

Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™



Beginning Perl

Perl入门经典

[美] Curtis “Ovid” Poe 著
朱允刚 韩雷 叶斌 译



清华大学出版社

TP312/4899

2013

Perl 入门经典

[美] Curtis “Ovid” Poe 著

朱允刚

韩 雷

叶 斌

译

北方工业大学图书馆



C00348594

清华大学出版社

北 京

Curtis "Ovid" Poe

Beginning Perl

EISBN: 978-1-118-01384-7

Copyright © 2012 by Wiley Publishing, Inc., Indianapolis, Indiana

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2013-0281

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Perl 入门经典/(美) 坡(Poe, O. C.) 著; 朱允刚, 韩雷, 叶斌 译. —北京: 清华大学出版社, 2013.10

书名原文: Beginning Perl

ISBN 978-7-302-33723-2

I. P… II. ①坡… ②朱… ③韩… ④叶… III. Perl 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2013)第 207749 号

责任编辑: 王 军 杨信明

装帧设计: 牛艳敏

责任校对: 成凤进

责任印制: 王静怡

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 39 字 数: 1046 千字

版 次: 2013 年 10 月第 1 版 印 次: 2013 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 78.00 元

产品编号: 043730-01

作者简介

Curtis “Ovid” Poe 早在 1982 年就开始了他的编程生涯，这其中，几乎专职编写 Perl 程序的时间长达 13 年。他目前是 Perl 基金会(Perl Foundation)的董事会成员，曾在许多国家举行的会议上发表演讲，但作为丈夫和父亲才是最令他感到自豪的事情。

技术编辑简介

Chromatic 是一位多产的作者和开发者，他最近为 Onyx Neon Press(<http://onyxneon.com/>)出版社出版了 *Modern Perl* 一书。

致 谢

和许多书一样，这一路上如果没有许多人的帮助，这本书也不可能得以付梓。在这里，我要特别感谢 Michael Rasmussen——我的“秘密评审员”，尽管不是正式的审阅人员，但他依然细致彻底地查看了每章内容，并反馈许多非常有益的意见，大大提升了本书的品质。

我还要感谢技术评审员 Chromatic，他不遗余力一次又一次地来烦我，指出那些我本应但却没有注意到的微妙问题。他是一个比我还要优秀的程序员，这可真气人。

Mary James 和 Maureen Spears 是我在 Wiley 的 Wrox 出版公司的主要联系人，和他们一起工作是一种享受，在写书工作看似要拖上比料想更多的时间时，这种享受确实有助于我保持高昂的斗志，他们的幽默感和帮助在整个制作过程中起到了无法估量的作用。我还要感谢 San Dee，虽然我不知道她是谁，但她的名字在整个创作过程中不断出现，非常感谢她为本书许多问题的捕获所付出的努力。

我还要感谢 Adrian Howard、Ævar Arnfjörð Bjarmason、Alejandro Lopez、Andy Armstrong、Aristotle、Michael Schwern、Ricardo Signes、Sean T Lewis 和 Simon Cozens，感谢他们傻乎乎地同意审阅这么厚的一本书。

最后，我要感谢 O'Reilly 的 Open Feedback Publishing System 系统的工作人员，感谢他们为本书创建了一个评审社区(<http://ofps.oreilly.com/titles/9781118013847/>)，并感谢这一站点带来的一切有益的意见。

还有太多的名字没有被一一提及，我事先在此向你们表示歉意。

就个人而言，我不得不说，很多时候在读到“所有错误在我(and all errors are mine)”这样的声明时，我会想：“但这就是为什么要有审阅人员的原因啊，不是吗？”不过现实情况远非如此。在撰写一本书时，审阅者会从中找出大量的问题，就像本书那些审阅人员所做的那样，但他们不可能注意到所有的问题。现在我意识到，在有着这种涉及面的作品中，我要为所有的错漏之处负起责任。审阅者一般没有作品酬劳，他们不会像我一样一直坐在那里，每夜花上几个小时，数月如一日，为每一个段落操心。他们找出大多问题，但这一作品中的其余缺陷却是我的责任，是我一个人的责任。

