String length in Go

Author : aklyachkin

Categories : Go Language

Tagged as : cookbook, go, golang, i18n, length, localization, runestring

Date : 10.07.2014

```
package main    import (      "fmt"      "bufio"      "os"      "unicode/utf8"
  )    func main() {      scanner := bufio.NewScanner(os.Stdin)      for scann
er.Scan() {          s := scanner.Text()          fmt.Println("Length: ",
         utf8.RuneCountInString(s),             " Bytes: ",             len(
s))      }  }
```

Built-in function len() shows number of bytes in string. For one-byte characters, such as ASCII, number of bytes in string and its length are the same value. But for multi-byte characters (Chinese, Russian and so on) these are two different values. Localized strings use more memory, than there are characters in them. That's why to find a number of characters in a string the function RuneCountInString from package "unicode/utf8" should be used.

Another way to obtain numer of characters in a string is to cast the string to []rune and use len() built-in:

```
fmt.Println(len([]rune(s)))
```