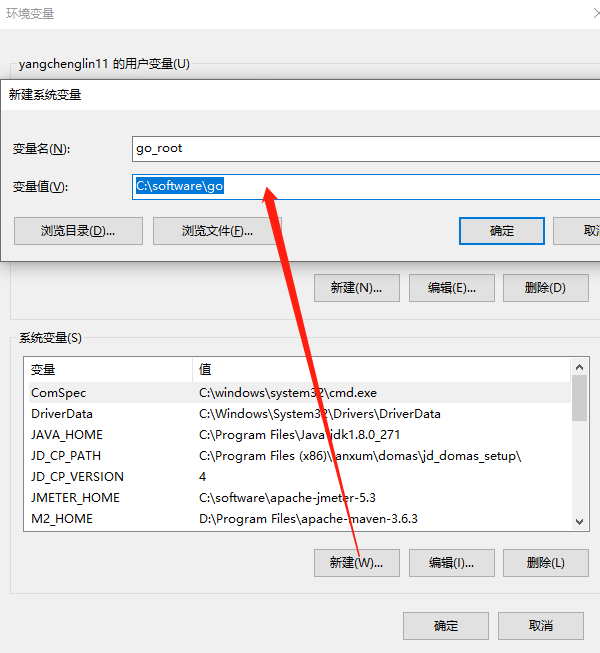
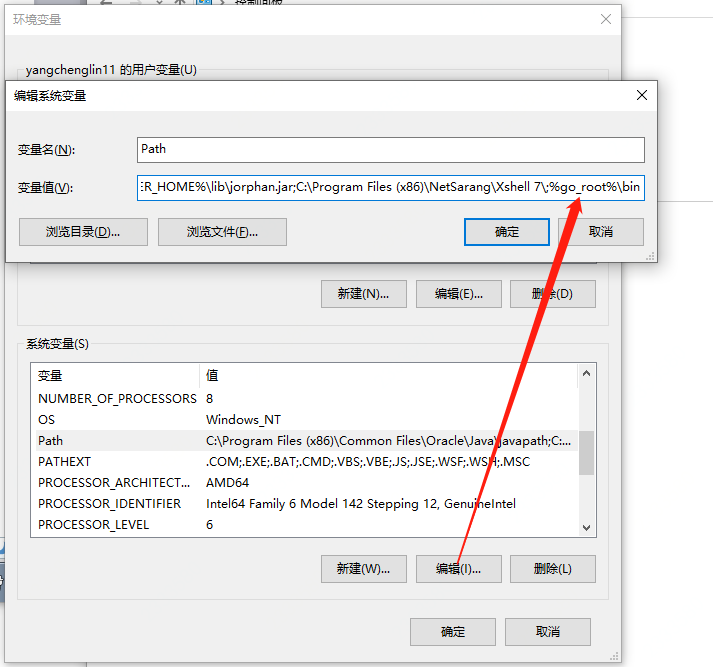
配置环境变量

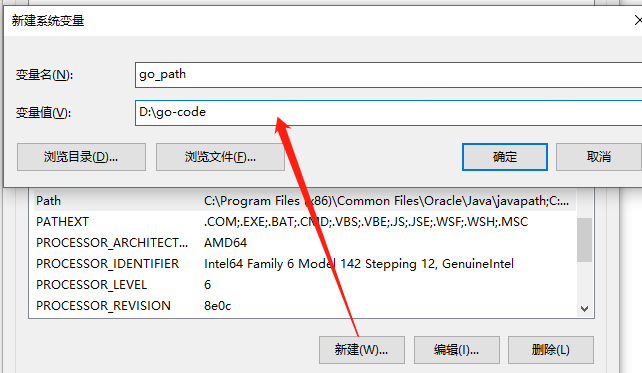
go\_root：指定sdk的安装路径；C:\software\go



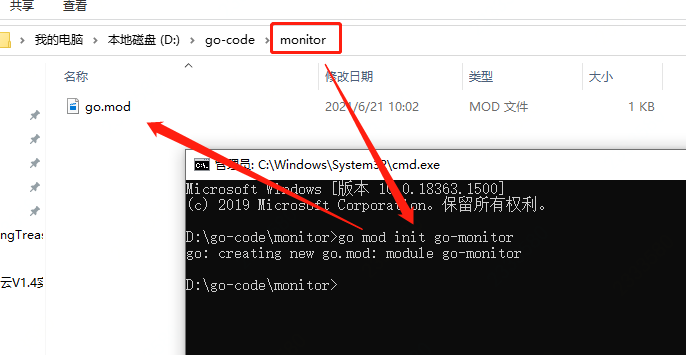
path：添加sdk的/bin目录；%go\_root%\bin，在path变量选择编辑，追加此go\_root值



