Go语言学习(一)变量

原创 2016年03月23日 19:58:56 标签：go语言 540

package main

import "fmt"

func main() { //程序入口

InitVariable()

AssignmentVariable()

AnonymousVariable()

}

/\*变量的声明

func DeclareVariable(){

// Go语言的变量声明方式与C和C++语言有明显的不同。对于纯粹的变量声明，Go语言引入了

// 关键字 var ，而类型信息放在变量名之后，示例如下：

var v1 int

var v2 string

var v3 [10]int // 数组

var v4 []int // 数组切片

var v5 struct {

f int

}

var v6 \*int // 指针

var v7 map[string]int // map，key为string类型，value为int类型

var v8 func(a int) int

// 变量声明语句不需要使用分号作为结束符。与C语言相比，Go语言摒弃了语句必须以分号作

// 为语句结束标记的习惯。

// var 关键字的另一种用法是可以将若干个需要声明的变量放置在一起，免得程序员需要重复

// 写 var 关键字，如下所示：

var (

v11 int

v22 string

)

//注意:变量声明后必须要使用,否则编译报错: "xx declared and not used"

}

\*/

/\*变量的初始化语句\*/

func InitVariable(){

fmt.Println("-------变量初始化--------")

//方式1:声明变量和类型,通过()可以声明一组变量

var (

x int

y int

)

x = 3

y = 4

fmt.Printf("x=%v, y=%v\n", x,y)

//方式2:声明变量,变量类型,并初始化

var i int = 1

var j int = 2

fmt.Println(i+j)

//方式3:声明变量,并初始化,由编译器自动推导变量的类型

var k = 1

var m = 2

fmt.Println("k + m = ",k+m)

//方式4:声明变量并初始化,指定类型已不再是必需的,出现在 := 左侧的变量不应该是已经被声明过的，否则会导致编译错误

o := 1

p := 2

fmt.Println(o \* p)

}

/\*变量的赋值\*/

func AssignmentVariable(){

fmt.Println("-------变量的赋值--------")

//在Go语法中，变量初始化和变量赋值是两个不同的概念

//普通赋值

var v int

v = 100

fmt.Println("v=",v)

//Go语言支持多重赋值,减少了中间变量的使用

i := 1

j := 2

fmt.Printf("交换前i=%v,j=%v\n",i ,j)

i, j = j, i

fmt.Printf("交换后i=%v,j=%v\n",i ,j)

}

/\*匿名变量\*/

func AnonymousVariable(){

fmt.Println("-------匿名变量--------")

\_,\_, lastName := GetName() //Go语言支持函数多返回值的使用,如果不想接收某个返回值可以使用\_来表示

fmt.Println("lastName=",lastName)

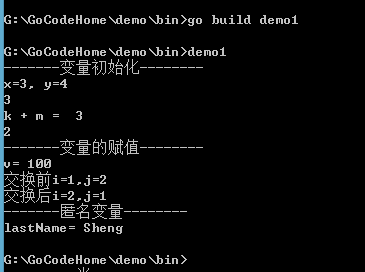
}

func GetName()(firstName,middleName,lastName string){

return "Chen","You","Sheng"

}

运行结果:



文章来源：<http://blog.csdn.net/mchenys/article/details/50966102>

版权声明：本文为博主原创文章，未经博主允许不得转载。