



Professional JavaScript Frameworks: Prototype, YUI,
Ext JS, Dojo and MooTools

JavaScript

框架高级编程

——应用Prototy

Dojo、MooTools

(美) Leslie M. Orchard 等著
Ara Pehlivanian 译
杨明军



清华大学出版社

JavaScript 框架高级编程

——应用 Prototype、YUI、Ext JS、Dojo、MooTools

(美) Leslie M. Orchard 等著
Ara Pehlivanian
杨明军 译

清华大学出版社

北 京

Leslie M. Orchard, Ara Pehlivanian, et al.

Professional JavaScript Frameworks: Prototype, YUI, Ext JS, Dojo and MooTools

EISBN: 978-0-470-38459-6

Copyright © 2009 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2009-7475

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

JavaScript 框架高级编程——应用 Prototype、YUI、Ext JS、Dojo、MooTools/(美)欧查德(Orchard, L.M.), (美)佩里瓦尼安(Pehlivanian, A.) 著; 杨明军 译. —北京: 清华大学出版社, 2011.2

书名原文: Professional JavaScript Frameworks: Prototype, YUI, Ext JS, Dojo and MooTools

ISBN 978-7-302-24783-8

I. J… II. ①欧… ②佩… ③杨… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2011)第 012845 号

责任编辑: 王 军 张立浩

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 何 芊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 49 字 数: 1317 千字

版 次: 2011 年 2 月第 1 版 印 次: 2011 年 2 月第 1 次印刷

印 数: 1~3000

定 价: 98.00 元

产品编号: 031836-01

译者序

JavaScript 是一种由 Netscape 的 LiveScript 发展而来的、基于原型继承的、面向对象的、动态类型的客户端脚本语言，旨在为用户提供更流畅的 Web 体验。JavaScript 是一门具有非常丰富功能的语言，它有着与其他编程语言一样的复杂性，甚至更加复杂。实际上，要想采用 JavaScript 编写比较复杂的程序，必须对 JavaScript 语言有扎实的理解才行。JavaScript 发明人 Brendan Eich 誉为“JavaScript 的精神领袖”的 Douglas Crockford 曾经撰文宣称“JavaScript 是世界上最被误解的语言”(<http://javascript.crockford.com/zh/javascript.html>)。

但是，AJAX 给了 JavaScript 复兴的机会。随着 Web 2.0/AJAX 技术的成功推广，JavaScript 也得到了发展的绝佳机会。为了解决 JavaScript/DOM 跨浏览器问题、代码重用甚至在设计取向方面存在的问题，各种 JavaScript 框架相继涌现，力争解决这些问题。JavaScript 语言设计的灵活性为 JavaScript 框架的设计和实现带来了无限的可能性，而且这些框架身处 Web 开发热潮，因此快速进化中，这导致我们很难全面了解这些框架。本书精选了 5 种最流行的 JavaScript 框架，旨在让读者了解这些框架的设计思想并掌握它们的应用技巧。

本书通过大量的实用示例和详细的论述，演示了 5 种主流框架解决主要的 Web 开发问题的方式以及它们背后的设计哲学。相信通过这 5 种框架的学习和对比，读者能够对 JavaScript 编程有深刻的理解，并能够在自己的 Web 项目中利用这些框架来简化编程，提高工作效率。

本书由杨明军翻译。Be Flying 工作室负责人肖国尊负责本书译员的选定、翻译质量和进度的控制与管理。敬请广大读者提供反馈意见，读者可以将意见发到 wkservice@vip.163.com，我们会仔细阅读读者发来的每一封邮件，以求进一步提高今后译著的质量。

作者简介

Leslie Michael Orchard 是一位来自底特律地区的作家和连续剧爱好者。他和心爱的妻子饲养了两只奥斯豹斑猫和一对袖珍兔子。忙里偷闲，他会在名为 **0xDECAFBAD** 的网站上 (<http://decafbad.com>) 分享一些代码、文档以及其他有趣的内容。

Ara Pehlivanian 自从 1997 年就一直从事 Web 相关工作。他曾经做过自由职业者、网站管理员，最近则成为 **Nurun** (一家全球交互式通信机构) 的前端架构师和实践引领者。**Ara** 的经历来自于他曾经在自己的职业生涯中做过 Web 开发方方面面的工作，而现在则热衷于基于 Web 标准的前端开发。除了在工作上教授有关最佳实践的知识以及编写代码之外，他还维护自己的个人网站 <http://arapehlivanian.com/>。

