

[笔记][尚硅谷韩顺平Go语言核心编程][01]

Go

[笔记][尚硅谷韩顺平Go语言核心编程][01]

- 001. Golang 可以做什么
- 002. 如何学习 Golang 更高效 1
- 003. 如何学习 Golang 更高效 2
- 004. 程序的基本概念
- 005. Go 语言发展简史
- 006. Go 语言的特性和快速体验
- 007. Golang 开发工具介绍
- 008. windows 下 VS Code 的安装和使用(1)
- 009. windows 下 VS Code 的安装和使用(2)
- 010. Linux 下 VS Code 的安装和使用
- 011. Mac 下 VS Code 的安装和使用
- 012. Windows 下搭建 Go 开发环境
- 013. Linux 下搭建 Go 开发环境
- 014. Mac 下搭建 Go 开发环境

001. Golang 可以做什么

Go 学习方向

- 区块链研发工程师
- Go 服务端/游戏软件工程师
- Go 分布式/云计算软件工程师

Go 语言，我们可以简单的写成 Go lang，lang 代表 language。

区块链实际上是一种分布式账本技术，核心思想是去中心化，让人人都可以参与数据库的记录。

Go 语言的主要设计者之一肯汤姆森，是 C 语言的设计者，所以 C 和 C++ 能做的，Go 也能做。

Go 在数据处理和大并发比较厉害。

Go 的应用领域

- 区块链技术，简称 **BT** (Blockchain Technology)，是一种互联网数据库技术，其特点是**去中心化、公开透明**，让**每个人均可参与数据库记录**。
- 后端服务器应用
 - 美团后台流量支撑程序
支撑主站后台流量（排序、推荐、搜索等），提供负载均衡，**cache**，容错，按条件分流，统计运行指标（**qps**，**latency**）等功能
 - 仙侠道
游戏服务端（通信、逻辑、数据存储）
- 云计算/云服务后台应用
 - 盛大云CDN（内容分发网络）
CDN 的调度系统、分发系统、监控系统、短域名服务，**CDN** 内部开放平台、运营报表系统以及其他一些小工具等。
 - 京东消息推送云服务/京东分布式文件系统

