

Pragmatic Guide to JavaScript
JavaScript
修炼之道

[法] Christophe Porteneuve 著
巩朋 张铁 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Pragmatic Guide to JavaScript

JavaScript

修炼之道

[法] Christophe Porteneuve 著
巩朋 张铁 译



人民邮电出版社

北京

图书在版编目(CIP)数据

JavaScript修炼之道 / (法) 波顿纽威
(Porteneuve, C.) 著 ; 巩朋, 张铁译. — 北京 : 人民
邮电出版社, 2011. 11

(图灵程序设计丛书)

书名原文: Pragmatic Guide to JavaScript

ISBN 978-7-115-26556-2

I. ①J… II. ①波… ②巩… ③张… III. ①
JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第209827号

内 容 提 要

本书是 JavaScript 的实战秘籍。作者将自己多年的编程经验融入其中, 不仅可以作为学习之用, 更是日常 JavaScript 开发中不可多得的参考手册, 使读者少走很多弯路。

本书的内容涵盖了当今流行的 JavaScript 库的运行机制, 也提供了许多应用案例。本书针对各任务采取对页式编排, 在对各任务的讲解中, 左页解释了任务的实现原理, 而右页则举出了该任务的代码片段以及可供对照参考的相关任务, 便于读者阅读和理解。

本书的读者既包括 JavaScript 编程的新手, 也包括已有不少 Web 应用编程经验的开发者。

图灵程序设计丛书

JavaScript修炼之道

◆ 著 [法] Christophe Porteneuve

译 巩 朋 张 铁

责任编辑 王军花

执行编辑 王一枝

◆ 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 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京艺辉印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 8.25

字数: 195千字

2011年11月第1版

印数: 1—3 000册

2011年11月北京第1次印刷

著作权合同登记号 图字: 01-2011-1376 号

ISBN 978-7-115-26556-2

定价: 29.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版权声明

Copyright © 2010 The Pragmatic Programmers, LLC. Original English language edition, entitled *Pragmatic Guide to JavaScript*.

Simplified Chinese-language edition copyright © 2011 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。

献词

谨以此书献给 Élodie——我最爱的妻子。

译者序

随着互联网的发展和移动上网设备数量的急剧增长,直接和用户打交道的网站前端变得越来越重要。而 JavaScript 作为前端开发的首选语言,其重要性不言而喻。

关于 JavaScript 的好书已有很多:入门级的有 Jeremy Keith 等人著的 *DOM Scripting*^①,进阶级的有 Nicholas Zakas 的 *Professional JavaScript for Web Developers*^②和 Douglas Crockford 的 *JavaScript: The Good Parts*^③。不过,对于前端开发来说,仅仅学习 JavaScript 这门语言是远远不够的。正如本书作者所言,使用原始的 JavaScript 编写网站前端就像用石斧和原木来建造一座摩天大厦:即便构建一个简单的 Web 应用,DOM 中的不一致、浏览器兼容性问题以及 JavaScript 语言本身存在的问题也会使它变得诡异莫测。

本书作者从实用性出发,把前端开发中的常见问题归纳为一个个任务,给出了常用框架下实现这些任务的方法和注意事项。这些任务的覆盖面很广,从 JavaScript 语言的基本技巧,到 DOM 操纵、事件和定时器,再到 UI 技巧,甚至还包含现在流行的混搭 (Mashup) 技术。读者既可以把这本书作为 JavaScript 开发的快速参考手册,也可以通读本书,以深入理解 JavaScript 和各个流行框架的使用方法。

本书第一部分、第二部分和附录由巩朋翻译,其余部分则由张铁完成。我们已尽力保证译文准确、通顺,但限于自身的程序设计能力和文字表达水平,难免有所遗漏,希望读者在容忍的同时能够给予指正。

译者

2011 年 8 月

① 此书中文版《JavaScript DOM 编程艺术 (第 2 版)》已由人民邮电出版社出版。* (标*号的是译者注,下同。)

② 此书中文版《JavaScript 高级程序设计 (第 2 版)》已由人民邮电出版社出版。*

③ 此书中文版《JavaScript 编程精粹》已由电子工业出版社出版。*

对本书的赞誉

“真希望我初学 JavaScript 那会就能有这本书！本书会让你在现实的 JavaScript 世界中游刃有余。它既讲述了当今流行的 JavaScript 库的运行原理，也提供了 JavaScript 正确实践的好建议和背景资料。作为最优秀的 JavaScript 开发者之一，作者将他多年的宝贵经验凝聚在本书中，使它成为日常 JavaScript 开发中的必读参考。”

