四）rune（相当于char）

- 可用来处理国际化字符
- 使用range遍历pos、rune对

```
s := "我爱你china"
fmt.Println()
  for i, ch := range []rune(s) {
    fmt.Printf("(%d %c)", i, ch)
}
// (0 我)(1 爱)(2 你)(3 C)(4 h)(5 i)(6 n)(7 a)
```

- 使用utf8.RuneCountInString获得字符数量

```
fmt.Println(utf8.RuneCountInString(s)) // 8
```

- 使用len获得字节长度

```
fmt.Println(len(s)) // 14
```

- 使用[byte获得字节

```
fmt.Println([]byte(s)) // [230 136 145 231 136 177 228 189 160 67 104 105 110 97]
```

五）其他字符串操作

- Fileds、Split、Join 字符串分割合并

```
// Fields 以连续的空白字符为分隔符，将 s 切分成多个子串，结果中不包含空白字符本身
// 空白字符有：\t, \n, \v, \f, \r, ' ', U+0085 (NEL), U+00A0 (NBSP)
// 如果 s 中只包含空白字符，则返回一个空列表
s1 := "Hello, 世界! Hello!"
s2 := strings.Fields(s1)
fmt.Printf("%q\n", s2) // ["Hello," "世界!" "Hello!"]
```

```
s3 := "我,爱,你,中,国"
fmt.Println(strings.Split(s3, ",")) // [我 爱 你 中 国]
```

```
s4 := "我爱你中国"
s5 := "我爱你香港"
fmt.Println(strings.Join([]string{s4, s5}, ",")) // 我爱你中国,我爱你香港
```

- Contains、Index 查找字串

```
s6 := "abcdefg"
fmt.Println(strings.Contains(s6, "de")) // true
fmt.Println(strings.Contains(s6, "de")) // false
```

```
s7 := "abcedf"
fmt.Println(strings.Index(s7, "df")) // 4
fmt.Println(strings.Index(s7, "de")) // -1
```

- ToLower、ToUpper转换大小写

```
s8 := "i love you china"
s9 := strings.ToUpper(s8)
fmt.Println(s9) // I LOVE YOU CHINA
fmt.Println(strings.ToLower(s9)) // i love you china
```

- Trim、TrimLeft、TrimRight

```
// func Trim(s string cutset string){}
// func TrimLeft(s string cutset string){}
// func TrimRight(s string cutset string){}

s10 := " i love you china, hello world "
fmt.Println(strings.Trim(s10, " ")) // "i love you china, hello world"
fmt.Println(strings.TrimLeft(s10, " ")) // "i love you china, hello world "
fmt.Println(strings.TrimRight(s10, " ")) // " i love you china, hello
world"
```