

2021 年 Go 语言入门

作者: [chirsgod](#)

原文链接: <https://ld246.com/article/1633932206163>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

2021 年 Go 语言入门

1. 前言

1.1 参考

本篇文章参考了煎鱼的[跟煎鱼学 Go](#)系列文章。教程写的非常好，推荐大家学习一下。

1.2 Go 语言是什么？

Go语言是一门编译性语言，在你运行程序之前，Go首先使用编译器把你的代码转换成机器能够读懂1和0。它会把你所有的代码编译成一个可执行文件（Windows环境下为.exe），在编译过程中，Go译器能够捕获一些错误。

不是所有语言都使用该方式，Python、Ruby等很多语言都是解释性语言。

1.3 Go语言的由来和评价

我们想要一个安全的、静态编译的、高性能的、类似C++和Java这样的语言，但是得更轻量级并且要Python这种动态解释型语言这样有趣。——Rob Pike

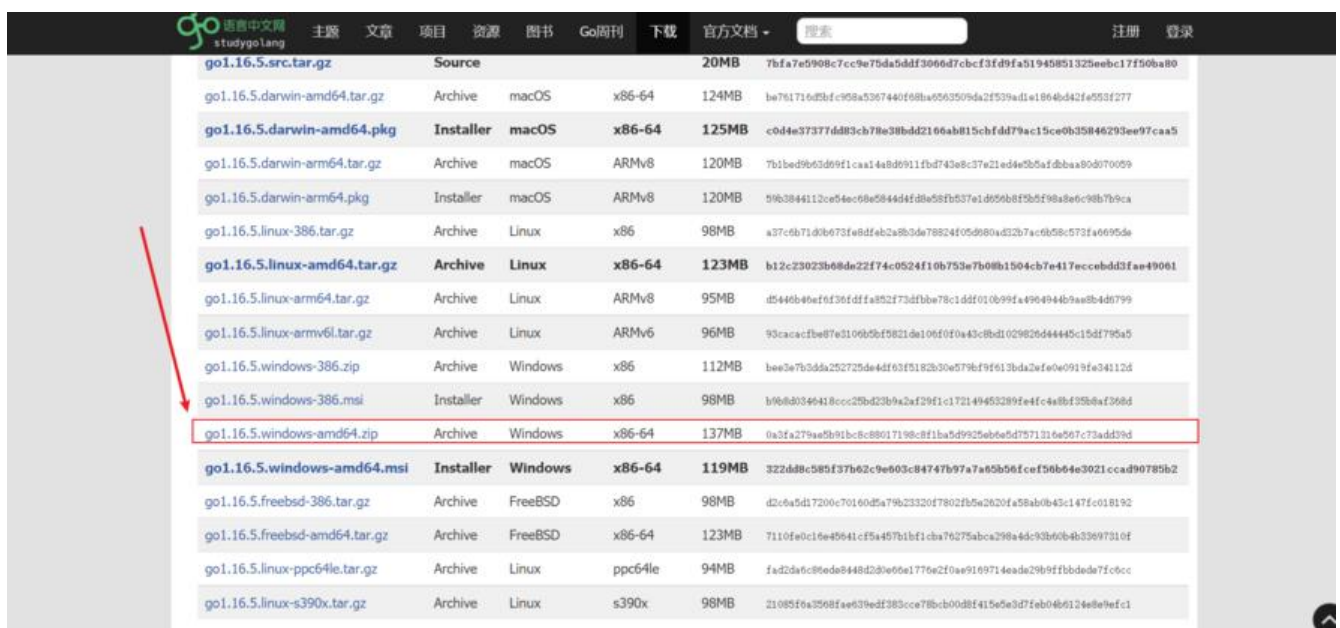
Go 是一个开源的编程语言，能够构建大规模简单、高效、可靠的软件。

2. Go 语言的环境安装与配置

2.1 Windows 系统下的安装

推荐进[Go 语言中文网](#)进行下载

Windows 环境下有现成的安装包，但是会自动安装环境变量。因为 Go 历史环境所导致的影响，所环境变量最好咱们自己配置。

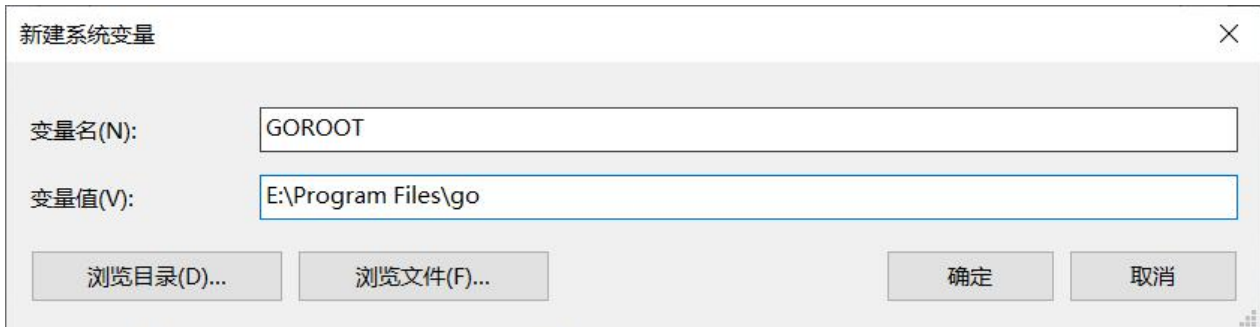


| 名称 | 类型 | 平台 | 架构 | 大小 | 哈希 |
|-------------------------------|-----------|---------|---------|-------|--|
| go1.16.5.src.tar.gz | Source | | | 20MB | 7bfa7e998c7cc9e75da5dd13066d7c9cf3fd9fa5194085132beebc17f50ba80 |
| go1.16.5.darwin-amd64.tar.gz | Archive | macOS | x86-64 | 124MB | be761716d5bfc958a5367440f68ba6563509da2f539ad1e1864b42fe593f277 |
| go1.16.5.darwin-amd64.pkg | Installer | macOS | x86-64 | 125MB | c0d4e37377dd83cb78e38bd42166ab815chfd99ac15ce0b35846293ee97caa5 |
| go1.16.5.darwin-arm64.tar.gz | Archive | macOS | ARMv8 | 120MB | 7b1bed963d09f1caa14a8d911fb743e8c37e21e4e9b5afdbaa80d070059 |
| go1.16.5.darwin-arm64.pkg | Installer | macOS | ARMv8 | 120MB | 99b3844112ce54e68e5844d4f8e58fb537e1d6568f5b5f98a8e6c98b7b9ca |
