

The  
Pragmatic  
Programmers

**Broadview**<sup>®</sup>  
www.broadview.com.cn

# Prototype 与 script.aculo.us 终极揭秘

Prototype and script.aculo.us  
*You Never Knew JavaScript Could Do This!*



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

[法] Christophe Porteneuve 著  
陆开一 译

# Prototype与script.aculo.us 终极揭秘

Prototype and script.aculo.us: You Never Knew JavaScript Could Do This!

[法] Christophe Porteneuve 著

陆开一 译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

Prototype 与 script.aculo.us 库能抹平不同浏览器之间的沟壑,使得一些常见的功能更加容易实现,通过本书你就能迅速掌握这些非常棒的库。深入研究 Prototype 后你将发现,Prototype 库居然能使 JavaScript 变得如此强大,使它看上去更像 Ruby。在 Prototype 中研究 DOM 和事件处理、征服 Ajax,将大大简化你的代码,使一切变得更加简单,而且可移植性更强。当谈及 UI 的高级特性时,script.aculo.us 使得 Web 开发者们的梦想变成现实:创建自动的文本输入用来 In-Place 编辑、提供可定制的拖曳行为、关注用户的需求,这些都只需要简单的代码而已。

本书适合于有一定用 JavaScript 进行 Web 开发经验的中级读者,通过对 script.aculo.us 的学习和研究,能使自己少走很多弯路;本书更加适合于有丰富的 JavaScript 开发经验的 Web 高级开发者,他们对 script.aculo.us 库的需求更迫切些。高级读者还能通过本书分析哲学库的设计原理,然后对其进行改进,使之为自己所用。

978-1-934356-01-2: Prototype and script.aculo.us: You Never Knew JavaScript Could Do This! by Christophe Porteneuve.

All rights reserved. Authorized translation from the English language edition published by The Pragmatic Programmers, LLC.

本书简体中文专有翻译出版权由 The Pragmatic Programmers, LLC. 授予电子工业出版社。未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字: 01-2008-2464

### 图书在版编目(CIP)数据

Prototype 与 script.aculo.us 终极揭秘 / (法)波顿纽威 (Porteneuve,C.) 著; 陆开一译. —北京: 电子工业出版社, 2008.10

书名原文: Prototype and script.aculo.us: You Never Knew JavaScript Could Do This!

ISBN 978-7-121-07309-0

I. P… II. ①波…②陆… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 134327 号

责任编辑: 王继花

印 刷: 北京智力达印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 28.25 字数: 522 千字

印 次: 2008 年 10 月第 1 次印刷

定 价: 65.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。服务热线: (010) 88258888。

## 读者评论

### What readers are saying about *Prototype* and *script.aculo.us*

我一直在工作中使用 Prototype 和 script.aculo.us，但是通过本书我学到了很多。我常常回来翻阅它以查找一些在其他地方找不到的内容。这是一本由很有经验的老师编写的对编程人员很有帮助的书，它会对所有希望能够轻易地为他们的应用添加 JavaScript 特性的读者大有帮助。

► Stéphane Akkaoui

Ruby on Rails 开发者，Feedback 2.0

如果您（或您的小组）打算学习 JavaScript 框架，这本书将是一位指引您毫不费力地一步步通往 Prototype 和 script.aculo.us 的向导。它以最简洁的风格编写从基础的到高级的代码。您会惊讶地发现 JavaScript 可以做得如此完美。

► Arnaud Berthomier

Web 开发者，Weborama

这本书是所有的 Prototype 和 script.aculo.us 开发者都应该拥有的。它不只是一本参考书；您会找到您想知道的关于这两个框架的所有东西，您会学到很好的 JavaScript 习惯。我已经学到了很多 script.aculo.us 技巧，并且掌握了很多 Prototype 方法，它们使得我的代码在提高功效的同时也更加简洁、更具可读性。感谢这本书！

► Sébastien Gruhier

Xilinus 创始人和 CTO

厌倦了页面加载一次又一次的等待了么？如果您像我一样，正在寻找一个智能的、优雅的方法来为您的应用注入 Ajax 的话。那好，您会在这本书中发现了解 Prototype 和 script.aculo.us 所需的一切。这本书在助您掌握最佳实践的同时也不失趣味性！

► Amir Jaballah

Fastconnect 技术组长

我们为所有的 Web 项目使用 Prototype 和 Script.aculo.us。当我们培训其他开发者的时候，总会提到两个问题：从哪里可以得到更多关于库的信息？从哪里可以学习现代的、函数风格的 JavaScript 编程？

本书同时回答了这两个问题。

Christophe 演示了这些库以及惯用的 JavaScript 风格的力与美。他不流于表面，他的介绍显示了高级的 JavaScript 用法，甚至比有些整本介绍 JavaScript 的书还要深入。虽然已使用了多年的 Prototype 和 script.aculo.us，我还是从每一章中学到了新的东西。感谢 Christophe!

