# 初识Go语言

本章目的是带领大家初步认识Go语言。先给大家讲解Go语言的发展历史，设计Go语言的初衷以及Go语言的主要特征。接下来带领大家安装Go语言的运行环境Golang，以及Go语言集成开发工具（IDE）——Goland。之后便是Go语言版本的HelloWorld程序，以及通过对HelloWorld的分析，掌握Go语言的构成及编码规范。

本章重点为大家介绍如下的内容：

* Go语言发展历史
* Go语言核心特性
* Golang的安装及环境变量配置
* 安装Goland开发环境及HelloWorld
* helloworld程序解读及Go语言编码规范

## Go语言的发展历史

### 知名编程语言发展简史

B语言：Ken Thompson（肯.汤普森）。该语言得名于汤姆森的妻子Bonnie，一门剑桥于60年代中期开发的语言。

C语言：美国著名计算机专家Dennis Ritchie（丹尼斯·里奇）在1969-1973年期间发明了C语言和Unix操作系统。

C++：Bjarne Stroustrup（本贾尼·斯特劳斯特卢普）。1982年，美国贝尔实验室的Bjarne Stroustrup博士在C语言的基础上引入并扩充了面向对象的概念，发明了—种新的程序语言。为了表达该语言与C语言的渊源关系，它被命名为C++。

Python：Guido von Rossum（吉多.范.罗苏姆）。1989年，为了打发圣诞节假期，Guido开始写Python语言的编译/解释器。1994年发布1.0版本。

Java：James Gosling（詹姆斯.高斯林）。1991年开发Oak，1994年更名为Java。1995年5月正式发布。

Javascript：Brendan Eich（布兰登.艾奇)。艾奇对Java一点兴趣也没有，为了应付公司安排的任务，他于1995年5月只用10天时间就把Javascript设计出来了。

Go语言：Go的三个作者分别是：Robert Griesemer（罗伯特.格利茨默）, Rob Pike（罗伯.派克） 和 Ken Thompson（肯.汤普森）。

Robert在开发Go之前是Google V8、Chubby和HotSpot JVM的主要贡献者；

Rob主要是Unix、UTF-8、plan9的作者；

Ken主要是B语言、C语言的作者、Unix之父。



图 1.1

### Go语言主要发展过程

2007年9月，Rob Pike（罗伯.派克） 正式命名为Go；

2008年5月，Google全力支持该项目；

2009年11月，Go将代码全部开源，它获得了当年的年度语言；

2012年3月28日，Go1.0发布；

Go1.0版发布时，Go Team承诺后续的版本都会兼容之前的版本。这对于开发者来说非常重要，Go后续的版本也一直在提升内功，从而可以让用户无缝的升级Go版本。

2020年2月26日，Go1.14版本发布，Go Module已经具备在生产环境中使用的条件，并鼓励所有用户迁移到Go Module进行依赖关系管理。

## Go语言的特点及优势

### Go语言设计初衷

设计Go语言是为了解决当时Google开发遇到的问题：

大量的C++代码，同时又引入了Java和Python；

成千上万的工程师；

数以万计行的代码；

分布式的编译系统；

数百万的服务器。

Google开发中的痛点：

编译慢；

失控的依赖；

每个工程师只是用了一个语言里面的一部分；

程序难以维护（可读性差、文档不清晰等）；

更新的花费越来越长；

交叉编译困难。

如何解决当前的问题和痛点？

Go希望成为互联网时代的C语言。多数系统级语言（包括Java和C#）的根本编程哲学来源于C++，将C++的面向对象进一步发扬光大。但是Go语言的设计者却有不同的看法，他们认为值得学习的是C语言。C语言经久不衰的根源是它足够简单。因此，Go语言也是足够简单。

他们当时设计Go的目标是为了消除各种缓慢和笨重、改进各种低效和扩展性。Go是由那些开发大型系统的人设计的，同时也是为了这些人服务的；它是为了解决工程上的问题，不是为了研究语言设计；它还是为了让我们的编程变得更舒适和方便。

但是结合Google当时内部的一些现实情况，如很多工程师都是C系的，所以新设计的语言一定要易学习，最好是类似C的语言；20年没有出新的语言了，所以新设计的语言必须是现代化的（例如内置GC）等情况。最后根据实战经验，他们向着目标设计了Go这个语言。

