

TURING

图灵程序设计丛书 Web 开发系列

Apress®

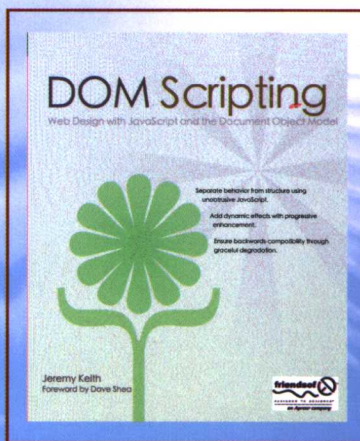
DOM Scripting

Web Design with JavaScript and the Document Object Model

JavaScript DOM 编程艺术

[英] Jeremy Keith 著
杨涛 王建桥 杨晓云 等译

- Amazon 超级畅销书
- 从 JavaScript 到 Ajax 的必由之路
- 释放 JavaScript 和 DOM 编程的惊人潜力



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

JavaScript DOM编程艺术

DOM Scripting

Web Design with JavaScript and the Document Object Model

[英] Jeremy Keith 著

杨涛 王建桥 杨晓云 等译

人民邮电出版社

北京

图书在版编目 (CIP) 数据

JavaScript DOM 脚本编程艺术 / (英) 基思著; 杨涛, 王建桥, 杨晓云译.

—北京: 人民邮电出版社, 2007.1

(图灵程序设计丛书)

ISBN 978-7-115-13921-4

I. J... II. ①基...②杨...③王...④杨... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 133751 号

内 容 提 要

本书讲述了 JavaScript 和 DOM 的基础知识, 但重点放在 DOM 编程技术背后的思路和原则: 预留退路、循序渐进和以用户为中心等, 这些概念对于任何前端 Web 开发工作都非常重要。本书将这些概念贯穿在书中的所有代码示例中, 使你看到用来创建图片库页面的脚本、用来创建动画效果的脚本和用来丰富页面元素呈现效果的脚本, 最后结合所讲述的内容创建了一个实际的网站。

本书适合 Web 设计师和开发人员阅读。

图灵程序设计丛书

JavaScript DOM 编程艺术

-
- ◆ 著 [英] Jeremy Keith
 - 译 杨 涛 王建桥 杨晓云 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
新华书店总店北京发行所经销
 - ◆ 开本: 800×1000 1/16
印张: 19.75
字数: 463 千字 2007 年 1 月第 1 版
印数: 1-5 050 册 2007 年 1 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-6445 号

ISBN 978-7-115-13921-4/TP·4908

定价: 39.00 元

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

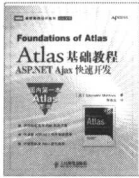
站在巨人的肩上
Standing on Shoulders of Giants



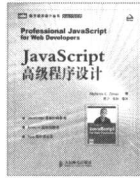
www.turingbook.com

人民邮电出版社图灵公司专注于引进国外经典教材与优秀科技图书，出版国内高校教师编写的专业教材，专业方向包括：计算机、数学、统计学和电子电气等。合作伙伴包括Prentice-Hall、Addison-Wesley、Wiley、McGraw-Hill、剑桥大学出版社、Elsevier、IEEE Press、Wrox、SIAM等多家世界知名出版公司，具有丰富的选题资源和雄厚的出版力量。

目录、前言等出版物详细信息，请浏览www.turingbook.com



书名: Atlas基础教程——ASP.NET Ajax 快速开发
原书名: Foundations of Atlas: Rapid Ajax Development with ASP.NET 2.0
作者: Laurence Moroney
译者: 陈黎夫
订书号: 7-115-15321-3
定价: 39.00元
出版时间: 2006年10月



书名: JavaScript高级程序设计
原书名: Professional JavaScript for Web Developers
作者: Nicholas C. Zakas
译者: 曹力 张欣等
订书号: 7-115-15209-8
定价: 59.00元
出版时间: 2006年9月