▶ Stuart Halloway  
Relevance, Inc. 的 CEO  
[www.thinkrelevance.com](http://www.thinkrelevance.com)

To Élodie, my love, ever supportive.

You're my true home.

## 推荐序

2005 年对于 Web 开发来说是一个伟大的年份。在这一年中，有两项技术异军突起，一项是 Ajax，另一项是 Ruby on Rails。这两项技术的出现改变了 Web 开发的面貌，甚至打乱了 JavaEE 前进的步伐。多年以来，JavaEE 设计者们为自己所设计的无所不包的复杂架构而陶醉，新的 buzz word 层出不穷，一出来就会得到广泛的关注，相关的图书也会热卖。辉煌的 JavaEE 版图中居然还有完全被忽略的死角，这是 JavaEE 设计者们始料不及的。事实上他们在 JavaEE 的架构中从来就没有考虑过浏览器端的处理能力，浏览器对于 JavaEE 来说是完全无智能的瘦客户端。他们这样设计情有可原，因为他们只能把设计局限在 Java 能够控制的区域内，在 Java Applet 彻底失败之后，Java 能够控制的区域退守到了服务器端。毫无疑问，JavaEE 取得了巨大的成功，但是这种完全基于服务器端的解决方案很难向用户提供优秀的交互体验，同时也很难实现最佳的可伸缩性（Scalability）。

Ajax 是用户和市场的选择，这个由群众创造出来的 buzz word 时常响在 JavaEE 设计者的耳边，令他们非常烦躁。甚至有人喊出了“JavaEE without Ajax”的口号，和 2004 年 Rod Johnson 所提出的“J2EE without EJB”相对应。但是仔细考察这个“without Ajax”，我们却发现它和“without EJB”说的并不是一个意思。“without EJB”的意思就是不使用 EJB，而“without Ajax”的意思却不是不使用 Ajax，而是设法将 Ajax 隐藏掉，使得普通的 JavaEE 开发者不需要学习 Ajax。在这个口号的发明者所设计的 JSF 框架中，内嵌了一个 Ajax 组件库 ExtJS，看来他们还是需要 Ajax 的帮助的。所谓的“JavaEE without Ajax”其实是一个伪命题，只是用来吸引眼球的噱头，主要的目的是为了迎合那些讨厌 Ajax、对于浏览器端脚本编程没有兴趣的 JavaEE 开发者。其实无论开发 Web 应用，还是开发 B/S 结构的企业应用，交互体

验都是非常重要的。在交互体验越来越受到重视的今天，想要把 Ajax 排除在外，可以说是一件不可能的任务。

尽管 Ajax 开发确实很重要，但无须讳言的是 Ajax 开发，特别是 DHTML 开发至今仍然是一件很复杂的工作，必须具备专业的技能。除了高超的开发技能之外，还需要开发者对于 Web 可用性和交互设计具有足够的理解。这超出了 Web 页面设计师和制作人员的能力范围；而对于服务器端开发者来说，这些显然也不是他们的特长（他们的特长是处理业务逻辑和与数据库打交道，而不是界面开发和交互设计）。在 Web 页面设计师和服务器端开发者之间出现了巨大的断层，这使得高水平的 Ajax 开发者一下子变得炙手可热。