前 言

“找份工作吧，嬉皮士！”

拒绝使用这句话作为本书的副标题是很遗憾的事情，但也很明智。不过，它传达出了关于本书我很看重的两件事，那就是：在学习所需的技能时做到赢取工作和享受乐趣两不误，好吧，是在学习如何编程的同时适度地享受一些乐趣了。

尽管许多书都没有直言这样的意图，但我还是在此先把它给挑明了吧：这是一本与赚钱有关的书。信息技术(Information Technology, IT)工作者的需求量很大，甚至是在当前经济不景气的时候也是如此，本书这样说凭借的不仅是我 13 年的 Perl 编程语言经验，还有就“野生状态的 Perl”这一主题所做的一些调查。这就是为什么你将会在书中发现一个令人吃惊的决定：本书把关注的重点放在 Perl 的 5.8 和 5.10 版本上，它们已不再是官方支持的版本，但却是大多数的公司仍在使用的版本。幸运的是，Perl 5 Porters(也被称作 P5P)努力保持着向后兼容，因此本书中的代码依然能够运行在最新版本 Perl 上。作为这一重点关注的结果，在完成本书的学习后，你将拥有在接受许多 Perl 工作时必须具备的技能。

最初我设想这是一本以开发工作技能为目标的 Perl 语言书，当时我住在俄勒冈州的波特兰市；后来，我搬到了伦敦，并做了一些跟写作本书有关的调查，但没有起到什么作用；再后来，我搬到了阿姆斯特丹，开始与 Wrox 合作撰写本书；现在，我住在巴黎，并在此完成本书的写作。所有这些城市呈现出来的共同迹象是，Perl 打开了工作的大门。在许多其他优秀的动态语言，比如说 PHP、Python 和 Ruby 正在为同一块馅饼争斗时，一些 Perl 开发者却转向了其他语言，离开了那些需要开发者使用 Perl 语言来维护代码和构建新系统的公司，Perl 开发者的需求量居高不下，本书的出现就是为了满足这一需要。

为了免得你认为 Perl 的作用就是维护遗留代码，我向你保证，许多公司，包括大型的和小型的，仍都在转向 Perl，把它作为编程语言的首选。该语言功能强大、切实可靠，Perl 综合典藏网(Comprehensive Perl Archive Network, CPAN)仍是面向单门语言的最大开源代码集。许多时候你会发现，不一定需要编写新的代码来解决某个棘手问题，你可以求助于 CPAN，在上面找出他人已写好的这类代码。

我曾用多种语言编程赚钱，其中包括了 6809 汇编语言(糟糕，我是不是刚说漏了自己的年龄?)、Basic、C、Java、COBOL、FOCUS、JCL(Job Control Language)、VBA 和 JavaScript 等，但一直会用 Perl，为什么会这样？嗯，为什么不呢？这是一门功能强大的语言。如果编程的需要受限于 CPU，比如说在实时的光线跟踪方面，那么 Perl 可能不是最佳选择，但在其他方面，Perl 则是一门非常优秀的语言。我的开发工作偏重于大型的数据库驱动应用，这些应用的性能问题通常会出现在网络、数据库或是文件系统中。无论使用哪种编程语言，你都会遇到类似的性能问题，所以，不妨选择一门自己喜欢的语言。

那么，我使用 Perl 开发过哪些应用呢？这其中可能最有名气的一个例子与电影相关。如果你在报纸上读到自己喜欢的电影在上周末创造了数百万美元的票房，那么这些数字很有可能就是经我编

写的 Perl 软件处理后得出的(我应该在处理前加上实时两个字), 因为这些数字报道通常都来自一家公司。

我还为世界上最大的广播公司 BBC 工作了好几年, 为他们开发中央元数据资源库。当时 BBC 的另一个团队需要一些节目数据(如果你住在 UK, 那么可能听说过被称作 iPlayer 的这个好东西), 他们就有可能调用了我编写的 Perl 软件。

