



链滴

go 基础

作者: [468336329Zc](#)

原文链接: <https://ld246.com/article/1573911763039>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



go基础

变量声明

//var 变量名 变量类型

var a int

□

//自动推导，用于初始化，同一个变量只自动推导一次

a:=10

a=20

□

//数组自动推导 变量名:=[]类型{1,2,4}

array:=[]int{1,2,4,5,6}

□

定义函数

/*func 函数名(形参 类型)(返回值变量名 类型)

都是形参，返回值形参back 与真正返回的res无关，不是要一样

*/

func Test(a int)(back int){

res:=a

return res

}

□

□

□

输入输出

//输入

/*1.格式化输入

常见格式化控制符 "%d" 整型 "%s" 字符串 "%c"字符 "f%"浮点型

*/

var a int

fmt.Scanf("%d",&a)

□

/*2.非格式化输入

*/

var str string

fmt.Scan(&str)

□

□

//输出

/*1.格式化输出

*/

```
a:=1
fmt.Printf("我是整型数字%d",a)
[]
[]
[]
```

切片

//1.第一方式：引用创建的数组，来作为一个切片

```
var array []int = []int{1,2,3,4,5,6}
```

//引用存在了的数组

```
slice:=array[:6]
```

```
[]
```

```
[]
```

//2.第二种方式--使用make函数创建 make([]type,len,[cap]) len:长度 cap: 可变参数，指容量

```
slice:=make([]int,5)
```

```
[]
```

//3.第三种方式

```
var slice []int 这是声明一个切片。[]里面没有数字，表示这是切片而不是数组。
```

/*第一种方式是直接引用数组，这个数组是事先存在的，程序员可见

第二/三种方式是通过make来创建切片，make也会创建一个数组，是由切片在底层维护，程序员不见

*/

切片作为参数传参

//切片作为参数传参

```
[]
```

```
var arr []int = []int{1,2,3,4,5}
```

```
[]
```

//引用创建的数组arr

```
slice:=arr[0:5]
```

```
[]
```

```
func main(){
```

```
sum:=sliceSum(slice)
```

```
fmt.Println(sum)//15
```

```
}
```

```
[]
```

```
//求和函数
func sliceSum(a[] int)(sum int){
res:=0
/*range遍历 两个参数 i是下标 data是值
for i,data range a{

} */

//这里求和，下标用不着，用匿名参数_代替
for _,data :=range a{
res=data+res
}
return res
}
```