Go语言的特色：

没有继承多态的OO；

强一致类型；

interface不需要显式声明(Duck Typing)；

没有异常处理(Error is value)；

基于首字母的可访问特性；

不用的import或者变量引起编译错误；

完整而卓越的标准库包。

### Go语言的优势

Go语言语法简单，包含了类C语法。它只提供了基本功能而没有多余的东西。Go引入了“defer”声明及带有goroutines和通道的并发性管理。

Go语言编译时间短，开发效率和运行效率高。开发过程中相较于 Java 和 C++呆滞的编译速度，Go 的快速编译时间是一个主要的效率优势。Go拥有接近C的运行效率和接近PHP的开发效率。Go语言可以说是开发效率和运行效率二者的完美融合，天生的并发编程支持。

Go语言支持当前所有的编程范式，包括过程式编程、面向对象编程、面向接口编程、函数式编程。程序员们可以各取所需、自由组合、想怎么玩就怎么玩。

Go 是一种非常高效的语言，高度支持并发性。Go是为大数据、微服务、并发而生的一种编程语言。作为一门语言致力于使事情简单化。它并未引入很多新概念，而是聚焦于打造一门简单的语言，它使用起来异常快速并且简单。其创新之处是 goroutines 和通道。Goroutines 是 Go 面向线程的轻量级方法，而通道是 goroutines 之间通信的优先方式。

goroutines 的成本很低，只需几千个字节的额外内存，正由于此，才使得同时运行数百个甚至数千个 goroutines 成为可能。可以借助通道实现 goroutines 之间的通信。goroutines 以及基于通道的并发性方法使其非常容易使用所有可用的 CPU 内核，并处理并发的 IO。相较于 Python/Java，在一个 goroutine 上运行一个函数需要最小的代码。

Go拥有强大的编译检查、严格的编码规范和完整的软件生命周期工具，具有很强的稳定性，稳定压倒一切。

### Go语言的核心特性和优势

Go主要有静态语言、函数多返回值、天生并发、内置GC（自动垃圾回收）、安全性高、语法简单、编译快速这几个方面的特性。这些特性决定了Go的三个高富帅特性：运行快、开发快和部署快。

静态类型语言是指在编译时变量的数据类型即可确定的语言，要求在使用变量之前必须声明数据类型（具有类型推导能力的现代语言可能能够部分减轻这个要求）；

动态类型语言是在运行时确定数据类型的语言，变量使用之前不需要类型声明，通常变量的类型是被赋值的那个值的类型。

### 采用Go语言的国内外知名企业

Go发布之后，很多公司特别是云计算公司开始用Go重构他们的基础架构，很多都是直接采用Go进行了开发。

采用Go的一些国外公司，如Google、Docker、Apple、Cloud Foundry、CloudFlare、Couchbase、CoreOS、Dropbox、MongoDB、AWS等公司。

采用Go开发的国内企业：如阿里云CDN、百度、小米、七牛、PingCAP、华为、金山软件、猎豹移动、饿了么等公司。

### Go语言学习的开发文档

https://www.studygolang.com/pkgdoc



图 1.2

## Golang安装和配置

### 下载

在Mac、Windows和Linux三个平台上都支持Golang。您可以从https://golang.org/dl/下载相应平台的二进制文件。该网站在国内不容易访问，所以可以访问https://www.studygolang.com/dl 进行安装软件的下载。

