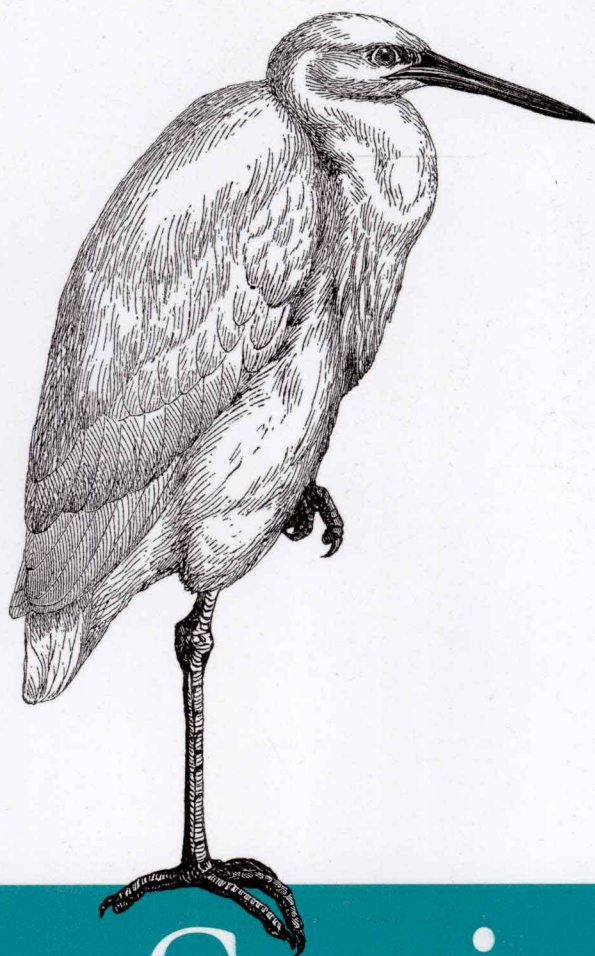


JavaScript Cookbook



JavaScript 经典实例

O'REILLY®

中国电力出版社



Shelley Powers 著

李强 译

JavaScript经典实例

Shelley Powers 著
李强 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

图书在版编目 (CIP) 数据

JavaScript经典实例/ (美) 鲍尔斯编著; 李强译. —北京: 中国电力出版社, 2011.8

书名原文: JavaScript Cookbook

ISBN 978-7-5123-2058-1

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

中国版本图书馆CIP数据核字 (2011) 第172070号

北京市版权局著作权合同登记

图字: 01-2010-7925号

©2010 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2011.

Authorized translation of the English edition, 2010 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2010。

简体中文版由中国电力出版社出版2011。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。

书 名/ JavaScript经典实例

书 号/ ISBN 978-7-5123-2058-1

责任编辑/ 刘炽

封面设计/ Karen Montgomery, 张健

出版发行/ 中国电力出版社 (<http://www.cepp.sgcc.com.cn>)

地 址/ 北京市东城区北京站西街19号 (邮政编码100005)

经 销/ 全国新华书店

印 刷/ 航远印刷有限公司

开 本/ 787毫米×980毫米 16开本 33.125 印张 625千字

版 次/ 2012年3月第一版 2012年3月第一次印刷

印 数/ 0001—3000册

定 价/ 78.00元 (册)

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”，创建第一个商业网站（GNN），组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

目录

前言	1
第1章 使用JavaScript字符串	11
1.0 简介	11
1.1 连接两个或多个字符串	13
1.2 连接字符串和另一种数据类型	14
1.3 条件比较字符串	15
1.4 在字符串中查找子字符串	18
1.5 从一个字符串提取子字符串	20
1.6 检查一个存在的、非空的字符串	21
1.7 将一个关键字字符串分解为单独的关键字	23
1.8 插入特殊字符	25
1.9 处理textarea的单个行	27
1.10 去除字符串末尾的空白	28
1.11 左补充或右补充一个字符串	29
第2章 使用正则表达式	32
2.0 简介	32
2.1 测试一个子字符串是否存在	36
2.2 测试不区分大小写的子字符串匹配	37
2.3 验证社会安全号码	38
2.4 找到并突出显示一个模式的所有实例	39
2.5 使用新字符串替换模式	43

