

# Go语言多平台开发环境如何搭建

---

## Go语言多平台开发环境如何搭建

- 一、课前准备
- 二、课堂主题
- 三、课堂目标
- 四、知识点
  - 1. 为什么要学习和使用Go语言（5分钟）
    - 1.1 为什么要用Go?
    - 1.2 Go语言能做什么?
    - 1.3 Go语言该怎么学?
  - 2. 环境搭建（10分钟）
    - 2.1 linux安装（1分钟）
    - 2.2 macOS安装（5分钟）
    - 2.3 windows安装（2分钟）
    - 2.4 Go语言工具集介绍
    - 2.5 小结
    - 2.6 互动
  - 3. Go语言IDE环境安装（2分钟）
  - 4. 初识golang
- 五、拓展点
- 六、总结
- 七、大作业
- 八、集中答疑
- 九、检测题
- 十、下节课预告

## 一、课前准备

---

说明：提前需要让学生做的课前准备，比如环境安装部署、工具安装、插件安装等，需要学生提前做的都放这，并且给出下载链接或信息源；

- 1. 下载Golang安装包，[下载链接](#)；
- 2. 安装Go-IDE环境，[下载链接](#)；
- 3. [Go语言官方包在线文档](#)；

## 二、课堂主题

---

说明：本堂课的总体概述，明确课堂主题和主体；

本小结主要介绍为什么要学习和使用Go语言，Go语言到底有什么优势，使用Go语言可以做什么，Go语言环境如何安装，IDE工具如何选择和安装等；

## 三、课堂目标

---

说明：主要是让学生了解，学了本堂课后，能达到的一个期望值，要量化；

1. 掌握Go语言的特点和优势；
2. 掌握Go语言开发环境的安装；
3. 掌握Go语言环境的配置；
4. 掌握Go语言IDE的安装；

## 四、知识点

---

说明：

1. 知识点是课堂主体，本堂课所有知识点都列出来；
2. 预估每个知识点需要讲解的时间；
3. 研发逻辑就是讲解逻辑，一般从上往下，遵循：What - Why - How 或 Why - What - How 思路；

### 1. 为什么要学习和使用Go语言（5分钟）

#### 1.1 为什么要用Go？



Go（又称Golang）是Google开发的一种静态强类型、编译型、并发型，并具有垃圾回收功能的编程语言。创作者包括罗伯特·格瑞史莫（Robert Griesemer），罗勃·派克（Rob Pike）及肯·汤普逊（Ken Thompson）、Ian Lance Taylor、Russ Cox。这些在google都是响当当的角色，如果对unix历史有所了解的小伙伴就会知道，Thompson和Dennis.Ritchie被合称为unix之父以及C语言之父，由此可见，go语言创造团队相当强大！学习Go语言的理由可以有以下几点：

- 并发
- 语法简洁
- 开发周期短
- 内存回收
- 运行效率高
- 统一的编程规范
- 活跃的社区文化

## 1.2 Go语言能做什么？

Go语言利用其天生的并发特性，可以很好的开发服务器后端，目前在云计算领域以及区块链行业，Go都受到了热烈追捧，顶顶大名的Docker也是用Go语言开发的，Go语言的产业链正在日趋成熟。

## 1.3 Go语言该怎么学？

对于这么优秀的语言，很多人可能认为学习起来一定非常困难。Go语言并没有，相反它的学习曲线非常平缓，一个开发者学习Go语言非常容易，而且极容易上瘾。学习路径参考如下：

- 语法
- 特性
- 项目实战
- 框架开发
- 源码进阶

## 2. 环境搭建（10分钟）

开发环境始终是学习一门语言最重要的一环，Go语言作为2009年才被推出的新面孔来说，跨平台是必须支持的。Go语言的安装很简单，在Go语言的官网可以下载到安装包，但是由于某些原因，我们访问会有一些问题，但是也有一些好心人将Go语言的安装包同步到了[Go语言中文网](#)，在这里可以很顺利的下载。