CDN 的目标就是让用户更快的访问网络。

002. 如何学习 Golang 更高效 1



Go 的吉祥物是金花鼠 **gordon**。

学习 **Go** 编程方法的理解：

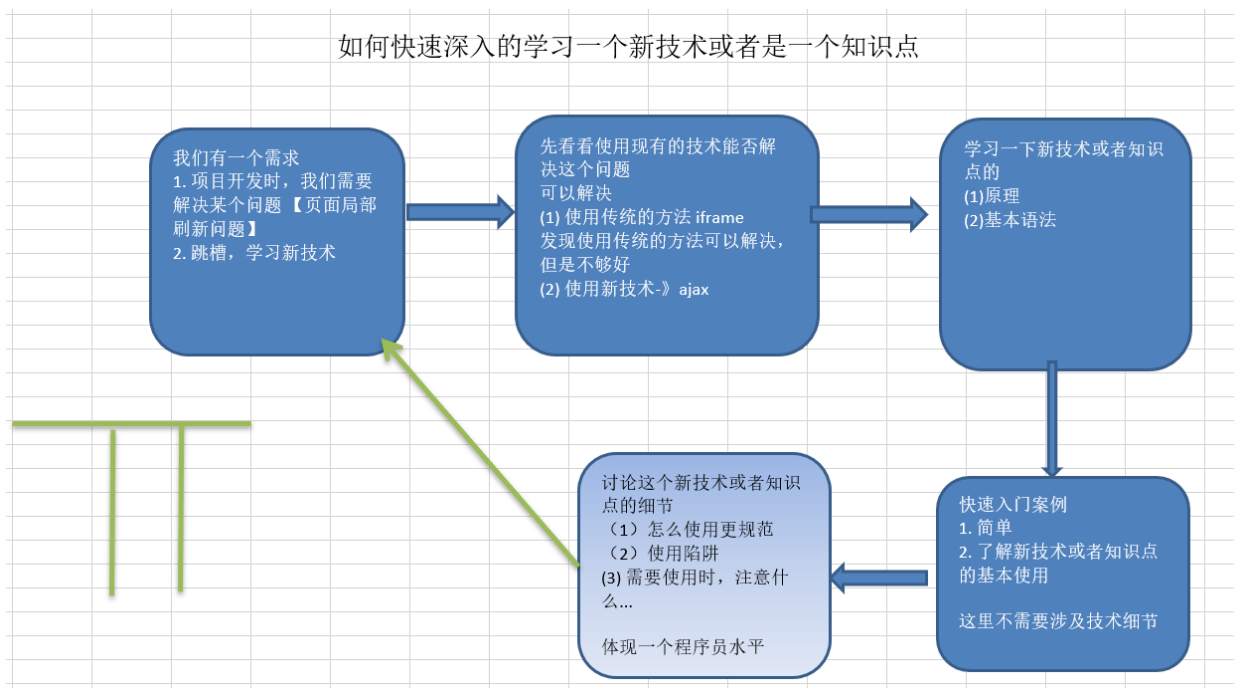
- 高效而愉快的学习
- 先建立一个整体框架，然后细节
- 在实际工作中，要培养用到什么，能够快速学习什么的能力
- 先 know how，再 know why
- 软件编程是一门“做中学”的学科，不是会了再做，而是做了才会
- 适当的囫圇吞枣
- 学习软件编程是在琢磨别人怎么做，而不是我认为应该怎么做的过程

003. 如何学习 Golang 更高效 2

授课方式说明:

- 努力做到通俗易懂
- 注重 Go 语言体系，同时也兼顾技术细节
- 在实际工作中，如何快速的掌握一个技术，同时也是我们授课的思路。
(怎么讲解或者学习一个技术)

