

SAMS  
**Teach  
Yourself**

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的良好起点
- “Read Less, Do More”（精读多练）的教学理念
- 以示例引导读者完成常见的任务

每章内容针对初学者精心设计，**1**小时轻松阅读学习，  
**24**小时彻底掌握关键知识

每章**案例与练习题**助你轻松完成常见任务，  
通过**实践**提高应用技能，巩固所学知识

# Go 语言

## 入门经典

[英] 乔治·奥尔波 (George Orno) 著  
张海燕 译

# Go 语言 入门经典

[英] 乔治·奥尔波 (George Orno) 著  
张海燕 译

人民邮电出版社

北京

## 图书在版编目 (C I P) 数据

Go语言入门经典 / (英) 乔治·奥尔波  
(George Orngo) 著 ; 张海燕译. — 北京 : 人民邮电出版社, 2018. 8

ISBN 978-7-115-48503-8

I. ①G… II. ①乔… ②张… III. ①程序语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2018)第125445号

## 版权声明

George Orngo:Sams Teach Yourself Go in 24 Hours

ISBN: 978-0-672-33803-8

Copyright © 2018 by Pearson Education, Inc.

Authorized translation from the English languages edition published by Pearson Education, Inc.

All rights reserved.

本书中文简体字版由美国 Pearson 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

- 
- ◆ 著 [英] 乔治·奥尔波 (George Orngo)
  - 译 张海燕
  - 责任编辑 陈聪聪
  - 责任印制 焦志炜
  
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
  - 邮编 100164 电子邮件 315@ptpress.com.cn
  - 网址 <http://www.ptpress.com.cn>
  - 涿州市京南印刷厂印刷
  
  - ◆ 开本: 787×1092 1/16
  - 印张: 17.5
  - 字数: 425 千字 2018 年 8 月第 1 版
  - 印数: 1-2 400 册 2018 年 8 月河北第 1 次印刷
- 著作权合同登记号 图字: 01-2017-9041 号
- 

定价: 59.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

# 内容提要

Go 语言是谷歌推出的一种全新的编程语言，旨在不损失应用程序性能的情况下降低代码的复杂性，具有“部署简单、并发性好、语言设计良好、执行性能好”等优势，目前国内诸多 IT 公司均已采用 Go 语言开发项目。

本书分为 24 章，讲解了使用 Go 语言编写高质量程序的方法，其内容涵盖了 Go 语言特性和标准库安装包，Go 与 JavaScript 的对比，Go 命令行工具，Go 中的基本概念（比如类型、变量、函数、控制结构、指针、接口等）、错误处理、Goroutine 和通道、Go 代码测试、使用 Go 编写 HTTP 客户端与服务器、处理 JSON 和文件、部署 Go 代码等。

本书适合想要掌握 Go 语言的零基础读者以及对 Go 语言感兴趣的程序员学习，还可作为高等院校教授 Go 语言课程的教材。

# 作者简介

---

**George Ornbo** 是一位软件工程师、博主和作家，拥有 14 年的软件开发经验，其客户既有初创公司，也有大型企业客户；熟悉众多编程语言、UNIX 和 Web 底层协议。当前供职于伦敦的一家区块链初创公司。

# 献辞

谨以此书献给 Bea 和 Fin，有你们俩真是我天大的福分！

# 致谢

感谢 Laura Lewin 和 Pearson 编辑团队给我写作本书的机会。还要感谢 Sheri Replin 所做的出色工作！

感谢 Bala Natarajan 和 Yoshiki Shibata 为本书所做的技术审校工作，感谢他们提出的大量卓越的改进建议。

感谢 Robert Griesemer、Rob Pike 和 Ken Thompson 设计出了 Go 语言。

# 前言

Go (Golang) 语言是编程语言设计的又一次尝试，是对类 C 语言的重大改进。它让您能够访问底层操作系统，还提供了强大的网络编程和并发编程支持。

Go 语言用途众多，其中包括如下几种。

- 网络编程。
- 系统编程。
- 并发编程。
- 分布式编程。

很多重要的开源项目都是使用 Go 语言开发的，其中包括 Go-Ethereum、Terraform、Kubernetes 和 Docker。Go 语言给开源界带来了重大影响，有望提高开源项目的成功率。