JavaEE 设计者们简化 Ajax 开发的努力是可以理解的。但是我们需要讨论清楚一个问题：究竟怎样做才能真正简化 Ajax 开发？

在我看来，JavaEE 社区简化 Ajax 开发的努力都不是很成功。将 ExtJS 嵌入到 JSF 框架中，确实可以实现一些简单的交互需求，但是这种做法其实阉割掉了 Ajax 的主要能力——直接在浏览器端处理用户的事件。JSF 的事件模型是完全位于服务器端的，界面的任何改变都需要发送事件到服务器端做处理。即使引入了 ExtJS，也不可能改变这一点。在 Struts2/WebWork 和 jMaki 中通过使用 taglib 集成 Ajax 组件库的做法也会令人感觉笨拙而不自然。之所以集成的效果不理想，是因为传统的 JavaEE 框架和 Ajax 组件库的核心架构设计都没有考虑到对方的需求，它们在架构设计上存在着内在的冲突，拉郎配的结果是可想而知的。这些 JavaEE 表现层框架集成 Ajax 组件库，是希望将 Ajax 组件库改造为 JavaEE 世界的顺民，服从 JavaEE 框架的架构约束，但是这样做不可避免地会削弱 Ajax 组件库的能力。要简化 Ajax 开发并且充分用好 Ajax 技术，仅仅使用 taglib 来对 Ajax 组件库做封装是不够的，服务器端的架构也必须加以改造，甚至需要设计出一套全新的架构，这是传统的 JavaEE 表现层框架无法做到的。Google 的 GWT 同时具有浏览器端和服务器端，似乎是一个很好的选择，但是 GWT 主要是为实现一类 One Page 的 Ajax 应用而设计的，它模仿的是桌面应用的交互模型，而绝大多数的 Web 应用是不可能采用这种交互模型的。DWR 也是一个跨浏览器端和服务器端的框架，但是 DWR 的 RPC 风格的 API 是我不喜欢的。RPC 架构和 REST 架构在可伸缩性方面的差距是巨大的，Ajax 应用的最佳架构是 REST 而不是 RPC，同时 REST 在简化编程模型方面的效果也要比 RPC 更好。

要简化 Ajax 开发, 我认为需要满足以下 3 个要求:

1. 由专业的 DHTML 开发者开发出更加全面和强大的 Widget 库。这部分的开发应该由专业人士来做, 这样就可以把负担从普通的 Web 开发者身上卸掉。开源软件是开发 Widget 库的必由之路。
2. Ajax 组件库要与服务器端框架做更加紧密的集成。服务器端框架的架构需要加以改造以适应 Ajax 和 REST 的需求, 不能指望以不变应万变。
3. 简化的编程模型。这包括两个方面: a) 通过与服务器端框架紧密集成, 使得服务器端开发者仅仅使用服务器端编程语言就能够实现一些普通的 Ajax 需求; b) 更加复杂的交互需求如果必须要做 DHTML 开发, 需要大幅降低 DHTML 开发的难度。

当我把眼光投向了 Ruby on Rails 之后, 包括 Ajax 开发在内, 一切都变得容易了很多。Rails 通过其 RJS 模板集成了两个 Ajax 组件库 Prototype 和基于 Prototype 的 script.aculo.us。Rails 集成这两个组件库的方式显得非常自然, 因为这两个组件库正是 Rails 和 Ajax 相结合的产物, 它们都是由 Rails 的核心开发人员所开发的, 充分考虑到 Rails 在架构方面的需求。现在这两个组件库已经长大成人, 完全可以独立应用在非 Rails 环境中。Rails + Prototype/script.aculo.us 满足了上述 3 个要求, 因此是目前做 Ajax 开发的最佳组合。当然, 上述 3 个要求每一个都还有很大的改进余地, 但是 Rails + Prototype/script.aculo.us 走在了正确的道路上, 前景是非常光明的。

随着 Ajax 技术的普及, 传统的 DHTML 组件库恢复了生机, 可以加以改造以适应新的交互需求。这些 DHTML 组件获得了一个时髦的新名称——Widget。我一般是将 Ajax 组件库分成 3 部分:

1. 基础库。包括: a) 对于 JavaScript 语言的增强, 为目前浏览器所支持的 JavaScript 1.x 版提供继承、包管理等面向对象编程所需要的支持; b) 封装 HTML DOM 的 API, 提供编程模型更加简化的 API。
2. Web Remoting 库, 通过 XMLHttpRequest/IFrame/JSONP 等机制与服务器交互, 从服务器获取数据。
3. 丰富的 Widget 库, 可用来实现各种交互需求, 以改善用户的交互体验。

Prototype 库实现了第 1 部分和第 2 部分, script.aculo.us 库实现了第 3 部分。在这 3 部分中, 开发工作量最大的是 Widget 库。正是因为市场上对于 Widget 库有着巨大的需求, 在 2007 年涌现出了大量 Widget 库, 开源的包括 script.aculo.us、Dojo、YUI、ExtJS、Mootools、jQuery UI、Spry、Tibco GI、Qooxdoo; 私有的包括 ASP.NET

