

零壹快学程序设计系列丛书

零壹
快学

书网合一

小白实战手册

零基础Go语言 从入门到精通

立体化教学模式 零基础快速入门

零壹快学 编著

真人教学，手把手教你学Go语言

- 丰富示例，贴近场景——丰富的代码示例，编程场景化
- 视频教学，动手操作——扫码即可学习配套视频，通俗易懂
- 线上问答，及时解惑——在线论坛，解答学习中遇到的疑问
- 海量题库，随时学习——大量题目练习，加快理解



扫一扫
获取视频激活码

SPM 南方出版传媒 广东人民出版社

◇—— 零壹快学简介 ——◇

零基础，一起学！

零壹快学是一个集教育、培训、交流于一体的计算机学习融媒体平台，覆盖Web、APP和微信小程序等场景，为广大IT学习者提供计算机各学科的教学资源、培训课程、测评系统等。通过人工智能手段，模拟真实课堂环境，力求打造成为一个全面、权威、高效的一流计算机在线教育平台。目前，平台已开发了包括计算机编程、云计算、区块链等在内的多门课程，提供计算机等级考试在线题库系统，将陆续开发计算机技术与软件专业技术资格（水平）考试、CCIE等在线题库系统，同时拥有一支由一线互联网公司工程师、知名高校教师组成的专家团队，精心为广大IT学习者提供优质的学习资源、课程和服务。

零基础 GO 从入门到精通

零壹快学 GO 团队出品

广东人民出版社

图书在版编目 (CIP) 数据

零基础Go语言从入门到精通 / 零壹快学编著. —广州: 广东人民出版社, 2020.4
ISBN 978-7-218-13975-3


I. ①零… II. ①零… III. ①程序语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字 (2019) 第247966号

Ling Jichu Go Yuyan Cong Rumen Dao Jingtong

零基础Go语言从入门到精通

零壹快学 编著

 版权所有 翻印必究

出版人: 肖风华

统筹策划: 李婧玮

责任编辑: 陈泽洪

封面设计: 画画鸭工作室

内文设计: 奔流文化

责任技编: 吴彦斌

出版发行: 广东人民出版社

地址: 广州市海珠区新港西路204号2号楼 (邮政编码: 510300)

电话: (020) 85716809 (总编室)

传真: (020) 85716872

网址: <http://www.gdpph.com>

印刷: 广东信源彩色印务有限公司

开本: 787毫米×1092毫米 1/16

印张: 26 字数: 500千

版次: 2020年4月第1版

印次: 2020年4月第1次印刷

定价: 79.00元

如发现印装质量问题, 影响阅读, 请与出版社 (020-32449105) 联系调换。

售书热线: 020-32449123

前 言

历经七十多年的发展，无论是对于国内数以十万计的学习者而言，还是在有着多年培训经验的编者看来，学习编程语言，仍存在不小的难度，甚至有不少学习者因编程语言的复杂多变、难度太大而选择了中途放弃。实际上，只要掌握了其变化规律，即使再晦涩难懂的计算机专业词汇也无法阻挡学习者们的脚步。对于初学者来说，若有一本能看得懂，甚至可以用于自学的编程入门书是十分难得的。为初学者提供这样一本书，正是我们编写本套丛书的初衷。

零壹快学以“零基础，一起学”为主旨，针对零基础编程学习者的需求和学习特点，由专业团队量身打造了本套计算机编程入门教程。本套丛书的作者都从事编程教育和培训工作多年，拥有丰富的教学经验，对于学习者常遇到的问题十分熟悉，在编写过程中针对这些问题花费了大量的时间和精力来加以阐释，对书中的每个示例反复推敲，加以取舍，按照学习者的接受程度雕琢示例涉及的技术点，力求成就一套真正适合初学者的编程书籍。

本套丛书涵盖了Java、PHP、Python、JavaScript、HTML、CSS、Linux、iOS、Go语言、C++、C#等计算机语言，同时借助大数据和云计算等技术，为广大编程学习者提供计算机各学科的视频课程、在线题库、测评系统、互动社区等学习资源。

◆ 课程全面，聚焦实战

本套丛书涵盖多门计算机语言，内容全面、示例丰富、图文并茂，通过通俗易懂的语言讲解相关计算机语言的特性，以点带面，突出开发技能的培养，既方便学习者了解基础知识点，也能帮助他们快速掌握开发技能，为编程开发设计积累实战经验。