Scott Koon 已经从事专业软件开发超过 13 年时间。他将大学期间的主要时间花在了研究一种遗留的、没有文档的符号语言(DNA)上，并最终取得了生物化学学位。他在 JavaScript 还被称为 LiveScript 时就已经在编写 JavaScript 代码，当时 Netscape 浏览器的右上角还有一个非常大的紫色字母“N”。他在 <http://lazycoder.com> 上面维护了一个博客，而且已经活跃在博客社区中将近 10 年时间。他经常光顾 Twitter (<http://www.twitter.com/lazycoder>)，并且为一个名为 Witty 的 WPF .NET Twitter 客户端做出贡献。他还与其他人共同创办了一个名为 **Herding Code** 的播客 (<http://www.herdingcode.com>)，用来处理各种技术方面的问题。他和妻子以及两个女儿一起住在华盛顿州西雅图市。

Harley Jones 目前是 **Perficient, Inc** (NASDAQ: PRFT) 的首席技术咨询师。他毕业于乔治亚州亚特兰大市的奥格尔绍普大学，获得英语文学学位。但是，他自从 10 岁起就一直在进行编程。他已经从事专业软件开发超过 10 年时间，酷爱研究编程语言。他积极地支持并教授几乎每个认真学习现代编程技术的人员。在不编写程序时，他会教育自己的孩子成为科学家。可以通过 harley.333@gmail.com 联系 Harley。

前 言

JavaScript 是一种用于 Web 应用程序的行业标准客户端脚本编程语言。本书研究了一些可用的顶级 JavaScript(JS)框架，提供了大量实用的示例和讲解来说明每个框架最擅长的领域。

在过去几年中，JavaScript 语言出现了复兴趋势。各种各样的项目如雨后春笋般出现，它们致力于构建可重用的 JS 库和框架，而在这方面，它们中有许多已经成熟并显示出持久力，值得在专业项目中认真考虑并依靠它们。

JavaScript 在 Web 的发展中得以流行，现在所有主要浏览器和新的 Web 技术都支持它。随着时间的推移，JavaScript 已经得到扩展，通过使用 Adobe Flash、AJAX 和 Microsoft Silverlight 等技术，可以交付高性能的、给人留下深刻印象的 Web 用户体验。

随着 JavaScript 越来越多地用于“正式的”Web 开发，开发人员将在这个过程中汲取的教训和发明的工具以库和框架的形式整理出来并进行分享。但由于 JavaScript 是一种如此灵活和动态的语言，每种框架都采用非常不同的方式来解决 Web 开发中的各种问题，每种方式都有各自的优缺点。

本书目标读者

本书面向热衷于学习 JavaScript 并研究各种 JavaScript 框架提供了哪些工具以及它们如何提高工作效率的 Web 开发人员。本书要求读者了解基本的 HTML、CSS 和 JavaScript 应用知识，而且具备一定的面向对象编程、服务器端 PHP 脚本编程经验，并了解诸如 AJAX 之类的现代 Web 开发技术。

本书涵盖内容

本书旨在成为一本在工作和查询其他联机资料来源时的简洁的、方便的手册，因此没有详细讲解有关 JavaScript 框架的更高级的、实验性的、尚在开发的方面。

使用本书的前提条件

一款浏览器、一个文本编辑器以及 Web 托管环境基本上就是使用本书示例所需的全部条件。强烈推荐使用了 Firebug 扩展的 Mozilla Firefox 作为浏览器，因为这个组合提供了非常强大的带有 JavaScript 日志和 DOM 探索工具的浏览器内开发环境。

可以在 <http://getfirefox.com/> 中下载 Mozilla Firefox，一旦运行 Firefox，就可以在 <http://getfirebug.com/>

中获取 Firebug 扩展。

此外，本书中的一些示例还要求服务器端脚本能够完全运行，而且示例代码采用 PHP 编写。因此，能够访问 Web 服务器上的 PHP 将有所帮助，这些脚本在 PHP 4 及以上的任何版本中都应该能够运行。

源代码

在读者学习本书中的示例时，可以手工输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/> 或 www.tupwk.com.cn/downpage 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有源代码。

注释：

由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-38459-6。

下载代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果你在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者更方便地学习，当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件我们就会检查你的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

P2P.WROX.COM

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于你张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给你传达感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自

己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 **Register** 链接。
- (2) 阅读使用协议，并单击 **Agree** 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 **Submit** 按钮。
- (4) 你会收到一封电子邮件，其中的信息描述了如何验证帐户，完成加入过程。

