



# Gorgo Go for Java Programmers

June 11, 2026



# Contents

<b>0</b>	<b>How to use this booklet</b>	<b>1</b>
	Code Review Rules . . . . .	1
	Idioms . . . . .	1
	Callouts . . . . .	1
	Function Signatures . . . . .	1
	Try It . . . . .	2
	Exercises . . . . .	2



# Chapter 0

## How to use this booklet

This is a short booklet to help a Java programmer learn Go.

Each chapter is meant to help you understand a topic, but you will still want to reference API descriptions for more specifics of the parameters and operating conditions. Hopefully, you'll have the context you need to understand API documentation.

### Code Review Rules

Go is an opinionated language — they don't even let us indent with spaces!!! You will find the Code Review Rules for Go in the appendix. These rules help developers understand the best practices for writing Go code. When we refer to these rules we will use the bold italics version of the [*short-rule-name*].

### Idioms

Idioms are found in every language — spoken and programmed. Go takes idioms next level. You are going to see *idiom* and *idiomatic* a lot, so it helps to get comfortable with the word. Wikipedia defines idiom as (Wikipedia contributors 2024)

a commonly-used way to code a relatively small construct in a particular programming context

You could also think of it as best practices. I like to think of it as “if you don't do it the idiomatic way, you risk looking like an idiot”.

### Callouts

**Tips** call out details that you need to pay special attention to. **Traps** warn you of common mistakes. **Wut** calls out a detail that is counter-intuitive, so make sure you pay attention.

### Function Signatures

When this book introduces a new function, it shows the function's **signature**. In Go, a function's signature is its parameter and result types — unlike Java, the return type is part of the signature. We show the signature together with the function's name so you know what it is called, what goes in, and what comes out. For example:

```
func Abs(n int) int
```

This tells you that `Abs` takes one `int` parameter and returns an `int`. You do not need to understand every detail the first time you see it, but it gives you three things at a glance: what the function is called, what goes in, and what comes out. As you work through API documentation on your own, signatures are the first thing you will look at, so getting comfortable reading them early pays off.

## Try It

As the intro to the most amazing programming language book ever written (Kernighan and Ritchie 1988) starts out:

The only way to learn a new programming language is by writing programs in it.

You need to write some code. Make sure you try writing some programs from scratch. At the end of each chapter is a starter program that you can type in and modify to play with. Don't use it as an excuse to avoid writing some of your own starter programs. It's the only way to master a language.

## Exercises

Don't skip the exercises at the end of the chapters. You can get the answer key, but don't look at the answer key before you work out the answer yourself. If you look at the answer key first, the concepts will not sink in. Kernighan, Brian W., and Dennis M. Ritchie. 1988. *The C Programming Language*. 2nd ed. Prentice Hall. Wikipedia contributors. 2024. "Programming idiom." Wikipedia. [https://en.wikipedia.org/wiki/Programming\\_idiom](https://en.wikipedia.org/wiki/Programming_idiom).