## 本书读者对象

本书不要求读者具备任何编程或计算机方面的经验，但如果对编程有基本的认识将大有裨益。由于 Go 代码主要是从终端运行的，因此熟悉终端以及如何在其中执行基本命令是十分必要的。最后，鉴于 Go 语言常用于系统编程和网络编程，因此对互联网的工作原理有些了解也将很有帮助，虽然这并非必要的。

## 为何要学习 Go 语言

如果您要创建系统程序或基于网络的程序，Go 语言是很不错的选择。作为一种相对较新的语言，它是由经验丰富且受人尊敬的计算机科学家设计的，旨在应对创建大型并发网络程序面临的挑战。如果您觉得 Java 或 C 语言的语法导致编程很难，那么 Go 语言将可能提供更好的体验。对具备诸如 Ruby、Python、JavaScript 等动态语言使用经验的程序员来说，Go 语言提供了类型安全，同时又不像传统语言那么死板。



## 组织结构

本书首先介绍 Go 语言基础知识，包括搭建 Go 语言编程环境以及运行第一个 Go 程序。接下来介绍一些重要的 Go 语言知识，包括字符串、函数、结构体和方法。您将学会如何使用 Goroutine 和通道，这些是 Go 语言特有的功能，它们避免了并发编程的大部分难题。

然后，您将学习如何调试和测试 Go 语言代码，并学会一些帮助编写独具 Go 语言风格代码的技巧。

接下来，您将学习如何编写基本的命令程序、HTTP 服务器和 HTTP 客户端，并学习如何处理 JSON 数据和文件。

最后，您将学习正则表达式相关的知识、如何处理时间以及如何将 Go 应用程序部署到生产环境中。

## 代码示例

本书每章都有多个代码示例，旨在帮助您学习 Go 语言，它们与正文一样重要，强烈建议您在阅读过程中运行这些示例代码。本书的示例代码可在异步社区本书页面中下载。

# 资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

## 配套资源

本书提供如下资源：

- 本书配套资源请到异步社区的本书购买页面中下载。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

## 提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，单击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。



The screenshot shows a web form for submitting勘误 (勘误). At the top, there are three tabs: '详细信息' (Detailed Information), '写书评' (Write a Review), and '提交勘误' (Submit勘误), with the latter being the active tab. Below the tabs, there are three input fields: '页码:' (Page Number), '页内位置 (行数):' (Position within page (Line Number)), and '勘误次数:' (Number of勘误). Below these fields is a large text area for entering the勘误 details. At the bottom right of the form, there is a '字数统计' (Character Count) label and a '提交' (Submit) button.

## 扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



## 与我们联系

我们的联系邮箱是 [contact@epubit.com.cn](mailto:contact@epubit.com.cn)。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 [www.epubit.com/selfpublish/submission](http://www.epubit.com/selfpublish/submission) 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

## 关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



异步社区



微信服务号

# 目 录

<b>第1章 起步</b> .....	1	2.4.1 带符号整数和无符号整数	13
1.1 Go 简介.....	1	2.4.2 浮点数.....	14
1.1.1 Go 语言简史.....	1	2.4.3 字符串.....	14
1.1.2 Go 是编译型语言.....	2	2.4.4 数组.....	14
1.2 安装 Go.....	2	2.5 检查变量的类型.....	15
1.2.1 在 Windows 系统中安装.....	3	2.6 类型转换.....	16
1.2.2 在 macOS 或 Linux 系统中 安装.....	4	2.7 小结.....	17
1.3 设置环境.....	4	2.8 问与答.....	17
1.4 编写第一个 Go 程序—— Hello World.....	5	2.9 作业.....	17
1.4.1 使用 go run 编译并运行 程序.....	6	2.9.1 小测验.....	17
1.4.2 Go 吉祥物.....	6	2.9.2 答案.....	17
1.5 小结.....	6	2.10 练习.....	18
1.6 问与答.....	7	<b>第3章 理解变量</b> .....	19
1.7 作业.....	7	3.1 变量是什么.....	19
1.7.1 小测验.....	7	3.2 快捷变量声明.....	21
1.7.2 答案.....	7	3.3 理解变量和零值.....	21
1.8 练习.....	7	3.4 编写简短变量声明.....	22
<b>第2章 理解类型</b> .....	8	3.5 变量声明方式.....	23
2.1 数据类型是什么.....	8	3.6 理解变量作用域.....	24
2.2 区分静态类型和动态类型.....	8	3.7 使用指针.....	25
2.3 使用布尔类型.....	11	3.8 声明常量.....	27
2.4 理解数值类型.....	12	3.9 小结.....	28
		3.10 问与答.....	28
		3.11 作业.....	29
		3.11.1 小测验.....	29