书名: 精通CSS: 高级Web标准解决方案
原书名: CSS Mastery: Advanced Web Standards Solutions
作者: Andy Budd
译者: 陈剑颢
订书号: 7-115-15316-7
定价: 39.00元
出版时间: 2006年10月



书名: Oracle9i&10g编程艺术: 深入数据库体系结构
原书名: Expert Oracle Database Architecture: 9i and Programming Techniques and Solutions
作者: Thomas Kyle
译者: 苏金国 王小振等
订书号: 7-115-15032-X
定价: 99.00元
出版时间: 2006年8月



书名: 设计模式解析(第2版)
原书名: Design Patterns Explained: A New Perspective on Object-Oriented Design
作者: Alan Shalloway, James R. Trott
译者: 徐言声
订书号: 7-115-15095-8
定价: 45.00元
出版时间: 2006年8月



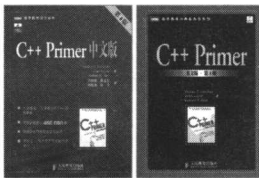
书名: .NET设计规范
原书名: Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries
作者: Krzysztof Cwalina, Brad Abrams
译者: 葛子昂
订书号: 7-115-14929-1
定价: 49.00元
出版时间: 2006年7月



书名: Ajax高级程序设计
原书名: Professional Ajax
作者: Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett
译者: 徐锋
订书号: 7-115-14867-8
定价: 49.00元
出版时间: 2006年7月



书名: Ajax实战
原书名: Ajax in Action
作者: Dave Crane, Eric Pascarella, Darren James
译者: ajaxcn.org
订书号: 7-115-14717-5
定价: 69.00元
出版时间: 2006年4月



大连理工大学
双语教材

书名: C++Primer(第4版)(中文版)
原书名: C++ Primer
作者: Stanley B. Lippman, Josee Lajoie, Barbara E. Moo
译者: 李师贤 蒋爱军 梅晓勇 林瑛
中文版书号: 7-115-14151-7
英文版书号: 7-115-15169-7
中文版定价: 99.00元
英文版定价: 99.00元
中文版出版时间: 2006年3月
英文版出版时间: 2006年10月



书名: Ajax基础教程
原书名: Foundations of Ajax
作者: Ryan Asleson
译者: 金灵等
订书号: 7-115-14481-8
定价: 35.00元
出版时间: 2006年2月



书名: JSP高级程序设计
原书名: Beginning JavaServer Pages
作者: Vivek Chopra, Jon Eaves, Rupert Jones, Sing Li, John T. Bell
译者: 朱清江 张文静
订书号: 7-115-14522-9
定价: 55.00元
出版时间: 2006年3月



书名: Exceptional C++ 中文版
原书名: Exceptional C++ Style: 40 New Engineering Puzzles, Programming Problems, and Solutions
作者: Herb Sutter
译者: 刘未鹏
订书号: 7-115-14225-4
定价: 39.00元
出版时间: 2005年12月



书名: C++编程规范: 101条规则、准则与最佳实践
原书名: C++ Coding Standards: 101 Rules, Guidelines, and Best Practices
作者: Herb Sutter, Andrei Alexandrescu
译者: 刘基诚
订书号: 7-115-14205-X
定价: 35.00元
出版时间: 2005年12月



书名: Java解惑
原书名: Java Puzzlers: Traps, Pitfalls, and Corner Cases
作者: Joshua Bloch, Neal Gafter
译者: 陈昊鹏
订书号: 7-115-14241-6
定价: 39.00元
出版时间: 2005年12月



书名: JSP程序设计
原书名: Beginning JavaServer Pages
作者: Vivek Chopra, Jon Eaves, Rupert Jones
译者: 张文静 林琪
订书号: 7-115-14152-5
出版时间: 2005年12月



书名: C++ 必知必会
原书名: C++ Common Knowledge: Essential Intermediate Programming, I.E
作者: Stephen C. Dewhurst
译者: 荣耀
订书号: 7-115-14101-0
定价: 29.00元
出版时间: 2005年11月

版 权 声 明

Original English language edition, entitled *DOM Scripting: Web Design with JavaScript and the Document Object Model* by Jeremy Keith , Cameron Moll , Simon Collison, published by Apress L.P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA.