——Thomas Fuchs
script.aculo.us 框架创始人

“这本书包含了一系列当今浏览器中既精妙又实用的 JavaScript 贴士和技巧，既有表单验证和 JSON 处理这样的基本技术，也有混搭和几何定位这样的应用实例。如果你想使用 JavaScript 编写更优秀的 Web 应用，我强烈建议你阅读本书。”

——Dylan Schiemann
SitePen 公司 CEO，Dojo 工具箱联合创始人

“现在市面上充斥着大量关于 JavaScript 的书，不过这些书大多都面向那些刚入门或是缺乏经验的 JavaScript 程序员。而本书并非如此，这本实用手册可谓出类拔萃。如果你是一个初学者，先去看些基础的书，等你具备一定经验后再读本书，效果会更好；如果你已经编写过不少 JavaScript 代码，你会在这本书中发现一系列你听过或见过的技术，但是本书会让你真正用好这些技术。总之，本书值得一读。”

——Stuart Langridge
kryogenix.org, @sil

引言

只要你在过去几年中对 JavaScript 有一丁点留意，你就会听过这个说法：JavaScript 将会是下一个重量级语言。这个曾经被用于半成品实现和滚动信息展示的语言，如今已经演进为一个世界级的、面向对象的动态语言，无论是在客户端还是在服务器端，都有超高速的实现。

一方面，JavaScript 的设计者通过 EcmaScript 5 (ES5) 标准为 JavaScript 注入了新鲜健康的血液；另一方面，各种强大的 JavaScript 引擎（比如 V8、JavaScriptCore、SpiderMonkey、Rhino 和 Carakan）以及不断涌现的标准和技术（在这方面，CommonJS^①和 Node^②处于领先地位），使 JavaScript 既适合在浏览器端工作，也适合独立在服务器端构建强大的架构。甚至连微软最新推出的 Internet Explorer 9 也大幅提高了 JavaScript 的运行速度。

JavaScript 不仅仅是一门强大的动态语言，而且是一个庞大的生态系统，这个系统由它的开发工具、基础设施、框架和工具箱构成。JavaScript 功能全面，易于编写，适用于各种编程任务，尤其是那些基于 Web 的应用和服务。

是时候进入 JavaScript 的世界了！

本书内容和读者对象

本书并没有打算从语言的角度来讲解 JavaScript。首先，JavaScript 的语法并不复杂，只要你拥有一些主流编程语言的经验——哪怕只了解基本的概念（诸如变量、循环等），就已经足够了。本书并不需要你拥有 JavaScript 的经验（当然有的话会更好），更不要求你是编程专家。

事实上，如果你打算研究 JavaScript 的语言精髓或是核心技术细节，那最好还是去读一些专门的材料，Opera 提供的 Web 标准课程中的“JavaScript 核心技能”部分就很不错^③。如果你需要

① 参见 <http://commonjs.org/>。

② 参见 <http://nodejs.org/>。

③ 参见 <http://www.opera.com/company/education/curriculum/>。

了解更复杂的语言实现细节，则可以去参考官方的语言标准，或是阅读那些圣经级的大部头，比如 David Flanagan 的 *JavaScript: The Definitive Guide*^①。

针对常见的客户端 JavaScript 任务，本书提供了快捷且定性的解决方案，这些任务从简单（比如获取一个 DOM 元素的引用）到复杂（比如基于 Ajax 的自动完成）。这也意味着我们需要使用 JavaScript、CSS、DOM、Ajax、JSON 等技术。本书绝大部分在讲客户端（浏览器端）技术，服务器端的内容则很少。为了便于说明，本书也用了一些 PHP 脚本，不过你可以使用你喜欢的技术，比如通过 Node 用 JavaScript 编写服务器端程序！

不要直接照搬本书中的代码。本书每个任务都围绕着某个核心概念、潜在问题或技术技巧展开论述。你要理解这些概念、问题和技巧并灵活运用，而不要拘泥于特定的任务。最终，学完本书，你将成为一名更出色的 JavaScript 程序员。

