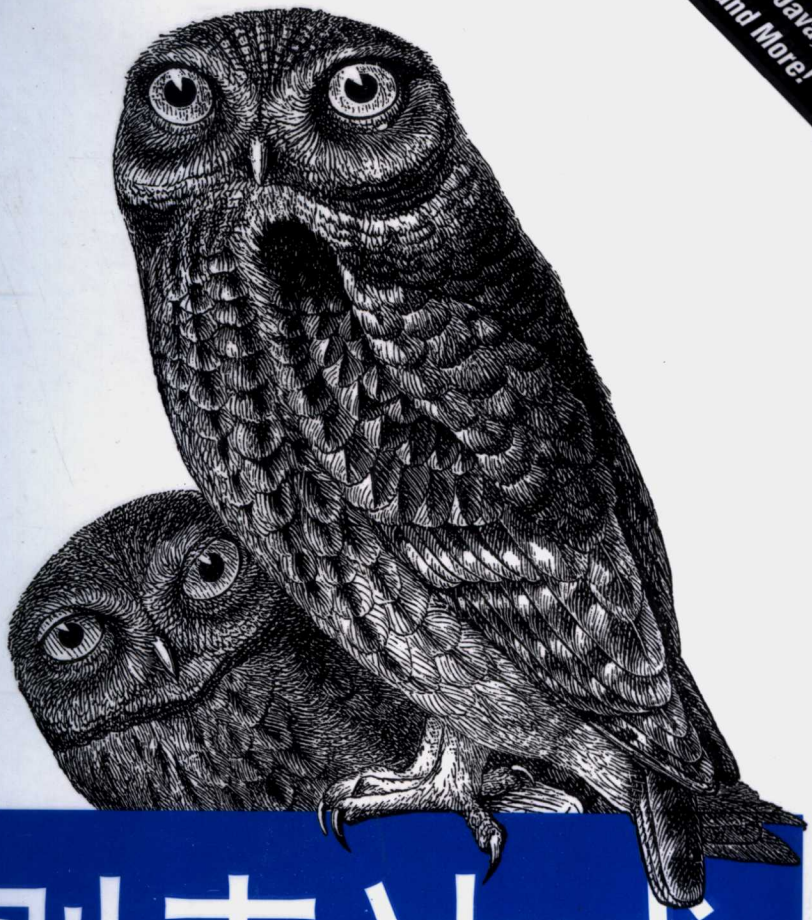
Mastering Regular Expressions

Understand Your Data and Be More Productive

第3版

For Perl, PHP, Java, .NET, Ruby, and More!



精通

正则表达式

Jeffrey E.F. Friedl 著

余晟 译

O'REILLY®

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

精通正则表达式（第3版）

Mastering Regular Expressions, 3rd Edition

[美] Jeffrey E.F. Friedl 著

余晟 译

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

O'Reilly Media, Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权电子工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在Unix、X、Internet和其他开放系统图书领域具有领导地位的出版公司，同时也是在线出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为20世纪最重要的50本书之一)到GNN(最早的Internet 门户和商业网站)，再到 WebSite（第一个桌面 PC 的Web服务器软件），O'Reilly Media, Inc. 一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

推荐序

一夫当关

IT 产业新技术日新月异，令人目不暇接，然而在这其中，真正能称得上伟大的东西却寥寥无几。1998 年，被誉为“软件世界的爱迪生”，发明了 BSD、TCP/IP、csh、vi 和 NFS 的 SUN 首席科学家 Bill Joy 曾经不无调侃地说，在计算机体系结构领域里，缓存是唯一能称得上伟大的思想，其他的一切发明和技术不过是在不同场景下应用这一思想而已。在计算机软件领域里，情形也大体相似。如果罗列这个领域中的伟大发明，我相信绝不会超过二十项。在这个名单当中，当然应该包括分组交换网络、Web、Lisp、哈希算法、UNIX、编译技术、关系模型、面向对象、XML 这些大名鼎鼎的家伙，而正则表达式也绝对不应该被漏掉。正则表达式具有伟大技术发明的一切特点，它简单、优美、功能强大、妙用无穷。对于很多实际工作来讲，正则表达式简直是灵丹妙药，能够成百倍地提高开发效率和程序质量。CSDN 的创始人蒋涛先生在早年开发专业软件产品时，就曾经体验过这一工具的巨大威力，并且一直印象深刻。而我的一位从事网络编辑工作的朋友，最近也领略了正则表达式的威力——他用 Perl 开发了一个不足 20 行的小程序，使用正则表达式将一项原本每天耗用 10 人时的工作在一分钟之内自动完成。而正则表达式在生物信息学和人类基因图谱的研究中所发挥的关键作用，更是被传为佳话。无论对于软件开发者，还是从事其他知识工作的专业人士，正则表达式都是最有利的工具之一。