gopath：go项目存放的路径，即工作目录；比Java多了这一步。

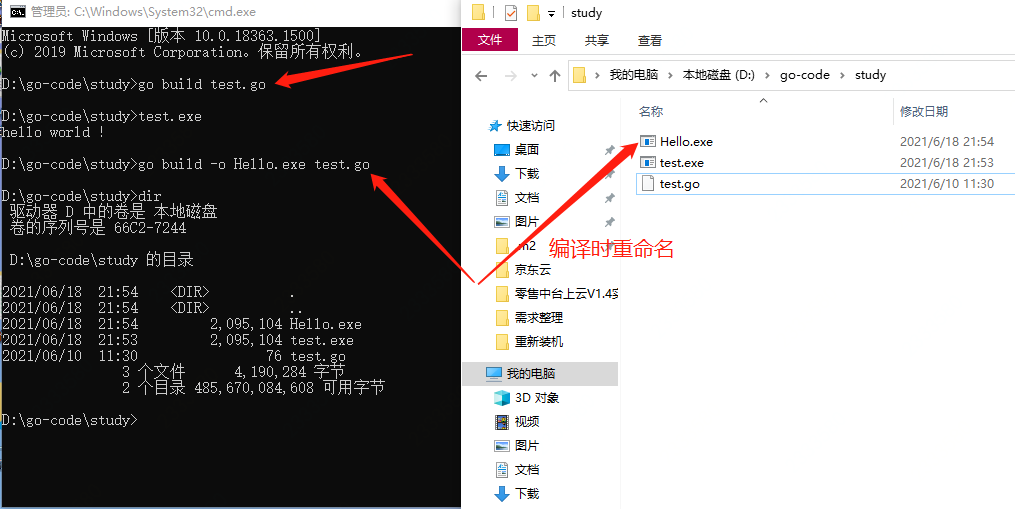
测试结果：

Go mod文件声明



go的编译说明

默认编译与编译重命名：

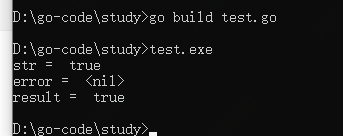
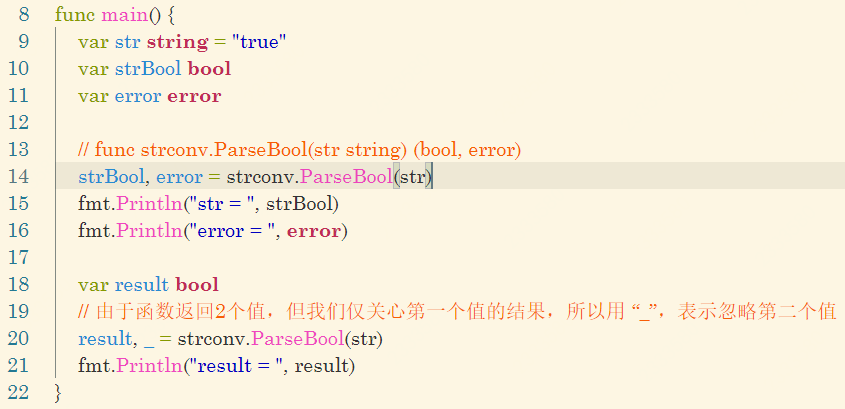


