



JavaScript

忍者秘籍

JavaScript Ninja

[美] John Resig 著
Bear Bibeault 著
徐涛 译

JavaScript

忍者秘籍

JavaScript
Ninja

[美] John Resig 著
Bear Bibeault
徐涛 译

人 民 邮 电 出 版 社
北 京

图书在版编目 (C I P) 数据

JavaScript忍者秘籍 / (美) 莱西格 (Resig, J.),
(美) 贝比奥特 (Bibeault, B.) 著; 徐涛译. -- 北京:
人民邮电出版社, 2015. 10
ISBN 978-7-115-40194-6

I. ①J… II. ①莱… ②贝… ③徐… III. ①JAVA语
言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第217250号

版 权 声 明

Original English language edition, entitled Secrets of the JavaScript Ninja by John Resig and Bear Bibeault published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, CT 06830. Copyright ©2013 by Manning Publications Co.

Simplified Chinese-language edition copyright ©2015 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Manning Publications Co.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。

-
- ◆ 著 [美] John Resig Bear Bibeault
 - 译 徐 涛
 - 责任编辑 陈冀康
 - 责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
 - 邮编 100164 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: 23.25
 - 字数: 497 千字 2015 年 10 月第 1 版
 - 印数: 1—4 000 册 2015 年 10 月北京第 1 次印刷
 - 著作权合同登记号 图字: 01-2009-3794 号
-

定价: 69.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315

内容提要

JavaScript 语言非常重要,相关的技术图书也很多,但没有任何一本书对 JavaScript 语言的最重要部分(函数、闭包和原型)进行深入、全面的介绍,也没有任何一本书讲述跨浏览器代码的编写。本书是 jQuery 库创始人编写的一本深入剖析 JavaScript 语言的书。

本书共分 4 个部分,从准入训练、见习训练、忍者训练和火影训练四个层次讲述了逐步成为 JavaScript 高手的全过程。全书从高级 Web 应用程序开发者所面临的挑战谈起,由浅入深地覆盖了测试和测试工具、函数、闭包、对象、正则表达式、定时器、运行时代码求值、with 语句、跨浏览器问题、元素的特性和属性、事件处理、DOM 以及 CSS 选择器等众多核心话题;将严谨的理论讲解和实用的代码示例相结合,引导读者更加深入地了解 JavaScript 的神奇,充分展示了 JavaScript 语言的各种特性。

本书适合具备一定 JavaScript 基础知识的读者阅读,也适合从事程序设计工作并想要深入探索 JavaScript 语言的读者阅读。

致谢

参与编写本书的人数会让大多数人感到意外。这一大本书（或在屏幕上阅读的电子书）的完成结合了许多编著者的共同努力。

Manning 的员工与我们共同努力，确保本书达到我们所期望的质量水平，我们非常感谢他们的付出。没有他们本书就不可能完成。本书的“结尾致谢”不仅包括我们的出版商 Marjan Bace、编辑 Mike Stephens，还包括以下编著者：Jeff Bleiel、Douglas Pudnick、Sebastian Stirling、Andrea Kaucher、Karen Tegtmeier、Katie Tennant、Megan Yockey、Dottie Marsico、Mary Piergies、Andy Carroll、Melody Dolab、Tiffany Taylor、Dennis Dalinnik、Gabriel Dobrescu 和 Ron Tomich。

要非常地感谢审稿专家，他们帮助我们为本书定稿，从寻找简单的错别字到纠正术语和代码中的错误，再到组织书中的章节。他们每个人都审查了整本书，极大地提高了本书的质量。非常感谢他们花时间审阅本书：Alessandro Gallo、André Roberge、Austin King、Austin Ziegler、Chad Davis、Charles E. Logston、Chris Gray、Christopher Haupt、Craig Lancaster、Curtis Miller、Daniel Bretoi、David Vedder、Erik Arvidsson、Glenn Stokol、Greg Donald、James Hatheway、Jared Hirsch、Jim Roos、Joe Litton、Johannes Link、John Paulson、Joshua Heyer、Julio Guijarro、Kurt Jung、Loïc Simon、Neil Mix、Robert Hanson、Scott Sauyet、Stuart Caborn 和 Tony Niemann。特别要感谢 Valentin Crettaz，他担任本书的技术编辑。除了检查多个环境中示例代码的每个例子，还对文本的技术准确性提出了宝贵的意见，找到了最初遗漏的信息，并及时更新了浏览器中的 JavaScript 和 HTML5 支持。