| go1.16.5.linux-386.tar.gz | Archive | Linux | x86 | 98MB | a37c0b71d0b073fe8dfab2a8b3da78824f05d980a432b7ac0b98c573fa6695de |
| go1.16.5.linux-amd64.tar.gz | Archive | Linux | x86-64 | 123MB | b12c23023b68de22f74c0524f10b753e7b08b1504cb7e417eccbddd3fae49061 |
| go1.16.5.linux-arm64.tar.gz | Archive | Linux | ARMv8 | 95MB | d544b49ef6f36fddf8a52f73dfb78c1ddf010b99fa4904944b9aa5b4d6799 |
| go1.16.5.linux-armv6l.tar.gz | Archive | Linux | ARMv6 | 96MB | 93cacacf8e7e3106b06f0821da106f09a43c8bd1029826d4445c15df795a5 |
| go1.16.5.windows-386.zip | Archive | Windows | x86 | 112MB | bee3e7b3da252725e4df63f5182b30e379ef9f613bda2ef0e0919fe34112d |
| go1.16.5.windows-386.msi | Installer | Windows | x86 | 98MB | b968d0349418ccc25bd23b9a2af29f1c172149453289fe4fc4a8bf358a4368d |
| go1.16.5.windows-amd64.zip | Archive | Windows | x86-64 | 137MB | 0a3fa279ae5b91bc8c8017198c8f1ba5d9925eb6e5d751316e567c73add39d |
| go1.16.5.windows-amd64.msi | Installer | Windows | x86-64 | 119MB | 322d88c585f37b62c9e603c84747b97a7a65b56fcef56b64e3021ccad90785b2 |
| go1.16.5.freebsd-386.tar.gz | Archive | FreeBSD | x86 | 98MB | d2c0a5d17200c70160d5a79b23320f7802fb5e2020a58ab0b43c147f018192 |
| go1.16.5.freebsd-amd64.tar.gz | Archive | FreeBSD | x86-64 | 123MB | 7110f0c1e49641cf54457b1bf1c8a70275abca298a46c93b0b4833897310f |
| go1.16.5.linux-ppc64le.tar.gz | Archive | Linux | ppc64le | 94MB | fad2d6c86ede8448d28e06e1776e2f0ae9169714eade29b9fbbd0e7fc0cc |
| go1.16.5.linux-s390x.tar.gz | Archive | Linux | s390x | 98MB | 21085f6a3268fae639e3f383c0e78bc00d8f415e5e3d7feb0466124e89efc1 |