3.11.2 答案 .....	29	6.2 使用切片 .....	56
3.12 练习 .....	29	6.2.1 在切片中添加元素 .....	56
<b>第4章 使用函数</b> .....	<b>30</b>	6.2.2 从切片中删除元素 .....	58
4.1 函数是什么 .....	30	6.2.3 复制切片中的元素 .....	58
4.1.1 函数的结构 .....	30	6.3 使用映射 .....	59
4.1.2 返回单个值 .....	31	从映射中删除元素 .....	60
4.1.3 返回多个值 .....	32	6.4 小结 .....	61
4.2 定义不定参数函数 .....	33	6.5 问与答 .....	61
4.3 使用具名返回值 .....	34	6.6 作业 .....	62
4.4 使用递归函数 .....	35	6.6.1 小测验 .....	62
4.5 将函数作为值传递 .....	36	6.6.2 答案 .....	62
4.6 小结 .....	38	6.7 练习 .....	62
4.7 问与答 .....	38	<b>第7章 使用结构体和指针</b> .....	<b>63</b>
4.8 作业 .....	38	7.1 结构体是什么 .....	63
4.8.1 小测验 .....	38	7.2 创建结构体 .....	65
4.8.2 答案 .....	38	7.3 嵌套结构体 .....	68
4.9 练习 .....	39	7.4 自定义结构体数据字段的 默认值 .....	69
<b>第5章 控制流程</b> .....	<b>40</b>	7.5 比较结构体 .....	71
5.1 使用 if 语句 .....	40	7.6 理解公有和私有值 .....	72
5.2 使用 else 语句 .....	42	7.7 区分指针引用和值 引用 .....	73
5.3 使用 else if 语句 .....	43	7.8 小结 .....	75
5.4 使用比较运算符 .....	44	7.9 问与答 .....	75
5.5 使用算术运算符 .....	45	7.10 作业 .....	76
5.6 使用逻辑运算符 .....	45	7.10.1 小测验 .....	76
5.7 使用 switch 语句 .....	46	7.10.2 答案 .....	76
5.8 使用 for 语句进行循环 .....	47	7.11 练习 .....	76
5.8.1 包含初始化语句和后续 语句的 for 语句 .....	49	<b>第8章 创建方法和接口</b> .....	<b>77</b>
5.8.2 包含 range 子句的 for 语句 .....	49	8.1 使用方法 .....	77
5.9 使用 defer 语句 .....	50	8.2 创建方法集 .....	79
5.10 小结 .....	52	8.3 使用方法和指针 .....	80
5.11 问与答 .....	52	8.4 使用接口 .....	83
5.12 作业 .....	53	8.5 小结 .....	86
5.12.1 小测验 .....	53	8.6 问与答 .....	86
5.12.2 答案 .....	53	8.7 作业 .....	87
5.13 练习 .....	53	8.7.1 小测验 .....	87
<b>第6章 数组、切片和映射</b> .....	<b>54</b>	8.7.2 答案 .....	87
6.1 使用数组 .....	54	8.8 练习 .....	87