◆ 专业团队，紧贴前沿

本套丛书作者由一线互联网公司高级工程师、知名高校教师和研究所技术人员等组成，线上线下同步进行专业讲解及点评分析，为学习者扫除学习障碍。与此同时，团队

在内容研发方向上紧跟当前技术领域热点，及时更新，直击痛点和难点。

◆ **全网覆盖，应用面广**

本套丛书已全网覆盖Web、APP和微信小程序等客户端，为广大学习者提供包括计算机编程、人工智能、大数据、云计算、区块链、计算机等级考试等在内的多门视频课程，配有相关测评系统和技术交流社区，互动即时性强，可实现在线教育随时随地轻松学。

随着大数据和云计算时代的到来，近些年来Go语言逐渐成为了程序开发人员最想了解 and 学习的编程语言之一。Go语言具有原生支持并发、代码风格清晰和编译速度快等特性，目前主要应用于微服务、高并发处理和网络编程。

本书基于Go 1.12编写而成，先从Go语言的基础知识入门，再到对Go开发的详细讲解，最终带领读者进入Go语言应用开发领域。对于零基础的读者而言，本书可以作为Go开发的快速入门教材。

• **本书内容**

◆ **简介和概况**：第1~2章，主要介绍了Go语言的概况和Go开发工具GoLand的使用方式。

◆ **基础知识**：第3~10章，主要介绍Go语言的基础知识，包括基本变量与类型、常量与运算符、流程控制、内置容器、函数、包和结构体等。

◆ **进阶知识**：第11~15章，主要介绍Go语言的进阶知识，包括文件操作、接口、并发与通道、反射等。

◆ **应用实战**：第16~20章，主要介绍如何通过Go语言来实现网络编程和数据库操作，并深入理解之前章节所学的知识点。

• **本书特点**

◆ **由浅入深，循序渐进**。本书先介绍Go语言基础，再介绍网络编程和数据库操作的实战应用，讲解过程详尽，通俗易懂。

◆ **示例丰富，贴近场景**。本书提供了丰富的代码示例，每个知识点均有对应示例代码进行演示，便于读者清晰理解。这些示例大部分来自于工作场景，有利于读者理解其中的使用逻辑，快速掌握。

◆ **视频教学，动手操作**。本书每一章都配有教学视频，直观展示了代码的运行效

果，并配有通俗易懂的解释。

◆ **知识拓展，难度提升。**本书的大部分章节结尾设有“知识拓展”，在讲解基础知识的同时提供了一些有一定难度的知识点，方便有能力的读者深入思考，强化学习，加深对Go开发的理解。

◆ **线上问答，及时解惑。**本书为确保广大读者的学习能够顺利进行，提供了在线答疑服务，希望通过这种方式及时解决读者在学习Go开发的过程中所遇到的困难和疑惑。

• 本书配套资源（可扫下方二维码获取）

- ◆ **大量的代码示例。**通过运行这些代码，读者可以进一步巩固所学的知识。
- ◆ **零壹快学官方视频教程。**力求让读者学以致用，知行并进，加强实战能力。
- ◆ **在线答疑。**为读者解惑，帮助读者解决学习中的困难，快速掌握要点难点。

• 本书适用对象

- ◆ 编程的初学者、爱好者与自学者
- ◆ 高等院校和培训学校的师生
- ◆ 职场新人
- ◆ 准备进入互联网行业的再就业人群
- ◆ “菜鸟”程序员
- ◆ 初、中级程序开发人员



零壹快学微信公众号

《零基础Go语言从入门到精通》由零壹快学汪心桐、廖旭编写。本书从初学者角度出发，详细讲述了Go语言应用开发所需的基础知识和开发实战中的必备技能。全书内容通俗易懂，示例丰富，步骤清晰，图文并茂，可以使读者轻松掌握Go语言应用开发的精髓，活学活用，是Go开发实战中必备的参考书。

