

Preface

Welcome to the eighth edition of *Learning Perl*, updated for Perl 5.34 and its latest features. This book is still mostly good even if you are still using Perl 5.8 (although, it's been a long time since it was released; have you thought about upgrading?).

If you're looking for the best way to spend your first 30 to 45 hours with the Perl programming language, you've found it. In the pages that follow, you'll find a carefully paced introduction to the language that is the workhorse of the internet, as well as the language of choice for system administrators, web hackers, and casual programmers around the world. We've designed this book based on the in-person classes we teach, so we've timed the book for a week's worth of work.

We hope you're reading this preface before you buy the book, because there's a historical hiccup that may cause some confusion. There's another language, Perl 6, that started off as a replacement for Perl 5 but then went out on its own with the new name “Raku” (although brian's book on that language is still [*Learning Perl 6*](#)).

Along with that, there's currently a move to make a new major version of Perl, Perl 7. That's supposed to be Perl v5.34 with different defaults as a baby step to evolving the language. Since it's basically Perl 5, it should be able to run Perl 5 programs, although perhaps with a compatibility switch. As we write this, we're not sure how that will shake out. After you finish this book, you may like to read another book by brian, [*Preparing for Perl 7*](#). Since much of that advice is simply modern good practices, we'll try to give you that same advice in this book.

As we write this, Perl 5 is probably the version you want. It's the widely installed and used language that people mean when they say simply “Perl.” It's going to be the interesting and most used version for a long time. It's the one you want if you don't know why this paragraph is here.

We can't give you all of Perl in just a few hours. The books that promise to do that are probably fibbing a bit. Instead, we've carefully selected a useful subset of Perl for you to learn, good for programs from 1 to 128 lines long (an arbitrary number), which end up being about 90% of the programs in use out there. And when you're ready to go on, you can get [*Intermediate Perl*](#), which picks up where this book leaves off. We've also included a number of pointers for further education.

Each chapter is short enough for you to read in an hour or two. Each chapter ends with a series of exercises to help you practice what you've just learned, with the answers in [*Appendix A*](#) for your reference. Thus, this book is ideally suited for a classroom "Introduction to Perl" course. We know this directly because the material for this book was lifted almost word for word from our flagship "Learning Perl" course, delivered to thousands of students around the world. However, we've designed the book for self-study as well. *brian* provides additional exercises and detailed answers in a separate companion book, [*Learning Perl Exercises*](#).

Perl lives as the "toolbox for Unix," but you don't have to be a Unix guru, or even a Unix user, to read this book. Unless otherwise noted, everything we're saying applies equally well to Windows [*ActivePerl from ActiveState*](#) and [*Strawberry Perl*](#) and pretty much every other modern implementation of Perl.

Although you don't need to know a single thing about Perl to begin reading this book, we recommend that you already have familiarity with basic programming concepts such as variables, loops, subroutines, and arrays, and the all-important "editing a source code file with your favorite text editor." We don't spend any time trying to explain those concepts.

Although we're pleased that we've had many reports of people successfully picking up *Learning Perl* and grasping Perl as their first programming language, of course we can't promise the same results for everyone.

Typographical Conventions

The following font conventions are used in this book:

Constant width

Used for method names, function names, variables, and attributes.
It is also used for code examples.

Constant width bold

Used to indicate user input.

Constant width italic

Used to indicate a replaceable item in code (e.g., *filename*, where you are supposed to substitute an actual filename).

Italic

Used for filenames, URLs, hostnames, commands in text, important words on first mention, and emphasis.

[37]

At the start of an exercise's text, we provide a (very rough) estimate of how many minutes you can expect to spend on that particular exercise.

Code Examples

This book is here to help you get your job done. You are invited to copy the code in the book and adapt it for your own needs. Rather than copying by hand, however, we encourage you to download the code from [the book's companion website](#). You do not need to contact us 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 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, authors, publisher, and ISBN. For example: “*Learning Perl*, 8th edition, by Randal L. Schwartz, brian d foy, and Tom Phoenix (O’Reilly). Copyright 2021 Enhydra Services, LLC, 978-1-492-09495-1.” 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.