<b>第 9 章 使用字符串</b> .....	88	11.8 问与答 .....	114
9.1 创建字符串字面量 .....	88	11.9 作业 .....	115
9.2 理解 rune 字面量 .....	89	11.9.1 小测验 .....	115
9.3 拼接字符串 .....	90	11.9.2 答案 .....	115
9.3.1 使用缓冲区拼接字符串 .....	92	11.10 练习 .....	115
9.3.2 理解字符串是什么 .....	93	<b>第 12 章 通道简介</b> .....	116
9.3.3 处理字符串 .....	94	12.1 使用通道 .....	116
9.4 小结 .....	97	12.2 使用缓冲通道 .....	119
9.5 问与答 .....	97	12.3 阻塞和流程控制 .....	120
9.6 作业 .....	97	12.4 将通道用作函数参数 .....	123
9.6.1 小测验 .....	97	12.5 使用 select 语句 .....	123
9.6.2 答案 .....	97	12.6 退出通道 .....	126
9.7 练习 .....	98	12.7 小结 .....	128
<b>第 10 章 处理错误</b> .....	99	12.8 问与答 .....	128
10.1 错误处理及 Go 语言的 独特之处 .....	99	12.9 作业 .....	128
10.2 理解错误类型 .....	101	12.9.1 小测验 .....	129
10.3 创建错误 .....	101	12.9.2 答案 .....	129
10.4 设置错误的格式 .....	102	12.10 练习 .....	129
10.5 从函数返回错误 .....	103	<b>第 13 章 使用包实现代码重用</b> .....	130
10.6 错误和可用性 .....	104	13.1 导入包 .....	130
10.7 慎用 panic .....	104	13.2 理解包的用途 .....	131
10.8 小结 .....	106	13.3 使用第三方包 .....	132
10.9 问与答 .....	106	13.4 安装第三方包 .....	132
10.10 作业 .....	106	13.5 管理第三方依赖 .....	133
10.10.1 小测验 .....	106	13.6 创建包 .....	135
10.10.2 答案 .....	106	13.7 小结 .....	137
10.11 练习 .....	107	13.8 问与答 .....	137
<b>第 11 章 使用 Goroutine</b> .....	108	13.9 作业 .....	137
11.1 理解并发 .....	108	13.9.1 小测验 .....	137
11.2 并发和并行 .....	110	13.9.2 答案 .....	138
11.3 通过 Web 浏览器来 理解并发 .....	110	13.10 练习 .....	138
11.4 阻塞和非阻塞代码 .....	111	<b>第 14 章 Go 语言命名约定</b> .....	139
11.5 使用 Goroutine 处理并发 操作 .....	112	14.1 Go 代码格式设置 .....	139
11.6 定义 Goroutine .....	114	14.2 使用 gofmt .....	140
11.7 小结 .....	114	14.3 配置文本编辑器 .....	141
		14.4 命名约定 .....	142
		14.5 使用 golint .....	143
		14.6 使用 godoc .....	144

14.7	工作流程自动化	147
14.8	小结	148
14.9	问与答	149
14.10	作业	149
14.10.1	小测验	149
14.10.2	答案	149
14.11	练习	149
<b>第 15 章</b>	<b>测试和性能</b>	<b>150</b>
15.1	测试：软件开发最重要的方面	150
15.1.1	单元测试	151
15.1.2	集成测试	151
15.1.3	功能测试	151
15.1.4	测试驱动开发	151
15.2	testing 包	151
15.3	运行表格驱动测试	154
15.4	基准测试	156
15.5	提供测试覆盖率	158
15.6	小结	159
15.7	问与答	159
15.8	作业	159
15.8.1	小测验	159
15.8.2	答案	159
15.9	练习	160
<b>第 16 章</b>	<b>调试</b>	<b>161</b>
16.1	日志	161
16.2	打印数据	164
16.3	使用 fmt 包	165
16.4	使用 Delve	168
16.5	使用 gdb	170
16.6	小结	171
16.7	问与答	171
16.8	作业	171
16.8.1	小测验	171
16.8.2	答案	172
16.9	练习	172
<b>第 17 章</b>	<b>使用命令行程序</b>	<b>173</b>
17.1	操作输入和输出	174
17.2	访问命令行参数	174
17.3	分析命令行标志	176
17.4	指定标志的类型	177
17.5	自定义帮助文本	178
17.6	创建子命令	179
17.7	POSIX 兼容性	182
17.8	安装和分享命令行程序	182
17.9	小结	184
17.10	问与答	184
17.11	作业	184
17.11.1	小测验	185
17.11.2	答案	185
17.12	练习	185
<b>第 18 章</b>	<b>创建 HTTP 服务器</b>	<b>186</b>
18.1	通过 Hello World Web 服务器宣告您的存在	186
18.2	查看请求和响应	187
18.2.1	使用 curl 发出请求	188
18.2.2	详谈路由	189
18.3	使用处理程序函数	189
18.4	处理 404 错误	190
18.5	设置报头	191
18.6	响应以不同类型的内容	192
18.7	响应不同类型的请求	194
18.8	获取 GET 和 POST 请求中的数据	195
18.9	小结	197
18.10	问与答	198
18.11	作业	198
18.11.1	小测验	198
18.11.2	答案	198
18.12	练习	199
<b>第 19 章</b>	<b>创建 HTTP 客户端</b>	<b>200</b>
19.1	理解 HTTP	200
19.2	发出 GET 请求	201
19.3	发出 POST 请求	202
19.4	进一步控制 HTTP 请求	204
19.5	调试 HTTP 请求	205
19.6	处理超时	207

