

Mastering Dojo
JavaScript and Ajax Tools for Great Web Experiences

精通Dojo

Rawld Gill

[美] Craig Riecke 著
Alex Russell

DOJO中国 译

- 深刻剖析Dojo工作原理
- Dojo之父执笔的权威之作
- 国内权威社区DOJO中国组织翻译

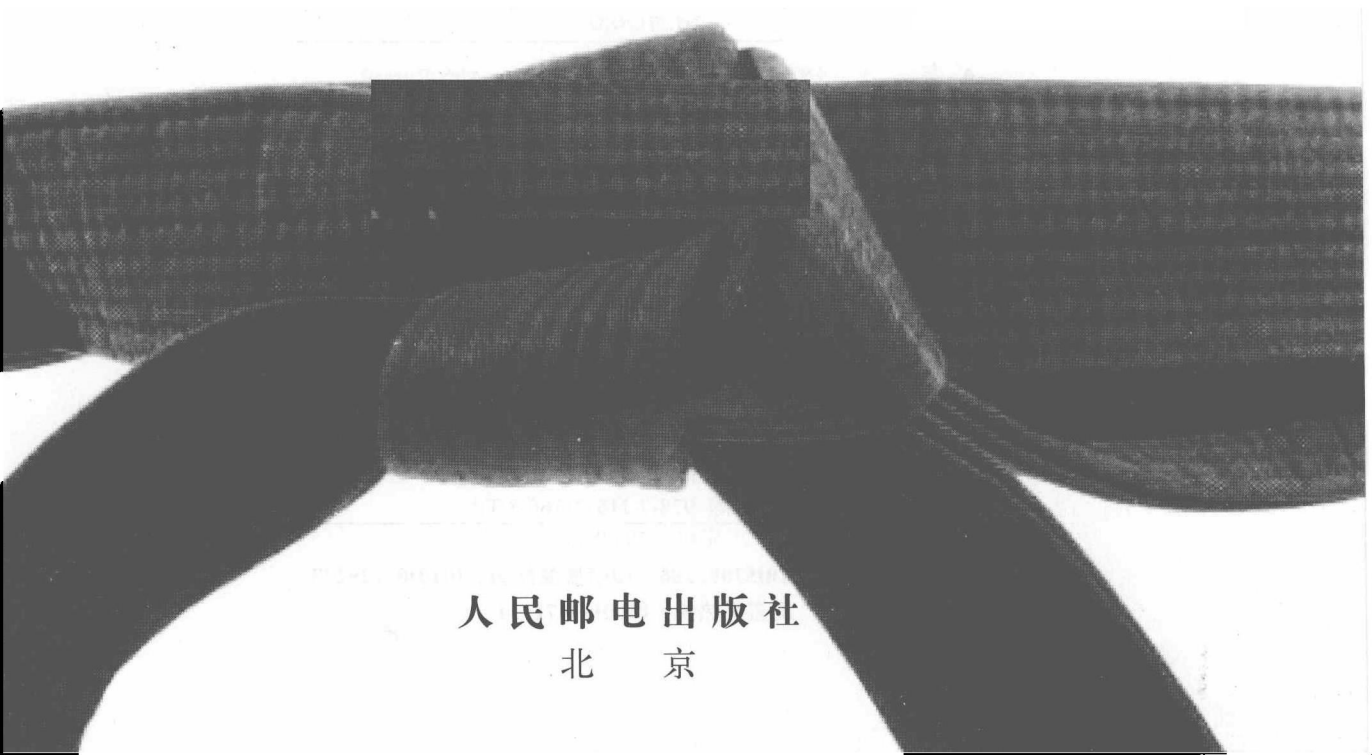


人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Web开发系列

Mastering Dojo
JavaScript and Ajax Tools for Great Web Experiences

精通Dojo



人民邮电出版社
北京

图书在版编目 (CIP) 数据

精通 Dojo / (美) 吉尔 (Gill, R.), (美) 里克 (Riecke, C.), (美) 拉塞尔 (Russell, A.) 著; DOJO 中国译. —北京: 人民邮电出版社, 2009.9
(图灵程序设计丛书)
ISBN 978-7-115-21166-8

I. 精… II. ①吉…②里…③拉…④D… III. 计算机网络—程序设计 IV. TP393.09

中国版本图书馆CIP数据核字 (2009) 第128684号

内 容 提 要

本书详尽阐述了 Dojo 和 Dijit。第一部分展示如何使用 Dojo 快速简便地构建强大的应用; 第二部分多方面讨论了 Dojo Core, 它是 Dojo 中所有功能的基础; 第三部分讲述 Dijit, 即 Dojo 小部件系统; 第四部分展示如何构建富因特网应用程序。书中各个部分都提供了真实有效的示例代码, 并经过了测试。

本书适合具备基本 Web 编程经验的开发人员阅读。

图灵程序设计丛书

精通Dojo

-
- ◆ 著 [美] Rawld Gill Craig Riecke Alex Russell
译 DOJO中国
责任编辑 傅志红
 - ◆ 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>
北京隆昌伟业印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 25.75
字数: 609千字
印数: 1—3 000册

2009年9月第1版

2009年9月北京第1次印刷

著作权合同登记号 图字: 01-2008-4300号

ISBN 978-7-115-21166-8/TP

定价: 69.00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

Copyright © 2008 Rawld Gill, Craig Riecke and Alex Russell. Original English language edition, entitled *Mastering Dojo: JavaScript and Ajax Tools for Great Web Experiences*.

