



## bool类型

布尔型的值只可以是常量 true 或者 false。一个简单的例子：var b bool = true

## 数值型

### 1. 整数型

可以简单讲解一下二进制和位数的关系，以及int和uint的关系

- int8 有符号 8 位整型 (-128 到 127) 长度：8bit
- int16 有符号 16 位整型 (-32768 到 32767)
- int32 有符号 32 位整型 (-2147483648 到 2147483647)
- int64 有符号 64 位整型 (-9223372036854775808 到 9223372036854775807)
- uint8 无符号 8 位整型 (0 到 255) 8位都用于表示数值：
- uint16 无符号 16 位整型 (0 到 65535)
- uint32 无符号 32 位整型 (0 到 4294967295)
- uint64 无符号 64 位整型 (0 到 18446744073709551615)

### 3. 浮点型

- float32 32位浮点型数

- float64 64位浮点型数

## 其他

- byte 等于 uint8
- rune 等于 int32
- uint 32 或 64 位

## 字符

Golang中没有专门的字符类型，如果要存储单个字符(字母)，一般使用byte来保存。

字符串就是一串固定长度的字符连接起来的字符序列。Go的字符串是由单个字节连接起来的。也就是说对于传统的字符串是由字符组成的，而Go的字符串不同，它是由字节组成的。

```
`package main

import (
    "fmt"
)

func main() {

    var a byte
    a = 'a'
    //输出ascii对应码值。。 这里说明一下什么是ascii码
    fmt.Println(a)
    fmt.Printf("a=%c", a)
}
`
```

代码块

预览

复制

复制成

功!

```
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
```

# 优质IT资源微信x923713

字符常量只能使用单引号括起来，例如：var a byte = 'a' var a int = 'a'

```
`package main

import (
    "fmt"
)

func main() {

    var a byte
    a = "a"
    //输出ascii对应码值。。 这里说明一下什么是ascii码
    fmt.Println(a)
    fmt.Printf("a=%c", a)
}
```

代码块

预览

复制

复制成

功!

```
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
```

字符本质是一个数字， 可以进行加减乘除

```
`package main

import (
    "fmt"
    "reflect"
)

func main() {
```

```
a := 'a'
```

//这里注意一下 1. a+1可以和数字计算 2.a+1的类型是32 3. int类型可以直接变成字符

```
fmt.Println(reflect.TypeOf(a+1))
```

```
fmt.Printf("a+1=%c", a+1)
```

```
}
```

```
`
```

代码块

预览

复制

复制成

功!

```
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
```

## 字符串

字符串就是一串固定长度的字符连接起来的字符序列。Go 的字符串是由单个字节连接起来的。Go 语言的字符串的字节使用 UTF-8 编码标识 Unicode 文本。

- 划线
- 写笔记