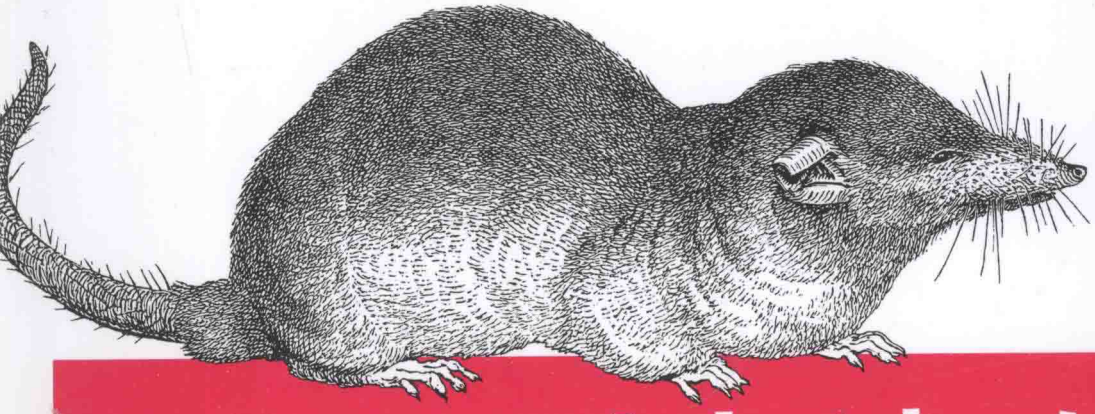
Regular Expressions Cookbook

第2版
Revised and Updated



正则表达式经典实例

[美] *Jan Goyvaerts* *Steven Levithan* 著
郭耀 迟骋 译
余晟 审校

O'REILLY®
人民邮电出版社
POSTS & TELECOM PRESS

O'REILLY®

正则表达式经典实例

（第2版）

[美] *Jan Goyvaerts Steven Levithan* 著

郭耀 迟骋 译

余晟 审校

人民邮电出版社

北京

图书在版编目（CIP）数据

正则表达式经典实例 : 第2版 / (美) 高瓦特斯 (Goyvaerts, J.) , (美) 莱文森 (Levithan, S.) 著 ; 郭耀, 迟骋译. -- 2版. -- 北京 : 人民邮电出版社, 2014.10

书名原文: Regular expressions cookbook, second edition

ISBN 978-7-115-36660-3

Ⅰ. ①正… Ⅱ. ①高… ②莱… ③郭… ④迟… Ⅲ. ①正则表达式 Ⅳ. ①TP301.2

中国版本图书馆CIP数据核字(2014)第200168号

◆ 著 [美] Jan Goyvaerts Steven Levithan

译 郭 耀 迟 骋

审 校 余 晟

责任编辑 杨海玲

责任印制 彭志环 杨林杰

◆ 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

三河市海波印务有限公司印刷

◆ 开本：787×1000 1/16

印张：35.25

字数：753 千字 2014 年 10 月第 2 版

印数：1-3 000 册 2014 年 10 月河北第 1 次印刷

著作权合同登记号 图字：01-2013-3676 号

定价：89.00 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

本书讲解了基于 C#、Java、JavaScript、Perl、PHP、Python、Ruby 和 VB.NET 等 8 种常用编程语言使用正则表达式的经典实例。书中提供了上百种可以在实战中使用的实例，帮助读者使用正则表达式来处理数据和文本。本书针对如何使用正则表达式来解决性能不佳、误报、漏报等常见的错误以及完成一些常见的任务，给出了基于 C#、Java、JavaScript、Perl、PHP、Python、Ruby 和 VB.NET 等编程语言的解决方案，旨在教会读者很多技巧以及避免特定语言的陷阱的方法，读者可以通过本书提供的实例解决方案库来解决实践中的复杂问题。

本书适合对正则表达式感兴趣的软件开发人员和系统管理员阅读。

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

前言

正则表达式在过去十多年间越来越普及。如今，所有常用的编程语言都会包含一个强大的正则表达式函数库，甚至在语言本身就内嵌了对于正则表达式的支持。许多开发人员都会利用这些正则表达式的功能，在应用程序中为用户提供使用正则表达式对其数据进行查找或者过滤的能力。正则表达式已经无处不在。

随着正则表达式的广泛采用，出现了许多相关的著作。其中大多数都很好地讲解了正则表达式的语法，并且还会提供一些例子以及参考实现。然而，我们还没有看到有任何一本书能够针对处理计算机和不同因特网应用上的文本所遇到的各种实际问题为读者提供基于正则表达式的解决方案。因此，本书作者 Steve 和 Jan 决定写一本书来满足这种需求。