学习方法介绍

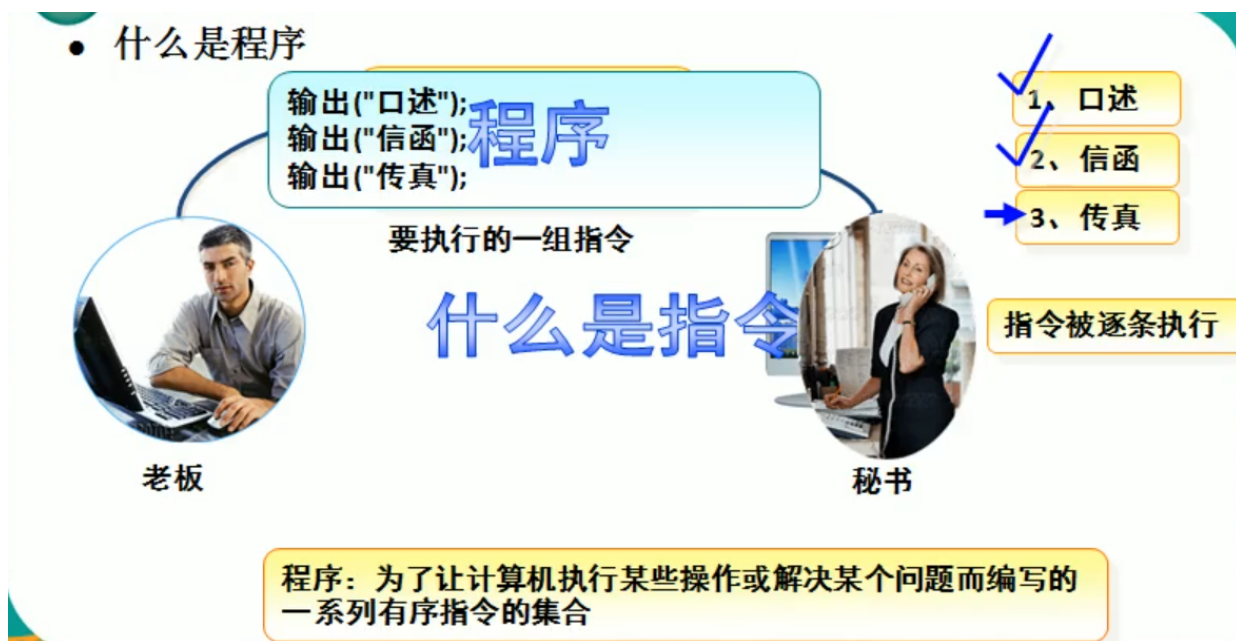
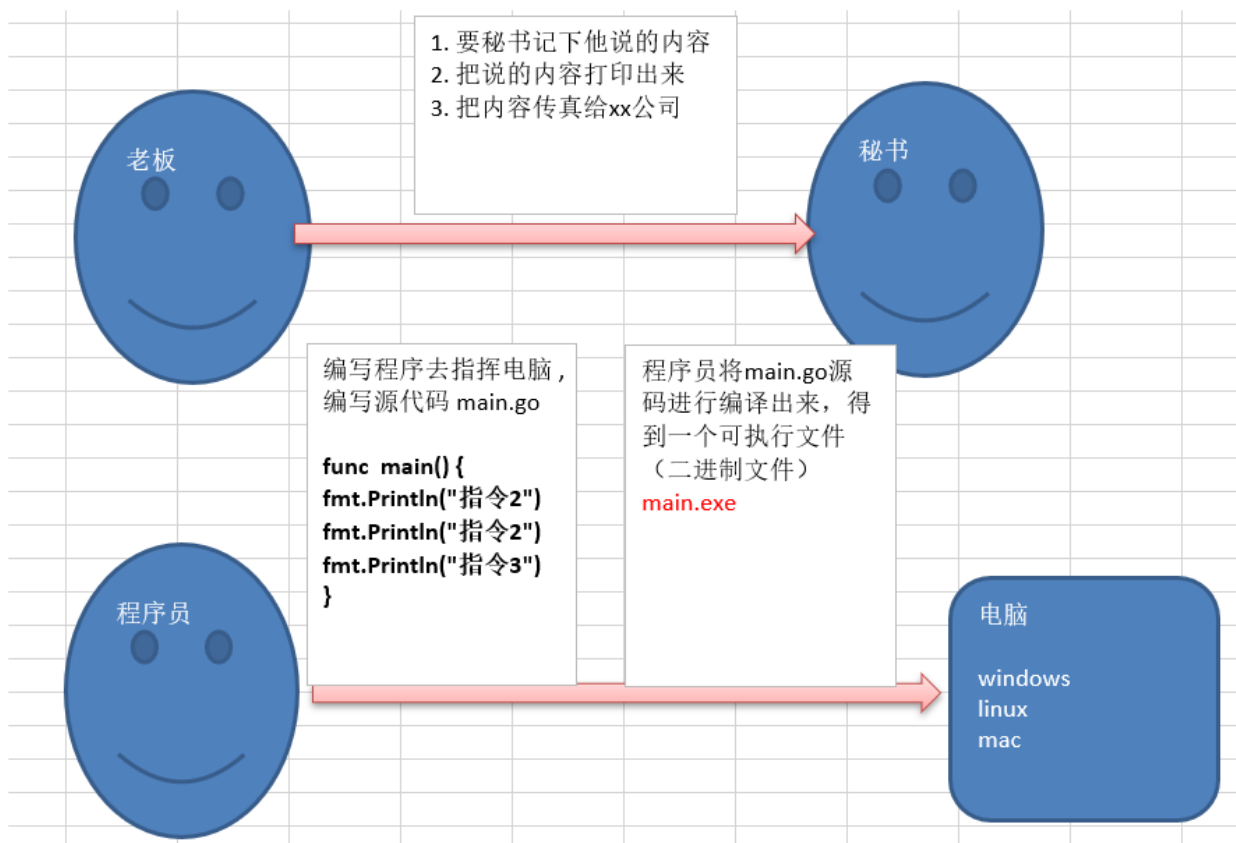