提示：

不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 I 部分 Prototype

第 1 章 扩展和增强 DOM 元素	3
1.1 扩展 DOM 元素	3
1.1.1 美元符号函数: <code>\$()</code>	4
1.1.2 <code>\$\$()</code>	4
1.1.3 <code>Element.extend()</code>	4
1.1.4 将 <code>Element</code> 对象用作构造函数	5
1.2 DOM 导航	6
1.2.1 <code>adjacent</code> 方法	6
1.2.2 <code>ancestors</code> 方法	7
1.2.3 <code>up/down/next/previous</code> 方法	8
1.2.4 <code>descendants/descendantOf/first</code> <code>Descendant/immediateDescendants</code> 方法	10
1.2.5 <code>getElementsBySelector</code> 和 <code>getElementsByClassName</code> 方法	10
1.2.6 <code>childElements</code> 函数	11
1.3 修改页面内容	11
1.3.1 <code>insert(element, content)</code> 和 <code>insert(element,</code> <code>{position:content})</code>	11
1.3.2 <code>remove</code>	11
1.3.3 <code>replace</code>	12
1.3.4 <code>update</code>	12
1.4 操作元素的大小、 位置和可见性	13
1.4.1 放置元素	13
1.4.2 处理偏移	14
1.4.3 显示和隐藏元素	15
1.4.4 调整元素的大小	16
1.5 处理 CSS 和样式	16

1.5.1 <code>addClassName</code> 、 <code>removeClassName</code> 和 <code>toggleClassNames</code>	16
1.5.2 <code>hasClassName</code> 和 <code>classNames</code>	19
1.5.3 <code>setStyle</code> 和 <code>getStyle</code>	19
1.6 使用自己编写的方法扩展 <code>Element</code> 对象	19
1.7 本章小结	20
第 2 章 处理跨浏览器事件	21
2.1 注册事件处理程序	21
2.2 响应事件	23
2.2.1 <code>event.target</code> 属性、 <code>this</code> 属性 和 <code>Event.element</code> 方法	23
2.2.2 <code>Event.extend(event)</code>	25
2.2.3 <code>Event.stop(event)</code>	26
2.3 触发调度事件	27
2.4 本章小结	30
第 3 章 简化 AJAX 和动态数据	31
3.1 建立到服务器的请求	31
3.1.1 <code>Ajax.Request</code>	32
3.1.2 回调	33
3.1.3 <code>Ajax.Response</code>	34
3.2 以全局方式响应数据变化	35
3.3 动态更新页面	35
3.3.1 <code>Ajax.Updater</code>	36
3.3.2 <code>Ajax.PeriodicalUpdater</code>	37
3.4 本章小结	38
第 4 章 处理表单	39
4.1 操作表单元素和数据	39
4.1.1 <code>Form</code> 对象	39
4.1.2 结合使用 <code>Form</code> 对象的方法	41
4.2 验证表单数据	44