我也曾在世界上最大的在线酒店预订公司工作过, 刚入职时, 公司正忙于把许多 Java 程序转换成 Perl 程序, 因为 Perl 好用得不得了。公司几乎所有的后端代码都是用 Perl 编写的, 对于它所取得巨大成功来说, Perl 功不可没。

目前我在为 Weborama 工作, 这家公司是欧洲在线营销技术方面的先驱之一。我要应对数量惊人的流量和数据, Perl 相当好地解决了所有的这些问题。实际上, Weborama 之所以抛弃了其他几种很流行的编程语言, 代之以 Perl, 就是因为 Perl 能把工作完成得很好。

最后要说的是, 使用 Perl 的乐趣很多, 虽然我还会涉猎其他一些语言(主要是 JavaScript, 不过目前看来是 Erlang 特别有吸引力), 但很愿意继续使用 Perl 来编码。有时我会匆忙写出一个立刻能用的 bash 脚本, 但只要脚本一开始变得复杂, 我就会懊恼地发现, 用 Perl 来编写会更容易一些。对我来说, Perl 真的太好用了。

本书读者对象

这本书是为你而写的吗? 我所能尽力保证的是, 即使是没有编程经验的人也能拿起本书学习 Perl。

没有编程经验的读者

不过, 若没有编程经验, 你需要把许多注意力放在第 1 章上, 该章内容介绍了许多不同的、对新入门的程序员很有帮助的资源。通常你会发现, Perl 社区是一个非常友善的地方, 始终以帮助他人学习为乐。在没有计算机背景的情况下, 第 2 章学起来可就要费点劲了, 该章讨论通过 CPAN 安装 Perl 代码的做法, 不过你可以翻回第 1 章的内容, 向这其中介绍的许多优秀资源寻求帮助(这其中包括了一些本地用户组, 在那里你会遇到其他一些 Perl 程序员)。在攻克了第 2 章的学习曲线之后, 你就会发现本书剩余部分的内容与其他的编程书没有什么不同, 一样的简单。

有经验的程序员

如果你是一位有经验的程序员, 正在寻找关于某种语言的全面资源, 那么这样的一本书正是你需要的。本书第 1 章主要谈及从哪些地方寻求帮助, 不过现在的你可能已懂得如何去查找编程的答案了。第 2 章讨论通过 CPAN 安装 Perl 模块, 这部分内容至少值得你略读一遍。不过从第 3 章开始, 你就要有所关注了, 在这一章中, 我们讨论了 Perl 的变量。Perl 并未把重点过多地放在你所使用的数据类型上, 它关注的是你组织数据的方式。Perl 假设你是称职的, 了解自己的数据, 它要做的是按照你需要的方式来使组织数据变成一件简单的事情。

本书涵盖的内容

虽然这会使有些人感到意外，但我们还是把重点放在了 5.8 和 5.10 这两个不再被支持的 Perl 版本上，这是因为，多个调查及我个人在多家公司的工作和顾问经验都表明，公司在升级编程语言方面态度保守，往往会使用旧的版本。幸运的是 P5P 非常看重 Perl 新版本的向后兼容性，故书中的所有例子都应该可以在 Perl 的最新版本上运行。在适当的时候，我们会讨论一些你可能会遇到的新功能，并会明确指出在什么情况下会遇到这些功能。

我们重点关注 Perl 语言的核心功能，然后再转到以 Web 技术为中心的数据库使用上。为什么以 Web 技术为中心？赚钱是目的，本书的目标是帮你赢取一份工作，如果不了解 SQL 或 HTML，那么找出一些在线教程来学习一点 SQL 和 HTML，这最终(在阅读第 15 章时)是会有好处的。使用本书不会用到太多这方面的知识，但对基础知识的了解有助于更容易地理解书中的一些例子，所以这是值得的。