所谓正则表达式，就是一种描述字符串结构模式的形式化表达方法。在发展的初期，这套方法仅限于描述正则文本，故此得名“正则表达式（regular expression）”。随着正则表达式研究的深入和发展，特别是 Perl 语言的实践和探索，正则表达式的能力已经大大突破了传统的、数学上的限制，成为威力巨大的实用工具，在几乎所有主流语言中获得支持。为什么正则表达式具有如此巨大的魅力？一方面，因为正则表达式处理的对象是字符串，或者抽象地说，是一个对象序列，而这恰恰是当今计算机体系的本质数据结构，我们围绕计算机所做的大多数工作，都归结为在这个序列上的操作，因此，正则表达式用途广阔。另一方面，与大多数其他技术不同，正则表达式具有超强的结构描述能力，而在计算机中，正是不同的结构把无差别的字节组织成千差万别的软件对象，再组合成为无所不能的软件系统，因此，描述了结构，就等于描述了系统。在这方面，正则表达式的地位是独特的。正因为这两点，在现在的软件开发和日常数据处理工作中，正则表达式已经成为必不可少的工具。如果一个开发工具不支持正则表达式，那它就会被视为玩具语言，如果一个编辑器

不支持正则表达式，那它就会被成为阳春应用。连人们原本并不指望应用正则表达式的商用数据库，各家厂商也竞相以支持正则表达式为卖点。正则表达式的声势之隆，是毋庸置疑的。

非常奇怪的是，这样一个了不起的技术，在我国却并没有得到充分推广。以其价值而言，正则表达式不但值得每一个专业程序员掌握，而且值得所有知识工作者去了解。然而现实情况是，不但一般知识工作者大多闻所未闻，很多专业程序员也视之为畏途。为什么会出现这种情况呢？原因有二。其一，正则表达式产生和发展在 UNIX 文化体系之中，而我国软件开发社群的知识结构长期受到微软的决定，UNIX 文化影响甚微。在 2002 年推出.NET 平台之前，微软在其各项主流平台、产品与开发工具当中，均未对正则表达式给予足够的重视，相应地，我们的开发者们对正则表达式也就知之不多。第二，也是更重要的原因，就是正则表达式并不是那么好掌握的，在通向驾驭正则表达式强大力量的道路上，还是有那么几只拦路虎的，而要打虎过岗，不但要花点功夫，还要有正确的方法。

学习正则表达式，入门不难，看一些例子，试着模仿模仿，就可以粗通，并且在工作中解决不少问题。然而大部分学习者也就就此止步，他们对自己说："正则表达式不过如此，我就学到这里了，以后现用现学就行了。"他们以为自己可以像学习其他技术一样，在实践中逐渐提高正则表达式的应用水平。然而事实上，正则表达式并不是每天都会用到，而其密码般的形象，随着时间的推移很容易被忘记，所以经常发生的情况是，开发者对于正则表达式的记忆迅速消褪，每次遇到新的问题，都要查资料，重新唤回记忆，对于稍微复杂一点的问题，只好求助于现成的解决方案。反反复复，长期如此，不但应用水平难以明显提升，而且会对这项技术逐渐产生一定的恐惧感和厌烦情绪。这还只是应用阶段，正则表达式应用的高级阶段，要求开发者还必须充分理解正则表达式的能力范围，能够将一些正则表达式技术组合应用，达成超乎一般想像的效果。为了高效、正确地解决实际问题，有的时候甚至要求深入理解正则表达式的原理，甚至对于如何实现正则表达式引擎都要有所了解，在此基础上，规避陷阱，优化设计，提高程序执行效率。要达到这样的程度，不经过系统的学习是不可能的。

