课程安排，请关注微信公众平台或者官方微博

编程语言： Golang 与 html5

编程工具： Goland 和 HBuilder

预计平均一周左右更新一或二节课程

授人以鱼，不如授人以渔业。

大家好，我是彬哥,
欢迎大家来到 Golang 语言社区云课堂课程的学习。Slice
社区论坛网址：**www.golang.ltd**
技术交流群 ： 221273219
微信公众号 ：**Golang 语言社区**
微信服务号 ：**Golang 技术社区**

第一季 Go 语言基础、进阶、提高课程

第十四节 Go 语言 切片（slice）

Go 语言切片是对数组的抽象。

Go 数组的长度不可改变，在特定场景中这样的集合就不太适用，Go 中提供了一种灵活，功能强悍的内置类型切片("动态数组"),与数组相比切片的长度是不固定的，可以追加元素，在追加时可能使切片的容量增大。

定义切片

你可以声明一个未指定大小的数组来定义切片：

```
var identifier []type
```

切片不需要说明长度。

或使用 make()函数来创建切片:

```
var slice1 []type = make([]type, len)

也可以简写为
```

```
slice1 := make([]type, len)
```

也可以指定容量，其中 capacity 为可选参数。

```
make([]T, length, capacity)
```

这里 len 是数组的长度并且也是切片的初始长度。

切片初始化

```
s :=[] int {1,2,3 }
```

直接初始化切片，[]表示是切片类型，{1,2,3}初始化值依次是 1,2,3.其 cap=len=3

```
s := arr[:]
```

初始化切片 s,是数组 arr 的引用

```
s := arr[startIndex:endIndex]
```

将 arr 中从下标 startIndex 到 endIndex-1 下的元素创建为一个新的切片

```
s := arr[startIndex:]
```

缺省 endIndex 时将表示一直到 arr 的最后一个元素

```
s := arr[:endIndex]
```

缺省 startIndex 时将表示从 arr 的第一个元素开始

```
s1 := s[startIndex:endIndex]
```

通过切片 s 初始化切片 s1

```
s :=make([]int,len,cap)
```

通过内置函数 make()初始化切片 s,[]int 标识为其元素类型为 int 的切片

append() 和 copy() 函数

如果想增加切片的容量，我们必须创建一个新的更大的切片并把原分片的内容都拷贝过来。

下面的代码描述了从拷贝切片的 copy 方法和向切片追加新元素的 append 方法。

```
package main


import "fmt"


func main() {
```

```
   var numbers []int
   printSlice(numbers)

   /* 允许追加空切片 */
   numbers = append(numbers, 0)
   printSlice(numbers)

   /* 向切片添加一个元素 */
   numbers = append(numbers, 1)
   printSlice(numbers)

   /* 同时添加多个元素 */
   numbers = append(numbers, 2,3,4)
   printSlice(numbers)

   /* 创建切片 numbers1 是之前切片的两倍容量*/
   numbers1 := make([]int, len(numbers), (cap(numbers))*2)

   /* 拷贝 numbers 的内容到 numbers1 */
   copy(numbers1,numbers)
   printSlice(numbers1)
}

func printSlice(x []int){
   fmt.Printf("len=%d cap=%d slice=%v\n",len(x),cap(x),x)
}
```

以上代码执行输出结果为：

```
len=0 cap=0 slice=[]
len=1 cap=1 slice=[0]
len=2 cap=2 slice=[0 1]
len=5 cap=6 slice=[0 1 2 3 4]
len=5 cap=12 slice=[0 1 2 3 4]
```

非学教宣