2.6 使用捕获圆括号交换一个字符串中的单词	43
2.7 使用正则表达式来去除空白	47
2.8 使用命名实体来替代HTML标签	48
2.9 搜索特殊字符	48
第3章 日期、时间和定时器	51
3.0 简介	51
3.1 打印出今天的日期	53
3.2 打印出UTC日期和时间	54
3.3 打印出一个ISO 8601格式日期	55
3.4 把一个ISO 8601格式的日期转换为Date对象可接受的一种格式	57
3.5 创建一个特定的日期	60
3.6 规划未来的一个日期	60
3.7 记录流逝的时间	61
3.8 创建一个延迟	62
3.9 创建重复性定时器	63
3.10 使用带有定时器的函数闭包	64
第4章 使用Number和Math	67
4.0 简介	67
4.1 保持一个递增的计数	69
4.2 把十进制数转换为一个十六进制值	71
4.3 创建一个随机数生成器	72
4.4 随机产生颜色	72
4.5 把表中的字符串转换为数字	74
4.6 把表中一列的所有数字加和	74
4.7 在角度和弧度之间转换	77
4.8 找到页面元素可容纳的一个圆的半径和圆心	77
4.9 计算圆弧的长度	80
第5章 使用数组和循环	81
5.0 简介	81
5.1 循环遍历数组	83

5.2 创建多维数组	84
5.3 从数组创建一个字符串	85
5.4 排序数组	86
5.5 按顺序存储和访问值	87
5.6 以相反的顺序存储和访问值	88
5.7 创建一个新数组作为已有数组的子集	89
5.8 在数组中搜索	91
5.9 将一个多维数组扁平化	92
5.10 搜索和删除或替换数组元素	93
5.11 对每个数组元素应用一个函数	94
5.12 对数组中的每个元素执行一个函数并返回一个新数组	96
5.13 创建一个过滤后的数组	97
5.14 验证数组内容	98
5.15 使用一个关联数组来存储表元素名和值	101
第6章 使用JavaScript函数构建重用性	105
6.0 简介	105
6.1 创建一段可重用的代码	106
6.2 把单个数据值传递到函数	107
6.3 把复杂的数据对象传递给函数	108
6.4 创建一个动态运行时函数	110
6.5 把一个函数当做参数传递给另一个函数	112
6.6 实现递归算法	113
6.7 创建能够记住其状态的函数	115
6.8 使用一个通用的科里化函数提高应用程序性能	118
6.9 使用缓存计算 (Memoization) 来提高应用程序性能	120
6.10 使用匿名函数包装全局变量	123
第7章 处理事件	126
7.0 简介	126
7.1 检测页面何时完成载入	129
7.2 使用Event对象捕获鼠标点击事件的位置	130

7.3 创建一个通用的、可重用的事件处理函数	133
7.4 根据修改的条件来取消一个事件	136
7.5 阻止事件在一组嵌套元素中传播	137
7.6 捕获键盘活动	140
7.7 使用新的HTML 5拖放	143
7.8 使用Safari方向事件和其他移动开发环境	151
第8章 浏览器模块	153
8.0 简介	153
8.1 请求Web页面访问者确认一项操作	154
8.2 创建一个新的、下拉式的浏览器窗口	155
8.3 找到关于浏览器的访问页面	155
8.4 警告Web页面访问者将要离开页面	157
8.5 根据颜色支持更改样式表	158
8.6 根据页面大小修改图像尺寸	159
8.7 在CMS模板页面中创建面包屑路径	161
8.8 将一个动态页面加入书签	164
8.9 针对后退按钮、页面刷新来保持状态	167
第9章 表单元素和验证	169
9.0 简介	169
9.1 访问表单文本输入值	169
9.2 动态关闭或打开表单元素	171
9.3 根据一个事件从表单元素获取信息	171
9.4 当点击单选按钮的时候执行一个动作	174
9.5 检查一个有效的电话号码	177
9.6 取消表单提交	178
9.7 阻止重复表单提交	179
9.8 隐藏和显示表单元素	181
9.9 根据其他表单选择修改一个选项列表	184
第10章 调试和错误处理	187
10.0 简介	187

