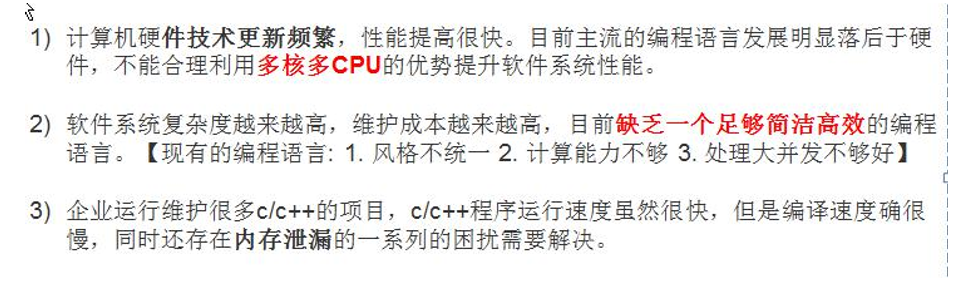
文档：

官网：https://golang.google.cn/

从事方向：区块链研发工程师、Go服务器/游戏软件工程师、golang分布式/云计算软件工程师。

# golang介绍

## 1.1Google创建 Golang的原因



## 1.2golang特点

Go既能达到静态编译语言的安全性和性能，又能达到动态语言开发维护的高效率。

Go=C+Python

1. 从c语言继承了很多理念，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。也保留了C语言一样的编译执行方式及弱化的指针