系统学习正则表达式并不是一件容易的事情，仅仅通过阅读一些"HOW TO"的快餐式文章是不行的，必须有更完整、更系统的资料指导学习。如果你在国外技术社区里询问如何才能系统学习正则表达式，几乎所有的领域专家都会向你推荐一本书——Jeffrey Friedl 的《精通正则表达式》，也就是本书。

这本《精通正则表达式》是系统学习正则表达式的唯一最权威著作。可以说，在今天，如果想理解和掌握正则表达式，想要建立关于这一技术的完整概念体系，想充分发挥其巨大能量，这本书几乎是无法绕开的必经之路。甚至可以说，如果你没有读过这本书，那么你

对于正则表达式的理解和应用能力一定达不到升堂入室的程度。本书第 1 版出版于十年之前，自那时起它就成为正则表达式领域最全面、最受欢迎的代表著作，数以万计的读者通过这本书掌握了正则表达式，成为行家里手。在任何时候，任何地方，只要提到正则表达式著作，人们都会提到这本书。这本书的质量之高，声誉之盛，使得几乎没有人企图挑战它的地位，从而在正则表达式图书领域形成独特的“一夫当关”的局面，称其为正则表达式圣经，绝对当之无愧。

为什么这本书能够表现得如此出色？我认为这其中有三个原因。其一，作者本人具有多年程序开发经验，理论基础深厚，实战经验丰富，对正则表达式这个主题透彻理解，因此在技术上得心应手，底气十足，对于技术上的难点不回避、不含糊。作者高超的技术水平是本书质量的强大保证。其二，作者思路对头，素材组织得当，用例丰富。正则表达式根植于数学理论，却又能在日常俗事上发挥巨大的效用。写这种类型的技术，思路稍微一偏差，就可能走歪路，不是太理论，就是太琐碎，不是太枯燥，就是太浅薄，实在很难把握。作者清楚地认识到，这本书的读者不是计算机科学家，但也不是满足于“知其然而不知其所以然”的快餐式代码小子，而是具有一定理论素养，却又始终以实践为本的专业开发者。他们需要的是面向实践的理论和思想，是实实在在的实战能力，只有满足这种需要，才能够真正打动读者。通读此书，可以说作者对这一路线的把握十分成功，保证了内容大方向的正确。其三，这本书的写法独具匠心，堪称典范。技术图书的主要使命是传播专业知识。而专业知识分为框架性知识和具体知识。框架性知识需要通过系统的阅读和学习掌握，而大量的具体知识，则主要通过日常工作的积累以及随用随查的的学习来逐渐填充起来。本书前六章，以顺序式记述的方式，将正则表达式的系统知识娓娓道来，读者像看故事书似的就建立起整个正则表达式的基本知识体系。而后面的内容，则是方便实际开发中频发查阅之用，包括各大主流语言对正则表达式的支持细节，包含有大量案例。这样的写法，完全符合一般人学习的特点，因此书读起来非常惬意，非常有趣，用的时候查起来又非常方便。这样的著述风格，实在值得学习。

读者可以在没有任何正则表达式的基础上开始阅读此书，只要勤动脑，加强理解，适当动手练习，将能够在不长的时间里掌握正则表达式的思想和技术精华，这一点已经被很多人验证过，我本人也是这本书的受益者之一。正因为这本书独一无二的地位和高度的可读性，也因为正则表达式作为一项了不起的技术发明所具有的巨大威力，我非常希望更多的读者能够通过认真地学习本书而掌握这一强大技术，并享受这项技术带来的快乐。

孟岩

2007 年 7 月于北京

译者序

《精通正则表达式（第 3 版）》（即 Mastering Regular Expression，3rd Edition）是一本好书。

我还记得，自己刚开始工作时，就遇到了关于正则表达式的问题（从此被逼上梁山）：若从文本中抽取 E-mail 地址，还可以用字符串来查找（先定位到@，然后向两端查找），若要抽取 URL，简单的文本查找就无能为力了。正当我一筹莫展之时，项目经理说："可以用正则表达式，去网上找找资料吧。"抱着这根救命稻草，我搜索了之前只是听说过名字的正则表达式的资料，并打印了 `java.util.regex`（开发用的 Java）的文档来看。摸索了半天，我的感觉就是，这玩意儿，真神奇，真复杂，真好用。