技术细节（第五步）的学习是最耗费时间的！

左边那个 π ，说的是程序员有广度也有深度，要能钻下去。

004. 程序的基本概念

程序：完成某个功能的指令集合。



005. Go 语言发展简史

Go 语言的核心开发团队：

- Ken Thompson，肯汤普森，和 Dennis Ritchie 是 Unix 的原创者。也是 C 语言的主要发明人。



- **Rob Pike**，罗布派克，肯汤普森的同事，共创出 **UTF-8** 字节编码。（他的妻子设计了金花鼠 **Logo**）



- **Robert Griesemer**，曾协助制作 **Java** 的 **HotSpot** 编译器，和 **Chrome** 浏览器的 **JavaScript** 引擎 **V8**。

Google 为什么要创造 **Go** 语言：

- 计算机硬件技术更新频繁，性能提高很快。目前的主流编程语言发展明显落后于硬件，不能合理利用**多核多CPU**的优势提升软件系统性能。
- 软件系统复杂度越来越高，维护成本越来越高，目前缺乏一个足够简洁高效的编程语言。
- 企业运行维护 **C/C++** 的项目，这些程序运行速度虽然很快，但是编译速度却很慢，同时还存在内存泄漏的一系列的困扰需要解决。

Go 语言发展简史

- 2007 年，谷歌工程师 Rob Pike, Ken Thompson 和 Robert Griesemer开始设计一门全新的语言，这是 **Go** 语言的最初原型。
- 2009 年 11 月 10 日，Google 将 **Go** 语言以开放源代码的方式向全球发布。
- 2015 年 8 月 19 日，**Go 1.5** 版发布，本次更新中移除了“最后残余的C代码”
- 2017 年 2 月 17 日，**Go** 语言 **Go1.8** 版发布。
- 2017 年 8 月 24 日，**Go** 语言 **Go1.9** 版发布。1.9.2版本
- 2018 年 2 月 16 日，**Go** 语言 **Go1.10** 版发布。

本课程讲的是 1.9.2 版本

006. **Go** 语言的特性和快速体验

简介:

Go 语言保证了既能到达静态编译语言的安全和性能，又达到了动态语言开发维护的高效率，使用一个表达式来形容 Go 语言： $Go = C + Python$ ，说明 Go 语言既有 C 静态语言程序的运行速度，又能达到Python动态语言的快速开发。

1) 从 C 语言中继承了很多理念，包括表达式语法，控制结构，基础数据类型，调用参数传值，指针等等，也保留了和 C 语言一样的编译执行方式及弱化的指针
举一个案例(体验):

```
//go语言的指针的使用特点(体验)
func testPtr(num *int) {
    *num = 20
}
```

2) 引入包的概念，用于组织程序结构，Go 语言的一个文件都要归属于一个包，而不能单独存在。

```
package main // 一个 go 文件需要在一个包

import "fmt"
func sayOK() {
    // 输出一句话
    fmt.Println("OK")
}
```

3) 垃圾回收机制，内存自动回收，不需要开发人员管理

4) 天然并发

- 从语言层面支持并发，实现简单
- goroutine 轻量级线程，可实现大并发处理，高效利用多核
- 基于 CPS 并发模型 (Communicating Sequential Processes) 实现

5) 吸收了管道通信机制，形成 Go 语言特有的管道 channel，通过管道 channel，可以实现不同的 goroutine 之间的相互通信。

6) 函数返回多个值

```
func getSumAndSub(n1 int, n2 int)(int, int) {
    // go 语言后面不要分号，编译器会给你加
    sum := n1 + n2
    sub := n1 - n2
    return sum, sub
}
```

