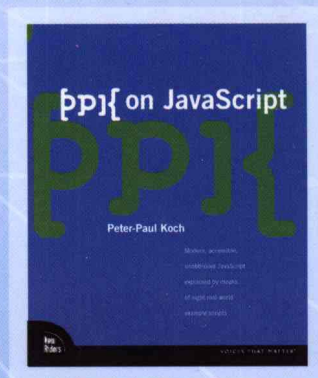


ppk on JavaScript

ppk谈JavaScript

[荷] Peter-Paul Koch 著
淘宝UED 译

- 世界级大师著作
- 提升JavaScript功力的捷径
- 代码全部来自实际商业项目



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Web开发系列

ppk on JavaScript

ppk谈JavaScript

[荷] Peter-Paul Koch 著
淘宝UED 译

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

ppk 谈 JavaScript / (荷) 科克 (Koch, P. P.) 著; 淘宝 UED 译. —北京: 人民邮电出版社, 2008.4
(图灵程序设计丛书)
ISBN 978-7-115-17545-8

I. p… II. ①科… ②淘… III. JAVA 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 013252 号

内 容 提 要

本书全方位介绍了 JavaScript, 主要讨论了浏览器兼容性、可访问性、底层语法以及与 HTML 结构层的协同等问题。书中既包括理论性的讲解, 又给出了相关的示例脚本以进行进一步阐述。通过 8 个真实项目示例, 介绍了 JavaScript 核心语言、BOM、事件处理、DOM、修改 CSS 样式表以及数据检索等内容。

本书适合具有一定网页开发经验的 Web 开发人员阅读。

图灵程序设计丛书

ppk 谈 JavaScript

-
- ◆ 著 [荷] Peter-Paul Koch
译 淘宝 UED
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>
三河市海波印务有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 800×1000 1/16
印张: 22.5
字数: 532 千字 2008 年 4 月第 1 版
印数: 1—5 000 册 2008 年 4 月河北第 1 次印刷
著作权合同登记号 图字: 01-2007-2672 号

ISBN 978-7-115-17545-8/TP

定价: 59.00 元

读者服务热线: (010)88593802 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 *ppk on JavaScript*, 1st Edition, 0321423305 by Peter-Paul Koch, published by Pearson Education, Inc, publishing as New Riders, Copyright © 2007 Peter-Paul Koch.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. Chinese Simplified language edition published by Posts and Telecom Press, Copyright © 2008.

本书中文简体字版由 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。

对本书的赞誉

不知道ppk的网站QuirksMode,说明你可能还没有真正成为资深的JavaScript程序员。多年来,ppk的不懈努力,帮助了全世界无数Web开发人员。如今,他把自己深厚功力和精湛技艺浓缩在一本书中。强烈推荐。

——Roger Johansson, 瑞典资深Web专家

ppk的功力非同一般,从来没有一本书能如此简洁而明确地讲述JavaScript的方方面面。

——陈贤安(Realazy),《精通JavaScript》译者

本书对于Web设计师来说也非常容易理解,必将有助于提升Web设计人员的编程水平。

——Harvey A. Ramer, 纽约著名Web设计师

看了4个月总算看完了,感觉增加了一甲子JavaScript功力。透过几个范例,把基本的知识都讲到了……内容很充实……想学JavaScript,绝对推荐从这本书开始。

——blog.othree.net

ppk在本书中倾注了大量心血,这必将是一部杰作。

——Jeremy Keith,《JavaScript DOM编程艺术》和《Bulletproof Ajax中文版》作者

本书非常全面,是一部提升你JavaScript功力的完美之作。

——Nate Klaiber, Clear Function公司营销总监,资深Web设计师

一部杰作。作者解释了面临各种实际问题时该选择什么技术,还提供了大量宝贵的技巧,强调设计的重要性,弥补了目前JavaScript图书的缺憾。

——JavaRanch.com

这是一本JavaScript传奇。很适合读完《JavaScript DOM编程艺术》之后,继续深入了解JavaScript的各种主题。

——ihower的blog

如果你想更深刻地理解JavaScript,成为一名真正的JavaScript程序员,本书正适合你。

——certforums.com

本书可以说是十全十美的!所有JavaScript新手都应该认真阅读。

——pen-and-ink.ca

译者序

作为译者，竭尽所能向读者介绍这是一本如何优秀的书，原本就是我们“应尽的义务”。但在JavaScript日益成为Web应用程序开发中最前沿、最具活力也最充满创造力的领域时，市场上以及读者的书架上，一定早已存在不少关于JavaScript的好书（而能看到这段文字，说明你也一定早已具备了挑选好书的眼光和能力）。