### 推荐下载

<b>源码</b> <a href="#">go1.12.7.src.tar.gz</a> (21MB)	<b>Apple macOS</b> macOS 10.8 or later, Intel 64-bit 处理器 <a href="#">go1.12.7.darwin-amd64.pkg</a> (121MB)	<b>Linux</b> Linux 2.6.23 or later, Intel 64-bit 处理器 <a href="#">go1.12.7.linux-amd64.tar.gz</a> (122MB)
<b>Microsoft Windows</b> Windows XP SP2 or later, Intel 64-bit 处理器 <a href="#">go1.12.7.windows-amd64.msi</a> (118MB)		

在上述图片中，我们可以看到，Go语言可以在macOS，linux及Windows平台安装，下面我们各个平台分别介绍一下。

### 2.1 linux安装（1分钟）

在linux平台可以下载安装包安装，也可以使用命令行来操作。以ubuntu系统举例（centOS需要使用yum进行安装），下面两句指令就可以搞定Go语言的安装：

```
sudo apt-get update
```

```
sudo apt-get install golang
```

命令行安装的好处是方便，不过默认安装的版本未必是最新的。下面我们介绍下载安装包的方式安装：

下面介绍使用安装包安装的方式，先下载安装包（amd64代表该版本支持64位操作系统，amd代表的则是cpu的架构，与之对应的是x86）

```
mkdir ~/install  
cd ~/install  
wget https://studygolang.com/dl/golang/go1.12.7.linux-amd64.tar.gz
```

将压缩包解压

```
tar -zxvf go1.12.7.linux-amd64.tar.gz -C ~/
```

设置环境变量

```
echo 'export GOROOT=$HOME/go' >> ~/.bashrc  
mkdir ~/gowork  
echo 'export GOPATH=$HOME/gowork' >> ~/.bashrc  
echo 'export GOBIN=$GOROOT/bin' >> ~/.bashrc  
echo 'export PATH=$GOBIN:$GOPATH/bin:$PATH' >> ~/.bashrc
```

来查看一下环境变量

```
parallels@parallels-vm:~$ go env  
GOARCH="amd64"  
GOBIN=""  
GOCACHE="/home/ubuntu/.cache/go-build"  
GOEXE=""  
GOFLAGS=""  
GOHOSTARCH="amd64"  
GOHOSTOS="linux"  
GOOS="linux"  
GOPATH="/home/ubuntu/gowork"  
GOPROXY=""  
GORACE=""  
GOROOT="/home/ubuntu/go"  
GOTMPDIR=""  
GOTOOLDIR="/home/ubuntu/go/pkg/tool/linux_amd64"  
GCCGO="gccgo"  
CC="gcc"  
CXX="g++"
```

```
CGO_ENABLED="1"
GOMOD=""
CGO_CFLAGS="-g -O2"
CGO_CPPFLAGS=""
CGO_CXXFLAGS="-g -O2"
CGO_FFLAGS="-g -O2"
CGO_LDFLAGS="-g -O2"
PKG_CONFIG="pkg-config"
GOGCCFLAGS="-fPIC -m64 -pthread -fmessage-length=0 -fdebug-prefix-map=/tmp/go-build052037651=/tmp/go-build -gno-record-gcc-switches"
```

## 2.2 macOS安装（5分钟）

在macOS上安装Go语言也可以选择下载安装包和命令行的形式，如果想下载安装包安装的话，可以参考之前在ubuntu上的安装做法，两者是相同的，只不过包名不同。

```
https://studygolang.com/dl/golang/go1.12.7.darwin-amd64.pkg
```

设置环境变量也基本类似，在mac我们设置到~/.bash\_profile文件中

```
echo 'export GOROOT=$HOME/go' >> ~/.bash_profile
mkdir ~/gowork
echo 'export GOPATH=$HOME/gowork' >> ~/.bash_profile
echo 'export GOBIN=$GOROOT/bin' >> ~/.bash_profile
echo 'export PATH=$GOBIN:$GOPATH/bin:$PATH' >> ~/.bash_profile
```