Ajax、Dorado、Zimbra，等等，全部列举下来不下二三十个。Apple 公司在这一年推出了 iPhone 手机，其中的 Widget 全部是使用 DHTML 开发的，达到了交互体验的极致，同时也展示出 Widget 开发的美好前景。很多人说 2007 年是 Widget 年，这个说法是有道理的。到了 2008 年，Google 公司推出了 Open Social，其核心是 Google 所提供的 Widget 开发 API 和 Widget 的部署容器。Google 管这些 Widget 叫做 Google Gadget，Gadget 将 Widget 的开发水平推向了新的高度。

无论是基于现有的 Widget 开发 Ajax 功能，还是开发新的 Widget，Prototype/script.aculo.us 都是我们手中的一件利器。网上的很多投票都显示出，Prototype/script.aculo.us 组合是最受欢迎的 Ajax 组件库。因为 Prototype 大受欢迎，其爱好者基于 Prototype 还开发了很多其他的组件库，Prototype 家族成员的数量还在不断增加。很多 Ajax 组件库最大的问题是缺乏详尽的文档，API 参考并不能代替文档。从文档的数量和质量来说，Prototype/script.aculo.us 的文档在所有 Ajax 组件库中都是最棒的。本书的出版，为爱好者深入学习这两个组件库提供了极大的便利。本书的副标题是“*You Never Know JavaScript Could Do This!*”，确实，正是通过这些优秀的 Ajax 组件库、这些勇于探索的 Ajax 顶尖高手们，我们才知道浏览器端脚本编程的世界原来是如此灿烂。基于 Prototype/script.aculo.us 做 Ajax 开发是一种很愉快的体验。学习越深入，这种愉快的感觉就越强烈。那么，你还在等什么呢？

▶ 李锟

上海印客网首席架构师

2008 年 7 月 6 日于上海

## 译者序

很多程序员比较注重服务端开发技术，而对于前台的 JavaScript 并不是很在意。随着网络技术的飞速发展，JavaScript 也在不断地得到改进和深入使用。

Prototype 更使得 JavaScript 如虎添翼。它提供了一整套“面向对象”的机制，虽然这些都是通过使用原型等技巧来模仿实现的，但是您现在也可以“面向对象”了。而且它还提供了很多功能函数，使一些常用操作更方便了。

Prototype 还对很多原生对象、DOM 操作、定时器，以及 XMLHttpRequest 等进行了封装和扩展。此外，它还引入了 Enumerable、Template 等很多实用的模块和对象。同时大大地提高了代码的简洁度和效率。

而 script.aculo.us 为我们提供了美妙的 UI 感受，以前很多 Web 程序员常常需要自己写一些效果，纵然效果写得很好，但也花费了不少的时间和精力。如果自己写了很多效果，能否把类似的效果归纳、组织一下，以便以后少写一些重复代码呢？script.aculo.us 就为我们做了这些。

自动完成、效果、拖曳，等等，一切都妙不可言。

作者的例子很好也很具有代表性，我们可以自己在网站上借鉴它们。本书源代码下载的网址为：<http://books.pragprog.com/titles/cpsu/prototype-and-script-aculo-us>。

由于译者水平有限，难免有疏漏和不妥之处，还望广大读者多多指正。

非常感谢博文视点的编辑晓菲的大力帮助。

► 陆开一

2008 年 6 月于上海

## 联系博文视点

您可以通过如下方式与本书的出版方取得联系。

读者信箱: [reader@broadview.com.cn](mailto:reader@broadview.com.cn)

投稿邮箱: [bvtougao@gmail.com](mailto:bvtougao@gmail.com)

北京博文视点资讯有限公司(武汉分部)

湖北省 武汉市 洪山区 吴家湾 邮科院路特1号 湖北信息产业科技大厦1402室

邮政编码: 430074

电 话: 027-87690813

传 真: 027-87690595

若您希望参加博文视点的有奖读者调查, 或对写作和翻译感兴趣, 欢迎您访问:

<http://bv.csdn.net>

关于本书的勘误、资源下载及博文视点的最新书讯, 欢迎您访问博文视点官方博客:

<http://blog.csdn.net/bvbook>

# 序

## Preface

Prototype 开始于 2005 年初，那个时候在多数开发者的印象中“JavaScript”依然是弹出广告、闪烁的文本、<script>标签的拷贝复制。即使一些 Web 应用程序，比如 Gmail 以及 Google Suggest 等，向世界显示了 JavaScript（以及新的被叫做“Ajax”的事物）实际上也可以用来改善用户的体验，但是在开发者自己的应用程序中使用这些新的技巧却是令人痛苦和沮丧的。每一种 Web 浏览器都有自己的一些怪僻，多数现存的代码在设计的时候没有利用 JavaScript 面向对象的天性或者强大的闭包能力。