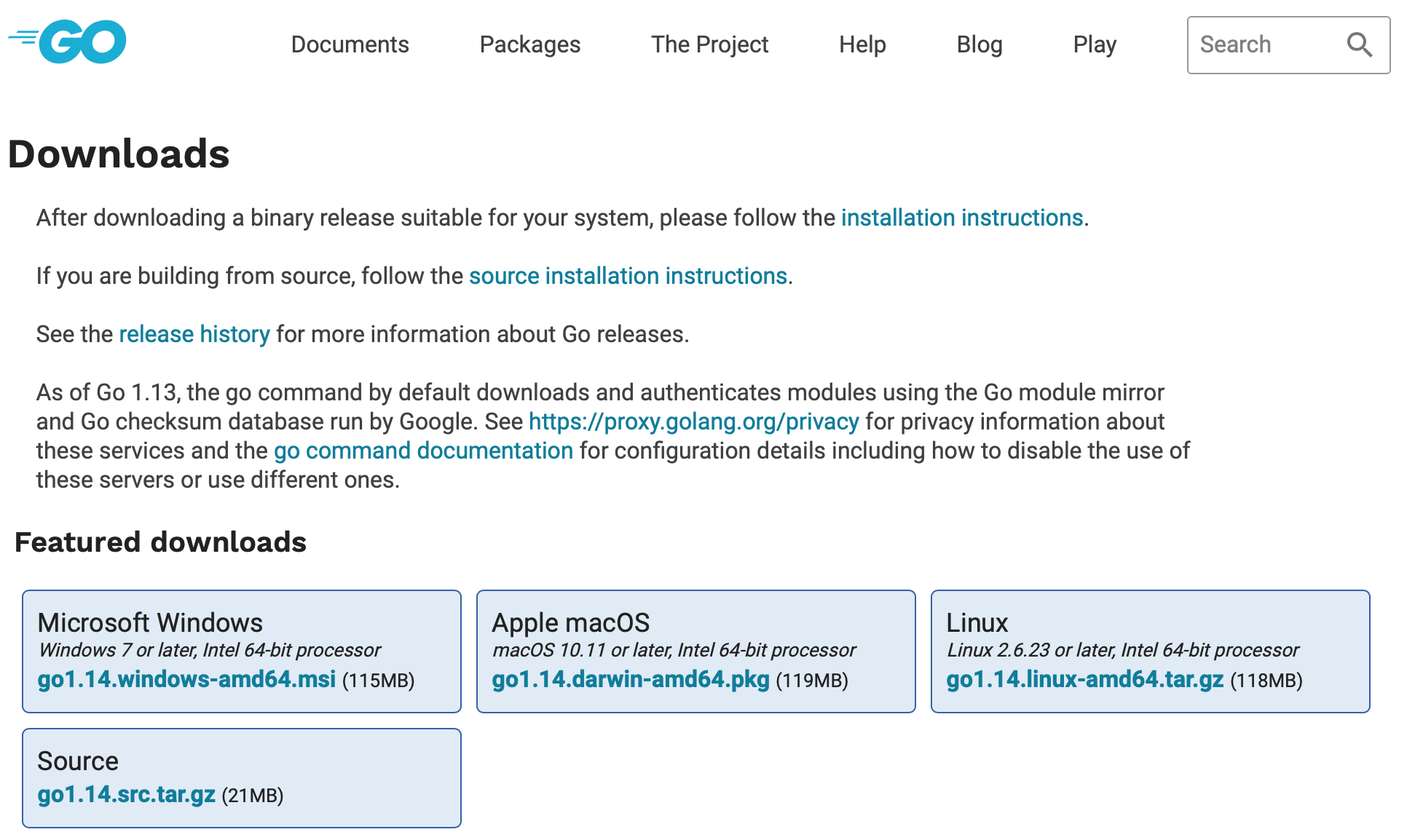


图 1.3



图 1.4

Mac OS 从https://golang.org/dl/下载osx安装程序。双击启动安装。按照提示，这应该在/usr/local/go中安装了Golang，并且还会将文件夹/usr/local/go/bin添加到您的PATH环境变量中。

Windows 从https://golang.org/dl/下载MSI安装程序。双击启动安装并遵循提示。这将在位置c中安装Golang:\Go，并且还将添加目录c:\Go\bin到您的path环境变量。

Linux 从https://golang.org/dl/下载tar文件，并将其解压到/usr/local。将/usr/local/go/bin添加到PATH环境变量中。这应该安装在linux中。

### windows系统下安装及配置环境变量

windows系统下安装步骤非常简单，一路到底。

windows系统下配置环境变量

注意：如果是msi安装文件，Go语言的环境变量会自动设置好。

我的电脑——右键“属性”——“高级系统设置”——“环境变量”——“系统变量”

