

精通 *Perl* (影印版)

*Mastering*

# Perl



O'REILLY®

東南大學出版社

*brian d foy* 著  
*Randal L. Schwartz* 序

---

精通Perl (影印版)

Mastering Perl

*brian d foy*

*foreword by Randal L. Schwartz*

**O'REILLY®**

*Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo*

O'Reilly Media, Inc. 授权东南大学出版社出版

南京 • 东南大学出版社

## 图书在版编目 (CIP) 数据

精通 Perl: 英文 / (美) 福瓦 (Foy, B.D.) 著. — 影印本. — 南京: 东南大学出版社, 2008.2

书名原文: Mastering Perl

ISBN 978-7-5641-1037-6

I . 精… II . 福… III . PERL 语言—程序设计—英文  
IV . TP312

中国版本图书馆 CIP 数据核字 (2007) 第 193117 号

江苏省版权局著作权合同登记

图字: 10-2007-219 号

©2007 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2008. Authorized reprint of the original English edition, 2007 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 2007.

英文影印版由东南大学出版社出版 2008。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 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重制。

## 精通 Perl (影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出版人: 江 汉

网 址: <http://press.seu.edu.cn>

电子邮件: [press@seu.edu.cn](mailto:press@seu.edu.cn)

经 销: 全国各地新华书店

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 21.5 印张

字 数: 361 千字

版 次: 2008 年 2 月第 1 版

印 次: 2008 年 2 月第 1 次印刷

书 号: ISBN 978-7-5641-1037-6/TP · 173

印 数: 1~3500 册

定 价: 48.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

## O'Reilly Media, Inc.介绍

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，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 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。

## 出版说明

随着计算机技术的成熟和广泛应用,人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而,计算机领域的技术更新速度之快也是众所周知的,为了帮助国内技术人员在第一时间了解国外最新的技术,东南大学出版社和美国 O'Reilly Meida, Inc.达成协议,将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作,以影印版或者简体中文版的形式呈献给读者。其中,影印版书籍力求与国外图书“同步”出版,并且“原汁原味”展现给读者。

我们真诚地希望,所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助,对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

最新出版的影印版图书,包括:

- 《深入浅出 PMP》(影印版)
- 《RESTful Web Services》(影印版)
- 《学习 WCF》(影印版)
- 《学习 Ruby》(影印版)
- 《Beautiful Code》(影印版)
- 《ActionScript 3.0 精髓》(影印版)
- 《Linux 系统管理》(影印版)
- 《精通 Perl》(影印版)
- 《深入浅出面向对象分析与设计》(影印版)
- 《Linux Kernel 技术手册》(影印版)
- 《ActionScript 3.0 Cookbook》(影印版)
- 《CSS:The Missing Manual》(影印版)

---

# Foreword

One of the problems we face at Stonehenge as professional trainers is to make sure that we write materials that are reusable in more than one presentation. The development expense of a given set of lecture notes requires us to consider that we'll need roughly two to four hundred people who are all starting in roughly the same place, and who want to end up in the same place, and who we can find in a billable situation.

With our flagship product, the *Learning Perl* course, the selection of topics was easy: pick all the things that nearly everyone will need to know to write single-file scripts across the broad range of applications suited for Perl, and that we can teach in the first week of classroom exposure.

When choosing the topics for *Intermediate Perl*, we faced a slightly more difficult challenge, because the “obvious” path is far less obvious. We concluded that in the second classroom week of exposure to Perl, people will want to know what it takes to write complex data structures and objects, and work in groups (modules, testing, and distributions). Again, we seemed to have hit the nail on the head, as the course and book are very popular as well.

Fresh after having updated our *Learning Perl* and *Intermediate Perl* books, brian d foy realized that there was still more to say about Perl just beyond the reach of these two tutorials, although not necessarily an “all things for all people” approach.

In *Mastering Perl*, brian has captured a number of interesting topics and written them down with lots of examples, all in fairly independently organized chapters. You may not find everything relevant to your particular coding, but this book can be picked up and set back down again as you find time and motivation—a luxury that we can't afford in a classroom. While you won't have the benefit of our careful in-person elaborations and interactions, brian does a great job of making the topics approachable and complete.

And oddly enough, even though I've been programming Perl for almost two decades, I learned a thing or two going through this book, so brian has really done his homework. I hope you find the book as enjoyable to read as I have.