本书的组织方式

每一章都构建在之前的章节之上，这就是本书的写作方式。

- 第 1 章和第 2 章：本书的头两章(很滑头地称为第 1 章和第 2 章)大都是背景资料，它们告诉你从哪里寻求额外的帮助，以及如何设置 CPAN 客户端来安装额外的 Perl 模块。
- 第 3~10 章：这几章涵盖了 Perl 语言的核心功能，在读完这些内容后，你会发现，编写 Perl 程序来处理许多常见的任务是很容易的事情。实际上，《Perl 入门经典》这一书名正是指这部分内容而言。
- 第 11~13 章：这几章开始谈及模块(一种组织代码的方式)和面向对象编程(一种创建可重用“专家”的强大做法，这些“专家”可处理一些公共的编程任务)。
- 第 14 章：这一章讨论测试，一个我倍感亲切的主题。许多程序员备受恐惧的驱动编程(fear-driven programming)的折磨，这一问题是指在使用一些大型系统时害怕对其进行修改，因为你不知道会破坏哪些内容。没错，测试可以解除你的这种恐惧感，甚至是在大型系统上，也能赋予你进行任何所需修改的自信。
- 第 15~18 章：这几章某种程度上是可选的，但别跳过它们。这些章节可以向你传授一点点中级到高级 Perl 程序员所需的技能，你会了解到使用 Perl 构建网站有多容易、如何使用数据库、如何处理许多常见的任务(比如说处理日期)以及如何使用命令行应用程序等。
- 第 19 章：本章总结了已了解的内容和仍需要学习的内容，以此作为整本书的结尾。此外，你还要构建一个 Web 应用来管理多媒体版权数据，回击 DMCA 的撤除通知。这是一项雄心勃勃的任务，不过你会看到，使用 Perl 和 CPAN 来完成这一任务是多么容易。
- 附录：本书每一章的最后都附有一组习题，目的是进一步加深对在该章中学到的技能的理解。这一附录给出了这些习题的答案，可别弄虚作假，先看了答案，这可是作弊。

本书用到的软件产品和工具

很幸运, Perl 代码几乎可以运行在每一种操作系统上, 在系统间进行切换时, 比如说从 Windows 切换到 Linux 上, 通常不需要做任何改动。在使用书中的例子时, 唯一需要用到的软件是 Perl 的 5.8 或更新版本, 书中后面有些章节要求通过 CPAN 安装代码, 不过这没问题, 第 2 章已彻底解释了 CPAN 的用法。

源代码

在演练本书中的例子时, 你可以选择手工输入所有代码, 或是使用本书附带的源代码文件。书中用到的所有源代码, <http://www.wrox.com> 都提供了下载, 本书为书中出现的每段代码或每个代码清单都提供了一个文件名, 这就是 www.wrox.com 网站上的源代码的文件名。进入该网站后, 只需(使用搜索框或是某个标题列表)找出本书的标题, 然后在图书的详细说明页面中单击 **Download Code** 链接就可以得到本书的所有源代码。

注意: 因为许多书都有着类似的标题, 所以你可能会发现通过 EISBN 来进行查找是最容易的, 本书的 EISBN 是 978-1-118-01384-7。

在下载代码之后, 只需使用你所喜欢的压缩工具来进行解压即可。此外, 你还可以通过 <http://www.wrox.com/dynamic/books/download.aspx> 这一地址来访问 Wrox 的代码下载页面, 查看本书和其他 Wrox 图书提供的代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误, 但是错误总是难免的, 如果您在本书中找到了错误, 例如拼写错误或代码错误, 请告诉我们, 我们将非常感激。通过勘误表, 可以让其他读者避免受挫, 当然, 这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件, 我们会检查您的反馈信息, 如果是正确的, 我们将在本书的后续版本中采用。在本书编辑过程中, 我们接受了热心读者白爽针对第 4 版中文译著提出的一些修改意见, 在此特向白爽表示衷心感谢。

要在网站上找到本书英文版的勘误表, 可以登录 <http://www.wrox.com>, 通过 **Search** 工具或书名列表查找本书, 然后在本书的细目页面上, 单击 **Book Errata** 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表, 网址是 www.wrox.com/misc-pages/booklist.shtml。

p2p.wrox.com

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。
- (4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。