这是一种必然的现象，每当一种具有生命力和市场号召力的新技术、新应用或新概念诞生的时候，总会伴随着一股潮流和运动，而大量相关的优秀文章与书籍的出现，即是这种潮流和运动的重要特征。

但是，不知读者是否和我们一样意识到，在这一波未平，一波又起的潮流中，鲜见中国人以创造者或推动者的角色出现。今天我学会了Ajax，明天你了解了可用性……而何时我们才能站到那圈子的中央，成为引发地动山摇的震源呢？

在翻译本书的过程中，我们的目标也渐渐地从向读者推荐一本好书，变成了向读者推荐一种平凡精神。这种精神是那么地平凡，很难想见其伟大。每当碰到恼人的浏览器问题，Peter-Paul Koch（下面简称ppk）不会像我们一样只会上网搜索方案，或者绕道而行。他会深入挖掘它的产生原因，搞明白是浏览器的bug还是脚本的bug。某些时候，他还会在“所有的”浏览器上测试某个特性，把它们列成一张兼容性表格。在这之后，他会将成果做成一个网页，发布在自己的网站上与大家分享。这样的工作，他坚持了好多年，他的个人网站收录的浏览器问题与解决方案，比浏览器厂商提供的还要全面和权威。特别值得我们钦佩的是，他的母语不是英语，但他却一直活跃在前端开发的前沿与核心。

所以，ppk用他的行为告诉我们，平凡的人也可以把平凡的事情做到伟大，只要我们坚持！

在阅读本书的时候，读者一定还会从字里行间感受到一种回报社区的感恩心态。正如ppk所说的，他在学习JavaScript的初期，也得到了前端开发社区的帮助和鼓励。所以书中的几个示例，虽然不难（对ppk来说肯定更加没有挑战性），但他却事无巨细地将它们剖开分析，指导初学者从更高的层次来看待JavaScript，引导他们更深入地学习。正是这样一种“平凡人做非凡事”的精神，使ppk成长为一位浏览器专家，一位前端技术专家，一位Web标准的布道者，一位平凡但对社区贡献巨大的开发者。

回头细想，这也是我们翻译此书的最大动力。显然我们高估了自己的实力，又低估了淘宝的发展速度（工作实在很忙），以致花费了数倍于计划的时间才完成本书的翻译。经过上百个加班之后再“加班”的夜晚、几十个令女朋友横眉的周末、两个不能完全放松休息的黄金周，是这样一种“回报社区”的想法，支持着我们的光标不断往前。

总而言之，希望大家阅读此书后能有所收获，特别是在心态和精神上有所收获。如果说ppk

的努力促成了我们对社区的一点贡献，那么，期望我们的努力也能促成读者对社区的一点贡献。如果大家不断地贡献社区，那么，我们也一定能成为潮流的创造者和推动者。

独乐乐，不如众乐乐。高手，必与众同乐。

所以，我们的口号是：坚持与分享。

最后，感谢人民邮电出版社图灵公司的编辑们，容忍了我们长期多次的“跳票”，并不断地鼓励我们前进。感谢淘宝UED的兄弟姐妹们，是这个团队的爱让我们充满斗志。还要感谢我们家中的领导，所有的功劳（和酬劳）都是她们的！

开始阅读吧，带着一种平凡的心态！

淘宝UED

小马 承志 轻侯

2007年底于杭州城西

前言

撰写本书是希望Web开发人员更容易理解一些JavaScript基础知识,这些知识是我成为一名职业Web开发人员以来一点一滴收集来的。

本书反映了我所掌握的知识,自然也同时反映了我的知识的局限性。我是一名Web开发人员,而不是应用程序开发人员,这意味着我将谈论的是浏览器兼容性、可访问性、底层语法以及与HTML结构层的协作,而不涉及模块、设计原则或者抽象层。