此后，用到正则表达式的地方越来越多，我也越来越感觉到它的重要，然而使用起来却总感觉捉襟见肘。当时是夏天，北京非常热，我决定下班之后不再着急赶车回家，而是在公司安心看看技术文档，于是邂逅了这本 *Mastering Regular Expression*。该书原文是相当通畅易懂的，看完全书大概花了我一周的业余时间，之后便如拨云见日，感觉豁然开朗——原来正则表达式可以这样用，真是奇妙，令人拍案叫绝。

此后我运用正则表达式便不用再看什么资料了，尤其量就是查查语言的具体文档，表达式的基本模型和思路，完全是在阅读本书时确立的。也正是因为细心阅读过本书，所以有时我能以正则表达式解决某些复杂的问题。我的朋友郝培强（Tinyfool，昵称 Tiny）曾问过我这样一个正则表达式的问题：在 Apache 服务器的 Rewrite 规则中，怎样以一个正则表达式匹配"除两个特定子域名之外的所有其他子域名"，其他人的办法都无法满足要求：要么只能匹配这两个特定的子域名，要么必须依赖程序分支才能进行判断。其实这个问题，是可以用一个正则表达式匹配的。事后，Tiny 说，看来，会用正则的人很多，但真正懂得正则的人很少。现实情况也确实如此，就我所见，不少同仁对正则表达式的运用，大多是从网上找些现成的表达式，套用在自己的程序中，但对到底该用几个反斜线转义，转义是在字符串级别还是表达式级别进行的，捕获型括号是否必须，表达式的效率如何，等等问题，往往都是一知半解，甚至毫无概念，在 Tiny 的问题面前，更是束手无策，一筹莫展。

就我个人来说，我所掌握的正则表达式的知识，绝大多数来自本书。正是依靠这些知识，我几乎能以正则表达式进行自己期望的任何文本处理，所以我相信，能够耐心读完这本书的读者，一定能深入正则表达式的世界，若再加以练习和思考，就能熟练地依靠它解决各种复杂的问题（其中就包括类似 Tiny 的问题）了。

去年，通过霍炬（Virushuo）的介绍，我参加了博文视点的试译活动，很幸运地获得了翻译本书的机会。有机会与大家分享这样一本好书，我深感荣幸。500 多页的书，拖拖拉拉，也花了半年多的时间。虽然之前读过原著，积累了一些运用正则表达式的经验，也翻译过数十万字的资料，但要尽可能准确、贴切地传达原文的阅读感觉，我仍感颇费心力。部分译文在确认理解原文的基础上，要以符合中文习惯的方式加以表述仍然颇费周折（例如，直译的“正则表达式确实容许出现这种错误”，原文的意思是“这样的错误超出了正则表达式的能力”，最后修改为“出现这样的错误，不能怪正则表达式”或“这样的问题，错不在正则表达式”）。另有部分词语，虽可译为中文，但为保证阅读的流畅，没有翻译（例如，“它包含特殊和一般两个部分，特殊部分之所以是特殊的，原因在于……”，此处 special 和 normal 是专指，故翻译为“它包含 special 和 normal 两个部分，special 部分之所以得名，原因在于……”），这样的处理，相信不会影响读者的理解。

在本书翻译结束之际，我首先要感谢霍炬，他的引荐让我获得了翻译这本书的机会；还要感谢博文视点的周筠老师，她谨慎严格的工作态度，时刻提醒我不能马虎对待这本经典之作；还有本书的责编晓菲，她为本书的编辑和校对做了大量细致而深入的工作。

另外我还要感谢东北师范大学文学院的王确老师，在我求学期间，王老师给予我诸多指点，离校时间愈长，愈是怀念和庆幸那段经历，可以说，没有与他的相识，便没有我的今天。

翻译过程中，我虽力求把握原文，语言通畅，但翻译中的错误或许是在所难免的，对此本人愿负全部责任。希望广大读者发现错误能及时与我和出版社联系以便重印时修正，或是以勘误的形式公布出来以惠及其他读者。如果读者有任何想法或建议，欢迎给我写信，我的邮件地址是：*yusheng.regex@gmail.com*。