编者

2020年3月

目 录

CONTENTS

第 1 章 走进 Go	1	第 3 章 基本变量与类型	22
1.1 Go 编程语言概述	1	3.1 Go 语言变量	22
1.1.1 Go 的历史	1	3.1.1 声明变量	22
1.1.2 Go 语言特性及应用场景	2	3.1.2 初始化变量	23
1.1.3 使用 Go 语言的项目	3	3.1.3 变量值交换	25
1.2 Go 语言官方文档	4	3.1.4 匿名变量	26
1.3 学好 Go 的建议	5	3.2 Go 语言数据类型	27
1.4 常用软件及网站	6	3.2.1 整型	27
1.5 小结	7	3.2.2 浮点型	28
第 2 章 安装和运行 Go	8	3.2.3 字符串	29
2.1 GOROOT 和 GOPATH	8	3.2.4 字符和转义字符	30
2.2 在 Windows 下安装 Go	9	3.2.5 布尔型	31
2.3 在 Linux 下安装 Go	11	3.2.6 数据类型判断	31
2.4 在 Mac OS 下安装 Go	12	3.2.7 数据类型转换	32
2.5 安装开发工具	13	3.3 指针	34
2.5.1 GoLand 下载	13	3.3.1 声明指针	34
2.5.2 GoLand 安装	13	3.3.2 取变量地址	34
2.5.3 GoLand 使用	16	3.3.3 获取指针所指向的内容	35
2.6 编写第一个 Go 程序	19	3.3.4 使用指针修改值	36
2.6.1 GoLand 编写	19	3.4 其他数据类型	37
2.6.2 文本编辑器编写	20	3.5 小结	38
2.7 小结	21	3.6 知识拓展	38

3.6.1 字符串的应用	38
3.6.2 Go 语言注释	43
第 4 章 常量与运算符	44
4.1 常量	44
4.1.1 常量的定义	44
4.1.2 常量组的定义	45
4.1.3 常量枚举	46
4.2 运算符	47
4.2.1 算术运算符	47
4.2.2 比较运算符	49
4.2.3 赋值运算符	50
4.2.4 位运算符	51
4.2.5 逻辑运算符	52
4.2.6 其他运算符	53
4.3 小结	54
4.4 知识拓展	54
第 5 章 流程控制	56
5.1 if 判断	56
5.2 循环控制	58
5.2.1 for 循环	58
5.2.2 break 跳出循环	59
5.2.3 continue 继续循环	61
5.3 switch 分支	61
5.4 goto 跳转	63
5.5 小结	64
5.6 知识拓展	64
5.6.1 斐波那契数列	64
5.6.2 循环嵌套	67

第 6 章 Go 语言内置容器	69
6.1 数组	69
6.1.1 声明数组	70
6.1.2 初始化数组	70
6.1.3 range 关键字	71
6.1.4 遍历数组	72
6.2 切片	72
6.2.1 从数组生成一个新的切片	73
6.2.2 从切片生成一个新的切片	74
6.2.3 直接生成一个新的切片	75
6.2.4 为切片添加元素	77
6.2.5 从切片删除元素	79
6.2.6 遍历切片	80
6.3 映射	81
6.3.1 声明映射	81
6.3.2 初始化映射	82
6.3.3 遍历映射	83
6.3.4 从映射中删除键值对	84
6.4 小结	85
6.5 知识拓展	86
第 7 章 函数	90
7.1 声明函数	90
7.1.1 函数参数的简写	91
7.1.2 函数返回值的简写	91
7.1.3 带有变量名的返回值	92
7.2 使用函数	92
7.3 函数变量	93
7.4 可变参数	95
7.4.1 可变参数函数的使用	95
7.4.2 可变参数与内置函数	96

7.4.3 可变参数的传递	96	8.10 知识拓展	124
7.5 匿名函数和闭包	98	第 9 章 结构体	126
7.5.1 定义和使用匿名函数	98	9.1 理解结构体	126
7.5.2 闭包的定义	99	9.2 定义结构体	127
7.5.3 闭包的“记忆力”	99	9.3 实例化结构体	128
7.6 延迟执行语句	101	9.3.1 标准实例化	128
7.7 小结	103	9.3.2 new 函数实例化	129
7.8 知识拓展	103	9.3.3 取地址实例化	129
7.8.1 函数参数传递的本质	103	9.3.4 访问成员	130
7.8.2 Go 内置函数	106	9.4 初始化结构体	131
第 8 章 包管理	109	9.4.1 键值对格式初始化	131
8.1 工作区	109	9.4.2 列表格式初始化	132
8.1.1 工作区结构	109	9.5 结构体方法	133
8.1.2 GOPATH	110	9.5.1 指针类型接收者	134
8.1.3 GOROOT	112	9.5.2 值类型接收者	135
8.1.4 GOBIN	112	9.6 结构体内嵌	136
8.2 包的声明	112	9.6.1 初始化结构体内嵌	138
8.3 包的导入	113	9.6.2 内嵌匿名结构体	139
8.3.1 导入声明	113	9.7 匿名结构体	140
8.3.2 远程导入	114	9.7.1 匿名结构体定义与初始化	140
8.3.3 别名导入	115	9.7.2 匿名结构体的应用	142
8.3.4 匿名导入	116	9.8 小结	143
8.4 main 包	117	9.9 知识拓展	143
8.5 包的初始化	118	第 10 章 错误处理	147
8.6 包的命名	119	10.1 错误处理的方式	147
8.7 依赖包管理	120	10.2 自定义错误	148
8.8 Go 语言命名规范	121	10.2.1 错误类型	148
8.8.1 驼峰式命名法	121	10.2.2 创建错误	149
8.8.2 导出标识符	122	10.2.3 自定义错误格式	149
8.9 小结	123		

