# 精通正则表达式（第3版）

## 作者：Jeffrey E.F.Friedl

**内容简介**

随着互联网的迅速发展，几乎所有工具软件和程序语言都支持的正则表达式也变得越来越强大和易于使用。本书是讲解正则表达式的经典之作。本书主要讲解了正则表达式的特性和流派、匹配原理、优化原则、实用诀窍以及调校措施，并详细介绍了正则表达式在 Perl、Java、.NET、PHP中的用法。

本书自第 1 版开始着力于教会读者“以正则表达式来思考”，来让读者真正“精通”正则表达式。该版对PHP的相关内容、Java1.5和 Java1.6的新特性作了可观的扩充讲解。任何有机会使用正则表达式的读者都会从中获益匪浅。



版权贸易合同登记号 图字：01- 2007-3143

**图书在版编目（CIP）数据**

精通正则表达式：第3版/（美）佛瑞德（Friedl,J.E.F.）著；余晟译.—北京：电子工业出版社，2012.7

书名原文：Mastering Regular Expressions,3rd Edition

ISBN 978-7-121-17501-5

I.①精… II.①佛…②余… III.①正则表达式 IV.①TP301.2

中国版本图书馆CIP数据核字(2012)第147494号

责任编辑：徐津平

印刷：北京智力达印刷有限公司

装订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开本：787×980 1/16 印张：35 字数：742 千字

印次：2012年 7月第 1次印刷

定价：89.00元

# 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，充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

# 重印牟言

**学到不会忘……**

博文视点的张春雨编辑告诉我，八次印刷的《精通正则表达式（第3版）》已经全部售罄了，O'Reilly与电子工业出版社续签了版权合同，准备重新上市，让我写一点东西。

该写什么好呢？

2007年《精通》上市时，我还在中关村，天气好的时候可以望见颐和园的佛香阁；而现在，窗外景色已经换成了珠江边的小蛮腰；对正则表达式的使用，也从随手拈来变得生疏——许多问题需要翻查《精通》，翻查自己写的《正则指引》。究其原因，与正则表达式直接相关的开发做得少了，古话说"勤则立，嬉则荒"，就是这个道理。

荒是荒了，毕竟还没荒废，虽然有很多细节需要查阅，但是我很清楚，某个问题能不能用正则表达式解决，该怎样解决。或者说，虽然手上生疏了，心里其实没有忘记，而这一切，归源都是之前死啃过《精通》的缘故。

在阅读《精通》之前，我已经查阅了网上的不少资料，对正则表达式有了基本了解，能像模像样地解决一些实际问题，可算"够用"了。这时候遇见《精通》这样"现实价值不那么大"的书，能静下心去阅读，其实带着点毕业不久的傻气，只是单纯想把它弄懂搞透。所以，遇到匹配原理这类看来没多少实用价值的知识，还会愿意花时间去揣摩、研习。回头想想，也正是因为当时有这种傻气，可算是意外的收获：工作中经常需要学习一些工具和原理，虽然当时也"学会"了，但不久就忘个精光；相比之下，正则表达式却是学到了"不会忘"的程度。更典型的例子是游泳，几乎人人都可以做到"一朝学会，终身不忘"。同样是"学会"，为什么差距这么大呢？

这个问题我想了很久，最后的答案是，"学会"的定义是不同的。

通常我们说"学会"了某项技术、某门语言，意思是"凑合能用"，或者说"可以对照文档（Google）解决问题"的程度——你用Python解决了一个问题，就说明你"学会"了Python，哪管是步步Google，还是照抄现成的代码。而我们说"学会"了游泳，意思是可以在水里行动而不沉下去，更重要的是在游泳时不需要时刻背诵各种口诀：吸气—伸手—划水—蹬腿—抬头—呼气……，如果你在泳池里必须谨记口诀，是绝对谈不上"学会"的。

两者虽然都叫"学会"，其实相差迥异：第一种"学会"是"照猫画虎"，第二种"学会"是"融会贯通"，虽然都可以解决问题，但从第一种"学会"到达第二种"学会"，其实需要经历漫长的过程。而且，两种"学会"都能解决问题，所以在达到第二种"学会"的漫长过程中，你很可能感觉不到自己的进步，反而会困惑继续学习的意义乃至放弃——既然能对着文档操作，既然有现成的资料，为什么要去理解背后的原理呢。

对我来说，第二种"学会"的好处是显而易见的，最重要的一点就是不会忘记——学习的时间增长一倍，遗忘的难度将会增加十倍、二十倍甚至一百倍。这些年来，我见到了太多这样的例子：有人每次用到正则表达式都会抓狂，都要四处极力搜索、反复盲目尝试，花很长时间才能凑出、蒙对解决方案；另一方面，他们又不愿意花时间潜心学习《精通》这样的经典。因为反复遗忘，需要反复学习，最终浪费了大量的时间。

许多人不愿意专门花时间来学习正则表达式，是认为它属于奇技淫巧，并非工作必须。但这理由是不成立的：我们大部分人不是作家，但为了在需要的时候写得出文章，还是必须专门花时间来练习写作。而且，专门花时间来学习"非必要"的技能，以后往往能有意想不到的收获。我真切体会到并且懂得这个道理，恰好也是与《精通》的翻译有缘。

在翻译《精通》时，为了省却重新编排索引的麻烦，需要做到中英文版页页对应，于是我专门学习了侯捷老师写的《Word排版艺术》，并且亲手尝试了每个例子，记熟了有关的概念和术语，从此学会了运

用格式和样式的角度定义文档，再不用为格式之类的问题烦恼。这些年来，虽然用得并不多，却没有忘记。去年写作《正则指引》时，我事先完整定义了各种格式、样式、引用等，交稿时节省了自己和出版社大量的时间。

另一个例子仍然与正则表达式有关。去年，为了写作《正则指引》中Unicode的章节，我专门花了时间研读Unicode规范，虽然最终《指引》中没有列出学到的全部知识，但我对Unicode的理解已经不再限于“在程序中设定 Unicode 编码即可”。前几天，有位同事遇到 Unicode字符Ä（U+00C4）无法打印的问题，于是我建议他使用A和¨（U+0041和U+0308）的两个Unicode字符来表示（按照Unicode规范，两个字符可以“组合”成一个字符），果然解决了问题。这段经历再次证明，真的学会了，就真的不会忘。

亚里士多德曾说：“所谓幸福，就是尽情地施展我们掌握的技能，等待期望的结果。”然而很多时候，虽然我们以为自己可以解决，但是之前学过的技能已经遗忘，于是施展起来步履沉重、举步维艰，最后只能精疲力竭地等待结果，自然与幸福绝缘。相反，如果我们能把重要的技能都真正学会，学到不会忘的程度，自然可以接近幸福。如果你想收获自如驾驭正则表达式的幸福，不妨从这本书开始吧。

# 目录