Copyright © 2006 by Jeremy Keith. Simplified Chinese-language edition copyright © 2006 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L.P.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。

谨以此书献给我的妻子和第一位读者 **Jessica**。

译者序

网上的生活越来越丰富多彩。从最初的(X)HTML 网页，到一度热炒的 DHTML 概念，再到近几年流行起来的 CSS，网站和网页的设计工作变得越来越简便，网上的内容越来越富于变化和色彩。但是，很多网页设计者和网民朋友都不太喜欢 JavaScript，这主要有以下几方面原因。第一，很多网页设计者认为 JavaScript 的可用性很差——早期的浏览器彼此很少兼容，如果想让自己编写出来的 JavaScript 脚本在多种浏览器环境里运行，就必须编写许多用来探测浏览器的具体品牌和具体版本的测试及分支代码（术语称之为“浏览器嗅探”代码）。这样的脚本往往到处是 if...else 语句，既不容易阅读，又不容易复查和纠错，更难以做到让同一个脚本适用于所有的浏览器。第二，对广大的网民来说，JavaScript 网页的可访问性很差——浏览器会时不时地弹出一个报错窗口甚至导致系统死机，让人乘兴而来、败兴而去。第三，JavaScript 被很多网站用来实现弹出广告窗口的功能，人们厌烦这样的广告，也就“恨”屋及乌地厌烦起 JavaScript 来了。第四，“JavaScript”这个名字里的“Java”往往让人们误以为其源于 Java 语言，而实际接触之后才发现它们根本没有任何联系。与 Java 语言相比，JavaScript 语言要简单得多。很多程序员宁肯钻研 Java，也不愿意去了解 JavaScript 的功能和用法。

不管什么原因，JavaScript 曾经不受欢迎的确是一个事实。

现在，情况发生了极大的变化。因为几项新技术的出现，JavaScript 的春天似乎来了。首先，W3C（万维网联盟）推出的标准化 DOM（Document Object Model，文档对象模型）已经一统江湖，目前市场上常见的浏览器可以说没有不支持的。这对网页设计者来说意味着可以用简单的“对象检测”代码来取代那些繁复的浏览器嗅探代码，而按照 DOM 编写出来的 JavaScript 页面不像过去那样容易出问题，这对网民来说意味着浏览体验变得流畅了。其次，最近兴起的 Ajax 技术以 DOM 和 JavaScript 语言（以及 CSS 和 XHTML）为基本要素，基于 Ajax 技术的网站离不开 JavaScript 和 DOM 脚本。

其实，人们对 JavaScript 的恶劣印象在很大程度上来源于早期的程序员对这种语言的滥用。如果程序员在编写 JavaScript 脚本的时候能够把问题考虑得面面俱到，就可以避免许多问题，但可惜的是如此优秀的程序员太少了。事实上，即使是在 JavaScript 已经开始流行起来的今天，如果程序员在编写 JavaScript 脚本的时候不遵守相关的标准和编程准则，也仍会导致各种各样的问题。

在 2002 年前后, CSS 也是一种不太受人们欢迎的 Web 显示语言, 除了用它来改变一下字体, 几乎没有人用它来干其他的事情。但没过多久, 人们对利用 CSS 设计网页布局的兴趣就一发而不可收, 整个潮流也从那时扭转了过来。现在, 掌握 CSS 已经成为许多公司在招聘网站开发人员时的一项要求。