如今正则表达式已经成为几乎所有主流编程语言中的必备元素：Java、Perl、Python、PHP、Ruby……莫不如此，甚至功能稍强大一些的文本编辑工具，都支持正则表达式。尤其是在 Web 兴起之后，开发任务中的一大部分甚至全部，都是对字符串的处理。相比简单的字符串比较、查找、替换，正则表达式提供了强大得多的处理能力（最重要的是，它能够处理“符合某种抽象模式”的字符串，而不是固化的、具体的字符串）。熟练运用它们，能够节省大量的开发时间，甚至解决一些之前看来是 mission impossible 的问题。

本书是讲解正则表达式的经典之作。其他介绍正则表达式的资料，往往局限于具体的语法和函数的讲解，于语法细节处着墨太多，忽略了正则表达式本身。这样，读者虽然对关于正则表达式的具体规定有所了解，但终究是只见树木不见森林，遇上复杂的情况，往往束手无策，举步维艰。而本书自第 1 版开始便着力于教会读者“以正则表达式来思考（think regular expression）”，向读者讲授正则表达式的精髓（正则表达式的各种流派、匹配原理、优化原则，等等），而不拘泥于具体的规定和形式。了解这些精髓，再辅以具体操作的文档，

读者便可做到“胸中有丘壑，下笔如有神”；即便问题无法以正则表达式来解决，读者也能很快作出判断，而不必盲目尝试，徒费工夫。

不了解正则表达式的读者，可循序渐进，依次阅读各章，即便之前完全未接触过正则表达式，读过前两章，也能在心中描绘出概略的图谱。第 3、4、5、6 章是本书的重点，也是核心价值所在，它们分别介绍了正则表达式的特性和流派、匹配原理、实用诀窍以及调校措施。这样的知识与具体语言无关，适用于几乎所有的语言和工具（当然，如果使用 DFA 引擎，第 6 章的价值要打些折扣），所谓“大象无形”，便是如此。读者如能仔细研读，悉心揣摩，之后解决各种问题时，必定获益匪浅。第 7、8、9、10 章分别讲解了 Perl、Java、.NET、PHP 中正则表达式的用法，看来类似参考手册，其实是对前面 4 章知识的包装，将抽象的知识辅以具体的语言规定，以具体的形式表现出来。所以，心急的读者，在阅读这些章节之前，最好先通读第 3、4、5、6 章，以便更好地理解其中的逻辑和思路。

相信仔细阅读完本书的读者，定会有登堂入室的感觉。不但能见识到正则表达式各种令人眼花缭乱的特性，更能够深入了解表达式、匹配、引擎背后的原理，从而写出复杂、神奇而又高效的正则表达式，快速地解决工作中的各种问题。

余晟

2007 年 6 月于北京

前言

Preface

本书关注的是一种强大的工具——“正则表达式”。它将教会读者如何使用正则表达式解决各种问题，以及如何充分使用支持正则表达式的工具和语言。许多关于正则表达式的文档都没有介绍这种工具的能力，而本书的目的正是让读者“**精通**”正则表达式。

许多种工具都支持正则表达式（文本编辑器、文字处理软件、系统工具、数据库引擎，等等），不过，要想充分挖掘正则表达式的能力，还是应当将它作为编程语言的一部分。例如Java、JScript、Visual Basic、VBScript、JavaScript、ECMAScript、C、C++、C#、elisp、Perl、Python、Tcl、Ruby、PHP、*sed* 和 *awk*。事实上，在一些用上述语言编写的程序中，正则表达式扮演了极其重要的角色。

正则表达式能够得到众多语言和工具的支持是有原因的：它们极其有用。从较低的层面上来说，正则表达式描述的是一串文本（a chunk of text）的特征。读者可以用它来验证用户输入的数据，或者也可以用它来检索大量的文本。从较高的层面上来说，正则表达式容许用户掌控他们自己的数据——控制这些数据，让它们为自己服务。掌握正则表达式，就是掌握自己的数据。