4.3	使用 AJAX 提交表单	47	7.1.1	get 方法	85
4.4	本章小结	48	7.1.2	getElementsByClassName 方法	86
第 5 章	操作通用数据结构和函数	49	7.1.3	getFirstChild 和 getLastChild 方法	87
5.1	增强原生对象并引入类	49	7.1.4	getFirstChildBy 和 getLastChildBy 方法	88
5.1.1	对象扩展	49	7.1.5	getChildren 和 getChildrenBy 方法	88
5.1.2	Class 对象	51	7.1.6	getElementsBy 方法	90
5.2	修改和分析字符串	52	7.1.7	getAncestorByTagName 方法	91
5.3	生成模板化内容	54	7.1.8	getAncestorByClassName 方法	94
5.4	绑定和操作函数	55	7.1.9	getAncestorBy 方法	95
5.4.1	绑定函数	56	7.1.10	Element 实用工具	96
5.4.2	操作函数的其他方法	57	7.1.11	Selector 实用工具	97
5.5	改进数组、散列和迭代器	58	7.2	操作内容	99
5.5.1	使用 for...in 循环会导致一些问题 的原因	58	7.2.1	insertBefore 方法	99
5.5.2	Enumerable 类	59	7.2.2	insertAfter 方法	99
5.5.3	改进 Array 对象	62	7.2.3	处理类名	99
5.5.4	引入 Hash 类	63	7.2.4	setStyle 方法	101
5.6	处理数值和日期	64	7.2.5	getStyle 方法	102
5.6.1	数值	64	7.2.6	setXY 方法	102
5.6.2	日期	65	7.3	本章小结	103
5.7	本章小结	65	第 8 章	处理跨浏览器事件	105
第 6 章	扩展 Prototype	67	8.1	注册页面事件和元素准备就绪事件	105
6.1	Script.aculo.us	67	8.1.1	onDOMReady 事件处理程序	106
6.2	Moo.fx for Prototype	73	8.1.2	执行作用域和参数传递	108
6.2.1	Fx.Tween	74	8.1.3	onAvailable 函数	110
6.2.2	Fx.Morph	74	8.1.4	onContentReady 函数	112
6.2.3	Fx.Transitions	74	8.1.5	on/addListener 函数	112
6.2.4	Fx.Slide	75	8.1.6	removeListener 函数	114
6.3	Rico	75	8.2	处理键盘和鼠标输入	116
6.3.1	组件	76	8.2.1	KeyListener 实用工具	116
6.3.2	动画效果	78	8.2.2	getCharCode 函数	119
6.3.3	圆角	78	8.2.3	getXY	121
6.3.4	拖放	79	8.2.4	getTarget 函数	121
6.4	本章小结	79	8.2.5	getRelatedTarget 函数	122
第 II 部分 YUI 库			8.2.6	preventDefault 函数	124
第 7 章	利用 YUI 库遍历和操作 DOM	85			
7.1	遍历 DOM 以及查找元素	85			

8.2.7	stopPropagation 函数	125	11.2.3	Panel	186
8.2.8	stopEvent 函数	126	11.3	使用选项卡和树状视图呈现内容	189
8.3	处理自定义事件	127	11.3.1	TabView	189
8.3.1	创建和订阅自定义事件	128	11.3.2	TreeView	198
8.3.2	退订自定义事件	130	11.4	本章小结	209
8.3.3	subscribeEvent 方法	131	第 12 章	利用窗口部件构建用户界面	
8.4	管理浏览器历史并修正后退按钮	131		(第二部分)	211
8.5	本章小结	135	12.1	装配按钮、滑块和菜单	211
第 9 章	使用动画和拖放	137	12.1.1	按钮	211
9.1	组合基本的动画序列	137	12.1.2	样式化	211
9.1.1	Anim 类	138	12.1.3	滑块	216
9.1.2	Motion 类	141	12.1.4	菜单	223
9.1.3	Scroll 类	143	12.2	提供日期选择功能	233
9.1.4	ColorAnim 类	144	12.2.1	简单的日历	233
9.2	平滑动画路径和运动	148	12.2.2	事件	236
9.2.1	缓动	148	12.2.3	多页日历	238
9.2.2	曲线路径(贝塞尔曲线)	150	12.3	启用富内容编辑	239
9.3	带有拖放功能的交互动画	155	12.3.1	事件	244
9.3.1	DD	155	12.3.2	实际使用编辑器	246
9.3.2	DDProxy	155	12.4	本章小结	247
9.4	本章小结	159	第 13 章	利用 YUI 核心增强开发	249
第 10 章	简化 AJAX 和动态加载	161	13.1	应用名称空间和模块性	249
10.1	建立 HTTP 请求并获取数据	161	13.1.1	名称空间	249
10.1.1	asyncRequest 函数	162	13.1.2	语言扩展	250
10.1.2	JSON	165	13.1.3	模拟类继承关系	251
10.2	动态加载库和组件	168	13.2	检测浏览器环境和可用模块	261
10.2.1	Get Utility	168	13.2.1	YAHOO.env.ua	261
10.2.2	YUI Loader Utility	171	13.2.2	YAHOO.env.getVersion	261
10.3	本章小结	175	13.2.3	YAHOO_config	263
第 11 章	利用窗口部件构建用户界面		13.3	日志记录和调试	265
	(第一部分)	177	13.4	本章小结	267
11.1	AutoComplete 实用工具与表单字段结合使用	177	第 14 章	处理数据、表和图表	269
11.2	为内容构建容器	183	14.1	格式化日期和数字	269
11.2.1	Module	183	14.1.1	日期	269
11.2.2	Overlay	185	14.1.2	数字	271
			14.2	获取数据源	273