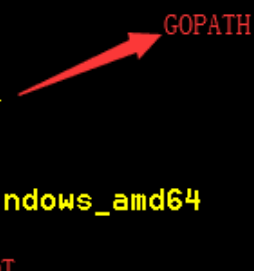
命令行安装也是非常简单，使用brew工具即可！

```
brew install golang
```

## 2.3 windows安装（2分钟）

对于windows的用户来说，下载windows版的msi文件即可，当然安装的过程也非常的傻瓜，各种下一步即可。默认情况下Go语言很关键的环境变量GOROOT和GOPATH在安装的时候会设定好，如果没有，需要手动设置一下这两个环境变量。GOROOT就是安装路径，而GOPATH则是后续要用到的重要路径，可以简单粗暴的认为它是开发路径。在它的下面，会有bin、pkg、src三个子目录，bin是存放二进制文件的目录，pkg是存放包编译后的.a文件，src就是我们的源码路径。（GOROOT与GOPATH的原理在所有平台是相同的）在命令行窗口，可以通过go env指令来查看Go语言的环境变量情况。

```
C:\Users\Administrator>go env
set GOARCH=amd64
set GOBIN=
set GOEXE=.exe
set GOHOSTARCH=amd64
set GOHOSTOS=windows
set GOOS=windows
set GOPATH=D:\myprojects\gowork
set GORACE=
set GOROOT=C:\Go
set GOTOOOLDIR=C:\Go\pkg\tool\windows_amd64
set GCCGO=gccgo
set CC=gcc
set GOGCCFLAGS=-m64 -mthreads -fmessage-length=0 -fdebug-prefix-map=C
91=/tmp/go-build -gno-record-gcc-switches
set CXX=g++
set CGO_ENABLED=1
set CGO_CFLAGS=-g -O2
set CGO_CPPFLAGS=
set CGO_CXXFLAGS=-g -O2
set CGO_FFLAGS=-g -O2
set CGO_LDFLAGS=-g -O2
set PKG_CONFIG=pkg-config
```



## 2.4 Go语言工具集介绍

说明：对本知识点做一个小总结，总结本知识点特色、难点和应用场景，提示学习方法；

我们在使用go help可以查看go所有指令的帮助，对于初学者来说，先记住几个常用的即可。

```
localhost:bin teacher$ go help
Go is a tool for managing Go source code.

Usage:

    go command [arguments]

The commands are:

    build      compile packages and dependencies
    clean      remove object files and cached files
    doc        show documentation for package or symbol
    env        print Go environment information
    bug        start a bug report
    fix        update packages to use new APIs
    fmt        gofmt (reformat) package sources
    generate    generate Go files by processing source
    get        download and install packages and dependencies
    install    compile and install packages and dependencies
    list       list packages
```

run	compile and run Go program
test	test packages
tool	run specified go tool
version	print Go version
vet	report likely mistakes in packages

Use `"go help [command]"` for more information about a command.

Additional help topics:

c	calling between Go and C
buildmode	build modes
cache	build and test caching
filetype	file types
gopath	GOPATH environment variable
environment	environment variables
importpath	import path syntax
packages	package lists
testflag	testing flags
testfunc	testing functions

Use `"go help [topic]"` for more information about that topic.


- go build 编译代码
- go run 编译+运行代码
- go get 下载第三方源码，多数情况下在github下载
- go doc 查看包手册
- go install 编译并且安装包
- go test 测试代码，要求文件必须以test开头
- fmt 重新格式化代码

## 2.5 小结

## 2.6 互动

# 3. Go语言IDE环境安装（2分钟）

Go语言的IDE开发工具有很多，开发者完全可以根据自己喜好来选择，比如GoLand，VsCode，sublime text3等等，本人比较喜欢用LiteIDE。开发人员可以前往[github地址下载](#)。