本书的价值

The Need for This Book

本书的第 1 版写于 1996 年，以满足当时存在的需求。那时还没有关于正则表达式的详尽文档，所以它的大部分能力还没有被发掘出来。正则表达式文档倒是存在，但它们都立足于“低层次视角”。我认为，那种情况就好像是教一些人英文字母，然后就指望他们会说话。

第 2 版与第 1 版间隔了五年半的时间，这期间，互联网迅速流行起来，正则表达式的形式也有了极大的扩张，这或许并不是巧合。几乎所有工具软件和程序语言支持的正则表达式也变得更加强大和易于使用。Perl、Python、Tcl、Java 和 Visual Basic 都提供了新的正则支持。新出现的支持内建正则表达式的语言，例如 PHP、Ruby、C#，也已经发展壮大，流行开来。在这段时间里，本书的核心——如何真正理解正则表达式，以及如何使用正则表达式——仍然保持着它的重要性和参考价值。

不过，第 1 版已经逐渐脱离了时代，必须加以修订，才能适应新的语言和特性，也才能对应正则表达式在互联网世界中越来越重要的地位。第 2 版出版于 2002 年，这一年的里程碑是 `java.util.regex`、Microsoft .NET Framework 和 Perl 5.8 的诞生。第 2 版全面覆盖了这些内容。关于第 2 版，我唯一的遗憾就是，它没有提及 PHP。自第 2 版诞生以来的 4 年里，PHP 的重要性一直在增加，所以，弥补这一缺憾是非常迫切的。

第 3 版在前面的章节中增加了 PHP 的相关内容，并专门为理解和应用 PHP 的正则表达式增加了一章全新的内容。另外，该版对 Java 的章节也进行了修订，做了可观的扩充，反映了 Java1.5 和 Java1.6 的新特性。

目标读者

Intended Audience

任何有机会使用正则表达式的人，都会对本书感兴趣。如果您还不了解正则表达式能提供的强大功能，这本书展示的全新世界将会让您受益匪浅。即使您认为自己已经是掌握正则表达式的高手了，这本书也能够深化您的认识。第 1 版面世后，我时常会收到读者的电子邮件反映说“读这本书之前，我**以为**自己了解正则表达式，但**现在**我才真正了解”。

以与文本打交道为工作（如 Web 开发）的程序员将会发现，这本书绝对称得上是座金矿，因为其中蕴藏了各种细节、暗示、讲解，以及能够立刻投入到实用中的知识。在其他任何地方都难以找到这样丰富的细节。

正则表达式是一种思想——各种工具以各种方式（数目远远超过本书的列举）来实现它。如果读者理解了正则表达式的基本思想，掌握某种特殊的实现就是易如反掌的事情。本书关注的就是这种思想，所以其中的许多知识并不受例子中所用的工具软件和语言的束缚。

如何阅读

How to Read This Book

这本书既是教程，又是参考手册，还可以当故事看，这取决于读者的阅读方式。熟悉正则表达式的读者可能会觉得，这本书马上就能当作一本详细的参考手册，读者可以直接跳到自己需要的章节。不过，我并不鼓励这样做。

要想充分利用这本书，可以把前 6 章作为故事来读。我发现，某些思维习惯和思维方式的确有助于完整的理解，不过最好还是从这几章的讲解中学习它们，而不是仅仅记住其中的几张列表。

故事是这样的，前 6 章是后面 4 章——包括 Perl、Java、.NET 和 PHP——的基础。为了帮助读者理解每一部分，我交叉使用各章的知识，为了提供尽可能方便的索引，我投入了大量的精力（全书中有超过 1 200 处交叉引用，它们以符号加页码的形式标注）。

在读完整个故事以前，最好不要把本书作为参考手册。在开始阅读之前，读者可以参考其中的表格，例如第 92 页的图表，想象它代表了需要掌握的相关信息。但是，还有大量背景信息没有包含在图表中，而是隐藏在故事里。读者阅读完整个故事之后，会对这些问题有个清晰的概念，哪些能够记起来，哪些需要温习。