本书与 JavaScript 库

坦率地说，如果你正在进行重要的 JavaScript 开发，却没有使用已有的优秀框架，那你就已经做错了。在浏览器端编写高效的网页脚本是一项挑战。你会碰到各种各样的障碍：DOM 中的不一致、错误的语言实现、CSS quirk 和诡异的 Ajax bug 等。在服务器端，即使已经有了运行时的支持，你仍需要把数据存储、网络栈和模块系统等基本服务器模块组合在一起，这涉及大量的工作。

幸运的是，优秀的开发者已经为你解决了这些难题。现在有大量的优秀框架（尤其在客户端）供你选择。附录 C 详细介绍了主流的 JavaScript 框架。

任何称职且注重实效的 JavaScript 开发者都会依赖一个或多个优秀框架，因此，对于那些在我看来相当“基本”的任务，本书会介绍它们在所有主流客户端框架中的实现方法。我选择的框架包括 Prototype、jQuery、MooTools、YUI、Dojo 和 ExtJS，它们覆盖了开发者心仪的大部分框架。

对于不那么“基本”的任务，我主要用我的最爱 Prototype^②来处理（光箱特效除外，因为我认为在这个任务中 jQuery 插件更合适）。不过说实话，当你掌握了基本技巧之后，就可以用你常用的库来改写或重写我的解决方案。事实上，为了处理这种情况，我们在 GitHub 上建立了一个公用代码库^③。因此，如果你需要用其他库（比如 jQuery），只需要简单地点一下 GitHub 的分支（Fork）按钮，而且查找这些代码库的派生版本也很容易。

① 此书中文版《JavaScript 权威指南》已由机械工业出版社出版。*

② 本书作者是 Prototype 的铁杆粉丝。*

③ 参见 <http://github.com/tdd/pragmatic-javascript>。

此外，本书所有代码都被打包并存档在本书网站上^①。如果想测试这些代码的效果，请访问 <http://demo.pocketjavascript.com>。

本书内容简介

本书按主题划分成不同的部分，每个部分均包含了一系列任务。书的末尾有一些附录，你在阅读正文之前很可能需要查阅其中一些内容，尤其是附录 A（JavaScript 快速参考）和附录 B（介绍 JavaScript 调试技术）。

- 第一部分包含了 JavaScript 开发者经常忽视但又非常重要的 JavaScript 编码模式。这些编码模式和框架关系不大，但为了编写良好的 JavaScript 代码，它们不可或缺。务必从这一部分开始阅读。
- 第二部分主要讲述前面提到的“基本”任务，主要包括基本的 DOM 和 CSS 操纵，以及事件处理和定时器的使用。由于它们都很“基本”，所以我列出了所有主流框架下这些任务的实现代码，你可以根据需要来选择。最好结合附录 C 阅读这一部分，从而对这些主流框架有一个全面的认识，作出可靠的决策。
- 第三部分主要讲用户界面，尤其是视觉效果和简洁的 UI 理念：漂亮的 tooltip、光箱特效、图片预载入、无限滚动^②等。
- 第四部分是对第三部分的补充，主要讲绝大多数 Web 应用的重要组成部分——表单。此外，这部分会提到一系列用来辅助、简化和验证输入的工具。
- 第五部分主要讲客户端和服务端之间的关系，涉及的话题包括 cookie、JSON 以及 Ajax（域内和域间）。
- 第六部分是本书的最后一部分。这部分内容主要讲解如何利用混搭（mashup）思想和使用第三方服务。在这里，我选择了 3 个流行主题：Twitter 应用、Flickr 应用以及地理位置相关的 API。
- 附录 A 是我编写的一份 JavaScript 快速参考。其中总结了一些我认为很重要的 JavaScript 语言元素，并收录了一些实用的小贴士。希望它能对你有用。
- 附录 B 主要讨论如何调试 JavaScript。阅读这部分的所有内容是很有必要的，但愿它可以省去你在电脑前抓耳挠腮的大量时间，尤其是你使用 Internet Explorer 的时候。
- 附录 C 简要地介绍了本书收录的主流 JavaScript 框架。我已尽我所能地为这些框架逐一给出了精确的描述，并且展示出它们的特点，同时我还为如何选择 JavaScript 框架提供了一些建议。

① http://pragprog.com/titles/pg_js。

② 随着用户向下滚动页面，分批加载并显示更多内容。——编者注

- 附录 D 提供了一份快速参考，列出了有关 JavaScript 及其主流框架的帮助资源。这部分总结了附录 C 中的相关内容，并加入了一些额外资源（多数和 JavaScript 语言有关）。我把这部分放在本书的最后，以便查阅。

