

# 学习正则表达式 (图灵程序设计丛书)

作者: [美]Michael Fitzgerald

## 版权信息

书名: 学习正则表达式

作者: Michael Fitzgerald

译者: 王热宇

ISBN: 978-7-115-31149-8

本书由北京图灵文化发展有限公司发行数字版。版权所有，侵权必究。

---

您购买的图灵电子书仅供您个人使用，未经授权，不得以任何方式复制和传播本书内容。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

# 目录

[版权声明](#)

[O'Reilly Media, Inc.介绍](#)

[前言](#)

[目标读者](#)

[阅读要求](#)

[排版约定](#)

[示例代码](#)

[Safari® Books Online](#)

[联系我们](#)

[致谢](#)

[第1章 什么是正则表达式](#)

[1.1 从Regexpal开始](#)

[1.2 匹配北美电话号码](#)

[1.3 用字符组来匹配数字](#)

[1.4 使用字符组简写式](#)

[1.5 匹配任意字符](#)

[1.6 捕获分组和后向引用](#)

[1.7 使用量词](#)

[1.8 括选字符](#)

[1.9 应用举例](#)

[1.10 本章所学](#)

[1.11 相关资源](#)

[第2章 简单的模式匹配](#)

[2.1 匹配字符串面值](#)

[2.2 匹配数字](#)

[2.3 匹配非数字字符](#)

[2.4 匹配单词和非单词字符](#)

[2.5 匹配空白符](#)

[2.6 再谈匹配任意字符](#)

[2.7 给文本加标签](#)

[2.7.1 用sed为文本加标签](#)

[2.7.2 用Perl为文本加标签](#)

[2.8 本章所学](#)

[2.9 相关资源](#)

[第3章 边界](#)

[3.1 行的起始与结束](#)

[3.2 单词边界与非单词边界](#)

[3.3 其他锚位符](#)

[3.4 使用元字符的字面值](#)

[3.5 添加标签](#)

[3.5.1 使用sed添加标签](#)

[3.5.2 使用Perl添加标签](#)

[3.6 本章所学](#)

[3.7 相关资源](#)

[第4章 选择、分组和后向引用](#)

[4.1 选择操作](#)

[4.2 子模式](#)

[4.3 捕获分组和后向引用](#)

[命名分组](#)

[4.4 非捕获分组](#)

[原子分组](#)

[4.5 本章所学](#)

[4.6 相关资源](#)

[第5章 字符组](#)

[5.1 字符组取反](#)

[5.2 并集与差集](#)

[5.3 POSIX字符组](#)

[5.4 本章所学](#)

[5.5 相关资源](#)

[第6章 匹配Unicode和其他字符](#)

[6.1 匹配Unicode字符](#)

[使用vim](#)

[6.2 用八进制数匹配字符](#)

[6.3 匹配Unicode字符属性](#)

[6.4 匹配控制字符](#)

[6.5 本章所学](#)

[6.6 相关资源](#)

[第7章 量词](#)

[7.1 贪心、懒惰和占有](#)

[7.2 用\\*, +和?进行匹配](#)

[7.3 匹配特定次数](#)

[7.4 懒惰量词](#)

[7.5 占有量词](#)

[7.6 本章所学](#)

[7.7 相关资源](#)

[第8章 环视](#)

[8.1 正前瞻](#)

[8.2 反前瞻](#)

[8.3 正后顾](#)

[8.4 反后顾](#)

[8.5 本章所学](#)

[8.6 相关资源](#)

[第9章 用HTML标记文档](#)

[9.1 匹配标签](#)

[9.2 用sed转换普通文本](#)

[9.2.1 用sed进行替换](#)

[9.2.2 用sed处理罗马数字](#)

[9.2.3 用sed处理特定段落](#)

[9.2.4 用sed处理多行诗文](#)

[9.3 追加标签](#)

[使用sed命令文件](#)

[9.4 用Perl转换普通文本](#)

[9.4.1 用Perl处理罗马数字](#)

[9.4.2 用Perl处理特定段落](#)

[9.4.3 用Perl处理多行诗文](#)

[9.4.4 使用Perl命令文件](#)

[9.5 本章所学](#)

[9.6 相关资源](#)

[第10章 初级班毕业了](#)

[10.1 想上中级班](#)

[10.2 工具、实现程序以及程序库](#)

[10.2.1 Perl](#)

[10.2.2 PCRE](#)

[10.2.3 Ruby \(Oniguruma\)](#)

[10.2.4 Python](#)

[10.2.5 RE2](#)

[10.3 匹配北美电话号码](#)

[10.4 匹配电子邮件地址](#)

[10.5 本章所学](#)

[附录 正则表达式参考](#)

[QED中的正则表达式](#)