受到来自动态语言，比如 Ruby，表达性的灵感，我们着手创建可以期待使用的浏览器编程环境。我们从一小组允许我们和类以及函数共事的工具开始。然后我们从现存的应用程序中萃取常见的 Ajax 和 DOM 处理操作。在 2005 年 3 月，我们发布了作为 Ruby on Rails 框架一部分的 Prototype 1.0。从那以后，Prototype 成长了很多，但是它依然关注于为 JavaScript 开发者提供最好的开发环境。

至于 script.aculo.us，核心团队称之为“Scripty”，它开始是作为 Prototype 中的一个很小的部分，实现了现在流行的“黄褪技术”。为了使 Web 应用程序更加地用户友好，视觉愉悦，script.aculo.us 很有必要进行改善，因此它很快成长为一个完整的实时的、基于 DOM 的效果引擎、拖放框架和控件库。它的 1.0 版本在 2005 年 6 月发布。

您应该理解 script.aculo.us 和很多 UI 不同，因为它并不是设法把开发者和 DOM 隔开，而是扩展和改进 DOM，这样开发者和设计者可以利用他们现有的知识。

通过和 Prototype 结合, 它被设计为用来构建自己的窗体部件 (widget)、控件 (control), 以及其他很棒的东西, 而所用的时间要比配置重量级的、基于窗体部件的框架少得多。

Ruby 的设计深深地影响了这两个库, 作为 Ruby 格言 (Ruby——A Programmer's Best Friend) 的诠释, Prototype 和 script.aculo.us 是 “Web 程序员最好的朋友”。根据得到的反馈, 我们不是唯一深有此感的人。

最初版本发行两年半之后, Prototype 和 script.aculo.us 应用在很多最流行的网站上并且为各种创新的应用提供了动力。

它之所以能快速地吸收流行的事物是因为 Prototype 核心组成员的努力, Prototype 和 script.aculo.us 社团的数百贡献者数千小时的工作; 还有本书的作者 Christophe。Prototype 核心团队成員有 Seth Dillingham、Andrew Dupont、Mislav Marohnić、Justin Palmer、Christophe Porteneuve、Tobie Langel、Scott Raymond 以及 Dan Webb;

大力感谢他们和你们!

▶ Sam Stephenson (Creator of Prototype)

2007 年 10 月 15 日

▶ Thomas Fuchs (Creator of script.aculo.us)

2007 年 10 月 15 日

# 目 录

## Contents

|                                                |    |
|------------------------------------------------|----|
| 序.....                                         | 1  |
| 第 1 章 引言.....                                  | 1  |
| 1.1 关于时间.....                                  | 1  |
| 1.2 本书的内容及组织结构.....                            | 3  |
| 1.3 致谢.....                                    | 4  |
| 第 I 部分 Prototype.....                          | 7  |
| 第 2 章 探究 Prototype.....                        | 9  |
| 2.1 到底什么是 Prototype.....                       | 9  |
| 2.2 如何在项目中使用 Prototype.....                    | 10 |
| 2.3 在使用 Prototype 时, JavaScript 看起来会是什么样子..... | 10 |
| 2.4 Prototype 术语和概念.....                       | 25 |
| 2.5 那么 Prototypes 究竟是什么.....                   | 27 |
| 2.6 运行本书中的 Prototype 代码实例.....                 | 29 |
| 第 3 章 带有\$的快速帮助.....                           | 31 |
| 3.1 快捷方式应该简短.....                              | 31 |
| 3.2 使用\$快速获取巧妙的元素.....                         | 31 |
| 3.3 \$w, 因为数组直接量很烦人.....                       | 33 |
| 3.4 \$\$, 使用样式进行查找.....                        | 34 |
| 3.5 \$A, 集合统一器.....                            | 36 |
| 3.6 \$F, 表单域专家.....                            | 38 |
| 3.7 \$H, 生成事物的 Hash.....                       | 39 |
| 3.8 使用\$R 处理范围.....                            | 39 |

