Go 基础map使用

map

map是一种**无序**的基于***key-value***的数据结构，go语言中的map是**引用类型,不初始化是个nil**，必须初始化才能使用（切片也是引用类型）。（声明时没有开辟内存）。

一、map的定义

go中map的定义语法如下：

**map[key type] value type**

keytype:表示键的类型。

Value type;表示键对应的值的类型。

（1）声明

var map变量名 map[key type]value type

map类型的变量默认是nil，需要使用make()函数来分配内存。

**var a map[string]int**

**fmt.Println(a == nil)**

（2）map的初始化

map变量名 = make(map [key type]vauletype, 容量大小)

（3）map的基本使用

map中的数据都是成对出现的，map的基本使用方法如下：

**func main() { scoreMap := make(map[string]int, 8)**

//在函数内部，可以使用更简略的 := 方式声明并初始化变量 **scoreMap["张三"] = 90 scoreMap["小明"] = 100 fmt.Println(scoreMap) fmt.Println(scoreMap["小明"]) fmt.Printf("type of a:%T\n", scoreMap) }**

map支持在声明时，填充元素。

**func main() { b :=map[int]bool{**

**1:true,**

**2:false,**

**} fmt.Println(b) // }**

声明一个map，没有初始化，不能直接操作。

（4）判断某个键是否存在

**func main() {**

**var scoremap = make(map[string]int,8)**

**scoremap["小米"]=100**

**scoremap["小胡"]=200**

**//判断小张 在不在scoremap中**

**value ,ok :=scoremap["小张"]**

**if ok {**

**fmt.Println("小张在scoremap中", value)**

**}else {**

**fmt.Println("小张不在scoremap中")**

**}**

func main() {

//map的遍历re

scoremap :=make(map[string]int)

scoremap["李"]=90

scoremap["张"]=100

scoremap["胡"]=60

for k, v :=range scoremap{

fmt.Println(k,v)

//map的元素的顺序与添加的键值对的顺序无关。

}

//**有一点很重要，当使用 for range 遍历 map 时，不保证每次执行程序获取的元素顺序相同**

for k :=range scoremap{

fmt.Println(k)

}

//只遍历value

for \_, v:=range scoremap{

fmt.Println(v)

}

（5）删除键值对

使用delete()内置函数从map中删除一组键值对，delete()函数的格式如下：

delete(map, key)

* map:表示要删除键值对的map
* key:表示要删除的键值对的键

（6）指定顺序遍历map

**var scoremap =make(map[string]int,100)**

**//添加50个键值对**

**for i:=0; i<50; i++{**

**key:=fmt.Sprintf("stu%02d",i)**

**value :=rand.Intn(100)**

**scoremap[key] =value**

**}**

**//for k,v :=range scoremap{**

**// fmt.Println(k,v)**

**//}**

**//按照key从小到大顺序去遍历scoremap**

**keys :=make([]string ,0 ,100)**

**for k :=range scoremap{**

**keys =append(keys,k)**

**//对key 排序**

**sort.Strings(keys)//keys是有序**

**for \_,key :=range keys{//按照排序后的key对scoremap输出**

**fmt.Println(key,scoremap[key])**

**}**

（7）**Map 的相等性**

**map 之间不能使用 == 操作符判断，== 只能用来检查 map 是否为 nil。**

**（8）元素为map类型的切片；**

**（9）值为切片的类型的map；**