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，而且不能单独存在。

(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

(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gorute之间的相互通信。

(6)函数可以返回多个值

(7)新的创新：切片slice、延时执行defer

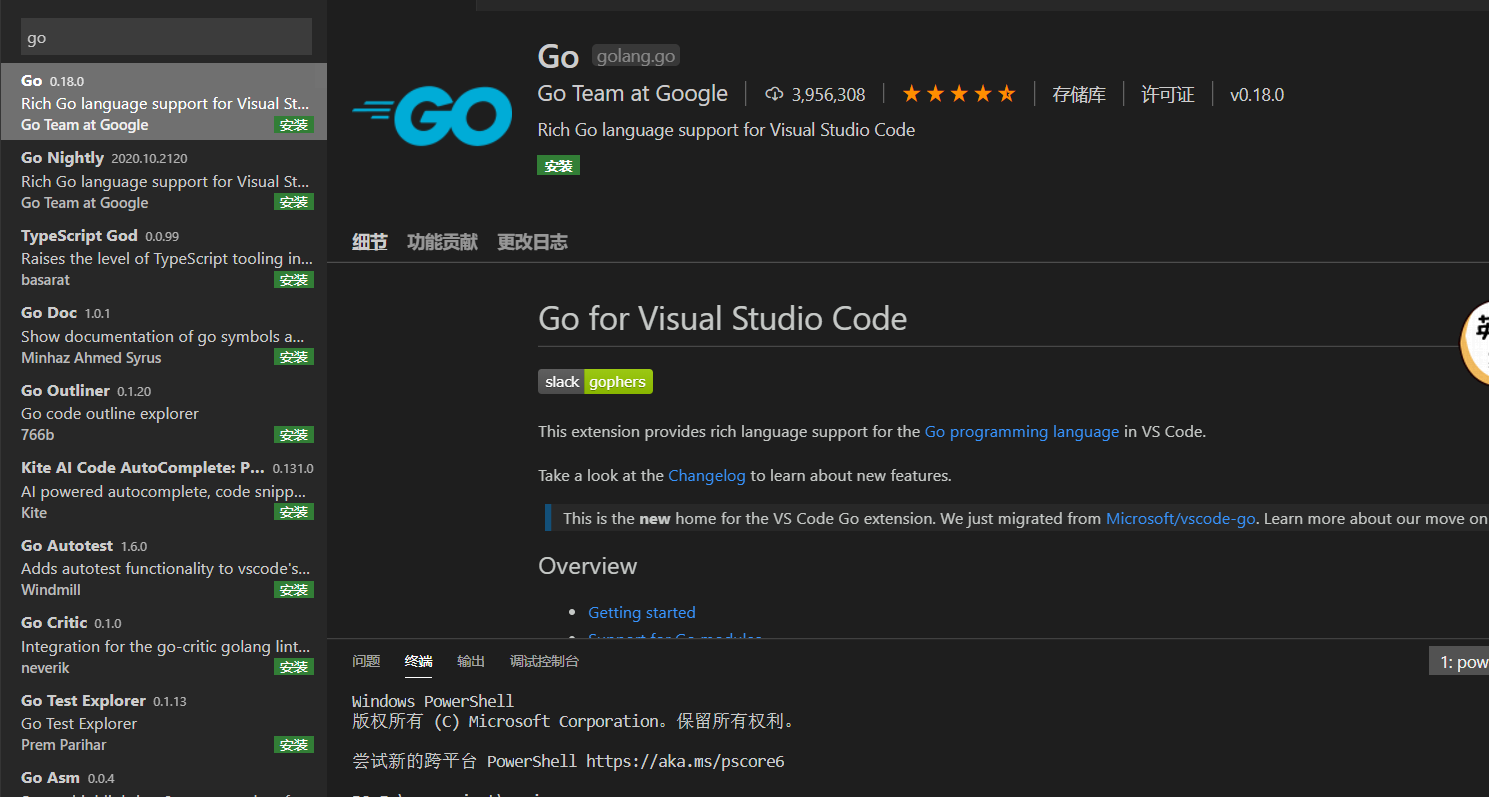
## 1.3安装

开发工具：vscode

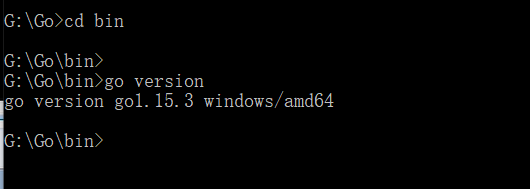