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

本书中文简体字版由The Pragmatic Programmers, LLC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。

对本书的赞誉

“看到这本书出版，我太高兴了。Dojo功能强大，但是要了解使用哪些API，何时使用，如何使用，还是相当复杂的。本书不仅详细解释了这些API，还进一步告诉开发人员如何高效地使用这些API。”

——IBM Dojo软件包项目Dijit主管Bill Keese

“本书告诉你如何构建出让最终用户印象深刻的Web应用程序，并让你充分享受整个构建过程。书中还解释了Dojo底层的JavaScript实现，既是为初学者考虑，也适合其他语言转型过来的读者。”

——软件工程师Ian Dees

“Dojo闪亮登场，本书为其红地毯铺路，鸣锣开道。作者向你展示无需安装任何东西就能轻松愉快地使用让人印象深刻的部件。我惊讶地发现，JavaScript决不是什么玩具语言，其上构建起Dojo，二者在Web开发项目中发挥的作用不可估量。快买这本书吧，拥有它，就像作者在与你并肩战斗，实在酷毙了。”

——bluecomIT.com应用开发人员Brian C.Reeve

“本书是学习Dojo工具集的一本易懂而深入的指南，是联机文档极好的补充材料。”

——WaveMaker软件公司高级工程师Steve Orvell

目 录

第 1 章 导论.....	1	第 3 章 连接外部服务.....	23
1.1 Dojo的一些主要方面.....	3	3.1 Dojo远程脚本.....	23
1.1.1 Dojo不只是一个程序库, 而是 一个工具包.....	3	3.2 与调用XHR相关的JavaScript习惯用语.....	24
1.1.2 Dojo着眼于广大的用户群.....	4	3.2.1 字面量和散列.....	24
1.1.3 Dojo是面向未来的.....	4	3.2.2 函数字面量.....	27
1.1.4 Dojo是开源的.....	4	3.3 使用dojo.data和dojox.grid.Grid 的Wish List.....	29
1.1.5 Dojo分为三个项目.....	5	3.3.1 JSON语言.....	31
1.1.6 Dojo具有高度的概念完整性.....	5	3.3.2 存根数据源.....	32
1.2 如何使用本书.....	6	3.3.3 数据可用的小部件, dojox.grid.Grid.....	34
1.2.1 预备知识.....	6	3.4 使用JSONP研究雪茄.....	37
1.2.2 示例代码.....	6	3.4.1 网格和事件.....	39
1.2.3 关于调试.....	6	3.4.2 XHR回调.....	40
1.2.4 本书大纲.....	7	3.5 使用dojo.xhrGet实现评论系统.....	42
1.3 致谢.....	7	3.6 错误与调试.....	45
第一部分 用 Dojo 玩转 Ajax		第二部分 Dojo API	
第 2 章 轻松创建功能强大的 Web 表单.....	10	第 4 章 深入 Dojo.....	50
2.1 顾客如何评价传统表单.....	10	4.1 模块化JavaScript.....	50
2.2 在服务器上安装Dojo.....	11	4.2 Dojo源代码组织.....	52
2.3 把Dojo和Dijit添加到网页.....	11	4.2.1 Dojo模块的分类.....	53
2.4 表单的布局.....	14	4.2.2 Dijit模块.....	54
2.4.1 准备页面.....	14	4.2.3 取得源代码.....	54
2.4.2 用选项卡来组织表单.....	15	4.3 载入Dojo.....	55
2.4.3 有问题吗.....	17	4.3.1 运行时配置.....	55
2.5 增强表单控制.....	19	4.3.2 dojo.require.....	57
2.5.1 验证字段.....	19	第 5 章 JavaScript 语言的扩展.....	58
2.5.2 简单输入日期.....	20	5.1 利用dojo.hitch实现绑定.....	58
2.6 组合起来.....	20		

