

Go 语言变量

变量来源于数学，是计算机语言中能储存计算结果或能表示值抽象概念。不同的变量可能保存的数据类型会不一样。

变量可以通过变量名访问。

声明变量

Go 语言变量名由字母、数字、下划线组成，其中首个字符不能为数字。

Go语言中的变量需要声明后才能使用，同一作用域内不支持重复声明。并且Go语言的变量声明后**必须使用**。

声明变量的一般形式是使用 var 关键字：

```
var identifier type
```

可以一次声明多个变量：

```
var identifier1, identifier2 type
```

var：声明变量关键字

identifier：变量名称

type：变量类型

例如：

```
package main

func main() {
    #单个变量声明
    var name string    //字符串类型
    var age int         //整型
    var istrue bool     //布尔型
    var price float32   //浮点型
    #多个变量声明
    var name,email string //同一类型多个变量声明
    #不同类型多个变量声明
    var (
        name string
        age int
        istrue bool
        price float32
    )
}
```

变量初始化

Go语言在声明变量的时候，会自动对变量对应的内存区域进行初始化操作。每个变量会被初始化成其类型的默认值，例如：**整型和浮点型变量的默认值为0**。**字符串变量的默认值为空字符串**。**布尔型变量默认为false**。**切片、函数、指针变量的默认为nil**。

变量初始化的标准格式如下：

```
var 变量名 类型 = 表达式
```

例如：

```
var name string = "包子"
var age int = 18
var isTrue bool = true
#或者一次初始化多个变量
var name string
var age int
var isTrue bool
name,age,isTrue = "包子",18,true
```

类型推导：

有时候我们会将变量的类型省略，这个时候编译器会根据等号右边的值来推导变量的类型完成初始化。

```
var name = "包子"    //字符串类型
var age = 18         //整型
var isTrue = true    //布尔型
#或者一次初始化多个变量
var name,age,isTrue = "包子",18,true
```

短变量声明

在**函数内部**，可以使用更简略的 `:=` 方式声明并初始化变量。**注意只能函数内部，外部不可以使用。**

```
name := "包子"    //字符串类型
age := 18         //整型
isTrue := true    //布尔型
#或者一次初始化多个变量
name,age,isTrue := "包子",18,true
```

匿名变量

在使用多重赋值时，如果想要忽略某个值，可以使用 **匿名变量**（`anonymous variable`）。匿名变量用一个**下划线**表示，就是不使用这个变量，忽略这个变量，下划线表示变量名称。例如：

```
func getInfo() (string, int) {  
    return "包子", 18  
}  
  
func main() {  
    name, _ := getInfo()  
    _, age := getInfo()  
}
```

匿名变量不占用命名空间，不会分配内存，所以匿名变量之间不存在重复声明。

需要注意的地方

- 函数外的每个语句都必须以关键字开始（var、const、func等）
- :=不能使用在函数外。
- _多用于占位，表示忽略值。