

JavaScript

精粹 (修订版)

The JavaScript Anthology

101 Essential Tips, Tricks & Hacks



[英] James Edwards 著
[澳] Cameron Adams 译
高铁军

JavaScript

精粹 (修订版)

The JavaScript Anthology

101 Essential Tips, Tricks & Hacks

[英] James Edwards 著
[澳] Cameron Adams 译
高铁军



YZLI0890169937

人民邮电出版社

北京

图书在版编目 (CIP) 数据

JavaScript精粹 / (英) 爱德华兹 (Edwards, J.),
(澳) 亚当斯 (Adams, C.) 著; 高铁军译. — 修订本
— 北京: 人民邮电出版社, 2013.1
ISBN 978-7-115-29722-8

I. ①J… II. ①爱… ②亚… ③高… III. ①
JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字 (2012) 第248133号

版 权 声 明

Copyright©2012Posts & Telecom Press

Authorized Chinese Simplified translation of the English edition of *JavaScript Anthology, First Edition*
(ISBN 9780975240267) ©2006 Sitepoint Pty.Ltd.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all
rights to publish and sell the same.

本书由 SitePoint Pty.Ltd 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何
方式或任何手段复制和传播。

版权所有, 侵权必究。

JavaScript 精粹 (修订版)

- ◆ 著 [英] James Edwards [澳] Cameron Adams
- 译 高铁军
- 责任编辑 陈冀康
- ◆ 人民邮电出版社出版发行 北京市东城区夕照寺街14号
- 邮编 100061 电子邮件 315@ptpress.com.cn
- 网址 <http://www.ptpress.com.cn>
- 三河市海波印务有限公司印刷
- ◆ 开本: 800×1000 1/16
- 印张: 26.25
- 字数: 600千字 2013年1月第1版
- 印数: 1-3000册 2013年1月河北第1次印刷

著作权合同登记号 图字: 01-2012-7993 号

ISBN 978-7-115-29722-8

定价: 59.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

内 容 提 要

本书主要介绍 JavaScript 应用中一些常见的问题及其解决方法，从最基础的数字、字符串、数组到进阶的 DOM、表单验证、cookie，再到较为高级的 AJAX，书中均有涉及。本书覆盖现在非常流行和通用的技术，提出很多出现频率较高的 Web 开发常见问题，并提供了大量的技巧和解决方案，具有很强的实用性和通用性，书中的代码 also 具有很强的兼容性。本书还介绍了一些面向对象编程内容，这部分内容可以帮助用户以更高的角度来看待自己的设计，以提高代码的质量和可维护性。最为难得的是，由于具有丰富的业界实践经验，作者在本书中不仅仅提出了对各种问题的解决方法，还展现了自己的思考方法和一些良好的程序设计习惯，与具体的细节性知识相比，这些无疑更加具有价值。

本书适合参与创建网站以及开发 Web 应用程序的工程技术人员和管理人员阅读参考。

前 言

对很多人来说，提起 JavaScript，常常伴随着很多不愉快的记忆，例如讨厌的弹出式窗口、恼人的鼠标尾迹，还有令人沮丧的禁止鼠标右键的脚本。如果常常遇到这些脚本，就知道那是多么乏味和无聊。不过，JavaScript 确实是一种成熟、专业的脚本语言，广泛地应用于现在的大多数网站，而且，它也几乎是所有基于 Web 的应用程序的关键性组件。别奇怪，我们确实是在谈论同一种技术。

当然，回顾过去，我认为很多人对 JavaScript 的感受还是比较准确的：JavaScript 确实名声不太好，随着大量网站的蓬勃兴起，JavaScript 用于产生弹出式窗口，隐藏本来需要严格审查的代码，产生各种特效。最近几年，JavaScript 常常被单列出来，作为各种问题的罪魁祸首。其实，这不是技术本身的错，完全是由于毫无规划地滥用给 JavaScript 带来了坏名声。

