

Go 语言常量

常量是一个简单值的标识符，在程序运行时，不会被修改的量。

常量中的数据类型只可以是布尔型、数字型（整数型、浮点型和复数）和字符串型。

常量的定义格式：

```
const identifier [type] = value
```

你可以省略类型说明符 [type]，因为编译器可以根据变量的值来推断其类型。

- 显式类型定义：`const b string = "abc"`
- 隐式类型定义：`const b = "abc"`

多个相同类型的声明可以简写为：

```
const c_name1, c_name2 = value1, value2
```

以下实例演示了常量的应用：

```
package main

import "fmt"

func main() {
    const LENGTH int = 10
    const WIDTH int = 5
    var area int
    const a, b, c = 1, false, "str" //多重赋值

    area = LENGTH * WIDTH
    fmt.Printf("面积为：%d", area)
    println()
    println(a, b, c)
}
```

以上实例运行结果为：

```
面积为：50
1 false str
```

常量还可以用作枚举：

```
const (  
    Unknown = 0  
    Female = 1  
    Male = 2  
)
```

数字 0、1 和 2 分别代表未知性别、女性和男性。

常量可以用len(), cap(), unsafe.Sizeof()函数计算表达式的值。常量表达式中，函数必须是内置函数，否则编译不过：

```
package main  
  
import "unsafe"  
const (  
    a = "abc"  
    b = len(a)  
    c = unsafe.Sizeof(a)  
)  
  
func main(){  
    println(a, b, c)  
}
```

以上实例运行结果为：

```
abc 3 16
```

iota

iota，特殊常量，可以认为是一个可以被编译器修改的常量。

iota 在 const关键字出现时将被重置为 0(const 内部的第一行之前)，const 中每新增一行常量声明将使 iota 计数一次(iota 可理解为 const 语句块中的行索引)。

iota 可以被用作枚举值：

```
const (  
    a = iota  
    b = iota  
    c = iota  
)
```

第一个 iota 等于 0，每当 iota 在新的一行被使用时，它的值都会自动加 1；所以 a=0, b=1, c=2 可以简写为如下形式：

```
const (  
    a = iota
```

```
b
c
)
```

iota 用法

```
package main

import "fmt"

func main() {
    const (
        a = iota    //0
        b           //1
        c           //2
        d = "ha"    //独立值, iota += 1
        e           //"ha"  iota += 1
        f = 100     //iota +=1
        g           //100  iota +=1
        h = iota    //7, 恢复计数
        i           //8
    )
    fmt.Println(a,b,c,d,e,f,g,h,i)
}
```

以上实例运行结果为：

```
0 1 2 ha ha 100 100 7 8
```

再看个有趣的的 iota 实例：

```
package main

import "fmt"
const (
    i=1<<iota
    j=3<<iota
    k
    l
)

func main() {
    fmt.Println("i=",i)
    fmt.Println("j=",j)
    fmt.Println("k=",k)
}
```

```
fmt.Println("l=",l)
}
```

以上实例运行结果为：

```
i= 1
j= 6
k= 12
l= 24
```

iota 表示从 0 开始自动加 1，所以 `i=1<<0`，`j=3<<1`（<< 表示左移的意思），即：`i=1`，`j=6`，这没问题，关键在 `k` 和 `l`，从输出结果看 `k=3<<2`，`l=3<<3`。

简单表述：

`i=1`：左移 0 位，不变仍为 1；

- `j=3`：左移 1 位，变为二进制 110，即 6；
- `k=3`：左移 2 位，变为二进制 1100，即 12；
- `l=3`：左移 3 位，变为二进制 11000，即 24。

← Go 语言变量

Go 语言运算符 →



2 篇笔记

写笔记



```
a = "hello"
unsafe.Sizeof(a)
```

输出结果为：16

字符串类型在 go 里是个结构，包含指向底层数组的指针和长度，这两部分每部分都是 8 个字节，所以字符串类型大小为 16 个字节。

yangzhuoyongcuo 2年前 (2017-09-07)



在定义常量组时，如果不提供初始值，则表示将使用上行的表达式。

```
package main

import "fmt"

const (
    a = 1
    b
    c
    d
```

```
)  
  
func main() {  
    fmt.Println(a)  
    // b、c、d没有初始化，使用上一行(即a)的值  
    fmt.Println(b)    // 输出1  
    fmt.Println(c)    // 输出1  
    fmt.Println(d)    // 输出1  
}
```

rain0120 2个月前 (01-30)