目前, DOM 编程技术的现状与 CSS 技术在 2002 年时的境况颇有几分相似。受 Google Maps 和 Flickr 等著名公司利用 DOM 编程技术推出的 Gmail、Google Suggest 等新型服务的影响和带动, 对 DOM 编程人才的需求正日益增加。有越来越多的人开始迷上了脚本编程技术并开始学习如何利用 DOM 技术去改善而不是妨碍网站的可用性和可访问性。

本书的作者 Jeremy Keith 是 Web 标准计划 DOM Scripting 任务组的负责人之一, 他在这本书里通过大量示例证明了这样一个事实: 只要运用得当, 再注意避开那些“经典的”JavaScript 陷阱, DOM 编程技术就可以成为 Web 开发工具箱里又一件功能强大甚至是不可或缺的好东西。

本书并不是一本参考大全类型的图书, 作者只重点介绍了几种最有用的 DOM 方法和属性。本书的精华在于作者在书中提到的关于 JavaScript 和 DOM 脚本编程工作的基本原则、良好习惯和正确思路。如果读者能通过书中的几个案例真正领悟这些原则、习惯和思路, 就一定能让自己的编程技术再上一个台阶。

这是一本非常实用的好书, 是一本值得一读再读的好书。作为本书的译者, 我们相信它会让每位读者、自建网站的设计者和来自建网站的访问者都受益匪浅。

参加本书翻译的人员还有韩兰、李京山、胡晋平、高文雅。

译者
2006 年 9 月

序

Dave Shea

“JavaScript? 别提了, 用它编写程序特别麻烦。靠它建立起来的网站也不好用, 动不动就弹出一个报错窗口什么的。你要是用了它, 说不定它会趁你没看见的时候踢你的狗一脚呢。”

这就是 JavaScript 给我留下的印象……

作为一名 Web 设计师 (或者称开发人员), 我对 JavaScript 的态度是能不用就不用, 你是不是也和我一样呢? 像我们这样的人肯定还不少。从 “.com” 盛极一时的年代开始, 因为过时的网络教程中有太多质量低劣的示例脚本, 所以人们对 JavaScript 产生抵触情绪, 这根本不让我感到意外。

幸运的是, 有一群像 Jeremy Keith 这样的人在努力地为人们指明正确的方向。在这本书中, 他向我们证明了事情并不像我们想像得那么糟糕; 只要运用得当, 再注意避开那些传统的 JavaScript 陷阱, DOM 脚本编程技术就可以成为 Web 开发工具箱中又一件功能强大甚至是不可或缺的好东西。

事实上, DOM 脚本编程技术的现状让我想起了 2002 年前后的 CSS。在那之前, CSS 一直被人们认为是一种古怪的 Web 显示语言, 除了用它来改变字体, 几乎没有什么人用它来干其他事情。

但没过多久, 人们对利用 CSS 设计网页布局的兴趣就一发而不可收了, 整个潮流也从那时扭转了过来。Wired 和 ESPN 等知名企业用 CSS 重新设计网站的做法改变了人们的旧思想。我也在 2003 年初加入了向 Web 设计人员推广 CSS 技术的 CSS Zen Garden 组织。到了那年年底, CSS 已经从少数人的个人爱好变成了许多公司对网站开发人员的一种预期和要求。

现在, 我们看到 DOM 脚本编程技术也正呈现出一种类似的上升趋势。受 Google Maps 和 Flickr 等著名公司在最近利用 DOM 脚本编程技术推出的新型服务的影响和带动, 对 DOM 脚本编程人才的需求正在日益增加。与过去相比, 有越来越多像你和我这样的人开始迷上了脚本编程技术, 并开始学习如何利用 DOM 的力量增强而不是妨碍网站的可用性。

我们是幸运的, 因为现在有这本书来指导我们。我也想像不出还有什么人能够比 Jeremy Keith 更适合做我们的领路人。作为 Web 标准计划 DOM Scripting 任务组的台柱, 他一直站在脚本编程技术领域各种最新研发方向的最前端。再说, 我从他那里“偷学”代码已经有好几年了, 手中有