10.1 优雅地处理无JavaScript支持的情况.....	187
10.2 检查函数中的错误.....	190
10.3 对于简单调试使用一条警告.....	191
10.4 捕获一个错误并提供优雅的错误处理.....	192
10.5 初始化可管理的错误.....	194
10.6 使用Firefox的Firebug.....	195
10.7 使用Firebug设置一个断点并查看数据.....	199
10.8 Firefox和Console.....	200
10.9 使用IE的内建调试器.....	204
10.10 使用IE Developer Tools设置一个断点.....	207
10.11 Opera的Dragonfly.....	209
10.12 使用Dragonfly设置一个断点.....	211
10.13 打开Safari的开发工具.....	212
10.14 使用Safari调试器设置断点.....	217
10.15 Chrome中的调试.....	219
第11章 访问页面元素.....	221
11.0 简介.....	221
11.1 访问一个给定的元素并找到其父元素和子元素.....	225
11.2 访问Web页面中所有的图像.....	226
11.3 在一篇文章中找出所有的图像.....	232
11.4 使用Selectors API找出文章中的所有图像.....	233
11.5 找出一组元素的父元素.....	236
11.6 突出显示每个元素中的第一个段落.....	237
11.7 对无序列表应用条纹主题.....	240
11.8 创建一个给定类的所有元素的一个数组.....	242
11.9 找出共享同一属性的所有元素.....	242
11.10 找出所有选中的选项.....	244
11.11 把一个表行中所有值加和.....	245
11.12 获取元素属性.....	247
11.13 获取一个元素的样式信息.....	249

第12章 创建和删除元素和属性	252
12.0 简介	252
12.1 使用innerHTML：一种添加内容的快速而容易的方法	252
12.2 在已有页面元素前插入元素	253
12.3 在页面的末尾附加一个新元素	257
12.4 触发IE的旧版来样式化新元素	258
12.5 插入一个新的段落	258
12.6 给新的段落添加文本	259
12.7 向一个已有元素添加属性	262
12.8 测试一个布尔值	263
12.9 删除一个属性	264
12.10 移动一个段落	265
12.11 使用脚注项目符号替代链接	265
12.12 向已有的表添加行	269
12.13 从一个div元素删除一个段落	271
12.14 从HTML表格删除行	273
12.15 修改元素的CSS样式属性	275
 第13章 使用Web页面空间	 279
13.0 简介	279
13.1 确定Web页面的区域	280
13.2 度量元素	282
13.3 在页面中定位元素	283
13.4 隐藏页面区段	286
13.5 创建可折叠的表单区段	287
13.6 添加一个页面覆盖	291
13.7 创建标签页	294
13.8 创建基于悬停的弹出信息窗口	299
13.9 折叠边栏或调整其大小	302

第14章 使用JavaScript、CSS和ARIA

创建交互式 and 可访问性效果..... 306

14.0 简介	306
14.1 显示隐藏的页面区段	308
14.2 创建警告消息	309
14.3 突出显示遗漏数据或数据不正确的表单字段	311
14.4 给页面覆盖添加键盘可访问性	317
14.5 创建可折叠的表单区段	321
14.6 显示一个带颜色的闪烁以表示一个动作	325
14.7 给标签页应用程序添加ARIA属性	329
14.8 动态区域	332

第15章 创建富媒体和交互应用程序 334

15.0 简介	334
15.1 在画布中创建基本的图形（使用canvas元素）	335
15.2 在Internet Explorer中实现画布应用程序	338
15.3 在画布中创建一个动态的线条图表	339
15.4 向一个SVG文件添加JavaScript	342
15.5 从Web页面脚本访问SVG	344
15.6 在Internet Explorer中模拟SVG	347
15.7 为嵌入到HTML中的SVG增加交互性	348
15.8 使用Math函数在SVG中创建一个实际的、走动的模拟时钟	354
15.9 在HTML中加入SVG和画布元素	357
15.10 在Firefox和WebKit/Safari中调试WebGL支持	359
15.11 当一个音频文件开始播放的时候运行一个例程	360
15.12 用JavaScript和video元素控制视频	362

第16章 JavaScript对象 367

16.0 简介	367
16.1 定义一个基本的JavaScript对象	368
16.2 保持对象成员私有	369
16.3 用原型扩展对象	370

16.4 给对象添加Getter/Setter	373
16.5 继承一个对象的功能	374
16.6 通过定义一个新的属性来扩展对象	377
16.7 枚举一个对象的属性	383
16.8 阻止对象可扩展性	386
16.9 阻止对象添加和修改属性描述符	387
16.10 阻止对对象的任何修改	388
16.11 一次性对象和为你的JavaScript提供命名空间	390
16.12 用Prototype.bind再次发现“this”	392
16.13 将对象方法链化	394
第17章 JavaScript库	397
17.0 简介	397
17.1 包装你的代码	398
17.2 使用JsUnit测试代码	400
17.3 简化你的库	404
17.4 寄存库	405
17.5 使用一个外部库：构建于jQuery框架之上	408
17.6 使用已有的jQuery插件	411
17.7 把库转换为一个jQuery插件	412
17.8 安全地把几个库组合到你的应用程序中	416
第18章 通信	420
18.0 简介	420
18.1 访问XMLHttpRequest对象	421
18.2 为传输准备数据	423
18.3 确定查询调用的类型	424
18.4 为Ajax请求添加一个回调函数	427
18.5 检查一个错误条件	429
18.6 处理一个文本结果	429
18.7 （使用JSONP）对另一个域进行Ajax请求	430
18.8 从服务器填充一个选项列表	432