还要特别感谢 Bert Bates，他为改进本书提供了宝贵的反馈意见。花在 Skype 上的大量时间也得到了回报。

John Resig

我要感谢我的父母多年来的支持和鼓励。他们为我提供了很多资源和工具，激发了

我对编程的兴趣——他们一直都在鼓励我。

Bear Bibeault

关于我出版的第五本书，我想要感谢的人有很多，包括 javaranch.com 的成员和员工。不加入 JavaRanch，我就不会有机会开始写作，所以我衷心感谢 Paul Wheaton 和 Kathy Sierra 让我开始了这项工作，以及其他给予我鼓励和支持的工作人员，包括（但不限于）Eric Pascarello、Ernest Friedman Hill、Andrew Monkhouse、Jeanne Boyarsky、Bert Bates 和 Max Habibi。

同样要感谢和我住在一起的伙伴 Jay、宠物狗 Little Bear 和 Cozmo，它们在家里忍受寂寞，而我始终埋头写书。除了向它们抱怨一下 Word、某个浏览器或者任何引起我愤怒的事情外，我很少从键盘上抬起头。

最后，我要感谢我的合著者 John Resig。没有他，本书就不会存在。

作者简介

John Resig 是可汗学院计算机科学学院的院长，也是 jQuery JavaScript 库的创造者。排名前 10000 的网站中，目前有 57% 的网站使用 jQuery (BuiltWith.com 统计)，并且有几百万其他网站也使用 jQuery，它已成为最受欢迎的用于构建网站的技术，可能也是一直以来最流行的编程技术之一。

他还创建了许多其他开源工具和项目，包括 Processing.js (Processing 语言到 JavaScript 的转换)、QUnit (测试 JavaScript 代码的测试套件) 和 TestSwarm (用于分布式 JavaScript 测试的平台)。

他目前正在可汗学院进一步开展计算机科学教育，他在学校开发计算机科学课程和工具，来教各个年龄层的人如何编程。可汗学院的目标是创造优秀的免费教育资源。他不仅教人们如何编程，还让每一位程序员体验写完自己的第一个程序后的兴奋。

目前，John 住在纽约布鲁克林，业余时间喜欢研究 Ukiyo-e (日本版画)。

Bear Bibeault 编写软件已经超过 30 年，一开始是通过一个 100 波特的电传打字机在控制数据网络超级计算机上编写井字程序。因为 Bear 有两个电气工程学位，因此应该是从事设计天线之类的工作，但从他在数字设备公司从事的第一份工作开始，他总是更着迷于编程。

Bear 还分别在 Lightbridge Inc.、BMC Software、Dragon Systems、Works.com 等其他几个公司工作过。Bear 甚至曾在美国军队服役，教步兵学习如何炸毁坦克，这些技能在每日例会上还能派上用场。

Bear 目前担任一家家庭网关设备和电视机顶盒领域领先供应商的软件架构师。

Bear 是其他一些 Manning 图书的作者：《jQuery in Action》(第一版和第二版)、《Ajax in Practice》和《Prototype and Scriptaculous in Action》，他还是 O'Reilly 出版的许多网络 Head First 图书的技术复审员，如《Head First Ajax》、《Head Rush Ajax》和《Head First Servlets and JSP》。

除了日常工作外，Bear 还写书（duh!），并经营着一家小型企业，致力于创建 Web 应用程序，提供其他媒体服务（但不是婚礼摄像——永远不会是婚礼摄像），并作为“引领者”（非常资深的版主）帮着打理 CodeRanch.com。

不在电脑前待着时，Bear 喜欢做大餐、涉足摄影和视频，也喜欢驾驶雅马哈 V-Star 和穿热带印花衬衫。

他工作和居住在得克萨斯州奥斯汀，他深爱这座城市，除了满城的疯狂司机外。

译者序

在拿到这本书的英文稿时，我如获至宝，不到一个礼拜便全部阅读了一遍，因为这才是我心目中的“圣经”。

虽然在市面上，几乎每个礼拜都有 JavaScript 相关的新书上市，而且几乎每一本都有不同的侧重点，或是实战练习，或是框架讲解，但却没有一本书籍对 JavaScript 的核心部分进行深入的讲解。

