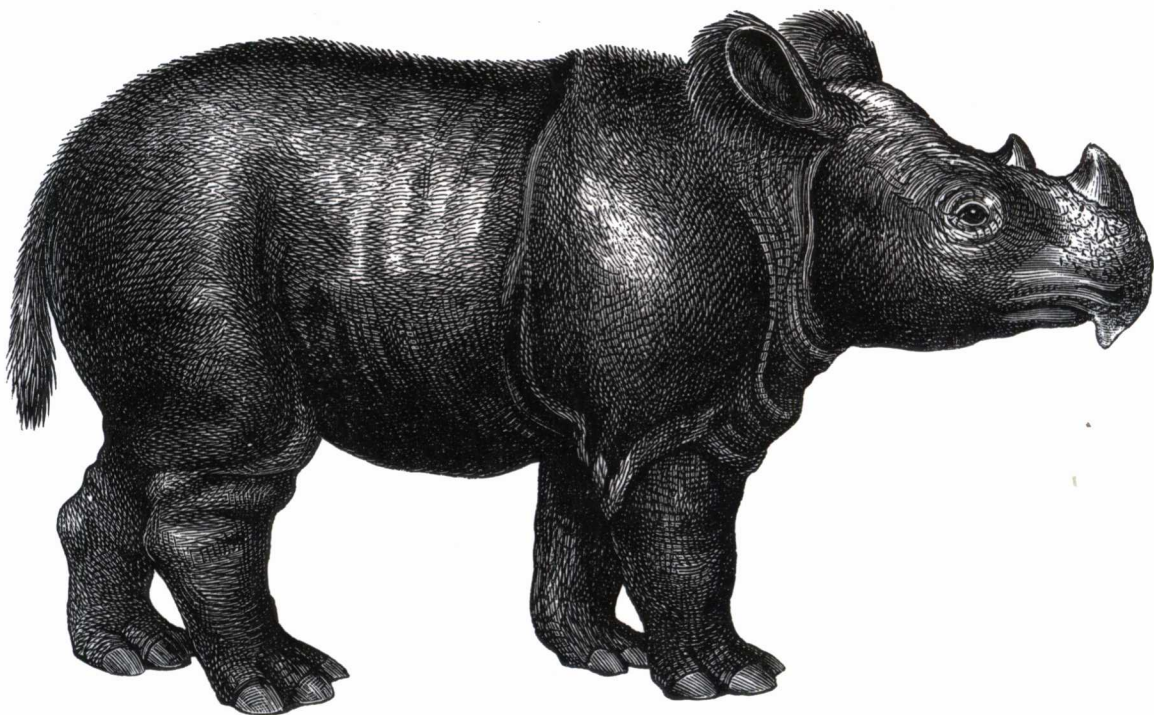


Learning JavaScript

涵盖
Ajax & DOM

JavaScript

核心技术



O'REILLY®



机械工业出版社
China Machine Press



Shelley Powers 著

苏敬凯 等译

JavaScript核心技术

Shelley Powers 著

苏敬凯 等译

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Paris · Sebastopol · Taipei · Tokyo

O'Reilly & Media, Inc. 授权机械工业出版社出版

机械工业出版社

图书在版编目 (CIP) 数据

JavaScript 核心技术 / (美) 帕瓦斯 (Powers, S.) 著. —北京: 机械工业出版社, 2007.6

书名原文: Learning JavaScript

ISBN 978-7-111-21297-3

I. J… II. 帕… III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2007) 第050391号

北京市版权局著作权合同登记

图字: 01-2007-1834号

© 2007 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2007. Authorized translation of the English edition, 2007 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2007。

简体中文版由机械工业出版社出版2007。英文原版的翻译得到O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

本书法律顾问 北京市展达律师事务所

书 名/ JavaScript核心技术

书 号/ ISBN 978-7-111-21297-3

责任编辑/ 刘立卿

封面设计/ Karen Montgomery, 张健

出版发行/ 机械工业出版社

地 址/ 北京市西城区百万庄大街22号 (邮政编码 100037)

经 销/ 新华书店北京发行所发行

印 刷/ 北京京北制版厂印刷

开 本/ 178毫米×233毫米 22印张

版 次/ 2007年6月第1版第1次印刷

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

O'Reilly Media, Inc.介绍

为了满足读者对网络 and 软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权机械工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc. 一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