### Windows搭建go开发环境

官网下载：<https://golang.google.cn/doc/install>

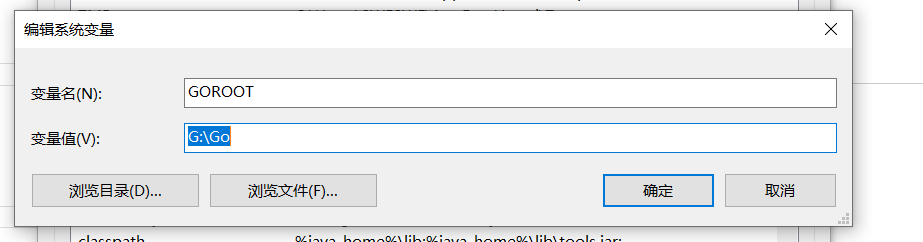
Go提示

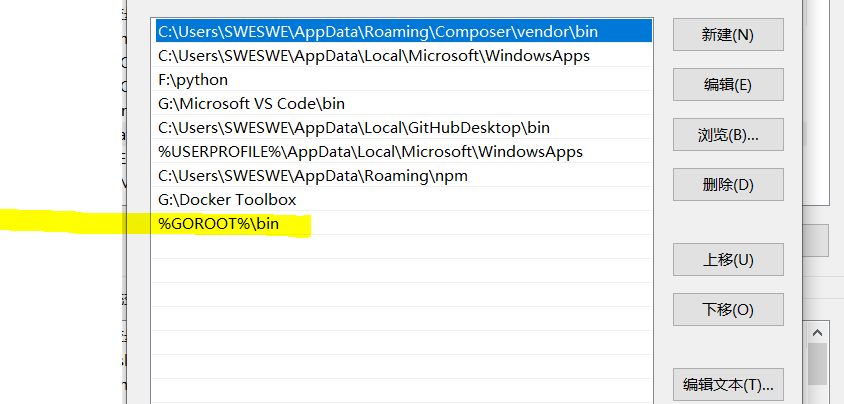


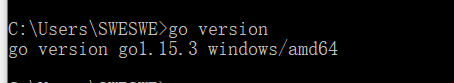
安装官网安装包，到安装目录bin下执行go version



配置环境变量：

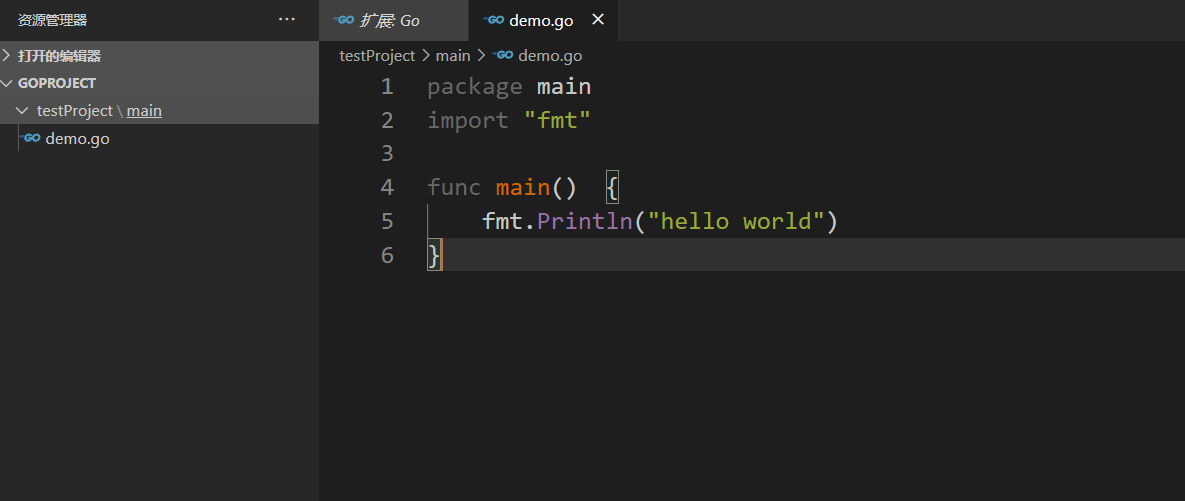




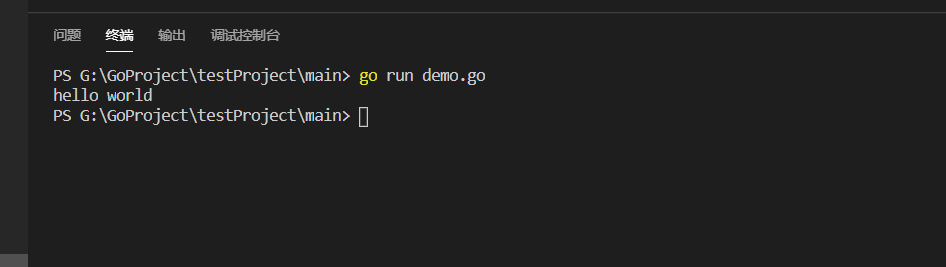


### Linux搭建go开发环境

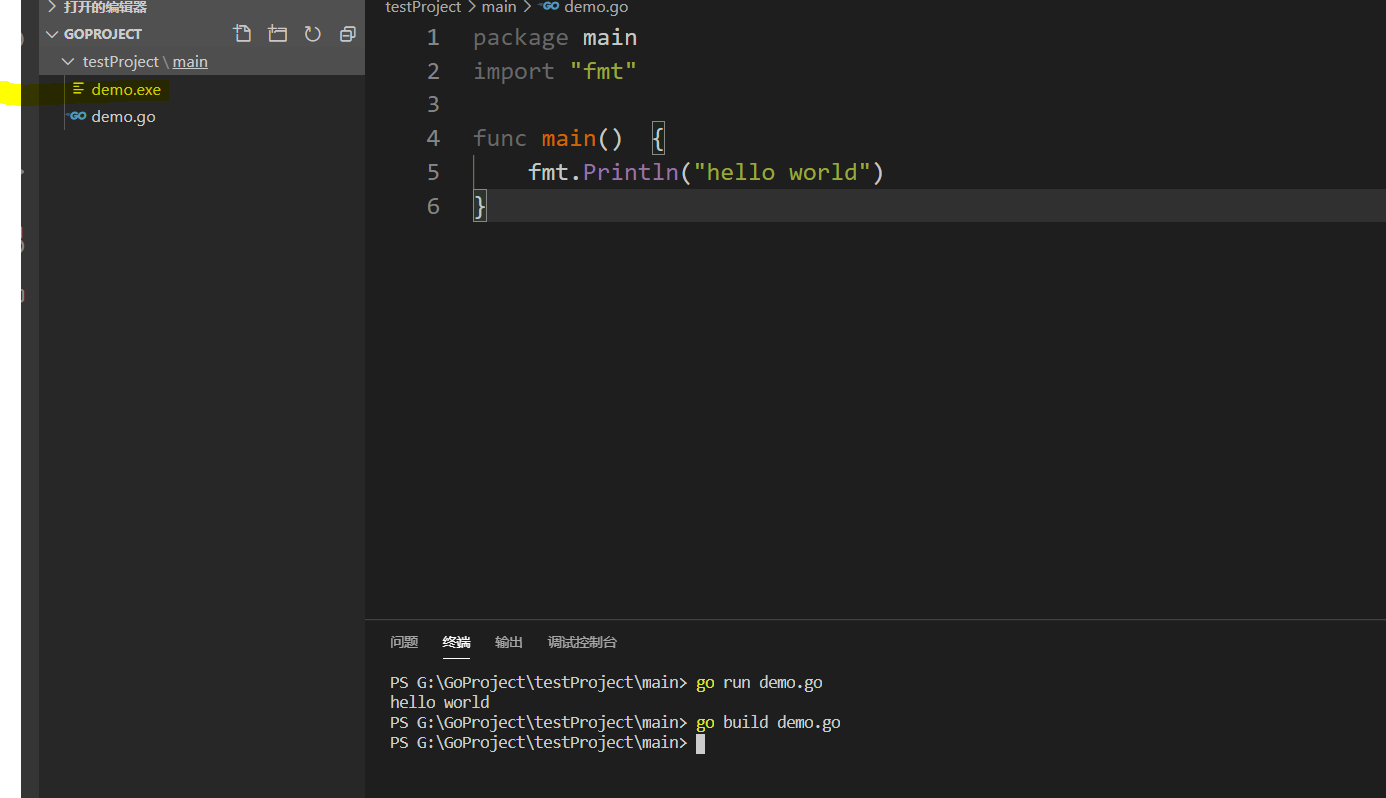