假设GO安装于C盘根目录

新建：

GOROOT：Go安装路径（例：C:\Go）

GOPATH：Go工程的路径（例：E:\go）。如果有多个，就以分号分隔添加

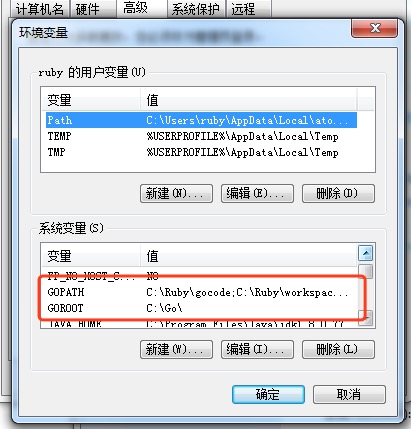


图 1.4

修改：

Path：在path中增加：C:\Go\bin;%GOPATH%\bin;

需要把GOPATH中的可执行目录也配置到环境变量中, 否则你自行下载的第三方go工具就无法使用了。

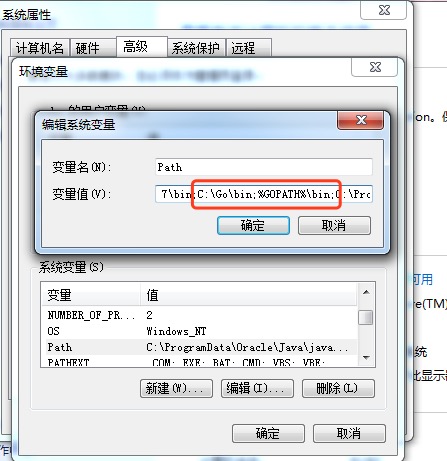


图 1.5

工作目录就是我们用来存放开发的源代码的地方，对应的也是Go里的GOPATH这个环境变量。这个环境变量指定之后，我们编译源代码等生成的文件都会放到这个目录下，GOPATH环境变量的配置参考上面的安装Go，配置到Windows下的系统变量里。

GOPATH之下主要包含三个目录: bin、pkg、src。bin目录主要存放可执行文件; pkg目录存放编译好的库文件, 主要是\*.a文件; src目录下主要存放go的源文件。

查看是否安装配置成功

使用快捷键win+R键，输入cmd，打开命令行提示符，在命令行中输入

go env # 查看得到go的配置信息

go version # 查看go的版本号

### mac系统下安装及配置环境变量

mac系统下安装

双击pkg包，顺着指引，即可安装成功。 在命令行输入 go version，获取到go的版本号，则代表安装成功。

mac平台下用户可以使用brew直接安装对应的go版本。

mac系统下配置环境变量

1. 打开终端输入cd ~进入用户主目录;
2. 输入ls -all命令查看是否存在.bash\_profile;
3. 存在既使用vim .bash\_profile 打开文件;
4. 输入 i 进入vim编辑模式；
5. 输入下面代码， 其中 GOPATH: 日常开发的根目录。GOBIN:是GOPATH下的bin目录。

* export GOPATH=/Users/steven/Documents/go\_project
* export GOROOT= /Usr/local/go
* export GOBIN=$GOROOT/bin
* export PATH=$PATH:$GOBIN

1. 点击ESC，并输入 :wq 保存并退出编辑。可输入vim .bash\_profile 查看是否保存成功。
2. 输入source ~/.bash\_profile 完成对golang环境变量的配置，配置成功没有提示。
3. 输入go env 查看配置结果

## Goland安装和配置

### 安装Goland开发工具

Goland是由JetBrains公司旨在为go开发者提供的一个符合人体工程学的新的商业IDE。这个IDE整合了IntelliJ平台的有关go语言的编码辅助功能和工具集成特点。它具有以下特点：