8年来,开发网站成为我谋生的手段。这期间,我频繁地使用JavaScript的一些特性,但同时也忽略了很多其他特性。我一直都只使用那些工作中需要用到的工具,而JavaScript工具箱里还存在一大堆我平常很少用的特性。

显然,我不能教你使用那些连我自己都不甚了解的工具,所以本书只探讨那些我工作中用到的语言特性。例如,本书不会涉及面向对象的JavaScript,因为我从来都不需要用它。另外,一本书的篇幅有限,所以我必须做出取舍(有时候真的很难)。最后我选择了那些中高级应用必须使用的JavaScript特性。

背景知识

我假设你已经非常熟悉HTML和CSS了。在第4章我们会看到整洁、有用的HTML结构对写出简洁、高效的脚本是多么关键。而第9章对于不了解CSS基本知识的人来说会是“天书”。

另外,我还假设你已经有一些JavaScript经验,哪怕只处在复制/粘贴的水平。看下面这段代码:

```
var x = document.getElementsByTagName('a');
for (var i=0;i<x.length;i++) {
    if (x[i].className != 'popup') continue;
    x[i].onclick = openPopup;
}
```

我假设你至少知道这段代码会找出页面上的所有链接,且将类为popup的链接定义成弹出。你不需要理解所有的细节(事实上本书的目的就是介绍这些细节),但是你应该对它有大致的了解。

理想情况下,通过阅读本书,我希望每一个初学者能达到中级水平,同时帮助每一个中级水平者走好通往高级的路。

配套网站

我的网站www.quirksmode.org是有关JavaScript的主要资源站点之一。在目录www.quirksmode.org。

org/book/下，你可以找到所有的示例脚本、勘误表等。

浏览器兼容性表格只在线提供，因为它们更新速度很快。放在网上的好处是，当浏览器的兼容模式变化时或发现错误时，我可以及时更新它们。到我的网站上去找它们吧，不管怎样它们已经呆在网上5年了。

本书也引用了一些其他网站。当然，你可以把URL输入到浏览器的地址栏。而更简单的办法是访问配套网站，本书提到的每一个网站在配套网站上都提供了链接。

示例脚本

任何JavaScript图书都需要示例脚本。作者每阐释一点原理，就应该接着展示这个原理在实践中是怎么应用的。示例脚本是最直白的办法。

理想的情况下，这些脚本都应该有实用价值。例如，创建一个脚本把你所有的宠物按名字排序显示出来，然后点击宠物名字就会弹出一张傻乎乎的大头照，这样虽然很不错，但即使这个脚本阐释了一些原理（如对象和弹出框），它的实用价值还是为零。

所以，本书最大特色是提供了8个脚本程序，它们可都是我做的真刀真枪的实际项目，客户花钱买了的。

这8个脚本程序是为7个特定客户的特定商业目标而写的，而且每一个客户都明确表示他们对这些脚本程序很满意。事实上，他们中的6个又给了我新的项目，而那第7个客户没有给我新项目的唯一原因就是他的公司实在太小，只需要一个网站就足够用了。

所以，没有比这些示例更现实和更实用的了。

如何使用示例脚本

用现实中的示例有一点不好，那就是我不能每一章讲一个示例。因为我的客户可不会因为我还没有向你解释过DOM就让他的脚本程序只能用到JavaScript的基本语法和事件。

因此，这些示例用来讲解全书内容，它们全都用到了事件处理、可访问性、DOM脚本编程、HTML结构以及运行时改变CSS。

我把脚本拆成了多个片段，并且把它们按主题进行了排列。你可以在<http://www.quirksmode.org/book/>找到所有的示例脚本。即使你不准备坐在电脑前阅读本书，我也建议你事先对这些脚本有个了解。本书假设你已经对它们有了一个全面的了解，知道它们是做什么的以及为什么要这么做。下面我分别简要介绍一下这些脚本。