10.3	Go 语言宕机	150	12.4	空接口	199
10.4	宕机恢复	151	12.4.1	将值保存到空接口	199
10.4.1	recover 捕获宕机	151	12.4.2	从空接口取值	200
10.4.2	recover 应用	153	12.4.3	空接口的常见使用	201
10.5	Go 语言错误的应用	154	12.5	类型断言	203
10.6	小结	155	12.5.1	ok-pattern	206
10.7	知识拓展	156	12.5.2	switch-type	207
第 11 章	文件操作	158	12.6	小结	209
11.1	目录基本操作	158	12.7	知识拓展	209
11.1.1	列目录	159	第 13 章	并发与通道	213
11.1.2	创建目录	163	13.1	概述	213
11.1.3	删除目录	166	13.1.1	并行与并发	214
11.2	文件基本操作	168	13.1.2	Go 并发优势	214
11.2.1	文件创建与打开	169	13.2	goroutine	215
11.2.2	文件读取	171	13.2.1	goroutine 定义	215
11.2.3	文件写入	174	13.2.2	创建 goroutine	216
11.2.4	删除文件	177	13.2.3	main 函数	218
11.3	处理 JSON 文件	178	13.2.4	使用匿名函数创建 goroutine	220
11.3.1	编码 JSON	179	13.2.5	runtime 包	220
11.3.2	解码 JSON	182	13.3	channel	225
11.4	小结	186	13.3.1	channel 类型	225
第 12 章	接口与类型	187	13.3.2	缓冲机制	227
12.1	接口定义	187	13.3.3	close 和 range	229
12.1.1	接口的定义	187	13.3.4	单向 channel	232
12.1.2	鸭子类型	188	13.3.5	定时器	233
12.2	接口的创建与实现	189	13.4	select	234
12.2.1	接口创建	189	13.4.1	select 作用	234
12.2.2	接口赋值	193	13.4.2	超时	236
12.3	接口嵌入	196	13.4.3	死锁	237
			13.5	小结	238

第 14 章 反射	239	15.5.3 覆盖率测试	277
14.1 反射定义	239	15.6 性能分析	278
14.1.1 反射的定义	239	15.6.1 安装 Graphviz	278
14.1.2 与其他语言的区别	239	15.6.2 通过文件方式	279
14.2 基本用法	241	15.6.3 通过 HTTP 方式	281
14.2.1 获取类型信息	243	15.7 小结	284
14.2.2 获取类型的值	245	第 16 章 正则表达式	285
14.2.3 使用反射调用函数	246	16.1 正则表达式介绍	285
14.3 对结构体的反射操作	248	16.2 正则表达式语法	286
14.3.1 获取结构体成员类型	248	16.2.1 普通字符	286
14.3.2 获取结构体成员字段的值	250	16.2.2 字符转义	288
14.3.3 反射执行结构体方法	251	16.2.3 元字符	288
14.4 反射三定律	252	16.2.4 限定符	290
14.4.1 接口到反射类型的转换	252	16.2.5 定位符	290
14.4.2 反射到接口类型的转换	253	16.2.6 分组构造	291
14.4.3 修改反射类型对象	254	16.2.7 匹配模式	291
14.5 反射的性能	255	16.3 regexp 包	292
14.6 小结	257	16.3.1 MatchString 函数	292
第 15 章 Go 命令行工具	258	16.3.2 FindStringIndex 函数	294
15.1 编译相关指令	259	16.3.3 ReplaceAllString 函数	295
15.1.1 build	259	16.4 小结	296
15.1.2 run	262	16.5 知识拓展	296
15.1.3 install	264	第 17 章 HTTP 编程 (上)	298
15.1.4 交叉编译	265	17.1 HTTP 简介	298
15.2 代码获取 (get)	267	17.1.1 HTTP 协议	298
15.3 格式化代码 (fmt)	268	17.1.2 URL 地址	300
15.4 注释文档 (doc)	269	17.1.3 Web 工作方式	300
15.5 代码测试 (test)	271	17.2 HTTP 客户端	301
15.5.1 单元测试	271	17.2.1 REST	301
15.5.2 基准测试	274		