不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 1 章 Perl 简介	1	第 3 章 变量	33
1.1 Perl 的发展现状	2	3.1 何为程序设计	34
1.2 获取 Perl	2	3.2 正式开始前的注意事项	34
1.2.1 在非 Windows 平台使用 Perl: perlbrew	3	3.2.1 strict、warnings 及 diagnostics	34
1.2.2 使用 Windows 平台	5	3.2.2 my 函数	35
1.3 Perl 社区	6	3.2.3 魔符	36
1.3.1 IRC	7	3.2.4 标识符	36
1.3.2 PerlMonks	7	3.3 标量	37
1.3.3 PerlMongers	8	3.3.1 字符串	38
1.3.4 StackOverflow	8	3.3.2 数字	41
1.4 使用 Perl 文档	9	3.4 数组	43
1.4.1 理解 perldoc 的结构	9	3.4.1 分解代码	43
1.4.2 开始使用 perldoc	10	3.4.2 访问元素	44
1.4.3 使用教程和常见问题解答	10	3.4.3 对数组进行迭代	46
1.4.4 使用 perldoc -f	11	3.5 哈希表	47
1.5 使用终端窗口	12	3.5.1 访问元素	47
1.5.1 使用命令行	12	3.5.2 对哈希表进行迭代	48
1.5.2 创建工作目录	13	3.5.3 对哈希增加数据	48
1.6 创建“Hello, World!”程序	15	3.6 切片	50
1.6.1 编写你的第一个程序	15	3.6.1 数组切片	50
1.6.2 shebang 行	17	3.6.2 哈希切片	50
1.7 本章小结	18	3.7 上下文	51
第 2 章 理解 CPAN	19	3.7.1 标量上下文	51
2.1 CPAN 和 metacpan	20	3.7.2 列表上下文	52
2.1.1 搜索和评价模块	21	3.8 作用域	54
2.1.2 下载和安装	22	3.8.1 my 变量	55
2.2 cpan 客户端	26	3.8.2 包变量	56
2.2.1 使用 CPAN.pm 客户端	26	3.9 strict、warnings 和 diagnostics	59
2.2.2 使用 cpanm 客户端	27	3.9.1 strict	60
2.2.3 ppm	28	3.9.2 warnings	60
2.2.4 CPAN::Mini	28	3.9.3 diagnostics	61
2.3 本章小结	31	3.9.4 在没有网络的情况下工作	62
		3.10 Perl 的内置变量	64
		3.10.1 \$_	64

3.10.2	%ENV	65	5.3	使用 while 或 until 循环	118
3.10.3	@ARGV	65	5.3.1	列表	119
3.10.4	其他的特殊变量	65	5.3.2	内置函数: last()、next()、redo() 和 continue()	120
3.11	本章小结	66	5.3.3	标签	122
第 4 章	数据处理	69	5.4	语句修饰符	123
4.1	使用标量	70	5.4.1	语句修饰符的类型	123
4.1.1	字符串处理	71	5.4.2	do while 或 do until	124
4.1.2	使用字符串操作符	78	5.5	given/when 语句	126
4.1.3	Scalar::Util	81	5.5.1	基本语法	126
4.1.4	数值型内置函数或操作符	81	5.5.2	Switch 模块	128
4.1.5	位操作符	86	5.6	本章小结	128
4.1.6	了解布尔操作	86	第 6 章	引用	131
4.1.7	赋值操作符	90	6.1	关于引用的基础知识	131
4.1.8	优先级和关联性	91	6.1.1	数组引用	132
4.2	数组和列表函数	93	6.1.2	哈希引用	133
4.2.1	内置数组函数	93	6.1.3	匿名引用	134
4.2.2	List::Util	97	6.1.4	其他引用	136
4.3	内置的哈希函数	97	6.2	使用引用	139
4.3.1	delete()	97	6.2.1	调试	139
4.3.2	exist()	98	6.2.2	复制	141
4.3.3	key()	98	6.2.3	切片	144
4.3.4	values()	98	6.3	本章小结	145
4.3.5	each()	98	6.4	练习	145
4.4	作用域关键字	99	第 7 章	子程序	147
4.4.1	my()	100	7.1	子程序的语法	148
4.4.2	local()	100	7.1.1	参数处理	148
4.4.3	our()	100	7.1.2	多重参数	149
4.4.4	state()	101	7.1.3	命名参数	150
4.5	本章小结	101	7.1.4	别名	152
第 5 章	流程控制	103	7.1.5	状态变量(5.10 之前版本和 5.10 之后版本)	153
5.1	使用 if 语句	103	7.1.6	传递列表、哈希或哈希引用?	154
5.1.1	理解基本的条件语句	104	7.2	返回数据	157
5.1.2	else/elsif/unless	105	7.2.1	返回真/假	157
5.1.3	三元操作符?:	109	7.2.2	返回单值或多值	158
5.2	for/foreach 循环	110	7.2.3	wantarray 函数	159
5.2.1	数组	110	7.3	错误	160
5.2.2	列表	112			
5.2.3	C 语言风格	113			