我们特别希望能够展现给读者的是：如何使用正则表达式来解决那些正则表达式经验有限的人认为无法解决的问题，或者软件纯粹主义者认为不能用正则表达式来解决的问题。因为如今正则表达式已无处不在，所以它们通常可以作为便利的工具被最终用户直接使用，而不需要程序员团队的参与。即使是对程序员来说，常常也可以在信息检索和修改的任务中采用一些正则表达式来节省大量时间，因为这些功能如果使用过程式代码来实现，可能会需要几小时甚至几天的时间，也可能会由于需要采用第三方的函数库，而不得不经过事先审查和管理人员的审批。

不同版本带来的混乱

与 IT 业界流行的东西一样，正则表达式也拥有许多种不同的实现，以及不同程度的兼容性。这就出现了许多不同的正则表达式*流派*（flavor），它们在处理一个特定正则表达式的时候并不总是拥有完全一样的表现，有时候甚至会无法正常工作。

在许多书中的确也提到了目前存在的不同流派，并且指出了其中的一些区别。但是，它们通常会选择在不同地方略掉一些流派——特别是当某种流派缺少特定功能的时候，而不是为之提供替代解决方案或者变通方法。而当需要在不同的应用程序或编程语言中使用不一样的正则表达式流派的时候，就会令人感到沮丧。

在文字方面，也常常可以看到一些不严格的表达，例如“所有人现在都在使用 Perl 风格的正则表达式”，但是这种说法轻视了大范围的不兼容。即使都是“Perl 风格”的函数库也有显著的区别，而与此同时 Perl 也在持续不断地发展。过度简单化的印象可能会导致程序员浪费半小时的时间来运行调试器却得不到任何结果，而不是去认真检查他们的正

则表达式的实现细节。甚至当他们发现所依赖的一些功能不存在的时候，都不知道该如何找到变通方法。

本书是市场上第一本面面俱到地讨论所有流行且强大的正则表达式流派的书，并且从头到尾贯穿全书。

目标读者

如果你经常在计算机上处理文本，不管是搜索一大堆的文档，还是在文本编辑器中处理文本，抑或是开发需要搜索或处理文本的软件，都应该读一读这本书。正则表达式对于上述这些工作来说是一个非常优秀的工具。本书会教给你需要了解的关于正则表达式的所有东西。你不需要任何先前的经验，因为我们会从正则表达式最基本的概念讲起。

如果你已经拥有使用正则表达式的经验，会发现其他书籍和网上文章中经常一带而过的大量细节。如果你曾经遭遇过正则表达式在一个应用程序中可用而在另外一个程序中不可用的情形，就会因为本书给出了世界上最流行的 7 种正则表达式流派的翔实的讲解，而感到受益颇多。我们把整本书组织成一本经典实例，因此你可以直接跳到想要细细阅读的话题。如果你从头到尾阅读了整本书，就会成为一个正则表达式的世界级“大厨”。

无论你是否是程序员，本书都会教给你使用正则表达式所需知道的所有知识，并且还会讲解更多其他内容。如果你想要在文本编辑器、查找工具，或是任意带有“使用正则表达式”选项输入框的应用程序中使用正则表达式，你根本不需要任何编程经验就可以阅读本书。本书中的大多数例子都拥有完全基于一个或多个正则表达式的解决方案。

如果你是程序员，那么第 3 章会讲解在源代码中应用正则表达式所需的所有信息。这一章假设读者熟悉所选用的编程语言的基本语言特性，但是并不假设你在源代码中曾经使用过任何正则表达式。

涉及的技术

.NET、Java、JavaScript、PCRE、Perl、Python 以及 Ruby，这些不只是一些用在封面上的热门词汇，它们是本书要讲到的 7 种正则表达式流派。我们把这 7 种流派等同对待，同时还会特别仔细地指出这些正则表达式流派中我们所能找到的所有不一致的地方。

关于编程的一章（第 3 章）中包含如下语言的代码示例：C#、Java、JavaScript、PHP、Perl、Python、Ruby 以及 VB.NET。同样，每一个实例都有这 8 种语言的解决方案和解释。虽然这样做会让这一章看起来有些啰唆，但是读者可以很轻松地跳过那些不感兴趣的语言的讨论，而不会错过所选用语言中应当知道的任何内容。