了这本书，我也就用不着再不好意思了。

这的确是一本值得一读再读的好书。在磕磕绊绊地看懂了前几段示例代码之后，我已经迫不及待地想看到更多的例子了。学完第 1 章后，我已经完全被吸引住了。Jeremy Keith 是极少数能把高深的概念用简明易懂的语言解释透彻的天才，他的著作不仅可以让我们知道应该怎么做，还能让我们明白为什么要那样做。

是抛开“浏览器嗅探”（browser sniffing）技术而拥抱“对象检测”（object detection）技术的时候了。再也不用假设你们网站的访问者都已经激活了 JavaScript 支持功能了。让我们舍弃那些内嵌在 HTML 文档里的事件处理函数吧，因为我们再也不需要那样做了。Web 一天一个样，而这本书里的技术会让我们每个人都获益。

前 言

这是一本讲述一种程序设计语言的书，但它也适合 Web 设计师阅读。具体地说，本书是为那些喜欢使用 CSS 和 XHTML 并愿意遵守编程规范的 Web 设计师们编写的。

本书由代码和概念两大部分构成。不要被那些代码吓倒：我知道它们乍看起来很唬人，可一旦抓住了代码背后的概念，你们就会发现用一种新语言去阅读和编写代码并没有多么困难。学习一种新的程序设计语言看起来很难，但事实却并非如此。DOM 脚本看起来似乎比 CSS 更复杂，可一旦领悟了它的语法，你们就会发现自己又多掌握了一样功能强大的 Web 开发工具。

归根结底，代码都是思想和概念的体现。我在这里要告诉大家一个秘密：其实没人能把一种程序设计语言的所有语法和关键字都记住。如果有拿不准的地方，查阅参考书就全解决了。

本书不是一本参考大全。我将只讨论编写和运行 JavaScript 脚本所必需的最基本的语法。我的真正目的是为了让你们理解 DOM 脚本编程技术背后的思路和原则。这些思路和原则或许已经是你们早就熟悉的了：预留退路、循序渐进、以用户为中心的设计。这些概念其实对任何前端 Web 开发工作都非常重要。

这些思路贯穿在本书的所有代码示例中。你们将会看到用来创建图片库页面的脚本、用来创建动画效果的脚本和用来丰富页面元素呈现效果的脚本。如果你们愿意，完全可以把这些例子剪贴到自己的代码中，但更重要的是理解这些代码背后的“如何”和“为什么”。

如果你们已经在使用 CSS 和 XHTML 来把设计思路转化为活生生的网页，就应该知道 Web 标准有多么的重要。还记得你们第一次意识到自己根本不必使用标签时感受到的震撼吗？还记得你们是在何时发现自己只需修改一个 CSS 文件就可以改变整个网站的视觉效果吗？DOM 技术有着同样强大的威力。

不过，能力越大，责任也就越大。因此，我不仅想让你们看到用 DOM 脚本实现的超酷效果，更想让你们看到怎样才能利用 DOM 脚本编程技术以一种既方便自己更体贴用户的方式去充实和完善你们的网页。

如果需要本书所讨论的相关代码示例的完整清单¹，到 <http://www.friendsofed.com> 网站搜索本书的主页就可以查到。你们还可以在这个网站找到 friends of ED 出版社出版的其他好书，它

1. 本书代码示例也可从图灵网站 (www.turingbook.com) 下载。——编辑注

们的内容涉及 Web 标准、Flash、DreamWeaver 以及许多细分的计算机领域。

你们对 JavaScript 的热情不应该在合上本书时就冷却下来。我已经在 <http://domscripting.com/> 处开设了一个网站，我将在那里继续与大家共同探讨现代的、标准化的 JavaScript。我希望你们能到该网站看看。与此同时，我更希望本书能够对大家有所帮助。祝你们好运！

