

<<Perl进阶>>

图书基本信息

书名：<<Perl进阶>>

13位ISBN编号：9787564138882

10位ISBN编号：7564138882

出版时间：2013-1

出版时间：东南大学出版社

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Perl进阶>>

内容概要

《Perl进阶(影印版)(第2版)》内容包括：包和命名空间；引用和作用域，包括正则表达式引用；操作复杂数据结构；面向对象编程；编写和使用模块；测试Perl代码；对CPAN作出贡献.....与《Perl语言入门》一样，书中中的素材紧密结合了作者自1991年起开始教授的广受欢迎的Perl入门课程。

<<Perl进阶>>

作者简介

作者：（美国）施瓦茨（Randal L.Schwartz）（美国）福瓦（Brian d Foy）（美国）菲尼克斯（Tom Pboenix） Randal L. Schwartz擅长软件设计、系统管理、安全、技术协作和培训。

他是多本必备书籍的作者之一，包括《Perl语言入门》、《Perl编程》、《Perl进阶》以及《精通Perl》（以上均由O'Reilly出版）。

brian d foy是一名多产的Perl讲师和作家，他创办了《The PerlReview》期刊来帮助用户使用和理解Perl。

他也是《Perl语言入门》、《Perl进阶》、《精通Perl》以及《EffectivePerl Programming》（Addison—Wesley出版）的作者之一。

Tom Phoenix在StonehengeConsulting Services公司讲授Perl语言。

他同时也在Usenet的comp.lang.perl.misc和complang.pefl.moderated新闻组回答问题。

他是《Perl进阶》的作者之一，也是Perl的参与者。

<<Perl进阶>>

书籍目录

Foreword Preface 1. introduction What Should You Know Already? strict and warnings Perl v5.14 A Note on Versions What About All Those Footnotes? What's With the Exercises? How to Get Help What If I'm a Perl Course Instructor? Exercises 2. Using Modules The Standard Distribution Exploring CPAN Using Modules Functional Interfaces Selecting What to Import Object-Oriented Interfaces A More Typical Object-Oriented Module: Math::BigInt Fancier Output with Modules What's in Core? The Comprehensive Perl Archive Network Installing Modules from CPAN CPANminus Installing Modules Manually Setting the Path at the Right Time Setting the Path Outside the Program Extending @INC with PERLSLIB Extending @INC on the Command Line local::lib Exercises 3. Intermediate Foundations. List Operators List Filtering with grep Transforming Lists with map Trapping Errors with eval Dynamic Code with eval The do Block Exercises 4. Introduction to References Doing the Same Task on Many Arrays PEGS: Perl Graphical Structures Taking a Reference to an Array Dereferencing the Array Reference Getting Our Braces Off Modifying the Array Nested Data Structures Simplifying Nested Element References with Arrows References to Hashes Checking Reference Types Exercises 5. References and Scoping More than One Reference to Data What If That Was the Name? Reference Counting and Nested Data Structures When Reference Counting Goes Bad Creating an Anonymous Array Directly Creating an Anonymous Hash Autovivification Autovivification and Hashes Exercises Manipulating Complex Data Structures Using the Debugger to View Complex Data Viewing Complex Data with Data::Dumper Other Dumpers Marshalling Data Storing Complex Data with Storable YAML JSON Using the map and grep Operators Applying a Bit of Indirection Selecting and Altering Complex Data Exercises Subroutine References Referencing a Named Subroutine Anonymous Subroutines Callbacks Closures Returning a Subroutine from a Subroutine Closure Variables as Inputs Closure Variables as Static Local Variables state Variables Finding Out Who We Are Enchanting Subroutines Dumping Closures Exercise 8. Filehandle References The Old Way The Improved Way Filehandles to Strings Processing Strings Line by Line Collections of Filehandles IO::Handle and Friends IO::File IO::Scalar IO::Tee IO::Pipe IO::Null and IO::Interactive Directory Handles Directory Handle References Exercises Regular Expression References Before Regular Expression References Precompiled Patterns Regular Expression Options Applying Regex References Regexpes as Scalars Build Up Regular Expressions Regex-Creating Modules Using Common Patterns Assembling Regular Expressions Exercises 10. Practical Reference Tricks Fancier Sorting Sorting with Indices Sorting Efficiently The Schwartzian Transform Multilevel Sort with the Schwartzian Transform Recursively Defined Data Building Recursively Defined Data Displaying Recursively Defined Data Avoiding Recursion The Breadth-First Solution Exercises 11. Building Larger Programs The Cure for the Common Code Inserting Code with eval Using do Using require The Problem of Namespace Collisions Packages as Namespace Separators Scope of a Package Directive Packages and Lexicals Package Blocks Exercises 12. Creating Your Own Perl Distribution Perl's Two Build Systems Inside Makefile.PL Inside Build.PL Our First Distribution h2xs Module: :Starter Custom Templates Inside Your Perl Distribution The META File Adding Additional Modules 13. Introduction to Objects 14. Introduction to Testing 15. Objects with Data 16. Some Advanced Object Topics 17. Exporter 18. Object Destruction 19. Introduction to Moose 20. Advanced Testing 21. Contributing to CPAN Appendix: Answers to Exercises Index of Modules in this Book Index

章节摘录

版权页： 插图： Sorting Efficiently As the Professor tries to maintain the community computing facility (built entirely out of bamboo, coconuts, and pineapples, and powered by a certified Perl-hacking monkey), he continues to discover that people are leaving entirely too much data on the single monkey-powered filesystem, so he decides to print a list of offenders. The Professor has written a subroutine called `ask_monkey_about`, which, given a cast-away's name, returns the number of pineapples of storage they use. We have to ask the monkey because he's in charge of the pineapples. An initial naive approach to find the offenders from greatest to least might be something like: In theory, this would be fine. For the first pair of names (Gilligan and Skipper), we ask the monkey "How many pineapples does Gilligan have?" and "How many pineapples does Skipper have?" We get back two values from the monkey and use them to order Gilligan and Skipper in the final list. However, at some point, we have to compare the number of pineapples that Gilligan has with another castaway as well. For example, suppose the pair is Ginger and Gilligan. We ask the monkey about Ginger, get a number back, and then ask the monkey about Gilligan... again. This will probably annoy the monkey a bit, since we already asked. But we need to ask for each value two, three, or maybe even four times just to put these seven values into order. This can be a problem because it irritates the monkey. How do we keep the number of monkey requests to a minimum? Well, we can build a table first. We use a map with seven inputs and seven outputs, turning each castaway item into a separate array reference, with each referenced array consisting of the cast-away name and the pineapple count reported by the monkey.

<<Perl进阶>>

编辑推荐

《Perl进阶(影印版)(第2版)》将继续你的Perl学习之旅。
通过《Perl进阶》，你将不再编写简单的脚本，而是使用那些让Perl成为通用语言的特性来开发更为庞大的程序。
这本轻松但又完备的指南将为你介绍模块、复杂数据结构以及面向对象编程。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>