译者序

Ajax是当今Web开发领域最流行的词汇。而JavaScript与CSS、XML和DOM几种老技术，加上XMLHttpRequest就构成了Ajax的四大基石。对于JavaScript，一些更资深的同事告诉我的感觉是失望。面对不同的浏览器和浏览器的不同版本，没有优秀的调试开发工具，JavaScript成了软件开发的泥潭。

而本书的出版则给我们增加了一丝解决这些问题的信心。

它从最简单的地方入手，不仅讲述了JavaScript的基础知识，还讲述了JavaScript如何操作CSS、DOM等Ajax基础技术。而关于跨浏览器兼容问题的解决，则贯穿了本书的始终。本书对于各种浏览器、浏览器版本、JavaScript版本、DOM版本的介绍，有助于我们理解所遇到的各种新旧代码，使我们能够对这些代码做出正确的取舍。

本书还提供了一些使用JavaScript的最佳实践。无论是新手还是老手，这些如何正确使用JavaScript的经验都能帮助他们养成良好的编程习惯。本书还介绍了一些调试和开发JavaScript的工具，这些工具无疑能够提高我们的开发效率。

本书最后对于Ajax和几个流行的JavaScript库的介绍，无疑会开阔我们在JavaScript使用上的思路。

本书是一本真正意义上的“新书”，不仅介绍了最新的JavaScript知识和方向，还完全覆盖了当今Web开发中关于JavaScript的所有重要话题，它使用了大量实例代码，图文并茂地讲解了使用JavaScript的各个层次和领域的内容。它不是一本参考手册，但却是一本值得拥有的教程。

感谢作者Shelley Powers。她在百忙之中回答了很多我在本书的翻译过程中遇到的问题。并且她还提供了英文版所没有的1.4.4一节，专门讲述DOCTYPE问题。本书的中文版对此也进行了及时的补充。

参加本书翻译的还有贺洁、尚计升、史红伟、祝国志、张峰峰、张文军、张艳军、王小财和周宝华。本书翻译力求忠于原著，但由于时间仓促，译者水平有限，翻译错误和不妥之处在所难免，欢迎广大读者批评指正。

译者于北京

2007.2

目录

前言	1
第1章 JavaScript初探	7
1.1 规范和实现相互交织的历史	8
1.2 跨浏览器的不兼容性和其他常见的JavaScript传说	10
1.3 你能用JavaScript来做什么	11
1.4 JavaScript初探: “Hello World!”	12
1.5 JavaScript沙箱	23
1.6 可访问性和JavaScript的最佳实践	24
第2章 JavaScript数据类型与变量	32
2.1 变量的标识	32
2.2 作用域	37
2.3 简单类型	41
2.4 常量: 有名称但不改变	50
2.5 习题	50
第3章 运算符和语句	51
3.1 JavaScript语句的格式	51
3.2 简单语句	53
3.3 条件语句和程序流	59
3.4 条件运算符	66
3.5 逻辑运算符	71
3.6 高级语句: 循环语句	72
3.7 习题	76

第4章 JavaScript对象	77
4.1 对象构造函数	77
4.2 Number对象	78
4.3 String对象	80
4.4 正则表达式与RegExp	85
4.5 有专门用途的对象: Date和Math	92
4.6 JavaScript数组	99
4.7 关联数组: 不是数组的数组	103
4.8 习题	104
第5章 函数	105
5.1 定义函数: 细数所有方式	105
5.2 回调函数	113
5.3 函数和递归	115
5.4 嵌套函数、函数闭包和内存泄漏	117
5.5 作为对象的函数	120
5.6 习题	121
第6章 捕捉事件	122
6.1 0级DOM上的事件句柄	123
6.2 2级DOM上的事件句柄	133
6.3 产生事件	139
6.4 习题	140
第7章 表单与即时验证	142
7.1 访问表单	142
7.2 把事件附加在表单上: 不同的方法	143
7.3 选择列表	144
7.4 单选按钮和复选框	148
7.5 输入字段和JIT正则表达式	154
7.6 习题	157
第8章 沙箱及之上的cookie、连通性和隐私	158
8.1 沙箱	159