7) 新的创新：比如切片、延时执行 defer 等

007. Golang 开发工具介绍

开发工具介绍

- VS Code
- Sublime Text
- Vim
- Emacs
- Eclipse IDE (装 GoEclipse 插件)
- LiteIDE 中国人开发的
- Goland

工具选择

- 如何选择开发工具
先选择 VS Code 或者 Vim 等文本编辑器，等到对 Go 有一定了解之后，再使用 Eclipse 等 IDE 开发工具。
- 为什么？
更深刻地理解 Go 语言技术，培养代码感，找到写代码的感觉。
有利于公司面试：给你纸，让你写程序。

本课程选择 VS Code 。

008. windows 下 VS Code 的安装和使用(1)

.tar.gz 表示解压后即可使用

演示在 windows 下安装 vscode 并使用
略

009. windows 下 VS Code 的安装和使用(2)

```
package main // 把这个 test.go 归属到 main 包
import "fmt" // 引入一个包 fmt

func main() {
    // 输出 hello
}
```

```
fmt.Println("hello")
}
```

010. Linux 下 VS Code 的安装和使用

- 首先下载安装文件
老师下载的是 `tar.gz` 安装包，我直接下载的是 `.deb`。
- 使用 `xftp5` 可以在 `windows` 和虚拟机的 `Ubuntu` 之间传递文件。
- 如果在 `Ubuntu` 安装，建议安装到 `/opt` 目录下，一般软件都会安装在这个目录。
- 切换到超级用户 `su root`，输入密码。然后拷贝传过来的 `tar.gz` 文件到 `/opt` 目录。
- 解压缩 `tar -zxvf xxx.tar.gz`
- 现在进入解压后的目录，即可运行 `VS Code`
- 绿色的文件就是可执行文件，`./code` 运行

```
package main
import "fmt"

func main() {
    fmt.Println("hello")
}
```

011. Mac 下 VS Code 的安装和使用

把 `VS Code` 传输到 `Mac` 的时候，默认 `Mac` 没有启动 `ssh` 服务，所以需要我们启动一下，才能远程传输文件。

- 开启 `sshd` 服务
`sudo launchctl load -w /System/Library/LaunchDaemons/ssh.plist`
- 关闭 `sshd` 服务
`sudo launchctl unload -w /System/Library/LaunchDaemons/ssh.plist`
- 查看 `sshd` 服务是否开启
`sudo launchctl list | grep ssh`

```
🔗 sudo launchctl list | grep ssh
- 0 com.openssh.sshd
```

解压缩指令 `unzip xxx.zip`

`test.go`


```
package main

import "fmt"

func main() {
    fmt.Println("hello")
}
```

012. Windows 下搭建 Go 开发环境

安装和配置SDK

SDK：软件开发工具包 [Software Development Kit](#)

SDK 是提供给开发人员使用的，其中包含了对应开发语言的工具包。

SDK下载

Go 官网：golang.org

如何选择合适的SDK版本？

1.9.2	
Filename	Size
go1.9.2.darwin-amd64.pkg	97 M
go1.9.2.darwin-amd64.tar.gz	97 M
go1.9.2.freebsd-386.tar.gz	86 M
go1.9.2.freebsd-amd64.tar.gz	97 M
go1.9.2.linux-386.tar.gz	87 M
go1.9.2.linux-amd64.tar.gz	99 M
go1.9.2.linux-arm64.tar.gz	84 M
go1.9.2.linux-armv6l.tar.gz	85 M
go1.9.2.linux-ppc64le.tar.gz	83 M
go1.9.2.linux-s390x.tar.gz	83 M
go1.9.2.src.tar.gz	15 M
go1.9.2.windows-386.msi	78 M
go1.9.2.windows-386.zip	91 M
go1.9.2.windows-amd64.msi	88 M
go1.9.2.windows-amd64.zip	104 M