本书的组织结构

本书的前 3 章讲解一些实用的工具和基本信息作为读者使用正则表达式的基础。随后的每一章则介绍各种不同的正则表达式以对文本处理的某个领域进行深入讲解。

第 1 章讲解正则表达式的作用，并介绍了一系列工具，这些工具会使你学习、创建和调试正则表达式更加容易。

第 2 章介绍正则表达式的每个元素和特性，以及高效使用正则表达式的重要指导。这一章构成完整的正则表达式指南。

第 3 章详细介绍了编码相关的技术，并且包含了在本书中涉及的每种编程语言中使用正则表达式的代码示例。

第 4 章中包含如何处理常见用户输入的实例，如日期、电话号码以及不同国家的邮政编码。

第 5 章探讨常见的文本处理任务，例如检查文本行中是否包含特定的单词。

第 6 章会讲解如何检测整数、浮点数以及这些输入的几种其他格式。

第 7 章讲解分析源代码和其他文本文件的基础，并演示了如何使用正则表达式处理日志文件。

第 8 章展示如何能够把在因特网上和 Windows 系统中常用的这些字符串拆分开来，并且利用它们来查找信息。

第 9 章讲解如何处理 HTML、XML、逗号分隔的数值（CSV），以及 INI 风格的配置文件。

排版约定

本书在排版上采用如下约定。

等宽字体

表示程序清单。

等宽斜体

表示应该用用户所提供的值或根据上下文确定的值来替换的文本。

‹Regular•expression› （正则表达式）

用来表示一个正则表达式，它可以单独出现，也可以出现在向某个应用程序的查找框中输入正则表达式的时候。正则表达式中的空格会使用一个灰色圆点来表示以使它们更明显。使用“宽松排列”（free-spacing）模式时，则不使用圆点表示空格，因为该模式下会忽略空格。

«Replacement•text» （替代文本）

用来表示在“查找和替换”的操作中，正则表达式所匹配的文本会被替换成的文本。在替代文本中的空格也会用一个灰色圆点来表示。

Matched text（匹配文本）

用来表示与正则表达式相匹配的目标文本（subject text）中的一部分。

…

在正则表达式中的省略号用来说明在使用该正则表达式之前必须“把这里的空白填好”。相应的文字解释中会告诉你在其中应该填入什么样的内容。

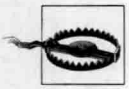
CR、LF 和 CRLF

CR、LF 和 CRLF 放在黑框中用来表示在字符串中实际出现的换行字符，而不是正则表达式中的字符转义序列（character escapes），如\r、\n 和\r\n。要创建这些字符，可以通过在应用程序的多行编辑控件中按回车键（Enter），或者也可以通过在源代码中使用多行字符常量，比如 C#中的逐字字符串（verbatim string），或是 Python 语言中的三引号字符串（triple-quoted string）。

↵

这个符号表示回车箭头，它与键盘上的回车（Return 或 Enter）键上的符号一样，用来表示必须打断才能使之符合印刷页面的宽度。当你在源代码中键入这些代码的时候，不需要按回车键，而是应该把所有内容都键入同一行之中。

这个图标表示提示、建议或者一般的注记。



这个图标用来说明警告或注意事项。

代码示例的使用

本书的目的就是要帮助读者完成手头的工作。一般来说，读者可以随意在程序和文档中使用本书中出现的代码。除非你打算复用本书中大量的代码，否则并不需要联系我们以获得许可。销售或者发布 O'Reilly 图书中包含示例的 CD-ROM 则必须要获得许可。引用本书或者引用其中的示例代码来回答问题并不需要获得许可。在你的产品文档中利用本书中的大量代码示例则需要获得许可。

如果读者在引用本书时提供出处，我们会很感激，虽然我们并不要求你一定这样做。提供出处的时候通常需要包括书名、作者、出版社和书号（ISBN）。例如：“*Regular Expressions Cookbook* by Jan Goyvaerts and Steven Levithan. Copyright 2012 Jan

Goyvaerts and Steven Levithan, 978-1-449-31943-4."。

如果你觉得对代码示例的使用可能会超出上面所给出的许可范围，或是属于合理使用的范围之外，那么请随时通过 permissions@oreilly.com 联系我们。

联系我们

如果你想就本书发表评论或有任何疑问，敬请联系出版社。

美国：

O'Reilly Media Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

我们还为本书建立了一个网页，其中包含了勘误表、示例和其他额外的信息。你可以通过如下地址访问该网页：

http://oreilly.com/catalog/978144319434

关于本书的技术性问题或建议，请发邮件到：

bookquestions@oreilly.com

欢迎登录我们的网站（http://www.oreilly.com），查看更多我们的书籍、课程、会议和最新动态等信息。

我们的其他联系方式如下：

Facebook: http://facebook.com/oreilly
Twitter: http://twitter.com/oreillymedia
YouTube: http://www.youtube.com/oreillymedia

致谢

我们要感谢 O'Reilly Media, Inc.的编辑 Andy Oram，他从头到尾给我们提供了莫大的帮助。我们还要感谢 Jeffrey Friedl、Zak Greant、Nikolaj Lindberg 和 Ian Morse 仔细地审阅了本书第 1 版技术内容，并感谢 Nikolaj Lindberg、Judith Myerson 和 and Zak Greant 审阅了本书第 2 版，他们提供的审阅意见使本书得以做到全面而且准确。