不过，随着远程脚本技术（一般指 AJAX 技术）的兴起，JavaScript 技术开始复苏。来自不同专业的设计者、开发者、程序员开始对这种曾经的主导技术产生兴趣，而且对它留下深刻的印象。随着 Web 应用和桌面应用之间的界限开始变得越来越模糊，浏览器提供商以及其他一些技术公司已经开始用一种新的目光审视这种强大语言的潜力。

JavaScript 已经成为开发大量新应用程序的关键性组件，是时候对它进行一番深入的探讨了。

读者对象

任何参与创建网站以及开发 Web 应用程序或者对其感兴趣的人都应该阅读本书。

如果您是用复制 / 粘贴的方式来解决日常问题的网站管理员，那么我们为您提供了更好的解决方案。如果您是有一些 JavaScript 经验的程序员，可以在本书的代码和讨论中发现对您的实际工作有很大裨益的高级部分。如果您是设计师，对编写代码感兴趣，或者是学生，想对此有个初步的了解，您会发现在本书中有大量精彩的例子，以很直观的方式给出了相关的概念和方法。

不管您现有的 JavaScript 知识处于何种水平，我们都希望这本书能够对您有用，并且能够成为指导您实践的最好的参考材料。

本书内容

第 1 章——JavaScript 初步 本章比起其他章节更加理论化，首先对 JavaScript 的能力和限制

做一个概述，同时也介绍一些将在其他章节中使用的实例。本章并不是初学者的入门手册，也不是对这种语言的一个高度概括，而是把注意力集中在寻找一些最佳的方法来实现一些基本任务，包括解决让脚本协同工作可能遇到的问题。

第 2 章——使用数字 本章介绍如何在 JavaScript 中使用和处理数字，包括基本的计算、四舍五入、随机数字的生成和控制，以及货币值、顺序值和其他一些格式的数字。

第 3 章——使用字符串 文本是 Web 的精华所在，处理文本几乎是 Web 脚本编程中最常见的任务。这一章涉及如何操作字符串来获取信息、存储数据、输出文本，也包括对 JavaScript 的正则表达式的一个全面的介绍。

第 4 章——使用数组 本章介绍 JavaScript 中用于存储数据的最强大的数据结构的一种：数组。这部分内容讨论如何读写数组中的数据，如何处理数组和对数组排序，以及多维数组的使用，也讨论一个比较相似的数据结构：对象显式声明。

第 5 章——导航文档对象模型 DOM 是操纵文档的各个独立部分的接口。本章将介绍并讨论 DOM，重点在怎样通过元素、属性、文本来创建并读取数据。

第 6 章——处理和验证表单 本章讨论通过各种表单部件来读写数据，完成表单数据的验证和处理，以及讨论如何提高基于表单的接口的可用性。

第 7 章——窗口和 Frame 本章小心翼翼地讨论了窗口操作和跨 Frame 的脚本。这些正是这种语言最具争议性的部分，因为它们可能造成使用上的巨大麻烦而难以被人接受，所以本章严格限制在讨论如何避免和缓解这些问题的技术上。

第 8 章——cookie cookie 是 JavaScript 中最简单也是最值得信赖的状态持久化技术，这种技术允许页面和应用程序“记住”一个人是谁以及他在做什么。本章将介绍 cookie 以及如何有效地使用它。

第 9 章——日期和时间 本章介绍使用 JavaScript 怎样获得时间和日期，怎样比较和处理时间和日期，以及怎样以不同的格式和形式来展现它们。

第 10 章——图片 在 Web 设计中，图片的重要性不言而喻，本章讨论有关图片的一些基本的脚本技术。本章中将给出一些比较简单的例子，例如预读取、随机选取、交换以及图片过渡显示，还有一些较复杂的例子，例如幻灯片效果、进度指示条以及基于图片的时钟。

第 11 章——检测浏览器 本章篇幅较短，介绍一些处理不同浏览器和模式的技术，并解释在什么时候、什么地方、如何适当地使用浏览器检测和对象检测技术，以及怎样将这些技术结合起来，编写出更健壮和更适用的代码。