8.2 关于cookie的一切	161
8.3 备选存储技术	166
8.4 跨站脚本 (XSS)	170
8.5 习题	172
第9章 基础浏览器对象	174
9.1 BOM总述	174
9.2 window对象	175
9.3 框架和地址	184
9.4 history、screen和navigator	190
9.5 document对象	195
9.6 有旧、有新	201
9.7 习题	202
第10章 DOM: 文档对象模型	203
10.1 两种接口的故事	204
10.2 DOM和兼容浏览器	205
10.3 DOM的HTML API	206
10.4 理解DOM: 核心API	213
10.5 DOM的核心文档对象	220
10.6 Element及在上下文中访问元素	223
10.7 修改树	225
10.8 习题	228
第11章 创建定制的JavaScript对象	229
11.1 JavaScript对象和原型	230
11.2 创建自己的定制JavaScript对象	231
11.3 对象检测、封装和跨浏览器对象	236
11.4 链式构造函数和JS继承	241
11.5 一次性对象	243
11.6 高级错误处理技术 (try、throw和catch)	245
11.7 JavaScript中的新特性	249
11.8 习题	251

第12章 构建动态网页：在脚本中加入样式	253
12.1 DHTML: JavaScript、CSS和DOM	254
12.2 字体和文本	259
12.3 定位和移动	262
12.4 调整大小和修剪形状	270
12.5 属性display、visibility和opacity	275
12.6 习题	281
第13章 使用Ajax	282
13.1 Ajax: 不仅是代码	283
13.2 Ajax如何工作	285
13.3 Hello Ajax World!	286
13.4 Ajax对象: XMLHttpRequest和IE的ActiveX对象	289
13.5 是否使用XML	293
13.6 Google Maps	300
13.7 习题	303
第14章 好消息：生动的程序库！令人惊异的Web服务！有趣的API！	304
14.1 在开始之前，要注意的一句话	304
14.2 使用Prototype	305
14.3 Script.aculo.us介绍	311
14.4 Sabre的Rico	314
14.5 Dojo	317
14.6 Yahoo! UI	321
14.7 MochiKit	324
14.8 习题	328
附录 习题答案	330

前言

JavaScript最初是打算作为一个脚本接口，用于在浏览器客户端（那时是Netscape Navigator）加载的网页和服务端的应用之间。自1995年引入以来，JavaScript已变成了Web开发的一个关键部分，而且在其他地方也有不少用途。

本书讲述了JavaScript语言，首先是JavaScript的最基本的数据类型，这是开始学习一门语言时首先要学习的，然后是一些最复杂的JavaScript特性，包括那些涉及Ajax和DHTML的部分。读完这本书的时候，你将掌握有关JavaScript的一些基础知识，这些基础知识即使在你今后面对最复杂的库和Web应用程序时也是很有用的。

读者

本书的读者应当熟悉网页技术，包括CSS和HTML/XHTML，还应该见过一些网页中的JavaScript。此外，对于读者并不要求以前有编程经验，只是之前没有编程经验的话可能某些章节需要额外地反复看几次才能看懂。

本书的读者对象包括：

- 任何想要或需要把JavaScript集成到自己的个人主页或其他网站中的人。
- 任何使用内容管理工具（比如博客工具）并想更好地理解那些纳入到工具模板中的脚本构件的人。
- 寻求将JavaScript与某些DHTML/Ajax特性集成到其网站中的Web开发人员。
- 想要为新客户市场开发Web服务的开发人员。
- 讲授Web技术相关内容的教师。
- 想要深入理解如何使用交互或动画效果使自己的设计变得更加生动的网页设计师。
- 任何对Web技术感兴趣的人。

阅读前提和阅读方法

正如前面说过的，本书假设读者已有(X)HTML和CSS方面的经验，且对Web应用程序的运行原理有所了解。编程经验不是必需的，但是本书涵盖了JavaScript的方方面面，其中有些内容会相对复杂一些。虽然较难的章节很少，但是还是需要读者对JavaScript有足够的了解才能使用较新的Ajax程序库。

本书分为4部分：

第1章至第3章介绍了JavaScript应用的结构，包括该语言所支持的简单数据类型，还有基本语句和控制结构。这为在后续的章节中理解JavaScript打下了基础。

第4章至第8章介绍了主要的JavaScript对象，包括关键的函数、访问网页表单的脚本、事件处理、脚本安全性和使用cookie。这些主题组合在一起就构成JavaScript的核心。读完这些章节之后，读者就可以验证表单的元素、设置和检索cookie、捕捉事件并为事件提供功能，甚至还能创建JavaScript程序库。这几章所讲述的功能近10年来一直都是JavaScript的基础，并且在未来10年中还会是这样。