限长的文本输入区

2004年9月，一家名叫Lost Boys的网站代理，请我帮他们开发一个脚本程序，可以记住用户在文本输入区（textarea）中键入的字符数量，这就是限长的文本输入区脚本。随后我在A List Apart（网站）发表了一篇名为“JavaScript Triggers（JavaScript触发器）”的文章，用很大篇幅探讨了这个脚本方方面面的内容，接着在自己的博客评论区实现了这段脚本，并且在QuirksMode添加了一段描述。你可以在<http://alistapart.com/articles/scripttriggers>找到这篇文章。

这是一个非常简单的DOM脚本，略微增强了页面。

易用的表单

2003年4月, 荷兰ING银行委托我创建一个演示用的抵押贷款申请表单。这几乎是现实存在的最复杂的表单, 必须想办法隐藏无关的表单域, 只在必要时才显示。例如, “离婚日期”域只有当填表人明确已离婚时才需要显示, 其他任何时候都得把它隐藏起来。

所以, 我写了易用的表单脚本, 为庆祝这一成果, 我还写了“Forms, Usability, and the W3C DOM (表单、可用性和W3C DOM)”一文, 你可以在http://www.digital-web.com/articles/forms_usability_and_the_w3c_dom/找到这篇文章。你会注意到文章中脚本的版本比我们这本书用到的要老一些。

这个较为复杂的DOM脚本会在运行时不停地改变文档的结构。这个例子包含许多有价值的内容, 如自定义属性、HTML元素之间关系的定义、标记的使用、全局事件以及在处理表单域时0级DOM的优势。

表单验证

为网站开发公司SQR开发的表单验证脚本是在2004年春季完工的。他们委托我为一个有大量表单的网站创建模板。很显然, 表单需要验证脚本, 所以我创建了这个表单验证脚本。它与自定义的validation属性绑定在一起, 在用户出错时不是弹出一个警告窗口, 而是在表单域旁边显示错误消息。

表单验证脚本包括基本的表单处理功能, 它会生成一些HTML, 而且用一个很有趣的方法来使用JavaScript关联数组。

下拉菜单

2004年1月, Orchid for Change联系到为他们建立站点。

Orchid需要一个脚本是永恒的经典: 下拉菜单。虽然它的可用性在某些时候还值得怀疑, 但在过去的这么多年里以及未来一两年内, 它似乎还是会继续成为需求量最大的脚本之一。

这个脚本极好地解释了如何让CSS和JavaScript紧密协作来保证良好的可访问性, 如何兼容键盘的操作, 怎样解决因为mouseout事件引发的问题, 以及更多的高级特性。

编辑样式表

下一个例子也是为Orchid写的脚本。

Orchid有很多HTML模板, 带有定义了大量页面元素(页头、导航、主要内容等)位置的样式表。客户只要选择一个预定义的模板就可以创建一个网站。

然而, 为了让站点更有个性, 需要向客户提供微调样式的机会, 主要就是颜色、背景色、字体和排版样式。为此, 每个站点都有另一个样式表, 可以通过它加入独特的设计元素。

我编写了“编辑样式表”脚本来允许客户修改第二个样式表。客户只要要在表单中输入希望的颜色和排版样式, 就可以改动样式表并立刻显示出效果。

遗憾的是, 这个脚本一直处于原型阶段, 因此缺乏不少特性, 特别是无法向服务器提交改变后的样式表并应用到整个站点上。尽管如此, 这个脚本还是提供了一个编辑样式表的范例, 并为HTML表单与脚本协同工作做了准备。

三明治拾取器

这是我最喜欢的脚本范例。

2004年10月,我为Keurslager Drost建网站,他是一位肉店老板,还提供规模不大但很成功的速食餐饮服务。这个站点想要实现食物订购的流水线作业,并让访问者看到他能提供的全部食物。

订购三明治成为这个站点的主要业务。因为超出预算,所以不可能使用一个完整的数据库解决方案,100来种的三明治就这么纵向排列在一个静态的HTML文件中,用户只好在庞大的列表中搜索。

在此情况下,我创建了“三明治拾取器”脚本。一个巨大的数据表包含着所有三明治的数据,而两张开始数据为空的表会包含搜索结果和订购信息。位于页面顶部的搜索框用来搜索三明治,脚本将正确的

