

字符串处理函数

目录：

1. strings包常用函数
2. strconv包常用函数

一、概述

(一)、strings包简介：

1、详见《go语言标准库》：<https://studygolang.com/pkgdoc>

腾讯云开发者手册：<https://cloud.tencent.com/developer/section/1144338>

2、学习strings库的目的：

- 熟悉编程语言字符串处理方法。
- 大多数语言的字符串处理库提供的函数都大同小异，且越高级的语言提供的函数越多，比如c提供的字符串处理函数go基本都有，但是go提供的c未必有。
- 掌握基本的字符串处理函数后，更丰富的字符串处理函数都是通过封装基本的处理函数实现。所以熟悉go strings库后基本就能借此封装符合自己需求的，应用于特定场景的字符串处理函数了。

(二)、strconv包简介：

1、strconv包实现了字符串与其它基本数据类型之间的类型转换。

- 详见《go语言标准库》：<https://studygolang.com/pkgdoc>
- 腾讯云开发者手册：

<https://cloud.tencent.com/developer/section/1144302>

(三)、string的基本用法（可以理解成特殊的slice）

- 1、字符串与字符
- 2、字符串遍历
- 3、切片是引用类型，字符串是值类型
- 4、示例代码：

```
package main

import (
    "fmt"
    "unicode/utf8"
)

func main() {
    s := "我爱Go语言"
    fmt.Println(s)
    fmt.Println("字节长度：", len(s))
    fmt.Println("-----")

    //获取字符串引用
    for i, ch := range s {
        fmt.Printf("%d:%c \n", i, ch)
    }
    fmt.Println("-----")

    //遍历所有字节
    for i, ch := range []byte(s) {
        fmt.Printf("%d:%X \n", i, ch)
    }
    fmt.Println("字节长度:", len(s))
    fmt.Println("-----")

    //获取所有字符
    //rune相当于go的char（字符型）
    count := 0
    for i, ch := range []rune(s) {
        fmt.Printf("%d:%c \n", i, ch)
```

```
        count++
    }
    fmt.Println("字符长度:", count)
    fmt.Println("字符长度: ", utf8.RuneCountInString(s))
    fmt.Println("-----")
}
```