第9章到第11章将深入到网页开发的一些更复杂的方面。这几章讲述了浏览器对象模型（Browser Object Model）和较新的文档对象模型（Document Object Model），还向读者展示了如何创建自己的定制对象。如果你希望创建一个新窗口，或者独立地访问、修改甚至动态地创建一个页面元素的话，理解这些模型非常重要。另外，有了定制对象，你所获得的能力就会超越语言或浏览器本身所预置的那些能力。

第12章到第14章将进入到JavaScript的高级应用，包括DHTML、Ajax以及在众多对两者都支持的一些精彩新程序库。

第1章，JavaScript初探

介绍JavaScript并给出一个小型的网页应用进行快速的初体验。本章还讲述了一些与使用JavaScript相关的问题，包括很多可用的工具，还有安全性和可访问性的问题。

第2章，JavaScript数据类型与变量

提供了JavaScript中的基本数据类型的概览，以及对于语言变量、标识符和JavaScript语句结构的概述。

第3章，运算符和语句

讲述了JavaScript的基本语句，包括赋值语句、条件语句和控制语句，还有这三种语句所必需的运算符。

第4章, JavaScript对象

介绍了一些首要的JavaScript内置对象, 包括Number、String、Boolean、Date和Math。本章还介绍了RegExp对象, 它提供了进行正则表达式模式匹配的功能。在检查表单字段时, 正则表达式很重要。

第5章, 函数

关注于另一个JavaScript内置对象: 函数。函数是创建定制对象的关键, 它还能将JavaScript代码块打包, 这样就可以在不同的JavaScript应用中一次又一次地反复使用。JavaScript函数相对简单, 但某些方面也较复杂, 其中就包括递归和闭包。这两个问题都在本章进行了介绍, 而在第11章中还有更详细的介绍。

第6章, 捕捉事件

关注于事件处理, 包括最初的事件处理形式(依然广泛使用于很多应用中), 还有较新的基于DOM的事件处理形式。

第7章, 表单与即时验证

介绍JavaScript在表单和表单字段上的使用, 包括如何访问每种类型(比如文本输入字段和下拉列表)的字段以及对检索出的数据进行验证。在表单提交给Web服务器之前进行表单验证, 能够避免与服务器之间进行不必要的往返数据传输, 这就节省了时间和资源。

第8章, 沙箱及之上的cookie、连通性和隐私

讲述基于脚本的cookie(它在客户机上存储了小块的数据)。有了cookie, 你可以存储用户名、密码和其他信息, 这样用户就不必总是重新录入数据了。另外, 由于讨论cookie不可避免地要引发对安全性的讨论, 本节还讲述了一些与JavaScript相关的安全问题。

第9章, 基础浏览器对象

开始介绍从JavaScript可访问到的对象模型。首先是浏览器对象模型(Browser Object Model), 它是对象的一个层次结构, 其中的对象包括window、document、forms、history和location等等。通过BOM, JavaScript可以打开窗口, 访问forms、links和images之类的页面元素, 甚至实现一些基本的动态效果。

第10章, DOM: 文档对象模型

关注于文档对象模型, DOM是一个简单但并非无足轻重的对象模型, 它提供了对于所有文档元素和属性的访问。你将看到基于XML(比如XHTML)和HTML的

文档。虽然模型容易理解并且作用范围也相当明了，但对于编程新手来说，本章的某些内容还是具有一些挑战性。

第11章，创建定制的JavaScript对象

展示如何在JavaScript中创建定制对象，讲述此语言中用以创建这类定制结构的全部原型结构。讨论了一些编程语言的概念，比如继承和封装，但你并不需要有过这些概念的经验。

第12章，构建动态网页：在脚本中加入样式

对一些较常用的动态HTML效果进行了概括介绍，包括拖放操作、折叠/展开一部分页面内容、可见性效果和移动效果。要求对CSS有一定程度的了解。

第13章，使用Ajax

本章介绍Ajax。无论产生了怎样令人激动的效果，Ajax实际上并不是JavaScript的复杂运用。除了讲述Ajax的部件之外，本章还提供了一个Google Maps的例子。在对于Ajax的推动上，这个应用可能比任何其他应用都要大。