14.3	呈现表数据	280
14.4	绘制图表和图形	284
14.5	本章小结	291
第 15 章	使用 YUI CSS 工具	293
15.1	建立跨浏览器一致性	293
15.2	控制字体	295
15.3	利用网格构建布局	297
15.3.1	模板	299
15.3.2	嵌套网格	300
15.4	本章小结	303
第 16 章	构建和部署	305
16.1	来自 Yahoo! 的共享 YUI 文件	305
16.2	减少和优化加载时间	308
16.3	本章小结	310

第 III 部分 Ext JS

第 17 章	架构和库约定	313
17.1	何时使用 Ext JS	313
17.2	如何使用 Ext JS	314
17.3	Ext JS 的面向对象设计	315
17.3.1	Ext.namespace	316
17.3.2	Ext.override	316
17.3.3	Ext.extend 和构造函数约定	317
17.3.4	Ext.apply	318
17.3.5	Ext.applyIf	318
17.4	功能强大的实用工具函数	319
17.4.1	Function.createCallback	319
17.4.2	Function.createDelegate	320
17.4.3	Function.createInterceptor	320
17.4.4	Function.createSequence	321
17.4.5	Function.defer	322
17.5	Ext JS 的基于事件的设计	322
17.5.1	Ext.util.Observable. addEvents	322
17.5.2	Ext.util.Observable.addListener /on	323

17.5.3	Ext.util.Observable. removeListener /un	323
17.5.4	Ext.util.Observable.fireEvent	323
17.5.5	Ext.util.Observable .addListener	324
17.5.6	Ext.util.Observable .purgeListeners	324
17.5.7	Ext.util.Observable .relayEvents	324
17.5.8	Ext.util.Observable.suspendEvents /resumeEvents	324
17.5.9	Ext.util.Observable.capture /releaseCapture	324
17.6	本章小结	325

第 18 章	元素、DomHelper 和模板	327
18.1	元素操作	327
18.1.1	Ext.Element	327
18.1.2	Ext.Element 方法	331
18.2	DOM 遍历	335
18.2.1	Ext.DomQuery	336
18.2.2	Ext.DomQuery 方法	337
18.3	DOM 操作	338
18.3.1	Ext.DomHelper	338
18.3.2	Ext.Template	341
18.3.3	Ext.XTemplate	343
18.4	CSS 操作	344
18.5	本章小结	346
第 19 章	组件、布局和窗口	347
19.1	Ext JS Component 系统	347
19.1.1	Ext.Component	347
19.1.2	Ext.ComponentMgr	348
19.1.3	Ext.BoxComponent	350
19.1.4	Ext.Container	350
19.2	Ext JS Component 生命周期	352
19.2.1	初始化	352
19.2.2	呈现	352
19.2.3	销毁	353
19.3	Ext.Viewport	354

19.4	Ext.Container 布局	355	21.2.1	Ext.grid.ColumnModel	394
19.4.1	Ext.layout.ContainerLayout	355	21.2.2	Ext.grid.AbstractSelection Model	397
19.4.2	Ext.layout.BorderLayout	356	21.2.3	Ext.grid.CellSelectionModel	397
19.4.3	Ext.layout.ColumnLayout	357	21.2.4	Ext.grid.RowSelectionModel	397
19.4.4	Ext.layout.TableLayout	357	21.2.5	Ext.grid.CheckboxSelection Model	397
19.4.5	Ext.layout.AnchorLayout	358	21.2.6	Ext.grid.GridView	398
19.4.6	Ext.layout.AbsoluteLayout	358	21.2.7	Ext.grid.GroupingView	398
19.4.7	Ext.layout.FormLayout	359	21.2.8	其他的定制方法	399
19.4.8	Ext.layout.FitLayout	361	21.3	本章小结	399
19.4.9	Ext.layout.Accordion	361			
19.4.10	Ext.layout.CardLayout	362			
19.4.11	创建自定义布局	362			
19.5	面板和窗口	363	第 22 章	表单控件、验证及其他	
19.5.1	Ext.Panel	363		功能	401
19.5.2	Ext.Window	365	22.1	表单控件介绍	401
19.5.3	Ext.WindowGroup	365	22.1.1	Ext.form.Label	402
19.5.4	Ext.WindowMgr	365	22.1.2	Ext.form.Field	402
19.6	本章小结	365	22.1.3	Ext.form.TextField	404
			22.1.4	Ext.form.FormPanel 和 Ext.form.BasicForm	405
第 20 章	数据处理以及服务器通信	367	22.1.5	其他表单控件	407
20.1	获取数据	367	22.1.6	Ext.form.NumberField	407
20.1.1	Ext.data.DataProxy	368	22.1.7	Ext.form.TextArea	407
20.1.2	Ext.data.HttpProxy	368	22.1.8	Ext.form.TriggerField	407
20.1.3	Ext.data.MemoryProxy	371	22.1.9	Ext.form.DateField	407
20.1.4	Ext.data.ScriptTagProxy	372	22.1.10	Ext.form.ComboBox	408
20.2	重新建模数据	373	22.1.11	Ext.form.TimeField	408
20.2.1	Ext.data.Record	373	22.1.12	Ext.form.Checkbox	408
20.2.2	Ext.data.DataReader	375	22.1.13	Ext.form.Radio	408
20.3	本地存储数据	379	22.1.14	Ext.form.CheckboxGroup	409
20.3.1	Ext.data.Store	379	22.1.15	Ext.form.RadioGroup	410
20.3.2	Ext.data.Record(回顾)	382	22.1.16	Ext.form.HtmlEditor	411
20.3.3	Ext.StoreMgr	383	22.1.17	Ext.form.Hidden	411
20.4	集成所有类	383	22.2	表单字段和表单验证	411
20.5	本章小结	385	22.2.1	验证消息	412
			22.2.2	高级验证技术	413
			22.2.3	表单级验证	414
第 21 章	DataView 和网格	387	22.3	其他功能	414
21.1	Ext.DataView	387	22.3.1	状态管理	414
21.1.1	操作 DataView	389			
21.1.2	DataView 事件	392			
21.2	Ext.grid.GridPanel	393			