第 12 章——在 JavaScript 中使用 CSS 这一章关注怎样读写一个元素或一组元素的样式、创建样式表或者从现有的样式表中读写 CSS 规则，以及怎样构建一个样式表转换器。

第 13 章——动态 HTML 基础 动态 HTML (DHTML) 通过 HTML、DOM、CSS 来使得静态的内容变得生动起来，虽然 DHTML 在很多时候被人轻视，我们仍然相信它是与这个脚本语言紧密相关的有用部分。这一章介绍事件处理、对象的位置和尺寸的检测、鼠标追踪、隐藏和显示对象等，也介绍一种通过脚本重新动态排列 DOM 使之整齐的表格化的方法。

第 14 章——计时器和动画 这一章在第 13 章的基础上更加前进了一步，讨论更加复杂的动画部分。在这一章中读者将学习 JavaScript 中的计时器，并使用它来创建一些简单或复杂的动画。这一章还讨论拖曳功能，并使用这一功能对信息进行选择和排序，同时也包括创建滚动条、滑动条以及转换器等。

第 15 章——DHTML 菜单和导航 这一章进入到 DHTML 比较复杂的部分——下拉式菜单和一个文件天树（或者称为可发展式菜单）。对于每种菜单，我们都会用整洁清晰的代码，先创建一个核心的用于导航的数据结构，然后通过改善其可用性和易用性，逐步改进这些代码，包括加入子菜单指示箭头、开关计时器以及自动定位功能（这使得菜单永远不会跑到页面区域之外）。本章也为 Windows IE 6 和 IE 5 中的此类菜单的元素重叠问题提供了解决方法。

第 16 章——JavaScript 和可用性 这一章关心 JavaScript 和可用性的关系，关注用什么样的技术和方法使 JavaScript 代码可以通过键盘驱动，以及使用不同的脚本会如何影响那些有学习或认知方面的困难的用户。在这一章中还将检查一系列的代码，包括 AJAX 程序，看看它们是怎样与用户交互的。

第 17 章——在 JavaScript 中使用 Flash 这一章将介绍这两种技术的结合使用，读者将学习到怎样通过脚本检测用户是否安装过 Flash 插件，并掌握 JavaScript 和 Flash 通信的方法。

第 18 章——使用 JavaScript 构建 Web 应用程序 这一章进入在线程序设计领域中最令人振奋的部分，包括通过 XMLHttpRequest 获取数据，通常同样的功能是通过 iframes 方法实现的。本章也谈到了创建定制对话框、创建可以编辑的元素（例如 RichText）以及控制和创建文本选项以产生自动补全搜索域。

第 19 章——JavaScript 中的面向对象技术 在大项目中，面向对象编程通常被认为是一种最佳的编程方法。这一章将会介绍面向对象编程，讨论它的一些核心概念和优点。涉及的内容还包括创建面向对象的脚本、域、继承以及对对象命名空间。

第 20 章——高级话题 这一章介绍一些很实用的技术，有助于编写更快速有效、更简单、更低开销的代码。这一章还讨论一些用于优化和模糊化我们的产品代码的技术，但是必须声明在先，这种优化往往会增加更多的麻烦，从而降低其使用价值。

本书的网站

本书的网上链接是 <http://www.sitepoint.com/books/jsantl>，这个网址会提供下面一些资料和信息。

（1）代码

在阅读本书的过程中，您会发现本书引用了很多相关代码，这些例子代码都包含在一个 ZIP 压缩包中，可以从 <http://www.sitepoint.com/books/jsantl/code.php> 获取。

（2）更新和勘误

在本书的网上链接中，我们将对勘误表进行随时更新，更正已知的印刷错误和代码错误，而

且对于技术变化也会保持必要的更新。勘误表可以通过 <http://www.sitepoint.com/books/jsantl/errata.php> 访问。

SitePoint 论坛

虽然我们已经尽量猜测您在书中可能遇到的问题，并且尝试做出了解答，但是，在 Web 开发中使用 JavaScript 会遇到各种各样的问题和困扰，而没有任何一本书可以涵盖所有这些问题。如果对于本书有任何问题，最快获得解答的地方将是 <http://www.sitepoint.com/forums/>，这是 SitePoint 颇具人气的知识社区。