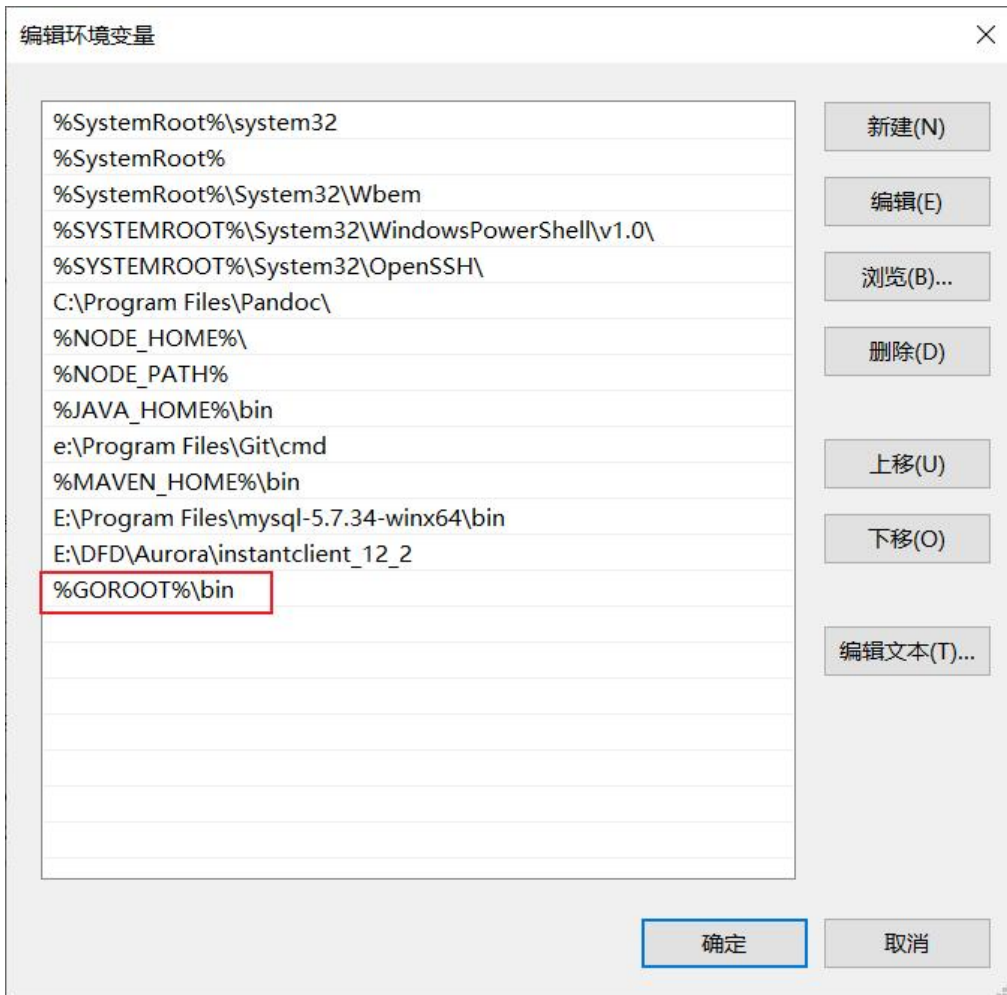
下载完成后，将压缩包如同 JDK 一样解压到你想要指定的目录就好

2.2 环境变量配置

首先在系统变量里新建一个变量为**GOROOT**，然后指定刚才的解压路径。



然后进入系统变量**Path**，增加对**GOROOT**的引用



打开 CMD 窗口输入下列命令验证，出现版本号信息即为成功。

go version

```
Microsoft Windows [版本 10.0.19043.1055]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Ljy>go version
go version go1.16.5 windows/amd64
```

2.3 开启 Go Modules

在很久很久之前 Go 的包管理一直就是个大问题，而且写 Go 的代码只能写到 GOPATH 这个指定好文件夹里。Go 语言的初衷是一个工程化的语言，于是在社区呼吁着解决该问题的人们越来越多。于是 Go Modules 在 Go 的主理人 Russ Cox 的强力推动下诞生了。从此以后 Go 语言有了靠谱的包管理咱们的代码也不用指定写到某一个位置上了。

打开CMD窗口输入下列命令开启：

```
go env -w GO111MODULE=on
```

2.4 更改 GOPROXY

可以这么理解一些 Go 的代码和框架都上传到了一个仓库，可以通过拉取等手段为我们所用，但是这外网。。。。。。类似于 Java 的 Maven 设置阿里云仓库，咱们 Go 语言设置七牛云仓库。

打开CMD窗口输入下列命令开启：

```
go env -w GOPROXY=https://goproxy.cn,direct