[元字符](#)

[字符简写式](#)

[空白符](#)

[Unicode空白字符](#)

[控制字符](#)

[字符属性](#)

[各种字符属性的脚本名称](#)

[POSIX字符组](#)

[选项与修饰符](#)

[正则表达式与ASCII码表](#)

[相关资源](#)

[术语表](#)

[作者及封面简介](#)

## 版权声明

© 2012 by Michael Fitzgerald.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2013. Authorized translation of the English edition, 2012 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc.出版2012。

简体中文版由人民邮电出版社出版，2013。英文原版的翻译得到O'Reilly Media, Inc.的授权。此简体中文版的出版和销售得到出版权和销售权所有者的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式复制。

## O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、在线服务、杂志、调查研究和会议等方式传播创新者的知识。自1978年开始，O'Reilly一直都是发展前沿的见证者和推动者。超级极客正在开创未来，我们关注着真正重要的技术趋势，通过放大那些“微弱的信号”来刺激社会对新科技的采用。作为技术社区中活跃的参与者，O'Reilly的发展充满着对创新的倡导、创造和发扬光大。

作为出版商，O'Reilly为软件开发人员带来革命性的“动物书”，创造了第一个商业网站（GNN），组织开放源代码峰会，以至于开源软件运动以此命名，通过创立Make杂志成为DIY革命的主要先锋，公司一如既往地用各种方式和渠道连接人们和他们所需要的信息。O'Reilly的会议和峰会聚集了超级极客和高瞻远瞩的商业领袖，共同描绘将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计算机用户。无论是通过印刷书籍、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信念——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是位少有的商人，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

本书通过示例介绍如何编写正则表达式，旨在让读者轻松掌握正则表达式。事实上，笔者几乎将所涉及的每一个概念都通过示例展示了出来，读者很容易模仿尝试。

正则表达式有助于找到文本字符串中的各种模式。更确切地说，正则表达式是经过专门编写的文本字符串，用来匹配字符串（尤其是文件内字符串）集合中符合该模式的所有字符串。

正则表达式最早出现于美国数学家斯蒂芬·克莱尼编写的*Introduction to Metamathematics*一书中（1952年Van Nostrand公司出版）。但其实这个概念早在20世纪40年代初就已形成。到了70年代，随着Unix操作系统及其实用程序sed、grep等问世，正则表达式得到了计算机科学家更为广泛的使用。Unix操作系统是美国电话电报公司下属贝尔实验室的Brian Kernighan、Dennis Ritchie、Ken Thompson以及其他工作人员的杰作。

据我所知，最早出现正则表达式的计算机应用程序是QED编辑器。QED是Quick Editor的缩写，它是为运行在Scientific Data Systems公司<sup>(1)</sup> SDS 940计算机中的Berkeley Timesharing System编写的。1970年的记录显示，QED是由Ken Thompson在之前MIT的Compatible Time-Sharing System中另外一个编辑器基础上重写而成的。从此，计算技术领域有了真正的正则表达式实现。（附录中的表A-1列出了QED的正则表达式特性。）

(1) Scientific Data Systems（英文缩写SDS），是Max Palevsky于1961年在美国成立的一家计算机公司，也是最早在计算机设计中使用集成电路和硅晶体管的公司。SDS计算机主要针对大型科学计算，物美价廉。“太空竞赛”期间NASA曾购买了很多台SDS计算机。SDS在1969年被施乐（Xerox）公司收购，1975年由于管理不善和销售下滑被关闭。在施乐管理期间，该公司一度被称为XDS。——编者注

本书中用来展示示例的工具很多，但多数都容易获取，而且也很实用。只有少数工具目前还没有好用的Windows版本。如果你觉得哪个工具不好用，完全可以不用。但要真正学习正则表达式，我还是建议在Unix环境中学习。我使用Unix环境长达25年，每天仍然能够学到不少新东西。

“不懂Unix的人注定还要重新发明一个蹩脚的Unix。”——Henry Spencer<sup>(2)</sup>

(2) Henry Spencer是加拿大程序员，著名正则表达式库regex的作者。这个正则表达式库被许多程序包或编程语言采用，比如Perl、Tcl和MySQL，等等。在多伦多大学工作期间，Henry Spencer从1981年开始运作美国之外的第一个Usenet站点。这个站点后来被谷歌收购，作为1980年代Usenet的公开档案。另外，他还写过“10 Commandments for C Programmers”（程序员十诫，<http://www.seebs.net/c/10com.html>）。——编者注

部分工具可以通过浏览器在线使用，这对于许多读者是十分方便的。其余的工具需要使用命令和shell脚本，还有一些工具是在桌面上运行的。如果你手头没有这些工具，从网上下载也很方便。其中大多数工具是免费的，偶尔有需要付费的也不贵。