22.3.2	浏览器历史	415	25.2.1	将内联处理程序连接到事件	464	
22.3.3	视觉效果	416	25.2.2	将全局函数连接到事件	464	
22.3.4	拖放	416	25.2.3	将对象方法连接到事件	465	
22.3.5	工具栏和菜单	416	25.2.4	断开与事件的连接	466	
22.3.6	主题	417	25.2.5	特殊的事件处理与事件对象	467	
22.3.7	树	417	25.3	连接到对象方法	468	
22.3.8	键盘导航	417	25.4	利用 NodeList 建立连接	470	
22.3.9	其他更多技术	417	25.5	发布与订阅事件主题	473	
22.4	本章小结	418	25.5.1	结合使用事件主题与 DOM 事件处理程序	473	
第IV部分 Dojo			25.5.2	将对象方法用作订阅者	476	
第 23 章 利用 Dojo 核心增强开发			421	25.5.3	取消订阅已发布的消息	477
23.1	获取 Dojo	421	25.5.4	将对象方法转换成发布者	477	
23.1.1	通过 AOL CDN 使用 Dojo	421	25.6	使用 Dojo 行为	479	
23.1.2	下载最新的 Dojo 发行版本	422	25.6.1	利用行为查找节点并建立连接	479	
23.1.3	尝试尚处于开发阶段的 Dojo	422	25.6.2	利用行为连接对象方法	481	
23.2	尝试使用 Dojo	422	25.6.3	利用行为发布事件主题	482	
23.3	研究 Dojo 核心	427	25.7	本章小结	483	
23.3.1	声明、加载和提供依赖项	428	第 26 章 编排动画			
23.3.2	定义类和使用继承关系	430	26.1	对 CSS 样式属性制作动画	485	
23.3.3	在 HTML 标记中声明对象	433	26.2	使用淡入淡出转换	488	
23.4	本章小结	438	26.3	使用擦除转换	489	
第 24 章 操作 DOM			439	26.4	使用滑动动画移动元素	490
24.1	查找 DOM 元素	439	26.5	使用缓动控制运动	492	
24.1.1	利用 dojo.byId 查找 DOM 元素	440	26.6	顺序链接动画	495	
24.1.2	利用 dojo.query 查找元素	440	26.7	以并行方式组合动画	496	
24.2	处理 DOM 元素列表	443	26.8	使用 NodeList 动画方法	498	
24.2.1	过滤和优化节点列表	443	26.9	研究动画对象	500	
24.2.2	处理节点列表	452	26.10	本章小结	505	
24.2.3	使用 NodeList 的其他方法	460	第 27 章 处理 AJAX 和动态数据			
24.3	本章小结	460	27.1	建立简单的 Web 请求	507	
第 25 章 处理事件			461	27.1.1	建立简单的请求并处理响应	508
25.1	响应页面加载和卸载事件	461	27.1.2	使用一个处理程序同时处理错误和成功响应	510	
25.2	连接到 DOM 事件	463				

