

Go Programming Language, The (Addison-Wesley Professional Computing Series)

by

Brian W. Kernighan

Look inside ↴



EBOOK DOWNLOAD

Synopsis

The Go Programming Language is the authoritative resource for any programmer who wants to learn Go. It shows how to write clear and idiomatic Go to solve real-world problems. The book does not assume prior knowledge of Go nor experience with any specific language, so you'll find it accessible whether you're most comfortable with JavaScript, Ruby, Python, Java, or C++. The first chapter is a tutorial on the basic concepts of Go, introduced through programs for file I/O and text processing, simple graphics, and web clients and servers. Early chapters cover the structural elements of Go programs: syntax, control flow, data types, and the organization of a program into packages, files, and functions. The examples illustrate many packages from the standard library and show how to create new ones of your own. Later chapters explain the package mechanism in more detail, and how to build, test, and maintain projects using the go tool. The chapters on methods and interfaces introduce Go's unconventional approach to object-oriented programming, in which methods can be declared on any type and interfaces are implicitly satisfied. They explain the key principles of encapsulation, composition, and substitutability using realistic examples. Two chapters on concurrency present in-depth approaches to this increasingly important topic. The first, which covers the basic mechanisms of goroutines and channels, illustrates the style known as communicating sequential processes for which Go is renowned. The second covers more traditional aspects of concurrency with shared variables. These chapters provide a solid foundation for programmers encountering concurrency for the first time. The final two chapters explore lower-level features of Go. One covers the art of metaprogramming using reflection. The other shows how to use the unsafe package to step outside the type system for special situations, and how to use the cgo tool to create Go bindings for C libraries. The book features hundreds of interesting and practical examples of well-written Go code that cover the whole language, its most important packages, and a wide range of applications. Each chapter has exercises to test your understanding and explore extensions and alternatives. Source code is freely available for download from <http://gopl.io/> and may be conveniently fetched, built, and installed using the go get command.

go programming language the addisonwesley professional computing series go programming language the addisonwesley professional computing series pdf the go programming language addisonwesley professional computing series pdf download go programming language addison wesley pdf the go programming language 2nd edition the go programming language alan donovan pdf the go programming language exercises what is the go programming language used for the go programming language donovan pdf go programming language the go programming language for dummies go programming language for android go programming language for beginners go programming language for dummies pdf the go programming language book the go programming language ebook go programming language for ai go programming language for loop go programming language for data science go programming language for 32 bit go programming language for web development go network programming

TCP/IP Illustrated, Vol. 2: The Implementation (Addison-Wesley Professional Computing Series) UNIX System V Network Programming (Addison-Wesley Professional Computing Series) Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library (Addison-Wesley Professional Computing Series) TCP/IP Illustrated, Volume 1: The Protocols (Addison-Wesley Professional Computing Series) Elemental Design Patterns (Addison-Wesley Professional Computing Series) Mathematical Foundations of Computer Networking (Addison-Wesley Professional Computing Series) Advanced Programming in the UNIX Environment (Addison-Wesley Professional Computing Series) Go Programming Language, The (Addison-Wesley Professional Computing Series) BPF Performance Tools (Addison-Wesley Professional Computing Series) Large-Scale C++ Volume I: Process and Architecture (Addison-Wesley Professional Computing Series) Understanding Software Dynamics (Addison-Wesley Professional Computing Series) Learning Go: An Idiomatic Approach to Real-World Go Programming Mastering Go: Harness the power of Go to build professional utilities and concurrent servers and services, 3rd Edition Concurrency in Go: Tools and Techniques for Developers Distributed Services with Go: Your Guide to Reliable, Scalable, and Maintainable Systems Introducing Go: Build Reliable, Scalable Programs Go in Practice: Includes 70 Techniques Mastering Go: Create Golang production applications using network libraries, concurrency, machine learning, and advanced data structures, 2nd Edition Network Programming with Go: Code Secure and Reliable Network Services from Scratch

What people say about this book

L. Hall, "Carefully crafted to make you competent - if you are patient enough give yourself the time to learn. Not a book for the impatient; but if you're a novice, this is the book to trust to take you through from beginner to competence and well on your way to proficiency. I've been a professional C/UNIX developer since the mid-80's and Java and Python for the past 20 years. Don't get me wrong: Go is not a difficult language to learn; but I thought I could pick it up in a few hours. There are serious ground-shattering differences between Go and everything else. Although it borrows liberally from C, Java, and Python - it really is a different approach because the language authors aren't afraid to address the shortcomings made in those and other languages for the sake of conforming with the ideas long held to be "norms". Case in point: Go builds statically-bound executables. No more runtime dependency woes from mis-matched DLL / .so versions. Another example: a radically different approach to polymorphism and encapsulation leading to an easier and cleaner object model than any other. The more I learn about Go, the more I am convinced that it will eventually overtake C/C++ as the defacto standard for system level development - and may even challenge Java and the dynamic languages for business-critical applications."