如何阅读本书

在本书中，每一个任务都编排成对开的两页：一页是文字解说，另一页则是代码。如果你阅读的是纸质版，这样的排版方式会让你感到流畅自然；如果你阅读的是电子版，请把阅读器设置成双页模式，以获得最佳的阅读效果。

致 谢

写书从来不是一件容易的事情。尽管技术书不需要复杂的情节，也不需要绞尽脑汁地来充实每页内容，但是迫于同行的压力和职业的责任，技术书的作者需要尽可能保证内容的准确和实用，从而为读者提供最佳的实践。因此，编写技术书必须付出巨大的努力，而技术书作者也会竭尽所能来保证书的质量。

在编写本书的过程中，我首先要感谢本书中提及并用到的框架的创造者们：一方面，他们的作品使得 Web 前端工作变得更加轻松；另一方面，他们对本书的严格审阅保证了本书的质量。在这里，我特别感谢 Sam Stephenson、Thomas Fuchs、John Resig、Alex Russell、Jack Slocum 等核心框架开发者。此外，我还要感谢 Prototype Core 团队的成员。他们不仅乐于助人，更是非常杰出的开发者，我从他们身上学到了很多，这里尤其要感谢 Andrew Dupont、Tobie Langel 和 Juriy Zaytsev。

为了提高本书的技术准确性和整体质量，很多人（一部分刚才已经提到了）仔细审阅过本书。感谢他们的杰出工作，使本书的质量得到保证！我特别要感谢的是 Dion Almaer、Arnaud Berthomier、Aaron Gustafson、Christian Heilmann、Dylan Schiemann 和 Sam Stephenson。

这是我与 Pragmatic Programmers 合作出版的第二本书^①。同以往一样，Dave Thomas 和 Andy Hunt 给了我这个宝贵的机会，使得我可以与 Pragmatic Guides 系列的优秀工作人员一起工作，他们是：Pragmatic Guides 系列编辑 Susannah Davidson Pfalzer、本书编辑 David McClintock（这是他在 Pragmatic Programmers 中的首次亮相）、眼光锐利的文字编辑 Kim Wimpsett、神奇的 Sara Lynn Eastler（她魔术般地完成了本书的索引）以及技术精湛的 Steve Peter（他精心排版了本书）。

最后，感谢我亲爱的妻子 Élodie。感谢她容忍我在过去五年中编写了四本书，并一直给予我支持和鼓励。她的爱让我感到自己是这个世界上最幸运的人，我无法想象没有她的生活将会是什么样子。所以，同以前一样，这本书为她而写。

^① 本书作者的第一本 Pragmatic Programmers 书的书名为 *Prototype and script.aculo.us: You Never Knew JavaScript Could Do This!*，该书的中文译本《Prototype 与 script.aculo.us 终极揭秘》已由电子工业出版社出版。——译者注

目 录

第一部分 JavaScript 必备操作

任务 1 动态选择方法及属性	2
任务 2 通过模块模式实现代码访问控制	4
任务 3 使用可选/可变/命名参数	6

第二部分 DOM、事件及定时器

任务 4 获得 DOM 元素的引用	10
任务 5 动态修饰内容	12
任务 6 修改元素的内容	14
任务 7 在 DOM 加载完成后运行脚本	16
任务 8 监听及停止监听事件	18
任务 9 利用事件委托	20
任务 10 将行为和自定义事件解耦	22
任务 11 模拟后台处理	24

第三部分 UI 技巧

任务 12 打造漂亮的 tooltip	28
任务 13 制作友好的弹窗	30
任务 14 预载入图片	32
任务 15 创造光箱特效	34
任务 16 实现“无限翻页”	36
任务 17 在载入内容时保持显示区域	38

第四部分 表单技巧

任务 18	暂时禁用提交按钮	42
任务 19	提供输入长度反馈	44
任务 20	同时选择或反选多个 checkbox	46
任务 21	表单验证：基本技巧	48
任务 22	表单验证：进阶技巧	50
任务 23	表单验证：高级技巧	52
任务 24	在表单中提供动态的帮助 tooltip	54
任务 25	自动完成输入	56
任务 26	使用动态多文件上传	58