5.1.1 绑定上下文	58	7.3.1 操纵DOM树示例	114
5.1.2 绑定参数	61	7.3.2 使用dojo.place来移动节点	115
5.2 JavaScript 1.6的数组方法	63	7.4 定位DOM节点	116
5.3 支持多态	66	7.4.1 用CSS和Dojo定位	116
5.4 组合、结构化与复制对象	68	7.4.2 动态定位节点	119
5.4.1 混合对象	68	7.5 动画	121
5.4.2 复制对象	69	7.5.1 DHTML 动画基础	121
5.4.3 复制类似数组的对象	70	7.5.2 使用dojo.animateProperty 来实现动画	122
第6章 异步编程	71	7.5.3 使用dojo.anim来实现动画	126
6.1 利用Dojo进行DOM事件编程	71	7.5.4 为问题列表挑战添加动画效果	126
6.1.1 编写事件处理函数	72	第8章 基于XHR、script和iframe 编写远程脚本	130
6.1.2 处理函数签名	72	8.1 原生远程脚本	130
6.1.3 键盘事件对象	74	8.2 使用Dojo的XHR框架	131
6.1.4 事件传播	75	8.2.1 调用dojo.xhr*	132
6.1.5 默认处理	76	8.2.2 args、ioArgs以及处理函数	133
6.1.6 关联处理函数	77	8.2.3 内容处理方式	138
6.1.7 利用dojo.addOnLoad执行初始 化代码	79	8.2.4 控制HTTP事务	142
6.1.8 DOM事件分类	83	8.2.5 使用表单	144
6.2 利用Dojo关联用户自定义事件	83	8.2.6 POST以及其他的HTTP方法	145
6.2.1 截获JavaScript函数	83	8.2.7 Dojo XHR的其他知识点	146
6.2.2 关联用户定义事件	84	8.3 使用script实现远程脚本	146
6.3 发布-订阅	86	8.3.1 探测动态加载的脚本	147
6.4 利用dojo.Deferred管理回调函数	87	8.3.2 Dojo对于动态Script元素的 支持	149
6.4.1 示例: 构建一个高性能显示 引擎	88	8.4 基于iframe编写远程脚本	152
6.4.2 实现进程控制	90	8.4.1 Dojo对于使用iframe的脚本的 支持	152
6.4.3 利用dojo.Deferred注册回调 函数	91	8.4.2 iframe的工具函数	154
6.4.4 错误处理	93	8.5 使用远程脚本访问Web服务	155
6.4.5 声明回调函数和错误处理函数	94	8.5.1 明确定义远程过程调用	155
6.4.6 取消回调函数处理	96	8.5.2 使用dojo.rpc执行远程过程 调用	156
第7章 DOM工具	100	8.5.3 在Yahoo中使用Dojo RPC	159
7.1 核心Dojo DOM工具函数	100	8.6 书签和不带导航的后退按钮	161
7.2 查找和编辑节点	104	8.6.1 简单的单页Web应用	161
7.2.1 dojo.query	105	8.6.2 使用dojo.back导航单页Web 应用	163
7.2.2 dojo.query选择器	107		
7.2.3 dojo.NodeList能做什么	111		
7.3 插入、移动和删除的DOM节点	113		

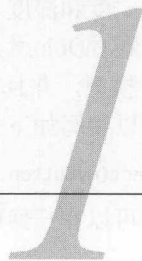
第 9 章 使用 dojo.declare 定义类	166	URL	203
9.1 为什么在JavaScript中使用面向对象编程	166	10.4 从JavaScript调用读取方法	204
9.2 定义简单的类	167	10.4.1 fetch方法和分页	204
9.2.1 原型以及原型的继承	167	10.4.2 排序	205
9.2.2 Shape类的类继承	168	10.4.3 fetch的处理函数	205
9.2.3 使用dojo.declare定义一个简单的类	169	10.4.4 更多的读取方法	206
9.2.4 dojo.declare创建的对象	170	10.5 Yahoo搜索驱动	207
9.2.5 标准的使用模式	172	第 11 章 Dojo 加载器和构建系统	213
9.3 使用单继承定义子类	172	11.1 概述	213
9.3.1 Circle对象空间	174	11.1.1 加载器和构建系统协同工作	213
9.3.2 调用父类中的方法	175	11.1.2 跨域加载器	214
9.3.3 手工添加父类方法	176	11.2 Dojo加载器	216
9.4 聚合和多继承	177	11.2.1 dojo.require	216
9.4.1 使用聚合派生新类	178	11.2.2 模块路径	216
9.4.2 各个实例的初始化	180	11.2.3 设置模块路径	217
9.4.3 使用聚合类进行类型测试	182	11.2.4 使用模块路径控制版本发布	218
9.5 预处理构造函数参数	184	11.2.5 dojo.provide	219
9.5.1 处理构造函数参数的方法	185	11.2.6 模块模式	219
9.5.2 使用散列来指定参数	186	11.2.7 为跨域加载器编程	220
9.6 解决属性名称冲突的问题	187	11.3 使用Dojo构建系统优化部署	222
9.7 两阶段构造	189	11.3.1 使用build打包Release程序	223
9.8 不使用构造函数创建自定义对象	191	11.3.2 Acmecorp的Magi-Browse	223
第 10 章 dojo.data	193	11.3.3 步骤1: 全部按照默认设置	224
10.1 鸟瞰	193	11.3.4 步骤2: 打包	227
10.1.1 捆绑的驱动类	194	11.3.5 步骤3: 层打包	230
10.1.2 数据存储器不是数据库	194	11.3.6 步骤4: 使用跨域加载器	231
10.1.3 特性	197	11.3.7 步骤5: 压缩	233
10.1.4 使用dojo.data.ItemFileReadStore的例子	198	11.4 使用Dojo-Rhino压缩JavaScript资源	233
10.2 dojo.data和增量搜索	199	第三部分 高级 Dijit	
10.2.1 建立一个数据存储器对象	199	第 12 章 编写小部件脚本	238
10.2.2 拉取提示数据	200	12.1 到底什么是小部件	238
10.2.3 自动完成	201	12.1.1 声明式小部件	239
10.2.4 查询	201	12.1.2 程式小部件	240
10.3 使用QueryReadStore进行作业划分	202	12.2 查找并操作声明式小部件	240
10.3.1 为什么你可能需要作业划分	202	12.3 程式地创建实例	243
10.3.2 将dojo.data的请求翻译为		12.4 扩展点	247
		12.5 实例: 实时表单	252