2012 年的时候，我曾经零零散散整理了大约 50 篇关于 JavaScript 核心内容的文章，并发布到了博客园网站，这些文章虽然都有提及这些核心内容，但却没有很强的系统性。直到阅读完本书，我发现这才是我心目中的“圣经”，它不是规范的照搬，也不是示例的讲解，也不是 jQuery 的源码分析，而是作者结合开发 jQuery 时的经验以及其他专家多年来经验的总结。

本书没有对 JavaScript 的基本类型、操作符、对象等基本概念进行讲解，在介绍完基本的测试和调试知识后，就开始讲解 JavaScript 语言的最重要部分：函数、闭包以及原型。本书用了整整两章对函数进行讲解，从普通函数、匿名函数、递归函数以及函数表达式等的声明，到函数的调用，每个细节都深入地进行了讲解，同时穿插讲解了函数的缓存、重载、存储等方面的知识。在讲解闭包的时候，深入讲解了作用域和作用域链，通过对函数上下文的理解，我们又看到了偏应用函数、即时函数、函数包装等一系列与闭包相关的各种技巧和原理解释。而在原型讲解的时候，不仅讲解了基于原型和原型链的各种继承方式，同时也讲解了基于类（Class）风格的面向对象编程实现方式。

正则表达式、线程和定时器，这三项内容，不管在哪种语言学习中，都是最难以理解的部分。作者花了整整两章内容对其进行讲解，足以显示出对这些内容的重视，同时，作者在开发 jQuery 库的时候也是大量利用了这三项技术内容。

接下来的内容，是我在其他书上从来没有见过的，分别是代码求值、with 语句、跨浏览器策略、HTML 元素的 Attribute、Property、Class 等相关内容的深入讲解，这些内容是开发 jQuery 的基石，没有对这些内容的深入理解，作者也根本无法开发出这么受欢迎

迎的 jQuery 库来。

最重要的部分通常都是放在最后，事件、DOM 操作、CSS 选择器引擎，这三部分虽然不属于 JavaScript 语言规范本身的部分，但却是前端开发中最重要的部分了，不深入理解这三部分，基本上做不出太好的前端程序。在事件方面，除了讲解了不同浏览器的冒泡和委托机制，作者还讲述了自己 jQuery 库中是如何实现事件模块的，通过事件中枢管理系统提供一个高效的事件管理中心，从而对各种浏览器的各种事件做出实时的响应。在 DOM 操作章节，作者并没有提及 jQuery 中的 attr、html、text、addClass 等快捷方式的使用，而且通过各种基本知识，讲解如何实现复杂的 DOM 操作。在 CSS 选择器引擎方面，作者讲解了 4 种不同的选择器实现方式，通过逐一讲解，让我们了解到了每一种实现方式的优缺点，以及使用时的注意事项。

本书教授的是核心原理，而不是其他书中常见的用法示例。理解了原理，自然就可以一通百通，其他剩余的只不过是各种各样的技巧罢了。读完本书，再回过头来看看其他的 JavaScript 书籍，你的阅读体验自然就会完全不同了。

徐涛

2015 年 9 月

序

2008 年年初，在我开始编写本书时，我发现了一个真正的需求：没有任何一本书对 JavaScript 语言的最重要部分（函数、闭包和原型）进行深入、全面的介绍，也没有任何一本书讲述跨浏览器代码的编写。可惜的是，到目前为止情况并没有改善，真令人惊讶。

人们把越来越多的精力投入到了开发新技术上，例如源于 HTML5 的新技术或 ECMAScript 的新版本。但如果不能很好地理解 JavaScript 语言的基本特性，就无法深入地钻研新技术，或者使用最热门的库。虽然浏览器开发的未来是光明的，但事实上，大多数的开发都需要确保代码适用于大多数浏览器和大部分的潜在用户。

虽然本书酝酿了很长一段时间，但幸好还没过时。我的合著者 Bear Bibeault 对本书进行了很好的修正，他确保了本书的内容在很长一段时期内都很实用。

花这么长时间编写本书的主要原因是，我在不断地积累经验，为编写后面章节的跨浏览器代码做准备。我对跨浏览器开发的大部分了解是来自于 jQuery JavaScript 库的实践。在编写后面跨浏览器开发章节时，我意识到 jQuery 的大部分核心内容都可以用不同的方式编写、优化，从而用于更广泛的浏览器。

