



Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

map

目录



PART 01
map简述



PART 02
map常用操作



map简述

1.map简述



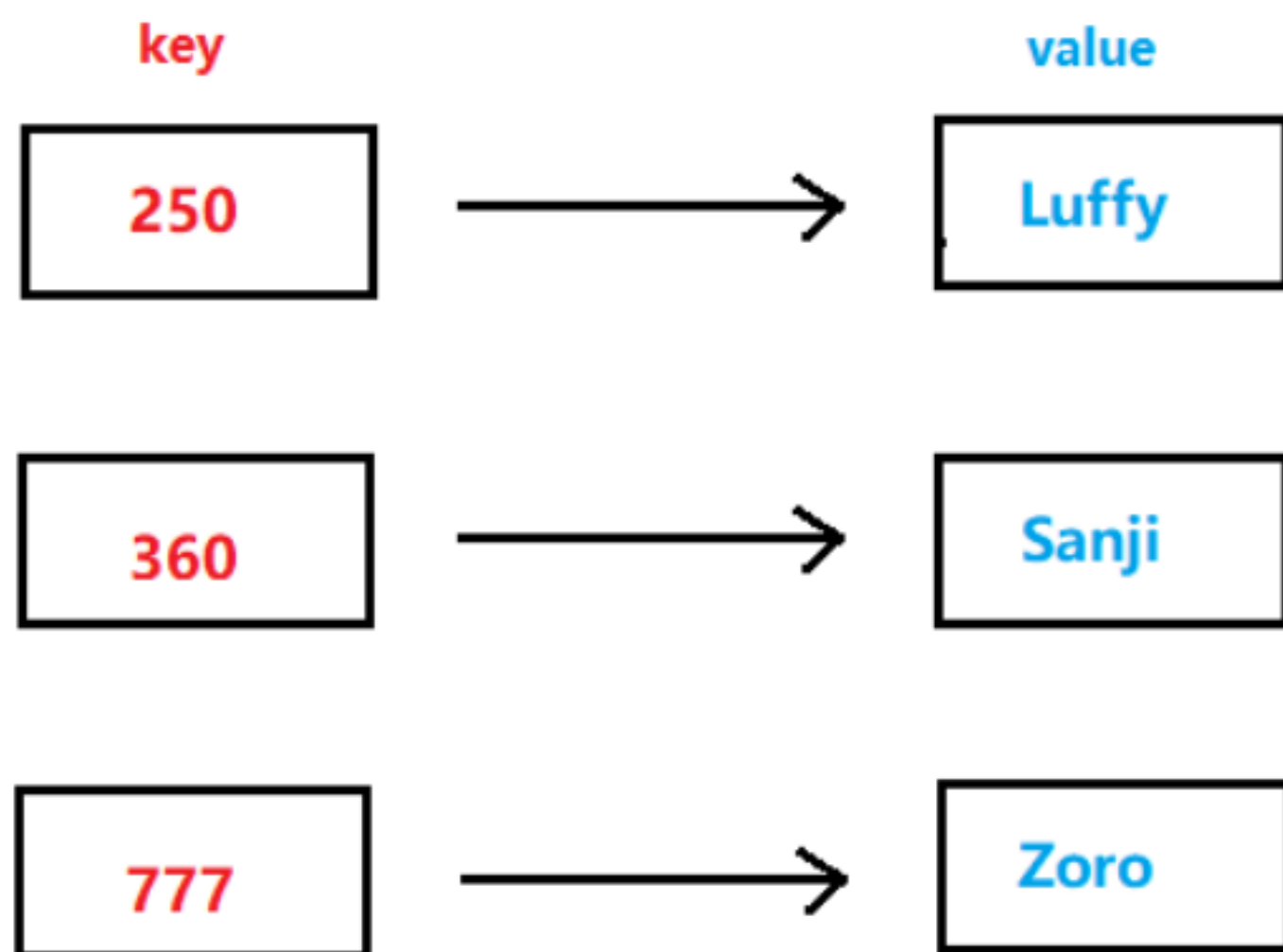
map简述

Go语言中的**map(映射、字典)**是一种内置的数据结构。

它是一个无序的key-value对的集合，比如以身份证号作为唯一键来标识一个人的信息。

Go语言中并没有提供一个set类型，但是map中的key也是不相同的，可以用map实现类似set的功能。

```
pirate := map[int]string {250:"Luffy", 360:"Sanji", 777:"Zoro"}
```





map常用操作

1.map常用操作



(1)、赋值

```
m1 := map[int]string{1: "Luffy", 2: "Sanji"}
m1[1] = "Nami"      //修改
m1[3] = "Zoro"      //追加， go底层会自动为map分配空间
fmt.Println(m1)     //map[1:Nami 2:Sanji 3:Zoro]

m2 := make(map[int]string, 10)    //创建map
m2[0] = "aaa"
m2[1] = "bbb"
fmt.Println(m2)                  //map[0:aaa 1:bbb]
fmt.Println(m2[0], m2[1])        //aaa bbb
```