第 13 章 树	255	第 15 章 表单控件	296
13.1 简单树结构	255	15.1 表单控件的特性	296
13.2 带有层次结构的数据存储器	257	15.1.1 Dijit 表单控件和与之对应的 HTML 代码	296
13.2.1 ItemFileReadStore 和树	257	15.1.2 通用方法	297
13.2.2 JavaScript 中访问复杂数据	259	15.1.3 提交表单	297
13.3 扩展点	261	15.2 流水线编辑	298
13.3.1 盒子树	261	15.2.1 对话框和工具提示对话框	298
13.3.2 onClick 扩展点	262	15.2.2 内联编辑器	301
13.4 操作树结构	262	15.2.3 文本编辑	301
13.4.1 添加、删除、修改树节点	263	15.3 反馈	304
13.4.2 把树结构保存到服务器	264	15.3.1 工具提示	305
13.5 拖放	266	15.3.2 进度	306
13.5.1 一个小的重构	267	15.3.3 使用 Toaster 进行通知	307
13.5.2 _TreeNodes 和关联对象	269	15.4 日期、数值与国际化	308
13.5.3 检查施放操作	270	15.4.1 标准格式	309
13.5.4 施放和添加元素	272	15.4.2 约束与格式化	309
第 14 章 网格	275	15.4.3 取消标准形式	312
14.1 网格显示和设计	275	15.4.4 反馈	314
14.1.1 子行	277	15.5 动作按钮、工具栏和菜单	314
14.1.2 视图	278	15.6 ally	317
14.2 程式化地创建结构	279	15.6.1 Tab 顺序	318
14.2.1 JavaScript 中网格结构的定义	279	15.6.2 获取扩展点的角色和状态	318
14.2.2 行选择条	281	第 16 章 Dijit 主题、设计与布局	321
14.3 扩展点	282	16.1 主题结构	321
14.3.1 单元格层次扩展点	282	16.1.1 制作主题所需的文件	322
14.3.2 聚合函数和 onBeforeRow 扩展点	283	16.1.2 样式命名空间	323
14.3.3 行选择	287	16.1.3 样式继承	324
14.3.4 鼠标和键盘事件	288	16.1.4 针对不同的浏览器的特定的 样式定义	325
14.4 单元格编辑	289	16.2 改变网页的视觉效果	326
14.4.1 使用单元格编辑器	290	16.2.1 单个 HTML 元素	327
14.4.2 单元格格式化: formatter 扩展点	291	16.2.2 元素类	327
14.4.3 数据同步	292	16.2.3 主题段	328
14.5 网格操作	292	16.2.4 从零开始制作主题	328
14.5.1 排序	292	16.2.5 动态修改	329
14.5.2 过滤	293	16.3 无障碍和主题	330
		16.4 面板: ContentPane 和 TitlePane	331
		16.5 停靠容器: BorderContainer	335

16.6 栈容器	337	18.2.1 组织结构	364
第 17 章 创建和扩展小部件类	341	18.2.2 创建主程序	365
17.1 使用dijit.Declaration定义小部件类	341	18.3 第2步: 主菜单和命令系统	368
17.2 通过dojo.declare定义小部件类	346	18.3.1 命令元素存储器	368
17.3 小部件生命周期	349	18.3.2 主菜单	371
17.3.1 dojo.parser和小部件	350	18.4 第3步: 可定制的状态栏小部件	375
17.3.2 属性图	350	第 19 章 向 RIA 中添加动态内容	379
17.3.3 呈现小部件	351	19.1 第4步: 导航器面板及随需而变的数 据存储	379
17.4 扩展小部件	352	19.1.1 按需检索树数据	379
17.5 示例: 一个Yahoo问答小部件	352	19.1.2 实现一个延时树数据驱动	381
第四部分 富因特网应用程序		19.1.3 实现模拟服务	383
第 18 章 创建富因特网应用程序	358	19.1.4 连接各个部分	385
18.1 概述	359	19.2 第5步: 工作区对象	387
18.1.1 浏览器是一个UI平台	359	19.2.1 实现WSO小部件	388
18.1.2 浏览器应用框架	359	19.2.2 实现静态文本小部件	392
18.1.3 工作空间对象简介	360	19.2.3 将WSO连接到框架中	395
18.1.4 命令系统	361	第 20 章 继续学习	399
18.1.5 状态栏	362	20.1 基础库	399
18.1.6 可以按需载入数据的数据存 储器	363	20.2 图形	400
18.1.7 模拟服务	363	20.3 Dojo Data和存储器	401
18.1.8 浏览器应用程序	363	参考书目	402
18.2 第1步: 创建应用程序骨架	363		