这个脚本是一个有趣的由用户参与修改文档结构的例子,它也展示了JavaScript和CSS协作的一些要点。

XMLHTTP 测速计

2005年9月,一家名叫Eend的网络公司委托我为其ISP Wanadoo公司写一个测试ADSL下载速度的脚本。需求非常简单:用户在一个表单中填入邮编和门牌号,然后一段XMLHttpRequest脚本就会在后台悄悄地向服务器发送这些信息,而服务器则会返回包含这用户所能得到的下载速度的一小段XML。

最后的结果(下载速度)会以动画形式显示出来(在最小速度和最大速度之间摆动)。除了视觉上要比较有趣之外,它还要清晰地表达出该下载速度是用户可以期待达到的速度,而不是ISP保证的。这对Wanadoo来说很重要,因为下载速度基本上是不可精确预测的。

这个脚本是一个使用XMLHTTP的简单例子,说明了怎么读服务器的反馈数据,演示了JavaScript动画的原理,还非常清楚地解释了setTimeout和setInterval之间的区别。

在线调查

在XMLHTTP测速计完工后不久,我开始为研究机构RM Interactive写一个不寻常又有趣的脚本。

这家公司受雇于大型网站,负责跟踪用户访问了哪些页面,并且在用户离开网站之后展示一个自愿参加的调查问卷。所有这些都是通过弹出窗口来实现的。

完成这些工作的脚本已经很老了,不能对付屏蔽弹出窗口的软件,对那些不支持JavaScript的用户来说也是不可访问的。而此时RM Interactive即将接手荷兰政府的项目,所以这些可访问性问题必须解决。

因此我完全重写了这个脚本,强化了可用性和可访问性。这个脚本将示范怎么绕过弹出窗屏蔽软件、监视主窗口中发生的事件、恢复弹出窗口与主窗口之间的通信,以及如何确定用户已经离开了网站。

请注意,“在线调查”是唯一使用了两个JavaScript文件的范例:主站的所有页面都加上了

survey.js，而弹出窗口用了popup.js。

错误

这些真实的脚本必然会存在真实的错误。在写这本书的时候我注意到有两个比较特殊的错误。

- ❑ 三明治拾取器有一个可访问性错误。理论上，它的页面对不支持脚本的用户是可访问的，但是事实上在不支持脚本的情况下可用性会被严重阻碍，我们将在2G讨论这点。
- ❑ 表单验证用到了自定义的validation属性（即不属于(X)HTML规范的属性）。写第4章的时候，我发现它和“易用的表单”及“限长的文本输入区”有所不同，其实它可以不用自定义属性。把这些信息转而写到类属性里会让脚本与CSS表现层更紧密地合作。这点将在4B讨论。

我决定不更正这些错误，因为它们都很有教育意义，让我们可以更深入地了解JavaScript与HTML、可访问性、Web标准、CSS之间的关系。

本书结构

我们会在前4章讨论一些比较高级的主题，后6章将关注技术主题。

第1章讨论JavaScript的目标、技术概览以及JavaScript的历史，以说明它的目标是如何被重新定义的。

第2章讨论JavaScript的上下文环境，应是一个满足可用性、可访问性和符合标准的网站，它的结构层、表现层和行为层是分离的。

第3章讨论浏览器之间的兼容性问题 and 处理它们的策略。

第4章讨论JavaScript与HTML结构层的交互、初始化和<script>标签。

第5章～第10章全方位讨论JavaScript技术：核心、BOM（浏览器对象模型）、事件、DOM（文档对象模型）、改变CSS以及数据获取。1B包括这些主题的概述。

本书约定

最后来说明一下本书中的一些版式约定。

节

本书分为10章，每章又划分为标号为A、B、C等的节。

书中涉及的主题密不可分。印刷版图书的一个缺点就是无法使用超链接，因此，我使用了一些章号和节号间的交叉引用。例如，“我们将在4B中详细介绍这个内容”。

代码示例

很多章都布满了代码示例，类似

```
var rows = document.getElementsByTagName('tr');
for (var i=0;i<rows.length;i++) {
    var cells = rows[i].getElementsByTagName('td');
}
```