O’Reilly Online Learning

NOTE

For more than 40 years, [O’Reilly Media](#) has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O’Reilly’s online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O’Reilly and 200+ other publishers. For more information, visit <http://oreilly.com>.

How to Contact Us

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 or local)
- 707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at

<https://oreil.ly/learning-perl-8e>.

Email bookquestions@oreilly.com to comment on or ask technical questions about this book.

For news and information about our books and courses, visit

<http://oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>.

Follow us on Twitter: <http://twitter.com/oreillymedia>.

Watch us on YouTube: <http://www.youtube.com/oreillymedia>.

History of This Book

For the curious, here's how Randal tells the story of how this book came about:

After I had finished the first [Programming.perl](#) book with Larry Wall (in 1991), I was approached by Taos Mountain Software in Silicon Valley to produce a training course. This included having me deliver the first dozen or so courses and train their staff to continue offering the course. I wrote the course for them¹ and delivered it for them as promised.

On the third or fourth delivery of that course (in late 1991), someone came up to me and said, “You know, I really like [Programming.perl](#), but the way the material is presented in this course is so much easier to follow—you oughta write a book like this course.” It sounded like an opportunity to me, so I started thinking about it.

I wrote to Tim O'Reilly with a proposal based on an outline that was similar to the course I was presenting for Taos—although I had rearranged and modified a few of the chapters based on observations in the classroom. I think that was my fastest proposal acceptance in history—I got a

message from Tim within 15 minutes, saying “We’ve been waiting for you to pitch a second book—[*Programming.perl*](#) is selling like gangbusters.” That started the effort over the next 18 months to finish the first edition of *Learning Perl*.

During that time, I was starting to see an opportunity to teach Perl classes outside Silicon Valley,² so I created a class based on the text I was writing for *Learning Perl*. I gave a dozen classes for various clients (including my primary contractor, Intel Oregon), and used the feedback to fine-tune the book draft even further.

The first edition hit the streets on the first day of November 1993,³ and became a smashing success, frequently even outpacing [*Programming.perl*](#) book sales.

The back-cover jacket of the first book said “written by a leading Perl trainer.” Well, that became a self-fulfilling prophecy. Within a few months, I was starting to get email from all over the United States asking me to teach at their site. In the following seven years, my company became the leading worldwide on-site Perl training company, and I had personally racked up (literally) a million frequent-flier miles. It didn’t hurt that the web started taking off about then, and the webmasters and webmistresses picked Perl as the language of choice for content management, interaction through CGI, and maintenance.

For two years, I worked closely with Tom Phoenix in his role as lead trainer and content manager for Stonehenge, giving him charter to experiment with the “Llama” course by moving things around and breaking things up. When we had come up with what we thought was the best major revision of the course, I contacted O’Reilly and said, “It’s time for a new book!” And that became the third edition.

Two years after writing the third edition of the Llama, Tom and I decided it was time to push our follow-on “advanced” course out into the world as a book, for people writing programs that are “100 to 10,000 lines of code.” And together we created the first Alpaca book ([*Learning Perl Objects, References, and Modules*](#)), released in 2003.

But fellow instructor brian d foy had noticed that we could use some rewriting in both books, because our courseware still needed to track the changing needs of the typical student. So, he pitched the idea to O'Reilly to take on rewriting both the Llama and the Alpaca. This edition of the Llama reflects those changes. brian has really been the lead writer here, working with my occasional guidance, and has done a brilliant job of the usual “herding cats” that a multiple-writer team generally feels like.

On December 18, 2007, the Perl 5 Porters released Perl 5.10, a significant new version of Perl with several new features. The previous version, 5.8, had focused on the underpinnings of Perl and its Unicode support. The latest version, starting from the stable 5.8 foundation, was able to add completely new features, some of which it borrowed from the development of Perl 6. Some of these features, such as named captures in regular expressions, are much better than the old ways of doing things, thus perfect for Perl beginners. We hadn't thought about a fifth edition of this book, but Perl 5.10 was so much better that we couldn't resist.