致谢

没有我的朋友和同事 Andy Budd¹ (www.andybudd.com) 和 Richard Rutter (www.clagnut.com) 的帮助，本书的面世就无从谈起。Andy 在我们的家乡 Brighton 开设了一个名为 Skillswap (www.skillswap.org) 的免费培训网站。在 2004 年 7 月，Richard 和我在那里做了一次关于 JavaScript 和 DOM 的联合演讲。演讲结束后，我们来到附近的一家小酒馆，在那里，Andy 建议我把演讲的内容扩展成一本书。

我接受了这个想法，并就此事请教了 friends of ED 出版社的 Chris Mills。Chris 非常支持我的想法，完全没有顾虑到我以前从未写过书的事实。friends of ED 的每个人一直都在帮助和鼓励我。我要特别感谢我的项目经理 Beckie Stones 和我的文字编辑 Julie Smith 对我这个初出茅庐的作者给予的支持和谅解。

如果没有两方面的帮助，我大概永远也学不会编写 JavaScript 代码。一方面是几乎每个 Web 浏览器里都有的“view source”（查看源代码）选项。谢谢你，“view source”。另一方面是那些多年来一直在编写让人叹为观止的代码并解说重要思路的 JavaScript 大师们。Scott Andrew、Aaron Boodman、Steve Champeon、Peter-Paul Koch、Stuart Langridge 和 Simon Willison 只是我现在能想到的几位。感谢你们所有人让我分享你们的聪明才智。

感谢 Molly Holzschlag 与我分享她的经验和忠告，感谢她对本书初稿给予反馈意见。感谢 Derek Featherstone 与我多次愉快地讨论 JavaScript 问题，我喜欢他思考和分析问题的方法。

我还要特别感谢 Aaron Gustafson，他在我写作本书期间向我提供了许多宝贵的反馈和灵感。

在写作本书期间，我有幸参加两次非常棒的盛会：在得克萨斯州 Austin 举办的“South by Southwest”和在伦敦举办的@media。我要感谢这两次盛会的组织者 Hugh Forrest 和 Patrick Griffiths，是他们让我有机会结识那么多最友善的人——我从没想过我能有机会与他们结为朋友和同事。