第1章

导 论



啊，凯蒂，那扭扭捏捏的风俗习惯，碰见了伟大的君王就该退避三舍！亲爱的凯蒂，你跟我俩是不能让一国的风俗——那脆弱的绳子——来束缚住的。因为我们就是礼节的创造者呀！

莎士比亚话剧《亨利五世》第五幕第二场

一位新国王正在加冕登基。

过去两三年里，我们目睹了一场由新技术引起的革命，它改变了以往服务器端Web应用程序开发的模式。现在，轮到客户端Web应用程序出场了。请准备好舍弃当前客户端开发的“陈规陋习”吧。利用Dojo，我们将进入一个基于浏览器开发应用程序的新纪元。

Dojo是一套工具，用来构建更好的基于浏览器的应用程序。Dojo绝大部分使用客户端JavaScript来构建，大大地扩展了现代浏览器（甚至是IE）的能力，几乎抹去了本地应用程序与基于浏览器的应用程序之间的界限。

这是一个相当重要的论断。因为它表明，基于浏览器的（因此也是基于Web的）用户界面可以与那些最好的本地应用程序没有什么不同。同时也表明本地应用程序的用户界面也可以在浏览器中实现，而且并不需要使用重量级的、复杂而又平台相关的本地GUI框架（如Windows、Tk、Qt、Fox、AWT、SWT、Swing、Cocoa及其他）。

浏览器的普及使其成了新的用户界面平台。后台程序无论是在远端的HTTP服务器中还是在实现了HTTP协议的本地程序中，都已经不是问题。

然而遗憾的是，“现代”浏览器提供了一个不完整、不方便并且不兼容的编程环境。虽然你可以通过引入插件（如Flash）来解决这个问题，然而这将会违背一个重要的规则：把应用程序安装到本地是被严格禁止的。虽然对于本地程序而言打破这个规则是可能的，但是基于Web的应用程序还无法突破这个限制^①。这时，Dojo上场了。

Dojo解决了浏览器的缺陷（如浏览器不兼容性和内存泄露），并增加了新的功能（如HTML用户界面控件以及DOM查询）。虽然有很多JavaScript库也可以做到这点，但是它们绝大部分只关注某个特别的方面。例如，某些JavaScript库包含特效库，另一些关注JavaScript内核中已知的漏洞，还有一些实现了一个或多个HTML图形组件（用户界面控件）。相比之下，Dojo广泛地覆盖

^① 指分配的内存没有被释放，而不是指内存内容被泄漏给第三方。这是IE 6/IE 7的特殊问题。更深层次的原因来自于IE上JavaScript的COM实现。这最终与安全性无关，而与效率和承载力有关。——译者注

了所有这些（甚至更多的）功能区域。从这个角度而言，我们认为Dojo具有其他任何开源解决方案所不具备的广度和深度。

让我们来看看Dojo所具备的深度吧。Dojo将各个常用浏览器的事件系统进行规范化（IE没有实现W3C事件模型，并且它泄漏内存；其他的浏览器正好相反），如果你想要在Dojo中连接点击事件，那么可以编写如下代码：

```
dojo.connect(myButton, "click", myFunction);
```

这行代码可以在任何Dojo支持的浏览器中运行。再比如，你可以用下面的代码向网页中加入一个树形控件：

```
<div dojoType="dijit.Tree" label="Order" id="ordTree" store="ordJson" ></div>
```

这要比大多数本地框架中的做法简单很多。

Dojo同时具备不可思议的广度。它包含大约40个用户界面控件、1个图形框架、cometd^①支持、1个打包系统，还有很多其他的东西。如今，Dojo因其多样的功能而独树一帜。

也许你认为这代表复杂度也在不断增加，而我们认为正好相反——至少当你对比同类解决方案的时候结论不是这样的。Dojo的功能体系被组织成层次结构，因此你可以只关注对当前的工作有用的部分。当需求改变或增加的时候，Dojo将以其深度和广度随时满足你的需要。相比于其他更小、无论深度广度都很欠缺的库——就算它们声称可以让程序员一天就学会它（而且我们也不太相信能做到），但是3个月或6个月后，当你的需求超出了这个库的能力范围，你只能要么学习一个新的Ajax库，要么自己实现一个。这时不管何去何从，与从一开始就使用Dojo相比，都将面临更高的复杂性，付出更大的代价。

更进一步，Dojo社区的核心价值观之一就是“摒弃复杂性”。Dojo设计中的所有关键特性都在这个核心价值观下被仔细地研究、实现、重写、测试和使用。Dojo社区没有无视缺点的传统，也不会抱有在沙滩上造大楼的幻想，事实上它能融合成熟、理性甚至多疑的工程师所做的努力以及年轻黑客的前卫想法。整个社区花费了2007年大部分的时间来重构他们的系统，这显然体现了对卓越系统和专业精神的不懈追求，而这通常是开源项目所缺乏的。

行话解析

Ajax、Web 2.0以及RIA（富因特网应用程序）这些概念最近非常时髦，然而它们的含义并不十分清晰。从使用现代浏览器开发的观点来看，Ajax可以理解为“创造现代Web应用程序的手段”。我们也正是这么用的。Web 2.0则像是一个商业概念而不是技术词汇，本书将避免使用它。而一个基于浏览器的本地应用程序，如果可以与其他主机的HTTP服务器通信，我们则认为它是RIA。虽然这样的应用并不是因特网应用程序，但是这个词已经惯用，所以我们将继续使用它。不过当我们提到RIA时，是指一个很大程度上基于浏览器的单页应用程序。嗯，够精确。