* 编码辅助功能
* 符合人体工程学的设计
* 工具的集成
* IntelliJ插件生态系统

### 下载及安装

官网下载地址：https://www.jetbrains.com/go/download/。下载完成后，在本地执行解压，安装。

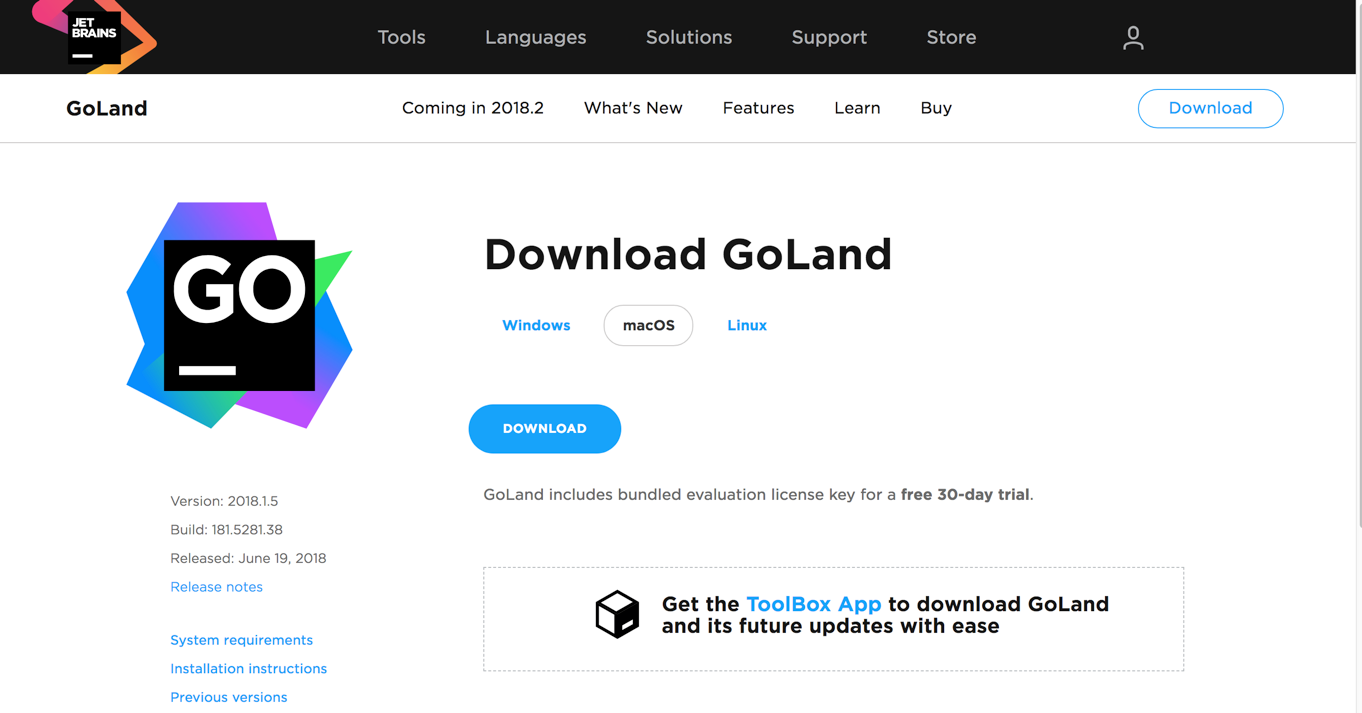


图 1.6

点击“next”按钮，选择要安装的路径，然后点击“next”，会出现安装选项。根据你自己电脑的型号，选择合适的版本后点击“next”按钮。接着保持默认的程序启动目录，点击“install”进行安装。整个安装过程很快，几乎一路next到底。