最后，我要感谢我的妻子 Jessica Spengler，这不仅是因为她一直在默默地支持我，更因为她对本书初稿做出的专业帮助。谢谢你，我的人生伴侣。

1. Andy Budd 是 Amazon 超级畅销书《精通 CSS》的作者，该书已由人民邮电出版社出版。

目 录

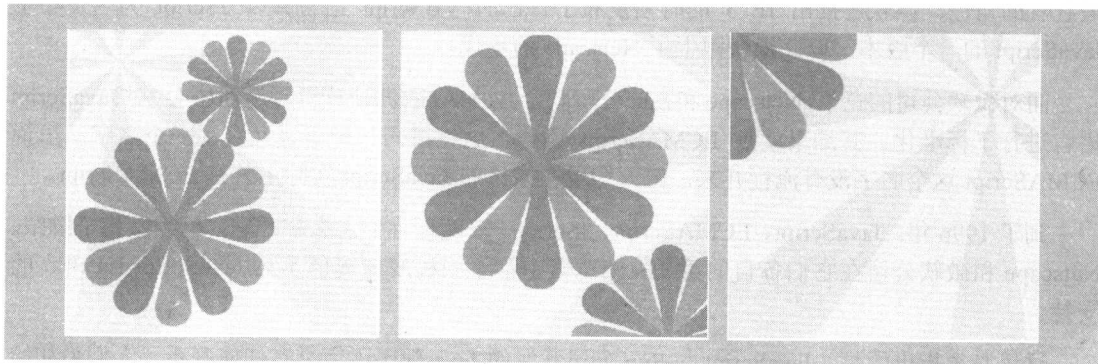
第 1 章 JavaScript 简史	1	3.3 模型：DOM 中的“M”	36
1.1 JavaScript 的起源	2	3.3.1 节点	38
1.2 浏览器之争	3	3.3.2 getElementById()方法	42
1.2.1 DHTML	3	3.3.3 getElementsByTagName()方法	44
1.2.2 浏览器之间的冲突	4	3.4 趁热打铁	46
1.3 制定标准	5	3.4.1 getAttribute()方法	46
1.3.1 浏览器以外的考虑	5	3.4.2 setAttribute()方法	48
1.3.2 浏览器之争的结局	6	3.5 小结	49
1.3.3 新的开始	6	第 4 章 案例研究：JavaScript 美术馆	51
1.4 小结	7	4.1 编写标记语言文档	52
第 2 章 JavaScript 语法	9	4.2 编写 JavaScript 函数	54
2.1 准备工作	9	4.2.1 DOM 之前的解决方案	55
2.2 语法	11	4.2.2 showPic()函数的代码清单	56
2.3 语句	11	4.3 JavaScript 函数的调用	56
2.4 变量	13	4.4 对 JavaScript 函数进行功能扩展	59
2.4.1 数据类型	15	4.4.1 childNodes 属性	59
2.4.2 数组	17	4.4.2 nodeType 属性	60
2.5 操作	20	4.4.3 在 HTML 文档里增加一段 描述性文本	61
2.6 条件语句	22	4.4.4 用 JavaScript 代码改变<p>元素的 文本内容	62
2.6.1 比较操作符	23	4.4.5 nodeValue 属性	63
2.6.2 逻辑操作符	24	4.4.6 firstChild 和 lastChild 属性	63
2.7 循环语句	25	4.4.7 利用 nodeValue 属性刷新<p> 元素的文本内容	64
2.7.1 while	25	4.5 小结	68
2.7.2 do...while	26	第 5 章 JavaScript 编程原则和良好 习惯	69
2.7.3 for	27	5.1 不要误解 JavaScript	69
2.8 函数	27	5.1.1 不要归罪于 JavaScript	70
2.9 对象	31	5.1.2 Flash 的遭遇	71
2.9.1 内建对象	32	5.1.3 质疑	72
2.9.2 宿主对象	33	5.2 预留退路	72
2.10 小结	33		
第 3 章 DOM	35		
3.1 文档：DOM 中的“D”	35		
3.2 对象：DOM 中的“O”	35		

5.2.1 “javascript:” 伪协议	74	7.4.2 “JavaScript 美术馆” 二次 改进版	127
5.2.2 内嵌的事件处理函数	74	7.5 小结	131
5.2.3 有何不好	74	7.6 下章简介	132
5.3 向 CSS 学习	75	第 8 章 充实文档的内容	133
5.4 分离 JavaScript	78	8.1 不应该做的事情	133
5.5 向后兼容性	80	8.2 把“不可见”变成“可见”	134
5.6 小结	82	8.3 原始内容	135
第 6 章 案例研究: JavaScript 美术馆 改进版	83	8.4 XHTML 文档	137
6.1 快速回顾	84	8.5 CSS	138
6.2 解决“预留退路”问题	84	8.6 JavaScript 代码	139
6.3 解决“分离 JavaScript”问题	85	8.6.1 显示“缩略词语表”	139
6.3.1 添加事件处理函数	86	8.6.2 显示“文献来源链接表”	152
6.3.2 进行必要的检查	87	8.6.3 显示“快速访问键清单”	158
6.3.3 创建必要的变量	89	8.7 小结	161
6.3.4 创建循环	90	8.8 下章简介	162
6.3.5 完成必要的操作	91	第 9 章 CSS-DOM	163
6.3.6 完成 JavaScript 函数	91	9.1 三位一体的网页	163
6.3.7 把多个 JavaScript 函数绑定到 onload 事件处理函数上	92	9.1.1 结构层	163
6.4 JavaScript 函数的优化: 不要做太多的 假设	94	9.1.2 表示层	164
6.4.1 不放过每个细节	97	9.1.3 行为层	164
6.4.2 键盘浏览功能	99	9.1.4 分离	165
6.4.3 慎用 onkeypress 事件处理 函数	100	9.2 style 属性	165
6.4.4 把 JavaScript 与 CSS 结合 起来	102	9.2.1 样式信息的检索	167
6.5 DOM Core 和 HTML-DOM	105	9.2.2 设置样式信息	172
6.6 小结	106	9.3 何时该用 DOM 脚本去设置样式 信息	174
第 7 章 动态创建 HTML 内容	109	9.3.1 根据元素在节点树里的位置来 设置样式信息	174
7.1 document.write()方法	110	9.3.2 根据某种条件来设置样式 信息	178
7.2 innerHTML 属性	111	9.3.3 对事件做出响应	182
7.3 DOM 提供的方法	114	9.4 className 属性	184
7.3.1 createElement()方法	115	9.5 小结	188
7.3.2 appendChild()方法	116	第 10 章 用 JavaScript 实现动画效果	191
7.3.3 createTextNode()方法	117	10.1 何为动画	191
7.4 重回“JavaScript 美术馆”	121	10.1.1 位置	192
7.4.1 insertBefore()方法	123	10.1.2 时间	194
		10.1.3 抽象化	198
		10.2 实用的动画	205