17.2.2	Client 和 Request	302	19.1.3	IPv4 和 IPv6	353
17.2.3	发起 GET 请求	307	19.1.4	子网掩码	354
17.2.4	发起 POST 请求	312	19.2 Socket 基础	355	
17.3 爬虫框架 gocolly	315		19.2.1	Socket 简介	355
17.3.1	gocolly 简介	316	19.2.2	netcat 工具简介	356
17.3.2	实战爬取西刺代理 IP	320	19.3 TCP 编程	357	
17.4 小结	324		19.3.1	TCP 简介	357
17.5 知识拓展	324		19.3.2	TCP 客户端	359
第 18 章 HTTP 编程 (下)	327		19.3.3	TCP 服务端	364
18.1 错误处理的方式	327		19.4 UDP 编程	369	
18.1.1	最常见的 Web 服务端	328	19.4.1	UDP 简介	369
18.1.2	请求方法	328	19.4.2	UDP 客户端	369
18.1.3	响应状态码	329	19.4.3	UDP 服务端	372
18.2 HTTP 服务端	329		19.5 小结	376	
18.2.1	Server 和 Response	330	19.6 知识拓展	377	
18.2.2	启动 HTTP 服务器	332	第 20 章 数据库编程	381	
18.2.3	启动 HTTPS 服务器	333	20.1 MySQL 简介	381	
18.2.4	静态文件	336	20.1.1	安装 MySQL	382
18.3 路由	338		20.1.2	MySQL 常见命令	382
18.4 中间件	341		20.2 database/sql	383	
18.5 Web 框架 gin	343		20.3 数据库基本操作	384	
18.5.1	启动 HTTP 服务器	344	20.3.1	连接数据库	384
18.5.2	静态文件	344	20.3.2	创建数据表	386
18.5.3	路由	345	20.3.3	插入数据	389
18.6 小结	347		20.3.4	查询数据	392
18.7 知识拓展	347		20.3.5	更改数据	395
第 19 章 Socket 编程	350		20.3.6	删除数据	397
19.1 计算机网络简介	350		20.3.7	MySQL 事务	398
19.1.1	OSI 七层网络模型	350	20.4 小结	401	
19.1.2	TCP/IP 协议	352	20.5 知识拓展	401	

» 第 1 章

走进 Go «

1.1 Go编程语言概述

Go语言也叫Golang，是由谷歌（Google）公司在2007年推出的一款静态编译型语言。Go语言高效、简洁、容易上手，加上它的高并发和自带垃圾回收等特性，一出现就备受大家的喜爱。目前，越来越多的公司开始推广并使用Go语言，主要将其用于服务端开发、并发编程和网络编程等。

本章将带你走进Go语言，体会Go语言的独特魅力。

1.1.1 Go的历史

Go语言起源于2007年，于2009年正式对外发布。Go语言项目主要由三位IT届著名人士参与开发：肯·汤普逊（Ken Thompson）、罗布·派克（Rob Pike）和罗伯特·格里泽默（Robert Griesemer）。此处简单介绍一下他们的贡献。

肯·汤普逊：图灵奖得主，Unix发明人，B语言（C语言前身）作者。1983年，他与丹尼斯·里奇（Dennis Ritchie）共同获得了图灵奖。2006年，他进入谷歌公司工作，与其他人共同设计了Go语言。