第14章，好消息：生动的程序库！令人惊异的Web服务！有趣的API！

讲述一些较流行的可以免费下载和使用的程序库。其中包括Prototype、Sabre's Rico、Dojo、MochiKit、Yahoo! UI和script.aculo.us。通过这些程序库和本书，你将获得所有需要的技能来创建难以置信的、有用的Web应用。

排版约定

本书使用如下的排版约定：

斜体 (*Italic*)

表明是新术语、URL、Email地址、文件名和文件扩展名。

等宽字体 (Constant width)

表明是广义的计算机代码，包括命令、数组、元素、语句、选项、分支、变量、元素属性、关键字、函数、类型、类、名字空间、方法、模块、对象属性、参数、值、对象、事件、事件句柄、XML标记、HTML标记、宏、文件的内容和命令的输出结果。

等宽粗体 (Constant bold)

表示应由用户逐字输入的命令或其他文本。

等宽斜体 (*Constant width italic*)

表示应当由用户所提供的值或根据上下文环境所确定的值进行替换的文本。



这一图标表示一个提示、建议或一般性的注解。



这一图标表示一个提醒或警告。

本书中所提到的网站和网页能够帮助你找到一些可能有用的在线信息。通常，我都指出了这些页面的地址 (URL) 和名称 (标题)。有些地址相对复杂一些，但通过你自己喜欢的搜索引擎，根据网页的名称进行搜索可能会更容易找到这些网页，通常应把名称写在引号中。如果根据书中给出的地址找不到相应网页，搜索也会对你有所帮助；网页可能被移到别的地方去了，但名称可能仍然起作用。

代码示例的使用

本书的目的是帮助你完成自己的工作。总体来说，你可以在自己的程序和文档中使用本书的源代码。你不需要联系我们以获得许可，除非你复制本书的大量代码。例如，编写一个程序时，使用几段来自本书的代码，这并不需要我们的许可；销售或分发带有 O'Reilly 图书中的例子的 CD-ROM，需要许可；回答问题时，引证本书或引用本书中的示例代码也不需要许可；将本书中的大量例子代码纳入到你的产品文档中，需要许可。

我们感谢注明出处，但并不要求必须这样做。注明出处时，通常应包括标题、作者、出版商和 ISBN 号。例如，“Learning JavaScript by Shelley Powers. Copyright 2007 O'Reilly Media, Inc., 978-0-596-52746-4”。

如果你感觉自己代码例子的使用已超出了正当使用或上述给出的许可，请和我们联系，邮件地址为 permissions@oreilly.com。

如何联系我们

关于本书的评论和问题请发给出版商：

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

800-998-9938 (美国或加拿大)

707-829-0515 (国际或本地)

707-829-0104 (传真)

我们为本书设立了一个网页，上面列出了勘误表、例子和一些额外的信息。你可以用如下网址访问它：

<http://www.oreilly.com/catalog/learningjvsript>

<http://www.oreilly.com.cn/book.php?bn=978-7-111-21297-3>

你还可以访问本书作者的网站：

<http://learningjavascript.info>

评论或咨询与本书有关的技术性问题，请发邮件至：

into@mail.oreilly.com.cn

bookquestions@oreilly.com

在我们的网站上，有更多关于我们的图书、会议、资源中心和O'Reilly网络的信息，欢迎访问：

<http://www.oreilly.com.cn>

<http://www.oreilly.com>

致谢

对于有些书，在你的身后有一个非常好的团队，本书就是这样。我想感谢我的编辑Simon St.Laurent，感谢他的耐心、热情和在本书写作过程中出现偏差时所给予的指导。另外，我要感谢技术及内容的审校人Steven Champeon、Roy Owens和Alan Herrell，感谢他们出色的建议和发现漏洞和不当方面给予的帮助。

我还要感谢本书的制作编辑Rachel Monaghan、文字编辑Mary Anne Weeks Mayo、索引制作人Johnna VanHoose Dinse和制作经理Marlowe Shaeffer。