第五部分 服务器端技术

任务 27	读取及写入 cookie	62
任务 28	通过 Ajax 载入内容（同域名）	64
任务 29	使用 JSON	66
任务 30	使用 JSON-P	68
任务 31	跨域“Ajax”（方法收集 1）	70
任务 32	跨域“Ajax”（方法收集 2）	72

第六部分 使用混搭

任务 33	Twitter 的同步更新	76
任务 34	Flickr 的同步更新	78
任务 35	获得地理位置及该位置的照片	80

第七部分 附录

附录 A	JavaScript 快速参考	84
附录 B	JavaScript 调试指南	91
附录 C	JavaScript 框架概览	104
附录 D	求助指南	112
参考文献		117

Part 1

第一部分

JavaScript 必备操作

序幕正式拉开。这部分介绍了 JavaScript 语言的必备操作，以便你在接下来的任务中大展拳脚。由于这部分中使用的都是纯粹的 JavaScript 语言，所以此处 3 个任务的代码不会依赖于任何框架或库。

- 通过任务 1 会学到如何动态访问对象的属性和方法（在代码里决定它们的名称）。
- 在任务 2 中，学习如何将代码控制在一个私有的作用域，从而避免“污染”别人的代码，同时也使得自己的代码具有自包含性。
- 最后，在任务 3 中，学习如何创建具有多样参数的函数。

别忘了，你可以在本书的网站上得到书中所有的源代码。你也可以直接在本书站点的演示页^①中直接访问它们。此外，请准备好一个简单的空页面（在接下来的章节中可能需要加载一些库或框架），打开 JavaScript 控制台，以便测试书中的代码。

^① 参见 <http://demos.pocketjavascript.com>。

任务 1 动态选择方法及属性

在实际工作中，我们经常会遇到这种情况：根据某个条件来调用两个方法^①中的一个，或是在两个属性^②中的一个上面进行读写操作。下面的代码展示了这种情形：

```
if (condition) {  
    myObj.method1(someArg);  
} else {  
    myObj.method2(someArg);  
}
```

JavaScript提供了一种简单的语法，即使用方括号操作符（[]）来动态地选择方法和属性。正如下面的代码所示，JavaScript有两种等价的成员访问语法（这个特征在动态语言里很常见）：

```
obj[expressionResultingInMembername] == obj.memberName
```

如果你曾用整数下标来访问数组中的某个元素，那你已经开始用方括号操作符来进行动态成员选择了。这是因为，数组对象本身就包含以数字下标命名的属性（以及length属性）。不过，JavaScript并不允许你使用点操作符（.）直接访问这些属性，因此myArray.0在语法上是非法的（太遗憾了，这本来是个挺酷的语法）。

为什么方括号操作符比点操作符表示法更强大呢？这是因为你可以在方括号中使用任何代表成员名称的内容来访问对象的成员。这些内容包括字面量、保存着成员名称的变量、名称组合（多数情况下是字符串的拼接）以及用三元操作符（condition ? valueIfTrue : valueIfFalse）实现的快速if/then选择。所有的这些内容都会被处理成一个字符串，然后JavaScript会用这个字符串来寻找对应的成员。

由于JavaScript中的函数本身也是对象，所以它可以像其他值一样被引用。如果一个表达式的结果是函数，你可以直接用括号操作符调用它，就像你直接用函数名称调用函数一样。

需要注意的是，如果你在向方法传递的参数上大量使用此类技巧，混乱的括号有可能会使代码变得难以阅读，此时使用常规的if/else结构更加明智。

① 方法是与对象相关联的函数。

② 属性是与对象相关联的变量。

使用方括号操作符

```
object['propertyName'] // => object.propertyName
object['methodName'](arg1) // => object.methodName(arg1)
```

切换行为

```
// 根据shouldBeVisible的值来调用show()或是hide()
element[shouldBeVisible ? 'show' : 'hide']();

// 避免IE中动画带来的巨大开销
// (假设这里已经有一个isIE变量)
element[isIE ? 'simpleEffect' : 'complexEffect']();
```

拼接方法名称

```
element[(enable ? 'add' : 'remove') + 'ClassName']('enabled');
```

请动手尝试下面代码

```
var love = { firstName: 'Élodie', lastName: 'Porteneuve' };
var useFirstName = true;
alert(love[useFirstName ? 'firstName' : 'lastName']); // => "Élodie"
```