(2)、遍历

Map的迭代顺序是不确定的，并且不同的哈希函数实现可能导致不同的遍历顺序。
在实践中，遍历的顺序是随机的，每一次遍历的顺序都不相同。
这是故意的，**每次都使用随机的遍历顺序可以强制要求程序不会依赖具体的哈希函数实现。**

```
m1 := map[int]string{1: "Luffy", 2: "Sanji"}  
//遍历1，第一个返回值是key，第二个返回值是value  
for k, v := range m1 {  
    fmt.Printf("%d ----> %s\n", k, v)  
    //1 ----> Luffy  
    //2 ----> yoyo  
}  
  
//遍历2，第一个返回值是key，第二个返回值是value（可省略）  
for k := range m1 {  
    fmt.Printf("%d ----> %s\n", k, m1[k])  
    //1 ----> Luffy  
    //2 ----> Sanji  
}
```




(3)、删除

使用delete()函数，指定key值可以方便的map中的k-v映射。

```
m1 := map[int]string{1: "Luffy", 2: "Sanji", 3: "Zoro"}

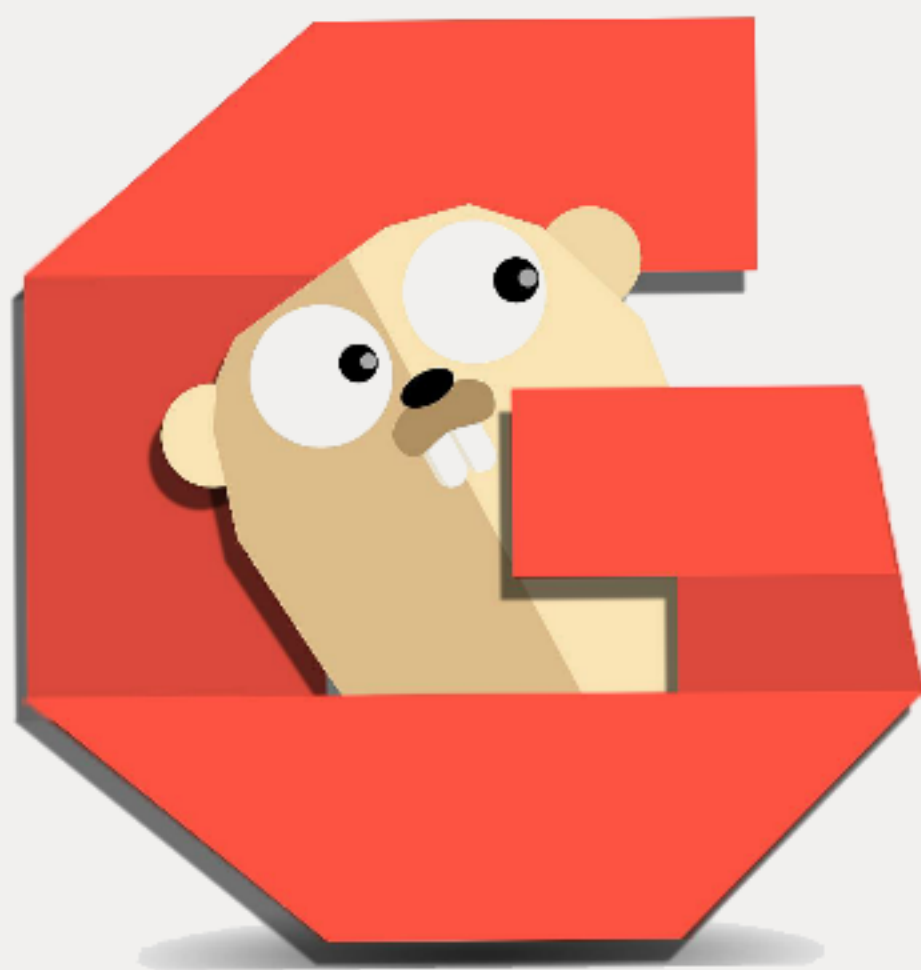
for k, v := range m1 { //遍历，第一个返回值是key，第二个返回值是value
    fmt.Printf("%d ----> %s\n", k, v)
}
//1 ----> Sanji
//2 ----> Sanji
//3 ----> Zoro
delete(m1, 2)           //删除key值为2的map

for k, v := range m1 {
    fmt.Printf("%d ----> %s\n", k, v)
}
//1 ----> Luffy
//3 ----> Zoro
```




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