二、strings包中的字符串处理函数

(一)、检索字符串

1、func Contains(s, substr string) bool

判断字符串s是否包含substr字符串

2、func ContainsAny(s, chars string) bool

判断字符串s是否包含chars字符串中的任一字符

3、func ContainsRune(s string, r rune) bool

判断字符串s是否包含unicode码值r

4、func Count(s, sep string) int

返回字符串s包含字符串sep的个数

5、func HasPrefix(s, prefix string) bool

判断字符串s是否有前缀字符串prefix

6、func HasSuffix(s, suffix string) bool

判断字符串s是否有后缀字符串suffix

7、func Index(s, sep string) int

返回字符串s中字符串sep首次出现的位置

8、func IndexAny(s, chars string) int

返回字符串chars中的任一unicode码值r在s中首次出现的位置

9、func IndexByte(s string, c byte) int

返回字符串s中字符c首次出现位置

10、func IndexFunc(s string, f func(rune) bool) int

返回字符串s中满足函数f(r)==true字符首次出现的位置

11、func IndexRune(s string, r rune) int

返回unicode码值r在字符串中首次出现的位置

12、func LastIndex(s, sep string) int

返回字符串s中字符串sep最后一次出现的位置

13、func LastIndexAny(s, chars string) int

返回字符串s中任意一个unicode码值r最后一次出现的位置

14、func LastIndexByte(s string, c byte) int

返回字符串s中字符c最后一次出现的位置

15、func LastIndexFunc(s string, f func(rune) bool) int

返回字符串s中满足函数f(r)==true字符最后一次出现的位置

(二)、分割字符串：

1、func Fields(s string) []string

将字符串s以空白字符分割，返回一个切片

2、func FieldsFunc(s string, f func(rune) bool) []string

将字符串s以满足f(r)==true的字符分割，返回一个切片

3、func Split(s, sep string) []string

将字符串s以sep作为分隔符进行分割，分割后字符串最后去掉sep

4、func SplitAfter(s, sep string) []string

将字符串s以sep作为分隔符进行分割，分割后字符最后附上sep

5、func SplitAfterN(s, sep string, n int) []string

将字符串s以sep作为分隔符进行分割，分割后字符最后附上sep，n决定返回的切片数

6、func SplitN(s, sep string, n int) []string

将字符串s以sep作为分隔符进行分割，分割后字符最后去掉sep，n决定返回的切片数

(三)、大小写转换：

1、func Title(s string) string

将字符串s每个单词首字母大写返回

2、func ToLower(s string) string

将字符串s转换成小写返回

3、func ToLowerSpecial(_case unicode.SpecialCase, s string) string

将字符串s中所有字符按_case指定的映射转换成小写返回

4、func ToTitle(s string) string

将字符串s转换成大写返回

5、func ToTitleSpecial(_case unicode.SpecialCase, s string) string

将字符串s中所有字符按_case指定的映射转换成大写返回

6、func ToUpper(s string) string

将字符串s转换成大写返回

7、func ToUpperSpecial(_case unicode.SpecialCase, s string) string

将字符串s中所有字符按_case指定的映射转换成大写返回

(四)、trim修剪函数：

1、func Trim(s string, cutset string) string

将字符串s中首尾包含cutset中的任一字符去掉返回

2、func TrimFunc(s string, f func(rune) bool) string

将字符串s首尾满足函数f(r)==true的字符去掉返回

3、func TrimLeft(s string, cutset string) string

将字符串s左边包含cutset中的任一字符去掉返回

4、func TrimLeftFunc(s string, f func(rune) bool) string

将字符串s左边满足函数f(r)==true的字符去掉返回

5、func TrimPrefix(s, prefix string) string

将字符串s中前缀字符串prefix去掉返回

6、func TrimRight(s string, cutset string) string

将字符串s右边包含cutset中的任一字符去掉返回

7、func TrimRightFunc(s string, f func(rune) bool) string

将字符串s右边满足函数f(r)==true的字符去掉返回

8、func TrimSpace(s string) string

将字符串s首尾空白去掉返回

9、func TrimSuffix(s, suffix string) string

将字符串s中后缀字符串prefix去掉返回

(五)、比较字符串

1、func Compare(a, b string) int

按字典顺序比较a和b字符串大小

2、func EqualFold(s, t string) bool

判断s和t两个utf8字符串是否相等，忽略大小写

3、func Repeat(s string, count int) string

将字符串s重复count次返回

4、func Replace(s, old, new string, n int) string

替换字符串s中old字符为new字符并返回，n<0是替换所有old字符串

5、func Join(a []string, sep string) string

将a中的所有字符串连接成一个字符串，使用字符串sep作为分隔符

三、strconv包中常用函数

(一)、Parse类函数（将字符串转其它类型）

1、func Atoi(s string) (int, error)

Atoi 返回 ParseInt(s, 10, 0) 转换为 int 类型的结果。【alphabet: 字母】

2、func ParseInt(s string, base int, bitSize int) (i int64, err error)

ParseInt 解释给定基础（2到36）中的字符串 s 并返回相应的值 i。如果 base == 0，则基数由字符串的前缀隐含：base 16代表“0x”，base 8代表“0”，否则以10为底数。

3、func ParseUint(s string, base int, bitSize int) (uint64, error)

ParseUint 就像 ParseInt，但是对于无符号数字。

4、func ParseFloat(s string, bitSize int) (float64, error)

ParseFloat 将字符串 s 转换为浮点数，精度由 bitSize: 32指定，float32为64; float64为64。当 bitSize = 32时，结果仍然具有 float64 类型，但可以在不更改其值的情况下将其转换为 float32。

5、func ParseBool(str string) (bool, error)

ParseBool 返回字符串表示的布尔值。

- 它接受1, t, T, TRUE, true, True,
- 0, f, F, FALSE, false, False。
- 任何其他值都会返回错误。

(二)、Format类函数 (将其它类型格式化成字符串)

1、func Itoa(i int) string

Itoa 是 FormatInt(int64(i), 10) 的缩写。

2、func FormatInt(i int64, base int) string

FormatInt 返回给定基数中的i的字符串表示, 对于 $2 \leq \text{base} \leq 36$.结果对于数字值 ≥ 10 使用小写字母 'a' 到 'z' 。

3、func FormatUint(i uint64, base int) string

FormatUint 返回给定基数中的 i 的字符串表示, 对于 $2 \leq \text{base} \leq 36$.结果对于数字值 ≥ 10 使用小写字母 'a' 到 'z' 。

4、func FormatFloat(f float64, fmt byte, prec, bitSize int) string

FormatFloat 根据格式 fmt 和 precision prec 将浮点数f转换为字符串。它将结果进行四舍五入, 假设原始数据是从 bitSize 位的浮点值获得的 (float32为32, float64为64) 。

格式 fmt 是 'b', 'e', 'E', 'f', 'g'或 'G'。

5、func FormatBool(b bool) string

FormatBool 根据 b 的值返回“true”或“false”

