

GoLang 中基本数据类型之间的转换

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

1、关于 golang 中的数据类型转换.....	1
2、数值类型之间的相互转换.....	1
3、其他类型转换成 String 类型.....	2
4、String 类型转换成数值类型.....	4
5、数值类型没法和 bool 类型进行转换.....	5

1、关于 golang 中的数据类型转换

Go 语言中只有强制类型转换，没有隐式类型转换。.

2、数值类型之间的相互转换

数值类型包括：整形和浮点型

```
package main
import "fmt"
func main() {
    var a int8 = 20
    var b int16 = 40
    var c = int16(a) + b //要转换成相同类型才能运行
    fmt.Printf("值: %v--类型%T", c, c) //值: 60--类型 int16
}
```

```
package main
import "fmt"
func main() {
    var a float32 = 3.2
    var b int16 = 6
    var c = a + float32(b)
```

```
fmt.Printf("值: %v--类型%T", c, c) //值: 9.2--类型 float32
}
```

转换的时候建议从低位转换成高位，高位转换成低位的时候如果转换不成功就会溢出，和我们想要的结果不一样。

比如：

```
package main
func main() {
    var a int16 = 129
    var b = int8(a)      // 范围 -128 到 127
    println("b=", b) //b= -127    //错误
}
```

比如计算直角三角形的斜边长时使用 `math` 包的 `Sqrt()` 函数，该函数接收的是 `float64` 类型的参数，而变量 `a` 和 `b` 都是 `int` 类型的，这个时候就需要将 `a` 和 `b` 强制类型转换为 `float64` 类型。

```
var a, b = 3, 4
var c int
// math.Sqrt() 接收的参数是 float64 类型，需要强制转换
c = int(math.Sqrt(float64(a*a + b*b)))
fmt.Println(c)
```

3、其他类型转换成 String 类型

1、sprintf 把其他类型转换成 string 类型

注意： `sprintf` 使用中需要注意转换的格式 `int` 为 `%d` `float` 为 `%f` `bool` 为 `%t` `byte` 为 `%c`

```
package main
import "fmt"
func main() {
    var i int = 20
    var f float64 = 12.456
    var t bool = true
    var b byte = 'a'
    var strs string
```

```

    strs = fmt.Sprintf("%d", i)
    fmt.Printf("str type %T ,strs=%v \n", strs, strs)

    strs = fmt.Sprintf("%f", f)
    fmt.Printf("str type %T ,strs=%v \n", strs, strs)

    strs = fmt.Sprintf("%t", t)
    fmt.Printf("str type %T ,strs=%v \n", strs, strs)

    strs = fmt.Sprintf("%c", b)
    fmt.Printf("str type %T ,strs=%v \n", strs, strs)
}

```

输出:

```

d:\golang\src\demo01>go run main.go
str type string ,strs=20
str type string ,strs=12.456000
str type string ,strs=true
str type string ,strs=a

```

2、使用 strconv 包里面的几种转换方法进行转换

```

package main

import (
    "fmt"
    "strconv"
)

func main() {
    //1、 int 转换成 string
    var num1 int = 20
    s1 := strconv.Itoa(num1)
    fmt.Printf("str type %T ,strs=%v \n", s1, s1)

    // 2、 float 转 string
    var num2 float64 = 20.113123
    /*
        参数 1: 要转换的值
    */
}

```

参数 2: 格式化类型

'f' (-ddd.ddd)、
'b' (-ddd±ddd, 指数为二进制)、
'e' (-d.ddd±dd, 十进制指数)、
'E' (-d.dddE±dd, 十进制指数)、
'g' (指数很大时用'e'格式, 否则'f'格式)、
'G' (指数很大时用'E'格式, 否则'f'格式)。

参数 3: 保留的小数点 -1 (不对小数点格式化)

参数 4: 格式化的类型

```
*/
s2 := strconv.FormatFloat(num2, 'f', 2, 64)
fmt.Printf("str type %T ,strs=%v \n", s2, s2)

// 3、bool 转 string
s3 := strconv.FormatBool(true)

fmt.Printf("str type %T ,strs=%v \n", s3, s3)

//4、int64 转 string
var num3 int64 = 20
/*
第二个参数为 进制
*/
s4 := strconv.FormatInt(num3, 10)
fmt.Printf("类型 %T ,strs=%v \n", s4, s4)
}
```

4、String 类型转换成数值类型

1、string 类型转换成 int 类型

```
var s = "1234"
i64, _ := strconv.ParseInt(s, 10, 64)
fmt.Printf("值: %v 类型: %T", i64, i64)
```

2、string 类型转换成 float 类型

```
str := "3.1415926535"
v1, _ := strconv.ParseFloat(str, 32)
v2, _ := strconv.ParseFloat(str, 64)

fmt.Printf("值: %v 类型: %T\n", v1, v1)
fmt.Printf("值: %v 类型: %T", v2, v2)
```

3、string 类型转换成 bool 类型（意义不大）

```
b, _ := strconv.ParseBool("true") // string 转 bool
fmt.Printf("值: %v 类型: %T", b, b)
```

4、string 转字符

```
s := "hello 张三"
for _, r := range s { //rune
    fmt.Printf("%v(%c) ", r, r)
}
fmt.Println()
```

5、数值类型没法和 bool 类型进行转换

注意：在 go 语言中数值类型没法直接转换成 bool 类型 bool 类型也没法直接转换成数值类型