### 使用Goland

1.打开Goland工具

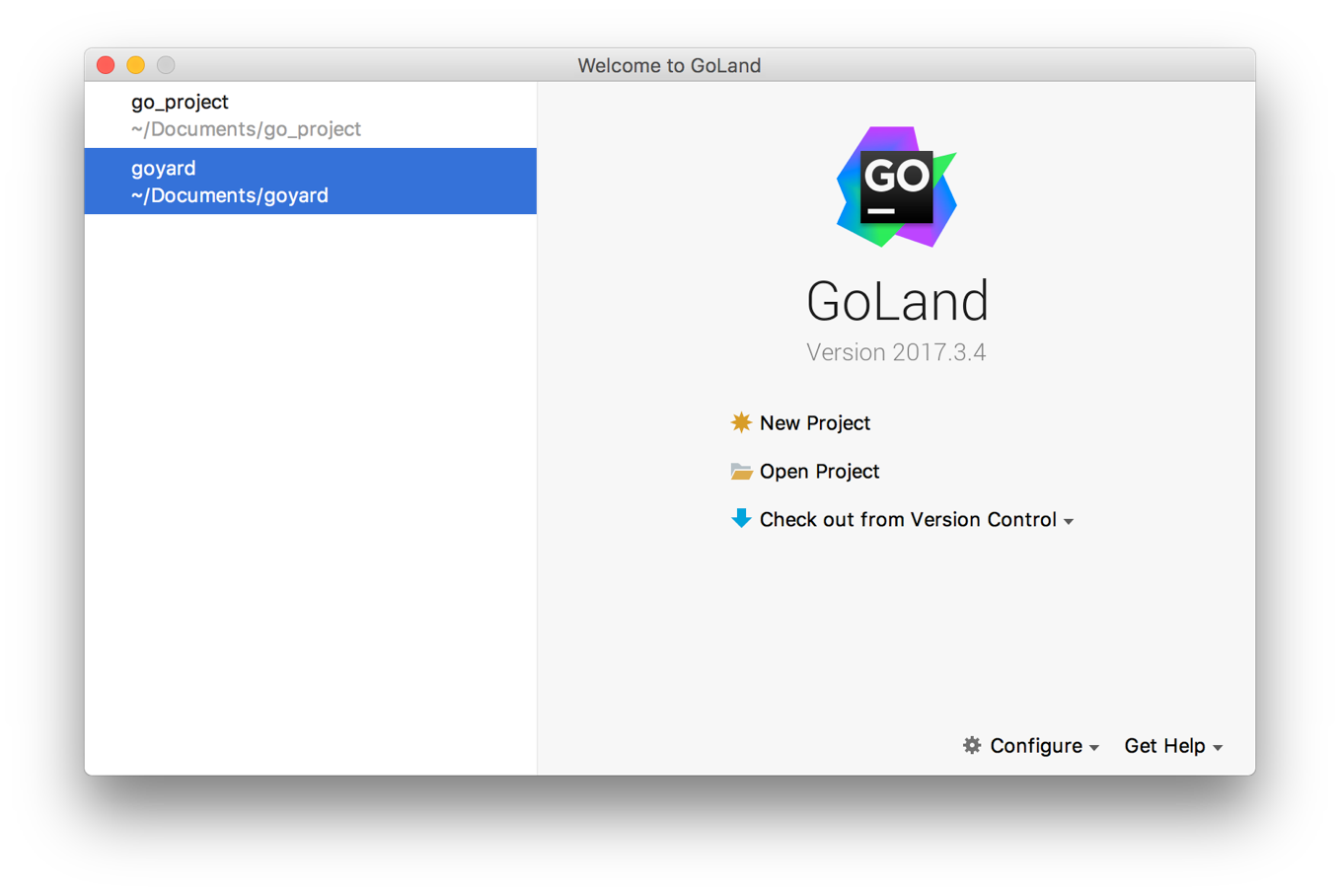


图 1.7

2.创建项目：

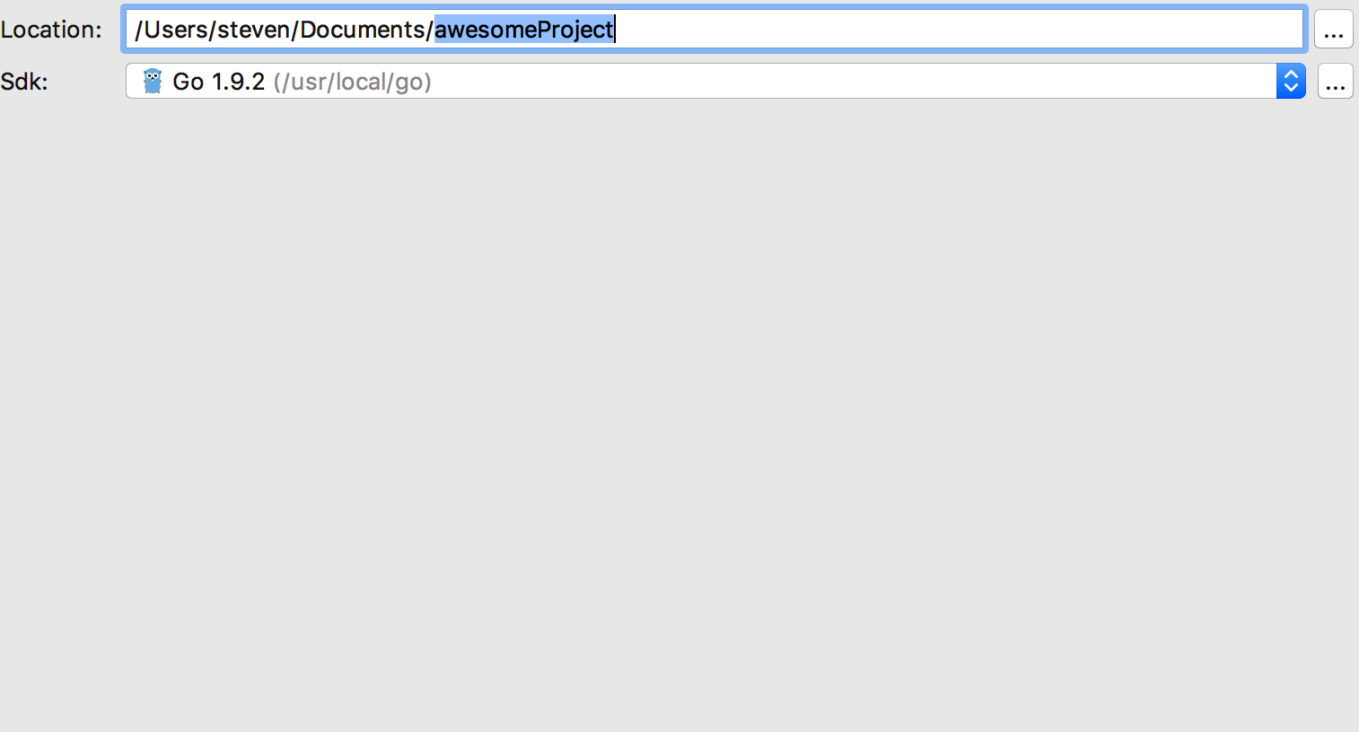


图 1.8

### 编写第一个程序HelloWorld

1.打开编辑器创建一个新的helloworld.go文件，并输入以下内容：

package main

import "fmt"

func main() {

/\* 输出 \*/

fmt.Println("Hello, World!")

}

2.执行go程序

执行go程序由几种方式

方式一：使用go run命令

* step1：使用快捷键win+R，输入cmd打开命令行提示符
* step2：进入helloworld.go所在的目录
* step3：输入go run helloworld.go命令并观察运行结果。

方式二：使用go build命令

* step1：使用快捷键win+R，输入cmd打开命令行提示符
* step2：进入helloworld.go所在的目录
* step3：输入go build helloworld.go命令进行编译，产生同名的helloworld.exe文件
* step4：输入helloworld.exe，执行

## Go语言编码规范

### HelloWorld程序解析

package main

import "fmt"

func main() {

/\* 这是第一个简单的程序 \*/

fmt.Println("Hello, World!")