```

2.5 验证

打开CMD窗口输入下列命令验证：

```
go env
```

看一下GOROOT，GO111MODULE和GOPROXY三个值的情况

```
C:\Users\Ljy>go env -w GOPROXY=https://goproxy.cn,direct
C:\Users\Ljy>go env
set GO111MODULE=on
set GOARCH=amd64
set GOBIN=
set GOCACHE=C:\Users\Ljy\AppData\Local\go-build
set GOENV=C:\Users\Ljy\AppData\Roaming\go\env
set GOEXE=.exe
set GOFLAGS=
set GOHOSTARCH=amd64
set GOHOSTOS=windows
set GOINSECURE=
set GOMODCACHE=C:\Users\Ljy\go\pkg\mod
set GONOPROXY=
set GONOSUMDB=
set GOOS=windows
set GOPATH=C:\Users\Ljy\go
set GOPRIVATE=
set GOPROXY=https://goproxy.cn,direct
set GOROOT=E:\Program Files\go
set GOSUMDB=sum.golang.org
set GOTMPDIR=
set GOTOOLDIR=E:\Program Files\go\pkg\tool\windows_amd64
set GOVCS=
set GOVERSION=go1.16.5
set GCCGO=gccgo
set AR=ar
set CC=gcc
set CXX=g++
```