19.7	小结	208	22.1	定义正则表达式	236
19.8	问与答	209	22.2	熟悉正则表达式语法	238
19.9	作业	209	22.3	使用正则表达式验证数据	239
19.9.1	小测验	209	22.4	使用正则表达式来变换数据	240
19.9.2	答案	209	22.5	小结	241
19.10	练习	210	22.6	问与答	241
<b>第 20 章</b>	<b>处理 JSON</b>	<b>211</b>	22.7	作业	241
20.1	JSON 简介	211	22.7.1	小测验	242
20.2	使用 JSON API	213	22.7.2	答案	242
20.3	在 Go 语言中使用 JSON	213	22.8	练习	242
20.4	解码 JSON	217	<b>第 23 章</b>	<b>Go 语言时间编程</b>	<b>243</b>
20.5	映射数据类型	219	23.1	时间元素编程	243
20.6	处理通过 HTTP 收到的 JSON	220	23.2	让程序休眠	245
20.7	小结	221	23.3	设置超时时间	245
20.8	问与答	221	23.4	使用 ticker	246
20.9	作业	222	23.5	以字符串格式表示时间	247
20.9.1	小测验	222	23.6	使用结构体 Time	248
20.9.2	答案	222	23.7	时间加减	249
20.10	练习	222	23.8	比较两个不同的 Time 结构体	249
<b>第 21 章</b>	<b>处理文件</b>	<b>223</b>	23.9	小结	250
21.1	文件的重要性	223	23.10	问与答	250
21.2	使用 ioutil 包读写文件	224	23.11	作业	251
21.2.1	读取文件	224	23.11.1	小测验	251
21.2.2	创建文件	225	23.11.2	答案	251
21.3	写入文件	227	23.12	练习	251
21.4	列出目录的内容	228	<b>第 24 章</b>	<b>部署 Go 语言代码</b>	<b>252</b>
21.5	复制文件	229	24.1	理解目标	252
21.6	删除文件	230	24.2	压缩二进制文件的大小	255
21.7	使用文件来管理配置	231	24.3	使用 Docker	256
21.7.1	使用 JSON 文件	231	24.4	下载二进制文件	258
21.7.2	使用 TOML 文件	232	24.5	使用 go get	259
21.8	小结	234	24.6	通过包管理器发布代码	260
21.9	问与答	234	24.7	小结	260
21.10	作业	234	24.8	问与答	260
21.10.1	小测验	235	24.9	作业	261
21.10.2	答案	235	24.9.1	小测验	261
21.11	练习	235	24.9.2	答案	261
<b>第 22 章</b>	<b>正则表达式简介</b>	<b>236</b>	24.10	练习	261



# 第 1 章

## 起步

---

本章介绍如下内容。

- Go 简介。
- 安装 Go。
- 设置环境。
- 编写您的第一个 Go 程序——Hello World。

通过阅读本章，您将知道 Go 是什么以及创建它的动机是什么。您将学习如何安装 Go 并运行第一个程序。您还将了解 Go Gopher，它会在本书中时不时地蹦出来。

### 1.1 Go 简介

Go（或 Golang）是 Google 在 2007 年开发的一种开源编程语言，出自 Robert Griesemer、Rob Pike 和 Ken Thompson 之手。2009 年 11 月 10 日，Google Open Source Blog 向全球发布了这款语言；公告指出 Go 的主要目标是“兼具 Python 等动态语句的开发速度和 C 或 C++ 等编译型语言的性能与安全性”。

#### 1.1.1 Go 语言简史

对语言进行评估时，明白设计者的动机以及语言要解决的问题很重要。Go 语言的设计者都是计算机科学领域的重量级人物。在 20 世纪 70 年代，Ken Thompson 设计并实现了最初的 UNIX 操作系统，仅从这一点说，他对计算机科学的贡献怎么强调都不过分。他还与 Rob Pike 合作设计了 UTF-8 编码方案。除帮助设计 UTF-8 外，Rob Pike 还帮助开发了分布式多用户操作系统 Plan 9，并与人合著了 *The Unix Programming Environment*，对 UNIX 的设计理念做了正统的阐述。Robert Griesemer 就职于 Google，