

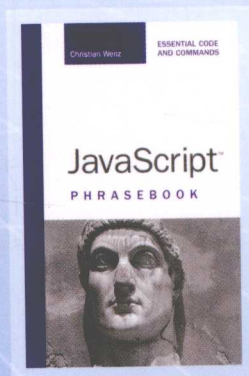
JavaScript Phrasebook

JavaScript

袖珍速查手册

[德] Christian Wenz 著
杜志刚 译

- 简明扼要，易于查询
- 涵盖130多种JavaScript典型编程任务
- 丰富精彩的代码，可立即用于实战



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Web开发系列

TP312/3401

2009

JavaScript Phrasebook
JavaScript
袖珍速查手册

[德] Christian Wenz 著
杜志刚 译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

JavaScript 袖珍速查手册 / (德) 文茨 (Wenz, C.) 著;
杜志刚译. —北京: 人民邮电出版社, 2009.9
(图灵程序设计丛书)
书名原文: JavaScript Phrasebook
ISBN 978-7-115-20071-6

I. J… II. ①文…②杜… III. JAVA 语言—程序设计—
手册 IV. TP312.62

中国版本图书馆CIP数据核字 (2009) 第114108号

内 容 提 要

本书从 JavaScript 开发实践中, 总结提炼出 130 余种典型的任务, 主要涵盖了 JavaScript 基础知识、常用语句、图像与动画、CSS、DOM、Ajax 和 Web 服务等内容。代码短小精悍、讲解简洁到位, 是一本简单、实用、轻量级的 JavaScript 语句查询手册。

本书适合初中级 Web 前端开发人员以及 JavaScript 语言学习者阅读和参考。

图灵程序设计丛书

JavaScript 袖珍速查手册

- ◆ 著 [德] Christian Wenz
译 杜志刚
责任编辑 傅志红
执行编辑 杨 爽
- ◆ 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: 850×1168 1/32
印张: 7.125
字数: 220千字 2009年9月第1版
印数: 1—3 000册 2009年9月北京第1次印刷
著作权合同登记号 图字: 01-2009-3803号

ISBN 978-7-115-20071-6/TP

定价: 29.00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 Phrasebook* by Christian Wenz, published by Pearson Education, Inc., publishing as Sams. Copyright © 2007 by Sams Publishing.

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.

Simplified Chinese-language edition copyright © 2009 by Posts & Telecom Press. All rights reserved.

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。

译者序

JavaScript是在页面中与用户交互的主要技术。随着互联网的发展，越来越强调用户体验，如何恰当地使用JavaScript开发出方便用户的功能，已经成为广大Web开发者和前端设计师关注的焦点。

市面上有很多介绍JavaScript语言理论的书籍，对于开发人员而言，系统学习JavaScript固然重要，然而我们日常所面对的更多是实际项目中遇到的问题，本书对理论和实践进行了很好的融合，更侧重于实践。每一节都以实际开发中可能遇到的问题或要实现的功能为标题（查看本书目录，如果你是开发者，肯定会对其中许多标题感兴趣），再给出解决问题或实现功能的核心语句，然后进行详细的讲述及延伸。文中讲述的解决问题方法都堪称最佳实践，普遍考虑了不同浏览器的兼容性问题和其他可能遇到的实际问题。按照本书所讲述的方式方法去做，可以让你省去一些烦琐的工作量，也会让自己的思维更加清晰，从中找到解决类似问题的规律。译者本人也经常要面对Web前端的开发工作，翻译本书也让我学到了解决一些问题更好的方式。

有必要说明的是本书写作时的主流浏览器版本是IE 6和Firefox 1.5，IE 7还未普及，书中内容主要提到的也是IE 6、IE 7和Firefox 1.5，现在主流的浏览器版本已经是IE 7、Firefox 3.x，然而本书的内容并未过时，兼容IE 6、IE 7和Firefox 1.5的JavaScript代码在新版浏览器中仍然能够正常运行。随着Web浏览器的发展，不同浏览器对JavaScript的实现又走向大一统的趋势。