—Randal L. Schwartz

---

# Preface

*Mastering Perl* is the third book in the series starting with *Learning Perl*, which taught you the basics of Perl syntax, progressing to *Intermediate Perl*, which taught you how to create reusable Perl software, and finally this book, which pulls everything together to show you how to bend Perl to your will. This isn't a collection of clever tricks, but a way of thinking about Perl programming so you integrate the real-life problems of debugging, maintenance, configuration, and other tasks you'll encounter as a working programmer. This book starts you on your path to becoming the person with the answers, and, failing that, the person who knows how to find the answers or discover the problem.

## Structure of This Book

### Chapter 1, *Introduction: Becoming a Master*

An introduction to the scope and intent of this book.

### Chapter 2, *Advanced Regular Expressions*

More regular expression features, including global matches, lookarounds, readable regexes, and regex debugging.

### Chapter 3, *Secure Programming Techniques*

Avoid some common programming problems with the techniques in this chapter, which covers taint checking and gotchas.

### Chapter 4, *Debugging Perl*

A little bit about the Perl debugger, writing your own debugger, and using the debuggers others wrote.

### Chapter 5, *Profiling Perl*

Before you set out to improve your Perl program, find out where you should concentrate your efforts.

### Chapter 6, *Benchmarking Perl*

Figure out which implementations do better on time, memory, and other metrics, along with cautions about what your numbers actually mean.

### Chapter 7, *Cleaning Up Perl*

Wrangle Perl code you didn't write (or even code you did write) to make it more presentable and readable by using `Perl::Tidy` or `Perl::Critic`.

### Chapter 8, *Symbol Tables and Typoglobs*

Learn how Perl keeps track of package variables and how you can use that mechanism for some powerful Perl tricks.

### Chapter 9, *Dynamic Subroutines*

Define subroutines on the fly and turn the tables on normal procedural programming. Iterate through subroutine lists rather than data to make your code more effective and easy to maintain.

### Chapter 10, *Modifying and Jury-Rigging Modules*

Fix code without editing the original source so you can always get back to where you started.

### Chapter 11, *Configuring Perl Programs*

Let your users configure your programs without touching the code.

### Chapter 12, *Detecting and Reporting Errors*

Learn how Perl reports errors, how you can detect errors Perl doesn't report, and how to tell your users about them.

### Chapter 13, *Logging*

Let your Perl program talk back to you by using `Log4perl`, an extremely flexible and powerful logging package.

### Chapter 14, *Data Persistence*

Store data for later use in other programs, a later run of the same program, or to send as text over a network.

### Chapter 15, *Working with Pod*

Translate plain ol' documentation into any format that you like, and test it, too.

### Chapter 16, *Working with Bits*

Use bit operations and bit vectors to efficiently store large data.

### Chapter 17, *The Magic of Tied Variables*

Implement your own versions of Perl's basic data types to perform fancy operations without getting in the user's way.

### Chapter 18, *Modules As Programs*

Write programs as modules to get all of the benefits of Perl's module distribution, installation, and testing tools.

### Appendix A

Explore these resources to continue your Perl education.

### Appendix B

My popular step-by-step guide to solving any Perl problem. Follow these steps to improve your troubleshooting skills.



## Conventions Used in This Book

The following typographic conventions are used in this book:

### Constant width

Used for function names, module names, environment variables, code snippets, and other literal text

### Italics

Used for emphasis, Perl documentation, filenames, and for new terms where they are defined

## Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact O'Reilly for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Mastering Perl* by brian d foy. Copyright 2007 O'Reilly Media, Inc., 978-0-596-52724-2."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at [permissions@oreilly.com](mailto:permissions@oreilly.com).

## Safari® Enabled



When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://safari.oreilly.com>.

## Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
800-998-9938 (in the United States or Canada)  
707-829-0515 (international/local)  
707-829-0104 (fax)

The web page for this book, which lists errata, examples, or any additional information, can be found at:

<http://www.oreilly.com/catalog/9780596527242>

Additionally, while the book was being written, the author maintained a web site for this book where you can find additional information, including links to related resources and possible updates to the book:

[http://www.pair.com/comdog/mastering\\_perl](http://www.pair.com/comdog/mastering_perl)

To comment or ask technical questions about this book, send email to:

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

For more information about books, conferences, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:

<http://www.oreilly.com>

## Acknowledgments

Many people helped me during the year I took to write this book. The readers of the *Mastering Perl* mailing list gave constant feedback on the manuscript and sent patches, which I mostly applied as is, including those from Andy Armstrong, David H. Adler, Renée Bäcker, Anthony R. J. Ball, Daniel Bosold, Alessio Bragadini, Philippe Bruhat, Katharine Farah, Shlomi Fish, David Golden, Bob Goolsby, Ask Bjørn Hansen, Jarkko Hietaniemi, Joseph Hourcle, Adrian Howard, Offer Kaye, Stefan Lidman, Eric Maki, Josh McAdams, Florian Merges, Jason Messmer, Thomas Nagel, Xavier Noria, Les Peters, Bill Riker, Yitzchak Scott-Thoennes, Ian Sealy, Sagar R. Shah, Alberto Simões, Derek B. Smith, Kurt Starsinic, Adam Turoff, David Westbrook, and Evan Zacks. I'm quite reassured that their constant scrutiny kept me on the right path.

Tim Bunce provided gracious advice about the profiling chapter, which includes `DBI::Profile`, and Jeffrey Thalhammer updated me on the current developments with his `Perl::Critic` module.

Perrin Harkins, Rob Kinyon, and Randal Schwartz gave the manuscript a thorough beating at the end, and I'm glad I chose them as technical reviewers because their advice is always spot on.

Allison Randal provided valuable Perl advice and editorial guidance on the project, even though she probably dreaded my constant queries. Near the end of the year, Andy Oram took over as editor and helped me get the manuscript into shape so we could turn it into a book. The entire O'Reilly Media staff, from editorial, production, marketing, sales, and everyone else, was friendly and helpful, and it's always a pleasure to work with them. It takes much more than an author to create a book, so thank a random O'Reilly employee next time you see one.

Randal Schwartz, my partner at Stonehenge Consulting, warned me that writing a book was a lot of work and still let me mostly take the year off to do it. I started in Perl by reading his *Learning Perl* and am now quite pleased to be adding another book to the series. As Randal has told me many times "You'll get paid more at Starbucks and get health insurance, too." Authors write to share their thoughts with the world, and we write to make other people better programmers.

Finally, I have to thank the Perl community, which has been incredibly kind and supportive over the 10 years that I've been part of it. So many great programmers and managers helped me become a better programmer, and I hope this book does the same for people just joining the crowd.

---

# Table of Contents

<b>Foreword .....</b>	<b>xiii</b>
<b>Preface .....</b>	<b>xv</b>
<b>1. Introduction: Becoming a Master .....</b>	<b>1</b>
What It Means to Be a Master	2
Who Should Read This Book	3
How to Read This Book	3
What Should You Know Already?	4
What I Cover	4
What I Don't Cover	5
<b>2. Advanced Regular Expressions .....</b>	<b>7</b>
References to Regular Expressions	7
Noncapturing Grouping, (?:PATTERN)	13
Readable Regexes, /x and (?#...)	14
Global Matching	15
Lookarounds	19
Deciphering Regular Expressions	25
Final Thoughts	28
Summary	29
Further Reading	29
<b>3. Secure Programming Techniques .....</b>	<b>31</b>
Bad Data Can Ruin Your Day	31
Taint Checking	32
Untainting Data	38
List Forms of system and exec	42
Summary	44
Further Reading	44

<b>4.</b>	<b>Debugging Perl .....</b>	<b>47</b>
	Before You Waste Too Much Time	47
	The Best Debugger in the World	48
	perl5db.pl	59
	Alternative Debuggers	60
	Other Debuggers	64
	Summary	66
	Further Reading	66
<b>5.</b>	<b>Profiling Perl .....</b>	<b>69</b>
	Finding the Culprit	69
	The General Approach	73
	Profiling DBI	74
	Devel::DProf	83
	Writing My Own Profiler	85
	Profiling Test Suites	86
	Summary	88
	Further Reading	88
<b>6.</b>	<b>Benchmarking Perl .....</b>	<b>91</b>
	Benchmarking Theory	91
	Benchmarking Time	93
	Comparing Code	96
	Don't Turn Off Your Thinking Cap	97
	Memory Use	102
	The perlbench Tool	107
	Summary	109
	Further Reading	110
<b>7.</b>	<b>Cleaning Up Perl .....</b>	<b>111</b>
	Good Style	111
	perltidy	112
	De-Obfuscation	114
	Perl::Critic	118
	Summary	123
	Further Reading	123
<b>8.</b>	<b>Symbol Tables and Typeglobs .....</b>	<b>125</b>
	Package and Lexical Variables	125
	The Symbol Table	128
	Summary	136
	Further Reading	136