## 1.4快速开发入门



### 1.运行 go run demo.go

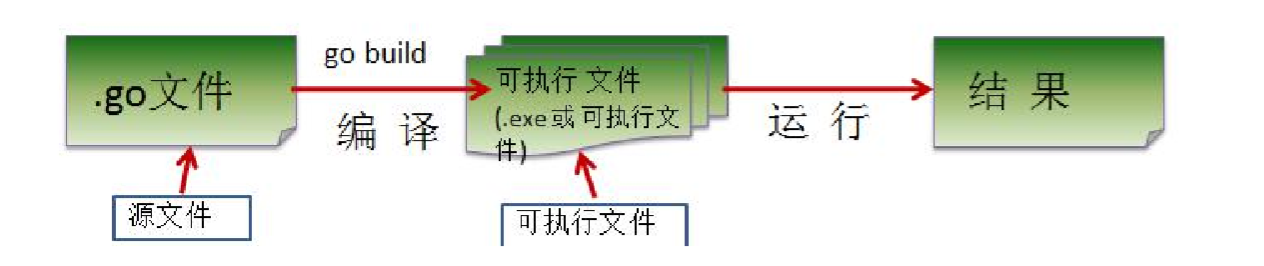


### 2.编译 go build生成.exe文件

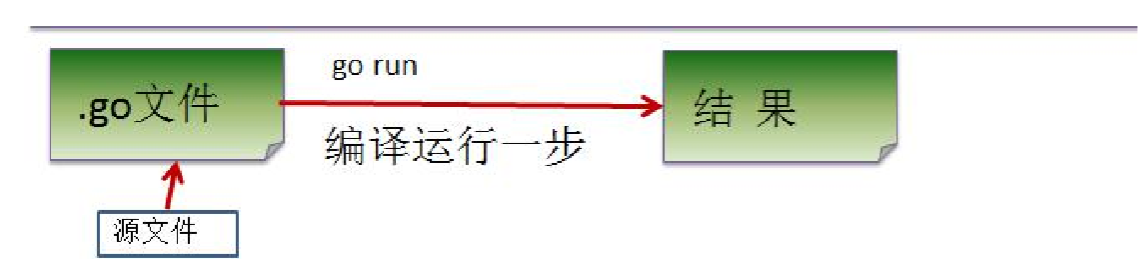


### 3.Go执行流程分析

对源码编译后在执行



直接go run 运行源码



**两种执行流程区别**

1. 对源码编译后在执行，没有go开发环境也能执行。直接go run运行源码在另一台机器上运行，该机器也需要go开发环境。
2. 在编译时，编译器会将程序运行依赖的库文件包含在可执行文件中，所以，可执行文件会变大。