或许在编写完本书以后，jQuery 的最大改变是从使用浏览器特征判断转化为功能特性判断。这可以让 jQuery 进行无限制使用，而无需假设浏览器总是有特定的 bug 或遗失某些特性。

这些变化的结果是，jQuery 预料到浏览器在过去几年经历的许多改进：谷歌发布了 Chrome 浏览器；由于移动计算越来越流行，用户代理的数量激增；Mozilla、谷歌和苹果已经进入了浏览器性能之争；微软终于开始对 Internet Explorer 进行实质性的改进。不能再假定浏览器渲染引擎（如 WebKit 或 Internet Explorer 中的 Trident）总是按同一种方式做事。技术在发生着快速的变化，也被越来越多的用户所使用。

使用本书所讲述的技术，jQuery 的跨浏览器功能能够保证，用 jQuery 编写的代码将能够在各种浏览器环境中运行。这使得 jQuery 在过去 4 年中高速增长，根据 BuiltWith.com

的记录，在排名前 10000 家的网站中，现在有超过 57% 的网站使用 jQuery。

JavaScript 的一些相对不变的功能，例如代码求值、有争议的 with 语句和计时器，其使用方式仍然很有趣。现在有许多很积极的编程语言基于 JavaScript 或者可以编译成 JavaScript，如 CoffeeScript 和 Processing.js 等。这些语言需要复杂的语言解析、代码求值和作用域操作，以实现有效运行。虽然动态代码求值因其复杂性和潜在的安全问题而一直遭到诋毁，但如果没有它，我们就不会有 CoffeeScript 这门编程语言，该语言影响了即将到来的 ECMAScript 规范本身。

我个人正在使用所有这些功能，即使是现在在可汗学院工作的时候也是一样。浏览器中的动态代码求值是一个非常强大的功能：可以构建浏览器内编程环境，做一些疯狂的事情，例如将代码注入到实时运行环境中。这种功能将为学习计算机编程带来一个非常吸引人的方式，它所提供的新功能是传统的学习环境所无法提供的。

浏览器开发的未来仍然非常光明，这主要是因为很多功能都被封装在 JavaScript 和浏览器 API 中了。准确掌握 JavaScript 语言的最重要部分，并且积极编写能够在很多浏览器中都能运行的代码，有助于创建优雅、快速、普适的代码。

前言

JavaScript 非常重要。这并不总是如此，但现在确实如此。

Web 应用程序将为用户提供丰富的用户界面体验，没有 JavaScript，我们也就只能显示几张小图片。Web 开发人员比以往任何时候都更需要熟练掌握能够为 Web 应用程序带来生命的语言。

像橙汁和早餐一样，JavaScript 不再只用于浏览器了。该语言推倒了浏览器的界限，可用在 Rhino 和 V8 等引擎的服务器上，也可以用于像 node.js 这样的框架。

虽然本书主要是介绍用于 Web 应用程序的 JavaScript，但本书第二部分提出的 JavaScript 基础适用范围也很广泛。

随着使用 JavaScript 的开发人员在逐渐增多，熟练掌握 JavaScript 基础比以往任何时候都更加重要，这样他们就可以成为真正的语言忍者。

目标读者

这本书不能作为你的第一本 JavaScript 图书。如果你是 JavaScript 完全新手，或者只是通过搜索网络获得代码片段了解了一些语句，那么本书还不适合你阅读。

本书面向至少掌握 JavaScript 基础的 Web 开发人员。你应该理解 JavaScript 语句的基本结构以及创建简单页面脚本的工作原理。你不需要是该语言的高级用户——这是本书的目标，但你不应是一位新手。

你还应该有 HTML 和 CSS 方面的工作经验。再次强调，不需要了解特别高级的知识，但你应该知道拼凑 Web 网页的基础知识。

如果你想获得一些好的必备资料，寻找一本关于 JavaScript 和 Web 开发方面的流行图书，然后解决这个问题。我们推荐的 JavaScript 图书：David Flanagan 的《JavaScript 权威指南》、Douglas Crockford 的《JavaScript 语言精粹》以及 Michael Morrison 的 *Head First JavaScript*。

