链滴

golang slice 排序

作者：cuua

原文链接：https://ld246.com/article/1609506782716

来源网站：链滴

如下示例为，在一个Person切片中，按年龄大小进行排序

```
package main

import (
    "fmt"
    "sort"
)

/*slice 排序示例*/
type Person struct {
    Age int
}

type PersonSlice []Person

func (s PersonSlice) Len() int           { return len(s) }
func (s PersonSlice) Swap(i, j int)      { s[i], s[j] = s[j], s[i] }
func (s PersonSlice) Less(i, j int) bool { return s[i].Age < s[j].Age }

func main() {
    persons := PersonSlice{
        Person{
            Age: 1,
        },
        Person{
            Age: 5,
        },
        Person{
            Age: 2,
        },
    }
    sort.Sort(persons)
    fmt.Printf("after sort:%+v", persons)
}
```

输出 after sort:[{Age:1} {Age:2} {Age:5}]

说明:被排序的结构体需要实现如下接口

```
type Interface interface {
    // Len is the number of elements in the collection.
    Len() int
    // Less reports whether the element with
    // index i should sort before the element with index j.
    Less(i, j int) bool
    // Swap swaps the elements with indexes i and j.
    Swap(i, j int)
}
```

sort包中有sort.Slice函数专门用于slice的排序，使用极简单方便

```
package main

import (
```

```
    "fmt"
    "sort"
)

/*slice 简单排序示例*/
func main() {
    //定义一个年龄列表
    ageList := []int{1, 3, 7, 7, 8, 2, 5}

    //排序，实现比较方法即可
    sort.Slice(ageList, func(i, j int) bool {
        return ageList[i] < ageList[j]
    })
    fmt.Printf("after sort:%v", ageList)
}
```

输出 after sort:[1 2 3 5 7 7 8]