字符串

字符串的相关操作

修改字符串的类型

类型转换

Go语言中的字符串以原生数据类型出现，使用字符串就像使用其他原生数据类型（int、bool、float32、float64 等）一样。 Go 语言里的字符串的内部实现使用 UTF-8 编码。 字符串的值为 双引号(") 中的内容，可以在Go语言的源码中直接添加非ASCII码字符，例如：

```
s1 := "hello"
s2 := "你好"
```

字符串转义符

Go 语言的字符串常见转义符包含回车、换行、单双引号、制表符等，如下表所示。

转义符	含义
\r	回车符（返回行首）
\n	换行符（直接跳到下一行的同列位置）
\t	制表符
\'	单引号
\"	双引号
\\	反斜杠

举个例子，我们要打印一个Windows平台下的一个文件路径：

Go语言中的字符串只能使用""包括

Go语言中''包裹的是字符

打印网络路径时，反斜线可能被认为是转移符号，在这些反斜线后面再加一个反斜线表示继续转义。

示例

```go
package main

import "fmt"

func main() {
    fmt.Println("'D:\\Go\\src\\code.oldboyedu.com\\studygo\\day01")
}
```

定义多行字符串：

定义多行字符串 Go

```go
func main() {
    //fmt.Println("'D:\\Go\\src\\code.oldboyedu.com\\studygo\\day01")
    s1 := `
     大家好
    我叫
        大帅逼
`

    fmt.Println(s1)
}
```

字符串的相关操作

字符串的常用操作

方法	介绍
len(str)	求长度
+或fmt.Sprintf	拼接字符串
strings.Split	分割
strings.contains	判断是否包含
strings.HasPrefix,strings.HasSuffix	前缀/后缀判断
strings.Index(),strings.LastIndex()	子串出现的位置
strings.Join(a[]string, sep string)	join操作

字符串操作示例 Go

```go
func main() {
    name := "理想"
    world := "大帅逼"
    s3 := "'D:\\Go\\src\\code.oldboyedu.com\\studygo\\day01"
    ss := name + world
    fmt.Println(ss)
    ss1 := fmt.Sprintf("%s%s", name, world)
    fmt.Println(ss1)
    //分割  遇到"\"以空格分开
    ret := strings.Split(s3, "\\")
    fmt.Println(ret)
    //包含
    fmt.Println(strings.Contains(ss, "理想"))
    //前缀
    fmt.Println(strings.HasPrefix(ss, "理想"))
    //后缀
    fmt.Println(strings.HasSuffix(ss, "理想"))
    //字串
    fmt.Println(strings.Index(ss, "大帅逼"))
    fmt.Println(strings.LastIndex(ss, "大帅逼"))
    //拼接
    fmt.Println(strings.Join(ret, "+"))
}
```

运行结果：

```
理想大帅逼
理想大帅逼
['D: Go src code.oldboyedu.com studygo day01]
true
true
false
6
6
'D:+Go+src+code.oldboyedu.com+studygo+day01
```

注意：英文字符时byte类型，中文及其他文字时rune类型

修改字符串的类型

```go
s2 := "白萝卜"
s3 := []rune(s2) //把字符串强制转化成rune切片
s3[0] = '红'
fmt.Println(string(s3)) //再将切片转化为字符串
```

类型转换

```go
n := 10
var f float64
f = float64(n)//将n由整形转为浮点型赋值给f
fmt.Println(f)
```