 <a href="#">liteidex36.archlinux-pkgbuild.zip</a>	2.98 KB
 <a href="#">liteidex36.linux64-qt4.8.7-system.tar.gz</a>	14.8 MB
 <a href="#">liteidex36.linux64-qt4.8.7.ApplImage</a>	22.6 MB
 <a href="#">liteidex36.linux64-qt4.8.7.tar.gz</a>	22.6 MB
 <a href="#">liteidex36.linux64-qt5.5.1-system.tar.gz</a>	14.9 MB
 <a href="#">liteidex36.linux64-qt5.5.1.ApplImage</a>	41 MB
 <a href="#">liteidex36.linux64-qt5.5.1.tar.gz</a>  linux	41.1 MB
 <a href="#">liteidex36.macos-qt5.9.5.zip</a>  macOS	29.7 MB
 <a href="#">liteidex36.windows-qt4.8.5.zip</a>	39.2 MB
 <a href="#">liteidex36.windows-qt5.9.5.zip</a>  windows	36.6 MB
 <a href="#">Source code (zip)</a>	

安装后效果如下图所示：



至于IDE的使用，本文不做详细说明，点一点就会了，不过需要注意一点的是，Go语言是面向工程型的语言，每个工程里只能有一个main，所以同一个目录下有多个包含main函数的文件时是不能使用IDE编译并运行的，这时仍然需要借助命令行的操作，我们将在后面进行介绍如何编译和运行Go代码。

## 4. 初识golang

安装好环境后，起码也要测试一下代码的运行效果，跑一次helloworld。Go语言版的helloworld如下：

```
package main

import (
    "fmt"
)

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

上述代码有三处关键字：

- package 定义包名
- import 导入包名
- func 函数定义关键字



main函数是程序执行的入口函数，这与其他语言是一致的，通过上述代码，我们可以发现Go语言引入了包的概念，而相关的API调用都可以通过“包名.函数名”的方式调用。作为main函数所在的文件，包名则必须是main，fmt.Println很好理解，就是fmt包为我们提供的一个格式化输出函数，该函数会自动的在输出后添加一个换行。下一个问题，这个代码怎么运行起来呢？

首先将代码保存为01-hello.go

```
localhost:day01 teacher$ go build 01-hello.go
localhost:day01 teacher$ ls
01-hello    01-hello.go
```

先用go build命令，可以对源码进行编译，这时在该目录下可以看到一个01-hello的可执行文件，运行该文件即可。

```
localhost:day01 teacher$ ./01-hello
hello world
```

除了这种方式，在前面介绍go语言工具集的时候，还看到过一个go run指令，当运行go run指令的时候，代码可以直接编译后运行，此时不会产生那个01-hello的可执行文件。

```
localhost:day01 teacher$ rm 01-hello
localhost:day01 teacher$ go run 01-hello.go
hello world
localhost:day01 teacher$ ls
01-hello.go
```

## 五、拓展点

说明：

1. 典型面试题、笔试题；
2. 新技术 or 经验分享；
3. 未来计划、行业趋势分享；

## 六、总结

说明：

1. 回顾本堂课所有知识点；
2. 提示 **注意点** 和 **重点**；
3. 提示学习方法；
4. 提示哪些需要记、哪些需要背、哪些代码需要敲；

## 七、大作业

---

说明：

1. 给出明确的作业要求，形成文字、图片或测试题等形式；
2. 给出明确的解答方式；
3. 频次低，可设计为：1次/周 或 1次/2周；

可从下面的检测题题库任意选择。

## 八、集中答疑

---

说明：完成本堂课所有知识点的讲解后，由学员集中提问，讲师逐一解答；

## 九、检测题

---

说明：

1. 针对本堂课，设计5-10个题，由学员课下完成（课下刷题）；
2. 出题范围可从检查点里挑选，可以是面试题或笔试题；
3. 题型要求是 单选、多选、判断、填空题中的一种或多种；
4. 频次高，原则上每次课都要有检测题；

## 十、下节课预告

---

说明：

1. Go语言的语法；
  2. 变量、常量
-