7.3.1	注意! 终止程序	161	8.3.1	Regex::Common	206
7.3.2	carp 和 croak	161	8.3.2	E-mail 地址	207
7.3.3	eval	162	8.3.3	HTML	207
7.3.4	evalGotchas	164	8.4	本章小结	209
7.3.5	Try::Tiny	165	第 9 章	文件与目录	213
7.4	子程序引用	165	9.1	基本的文件处理方法	214
7.4.1	已有的子程序	166	9.1.1	打开并读取文件	214
7.4.2	匿名子程序	166	9.1.2	文件测试操作符	220
7.4.3	闭包	166	9.1.3	菱形操作符	222
7.5	原型	169	9.1.4	临时文件	223
7.5.1	参数强制	169	9.1.5	DATA 文件	223
7.5.2	更多原型技巧	171	9.1.6	binmode	225
7.5.3	仿造内置函数	172	9.2	目录	227
7.5.4	前置声明	174	9.2.1	读取目录	227
7.5.5	原型总结	175	9.2.2	文件名代换	227
7.6	递归	177	9.3	Unicode	228
7.6.1	基本递归	177	9.3.1	Unicode 的概念	229
7.6.2	分治法	178	9.3.2	两条简单的规则	229
7.6.3	记忆子程序	179	9.3.3	一些复杂的规则	232
7.7	注意事项	183	9.4	本章小结	244
7.7.1	参数别名	183	第 10 章	sort、map 和 grep	247
7.7.2	作用域问题	183	10.1	基本排序	248
7.7.3	功能过多	184	10.1.1	按字母顺序排序	248
7.7.4	参数过多	184	10.1.2	按数字顺序排序	249
7.8	本章小结	184	10.1.3	反向排序	249
第 8 章	正则表达式	187	10.1.4	复杂的排序条件	250
8.1	基本匹配	187	10.1.5	编写一个排序子程序	251
8.1.1	限定符	189	10.1.6	排序和 Unicode 码	252
8.1.2	转义序列	191	10.2	map 函数和 grep 函数	256
8.1.3	提取数据	193	10.2.1	使用 grep 函数	256
8.1.4	修饰符和锚字符	195	10.2.2	使用 map 函数	261
8.1.5	字符类	197	10.2.3	别名问题	263
8.1.6	分组	198	10.2.4	关于实现过多功能	264
8.2	高级匹配	201	10.2.5	关于编写精巧的代码	265
8.2.1	替换	201	10.3	汇总	265
8.2.2	向前查找/向后查找锚位	201	10.3.1	Schwartzian 变换(也就是修饰、 排序以及去除修饰)	266
8.2.3	命名的子表达式(需 Perl 5.10 或 更高版本)	203	10.3.2	Guttman-Rosler 变换	267
8.3	常见的正则表达式问题	206			

