

二，内建容器

一) 数组

- 定义数组

```
1 var arr1 [5]int
2 arr2 := [3]int{1, 2, 4}
3 arr3 := [...]int{2, 4, 6, 7, 8, 9}
4 var grid [4][5]int // 4行5列, 默认数组元素都是0, [[0 0 0 0 0] [0 0 0 0 0] [0 0 0 0 0] [0 0 0 0 0]]
```

- 遍历数组

```
1 // 第一种
2 for i, v := range arr3 {
3     fmt.Println(i, v)
4 }
5 // 第二种
6 for i := 0; i < len(grid); i++ {
7     fmt.Println(grid[i])
8 }
```

- 数组是值类型
 - [10]int和[20]int是不同的类型
 - func foo(arr [10]int)会拷贝数组
 - 在Go中我们一般不使用数组，而会使用切片

```
1 func printArr(arr [5]int) {
2     arr[0] = 100
3     for i, v := range arr {
4         fmt.Println(i, v)
5     }
6 }
7 func main() {
8     arr2 := [...]int{3, 4, 5, 6, 7}
9     printArr(arr2)
10    //0 100
11    //1 4
12    //2 5
13    //3 6
14    //4 7
15    fmt.Println(arr2)
16    //[3 4 5 6 7]
17 }
```

使用指针

```
1 func printArr(arr *[5]int) {
2     arr[0] = 100
3     for i, v := range arr {
4         fmt.Println(i, v)
5     }
6 }
7 func main() {
8     arr2 := [...]int{3, 4, 5, 6, 7}
9     printArr(&arr2)
10    //0 100
11    //1 4
12    //2 5
13    //3 6
14    //4 7
15    fmt.Println(arr2)
16    //[100 4 5 6 7]
17 }
```

二) 切片 (Slice)

- 定义

数组: [5]int

切片: []int

```
1 func main() {
2     arr := [...]int{0,1,2,3,4,5,6,7}
3     fmt.Println("arr[2:6] = ", arr[2:6])
4     fmt.Println("arr[:6] = ", arr[:6])
5     fmt.Println("arr[2:] = ", arr[2:])
6     fmt.Println("arr[:] = ", arr[:])
7 }
8 //arr[2:6] =  [2 3 4 5]
9 //arr[:6] =  [0 1 2 3 4 5]
10 //arr[2:] =  [2 3 4 5 6 7]
11 //arr[:] =  [0 1 2 3 4 5 6 7]
```

- 切片是按引用传递

```
1 func updateSlice(s []int) {
2     s[0] = 100
3 }
4
5 func main() {
6     arr := [...]int{0,1,2,3,4,5,6,7}
7 }
```

```

8   s1 := arr[2:]
9   fmt.Println("s1 = ", s1)
10  s2 := arr[:]
11  fmt.Println("s2 = ", s2)
12  fmt.Println("After update slice")
13  updateSlice(s1)
14  fmt.Println(s1)
15  fmt.Println(s2)
16  fmt.Println(arr)
17  }
18  //s1 =  [2 3 4 5 6 7]
19  //s2 =  [0 1 2 3 4 5 6 7]
20  //After update slice
21  //[100 3 4 5 6 7]
22  //[0 1 100 3 4 5 6 7]
23  //[0 1 100 3 4 5 6 7]

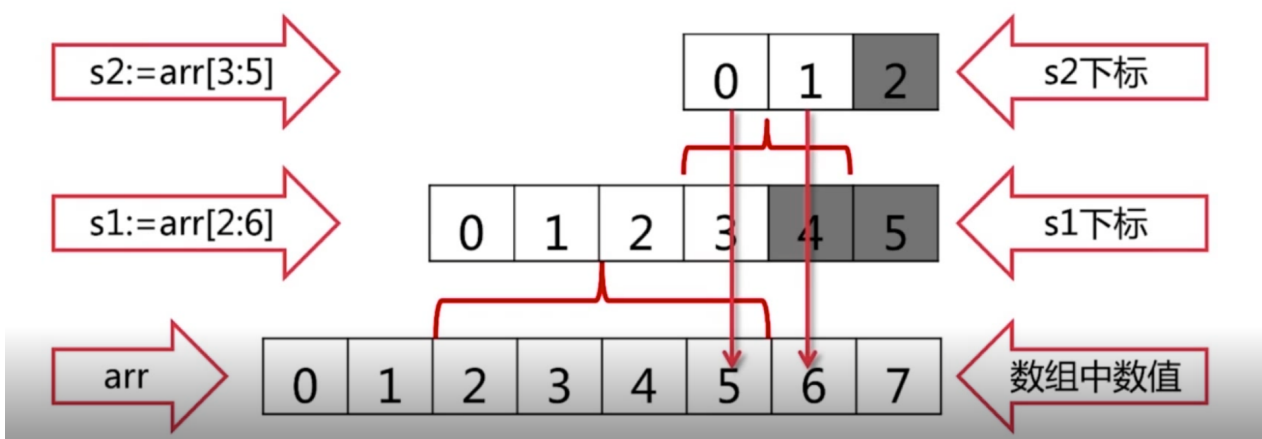
```