变量声明



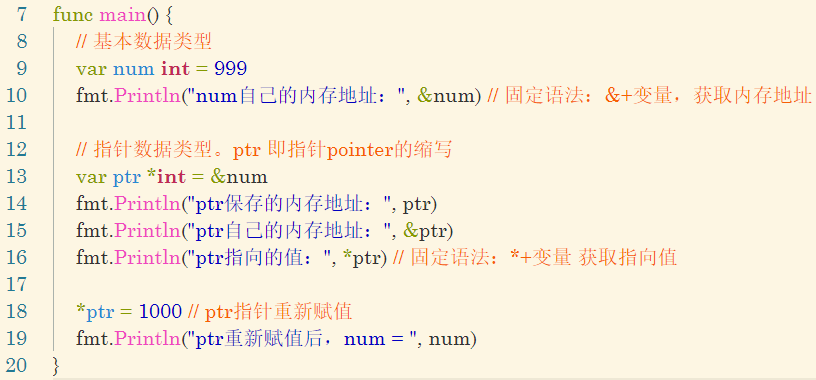
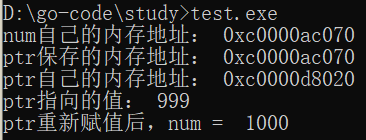
忽略函数返回值中的指定值

“\_”作为忽略返回值中的特定项使用，示例如下：



Go指针的详细说明

基本数据类型，变量存的就是值，也叫值类型。

指针类型与基本数据类型内存结构说明：

0x地址

堆内存

栈内存

0x002

0x001

num

999

0x004

0x003

0x002

ptr

栈保存声明的变量（比如基本数据类型变量、指针类型变量等），堆保存这个变量对应的值。

无论是栈内存还是堆内存，都有自己的0x\*\*\*\*\*内存地址。

值类型和引用类型

1. 值类型

基本数据类型：int/float/bool/string、数组和结构体struct。

1. 引用类型

指针、slice切片、map、管道channel、interface等。

golang %c，%d，%u分别代表输出的是什么类型的？

printf("%c",a)；输出单个字符。

printf("%d",a)；输出十进制整数。

printf("%f",a)；输出十进制浮点数.

printf("%o",a)；输出八进制数。

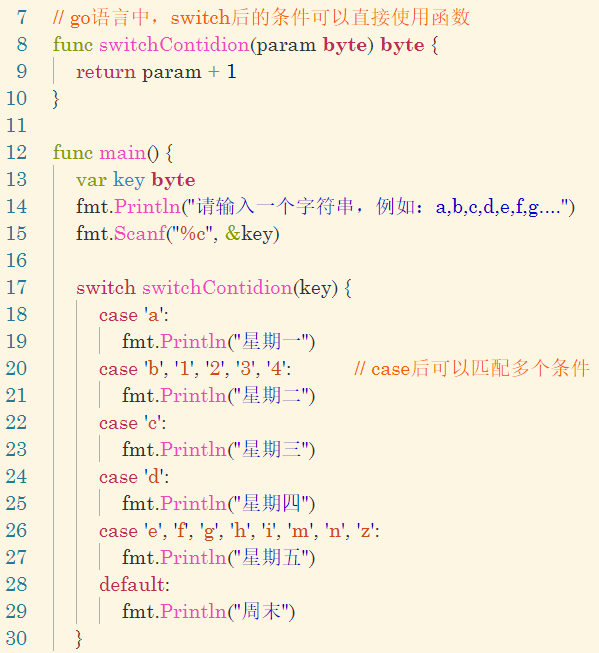
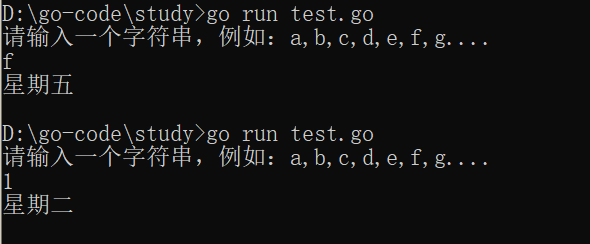
printf("%s",a)；输出字符串。