}

程序解析

1. package main 定义了包名。必须在源文件中非注释的第一行指明这个文件属于哪个包，如：package main。package main表示一个可独立执行的程序，每个 Go 应用程序都包含一个名为 main 包。

2. import "fmt" 告诉 Go 编译器这个程序需要使用 fmt 包，fmt 包实现了格式化 IO（输入/输出）的函数。

3. func main() 是程序入口。main 函数是每一个可执行程序所必须包含的，一般来说都是在启动后第一个执行的函数，如果有 init() 函数则会先执行init()函数。

4. /\*...\*/ 是注释，在程序执行时将被忽略。单行注释是最常见的注释形式，你可以在任何地方使用以 // 开头的单行注释。多行注释也叫块注释，均已以 /\* 开头，并以 \*/ 结尾，且不可以嵌套使用，多行注释一般用于文档描述或代码片段。

5. fmt.Println(...) 可以将字符串输出到控制台，并在最后自动增加换行字符 \n。 使用 fmt.Print("hello, world\n") 可以得到相同的结果。

### Go语言编码规范

1.注释

单行注释是最常见的注释形式，你可以在任何地方使用以 // 开头的单行注释

多行注释也叫块注释，均已以 /\* 开头，并以 \*/ 结尾，且不可以嵌套使用，多行注释一般用于文档描述或注释成块的代码片段

2.标识符

标识符用来命名变量、类型等程序实体。一个标识符实际上就是一个或是多个字母(A~Z和a~z)数字(0~9)、下划线\_组成的序列，但是第一个字符必须是字母或下划线而不能是数字。