目录

第 1 章

正则表达式简介

在你打开这本书的时候，很可能已经热切地期望，要在代码中插入本书中找到的那些包含诸多括号和问号的古怪字符串了。如果你已经准备好要“即查即用”，我们非常欢迎，第 4～9 章中会列出并讲解了各种实用的正则表达式。

但是如果阅读本书的前几章，你将为未来节省大量的时间。例如，本章会向读者介绍许多工具——其中一些工具是本书作者之一的 Jan 所开发的，这些工具可以帮你事先测试和调试正则表达式，而不用等到把它们塞到代码中之后再处理，那时候查找错误就非常困难了。而且开始这几章还会展示使用正则表达式的不同特性和选项，帮助你轻松应对遇到的问题，并帮助你理解正则表达式，从而提高它们的性能，以及学习不同语言——甚至是你最喜欢的编程语言的不同版本之间——在处理正则表达式的时候出现的细微差别。

因此，我们在这些背景知识上花费了大量的精力，相信读者在开始动手之前会阅读这些内容，或是在使用正则表达式时遇到挫折，而想要巩固你的理解的时候，会回头来阅读它们。

1.1 正则表达式的定义

在本书的上下文中，*正则表达式*（regular expression）是一种可以在许多现代应用程序和编程语言中使用的特殊形式的文本模式。它们可以用来验证输入是否符合给定的文本模式；在一大段文本中查找匹配该模式的文本；用其他文本来替换匹配该模式的文本或者重新组织匹配文本的片段；把一块文本切分成一系列更小的文本，当然如果使用不当也可能搬起石头砸自己的脚。本书会帮助你确切理解正在做的事情，从而避免可能会造成的灾难性后果。

学会了使用正则表达式的技巧，就可以简化许多编程和文本处理的任务，并且让许多没有正则表达式则根本无法实现的任务成为可能。从一个文档中提取所有的电子邮件地址，至少需要几十行，甚至是几百行过程式代码——这些代码编写起来费事，维护起来也麻烦。但是，如果采用了合适的正则表达式，如在实例 4.1 中所给的那样，就只需要很少的几行甚至只要一行代码就可以了。

术语“正则表达式”的历史

术语“正则表达式”来源于数学与计算机科学理论，它用来反映被称为“正则性”的数学表达式特点。这样一个表达式可以通过一个确定性有限自动机（DFA）用软件来实现。一个 DFA 是一个不使用回溯的有限状态机。

最早版本的 grep 工具所使用的文本模式是数学意义上的正则表达式。尽管名字看起来是一样的，然而如今流行的 Perl 风格的正则表达式已经完全不是数学意义上的正则表达式了。它们是采用非确定性的有限自动机（NFA）来实现的。你稍后就会学到和回溯有关的所有内容。关于这条说明，实干的程序员需要记住的所有内容就是：象牙塔里的一些计算机科学家，很不喜欢自己精心定义的术语被套用到现实世界中更为有用的技术。

但是，如果你试图用一个正则表达式来做太多的事情，或者是在根本不适合的情形中非要使用正则表达式，就会明白为什么会存在如下的说法[①]：

> 有些人每遇到一个问题，就会想“我知道怎么做，用正则表达式就可以了。”于是他们就有两个（而不是一个）问题需要解决了。

这些人所遇到的新问题指的就是他们并不会去阅读用户手册，也就是现在你手里的这本书。所以请继续读下去。正则表达式是一个强大的工具。如果你的工作涉及在计算机上编辑或是提取文本，牢固地掌握正则表达式就会为你少度过很多个不眠之夜。

1.1.1 众多正则表达式流派

上一小节的标题确实表述得不那么确切，我们并没有定义正则表达式到底是什么。我们也不可能给出定义。对于哪些文本模式是正则表达式，而哪些不是，并不存在正式的标准来给出严格精确的定义。可以想象得到，每种编程语言的设计人员，以及每款文本处理程序的开发人员，对于正则表达式应该是什么样子，都会有自己不同的想法。因此，我们就不得不面对这样一大堆不同的正则表达式流派。

幸运的是，绝大多数设计人员与开发人员都比较懒惰。如果可以照搬别人已经做好的工作，为什么非要自己创建一些全新的东西呢？正因为此，所有现代的正则表达式流派，包含本书要讨论的这些流派，其历史都可以追溯到 Perl 编程语言。我们把这些流

① Jeffrey Friedl 在他的博客 http://regex.info/blog/2006-09-15/247 中探讨了这句话的来源和历史。