Since then, Perl has been under constant improvement and is keeping a regular release cycle. Each new Perl release has brought exciting new features, many of which programmers have wanted for years. As long as Perl keeps doing that, we'll keep updating this book.

Changes from the Previous Edition

The text is updated for the latest version, Perl 5.34, and some of the code only works with that version. We note in the text when we are writing about a Perl 5.34 feature, and we mark those code sections with a special `use` statement that ensures you're using the right version:

```
use v5.34; # this script requires Perl 5.34 or greater
```

If you don't see that `use v5.34` in a code example (or a similar statement with a different version), it should work all the way back to Perl 5.8. To see which version of Perl you have, try the `-v` command-line switch:

```
$ perl -v
```

In some examples, we'll show a lower minimum Perl version because that's all the program needs. For instance, `say` was introduced in Perl v5.10:

```
use v5.10;  
  
say "Howdy, Fred!";
```

In most cases, we'll probably forgo a new feature to make the examples work on as many versions of Perl as possible. That doesn't mean you shouldn't use the new features or that we don't endorse them. We simply have a wide audience for this book.

We include Unicode examples and features where appropriate. If you haven't started playing with Unicode, you may want to read our primer in [Appendix C](#). You have to bite the bullet sometimes, so it might as well be now. You'll see Unicode throughout the book, most notably in the chapters on scalars ([Chapter 2](#)), input/output ([Chapter 5](#)), and sorting ([Chapter 14](#)).

Here's a quick summary of the updated or new things in this edition:

- We updated references to Perl 6 to call it by its new name, “Raku.”
- The *search.cpan.org* site was subsumed into MetaCPAN, so we removed references to the old site.
- [ActiveState has discontinued PPM](#), its Perl Package Manager, so we've removed references to that.

Acknowledgments

From Randal

I want to thank the Stonehenge trainers past and present (Joseph Hall, Tom Phoenix, Chip Salzenberg, brian d foy, and Tad McClellan) for their

willingness to go out and teach in front of classrooms week after week and to come back with their notes about what's working (and what's not), so we could fine-tune the material for this book. I especially want to single out my coauthor and business associate, Tom Phoenix, for having spent many, many hours working to improve Stonehenge's Llama course and to provide the wonderful core text for most of this book. And brian d foy for being the lead writer beginning with the fourth edition, and taking that eternal to-do item out of my inbox so that it would finally happen.

I also want to thank everyone at O'Reilly, especially our very patient editor and overseer for previous editions, Allison Randal (no relation, but she has a nicely spelled last name), editor Simon St. Laurent, and Tim O'Reilly himself for taking a chance on me in the first place with the Camel and Llama books.

I am also absolutely indebted to the thousands of people who have purchased the past editions of the Llama so that I could use the money to stay "off the streets and out of jail"; and to those students in my classrooms who have trained me to be a better trainer; and to the stunning array of Fortune 1000 clients that have purchased our classes in the past and will continue to do so into the future.

As always, a special thanks to Lyle and Jack, for teaching me nearly everything I know about writing. I won't ever forget you guys.

From brian

I have to thank Randal first, since I learned Perl from the first edition of this book, and then had to learn it again when he asked me to start teaching for Stonehenge in 1998. Teaching is often the best way to learn. Since then, Randal has mentored me not only in Perl but several other things he thought I needed to learn—like the time he decided we could use Smalltalk instead of Perl for a demonstration at a web conference. I'm always amazed at the breadth of his knowledge. He's the one who told me to start writing about Perl. Now I'm helping out on the book where I started. I'm honored, Randal.

I probably only actually saw Tom Phoenix for less than two weeks in the entire time I worked for Stonehenge, but I had been teaching his version of Stonehenge’s “Learning Perl” course for years. That version turned into the third edition of this book. By teaching Tom’s new version, I found new ways to explain almost everything, and learned even more corners of Perl.