^① cometd是一个低延迟通信技巧，服务器通过该技术可以将数据推送至浏览器。见 <http://cometd.com>。

最后必须强调的是, Dojo并非一个纸上谈兵的项目, 它已经为成百上千的公司所使用。如果你访问<http://www.myaol.com>并查看源代码, 便可以看到Dojo的身影。IBM是Dojo基金会的主要捐赠者, 并将Dojo应用于他们的WebSphere架构中; BEA和Sun在他们的产品中集成了Dojo。有了这些财富100强企业的使用和资助, 你可以确信, Dojo将产生更加深远的影响。

1.1 Dojo 的一些主要方面

让我们来鸟瞰Dojo。

1.1.1 Dojo 不只是一个程序库, 而是一个工具包

Dojo是一系列静态客户端JavaScript脚本, 不需要任何的客户端插件和服务器端组件, 包含下列内容。

- 一个用于统一浏览器差异的设计和实现, 允许同样的代码运行在多个浏览器中 (不需要用户进行浏览器或特性判断, 也不会有依赖浏览器的代码)。
- 一个函数库, 用于将有时难用、难理解而且不方便的W3C DOM编程模型抽象为方便、简洁而高效的接口。
- 一个函数库, 用于解决几个严重的浏览器错误, 例如内存泄露。这个库还实现了一些浏览器在几年之后很可能会提供的功能, 让你现在就可以用到它们。
- 一个已知最大的HTML图形组件集。
- 一个模块体系连同一个打包系统, 可以让你在开发以及将来发布的过程中将代码拆分成较小的、可管理的分块, 以便优化下载性能——这个过程中不需要更改任何代码。打包系统甚至允许将Dojo本身进行拆分, 使其适合你的项目。
- 一些独立的模块 (即可以根据需要加载它们), 它们实现了一些高级功能。

一些程序库组成了为下列目的所用的框架:

- 创建自定义的HTML小部件^① (widget)。
- 国际化 (Internationalization, i18n)。
- 本地化 (Localization, l10n)。
- 无障碍 (Accessibility, a11y)。

Dojo还包含一个称为构建系统的工具程序, 它可以将包含几百个文件的大项目打包成较小的、优化的压缩文件集, 以部署在产品服务器中。

到撰写本书时, Dojo正式支持IE (6及更高版本)、Firefox (1.5及更高版本)、Safari (3及更高版本) 和Opera (9及更高版本, Dijit不支持Opera)。

由于Dojo使用纯JavaScript编写, 所以它可以用于非浏览器环境、内嵌SpiderMonkey^②以及内

① 该词没有在中文中相对应的译法, 此处的译法遵照了IBM开发者网络dojo专题中的提法。请见: <http://www.ibm.com/developerworks/cn/web/wa-dojotoolkit/>。——译者注

② SpiderMonkey是历史上第一个JavaScript引擎的代号。该引擎由Netscape的Brendan Eich使用C编写, 之后SpiderMonkey开放源代码并由Mozilla基金会维护。——译者注

嵌Rhino^①环境中。当然，Dojo的大部分代码（用到了HTML、CSS以及XHR的所有部分）都不能用在这些环境中。然而，Dojo最核心的功能还是可用的，即加载器、语言扩展、异步编程、面向对象编程以及只依赖于JavaScript（不依赖于任何其他东西）的通用本地化数据仓库，都可以用在这些非浏览器环境中^②。

1.1.2 Dojo 着眼于广大的用户群

Dojo服务于一个很广的用户群，从简单Web站点的设计者到企业应用程序的开发人员。这是一个十分苛刻的需求，因为在做设计时，为一个用户群而做的优化，通常不适用于另一个用户群。大多数“Web设计”任务可以通过简单地加载dojo.js脚本来完成。从这个意义上讲，Dojo足以称为最具竞争力的、易用的轻量级程序库。另一方面，大型项目的需求更多。而Dojo包含一个能够按需加载不同的功能集的机制，这使得用户只需学习用以解决手头工作的部分就可以将Dojo投入使用，这实在是一个诱人的特性。

1.1.3 Dojo 是面向未来的

除了Dojo的设计原则和内容之外，基于浏览器的编程环境的现状是另外一个主要推动因素。假如浏览器适应某个标准（或者，至少它们是兼容的），假如JavaScript和DOM API解决了那些明显的缺陷，或者假如HTML包含了哪怕比1989年时更先进的用户界面控件，那么Dojo的很多内容可能都是不必要的。Dojo的架构师意识到，总有一天这些问题都将会被解决。而上述特性得到本地支持以后，我们希望能够不修改代码就直接使用它们。屏蔽基于浏览器的编程环境的关键缺陷，是设计Dojo的主要目的。这样，现代的、具有强大功能的程序（这些程序确实冲击着本地应用）可以预先定位于浏览器之上——由于本地浏览器环境正在不断改善，因而这实际上是在为将来的升级做准备。