### 编译和运行说明

1. 有了go源文件，通过编译器将其编译成机器可识别的二进制文件。
2. 编译之后会生成.exe文件

## 1.5注意事项

（1）go源文件以go为扩展名

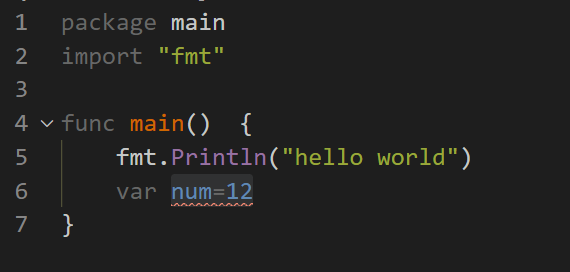
（2）go应用程序的执行入口为main()函数

（3）go严格区分大小写

（4）每条语句不需要加分号

（5）go编译器是一行行进行编译，因此一行只能写一条语句。多条语句写在同一行会报错

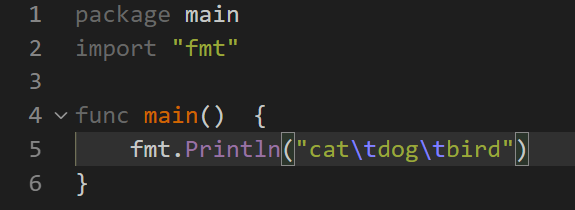
（6）go语言定义的变量或者import的包没有使用到，代码编译不能通过。

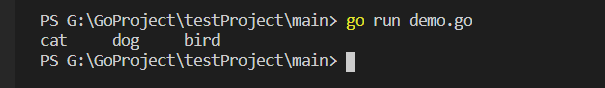


（7）大括号成对出现

## 1.6go语言的转义字符

（1）\t 表示一个制表符，通常使用它可以排版



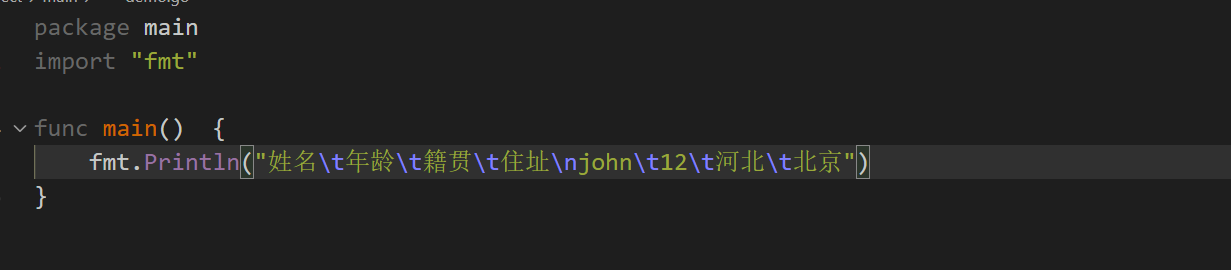


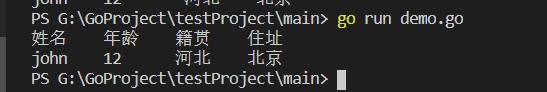
（2）\n 换行符

（3）\\ 一个\

（4）\” 一个”

（5）\r 一个回车





## 1.7开发常见问题

## 1.8注释

//

/\*\*/

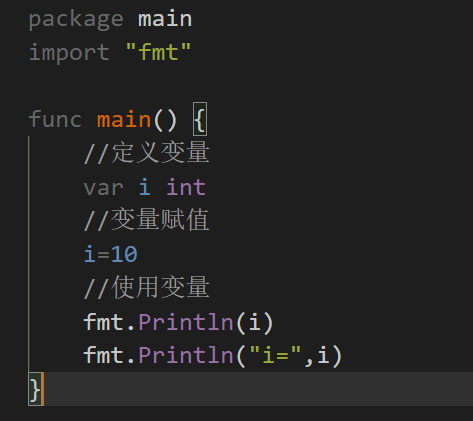
# 二．变量

## 1.使用变量步骤

（1）声明变量(定义变量)