组织结构

Organization

全书共 10 章，可以从逻辑上粗略地分为三类，下面是总体概览：

导引

第 1 章：介绍正则表达式的基本概念。

第 2 章：考察利用正则表达式进行文本处理的过程。

第 3 章：提供对于特性和工具软件的概述以及简史。

细节

第 4 章：揭示了正则表达式的工作原理的细节。

第 5 章：利用第 4 章的知识，继续学习各种例子。

第 6 章：详细讨论效率问题。

特定工具的知识

第 7 章：详细讲解 Perl 的正则表达式。

第 8 章：讲解 Sun 提供的 `java.util.regex` 包。

第 9 章：讲解.NET 的语言中立的正则表达式包。

第 10 章：讲解 PHP 中提供正则功能的 preg 套件。

导引部分会把完全的门外汉变成“对问题有感觉”的新手。对正则表达式有一定经验的读者完全可以快速翻阅这些章节，不过，即使是对于相当有经验的读者来说，我仍然要特别推荐第 3 章。

- **第 1 章 正则表达式入门**，是为完全的门外汉准备的。我以应用相当广泛的程序 *egrep* 为例来介绍正则表达式，我也提供了我的视角：如何“理解”正则表达式，来为后面章节所包括的高级概念打下坚实的基础。即使是有经验的读者，浏览本章也会有所收获。

- **第 2 章 入门示例拓展**，考察了支持正则表达式的程序设计语言的真实文本处理过程。附加的示例提供了后面章节详细讨论的基础，也展示了高级正则表达式调校背后的重要思考过程。为了让读者学会“正则表达式的套路”，这章出现了一个复杂问题，并讲解了两种全然不相关的工具如何分别通过正则表达式来解决它。

- **第 3 章 正则表达式的特性和流派概览**，提供了当前经常使用的工具的多种正则表达式的概览。因为历史的混乱，当前常用的正则表达式的类型可能差异巨大。此章同时介绍了正则表达式以及使用正则表达式的工具的历史和演化历程。本章末尾也提供了“高级话题引导”。此引导是读者学习此后高级内容的路线图。

细节
The Details

了解了基础知识之后，读者需要弄明白“如何使用”及“这么做的原因”。就像“授人以渔”的典故一样，真正懂得正则表达式的读者，能够在任何时间、任何地点应用关于它的知识。

- **第 4 章 表达式的匹配原理**，循序渐进地导入本书的核心。它从实践的角度出发，考察了正则引擎真实工作的重要的内在机制。懂得正则表达式如何处理工作细节，对读者掌握它们大有裨益。

- **第 5 章 正则表达式实用技巧**，教育读者在高层次和实际的运用中应用知识。这一章会详细讲解常见（但复杂）的问题，目的在于拓展和深化读者对于正则表达式的认识。

- **第 6 章 打造高效正则表达式**，考察真实生活中大多数程序设计语言提供的正则表达式的高效结果。本章运用第 4 章和第 5 章详细讲解的知识，来开发引擎的能力，并避免其中的缺陷。

特定工具的知识
Tool-Specific Information

学习完第 4、5、6 章的读者，不太需要知道特定的实现。不过，我还是用了 4 个整章来讲解 4 种流行的语言。

- **第 7 章 Perl**，详细讲解了 Perl 的正则表达式，Perl 大概是目前最流行的主要的正则表达式编程语言。在 Perl 中，与正则表达式相关的操作符只有四个，但它们组合出的选项和特殊情形带来了大量的程序选项——同时还有陷阱。对没有经验的开发人员来说，这种极其丰富的选项能够让他们迅速从概念转向程序，当然也可能是雷场。本章的详细介绍有助于给读者指出一条光明大道。

- **第 8 章 Java**，详细介绍了 `java.util.regex` 包，从 Java 1.4 以后，它已经成为了 Java 语言的标准部分。本章主要关注的是 Java 1.5，但也提及了它与 Java 1.4.2 和 Java 1.6 的差别。

- **第 9 章 .NET**，是微软尚未提供的.NET 正则表达式库的文档。无论使用 VB.NET、C#、C++、JScript、VBScript、ECMAScript 还是使用.NET 组件的其他语言，本章都提供了详细内容，让读者能够充分利用.NET 的正则表达式。