本书主要针对正在从事开发工作的读者，内容涵盖了开发问题的

方方面面，可以作为参考手册。对于JavaScript初学者，可以将本书作为实践读本，结合本书附带下载的代码进行练习，同样会有所受益。本书由杜志刚翻译，杨爽编辑、统稿及审校。我们感谢我们的家人和朋友，在翻译过程中他们给予了我们莫大的关心和支持。限于译者水平有限，译文中的疏漏和错误之处在所难免，欢迎广大读者批评指正。

前 言

1999年，我曾写了一本关于JavaScript的书。起初该书销量确实不错，后来有略微减少。到去年秋天，该书仍然在销售，已经出到了第7版，但是印数略有下降。

然而，去年年底情况有了很大变化，与同一题材的其他书籍类似，那本书的销量大幅上升。这是怎么回事呢？原因之一是AJAX。这种技术本身并不新，但这个词却是全新的。早在2005年2月，Jesse James Garrett创造了AJAX这一缩略词，从那以后，整个网络世界似乎开始为之疯狂起来。虽然AJAX技术实际上可以在几分钟之内解释清楚，但是它要求对JavaScript的各方面知识有很好的掌握。这正是对高级JavaScript知识需求增长的原因，本书即为适应此需求而撰写。

我们（Damon Jordan、Mark Taber和我）在2005年创作这一系列图书时，我们希望写成一种个性鲜明的语言速查手册（phrasebook）：将常用的语句和表达式翻译成一门语言——本书当然是指JavaScript。然而，与常规的速查手册不同，你还会在代码旁边找到相应解释。否则无论用什么语言，都会很容易产生一些理解困难的情况。本书不是教你JavaScript入门的。本书涵盖了基本的JavaScript技术要素，但我们也力求侧重于JavaScript中高级应用，特别是如果你的JavaScript知识不够，你会在本书中找到常见问题的解决方案，这正是撰写本书的目的。所以你可以将本书作为参考指南，可供快速解决在开发中面对的一些问题，并且还可能在本书中发现一些你之前从未想到过的JavaScript特性。

本书不会对于简短的问题提出冗长僵化的解决方案，主要使用代

码段简明扼要地展示如何处理问题。这样你能在应用程序和具体方案中灵活地使用书中介绍的技术。正因如此，本书展示的只有示例中最为重要的代码段。这些代码经常由<script>内的元素和一些其他HTML标签组合到一起。一个现代Web应用程序至少应当符合XHTML标准，但这并不是本书的焦点。

我们千方百计让代码能在多种浏览器上运行。目前IE和各种Mozilla浏览器（包括Firefox）主宰着市场，Opera、Safari和Konqueror也有一定的份额。所以本书以前两个浏览器为主，适时也会提及其他浏览器不兼容和须留意的地方。考虑市场占有率的因素，本书只讨论当今广为使用的浏览器，Netscape 4 和IE 4不会涉及。

书中的代码示例和更新在<http://javascript.phrasebook.org/>中可以找到。大多数的代码清单都有相应的文件名，所以你可以轻松找到每个语句对应的文件。如果需要反馈意见或是发送勘误，请告诉我！如果你希望找到的语句被遗漏了，也请让我知道。如果你提议在下一版中应当增加哪些新的内容，我会非常感激。（书很小，能容纳的内容有限。）需要写的东西远远超出了本书这个篇幅，我们不得不艰难地割舍一些内容——希望这样的选择能合你意。

最后，感谢下面几位为本书付出辛劳的人：Shelley Johnston 启动这个项目，并让我写了最早的*PHP Phrasebook*；Damon Jordan编辑完*PHP Phrasebook*，又来做本书的编辑；Phil Ballard是本书的技术审稿人。感谢你们为本书所做的大量工作！

同时，也感谢Judith Stevens-Lemoine，她从1999年开始编辑我的另一本JavaScript书。我想感谢她同意我写了本书。期待本书能出到第八版！