When I convinced Randal that I should help out on the Llama update, I was anointed as the maker of the proposal to the publisher, the keeper of the outline, and the version control wrangler. Our editor, Allison Randal, helped me get all of those set up and endured my frequent emails without complaining. After Allison went on to other things, Simon St. Laurent was extremely helpful in the role of editor and inside guy at O’Reilly, patiently waiting for the right phase of the moon to suggest another update. Zan McQuade and Jill Leonard, both from O’Reilly, provided enthusiastic support to publish the present edition.

From Tom

I’ve got to echo Randal’s thanks to everyone at O’Reilly. For the third edition of this book, Linda Mui was our editor, and I still thank her, for her patience in pointing out which jokes and footnotes were most excessive, while pointing out that she is in no way to blame for the ones that remain. Both she and Randal have guided me through the process of writing, and I am grateful. In a previous edition, Allison Randal took charge; then Simon St. Laurent became the editor. My thanks go to each of them in recognition of their unique contributions.

And another echo with regard to Randal and the other Stonehenge trainers, who hardly ever complained when I unexpectedly updated the course materials to try out a new teaching technique. You folks have contributed many different viewpoints on teaching methods that I would never have seen.

For many years, I worked at the Oregon Museum of Science and Industry (OMSI), and I’d like to thank the folks there for letting me hone my teach-

ing skills as I learned to build a joke or two into every activity, explosion, or dissection.

To the many folks on Usenet who have given me your appreciation and encouragement for my contributions there, thanks. As always, I hope this helps.

Also to my many students, who have shown me with their questions (and befuddled looks) when I needed to try a new way of expressing a concept. I hope that the present edition helps to relieve any remaining puzzlement.

Of course, deep thanks are due especially to my coauthor, Randal, for giving me the freedom to try various ways of presenting the material both in the classroom and here in the book, as well as for the push to make this material into a book in the first place. And without fail, I must say that I am indeed inspired by your ongoing work to ensure that no one else becomes ensnared by the legal troubles that have stolen so much of your time and energy; you're a fine example.

To my wife, Jenna, thanks for being a cat person, and everything thereafter.

From All of Us

We also thank our “correcters.” The O'Reilly Media system is one of continuous publishing. As people find mistakes, we try to fix them immediately. When it's time to print more books, or release a new ebook, you get the benefit of those post-publication corrections. For those, we thank Egon Choroba, Cody Czesler, Kieren Diment, Charles Evans, Keith Howanitz, Susan Malter, Enrique Nell, Peter O'Neill, Povl Ole Haarlev Olsen, Flavio Poletti, Rob Reed, Alan Rocker, Dylan Scott, Peter Scott, Shaun Smiley, John Trammel, Emma Urquhart, John Wiersba, Danny Woods, and Zhenyo Zhou. Additionally, David Farrell, André Philipp, Grzegorz Szpetkowski, and Ali Sinan Ünür carefully read through the entire book to find all (we hope) mistakes and lies. We learned from each of them.

Thanks also to our many students who have let us know what parts of the course material have needed improvement over the years. It's because of you that we're all so proud of it today.

Thanks to the many Perl Mongers who have made us feel at home as we've visited your cities. Let's do it again sometime.

And finally, our sincerest thanks to our friend Larry Wall, for having the wisdom to share his really cool and powerful toys with the rest of the world so that we can all get our work done just a little bit faster, easier, and with more fun.

- 1** In the contract, I retained the rights to the exercises, hoping someday to reuse them in some other way, like in the magazine columns I was writing at the time. The exercises are the only things that leapt from the Taos course to the book.
- 2** My Taos contract had a noncompete clause, so I had to stay out of Silicon Valley with any similar courses, which I respected for many years.
- 3** I remember that date very well, because it was also the day I was arrested at my home for computer-related activities around my Intel contract, a series of felony charges for which I was later convicted.

[Support](#) [Sign Out](#)