- **第 10 章 PHP**，简要介绍了 PHP 内嵌的多个正则引擎，并详细介绍了 preg 正则表达式套件（regex engine）的类型和 API，这些是由 PCRE 正则表达式库提供的。

体例说明
Typographical Conventions

在进行（或者谈论）详细的和复杂的文本处理时，保持精确性是很重要的。差一个空格字符，可能导致截然不同的结果，所以我会在本书中使用下面的惯例：

- 正则表达式以「`this`」的形式出现。两端的符号表示“里面有一个正则表达式”，而正则表达式文字（例如用来检索的表达式）以‘`this`’的形式出现。有时候，省略两端的符号和单引号也不会造成歧义，此时我会省略它们。同样，屏幕截图通常以原来的样子出现，而不会用到上面两种符号。

- 在文字文本和表达式内部的省略号会被特别标出。例如，[…]表示一对方括号，之间的内容无关紧要，而[...]表示一对方括号，其中包含三个句点。
- 如果没有明确数字，可能很难判断“a b”之间有多少空格，所以出现在正则表达式和文字文本中的空格以“·”表示。这样“a···b”就清楚多了。
- 我使用可见的制表符，换行符和回车字符：

·	空格字符
	制表符
	换行符
	回车字符

- 有时候，我会使用下画线或有色背景高亮标注文字文本或正则表达式的一部分。下面这句话中，下画线的部分表示表达式真正匹配的部分：

  Because 「cat」 matches 'It·indicates·your·cat·is···' instead of the word 'cat', we realize...

  这个例子中，下画线的部分高亮标记了表达式中添加的字符：

  To make this useful, we can wrap 「Subject|Date」 with parentheses, and append a colon and a space. This yields 「(Subject|Date):·」

- 本书包含了大量的细节和例子，所以我设置了超过 1 200 处的交叉引用，帮助读者理解。它们通常表示为“☛123”，意思是“请参阅第 123 页”。举个例子：“…的说明在表 8-2 中（☛367）”。

练习

Exercises

有时候我会问个问题，帮助读者理解正在讲解的概念，尤其是在前几章这种问题更多。它们并不是摆设，我希望读者在继续阅读之前认真想想。请记住我的话，不要忽略它们的重要意义，本书中这样的问题并不多。它们可以当作自我测试题：如果不是几句话就能说明白的问题，最好是在复习相关章节之后再继续阅读。

为了避免读者直接看到问题的答案，我使用了一点技巧：问题的答案都必须翻页才能看到。通常你必须翻过一页才能看到标着◆的答案。这样答案在你思考问题的时候没法直接看到，但又很容易获得。

链接、代码、勘误及联系方式

Links, Code, Errata, and Contacts

写第 1 版时，我发现修改书本上的 URL，保持与实际一致是件很费工夫的事情，所以，我没有在书后罗列一个 URL 附录，而是提供统一的地址：

http://regex.info

在这里你可以找到与正则表达式相关的链接，书中的所有代码，可检索的索引以及其他资源。本书也可能存在错误☺，所以我提供了勘误。

如果你找到书中的错误，或者仅仅是希望给我写几句话，请写邮件到：*jfriedl@regex.info*。

我们已尽力核验本书所提供的信息，尽管如此，仍不能保证本书完全没有瑕疵，而网络世界的变化之快，也使得本书永不过时的保证成为不可能。如果读者发现本书内容上的错误，不管是赘字、错字、语意不清，甚至是技术错误，我们都竭诚虚心接受读者指教。如果您有任何问题，请按照以下的联系方式与我们联系。

奥莱理软件（北京）有限公司

北京市 海淀区 知春路 49 号 希格玛公寓 B 座 809 室
邮政编码：100080
网页：*http://www.oreilly.com.cn*
E-mail：*info@mail.oreilly.com.cn*

与本书有关的在线信息如下所示。

http://www.oreilly.com/catalog/regex3/（原书）
http://www.oreilly.com.cn/book.php?bn=978-7-121-04684-1（中文版）

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）87690813 转 817
网页：*http://bv.csdn.net*
读者服务信箱：
sheguang@broadview.com.cn（读者信箱）
broadvieweditor@gmail.com（投稿信箱）

目录

Table of Contents