mac下的sdk
1. pkg 图形化安装包
2. tar.gz是解压就可以使用

unix 下的sdk

如果你的linux是32位系统： 386.tar.gz如果是：64位系统，选择amd.tar.g

windows下的sdk; 我们使用.zip 如果你是32的, 选择 -386.zip, 如果是64位

团子注：上面的截图来自于 [golang 中国](#)，现在找不到这个页面，下面是官网的版本展示。

go1.9.2 ▾

File name	Kind	OS	Arch	Size	SHA256 Checksum
go1.9.2.src.tar.gz	Source			16MB	665f184bf8ac89986cfd5a4460736976f60b57df6b320ad71ad4cef53bb143dc
go1.9.2.darwin-amd64.tar.gz	Archive	macOS	x86-64	98MB	73fd5840d55f5566d8db6c0ffdd187577e8ebe650c783f68bd27cbf95bde6743
go1.9.2.darwin-amd64.pkg	Installer	macOS	x86-64	97MB	8b4f6ae6deae1150d2e341d02c247fd18a99af387516540eeb84702ffd76d3a1
go1.9.2.freebsd-386.tar.gz	Archive	FreeBSD	x86	86MB	809dcb0a8457c8d0abf954f20311a1ee353486d0ae3f921e9478189721d37677
go1.9.2.freebsd-amd64.tar.gz	Archive	FreeBSD	x86-64	98MB	8be985c3e251c8e007fa6ecd0189bc53e65cc519f4464dd19fa11f7ed251134
go1.9.2.linux-386.tar.gz	Archive	Linux	x86	88MB	574b2c4b1a248e58ef7d1f825beda15429610a2316d9cbd3096d8d3fa8c0bc1a
go1.9.2.linux-amd64.tar.gz	Archive	Linux	x86-64	99MB	de874549d9a8d8d8062be0580509c09a88a248e77ec14eb77453530829ac02b
go1.9.2.linux-arm64.tar.gz	Archive	Linux	ARMv8	84MB	0016ac65ad8340c84f51bc11d8b24ee8265b0a4597dbf8d91776fc187456fa
go1.9.2.linux-armv6l.tar.gz	Archive	Linux	ARMv6	86MB	8a6758c8d390e28ef2bcea511f62dcb43056f38c1addc06a8bc996741987e7bb
go1.9.2.linux-ppc64le.tar.gz	Archive	Linux	ppc64le	84MB	adb440b2b6ae9e448c253a20836d8e8aa4236f731d87717d9c7b241998dc7f9d
go1.9.2.linux-s390x.tar.gz	Archive	Linux	s390x	83MB	a7137b4fbdec126823a12a4b696eeee2f04ec616e9fb8a54654c51d55884c1345
go1.9.2.windows-386.zip	Archive	Windows	x86	92MB	35d3be5d7b97c6d11ffb76c1b19e20a824e427805ee918e82c08a2e5793eda20
go1.9.2.windows-386.msi	Installer	Windows	x86	79MB	020ea4a53093dd98b5ad074c4e493ff52be0aa71eee89dc24ca7783cb528de97
go1.9.2.windows-amd64.zip	Archive	Windows	x86-64	104MB	682ec3626a9c45b657c2456e35cadad119057408d37f334c6c24d88389c2164c
go1.9.2.windows-amd64.msi	Installer	Windows	x86-64	90MB	daeb761aa6fdb22dc3954fd911963b347c44aa5c6ba974b9c01be7cbbd6922ba

安装路径不要有中文字符和空格。

团子注：老师用解压安装，我选择 `.msi` 文件。



`go.exe` 可以编译和运行 `go` 源码

如何测试 `go` 的 `sdk` 安装成功？

```
PS C:\software\Go> go version
go version go1.9.2 windows/amd64
```