路线图

本书旨在通过四个部分让你从学徒升级为忍者。

第一部分介绍了我们往下要学习的主题和所需要的一些工具。

第二部分主要关注 JavaScript 基础：该语言中你习以为常但不太清楚其工作原理的部分。这可能是本书中最重要的部分，即使你只阅读了该部分，也会对 JavaScript 语言有更深入的理解。

在第三部分，我们利用在第二部分中学到的基础知识解决浏览器留给我们的棘手问题。

第四部分总结本书内容，主要讨论从创建 jQuery 等高级 JavaScript 库中学到的东西。让我们看一下每一章的简要内容。

第 1 章介绍我们作为高级 Web 应用程序作者所面临的挑战。提出浏览器扩散引起的一些问题，并针对我们在开发包括测试和性能分析在内的应用程序时应该遵循的最佳实践给出建议。

第 2 章讨论测试，了解测试和测试工具的当前状态。还介绍一个小而强大的测试概念——断言，我们将在本书的其余部分大量使用它，以确保代码完成我们希望它完成的事情（有时是为了证明它不应该这样做！）。

有了这些工具，第 3 章开始我们开始涉足语言基础，首先要彻底研究 JavaScript 中的函数（function），这可能会出乎大家意外之外。虽然你可能希望对象（object）是首先关注的内容，但 object 是对函数的深入理解，并且 JavaScript 是一种函数式语言，其可以让我们从普通的 JavaScript 程序员开始转变成 JavaScript 忍者！

第 4 章还是继续介绍函数，将把在第 3 章中学到的基础知识应用于我们在创建应用程序过程所面临的问题上。我们将探讨递归，不只是因为它本身的含义，而是因为我们可以通过检测递归学到更多关于函数的知识——我们将学习如何利用 JavaScript 的函数式编程特性让代码变得优雅，同时更加强大和简洁。我们将学习如何处理可变参数列表，以及在非原生支持方法重载的面向对象概念的语言中重载函数的方式。

第 5 章是大家可以本书中学到的最重要概念之一的主题：闭包。函数式编程中的关键概念——闭包，允许我们对在程序中声明和创建的对象作用域进行细粒度控制。这些作用域的控制是 JavaScript 忍者编写代码的关键因素。即使不阅读本章后面的内容（但我们希望大家不要这样做），编程水平也会比刚开始学习时要好很多。

第 6 章解决对象问题，通过函数的 prototype 属性，我们可以学习创建对象的模式，我们将学习如何将对象绑定到它们定义的函数上——这是我们首先讨论函数的其中原因之一。

第 7 章主要是正则表达式，正则表达式是该语言常常被忽视的一项功能，如果正确使用，它可以完成很多行代码完成的工作。我们将学习如何构建并使用正则表达式，以

及如何利用正则表达式及其相关方法优雅地解决一些反复出现的问题。

第二部分语言基础将在第 8 章结束,在本章我们将学习定时器和间隔定时器是如何在 JavaScript 的单线程特性中工作的。HTML5 承诺使用 Web workers 为我们减少单线程限制,但大多数浏览器还无法做到,几乎所有现有的 JavaScript 代码都取决于对 JavaScript 单线程模型的深入理解。

第三部分从第 9 章开始,我们打开 JavaScript 运行时代码求值的黑盒子。我们来了了解求值代码的各种方式,包括在我们的选择范围内怎样做是安全的。我们将讨论几个实例,比如 JSON 求值、元语言(又名领域特定语言)、压缩和混淆,甚至是面向方面编程。

在第 10 章中,我们将研究有争议的 with 声明,用于在作用域内缩短引用。无论你是 with 爱好者还是批评者,它都存在于大量的代码中,不管你认为它是炸弹还是令人憎恶的事物,都应该要理解它。

第 11 章的主题是处理跨浏览器问题。我们将研究 5 个与这些问题有关的关键开发问题:浏览器差异、错误和 bug 修复、外部代码和标记、缺失的功能以及回归。详细讨论特性模拟和对象检测等策略,以帮助我们应对这些跨浏览器挑战。

第 12 章的重点是处理元素特性(attribute)、属性(property)和样式(style)。虽然各种浏览器在如何处理这些元素方面的差异随着时间的过去会慢慢融合,但仍然有许多棘手的问题,本章将讲述如何解决这些问题。

