课程安排，请关注微信公众平台或者官方微博

编程语言： Golang 与 html5

编程工具： Goland 和 HBuilder

预计平均一周左右更新一或二节课程

授人以鱼，不如授人以渔业。

大家好，我是彬哥，
欢迎大家来到 Golang 语言社区云课堂课程的学习。Slice
社区论坛网址：**www.golang.ltd**
技术交流群 ： 221273219
微信公众号 ：**Golang 语言社区**
微信服务号 ：**Golang 技术社区**

第一季 Go 语言基础、进阶、提高课程

第四节 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。

在每一个 const 关键字出现时，被重置为 0，然后再下一个 const 出现之前，每出现一次 iota，其所代表的数字会自动增加 1。

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。