罗布·派克：Unix团队和Plan 9操作系统计划的成员，曾经参与过贝尔实验室九号计划、操作系统Inferno与编程语言Limbo的开发。

罗伯特·格里泽默：曾协助Java的HotSpot编译器和Chrome浏览器的JavaScript引擎V8的制作。

Go语言主要大事件如下：

2007年，Go语言雏形设计。

2009年，首次公开发布。

2010年，当选TIOBE2009年度最佳编程语言，谷歌正式将其投入使用。

2011年，Google App Engine支持Go语言。

2017年，当选TIOBE2016年度最佳编程语言。

截至目前，Go语言的最新版本为Go 1.13。

另外，Go语言还有个十分可爱的吉祥物——金花鼠（图1.1.1），由罗布·派克的妻子芮妮·弗伦奇（Renée French）绘制。

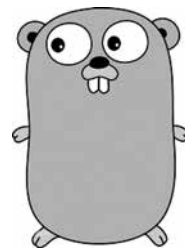


图1.1.1 Go语言吉祥物

1.1.2 Go语言特性及应用场景

随着人工智能、大数据和云计算时代的到来，Python、Java和PHP等编程语言风靡盛行，C/C++等编程语言也有自己的特性和应用场景，那么作为后起之秀的Go语言究竟有哪些特性和优势？这些特性又适用于哪些应用场景呢？

1. 容易上手

Go语言上手非常容易，许多零基础的初学者在学习大约一周的时间后就可以使用Go语言完成一些既定的任务。而C/C++等语言，则需要经过几个月的时间学习才开始上手，并且学习中的大量时间都用于避免语法错误和调试代码bug。

2. 编译速度快

在Go编程语言的1.7版本更新中，通过减少编译器所使用的内存量、减少链接时间来减少总的编译时间。根据测试，这项改进和Go1.5.3相比减少了20%~30%的总编译时间。

另外，Go语言工程结构简单，没有头文件和不允许包的交叉编译等规则，这也在很大程度上减少了编译所需的时间。

3. 原生支持并发

Go语言最主要的特性就是从语言层面原生支持并发，无须任何第三方库。Go的并发基于goroutine，可以理解为一种微线程。与Python的多线程性能限制于GIL全局锁不同，Go语言的并发可以充分利用CPU的资源，将goroutine合理地分配到每个CPU中，最大限度地使用CPU的性能。goroutine之间的通信可以使用Go语言的sync包和channel机制来实现。

4. 垃圾回收

一直以来，内存管理是程序员开发中的一大难题。传统的编程语言C/C++中，程序员必须对内存进行细致的管理操作，控制内存的申请及释放，否则就可能产生内存泄漏问题。

为了解决这个问题，Go语言实现了自己的垃圾回收器，在1.3版本前采用传统的Mark and Sweep算法。Go语言在1.5的版本更新中进行了较大改进，使用了三色标记算法，进一步改善了垃圾回收的机制，使程序员更加专注于业务本身，不用把过多的精力投入到垃圾回收中。

5. 代码风格清晰

Go语言语法在C语言的基础上进行优化，去掉了表达式和for循环中冗余的括号，强制统一了代码风格，如Go语言中，左括号必须紧接着语句不进行换行。有些开发者，特别是有其他编程语言开发经验的人，刚开始时会对这个特性感到不习惯，但随着对Go语言的不断深入熟悉，就会发现这个特性能让代码阅读者将更多精力投放到代码本身，而不是代码风格上去。

由于以上特性，Go语言主要应用于以下三个方面：（1）服务端开发；（2）高并发处理；（3）网络编程。

基于以上三个方面，Go语言可以具体用于服务端网络请求接收和解析、分布式系统开发、后端API接口开发和虚拟机处理等。

1.1.3 使用Go语言的项目

Go语言在2009年首次公开发布后受到越来越多的开发者的关注，Go语言社区不断壮大，开源项目不断涌现。下面列举使用Go语言进行开发的部分项目。

1. Docker项目

网址：<https://github.com/docker/docker-ce>

Docker是一个开源的应用容器引擎，基于Go语言开发并遵从Apache 2.0协议开源。Docker可以让开发者打包他们的应用和依赖包到一个轻量级、可移植的容器中，也可以实现虚拟化，更重要的是容器性能开销极低。