27.2	利用 Deferred 处理 Web 响应	511	28.1.1	使用 JavaScript 实例化窗口 部件	554
27.2.1	为成功和错误响应注册 处理程序	511	28.1.2	在 HTML 标记中声明 窗口部件	555
27.2.2	在一次调用中注册错误和 成功处理程序	512	28.1.3	利用正则表达式验证输入	556
27.2.3	注册一个同时处理错误和 成功响应的处理程序	513	28.1.4	在提交时实施表单验证	557
27.3	处理响应格式	514	28.1.5	处理数字与货币值	559
27.3.1	处理文本响应	514	28.1.6	处理日期和时间字段	561
27.3.2	处理 XML 响应	515	28.1.7	增强单选按钮和复选框	563
27.3.3	处理 JSON 响应	516	28.1.8	使用滑块进行离 散值的选取	568
27.3.4	处理注释过滤 JSON 响应	516	28.1.9	使用动态文本区域和富文本 编辑器	570
27.3.5	处理 JavaScript 响应	517	28.2	管理应用程序布局	571
27.4	指定请求方法	519	28.2.1	搭建应用程序布局页面	572
27.4.1	构建一个服务器端请求 回应工具	519	28.2.2	将 ContentPane 用作布局 构建块	573
27.4.2	尝试多种请求方法	522	28.2.3	利用 BorderContainer 管理 布局区域	574
27.5	使用请求参数和内容	524	28.2.4	利用 StackContainer 管理 内容可见性	576
27.5.1	建立带有查询参数的 GET 请求	524	28.2.5	利用 AccordionContainer 切换内容窗格	582
27.5.2	建立带有响应正文参数的 POST 请求	526	28.2.6	利用 TabContainer 构建选项卡 式的内容窗格	584
27.5.3	建立带有原始正文内容的 POST 请求	527	28.2.7	利用 SplitContainer 划分 布局区域	586
27.6	利用就地请求增强表单	528	28.3	创建应用程序控件和对话框	588
27.7	使用跨域的 JSON 源	535	28.3.1	构建可单击按钮并为其编写 脚本	589
27.7.1	通过轮询变量来加载 JSON	536	28.3.2	编排弹出式上下文菜单	591
27.7.2	利用回调加载 JSON	539	28.3.3	组合按钮和菜单	594
27.8	利用 IFrame 建立请求	541	28.3.4	利用按钮和菜单构建 工具栏	595
27.8.1	利用代理脚本打包 IFrame 数据	541	28.3.5	利用进度条提供完成进度 反馈	596
27.8.2	利用 IFrame 处理响应格式	542	28.4	对窗口部件应用主题	598
27.8.3	利用表单和 IFrame 上传文件	543	28.4.1	检查窗口部件的 DOM 结构	598
27.9	本章小结	550			
第 28 章	利用窗口部件构建用户界面	551			
28.1	构建并验证表单	551			

28.4.2	加载主题并将其应用于窗口部件	600	31.2	研究 MooTools Core	637
	窗口部件	600	31.2.1	检查 MooTools 版本	638
28.4.3	定制并检查可用主题	601	31.2.2	确定类型	638
28.5	本章小结	602	31.2.3	检查已定义的值	639
第 29 章	构建和部署 Dojo	603	31.2.4	选择已定义的值	639
29.1	研究 Dojo 构建	603	31.2.5	选取随机数	639
29.2	查找构建系统	604	31.2.6	获取当前时间	639
29.3	创建自定义构建配置文件	604	31.2.7	清除定时器和时间间隔	640
29.4	生成自定义构建	606	31.2.8	合并和扩展对象	640
29.5	检验并使用自定义构建	607	31.3	使用数组扩展	641
29.6	本章小结	609	31.3.1	使用 .each() 和 .forEach()	
第 30 章	扩展 Dojo	611	处理数组项	642	
30.1	研究 DojoX 子项目	611	31.3.2	过滤和映射数组项	644
30.2	尝试高级窗口部件	612	31.3.3	检查数组项的内容	644
30.2.1	构建鱼眼菜单	612	31.3.4	将数组项转换成对象属性	646
30.2.2	利用 Toaster 窗口部件建立动画通知	613	31.3.5	扩展与合并数组	646
30.3	采用高级表单验证辅助函数	614	31.3.6	展平嵌套数组	647
30.4	从模板生成内容	617	31.3.7	利用 .link() 应用选择规则	648
30.5	绘制形状以及呈现图表	621	31.4	使用散列数据结构	649
30.5.1	绘制形状和线	621	31.4.1	定义散列和散列快捷方式	649
30.5.2	呈现图表和曲线图	622	31.4.2	设置与获取键和值	649
30.6	使用编码和加密例程	624	31.4.3	映射和过滤散列	650
30.6.1	生成 MD5 散列值	625	31.4.4	使用 .every() 和 .some()	
30.6.2	采用 Base64 编码数据	626	检查散列	651	
30.6.3	采用 Blowfish 加密数据	628	31.4.5	扩展与合并散列	651
30.7	导航 JSON 数据结构	629	31.4.6	将散列转换成 URL 查询字符串	652
30.8	研究 DojoX 的其他功能	631	31.5	使用字符串扩展	652
30.9	本章小结	632	31.5.1	检查字符串内容	652
			31.5.2	将字符串转换成数字和颜色	653
			31.5.3	使用简单的替换模板	653
			31.5.4	执行其他的转换	654
			31.6	使用函数扩展	654
			31.6.1	将函数绑定到对象上下文	655
			31.6.2	间歇性地延迟和设置函数调用	656
			31.6.3	尝试带有潜在异常的函数调用	657
			31.7	使用面向对象编程方法	657
第 V 部分 MooTools					
第 31 章	利用 MooTools 增强开发	635			
31.1	获取 MooTools	635			
31.1.1	下载最新的 MooTools 发布版本	635			
31.1.2	尝试正在开发的 MooTools 版本	637			