3. IDE 配置

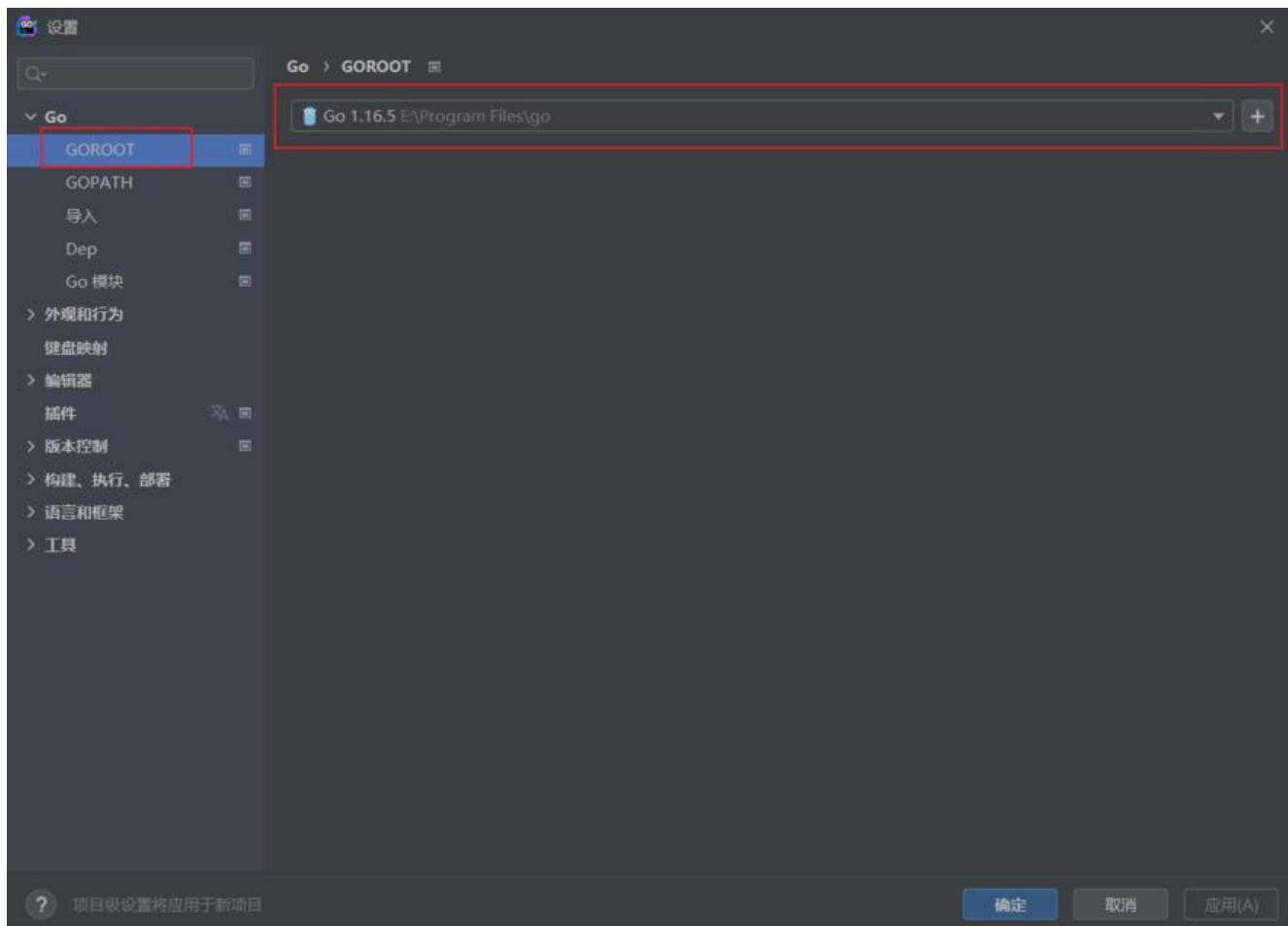
3.1 软件下载

关于 Go 语言的 IDE，推荐两个一个是[Goland](#)，另一个是 VsCode。

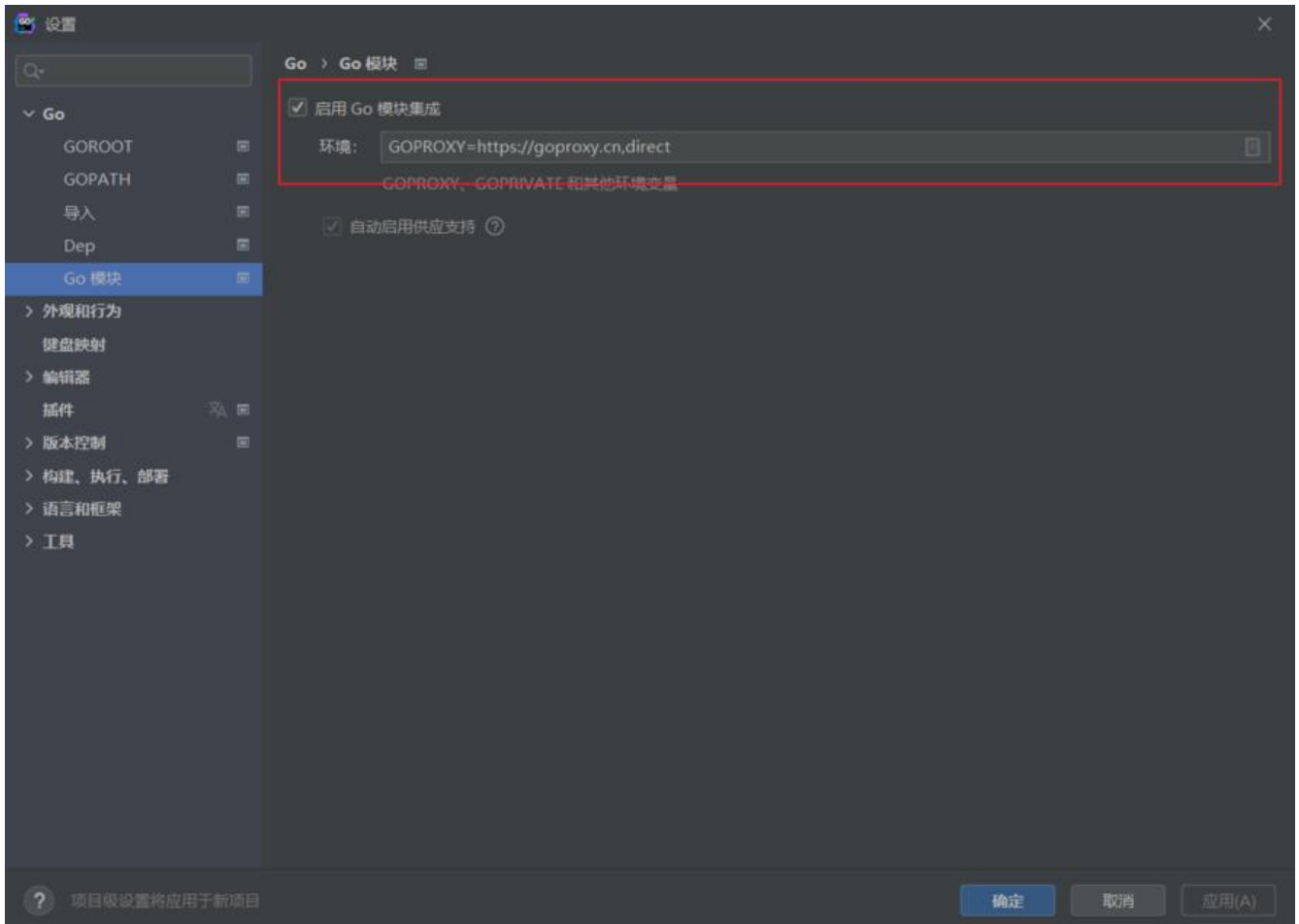
在这里只讲对于 Goland 的配置，点击[超链接](#)进行下载，安装过程不再过多赘述。

3.2 GOROOT 设置

打开 Goland 的设置，勾选系统的 Go 安装路径



3.3 Go 模块



`GOPROXY=https://goproxy.cn,direct`

4. 测试

新建项目，新建一个go文件，输入下列代码，进行测试。

```
package main

func main() {
    println("hello world")
}
```

执行构建命令

```
go build xx.go
```

执行exe文件

```
xx.exe
```

输出结果

```
hello world
```