本书中不会出现很多专业术语。我会在必要的时候告诉你正确的术语，但这种情况很少。因为多年的经验表明，专业术语常会造成障碍。换句话说，我会尽可能用通俗易懂的语言描述正则表达式，以免你晕头转向不知所措。因为本书的理念是“略知大概，即可实践”。

正则表达式的实现多种多样。你会发现在vi（vim）、grep及sed等Unix命令行工具中使用的正则表达式也可以在其他程序中找到。各种程序设计语言都支持正则表达式，比如Perl（当然啦<sup>(3)</sup>）、Java、JavaScript、C#、Ruby等。就连XSLT 2.0这样的声明式语言中也有正则表达式。你还会发现Notepad++、Oxygen及TextMate等应用程序同样支持正则表达式。

(3) Perl，后来被人们解释为Practical Extraction and Reporting Language的缩写。由这个非官方的“全称”——实用提取和报告语言——可知，Perl在处理文本文件和生成报表方面是非常强大的。1987年，Larry Wall在美国宾夕法尼亚州蓝铃（Blue Bell）地区的Unisys公司当程序员的时候发明了Perl。在该语言后来的发展中，正则表达式功能得到不断丰富和加强，最终成为Perl独树一帜的招牌特色。——编者注

大多数正则表达式实现各有异同。本书不会逐一讨论它们的差异，但也会涉及一些。如果我要把所有实现的全部不同点都列出来，恐怕非得把我累吐血不行。所以我在本书中就不纠缠这些细节了。总而言之，如果你期待一本正则表达式的入门书，那就选这本吧。

## 目标读者

本书适合从零开始学习正则表达式的读者。如果你准备开始学习正则表达式或编写程序，本书就是很好的起点。换句话说，所有听说过正则表达式且对其非常感兴趣，但还没有真正理解正则表达式的人，都是这本书的目标读者。如果你属于这种情况，本书很适合你。

我将采取由简到繁的顺序来介绍正则表达式的特性，也就是先从简单的开始。

如果你已经对正则表达式及其用法有所了解，或者已是位编程老手，本书可能不适合你。本书面向需要指导的初学者。如果你写过一些正则表达式，但希望加深对基本概念的理解，也可以阅读本书，只是我们的节奏可能比你想像的要慢一些。

我在此推荐几本学习完本书后应该阅读的书。第一本是Jeff Friedl的*Mastering Regular Expressions, Third Edition*（参见 <http://shop.oreilly.com/product/9781565922570.do>）。这本书对正则表达式进行了全面阐述，强烈推荐读者阅读。Jan Goyvaerts和 Steven Levithan执笔的*Regular Expressions Cookbook*（参见 <http://shop.oreilly.com/product/9780596520694.do>）也是一本相当不错的书<sup>(1)</sup>。Jan Goyvaerts开发的RegxBuddy是一个强大的桌面应用（参见 <http://www.regexbuddy.com>），而Steven Levithan开发了我们会用到的在线正则表达式处理程序RegxPal（参见 <http://www.regexpal.com>）。

(1) 这两本书的中文版分别是《精通正则表达式（第3版）》（电子工业出版社）和《正则表达式经典实例》（人民邮电出版社）。——编者注

## 阅读要求

为了更好地学习本书，你需要使用Unix或Linux操作系统上的一些工具。这些工具在Mac上的Darwin系统（BSD在Mac上的衍生系统）、Windows电脑中运行的Cygwin（参见<http://www.cygwin.com>及<http://www.gnu.org>）中都能找得到。

本书提供了大量示例供你实验，只是看一遍印象不会深刻。要真正掌握正则表达式，就应该按照这些示例的步骤自己操作，最好把所有示例都过一遍。最好的学习方式是亲身实践，而不是做个旁观者。你得通过本书学会使用高亮显示匹配结果以验证正则表达式的网站、强大的Unix命令行工具、分析正则表达式或用正则表达式搜索文本的桌面程序。

Github上托管着本书的示例代码（<https://github.com/michaeljamesfitzgerald/Introducing-Regular-Expressions>）。另外，通过<http://examples.oreilly.com/0636920012337/examples.zip> 也可以下载到本书所有的示例和测试文件<sup>(1)</sup>。学习本书之前，你最好先在自己的计算机上创建一个新文件夹，并将这些文件保存到该文件夹中。

(1) 读者也可以从图灵社区本书网页（<http://www.it-ebooks.com.cn/book/955>）随书下载部分下载本书示例代码。——编者注

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《学习正则表达式》[美]Michael Fitzgerald.epub

请登录 <https://shgis.cn/post/1856.html> 下载完整文档。

手机端请扫码查看：