|                                        |     |
|----------------------------------------|-----|
| 第4章 常规的 JavaScript 打了激素 .....          | 41  |
| 4.1 普通对象的操作 .....                      | 41  |
| 4.2 正确的函数绑定 .....                      | 47  |
| 4.3 您的函数实际上知道更多技巧 .....                | 51  |
| 4.4 数值 .....                           | 54  |
| 4.5 字符串 .....                          | 57  |
| 4.6 数组 .....                           | 64  |
| 4.7 全能的 JSON 支持 .....                  | 70  |
| 第5章 具有 Enumerable 的高级集合 .....          | 77  |
| 5.1 核心方法: 使用 each 遍历 .....             | 77  |
| 5.2 获得集合的综合信息 .....                    | 80  |
| 5.3 查找元素, 提供过滤 .....                   | 82  |
| 5.4 对元素进行分组, 把集合拼接起来 .....             | 84  |
| 5.5 计算衍生的集合和值 .....                    | 88  |
| 5.6 现在排序: 获取极值, 使用定制的调整 .....          | 91  |
| 5.7 把集合转换为数组或调试字符串 .....               | 93  |
| 5.8 Enumerable 实际上是个模块 .....           | 93  |
| 第6章 统一的事件处理 .....                      | 97  |
| 6.1 事件 .....                           | 97  |
| 6.2 事件名人堂 .....                        | 114 |
| 6.3 响应表单相关的内容变化 .....                  | 116 |
| 第7章 和 DOM 打交道最终是个乐趣 .....              | 119 |
| 7.1 扩展 DOM 元素 .....                    | 119 |
| 7.2 Element, 您最好的新朋友 .....             | 121 |
| 7.3 选择器 .....                          | 156 |
| 7.4 调试 DOM 相关的代码 .....                 | 157 |
| 第8章 表单管理 .....                         | 163 |
| 8.1 为了更好的用户界面 .....                    | 163 |
| 8.2 查找表单域 .....                        | 167 |
| 8.3 通过 Ajax 提交表单 .....                 | 168 |
| 8.4 密切监视表单和域 .....                     | 173 |
| 第9章 Ajax 从来没有这么简单过 .....               | 177 |
| 9.1 在我们开始之前 .....                      | 177 |
| 9.2 上路了: Ajax.Request .....            | 183 |
| 9.3 流水作业: Ajax.Updater .....           | 198 |
| 9.4 轮循检测: Ajax.PeriodicalUpdater ..... | 205 |

|                                       |            |
|---------------------------------------|------------|
| 9.5 监测 Ajax 活动: Ajax.Responders.....  | 209        |
| 9.6 调试 Ajax.....                      | 209        |
| 9.7 Ajax 有危险么? 可访问性及效率的考虑.....        | 210        |
| <b>第 10 章 更多有用的帮助对象.....</b>          | <b>215</b> |
| 10.1 把值存放到 Hash 中.....                | 215        |
| 10.2 表达任何事物的范围.....                   | 220        |
| 10.3 无重入风险的周期性执行.....                 | 221        |
| 10.4 用模板制造很简单.....                    | 223        |
| 10.5 检查当前的浏览器和 Prototype 库.....       | 226        |
| <b>第 11 章 性能考虑.....</b>               | <b>231</b> |
| 11.1 元素扩展和\$函数.....                   | 231        |
| 11.2 迭代 vs. 普通循环.....                 | 232        |
| 11.3 废弃的事件处理器.....                    | 233        |
| 11.4 您需要知道的新的速度提升.....                | 233        |
| 11.5 简练就是漂亮.....                      | 234        |
| <b>第 12 章 结尾.....</b>                 | <b>235</b> |
| 12.1 构建一个奇特的任务列表.....                 | 235        |
| 12.2 构建基础工作.....                      | 236        |
| 12.3 它只需要 40 行 JavaScript 代码.....     | 238        |
| <b>第 II 部分 script.aculo.us.....</b>   | <b>243</b> |
| <b>第 13 章 探究 script.aculo.us.....</b> | <b>245</b> |
| 13.1 script.aculo.us 的模块.....         | 245        |
| 13.2 在页面中使用 script.aculo.us.....      | 247        |
| <b>第 14 章 视觉效果.....</b>               | <b>249</b> |
| 14.1 那些效果是什么, 为什么要使用它们.....           | 249        |
| 14.2 核心效果.....                        | 251        |
| 14.3 深入效果.....                        | 257        |
| 14.4 组合效果.....                        | 261        |
| 14.5 释放酷因子: 效果队列.....                 | 264        |
| 14.6 效果助手.....                        | 266        |
| 14.7 如何创建我们自己的效果.....                 | 268        |