Go不允许在标识符中使用@、$和%等标点符号。

Go是一种区分大小写的编程语言。因此，Manpower和manpower是两个不同的标识符。

以下是无效的标识符：

* 1xy（以数字开头）
* case（Go 语言的关键字）
* chan（Go 语言的关键字）
* x+y（运算符是不允许的）

3.Go 语言的空格

Go 语言中变量的声明必须使用空格隔开，如：var age int;语句中适当使用空格能让程序更易阅读。在变量与运算符间加入空格，程序看起来更加美观，如：a = x + y;

4.语句的结尾

在 Go 程序中，一行代表一个语句结束。Go语言中是不需要类似于Java需要分号结尾，因为这些工作都将由 Go 编译器自动完成；

如果打算将多个语句写在同一行，它们则必须使用分号“;”人为区分，但在实际开发中并不鼓励这种做法。

5.可见性规则

Go语言中，使用大小写来决定标识符（常量、变量、类型、接口、结构或函数）是否可以被外部包所调用。

以一个大写字母开头，那么使用这种形式的标识符的对象就可以被外部包的代码所使用（使用时程序需要先导入这个包），如同面向对象语言中的 public。

如果以小写字母开头，则对包外是不可见的，但是他们在整个包的内部是可见并且可用的，像面向对象语言中的 private 。

### Go语言关键字及保留字

下面列举了 Go 代码中会使用到的 25 个关键字或保留字：

表1.1 Go语言关键字或保留字

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| break | default | func | interface | select |
| case | defer | go | map | struct |
| chan | else | goto | package | switch |
| const | fallthrough | if | range | type |
| continue | for | import | return | var |

除了以上介绍的这些关键字，Go 语言还有 36 个预定义标识符：

表1.2 Go语言预定义标识符

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| append | bool | byte | cap | close | complex | complex64 | complex128 | uint16 |
| copy | false | float32 | float64 | imag | int | int8 | int16 | uint32 |
| int32 | int64 | iota | len | make | new | nil | panic | uint64 |
| print | println | real | recover | string | true | uint | uint8 | uintptr |

### Go 程序结构组成

Go 文件的基本组成包括：包声明、引入包、函数、变量、语句 & 表达式、注释。

// 当前程序的包名

package main

// 导入其他包

import . "fmt"

// 常量定义

const PI = 3.14

// 全局变量的声明和赋值

var name = "gopher"

// 一般类型声明

type newType int

// 结构的声明

type gopher struct{}

// 接口的声明

type golang interface{}

// 由main函数作为程序入口点启动

func main() {

Println("Hello World!")

}

[第1章 初识Go语言 1](#_Toc1712565897)

[1.1 Go语言的发展历史 1](#_Toc371710138)

[1.1.1 知名编程语言发展简史 1](#_Toc302360243)

[1.1.2 Go语言主要发展过程 2](#_Toc822295299)

[1.2 Go语言的特点及优势 3](#_Toc1259821848)

[1.2.1 Go语言设计初衷 3](#_Toc1784523563)

[1.2.2 Go语言的优势 4](#_Toc730909339)

[1.2.3 Go语言的核心特性和优势 5](#_Toc786799733)

[1.2.4 采用Go语言的国内外知名企业 5](#_Toc1686297952)

[1.2.5 Go语言学习的开发文档 5](#_Toc1267989805)

[1.3 Golang安装和配置 6](#_Toc1624423454)

[1.3.1 下载 6](#_Toc725387067)

[1.3.2 windows系统下安装及配置环境变量 8](#_Toc315771050)

[1.3.3 mac系统下安装及配置环境变量 9](#_Toc731945613)

[1.4 Goland安装和配置 10](#_Toc1023587675)

[1.4.1 安装Goland开发工具 10](#_Toc2094041255)

[1.4.2 下载及安装 10](#_Toc1589365749)

[1.4.3 使用Goland 11](#_Toc2068542057)

[1.4.4 编写第一个程序HelloWorld 13](#_Toc373590716)

[1.5 Go语言编码规范 13](#_Toc1844463631)

[1.5.1 HelloWorld程序解析 13](#_Toc973801772)

[1.5.2 Go语言编码规范 14](#_Toc713508217)

[1.5.3 Go语言关键字及保留字 16](#_Toc383918271)

[1.5.4 Go 程序结构组成 16](#_Toc1473505109)