解决浏览器的问题并非一朝一夕的事，而Dojo的模块设计可以直接提供这些便利——可以针对具体问题也可以针对不同的浏览器，就如同它们已经正常工作一般。简而言之，正当我们步入浏览器的下一轮战争时，Dojo提供了一个稳定的、基于浏览器的编程环境。

1.1.4 Dojo 是开源的

Dojo的源代码是可以免费获取的。它受到两个许可的保护，即修改的BSD许可和学术免费许可2.1版本。BSD许可适合商业应用，允许在商业产品中使用或修改Dojo，而不需要公开任何你所做的工作。自然，你可以针对此类产品任意修改Dojo（在BSD许可下，你甚至可以出售未经修改的Dojo副本）。

Dojo的开发过程是严格而开放的。源代码在SVN容器中维护，缺陷和改进被Trac程序跟踪。这两个服务器均可被匿名用户访问。开发过程强制执行编码风格方针，而且代码必须在提交入关键发布集前经过单元测试。

① Rhino与SpiderMonkey类似，但使用了Java进行编写。由Mozilla基金会维护。——译者注

② 在浏览器以外使用Dojo超出本书的讨论范围。

Dojo通过论坛、邮件列表以及提供免费服务的公司向使用者提供免费支持。

与很多开源项目不同, Dojo得到Dojo基金会的资助。Dojo基金会属于501(c)(6)^①条款规定的非盈利性基金会, 成立的目的是推广Dojo, 并为各行各业提供健康的JavaScript工程环境。存在这个基金会的一个关键好处就在于, 它可以确保用户不会因使用Dojo代码而引发侵权问题(例如侵犯专利或版权)。

要了解有关Dojo的所有信息, 可以访问<http://dojotoolkit.org/>。

1.1.5 Dojo 分为三个项目

Dojo包含以下三个项目。

- Dojo: 是Dojo的基础, 所有其他的功能都建立在其上。总之, 它包含大约50个JavaScript脚本和几个其他资源。这些资源用于处理浏览器差异的统一、JavaScript模块化、JavaScript核心库扩展、W3C DOM API扩展(包括解析和查询DOM)、远程脚本编程、Firebug Lite^②、拖放、数据存储API、本地化和国际化, 以及一些其他的附加功能。
- Dijit: Dojo的小部件框架和内置的小部件集(大约40个HTML用户界面小部件)。
- Dojox: Dojo扩展库。这包含了从表格小部件到绘图库的所有功能。Dojox如同Dojo的“西部荒野”, 这里有许多非常精妙、稳定和已经为实际盈利项目所使用的程序库, 同时也有很多完全属于试验阶段的系统。每个Dojox库都具备一份自述文件来描述其功能。

这三个项目保存在各自的源代码目录中。一般而言, Dojo和Dijit同时发布。到目前为止, 虽然Dojox曾与Dojo和Dijit一起发布, 但是这种情况在将来可能会发生改变。在本书中, 我们将详尽地讨论Dojo和Dijit, 而只会提到两三个Dojox项目。

1.1.6 Dojo 具有高度的概念完整性

尽管Dojo十分庞大, 但是其设计和实现过程都显示出了高度的概念完整性。在Fred Brooks的经典软件工程巨著《人月神话》[Bro95]中, Brooks认为对任何程序项目而言概念完整性(即功能和复杂度的比例)是最重要的优点。虽然这个理论已经被公认, 但很多长周期软件项目常常会忽略这一点。我们已经提到, 压制复杂度是Dojo项目的核心价值观之一。进一步讲, Dojo和Dijit项目目录由两个人来分别管理, 并且管理的人还要负责指导和协调项目的推进过程。这满足了Brooks理论中取得概念一致性的另一个要求——任命唯一的系统架构师。最终, 当我们探讨Dojo时, 你会发现一切都显得恰如其分。它很自然, 不会使你感到惊讶。所有这些都得益于高度的概念完整性。

① 美国《国内收入法》501(c)节定义了享有减免税权利的28种不同的非盈利性组织。501(c)(6)款是特别为行业工会、商会、经济发展公司、房地产联合会、商业联合会、职业橄榄球联盟以及其他的商业组织所设立的。归纳于此条款的组织在收取会费时可以减税。——译者注

② 一个免费的轻量级的Firebug, 可以嵌入到其他的JavaScript脚本中。用以在IE/Safari/Opera上提供类似于Firebug的功能。所有权属于Firebug的厂商Parakey, Inc. ——译者注

1.2 如何使用本书

这里将给出一些导读信息，它们将帮助你最大化地挖掘本书的价值。

1.2.1 预备知识

我们假设你至少具备最基本的Web编程经验。Dojo建立在基于标准的技术之上，尤其是(X)HTML、CSS、DOM以及JavaScript。虽然我们将经常在讨论特定的Dojo功能块时为这些底层技术提供一些定性的讲解，但是我们将不会展开讲解这些技术——否则将至少需要另外4本书的篇幅！万一你对这些技术一无所知，那么可以参考如下建议。