最后，我要感谢那些在网上碰到的人——有的是在技术社区中，有的不是。在我写这本书时，我记得你们。在某种程度上，可以说这本书是为你们而写的，你们知道我说的是谁。

JavaScript初探

JavaScript是一种广泛使用的编程语言；它也是被误解最多的一种语言。近几年,它得到了飞速发展,大多数网站都以某种形式使用它。它基于构件的能力简化了那些日益复杂的程序库的创建工作。这些程序库在Web页面中所提供的效果,在过去大都需要安装一个外部应用程序才能实现。它也可以和服务端应用程序紧密集成在一起,而这些服务端应用程序往往使用各种各样语言开发并与多种数据库接口。正因为这些原因,JavaScript往往被视作一种轻量级的、不复杂的、不像一种“真正的编程语言”的编程语言。

在某些方面,JavaScript太容易使用了。对于它的批评者来说,它规范性欠缺;它的面向对象的能力不是真正的面向对象(OO);它存在于一个简化环境之中,只是功能的一个子集;它不安全;它是宽松类型(loosely typed)的;它不能编译成字节或位。我想起在几年前阅读过的一个JavaScript的介绍,告诫我们不要被JavaScript的名字所迷惑:JavaScript和Java没有什么关系。毕竟,Java很难学。

那么,实际情况又怎样呢?JavaScript到底是一种有趣的、轻量级的、有用但却不被认真对待的小脚本语言呢?还是一种可信任的、可用来实现网站的最重要功能的、强大的编程语言呢?JavaScript的真实情况是:它是将两种语言合二为一的语言;也正是这一真实情况令人混淆。

第一种语言是内置在Web浏览器和其他应用程序中的、用户友好的、易于使用的脚本语言,它能提供表单验证之类的功能,还有下拉菜单、数据更新时的颜色渐变和即席编辑页面之类的、很酷的新玩意。因为它实现于一个特定的环境中,通常是一个带有一些表单的Web浏览器中,这同时也是一个受保护的环境。JavaScript不需要具备管理文件、内存或很多其他编程语言的基础功能,这使得JavaScript更加精益,更加简单。无论你是否具有JS的经历,是否受过JS训练,甚至不论你以前是否有过编程经验,都可以开始JS的编程。

而第二种语言是一种成熟的、功能完备的、限制严格的、基于对象的语言。它确

实需要你对它有更深一层的理解。如果使用得当,在只对服务器端的应用做一点点修改甚至是不修改的情况下它就能帮助Web应用扩展规模(增加其用户数)。它能够简化网站的开发工作,同时又提高网站的成熟度,使一个好的站点在其访问者面前表现得更好。

如果使用不当,JavaScript也会产生站点的安全漏洞,尤其是在与其他功能(比如Web服务或数据库表单)组合使用的时候。它还会使页面变得无法使用、不可读和可访问性降低。

在本书中,将向你介绍上面所描述过的两种语言:有趣的脚本语言和强大的面向对象的编程语言。更重要的是,本书将向你展示如何正确地使用JavaScript。

1.1 规范和实现相互交织的历史

学习一种编程语言并不需要学习它的历史,但学习JavaScript却例外,因为它的历史反映在今天的Web页面中。

JavaScript是由Netscape公司发明的,最初开发它是为了进行LiveConnect服务器端的开发。Netscape当时想要一种能实现与服务器端构件接口的脚本语言,于是就创造了这一脚本语言,并起名叫“LiveScript”。后来,在和SUN(Java语言的所有者)合作之后,Netscape的工程师将LiveScript改名为JavaScript,尽管它过去和现在都与Java没有关系。著名的JavaScript权威Steven Champeon这样写道:

让我们回到1995年初。Netscape刚刚聘用了从MicroUnity Systems Engineering出来的Brendan Eich,让他负责设计和实现一种新语言。分配给他的任务是让新加入到Navigator中的Java支持能够更方便地被非Java程序员的访问,Eich最终断定:宽松类型的脚本语言才适合这样的环境;也就是说,读者(也就是数千的Web设计师和开发人员)既不需要使用字节码编译器,也不需要具备面向对象的软件设计知识,就能够把页面元素(比如表单、框架或图片)结合在一起。

为了反映语言的动态本质,他所创造的语言命名为“LiveScript”。但是没多久(在Navigator 2.0 beta开发结束之前),它就被更名为JavaScript。一个市场驱动的、并使Web设计师们困扰了好几年的错误就这样出炉了,他们在邮件列表和Usenet上不断地把这两种语言弄混。1995年12月4号,Netscape和Sun联合宣布了这一新语言的诞生,并称其是HTML和Java二者的补充。

(摘自《JavaScript: How Did We Get Here?》,O'Reilly Network,2001年4月)。