Christian Wenz

目 录

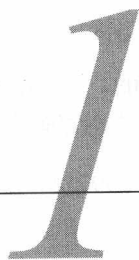
第 1 章 JavaScript 基础	1
1.1 理解JavaScript（和它的历史）	1
1.2 建立测试系统	3
1.3 配置Web浏览器	5
1.4 包含JavaScript代码	6
1.5 使用外部JavaScript文件	7
1.6 动态载入JavaScript文件	7
1.7 使用JavaScript伪URL	9
1.8 用事件处理函数执行JavaScript	9
1.9 应对不支持JavaScript的浏览器	10
第 2 章 常用语句	13
2.1 检测浏览器类型	13
2.2 检测浏览器功能	16
2.3 防止缓存	16
2.4 重定向浏览器	17
2.5 重新载入页面	17
2.6 创建随机数	18
2.7 日期和时间信息	18
2.8 用正则表达式查询	20
2.9 替换文本	21
2.10 在浏览器历史中导航	21
2.11 显示页面的修改日期	22
2.12 检索GET参数	22
2.13 提示用户进行确认	23
2.14 提示用户输入数据	24

第3章 图像与动画	25
3.1 创建鼠标悬浮按钮	25
3.2 预载图像	27
3.3 制作图形动画	29
3.4 拉伸图形	31
3.5 用进度条让页面载入状态可视化	32
第4章 CSS	35
4.1 访问CSS样式	35
4.2 访问CSS类	37
4.3 访问单个样式表	38
4.4 访问单个CSS规则	39
4.5 隐藏网站内容	42
4.6 将JavaScript用于CSS选择器	44
4.7 修改鼠标光标	46
第5章 DOM 与 DHTML	49
5.1 理解DOM	49
5.2 理解DHTML	50
5.3 访问特殊元素	51
5.4 访问标签	51
5.5 测定节点信息	53
5.6 移除元素	55
5.7 添加元素	56
5.8 创建文本元素	57
5.9 使用节点属性	58
5.10 克隆元素	59
5.11 替换元素	61
5.12 用JavaScript数据创建项目符号列表	61
5.13 用JavaScript数据创建表格	62
5.14 修改HTML片段	64
5.15 定位元素	65
5.16 移动元素	67

5.17 创建黏性导航	68
5.18 创建Flash弹出式广告	70
第 6 章 OOP 与事件	73
6.1 创建类	73
6.2 访问类成员	74
6.3 继承类	76
6.4 扩展内建JavaScript对象	78
6.5 响应JavaScript事件	79
6.6 使用键盘事件	81
6.7 用Enter键提交表单	83
6.8 使用鼠标事件	84
第 7 章 cookie	86
7.1 理解cookie	87
7.2 设置cookie	88
7.3 读取cookie	89
7.4 设置cookie失效日期	92
7.5 使用其他cookie选项	93
7.6 删除cookie	94
7.7 检查浏览器对cookie的支持	95
7.8 在单个cookie中存储多条信息	96
第 8 章 表单	98
8.1 从JavaScript的角度理解HTML表单	98
8.2 访问文本域	99
8.3 访问复选框	101
8.4 访问单选按钮	101
8.5 访问选择列表	103
8.6 访问多选列表	104
8.7 禁用表单元素	107
8.8 提交表单	109
8.9 防止表单提交	109
8.10 防止表单重复提交	110

8.11	为某个表单域设置焦点	112
8.12	选中表单域中的文本	113
8.13	在单击文本域时清空内容	115
8.14	验证文本域	117
8.15	验证复选框	118
8.16	验证单选按钮	118
8.17	验证选择列表	119
8.18	自动验证表单	121
8.19	用选择列表实现导航	124
8.20	用选择列表实现层次导航	125
8.21	置空单选按钮组	127
8.22	创建预先填写的日期选择列表	128
8.23	创建带有日期验证的选择列表	129
第 9 章	窗口与框架	132
9.1	使用窗口选项	133
9.2	打开模态窗口	135
9.3	测定屏幕大小	137
9.4	测定窗口大小	138
9.5	调整窗口大小	139
9.6	重定位窗口	140
9.7	打开居中显示的弹出窗口	141
9.8	打开全屏显示的窗口	142
9.9	在窗口的一角打开新窗口	143
9.10	创建网站地图	144
9.11	关闭窗口	145
9.12	检查弹出窗口的拦截方式	147
9.13	同时修改两个框架的内容	149
9.14	使用内嵌框架	152
第 10 章	Web 服务	153
10.1	用PHP创建Web服务	155
10.2	用ASP.NET创建Web服务	156