- 雅虎发表了一系列有关JavaScript和DOM编程的卓越讲座，作者是Douglas Crockford^①。
- 公认的JavaScript参考指南是《JavaScript权威指南》[Fla06]。它还包含一份非常好的DOM教程和文献。
- CSS（层叠样式表）通常神秘莫测。《CSS权威指南》[Mey06]给出了很好的尝试，但是依然还有很多文献讲述CSS，它们各有优缺点。
- 另一方面，HTML是相当容易掌握的。《HTML和XHTML权威指南》[MK08]是一本非常好的参考书，但是恐怕任何免费的在线文档对于学习HTML而言都足够了。

JavaScript是一门伟大的语言。与流行的（并且非常错误的）观点不同，它更接近于Lisp[®]而不是BASIC。它允许你快速和优美地表达极为强大的思想。Dojo把JavaScript发挥到了极致，我们也将这样做。

1.2.2 示例代码

我们为本书的各个部分编写了真实有效的示例代码，并使该示例代码有适宜的长度，便于你理解这些代码，而不会被长篇的代码打断阅读的连贯性。如果你觉得某代码片段缺少一些细节以至难以理解，请访问http://www.pragprog.com/titles/rgdojo/source_code，这里可以找到完整的可用示例代码。

1.2.3 关于调试

Web编程是一件非常动态的活动。最常见的情况是，写几行代码，在浏览器中点击刷新，然后观察到底发生了什么。然而，一个良好的调试环境对于最大限度地提高编程效率是非常重要的。由于Firefox + Firebug是这些调试环境中最好的选择（而且它们是免费的），因此我们将在后面的讨论中使用它们。如果你正在使用Firefox以外的浏览器（比如IE），那么你可能已经拥有了用于该浏览器上的良好的调试环境了。假如没有，Dojo包含Firebug Lite控制台，你可以在任何浏览器上运行它。参见4.3节的“Alex如是说”可获得更多的调试建议。

① 位于 <http://yuiblog.com/blog/2007/01/24/video-crockford-tjpl>、<http://yuiblog.com/blog/2006/11/27/video-crockford-advjs/> 与 <http://yuiblog.com/blog/2006/10/20/video-crockford-domtheory/>。

② Lisp是一种解释型语言，所有的数据结构以及程序都由表构成。函数本身也是一个表。故此也可以返回一个由函数动态构造出来的新函数。JavaScript借鉴了这种机制。——译者注

1.2.4 本书大纲

本书分为四部分。第一部分展示Dojo如何帮助你快速简便地构建强大的应用。其中的章节包含对给出示例的详细剖析，但有意避免了剖析这些示例所涉及的Dojo底层内容。第二部分对Dojo Core（所有Dojo中功能的基础）进行了多方面的考察。第三部分涉及Dijit，即Dojo小部件系统。第四部分展示如果构建一个富因特网应用程序。

1.3 致谢

首先，我们三个人要鼓励和感谢Dojo的贡献者们。没有这样的勇于奉献和高度智慧的社区，Dojo就不可能存在。我们希望本书正确地展示了他们无以伦比的成就。我们也特别感谢Bill Keese、Adam Peller、Ian Dees以及Brian Reeve。他们花费大量的时间和精力来阅读并评价我们的手稿。几乎每一页都包含了由于这些专家的建议而作出的改进。最后，我们向Pragmatic Programmers出版社的人员致敬，这是一个来自于程序员，服务于程序员的出版社。感谢他们提供的机会以及在出书过程中提供的所有帮助。

—— Rawld、Craig及Alex

我尤其要感谢我的合作者Craig和Alex，以及开发编辑Jackie。他们每个人都在自己的领域内有所建树，也为我能够完成更好的作品付出了不懈的努力。和这样一群人合作是件幸事。

—— Rawld

冒着被当成奥斯卡领奖演说的风险，我要感谢我的作文老师Carolyn Goodwin、James Alsop、Gerry Shapiro和Judith Sornberger。即使我没有多少写作的天赋，他们也鼓励我让我继续写下去。感谢Kathy，感谢她的星巴克优惠卡、她的爱以及无限的耐心，使得本书得以出版。如果生活是公平的，她一定会比我更早出书。

—— Craig

首先，我要感谢Craig和Rawld。这本书真正属于他们。他们在挖掘Dojo深层次内涵的过程中所做的贡献、展示出来的才华和愿景，造就了这本严谨、明朗、能够使Dojo开发者受益多年的图书。我很荣幸有机会在这个事业的一些方面做出贡献。Rawld、Craig和Jackie Carter使这个过程比我想象得更加从容。向他们致以最诚挚的感谢！

我要感谢促使Dojo如此杰出的那些捐赠者和开发者。他们为了Dojo付出了大量的时间与令人难以置信的努力，而不追求任何的物质回报。我有幸能够领导这个团队并在其中工作。他们富于乐观、锲而不舍、勇于奉献的精神。他们推进了Web向前发展。他们的工作改变了千百万人的日常生活。另外，感谢Dylan Schiemann、Tom Trenka、Bill Keese、David Schontzler、Paul Sowden、Eugene Lazutkin、Adam Peller、Becky Gibson、Pete Higgins、James Burke、Brad Neuberg和Owen Williams。我欠了太多人太多东西。

我最要谢谢我的妻子Jennifer，她在各个方面支持了我、激励了我，她的帮助、她的建议以及她的耐心是无限的。她不仅使Dojo成为了可能，同时也让我变成了一个更好的人。

—— Alex