致谢

我想感谢在写作本书过程中所有帮助、支持过我的人，特别是 Eddie、Debi、Jon 和 Kim，他们做了很多实际的支持，还给了我很多的鼓励。我也想感谢 Dave Evans——一个在我成长早期对我影响极大的开发者。

James Edwards

目 录

第 1 章 JavaScript 初步	1
1.1 JavaScript 的定义	1
1.2 JavaScript 的限制	2
1.3 JavaScript 的最佳实践	3
1.4 为没有 JavaScript 的用户着想（渐进增强）	4
1.5 内容和行为的隔离（谨慎地编码）	5
1.6 使用括号和分号结束符（一致的编码方式）	7
1.7 给页面添加脚本	9
1.7.1 用HTML注释框住代码	9
1.7.2 语言属性	10
1.8 让不同的脚本在同一页面中协同工作	10
1.9 隐藏 JavaScript 源代码	12
1.10 脚本调试	13
1.10.1 理解浏览器内建的错误报告	14
1.10.2 使用alert	15
1.10.3 使用try-catch	16
1.10.4 向页面和窗口中写入数据	17
1.10.5 使用外部调试器	17
1.11 严格的警告	18
1.12 小结	19
第 2 章 使用数字	20
2.1 JavaScript 中的数学计算	20
2.2 将数字转化为拥有 x 位小数位的形式	22
2.3 创建受约束的随机数	23
2.4 将数字转换为字符串	24
2.5 格式化货币值	25
2.6 将字符串转换为数字	26

2.7	将数字转换为序数	28
2.8	小结	29
第 3 章	使用字符串	30
3.1	在字符串中包含特殊字符	30
3.2	处理字符串中的字符	31
3.3	对 URL 编码	32
3.4	比较字符串	32
3.5	查找字符串中的子串	34
3.6	将字符串拆分成子串	35
3.7	创建正则表达式	36
3.8	测试字符串和正则表达式的匹配	38
3.9	测试字符串是否只包含数字	39
3.10	测试字符串是否是有效的电话号码	40
3.11	测试字符串是否是有效的电子邮件地址	40
3.12	使用正则表达式搜索和替换文本	41
3.13	小结	42
第 4 章	使用数组	43
4.1	使用数组显式声明	43
4.2	创建数组的数组	44
4.3	使用字符串代替数字做数组索引	46
4.4	将数组转换为字符串	47
4.5	增加和删除数组中的成员	48
4.6	将数组按照字母顺序或者数字顺序排序	50
4.7	对多维数组排序	51
4.8	随机地排序一个数组	52
4.9	小结	52
第 5 章	导航文档对象模型	53
5.1	访问元素	54
5.2	创建元素节点和文本节点	58
5.3	改变元素的类型	61
5.4	删除一个元素节点或者文本节点	62
5.5	读写元素属性	63

5.6 获得拥有特定属性值的所有元素	65
5.7 元素的 class 的增减	67
5.8 小结	68
第 6 章 处理和验证表单	69
6.1 读写文本区的数据	69
6.2 读取和设置 checkbox 的状态	71
6.3 读写 radio 按钮的状态	72
6.4 读取和设置 select box	74
6.5 验证强制性的文本区	75
6.6 验证数字	75
6.7 验证电子邮件地址	76
6.8 检查未选中的 radio 按钮	76
6.9 阻止不合法的表单被提交	77
6.10 验证数据项数目不详的表单	78
6.11 验证表单时打印内嵌错误信息	79
6.12 根据其他输入区域的值隐藏和显示某些表单输入区域	80
6.13 小结	83
第 7 章 窗口和 Frame	84
7.1 使用弹出窗口	85
7.1.1 弹出窗口的问题	85
7.1.2 怎样减轻问题的恶劣影响	86
7.2 在新窗口中打开链接	88
7.3 Frame 之间的通信	89
7.4 获取滚动位置	91
7.5 滚动页面到特定位置	93
7.6 获得视区的大小（窗口内可用的空间）	93
7.7 小结	94
第 8 章 cookie	95
8.1 写入 cookie	95
8.2 读取 cookie	96
8.3 设定 cookie 的过期时间	97
8.4 使 cookie 只能通过特定的域和路径访问	98

