



Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

切片1

目录



PART 01
切片的概念



PART 02
创建切片



PART 03
切片截取



PART 04
切片的底层实现



切片的概念

1.切片的概念



切片的概念

- 1) 切片的英文是 slice
- 2) 切片是数组的一个引用，因此**切片是引用类型**，在进行传递时，遵守引用传递的机制。
- 3) 切片的使用和数组类似，遍历切片、访问切片的元素和求切片长度 `len(slice)` 都一样。
- 4) 切片的长度是可以变化的，因此切片是一个**可以动态变化数组**。
- 5) 切片定义的基本语法:

`var 切片名 []类型`

比如: `var a [] int`



创建切片

1.创建切片

切片和数组的区别：声明数组时，`[]`内写明了数组的长度或使用...自动计算长度，而声明切片时，`[]`内没有任何字符。

切片创建方法：

1.自动推导类型创建slice

`s1 := []int{1, 2, 3, 4}` 创建有4个元素的切片，分别为：1234

2.借助make创建 slice，格式：make(切片类型，长度，容量)

`s2 := make([]int, 5, 10)` `len(s2) = 5, cap(s2) = 10`

3.make时，没有指定容量，那么 长度==容量

`s3 := make([]int, 5)` `len(s3) = 5, cap(s3) = 5`

注意：make只能创建slice、map和channel，并且返回一个有初始值(非零)的对象。



切片截取

1.切片截取

切片截取

操作	含义
<code>s[n]</code>	切片 <code>s</code> 中索引位置为 <code>n</code> 的项
<code>s[:]</code>	从切片 <code>s</code> 的索引位置 0 到 <code>len(s)-1</code> 处所获得的切片
<code>s[low:]</code>	从切片 <code>s</code> 的索引位置 <code>low</code> 到 <code>len(s)-1</code> 处所获得的切片
<code>s[:high]</code>	从切片 <code>s</code> 的索引位置 0 到 <code>high</code> 处所获得的切片, <code>len=high</code>
<code>s[low:high]</code>	从切片 <code>s</code> 的索引位置 <code>low</code> 到 <code>high</code> 处所获得的切片, <code>len=high-low</code>
<code>s[low : high : max]</code>	从切片 <code>s</code> 的索引位置 <code>low</code> 到 <code>high</code> 处所获得的切片, <code>len=high-low</code> , <code>cap=max-low</code>
<code>len(s)</code>	切片 <code>s</code> 的长度, 总是 $\leq \text{cap}(s)$
<code>cap(s)</code>	切片 <code>s</code> 的容量, 总是 $\geq \text{len}(s)$

截取可表示为 `s[low: high: max]`。

`low`: 表示下标的起点。 `high`: 表示下标的终点（左闭右开，不包括此下标）。

内置的 `len()` 和 `cap()` 函数分别返回 `slice` 的长度和容量。

长度 `len = high - low`。容量 `cap = max - low`。

长度对应切片中元素的数目，不能超过容量。

容量一般是从 `slice` 的开始位置到底层数据的结尾位置。



切片和底层数组关系

利用数组创建切片。

切片在操作过程中，是直接操作原数组。

因此，在go语言中，我们常常使用切片代替数组。

```
func main() {  
    arr := []int {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}  
    s1 := arr[2:5]           // 从arr[2]开始，取 5-2 个元素，组成切片s1。  
    fmt.Println("s1=", s1)  // s1= [2 3 4]  
}
```