（2）非变量赋值

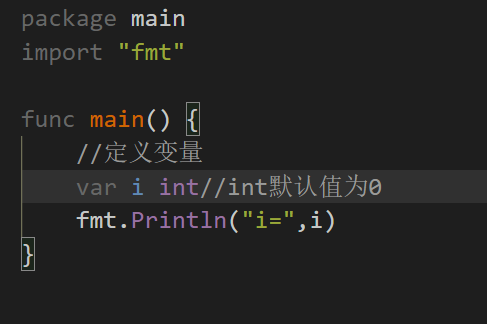
1. 使用变量



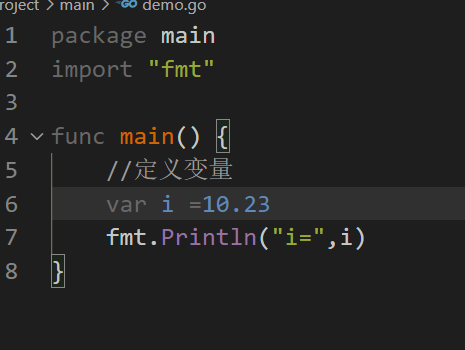
## 2.golang使用的三种方式

1. 指定变量类型，声明后不赋值，使用默认值

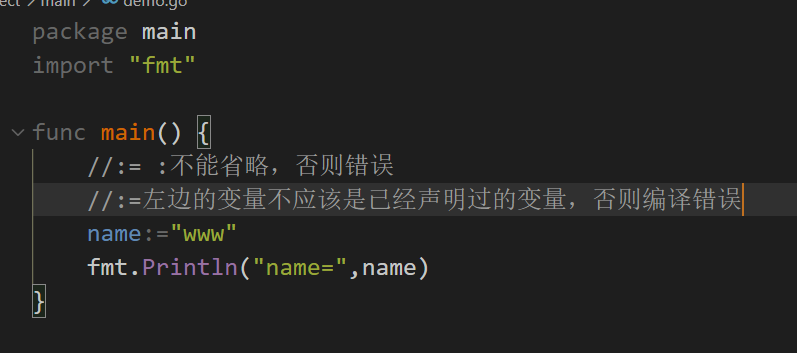
Int默认值为0 ，string默认值为空串，小数默认为0



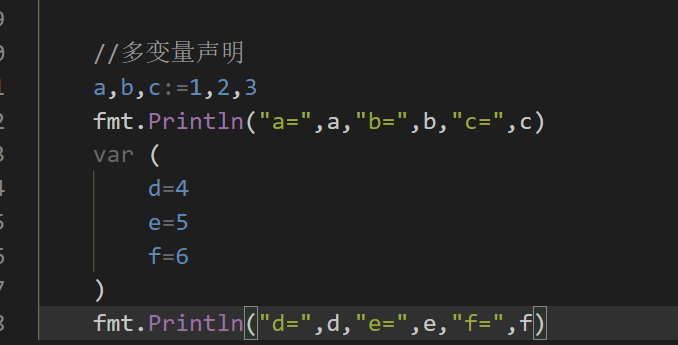
1. 根据值自行判断变量类型



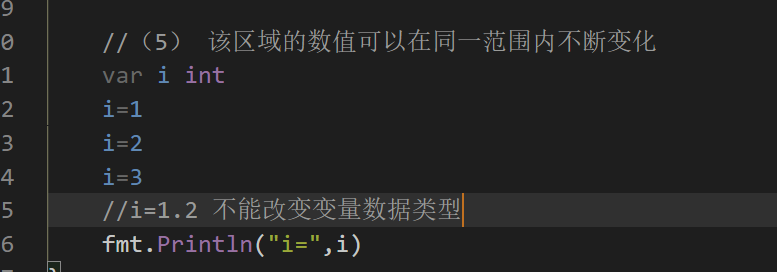
1. 省略var,



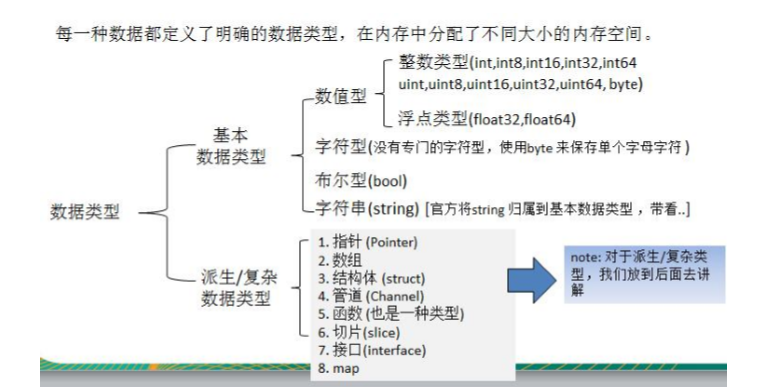
1. 多变量声明



1. 该区域的数值可以在同一范围内不断变化



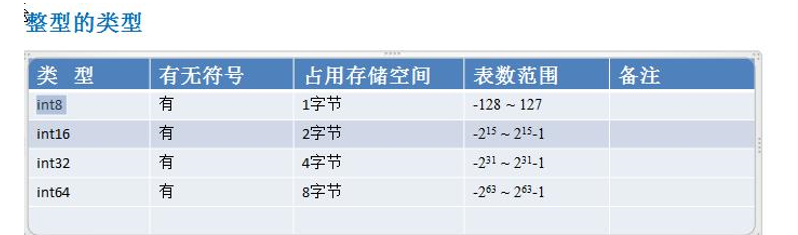
## 3．基本数据类型

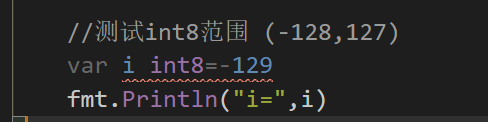


### 3.1整数型

整数类型分为有符号和无符号

#### Int有符号类型





#### Int 无符号类型



#### 其他类型



#### 查看变量的字节大小和数据类型

先引入unsafe包

Import “unsafe”

    //查看变量类型,所占字节数

    var i =23

    fmt.Printf("i的类型 %T，i占用字节数 %d",i,unsafe.Sizeof(i))

#### 保小不保大原则

Golang在使用整型变量时，遵守保小不保大原则。即：在保证程序正确和运行下，尽量使用占用空间小的数据类型。