8.5 突破浏览器对 cookie 数目的限制	99
8.6 小结	100
第 9 章 日期和时间	101
9.1 获得日期和时间	101
9.2 自定义 Date 格式转换	103
9.3 将时间格式化成 12 小时记法和 24 小时记法	105
9.4 比较日期	106
9.5 格式化日期期间的差异	109
9.6 小结	111
第 10 章 图片	112
10.1 图片的预装载	112
10.2 图片的切换	113
10.3 图片的随机显示	115
10.4 多个图片的幻灯片显示效果	117
10.5 图片的淡入淡出效果	119
10.6 基于图片的实时时钟的实现	123
10.7 进度条的实现	126
10.8 小结	128
第 11 章 检测浏览器	129
11.1 识别浏览器是否支持某些特性	129
11.2 识别特殊浏览器	131
11.3 检测 Quirks 模式和标准模式	134
11.4 小结	135
第 12 章 在 JavaScript 中使用 CSS	136
12.1 改变单个元素的样式	136
12.2 改变一组元素的样式	138
12.3 获得元素的计算样式	139
12.4 创建样式表转换器	140
12.5 创建处理多媒体类型的样式表转换器	145
12.6 读取和修改已经存在的样式表	147
12.7 添加新的样式表规则	150

12.8 从样式表中删除一个规则	152
12.9 创建一个新的样式表	152
12.10 小结	155
第 13 章 动态 HTML 基础	156
13.1 事件的处理	156
13.1.1 简便方法: 使用事件处理函数	157
13.1.2 W3C标准方法: 使用事件监听者 (Event Listeners)	159
13.2 获取元素尺寸	168
13.3 获取元素位置	169
13.4 检测鼠标光标的位置	171
13.5 当将鼠标放到某个元素上时显示工具提示	172
13.6 根据表的列对表进行排序	178
13.7 小结	183
第 14 章 计时器和动画	184
14.1 setTimeout 和 setInterval 的使用	184
14.2 沿着某个确定路径移动对象	187
14.3 创建平滑的动画效果	193
14.3.1 动画帧时间	193
14.3.2 两帧之间移动距离的变化	194
14.3.3 动画效果的复杂度	194
14.3.4 计算机的速度	194
14.3.5 浏览器的速度	195
14.4 拖曳行为的实现	195
14.5 利用拖曳功能对列表重新排序	203
14.6 创建新闻滚动条	209
14.7 创建基于裁剪的过渡效果	214
14.8 创建滑动控制器	219
14.9 小结	224
第 15 章 DHTML 菜单和导航	225
15.1 创建下拉式菜单或平展式菜单	227
15.2 添加箭头以显示子菜单的存在	235
15.3 添加计时器延迟菜单的开启和关闭	238