10.4	本章小结	268	12.3.3	使用 TV::Episode::Broadcast	323
第 11 章	包和模块	271	12.3.4	类数据和实例数据	327
11.1	名称空间和包	272	12.4	简单回顾	329
11.1.1	use 和 require	276	12.4.1	重载对象	329
11.1.2	包变量	278	12.4.2	使用 UNIVERSAL	332
11.1.3	版本号	281	12.5	了解私有方法	335
11.2	其他包中的子程序	281	12.6	错误陷阱	340
11.2.1	导出	281	12.6.1	非必需的方法	340
11.2.2	命名惯例	284	12.6.2	“内部接触”	340
11.3	BEGIN、UNITCHECK、CHECK、INIT 和 END	288	12.6.3	多继承	341
11.3.1	BEGIN 模块	289	12.7	本章小结	343
11.3.2	END 块	290	第 13 章	Moose	345
11.3.3	INIT、CHECK 和 UNITCHECK 块	290	13.1	了解基本 Moose 语法	346
11.4	置标语言(POD)	291	13.2	使用属性	348
11.4.1	文档结构	293	13.3	使用构造函数	351
11.4.2	标题	293	13.3.1	BUILDARGS	351
11.4.3	段落	293	13.3.2	BUILD	352
11.4.4	列表	294	13.4	了解继承	353
11.4.5	字面段落	294	13.5	谨慎使用 Moose	354
11.4.6	其他方面	295	13.6	高级 Moose 语法	358
11.5	创建与安装模块	296	13.6.1	使用类型约束	358
11.5.1	创建一个简单的模块	296	13.6.2	使用方法修改器	361
11.5.2	Makefile.PL 还是 Module::Build?	301	13.6.3	了解和使用角色	364
11.6	本章小结	301	13.6.4	探讨 MooseX	368
第 12 章	面向对象的 Perl	303	13.6.5	重写 Television::Episode	370
12.1	以私人买手 ÆVAR 为例讲解什么是对象	304	13.7	Moose 最佳实践	376
12.2	Perl OO 的三条规则	304	13.7.1	使用 namespace::autoclean 并把类设为不可变的	376
12.2.1	类就是包	305	13.7.2	绝不重写 new()	376
12.2.2	对象就是知道其所属类的引用	306	13.7.3	始终调用父类的 BUILDARGS 方法	376
12.2.3	方法就是子程序	308	13.7.4	如果属性是非必需的,就提供默认值	376
12.3	关于对象的另一种观点	320	13.7.5	默认只读	377
12.3.1	使用 TV::Episode	320	13.7.6	把自定义类型置于单独模块中并为它们指定名称空间	377
12.3.2	子类化	323	13.7.7	别使用多重继承	377
			13.7.8	始终同时消费所有角色	377
			13.8	本章小结	378

第 14 章 测试	381	15.2.3 借助程序填写表单	450
14.1 基本测试	382	15.3 本章小结	455
14.1.1 使用 Test::More	382	第 16 章 数据库	457
14.1.2 编写测试	384	16.1 使用 DBI	458
14.1.3 了解 prove 实用工具	385	16.1.1 连接数据库	458
14.2 了解 Test::More 的一些测试		16.1.2 使用 SQLite	460
函数	386	16.1.3 使用 DBD::SQLite	460
14.2.1 使用 ok	386	16.2 选择基本的数据	465
14.2.2 使用 is	387	16.2.1 使用 SELECT 语句	465
14.2.3 使用 like	389	16.2.2 使用绑定参数	468
14.2.4 使用 is_deeply	390	16.2.3 插入和更新数据	471
14.2.5 使用 SKIP	391	16.2.4 创建事务处理	472
14.2.6 使用 TODO	391	16.2.5 错误处理	473
14.2.7 使用 eval{}	392	16.3 本章小结	473
14.2.8 使用 use_ok 和 require_ok	393	第 17 章 使用其他外部工具	475
14.2.9 使用其他各类测试函数	393	17.1 命令行	475
14.3 使用其他一些测试模块	397	17.1.1 读取用户输入	476
14.3.1 使用 Test::Differences	397	17.1.2 处理命令行实参	478
14.3.2 使用 Test::Exception	399	17.1.3 perlrun	480
14.3.3 使用 Test::Warn	400	17.2 其他程序	485
14.3.4 使用 Test::Most	400	17.2.1 运行外部程序	485
14.4 了解 xUnit 风格的测试	401	17.2.2 读取其他程序的输出	487
14.4.1 使用 Test::Class	401	17.2.3 写入其他程序的输入	488
14.4.2 基本测试类	403	17.2.4 STDERR	490
14.4.3 扩展测试类	406	17.3 本章小结	492
14.4.4 使用测试控制方法	410	第 18 章 常见任务	495
14.4.5 调用父类的测试控制方法	412	18.1 使用 CSV 数据	496
14.5 本章小结	416	18.1.1 读取 CSV 数据	497
第 15 章 互联网	419	18.1.2 写入 CSV 数据	498
15.1 HTTP 简介	420	18.2 理解基本的 XML	499
15.1.1 Plack	421	18.2.1 读取 XML 数据	499
15.1.2 Hello World 程序	422	18.2.2 写入 XML 数据	503
15.1.3 处理参数	427	18.3 日期处理	506
15.1.4 模板	428	18.3.1 使用 DateTime 模块	507
15.1.5 处理 POST 请求	433	18.3.2 使用 Date::Tiny 模块和 DateTime::Tiny 模块	507
15.1.6 会话	436	18.4 理解你的程序	513
15.2 Web 客户端	447	18.4.1 使用调试器	513
15.2.1 提取网页中的链接	448		
15.2.2 提取网页中的注释	449		