10.3	从IE调用Web服务	158
10.4	从Mozilla浏览器调用Web服务	160
10.5	从Mozilla浏览器调用ASP.NET Web服务	163
第 11 章	AJAX (及相关主题)	165
11.1	初始化AJAX应用	166
11.2	发送GET请求	168
11.3	发送POST请求	169
11.4	发送同步请求	171
11.5	从服务器接收复合数据	172
11.6	中止HTTP请求	173
11.7	检索HTTP标头	174
11.8	从服务器接收XML	175
11.9	将JSON用于数据(反)序列化	179
11.10	创建等待画面	180
11.11	解决书签问题	182
11.12	解决返回按钮问题	183
11.13	使用XSLT	186
11.14	使用XML库	188
11.15	使用Yahoo! Web服务	190
第 12 章	嵌入式媒体	194
12.1	访问嵌入式媒体	194
12.2	检查插件	195
12.3	应对最新的IE版本	196
12.4	访问多媒体内容	198
12.5	访问Java内容	198
12.6	访问Flash内容	200
索 引		202



本章讲述JavaScript的一些基础内容。我们不去介绍语言的语法本身，因为一来已经有足够的教程和书籍对此做了阐述，二来这也不符合本书作为“袖珍手册”的思路。然而很多基本的方法，例如怎样把JavaScript代码放入页面，在本章中做了详细解释。本章另外介绍了一些JavaScript历史和浏览器大战的故事，可以让你更好地阅读后续章节的内容。

1.1 理解 JavaScript（和它的历史）

JavaScript是一种客户端脚本语言，即它运行在客户端的浏览器中。（JavaScript同样也能在服务器端或是在浏览器之外使用，但这并不是本书的主题。）如果某浏览器支持此语言，JavaScript就被允许访问当前页面，并让脚本定义客户端的属性，重定向用户到其他页面，访问cookie，以及做更多其他的事情。

5

JavaScript诞生于1995年9月，与Netscape浏览器的2.0版同时发布，这也是第一个带有脚本语言的版本。当时，该语言的名称叫做Moncha，而在发布后，改名为LiveScript。由于Netscape与Sun（Java的创造者）之间有着密切的市场合作，因而当年12月又将其改名为JavaScript。

这个名字吸引了众多追随者，微软公司从IE的第3版开始（1996年年中）引入了JavaScript支持。由于法律方面的原因，微软把它叫做JScript。JScript与JavaScript基本上兼容，但微软开始引入了一些仅在IE中有效的特性（除了一些特例，这些特性大多没有流行开来）。

1997年，标准ECMAScript^①（ECMA-262）发布了，JavaScript便是此标准的第一个实现。标准只描述了语言本身，未包含它的宿主特性（例如，如何访问当前浏览器窗口或是打开新窗口）。ECMAScript从1998年开始成为ISO^②标准。

1997至1998年前后，Netscape和微软的浏览器之争达到高潮，两者都在各自浏览器的第5版中加入了新的不兼容的功能。后来，Netscape放弃了发布第5版的想法，转而决定在第6版中一切从头再来，这便让IE抓住时机将其市场占有率扩大到90%。这让后来建立的Mozilla项目经过很多年才恢复元气。非常流行的Firefox浏览器就是基于Mozilla的，并开始从微软手里夺取市场份额。

从JavaScript的角度来看，过去几年并没有发生很多事件。2005年末发布的Firefox 1.5支持新的JavaScript 1.6版，但是改动极少——IE已经远离对该版本的支持。但随着IE 7和Firefox 2.0的发布^③，Web开发者又会面临新的选择。

顺便说一下其他浏览器，它们也支持JavaScript。仅有细微的差异，但开发所有浏览器通用的Web应用仍然是令人极端苦恼的事情。当前支持JavaScript的浏览器如下：