18.9 使用定时器以新数据自动更新页面	434
18.10 使用PostMessage跨窗口通信	438
第19章 使用结构化数据	442
19.0 简介	442
19.1 处理从Ajax调用返回的一个XML文档	443
19.2 从一个XML树提取相关信息	444
19.3 使用JSON产生一个JavaScript对象	449
19.4 解析一个JSON格式化字符串	451
19.5 使用JSON把一个对象转换为过滤的/转换的字符串	452
19.6 把hCalendar微格式注释转换为一个画布时间表	454
19.7 清除页面RDFa并且使用rdfQuery和jQuery RDF插件将其转换为JSON	457
第20章 持久化	463
20.0 简介	463
20.1 给URL附加持久性信息	464
20.2 创建一个Cookie来跨页面持久化信息	468
20.3 使用History.pushState方法和window.onpopstate来持久化信息	471
20.4 针对客户端存储使用sessionStorage	475
20.5 创建一个localStorage客户端数据存储项	482
20.6 使用关系数据存储来持久化数据	485
第21章 JavaScript创新用法	487
21.0 简介	487
21.1 创建一个浏览器插件或扩展	488
21.2 创建桌面和移动挂件	493
21.3 使用PhoneGap为iPhone、Android和BlackBerry开发JavaScript应用程序	498
21.4 使用JavaScript扩展工具	500
21.5 使用Web Workers和File API创建高效的桌面应用程序	504

前言

我在15年前编写了自己的第一本JavaScript图书，并且那时候必须匆忙地寻找足够的材料来充实一本图书。对于本书，我从数百种用法中选择要包含什么内容。毕竟，这些年看着JavaScript发展，我仍然对于JavaScript的用法可以变得如此深远而感到惊讶。在我看来，没有比它更加有用的编程语言或开发工具了。这是在HTML中唯一的如此广泛地应用的技术。

本书针对那些已经接触过JavaScript，并且想要尝试新技术的人，或者想要加强对JavaScript基础知识和高级功能的掌握的人。通过本书，我将介绍：

- 如何使用String、Array、Number和Math这样的JavaScript对象。
- 创建可重用的对象。
- 在文档对象模型（Document Object Model，DOM）中查询和创建新的元素。
- 使用新的Selectors API以更高效且有目标地查询。
- 将JavaScript与HTML 5新技术一起使用，例如，新的媒体元素video和audio。
- 创建可交互的应用程序。
- 管理Web页面空间。
- 以不同的方式存储数据，从简单的到复杂的。
- 使用JavaScript和Scalable Vector Graphics (SVG)以及canvas元素。
- 使用一些有趣的数据结构，例如微格式和RDFa。
- 将库打包以供其他人使用，以及在你的应用程序中使用其他的库。

- 通过Accessible Rich Internet Applications (ARIA)来确保你的应用程序的可访问性。
- 在典型桌面浏览器以外的环境中工作，例如，创建移动手机Web应用程序，用新的行为扩展Photoshop。
- 使用并创建jQuery插件。
- 开发Ajax应用程序。
- 使用浏览器的调试器来调试应用程序。
- 使用新的HTML 5拖放功能。
- 使用新的HTML 5跨文档技术来通信。
- 使用Web Workers实现并发编程。
- 使用File API直接在客户端JavaScript中访问一个桌面文件。

本书不是关于现今的JavaScript用法的百科全书，当然还没有一本书能够覆盖所有这些内容。但是，希望你读完本书后，对于能使用JavaScript做的事情能够有更广的认识。

你一定会感到津津有味！

目标读者、所需背景和阅读方法

本书的读者应该对Web开发和JavaScript的用法有所了解。此外，各个小节的形式，意味着我们会关注特定的任务，而不是提供一个全面的一般性介绍。我不会介绍JavaScript主题的每一个方面，如String。相反，我们会关注与该主题相关的、较为常见的任务或实战。

书中会有很多代码，一些是代码段，另一些是完整的应用程序。各个小节也是相互引用的，因此，如果在一个小节中我所介绍的特定话题，在另一个小节中也有所涉及，我会在该小节的“参见”部分包含这些信息。为了给你提供帮助，我还为所有的小节创建了示例代码，以便你可以下载并立即使用。