10.2.1 问题的提出	205	11.6.2 JavaScript 动画	242
10.2.2 问题的解决	207	11.6.3 内部浏览	247
10.2.3 CSS	208	11.6.4 JavaScript 美术馆	251
10.2.4 JavaScript 代码	210	11.6.5 改进表格	254
10.2.5 与变量的作用域有关的问题	214	11.6.6 改进表单	259
10.3 改进动画效果	216	11.7 小结	268
10.4 最后的优化	219	11.8 下章简介	268
10.5 小结	224	第 12 章 展望 DOM 脚本编程技术	269
第 11 章 学以致用：JavaScript 网站设计实战	225	12.1 Web 的现状	269
11.1 案例背景介绍	225	12.1.1 Web 浏览器	270
11.1.1 原始材料	226	12.1.2 Web 设计师	271
11.1.2 网站的结构	226	12.1.3 三条腿的凳子	272
11.1.3 网页的结构	227	12.2 Ajax 技术	274
11.2 页面视觉效果设计	228	12.2.1 XMLHttpRequest 对象	275
11.3 CSS	229	12.2.2 Ajax 技术的爆发	278
11.4 颜色	231	12.2.3 循序渐进：如何运用 Ajax 技术	279
11.4.1 布局	232	12.2.4 Ajax 技术的未来	281
11.4.2 字型	234	12.3 Web 应用	282
11.5 XHTML 文档	236	12.4 小结	283
11.6 JavaScript 脚本	236	附录 DOM 方法和属性	285
11.6.1 当前页面的标识	238		

第 1 章

JavaScript 简史



本章内容

- JavaScript 的起源
- 浏览器之争
- DOM 的演变史

对网页设计人员来说，这是一个充满着挑战和机遇的时代。近几年来，网页设计工作已经从一种混乱无序和即兴发挥的状态，逐渐发展为一种有着成熟的设计原则可供遵循的流水线作业。有越来越多的网页设计人员开始采用一种标准化的思路来建立网站，而实现这一思路和方法的具体技术则称为“Web 标准”。

当网页设计人员谈论起与 Web 标准有关的话题时，XHTML（可扩展的超文本标记语言）和 CSS（层叠样式表）通常占据着核心地位。不过，由 W3C（万维网联盟）批准并由所有与标准相兼容的 Web 浏览器支持的第三方技术称为 DOM（文档对象模型）。我们可以利用 DOM 去改善文档的可交互性，就像我们可以利用 CSS 给文档添加各种样式一样。

在开始学习 DOM 之前，我们先回顾一下使网页具备可交互性的程序设计语言。这种语言就是 JavaScript，它已经诞生相当长的时间了。