<b>9. Dynamic Subroutines .....</b>	<b>137</b>
Subroutines As Data .....	137
Creating and Replacing Named Subroutines .....	141
Symbolic References .....	143
Iterating Through Subroutine Lists .....	145
Processing Pipelines .....	147
Method Lists .....	147
Subroutines As Arguments .....	148
Autoloaded Methods .....	152
Hashes As Objects .....	154
AutoSplit .....	154
Summary .....	155
Further Reading .....	155
<b>10. Modifying and Jury-Rigging Modules .....</b>	<b>157</b>
Choosing the Right Solution .....	157
Replacing Module Parts .....	160
Subclassing .....	162
Wrapping Subroutines .....	167
Summary .....	169
Further Reading .....	170
<b>11. Configuring Perl Programs .....</b>	<b>171</b>
Things Not to Do .....	171
Better Ways .....	174
Command-Line Switches .....	177
Configuration Files .....	183
Scripts with a Different Name .....	187
Interactive and Noninteractive Programs .....	188
perl's Config .....	189
Summary .....	191
Further Reading .....	191
<b>12. Detecting and Reporting Errors .....</b>	<b>193</b>
Perl Error Basics .....	193
Reporting Module Errors .....	199
Exceptions .....	202
Summary .....	209
Further Reading .....	209
<b>13. Logging .....</b>	<b>211</b>
Recording Errors and Other Information .....	211

Log4perl	212
Summary	218
Further Reading	218
<b>14. Data Persistence .....</b>	<b>219</b>
Flat Files	219
Storable	228
DBM Files	232
Summary	234
Further Reading	234
<b>15. Working with Pod .....</b>	<b>237</b>
The Pod Format	237
Translating Pod	238
Testing Pod	245
Summary	248
Further Reading	249
<b>16. Working with Bits .....</b>	<b>251</b>
Binary Numbers	251
Bit Operators	253
Bit Vectors	260
The vec Function	261
Keeping Track of Things	266
Summary	268
Further Reading	268
<b>17. The Magic of Tied Variables .....</b>	<b>269</b>
They Look Like Normal Variables	269
At the User Level	270
Behind the Curtain	271
Scalars	272
Arrays	277
Hashes	286
Filehandles	288
Summary	290
Further Reading	291
<b>18. Modules As Programs .....</b>	<b>293</b>
The main Thing	293
Backing Up	294
Who's Calling?	294
Testing the Program	295

Distributing the Programs	302
Summary	303
Further Reading	303
<b>A. Further Reading .....</b>	<b>305</b>
<b>B. Brian's Guide to Solving Any Perl Problem .....</b>	<b>309</b>
<b>Index .....</b>	<b>315</b>



# **Introduction: Becoming a Master**

This book isn't going to make you a Perl master; you have to do that for yourself by programming a lot of Perl, trying a lot of new things, and making a lot of mistakes. I'm going to help you get on the right path. The road to mastery is one of self-reliance and independence. As a Perl master, you'll be able to answer your own questions as well as those of others.

In the golden age of guilds, craftsmen followed a certain path, both literally and figuratively, as they mastered their craft. They started as apprentices and would do the boring bits of work until they had enough skill to become the more trusted journeymen. The journeyman had greater responsibility but still worked under a recognized master. When he had learned enough of the craft, the journeyman would produce a “master work” to prove his skill. If other masters deemed it adequately masterful, the journeyman became a recognized master himself.

The journeymen and masters also traveled (although people disagree on whether that's where the “journey” part of the name came from) to other masters, where they would learn new techniques and skills. Each master knew things the others didn't, perhaps deliberately guarding secret methods, or knew it in a different way. Part of a journeyman's education was learning from more than one master.

Interactions with other masters and journeymen continued the master's education. He learned from those masters with more experience and learned from himself as he taught journeymen, who also taught him because they brought skills they learned from other masters.

The path an apprentice followed affected what he learned. An apprentice who studied with more masters was exposed to many more perspectives and ways of teaching, all of which he could roll into his own way of doing things. Odd teachings from one master could be exposed by another, giving the apprentice a balanced view on things. Additionally, although the apprentice might be studying to be a carpenter or a mason, different masters applied those skills to different goals, giving the apprentice a chance to learn different applications and ways of doing things.