切片的底层实现

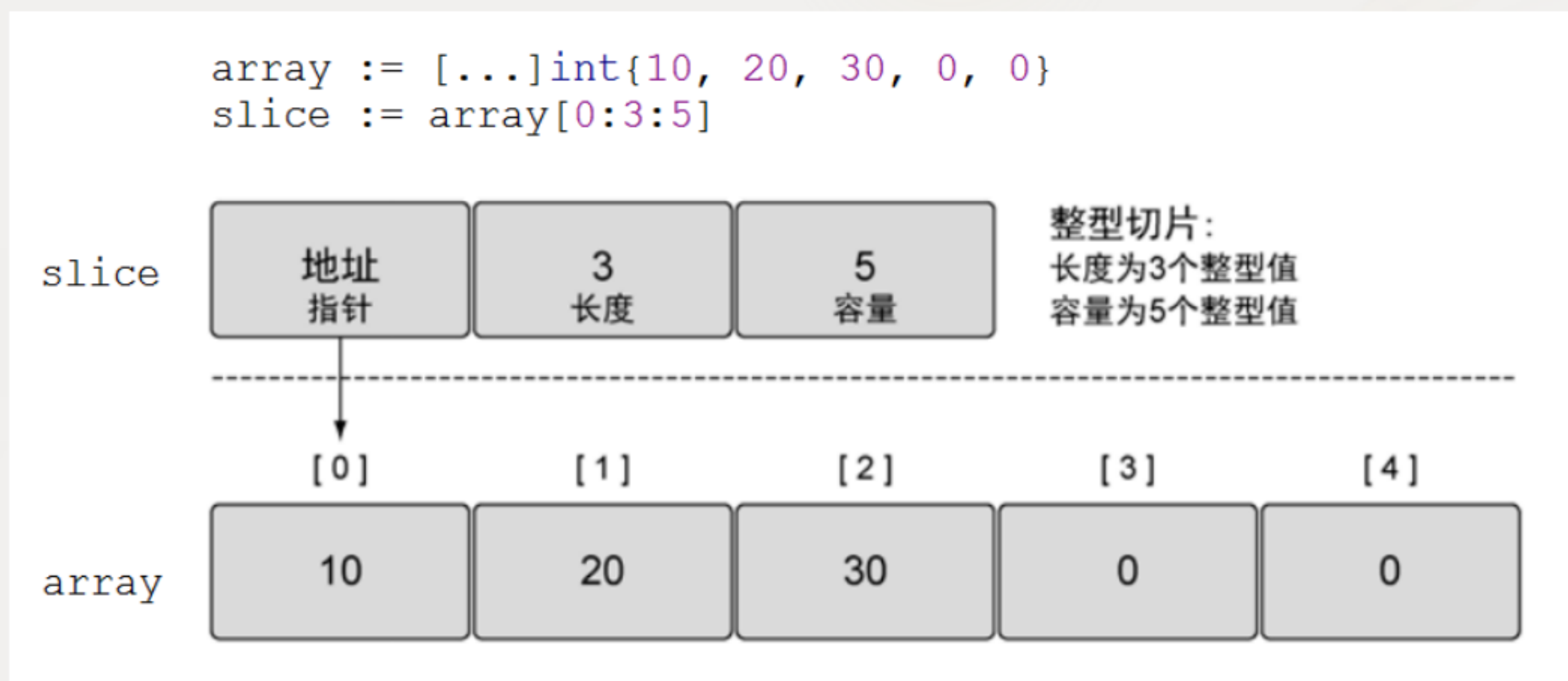
1.切片的底层实现

切片底层的实现

切片通过内部指针和相关属性引用数组片段，以实现变长方案。

slice并不是真正意义上的动态数组，而是一个数据结构，数据传递时属于传值操作。

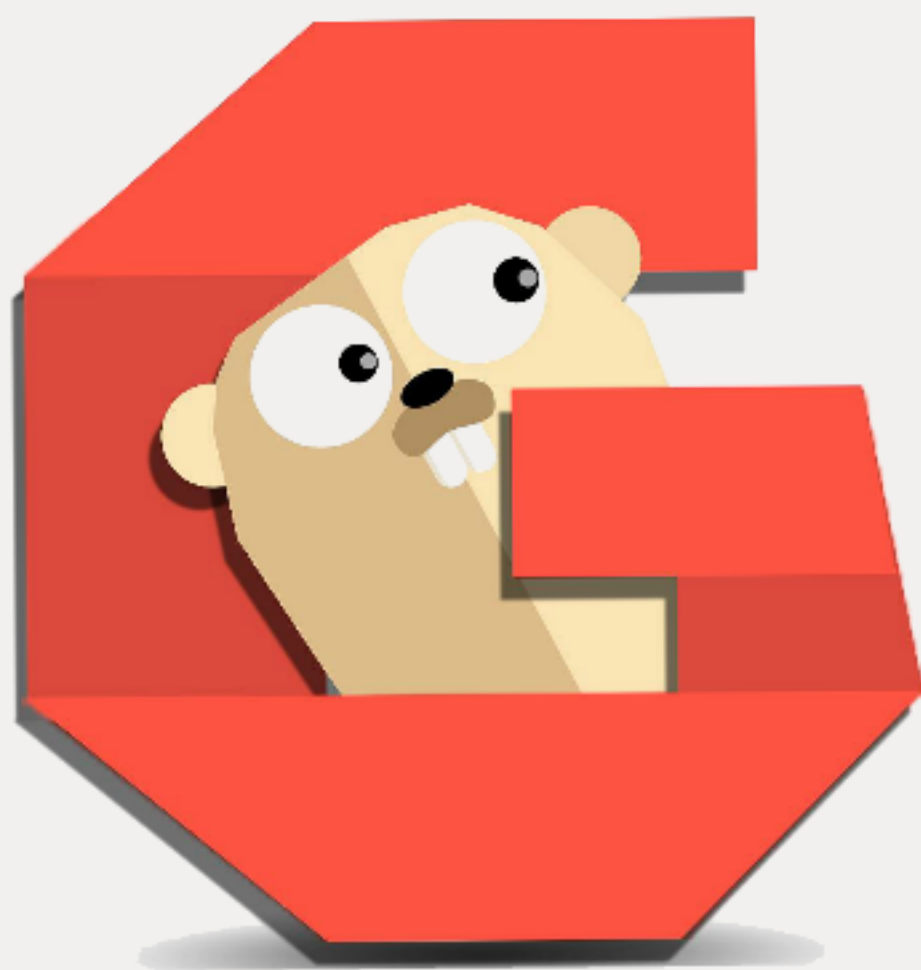
slice总是指向一个底层array，slice的声明也可以像array一样，只是不需要长度。





实例演示





Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

感谢您的聆听和观看