团子注：我使用 `.msi` 安装，已经配置好了环境变量，如果采用压缩包的方式，还需要配置 `path` 环境变量。

为什么需要配置环境变量

如果执行的程序或命令在当前目录下找不到，`windows` 系统会在已有的一个叫做 `path` 的环境变量指定的目录中查找。如果仍未找到，会提示“不是内部或外部命令，也不是可运行的程序或批处理文件”。

需要配置哪些环境变量？

环境变量	说明
GOROOT	指定SDK的安装路径 <code>d:/programs/go</code>
Path	添加SDK的/bin目录
GOPATH	工作目录，将来我们的go项目的工作路径

环境变量分为**系统环境变量**，意思是所有用户都能用，还有**用户环境变量**，只有当前用户能用得到。

如何配置环境变量？

- 首先配置 `GOROOT`，它是 `go` 语言 `sdk` 安装的路径
- 编辑 `Path` 环境变量，在最末尾增加 `;%GOROOT%\bin`（`win7` 需要这样配置）
- 配置 `GOPATH`，它是 `go` 项目的路径，即工作目录。
- 重新开启命令行，再试验一次 `go version`。

团子注：我没有配置 `GOPATH`，不知道这个有什么用。

013. Linux 下搭建 Go 开发环境

- 查看当前 `Linux` 到底是 `32` 位还是 `64` 位

```
jpch89@jpch89-ubuntu:~$ uname -a
Linux jpch89-ubuntu 5.3.0-26-generic #28~18.04.1-Ubuntu SMP Wed Dec 18
16:40:14 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
```

如果出现了 `64`，那么就是 `64` 位的。

- 下载 `Linux` 版本的 `SDK`，并且是 `64` 位的：`go1.9.2.linux-amd64.tar.gz`
- 拷贝到 `sudo cp go1.9.2.linux-amd64.tar.gz /opt`

- 切换到超级用户 `su -`，切换到 `/opt` 目录

团子注：`su -` 默认切换到超级用户，`su - username` 则切换到 `username` 用户。
带 `-` 时会切换到对应用户的环境变量，并且切换到对应用户的家目录 `~`
不带 `-` 时保持之前的环境变量不变，而且目录也保持之前的不变

- 解压缩 `tar -zxvf go1.9.2.linux-amd64.tar.gz`
- 然后 `cd /opt/go/bin` 验证是否安装成功
- `./go version`，注意，一定不能少 `./`，要不然会提示找不到命令

```
jpch89@jpch89-ubuntu:/opt/go/bin$ go version
```

Command 'go' not found, but can be installed with:

```
sudo apt install golang-go
sudo apt install gccgo-go
```

```
jpch89@jpch89-ubuntu:/opt/go/bin$ ./go version
go version go1.9.2 linux/amd64
```

- 配置环境变量：在 `/etc/profile` 文件下面添三条语句

```
# Golang
export GOROOT=/opt/go
export PATH=$PATH:$GOROOT/bin
export GOPATH=$HOME/goprojects
```

团子注：这样添加的**系统级别**的环境变量！`/etc/profile` 就是 `Linux` 系统的环境变量配置文件。


- 如果需要生效的话，需要注销用户再登录使用！

团子注：有人说，`source /etc/profile` 不用注销，但是这样再开一个终端还是没用。

- 再次 `go versions` 就可以看到版本号了。

014. Mac 下搭建 Go 开发环境

我直接下载了 `*.pkg` 安装，有安装向导，一路下一步。
直接就能在终端使用：

```
 go version  
go version go1.9.2 darwin/amd64
```

团子注：没有配置 `GOROOT` , `GOPATH` 不知道有没有影响

- 使用 `root` 用户，修改 `/etc/profile` , 加入下面三行

```
export GOROOT=$HOME/go_dev/go  
export PATH=$PATH:$GOROOT/bin  
export GOPATH=$HOME/goproject
```

- `wq!` 表示强制保存，如果是只读文件也能写入
- 注销用户，配置才会生效
注销的本质，就是让系统重新读取一下 `/etc/profile` 文件