- IE;
- Mozilla 和它的所有派生浏览器（Firefox、Epiphany、Camino、

① ECMAScript是一种由Ecma国际（前身为欧洲计算机制造商协会）通过ECMA-262标准化的脚本程序设计语言。JavaScript和Jscript是ECMA-262标准的实现和扩展。

——译者注

② International Standards Organization，国际标准组织。——译者注

③ 截至本书翻译完成，IE 8已经发布，Firefox最新的正式版本为3.0.10。——译者注

Galeon等);

- Opera;
- Konqueror;
- Safari。

1.2 建立测试系统

刚刚提到过,很多浏览器都支持JavaScript。开发时经常要考虑支持其中的大部分浏览器。例如,网站<http://marketshare.hitslink.com/report.aspx?qprid=3>显示,2006年3月IE和Firefox的浏览器市场占有率^①总和接近95%,紧随其后的是Safari(仅超过3%)。Netscape浏览器占据1%,Opera拥有0.5%,其他所有浏览器的占有率总和为0.5%(包括Konqueror)。

7

因此要考虑以最小的努力在尽可能多的系统^②上测试网站。

实际上,这取决于你的网站用户是谁。如果浏览网站的人大多数使用Mac操作系统,你就有必要为Safari浏览器做大量的测试,因为它是最新的Mac OS X系统的默认浏览器。

无论你在浏览什么的网站,可以看出IE仍然占有大量的市场份额,甚至在很多开源主题网站也是如此。所以你需要IE,理所当然也需要Windows系统来做测试(或者至少带有Windows虚拟机的Virtual PC、VMware这样的工具)。所有的Mozilla浏览器都使用相同的基础代码进行渲染并支持JavaScript,因此这与你使用什么样的平台无关(尽管有一些细微差异)。所以你可以在安装IE的同一台Windows机器上使用Firefox。

① 美国NetApplication公司2009年5月11日发布的Web浏览器市场占有率调查结果为:IE占66.02%,Firefox占20.25%,Safari占6.19%,其他浏览器占7.36%。——译者注

② 指的是操作系统,一般不同的操作系统对应不同的浏览器。——译者注

Opera也可以运行在Windows上（还可运行在Linux和Mac等其他系统上），换句话说Windows上可以安装多个浏览器。

还剩下的两个主流浏览器是Safari和Konqueror，当使用KDE^①窗口管理器时，Konqueror是默认的浏览器。幸运的是两者或多或少使用了相同的基础代码：Safari使用KHTML引擎，这也是Konqueror渲染的核心。因此，有如下两种选择。

- ❑ 安装一台带有KDE和Konqueror的Linux机器（或者Linux虚拟机）。
- ❑ 安装Mac操作系统（或是购买一台Intel Mac，用BootCamp配置Windows和OS X的双启动模式）。

这样至少能够让你拥有一个不错系统来开始测试。随着网站用户规模增大，支持尽可能多的目标系统就变得越发重要。总有一天，你没有别的选择，只能逐个安装并测试所有浏览器。

就IE来说，当前主流的是版本6，而5.x已经基本没人用了，4以前的版本早已绝迹。因此，通常在IE 6上测试基本上已经足够。^②虽然有时也需要测试不同版本的浏览器，但因为每个操作系统只允许安装一个版本的IE，就需要为每个版本安装相应的Windows操作系统。

我偶然发现了同时安装多个版本IE的解决方案。你可以在这里找到这个解决方案http://labs.insert-title.com/labs/Multiple-IEs-in-Windows_article795.aspx，更多的信息在这里<http://www.positioniseverything.net/article/multiIE.html>。

最后，确保分别在JavaScript开启和关闭的状态下测试网站。

① K Desktop Enviroment（K桌面环境）的缩写。是一种著名的运行于Vnix以及Linux、FreeBSD等类Vnix操作系统上面的自由桌面环境。——译者注

② 美国NetApplication公司2009年5月11日发布的Web浏览器市场占有率调查结果为：IE 7占44.51%、IE 6占17.52%、IE 8占3.99%，因此，IE 7已经成为主流版本，建议将IE 7作为主要的测试版本。——译者注