目标浏览器

整个本书中，我会提到目标浏览器。本书中主要的示例代码，都针对最常使用的浏览器的新版本而设计，并且进行了测试。这些浏览器是：

- 在Mac和Windows上运行的Firefox 3.6x。
- 在Mac和Windows上运行的Safari 4.0x。

- 在Mac和Windows上运行的Opera 10.x。
- 在Windows上运行的Chrome 5.x。
- 在Windows上运行的Internet Explorer 8。

我没有使用Linux的机器来测试这些环境，但是，运气好的话，大多数小节在Linux环境下使用Firefox也能工作。我也没有一个System 7来测试IE9的预览版，但是，大多数应用程序应该能够工作，包括那些使用了SVG（IE9新添加的功能）的应用程序。

一些小节需要特定的环境，例如，一个移动设备或模拟器，或者浏览器的beta版（或alpha版）发布。对于那些在目标浏览器中无法工作的示例，或者需要特定环境或浏览器的示例，我做出了说明。此外，我还介绍了只在某种浏览器的alpha/beta版中实现了的新技术和API。同样，我也对浏览器支持给出了说明。

很多示例在IE6中无效。在编写本书之前，我决定不对IE6提供支持，包括任何的解决方案代码也是如此。很多主流的站点现在都不再支持太早或不太安全的浏览器，包括Amazon、Google、YouTube和Facebook。此外，IE6所需的解决方案如此知名，并且在互联网上有很好的记载，这使我觉得不必再在本书中提及。

在IE8中有效的大多数示例，也会在IE7中有效，只有一部分例外。IE7不支持常见属性上的getAttribute/setAttribute，例如style、id和class，并且根本不支持hasAttribute。此外，IE7不支持CSS选择器，也不支持Selectors API方法，例如document.querySelectorAll（将在第11章中介绍）。

IE7无效的地方，要么我在可以下载的示例代码中提供特定于IE7的解决方案，或者我在该小节中注明不支持，或者二者都做到。

示例代码体例

全书中有很多代码片段和功能完备的示例。大多数都是基于HTML的，但是，也有一些基于XHTML，即HTML的XML序列化。此外，大多数示例都是基于HTML 5的，尽管我也已经使用了几个其他的HTML版本，特别是在SVG的示例中：

HTML5

```
<!DOCTYPE html>
```

XHTML5

```
<!DOCTYPE html>  
<html xmlns="http://www.w3.org/1999/xhtml">
```

XHTML+SVG

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN"
"http://www.w3.org/2002/04/xhtml-math-svg/xhtml-math-svg.dtd">
```

还有几个基于HTML的示例的不同版本。如果该示例是X/HTML5，你不需要对style或script元素使用type属性。此外，在很多XHTML示例页面中，我用一个CDATA区段把代码包围起来，如下所示：

```
<script>
//<![CDATA[
Preface | xv
...
//--><!]]>
</script>
```

对XHTML中的脚本块使用一个CDATA区段的原因是，像尖括号(< >)和&这样的字符，在JavaScript中是有意义的，但是，它们作为标记也是有意义的。当一个XML解析器看到这些字符的时候，它想要将其解释为标记。为了预防这一点，CDATA区段告诉解析器，忽略该区段。

我试图在同一页面中保持所有的样式设置和脚本，从而简化示例。然而，在现实世界中，你需要尽可能地将样式表和脚本分别放到不同的文件中。这么做是为了保持HTML文件的整洁，同时，使得样式和脚本都易于修改。

阅读方法

在本书中，我们首先从基本的JavaScript功能开始介绍，因为这仍然是应用程序开发必备的基础。我还使用一些小节，介绍了较新的功能，包括使用canvas元素，尝试新的跨域挂件通信技术（PostMessage），使用新的File API和Web Workers，将代码与流行的jQuery库整合，甚至使用新的HTML video和audio元素（其中有很多乐趣）。我还介绍了JavaScript的一些新用法，例如，在移动设备和离线桌面应用程序中的使用，以及访问页面中的微格式和RDFa这样的元数据。

组织方式

本书介绍的是相对深奥的话题，主要覆盖了在过去几年里，我所看到的这一领域有趣的和发展的地方。还包含了对新的ECMAScript 5和HTML5创新的介绍。

然而，本书确实包含了两个相对通用的部分：第一个部分关注已有的JavaScript功能和对象，第二个部分更多地关注JavaScript在环境中的应用，例如浏览器。如果你是一个JavaScript新手，建议你先学习完前10章中的所有小节，然后再学习本书后面的章节。