Michael Davidson, "Great book, but the Kindle edition sucks.. 5 stars for the book, 2 stars for the Kindle edition. I own both the print and kindle editions of this book. The book itself is great but I am very disappointed in the Kindle edition. This is the first time that I have attempted to read a text book on a Kindle and I found that navigation within the book was extremely cumbersome. To add insult to injury all of the online code examples are line wrapped because of the Kindle formatting and there are links to out of line versions that are correctly formatted (but in a horrible font). Honestly, is that the best that Kindle can do? I would have been better off with a PDF file of the book."

Clinton Begin, "Excellent book. It did what the others didn't. I tried another Go book, before tossing it aside in favor of this one. Excellent book. It did what the others didn't: it explained "why". Go is a weird language. So if the reasoning behind the decisions is left out, then the language is hard to swallow. But with proper explanation of the reasoning, one can see the wisdom of Go and forgive some of its odd design and syntax decisions."

G.S. Fordham, "Pretty good language, VERY good book.. Very good book to get me ready to use Golang. But not just that: it also helps get people into better coding habits. Golang is a language based on best practices, and I feel I not only learned a new language this past Summer, but I also learned more about being a good programmer in general. I would definitely consider using this language in bigger projects in the future, and I have already adopted several practices they listed across other languages I use. The only things I am not a fan of with this language are the ways it implements public vs private data members and interfaces. I feel the variable name

casing should not determine public or private members, because it is less explicit. I also did try a few things out, after reading, and I came across some nasty bugs that were not the easiest to track, due to the implicit interface implementation -- things would go from implementing part of my interface and not another to getting changed and implementing them in reverse, even with only the parts in question being changed; I would much rather say "implements x, y, z", and the compiler would know EXACTLY what I'm trying to do."

Kent R. Spillner, "The best introduction to the language. Personally, I haven't quite made up my mind about the Go language itself just yet, but this book is undeniably fantastic! Every programming language needs an introductory book like this: clear and concise, thorough coverage of the entire language without being dense or boring, and puts as much effort into teaching good taste & style as learning the language itself. (Obviously, we'd expect nothing less from anything bearing Brian Kernighan's name on the cover)I especially appreciated the way this book was organized: the very first chapter starts with several illustrative examples before even introducing the language itself. Then the next several chapters cover all of the basic & advanced features of the language. The final few chapters cover some of the other concerns of modern professional programmers beyond just the language itself: the build environment, packaging, testing, etc.If you want to learn the Go programming language, or if you already know how to program in Go but want to improve your understanding of the language, then this is the first & only book you'll need."

laymonk, "Encyclopedic Coverage Of All Important Aspects. If you are coming from another language and need a rapid coverage of the basics, this is probably not the best book to start with. You are better off using one of the numerous online resources for that. I personally found the free 'The Little Go Book' as a perfect fast-paced first scan of the language .This book is best as your second (or your first, if you are not in a great deal of a hurry), and more thorough coverage of the language, and as a ready reference while you tackle projects.If I had any complaints, it would be a wish that they used bigger print in the paperback ... and that an eBook version is made available and discounted for owners of the paperback.Highly recommended."

Ibbie Anderson, "Relevant, interesting, and fun!. Over the years I've read many programming books and this one I have to say is a standout book! The exercises, while sometimes pretty in-depth, are varied and hugely interesting. The content of the book is relevant, elegantly put forward, and full of detail. I'm enjoying this book so much I'm going through it page by page and doing all of the exercises.Maybe some will want a cheat-sheet type of book containing less detail but this book will actually help you grok Go. The authors are also approachable and helpful. Highly recommended!"

EliasVO, "The book. K&R C (second edition) still is the standard for books on the C

programming language. This is the same thing, for golang. Good book. It assumes some prior programming experience (eg it doesn't bother explaining scopes, functions, if's and else's). Quite refreshing to have a book that just gets on with it.”

The book by Brian W. Kernighan has a rating of 5 out of 4.7. 1,034 people have provided feedback.

Book Information

Language: English

File size: 33774 KB

Text-to-Speech: Enabled

Screen Reader: Supported

Enhanced typesetting: Enabled

X-Ray: Not Enabled

Word Wise: Not Enabled

Print length: 1390 pages

Lending: Not Enabled

Simultaneous device usage: Up to 5 simultaneous devices, per publisher limits

[DMCA](#)