18.4.2	性能分析	519
18.4.3	Perl::Critic 模块	528
18.5	本章小结	531
第 19 章	进阶	533
19.1	接下来的事情	534
19.1.1	本书涵盖的内容	534
19.1.2	本书略而不谈的内容	534
19.2	了解对象-关系映射器	535
19.2.1	了解 DBIx::Class	536
19.2.2	了解 DBIx::Class 的基本用法	536
19.2.3	了解 ORM 的优缺点	539
19.2.4	使用 DBIx::Class::Schema::Loader	544

19.3	使用 Template Toolkit	545
19.3.1	使用模板的理由	545
19.3.2	Template Toolkit 入门	546
19.4	使用 Catalyst 构建应用程序	554
19.4.1	MVC 之美	554
19.4.2	设立 Catalyst 应用程序	555
19.4.3	使用 Catalyst 视图	560
19.4.4	使用 Catalyst 模型	562
19.4.5	使用 Catalyst 控制器	565
19.4.6	CRUD: 创建、读取、更新和删除	566
19.5	本章小结	569

附录	习题答案	573
----	------	-----

第 1 章

Perl 简介

本章主要内容

- 获取和安装 Perl
- 了解社区
- 理解 Perl 文档
- 使用终端
- 编写你的第一个“Hello, World”程序

Perl 程序设计语言在计算和 Web 领域具有丰富的历史和巨大的影响力，本章只是对 Perl 语言的一些亮点进行简要介绍。

在你阅读完本章后，将能很好地了解 Perl 的历史，也将知道当你需要了解本书未能涉及的内容时，如何获取更多的帮助。如何寻找“开发过程中所遇到的问题的答案”可能是你需要学习的最重要的能力。

在安装 Perl 之前，你需要了解一些关于 Perl 的术语，以便能与其他 Perl 用户准确交流。

首先需要明确的是，这个程序设计语言的名称是 Perl，而不是 PERL。Perl 的创始人 Larry Wall 起初想使用一个具有积极内涵的名称，并将该语言命名为 Pearl，但在该语言发布之前，他发现了另一个叫 Pearl 的程序设计语言，因此他将该语言的名称缩短为 Perl。

这一编程语言的名称会引起一些混淆。当人们写成 Perl(首字母大写)时，它们指的是你在本书所学的程序设计语言。当人们写成 perl(首字母小写)时，它们指的是运行 Perl 语言所需的二进制可执行程序。

因此，perl 是二进制程序，而 Perl 是语言。前者对后者进行语法分析并运行后者：perl 对 Perl 进行语法分析并运行 Perl。如果某些人写出 PERL，那么你马上知道他并不熟悉 Perl 语言。所以有时你会看到一些有经验的程序员使用 PERL 来暗指编写得很差的 Perl 程序。

由于 Perl 起初附带的文档中的措辞，使得很多程序员认为 PERL 是“实践提取与报告语言 (Practical Extraction and Report Language)”只取每个单词首字母的缩写词。但 Perl 附带的文档 perlfaq1 给出如下描述：