printf("%u",a)；输出无符号十进制数。

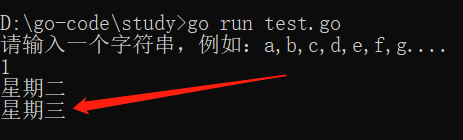
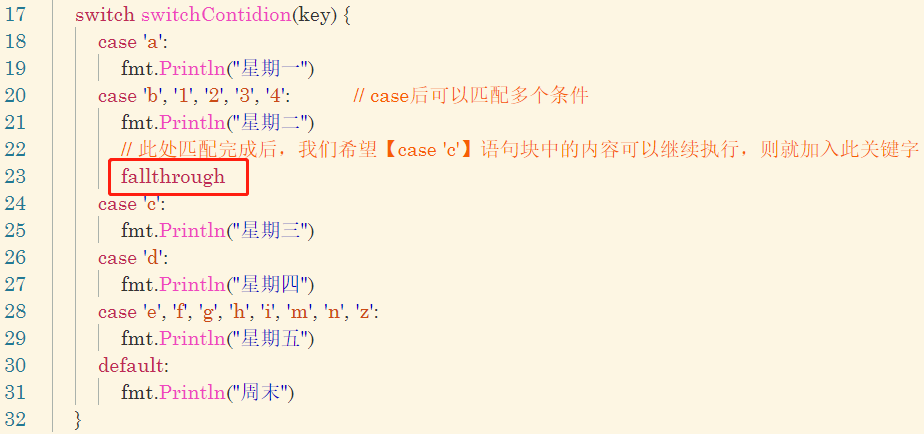
printf("%x",a)；输出十六进制数。

Case语句示例

1、基础使用

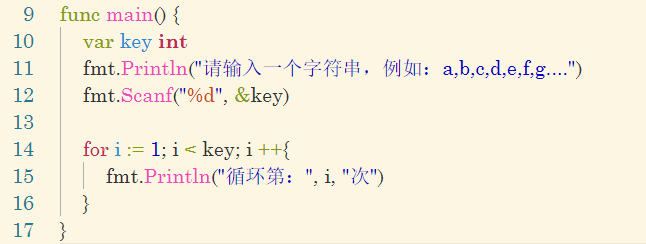
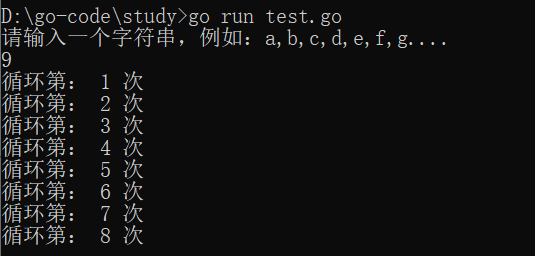
 

2、case穿透

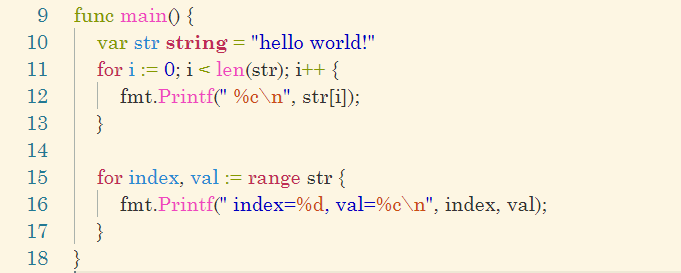
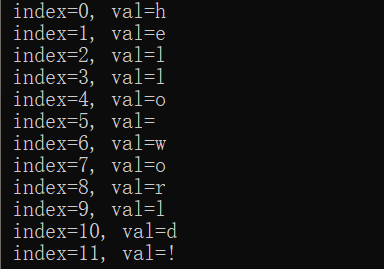


for循环

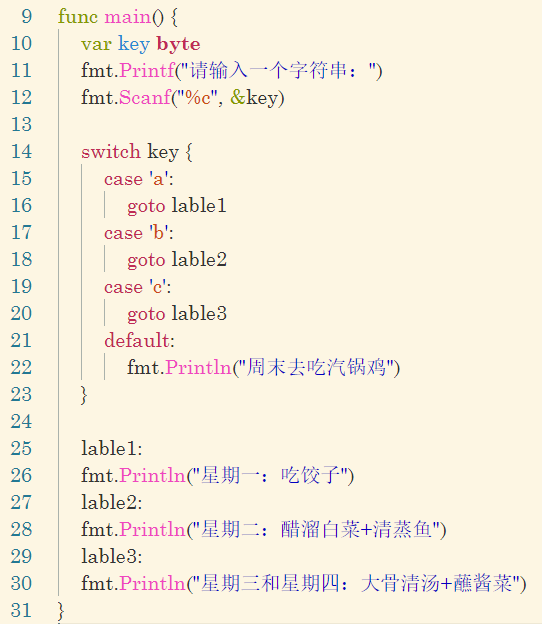
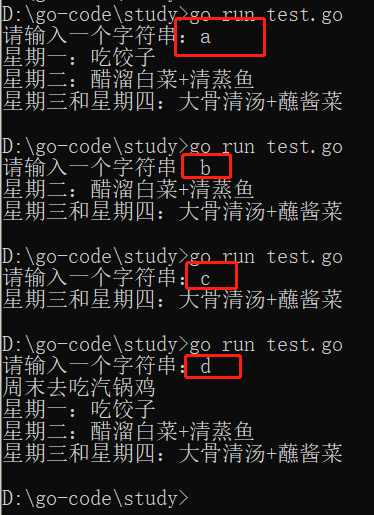
1、基础方式 – 字节方式遍历，中文会乱码