如果它们来自一个示例脚本，我会给出行号，如

[示例脚本名，第14~17行]

```
var rows = document.getElementsByTagName('tr');
for (var i=0;i<rows.length;i++) {
    var cells = rows[i].getElementsByTagName('td');
}
```

偶尔我也会简化示例脚本，因为我想集中讲述某个特性，而忽略其他内容。在这种情况下，会在行号中标出“节选”字样。“有改动”字样意思是根据旁边正文的说明，对脚本做了些改动。

代码中的某些行太长，书中会分为两行打印。这种情况下，行末会有特殊标记，表示“下一行的代码实际应在本行中”，例如，

```
var writerootRows = document.getElementById('writeroot').
    getElementsByTagName('tr');
```

JavaScript有时允许在标明位置分行，有时不允许。为了避免错误，最好是认为所有代码在源文件中位于一行上。

书中偶尔也会给出错误的反面示例，类似

```
 var rows = getElementsByTagName('tr');
    for (var i=0;i<rows.length;i++) {
        var cells = rows.getElementsByTagName('td');
```

其上下文会解释为什么这些示例是错误的。

代码示例有时也包含用方括号括起的文字，如

```
var rows = [get the rows in the start table];
for (var i=0;i<rows.length;i++) {
    var cells = rows[i].getElementsByTagName('td');
```

这表明，我们将先这么做，但暂不讨论如何做。我希望你把注意力放在示例中的实际代码上，虽然这些应该加以说明，但我并不打算详细讨论，以免分散你的注意力。

你不能将这样的方括号与数组中表示键的方括号相混淆，后者在5K和5L中会用到，它们通常是有明显区别的。

浏览器的名称

本书主要涉及4种浏览器，我通常称之为Explorer、Mozilla、Opera和Safari，其后接版本号。

Mozilla包括基于Gecko呈现引擎的各种浏览器，如Firefox、Camino和Netscape。

Explorer指Windows之上的微软Internet Explorer浏览器（5.0及以上版本）。

我使用这样的名称已经很久了，过去的7年中，我在www.quicksmode.org上也是一直这么使用的。我希望本书可以和这个网站相一致，因此也通篇使用了Explorer。

虽然有些读者会反驳说，它实际上就应该叫Internet Explorer，以区别Windows Explorer，但我认为不会有人会把WWW浏览器与文件管理器相混淆，特别是在它们紧密结合的今天。

如第3章所提到的，Windows版本的Explorer与Mac版本的Explorer是截然不同的浏览器。因为从JavaScript的角度看，Mac版的Explorer即将过时，而且也不是非常好的浏览器。没有限定词的

Explorer总是指Windows版本的。当我提到Mac版时，我会明确地说“Mac的”。

时间

本书最后完成于2006年5月。显然，它不涉及我在这之后所进行的开发活动。

在写本书期间，我用以下浏览器测试了我的脚本：

Explorer 6.0和7.0 beta 2 preview

Mozilla 1.7.12和Firefox 1.5

Opera 8.54和9 beta

Safari 1.3.2

Konqueror 3.0.5

iCab 3.0

目 录

第1章 目标	1	无脚本	29
A 概念回顾	1	没有鼠标	29
在浏览器中处理交互	2	屏幕阅读器	29
胖与瘦	3	屏幕阅读器和事件	31
JavaScript的目标	5	可访问性和可用性	32
B 技术概览	5	F 可访问性规则	32
标准化	6	条理分明的HTML	33
版本	7	产生对脚本用户有意义的内容	33
安全	9	重定向用户	35
C JavaScript的历史	10	键盘用户	36
错误的开始	10	<noscript>标签	36
事实标准	11	G 范例脚本	37
CSS变革	13	限长的文本输入区	37
新的开始	14	易用的表单	38
下一步	15	表单验证	39
第2章 大背景	17	下拉菜单	39
A CSS变革	17	编辑样式表	40
无侵入脚本编程	18	三明治拾取器	40
三个层	19	XMLHTTP测速计	42
分离关注点	20	站点调查	43
B 表现与结构的分离	21	第3章 浏览器	44
CSS修改	21	A 浏览器	44
更改结构还是表现	21	Mozilla家族	44
C 行为与结构的分离	22	Windows的Explorer	45
独立文件中的函数	22	Safari	46
从HTML中移除事件处理程序	23	Opera	46
D 行为和表现的分离	24	其他图形桌面浏览器	46
下拉菜单: hover或mouseover/ mouseout	25	移动电话	47
相同的效果与相似的效果	26	屏幕阅读器	48
E 可访问性概览	28	B 兼容性问题	48
什么是可访问性	28	问题	48
		解决方案	50