第 13 章总结了第三部分内容,深入研究浏览器中的事件处理,以及创建能够以浏览器不可知方式处理事件的统一子系统方式。这包括给浏览器添加其无法提供的功能,如自定义事件和事件委托。

在第四部分中,我们加快速度,深入研究由 jQuery 等 JavaScript 库的核心内容引出的主题。第 14 章将讨论如何构建 DOM 操作 API,在运行时操作文档对象模型,包括向 DOM 注入新元素的难题。

最后,在第 15 章中,我们将讨论如何构建 CSS 选择器引擎以及对选择器进行解析和求值的不同方式。这一章不适合胆小者,但可以用它来测验你是否能成为忍者。

代码规范

清单或文本中的所有源代码都是采用固定宽度字体,将其与普通文本分离。文本中的方法和函数名、属性、XML 元素和特性也采用相同的字体。

在某些情况下,原始源代码被重新格式化来适应页面。一般来说,编写原始代码时要考虑页面宽度限制,但有时你会发现本书中的代码和源下载提供的代码之间的格式有一点不同。在极少数情况下,不改变其意义,长长的代码行无法重新格式化,本书清单包含行接续符标记。

代码注释伴随有许多列表,用于突出重要概念。在许多情况下,数字编号对应着文

本中的解释。

源码下载

本书中所有运行示例的源代码（以及一些从未进入文本的额外代码）可以在本书的网页 www.manning.com/SecretsoftheJavaScriptNinja 上下载。

本书的代码示例按章节分类，每一章有单独的文件夹。布局由本地 Web 服务器来完成，如 Apache HTTP 服务器。简单将下载的代码解压缩到所选择的文件夹，并将该文件夹设置成为应用程序的根目录。

除了少数例外，大多数例子都不需要使用 Web 服务器，可以直接加载到浏览器中执行。

所有的例子都在各种现代浏览器中测试过（2012 年中期），包括 Internet Explorer 9、Firefox、Safari 和 Google Chrome。

在线交流

作者和 Manning 出版社邀请你访问 Manning 出版社运营的本书论坛，在那里你可以评论本书，询问技术问题，并获得作者和其他用户的帮助。访问和订阅论坛，在浏览器中登录 www.manning.com/SecretsoftheJavaScriptNinja，然后单击作者在线链接。这个页面提供有关如何注册登录论坛、可以获得哪些帮助以及论坛行为规则的信息。

Manning 的承诺，是为读者提供一个场所，让各位读者以及读者和作者之间进行有意义的交流。Manning 不要求作者的参与次数，对本书论坛的贡献仍是自愿的（无报酬）。我们建议大家试着询问作者一些具有挑战性的问题，以免他们转移兴趣！

只要本书在发售，在线交流论坛和先前讨论的帖子都可以在出版商的网站上进行访问。

关于封面插图

本书封面上的图像是“能乐剧演员，武士”，19 世纪中期一位不知名的日本艺术家制作的木刻板画。Noh（能乐剧）衍生自日语：天赋或技能，是从 14 世纪就开始出现的一种经典音乐剧。许多人物都戴着面具，男性扮演男性和女性角色。日本数百年来的英雄人物、武士，经常出现在表演中。在该版画中，精致的服装和威武的武士展示了这位艺术家的精湛技术。

武士和忍者都是日本艺术作品中善于打斗的勇士，他们非常勇敢和精明。武士是精英士兵，受过良好教育，文武双全，武士道是他们必须严格遵守的原则，自 10 世纪开始代代相传。他们来自贵族和上层阶级，相当于欧洲骑士。战争时，大批武士穿着精致的盔甲和彩色的服装，震慑住对方。忍者是靠武术技能选出，而不是社会地位或教育。他们穿着黑色服装，蒙面，单独或一小组出动，以诡计和隐身攻击敌人，千方百计完成

任务，他们唯一的编码是保密的。

封面插图是由 Manning 的一位编辑多年来收集到的三个日本人物版画中的一个，当我们在为本书寻找忍者封面时，这幅引人注目的武士版画受到了我们的关注，它的细节刻画得很精致，色彩鲜明，生动地描绘了一位威武的战士志在必得的情形。

有时候我们很难区分两本计算机图书，Manning 非常有创意，用 200 年前的插图做计算机图书的封面，这些插图描绘了世界各地的丰富多样的传统服装，印刷在封面上为其重新带来活力。