2、for-range方式 – 字符方式遍历，中文不乱码，一个中文占3个字节。

goto示例 – 尽量不使用

包的本质

1、基本概念

包的本质，实际上就是创建不同的文件夹，来存放程序文件。go的每一个文件爱你都是属于一个包的，也就是说go是以

包的形式来管理文件和项目目录结构的。包有三个作用：

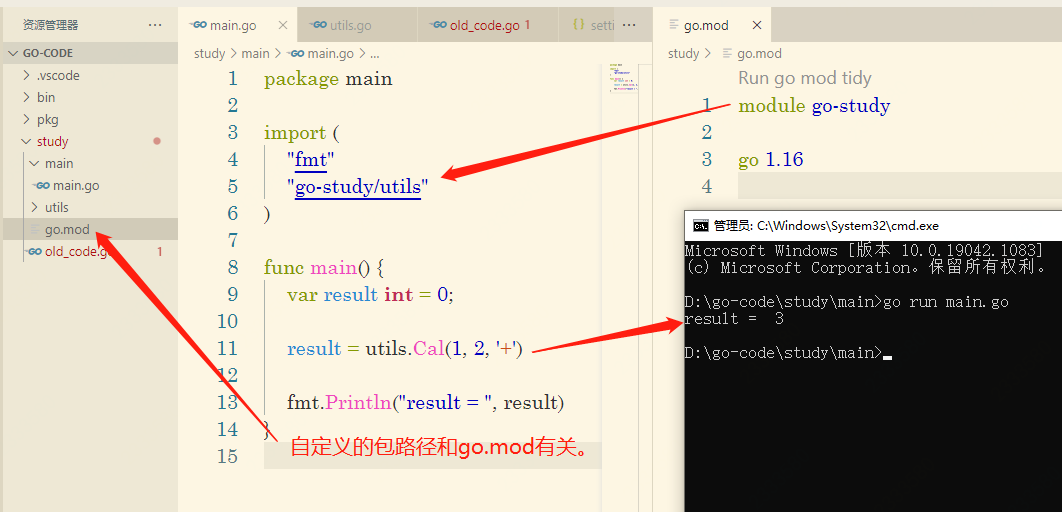
1. 区分同名函数、变量等标识符；
2. 当文件过多，能更好的管理项目；
3. 控制函数、变量的访问范围，即：作用域。

打包的基本语法：package util；引入包的语法：import "包的路径"。**通常**(但不是硬性规定)一个包名对应一个文件夹名，而且名称也是

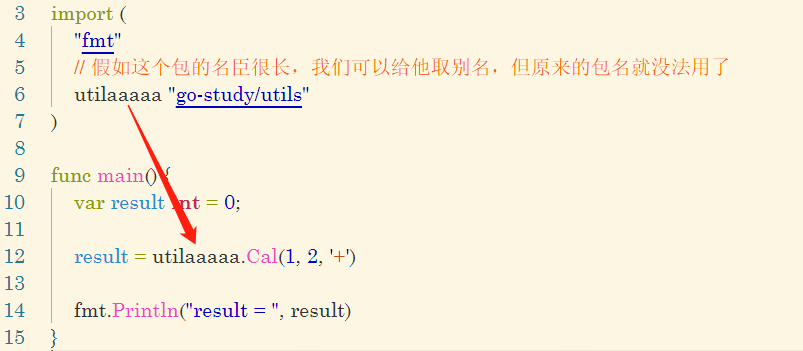
一样的小写字母，比如utils文件夹下的utils.go。方法名大写：该函数可导出。

示例如下：





2、包的别名



3、main包

如果你要将编译成一个可执行文件，那么就要将这个包声明为main，即：package main。在一个项目中main包只有一个。