C 对象检测	52	易用的表单	88
对象检测是如何工作的	53	表单验证	89
我们该走多远	54	下拉菜单	90
D 浏览器检测	55	编辑样式表	91
为什么浏览器检测行不通	55	三明治拾取器	92
浏览器检测的军备竞赛	57	XMLHTTP测速计	93
拆解浏览器字符串	58	站点调查	93
浏览器检测的正确使用	61	细节决策	94
E 调试	62	第5章 核心	95
错误消息	62	A 基础知识	96
处理浏览器的bug	65	区分大小写	97
范例	67	语句和分号	97
报告浏览器bug	69	注释	98
第4章 准备	71	代码块: {}	99
A HTML和CSS结构	71	运算符	100
HTML结构	71	值	101
CSS结构	72	B 数据类型	102
B 钩子	73	6种数据类型	102
id	73	数据类型转换	105
class	74	+号的问题	106
自定义属性	75	转换成布尔型	107
名/值对	76	显式数据类型转换	107
C 准备页面	78	C 普通运算符	108
设置事件处理程序	78	算术运算符	108
确定访问者状态	79	++和--	109
设置访问	79	=、==和===	109
产生内容	79	!=、<、>、<=和>=	111
定义关系	80	条件运算符?:	112
修改文档结构	81	D 变量	113
D <script>标签	81	变量名称	113
语法	81	关键字var	114
使用多个脚本	83	变量作用域	114
E 初始化	83	E 处理数字	116
load事件	84	八进制和十六进制的数字	116
批量初始化	85	Math对象	117
无侵入初始化	85	toFixed()	118
等待load事件	86	parseInt()和parseFloat()	118
最佳方案	87	F 与字符串打交道	119
F 范例脚本	87	引号	119
限长的文本输入区	87	连接符+	120

字符串对象	120	B 跨浏览器通信	167
G 处理布尔值	123	打开弹出窗口	167
布尔型的变量	124	指向其他窗口	168
布尔运算符	124	重新建立联系	169
检测对象是否存在	126	关闭窗口	172
布尔运算符的高级应用	127	弹出窗口拦截软件	173
H 控制结构	129	C 导航	174
if语句	129	location	174
switch	132	history	175
for、while和do/while	135	D 窗口几何学	176
break和continue	138	screen对象	176
try/catch	140	窗口尺寸	177
I 函数	141	移动、改变大小和滚动窗口	177
基本函数语法	141	获得焦点和失去焦点	178
参数	142	E 杂项	178
return	142	navigator对象	178
作为值的函数	144	alert、confirm和prompt	179
内嵌函数和变量作用域	145	timeout与interval	180
J 对象	147	F document对象	181
点	148	lastModified	181
定义对象	148	referrer	182
处理方法和属性	150	domain	182
this关键字	151	write()	182
全局对象	152	open()与close()	183
K 关联数组	152	G cookie	184
方括号与点符号	153	名/值对	185
关联数组	154	有效期	185
for in语句	155	域和路径	186
L 数组	156	设置cookie	186
嵌套数组	157	document.cookie	187
数组表示法	157	读取cookie	187
数组的使用	158	删除cookie	188
length	158	cookie示范用例	188
push、pop、shift和unshift	159	用户控制权	189
第6章 BOM	161	确定cookie支持	189
A window对象	162	第7章 事件	191
作为全局对象的window	162	A 浏览器兼容性	191
新的页面,新的window对象	163	按实际情况解决	192
作为新全局对象的新窗口	164	概览	192
窗口属性	166	B 事件	192