2. Golang项目

网址：<https://github.com/golang/go>

Go语言自1.5版本后，可以完全使用Go语言自身来实现。对Go语言的源码研究有助于深入理解Go的底层实现，想对Go语言有持续深入了解的读者可阅读此项目。

3. Kubernetes项目

网址：<https://github.com/kubernetes/kubernetes>

Kubernetes是自动化容器操作的开源平台，使用Kubernetes可以进行自动化容器的部署和复制、随时扩展或收缩容器规模和容器间的负载均衡等。

4. Beego项目

网址：<https://github.com/astaxie/beego>

Beego是一个使用Go编写的极轻量级、高可伸缩性和高性能的Web应用框架。

5. Codis项目

网址：<https://github.com/CodisLabs/codis>

Codis是一个国产开源的分布式redis解决方案，对于上层的应用透明，就像连接到原生的redis服务器那样。

1.2 Go语言官方文档

Go语言官方文档网址：<https://golang.org/doc/>。官方文档中有关于如何写Go代码的建议、编辑器推荐和Go语言的发布历史等内容。

Go语言官方文档网站主页如图1.2.1所示。

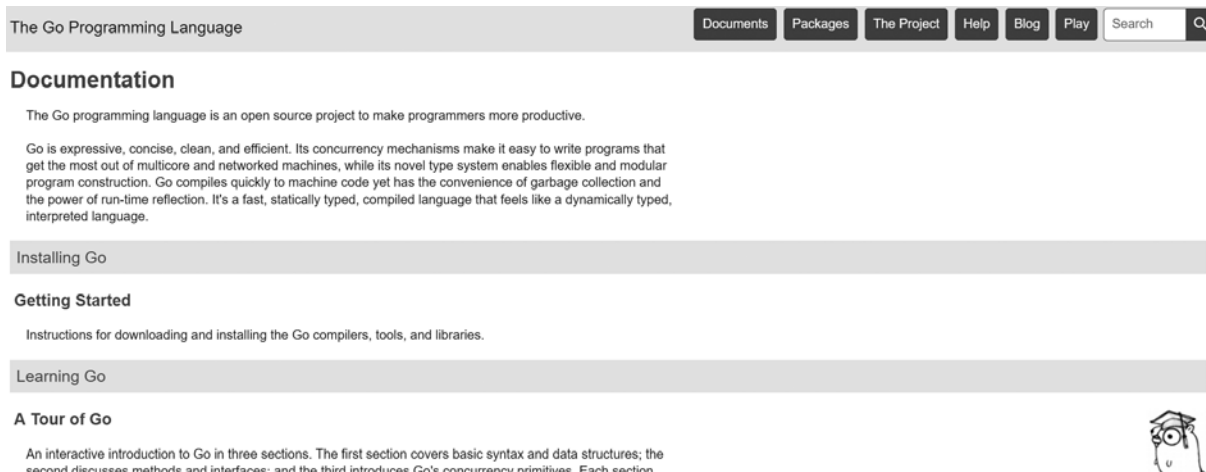


图1.2.1 Go语言官方文档网站主页

如果要查看Go语言标准库或其他库函数的用法，可参见官方的包索引与搜索引擎网站：<https://godoc.org/>，里面有详细的函数用法和样例，主页如图1.2.2所示。

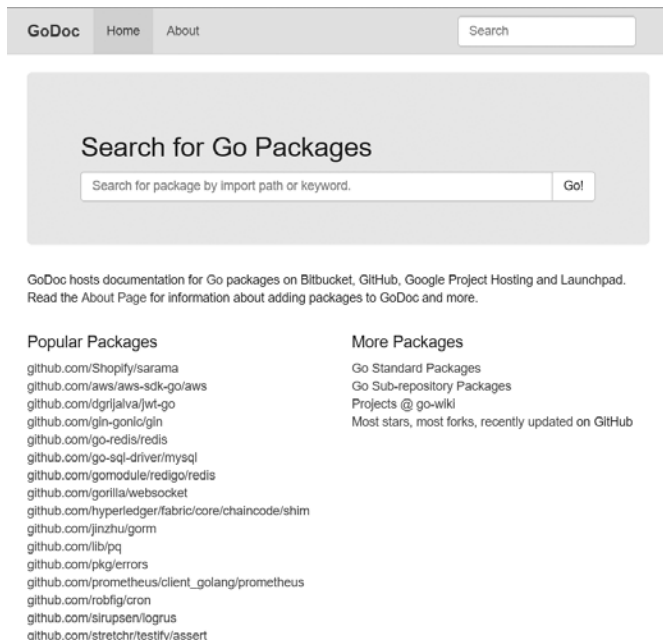


图1.2.2 Go语言官方包索引与搜索引擎网站主页