- 切片分割后，它仍然有原始切片的隐藏下标

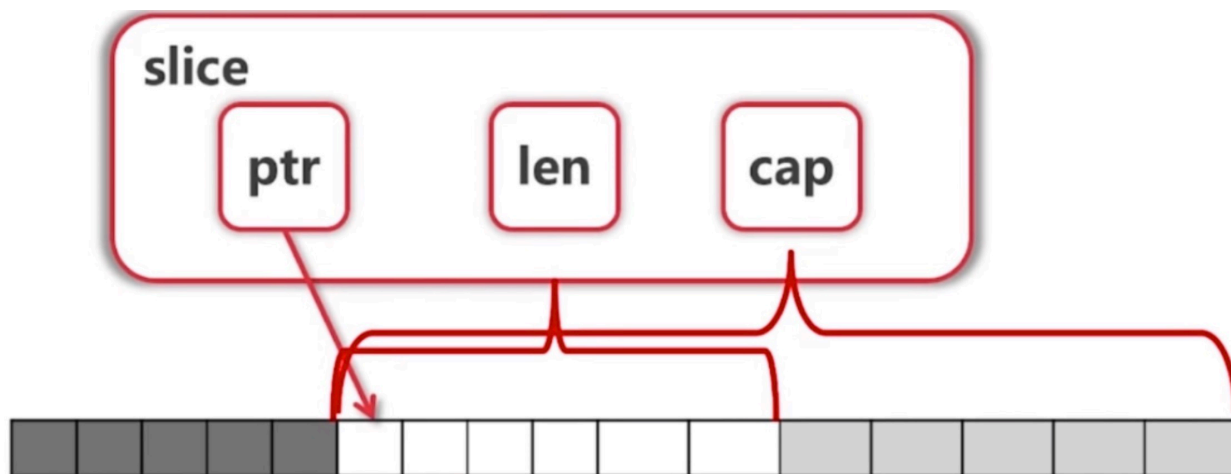
```

1  func main() {
2      arr := [...]int{0, 1, 2, 3, 4, 5, 6, 7}
3
4      s1 := arr[2:6]
5      s2 := s1[3:5]
6      fmt.Println(s1)
7      fmt.Println(s2)
8  }
9  //[2 3 4 5]
10 //[5 6]

```



切片的底层数据结构



- slice的扩展
 - slice可以向后扩展，不可以向前扩展
 - `s[i]`不可以超越`len(s)`，向后扩展不可以超越底层数组`cap(s)`

```

1 func main() {
2     arr := [...]int{0, 1, 2, 3, 4, 5, 6, 7}
3
4     s1 := arr[2:6]
5     s2 := s1[3:5]
6     fmt.Println(s1)
7     fmt.Println(s2)
8
9     fmt.Println("=====")
10    fmt.Printf("s1=%v, len(s1)=%d, cap(s1)=%d\n",
11        s1, len(s1), cap(s1))
12    fmt.Printf("s2=%v, len(s2)=%d, cap(s2)=%d\n",
13        s2, len(s2), cap(s2))
14
15    fmt.Println(s1[3:7]) // 越界
16 }
17 //[2 3 4 5]
18 //[5 6]
19 =====
20 //s1=[2 3 4 5], len(s1)=4, cap(s1)=6
21 //s2=[5 6], len(s2)=2, cap(s2)=3
22
23 // 越界
24 //panic: runtime error: slice bounds out of range

```