### 3.2浮点型

#### 浮点型分类

Golang浮点型默认是float64



func main() {

    //单精度

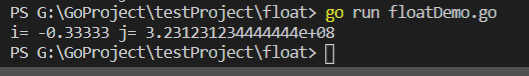
    var i float32=-0.33333

    //双精度相当于double

    var j float64=323123123.44444444444444

    fmt.Println("i=",i,"j=",j)

}



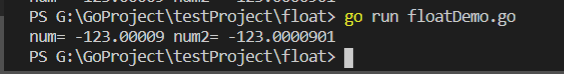
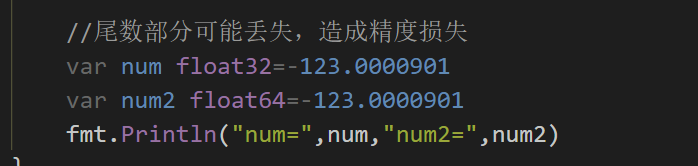
输出采用科学计数法：

323123123.44444444444444=3.2312312344444444444444乘10的8次方

1. 浮点数在机器中存放形式：浮点数=符号位+指数位+尾数位

浮点数都是有符号的

1. 尾数位部分可能丢失，造成精度缺失



说明float64的精度比float32的精度要准确

1. 浮点型的存储分为三部分：符号位+指数位+尾数位，在存储过程中精度会丢失

#### 浮点型常量两种表现形式

    //十进制形式

    num3:=5.212

    num4:=.1234//=>0.1234

    fmt.Println("num3=",num3,"num4=",num4)

    //科学计数法

    num5:=5.123e2//5.123\*10的2次方

    num6:=5.123E2//5.123\*10的2次方

    num7:=5.123-2//5.123\*10的2次方

### 3.3字符类型