15.4	确保菜单一直位于窗口内	243
15.5	菜单显示在 <code>select</code> 元素之上	249
15.6	创建文件夹树或扩展菜单	255
15.7	在菜单中可扩展分支的指示	263
15.8	任何时候只允许一个菜单分支打开	267
15.9	自动打开当前的子分支	268
15.10	小结	272
第 15 章 JavaScript 和可用性		273
16.1	JavaScript 不具备可用性吗	273
16.1.1	可用性	274
16.1.2	受影响的用户	275
16.2	创建易于用键盘操作的脚本	276
16.3	使用设备无关的事件处理函数	278
16.4	创建键盘和鼠标都能使用的脚本	279
16.4.1	Rollovers 与内容的显示	280
16.4.2	表单验证	282
16.4.3	拖放 (Drag-and-drop) 功能	283
16.4.4	AJAX 和其他远程脚本技术	284
16.5	创建显示在焦点上的 <code>title</code> 属性工具提示	285
16.6	创建可用键盘访问的 DHTML 菜单	291
16.7	创建通过键盘可用的 DHTML 菜单	299
16.8	创建可用键盘控制的 DHTML 滑动控件	305
16.9	创建屏幕读取器可访问的脚本	311
16.9.1	JavaScript 行为	313
16.9.2	技巧和深入研究	321
16.9.3	寻求最佳实践	323
16.10	小结	325
第 17 章 在 JavaScript 中使用 Flash		326
17.1	检测浏览器是否安装了 Flash	326
17.2	JavaScript 和 Flash 之间的沟通	329
17.2.1	FSCommand	330
17.2.2	Flash/JavaScript 集成工具包	331
17.3	小结	332

第 18 章 使用 JavaScript 构建 Web 应用程序	333
18.1 使用 XMLHttpRequest 获取数据	333
18.1.1 请求服务器上的数据	336
18.1.2 解析数据	337
18.1.3 缓存	338
18.1.4 AJAX 框架	339
18.2 不使用 XMLHttpRequest 获取数据	339
18.3 创建自定义对话框	343
18.4 创建可编辑元素	349
18.5 控制文本选择	354
18.6 创建自动完成的文本域	358
18.7 小结	368
第 19 章 JavaScript 中的面向对象技术	369
19.1 面向对象技术的优点	369
19.1.1 抽象性	370
19.1.2 封装性	370
19.1.3 类的继承	370
19.1.4 多态	371
19.2 基于对象的代码与面向对象的代码	372
19.3 创建一个面向对象的脚本	372
19.4 为对象创建方法	373
19.5 模拟继承	377
19.6 理解作用域 (Scope)	379
19.7 命名空间的实现	381
19.8 小结	382
第 20 章 高级话题	383
20.1 使脚本运行得更快	383
20.1.1 减少常用对象的引用	384
20.1.2 使用三元操作符以及 switch 语句	386
20.1.3 优化循环	388
20.1.4 避免使用 eval	389
20.1.5 避免严格警告的出现	390

20.1.6 为特定浏览器进行优化	390
20.2 减小脚本的代码量	393
20.2.1 将任务划分为函数（使用面向对象思想）	393
20.2.2 使用数组和循环以避免代码的重复	394
20.2.3 使用简洁的条件和返回语句	395
20.3 为 Web 优化脚本	396
20.3.1 去掉注释和不必要的空白符	396
20.3.2 使变量名和属性名简洁化	398
20.4 避免内存泄漏	399
20.4.1 避免循环引用	399
20.4.2 在使用之后进行清理	400
20.5 使脚本在 Load 事件之前运行	402
20.6 小结	404

第 1 章

JavaScript 初步

JavaScript 的确是一种令人惊奇的编程语言，有很多独特的优势，本书将主要介绍 JavaScript 的实际应用。只要花一点儿时间学习和思考，就可以使用 JavaScript 为网站带来更多的功能、更新颖的设计和更好的易用性。

下面先介绍 JavaScript 的概念，然后讨论它的由来以及使用方法。

1.1 JavaScript 的定义

JavaScript 是一种用于增强网页与应用程序的交互能力和动态能力的脚本语言。JavaScript 可以和网页的各种部件进行交互，例如 HTML、CSS，并且可以使它们实时地变化，以响应用户的操作。

毫无疑问，您一定看过网页中的 JavaScript 代码，大多是嵌入 HTML 元素中的，例如：

```
<a href="page.html" onclick="open('page.html'); return false;">
```

也可能是作为一个 script 元素，链接到其他文件中的：

```
<script type="text/javascript" src="myscript.js"></script>
```

还可能直接在文件中编码：

```
<script type="text/javascript">
function saySomething(message)
{
    alert(message);
}
saySomething('Hello world!');
```