31.7.1	构建类和子类	657	33.2.1	加载 JavaScript 和 JSON 源	705
31.7.2	将方法和属性注入到 现有类中	659	33.2.2	包含额外的 CSS 样式表	706
31.7.3	实现混入类	660	33.2.3	获取图像和图像集合	707
31.8	本章小结	665	33.3	建立 Web 请求	709
第 32 章	操作 DOM 以及处理事件	667	33.3.1	执行基本的 Web 请求	710
32.1	查找 DOM 元素	667	33.3.2	获取和更新 HTML 内容	716
32.1.1	使用 \$() 和 ID 查找元素	668	33.3.3	请求并使用 JavaScript 和 JSON 数据	719
32.1.2	使用 \$\$() 和 CSS 选择器 查找元素	668	33.4	本章小结	722
32.1.3	导航 DOM 结构	670	第 34 章	构建用户界面以及使用 动画	723
32.2	操作元素样式和属性	672	34.1	编排动画	723
32.2.1	操作元素 CSS 类	672	34.1.1	检查元素大小和位置	725
32.2.2	操作元素视觉样式	674	34.1.2	使用 MooTools Fx 编排 动画	727
32.2.3	操作元素属性	676	34.1.3	研究预制动画和效果	731
32.2.4	操作扩展的元素属性	678	34.1.4	使用 Fx.Slide 动画	732
32.2.5	使用元素存储机制安全地 管理元数据	682	34.1.5	使用 Fx.Scroll 动画	733
32.3	修改 DOM 结构	683	34.1.6	研究 MooTools Fx. Transitions	734
32.3.1	创建新元素	683	34.1.7	研究动画事件	737
32.3.2	复制元素	684	34.1.8	利用 Fx.Morph 对多个属性 制作动画	740
32.3.3	获取元素	685	34.1.9	利用 Fx.Elements 对多个元素 制作动画	742
32.3.4	注入元素	685	34.2	使用用户界面窗口部件	744
32.3.5	创建并附加文本节点	686	34.2.1	构建折叠布局	744
32.3.6	替换和包装元素	686	34.2.2	向页面导航中添加 平滑滚动	748
32.3.7	接纳元素	686	34.2.3	启用可拖动元素	749
32.3.8	销毁和清空元素	687	34.2.4	自动滚动窗口和元素	753
32.4	附加监听程序并处理事件	687	34.2.5	启用拖放目标	754
32.4.1	响应页面加载和卸载事件	688	34.2.6	构建可排序列表	756
32.4.2	添加和删除事件处理程序	689	34.2.7	使用工具提示	759
32.4.3	检查事件包装器对象	694	34.2.8	构建滑块控件	762
32.5	本章小结	699	34.3	本章小结	764
第 33 章	简化 AJAX 以及处理动态 数据	701			
33.1	操作浏览器 cookie	701			
33.1.1	使用 cookie 函数	702			
33.1.2	使用 cookie 支持的散列	703			
33.2	动态加载页面素材	704			