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。
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

第十六节 Go 语言 map

Map 是一种无序的键值对的集合。Map 最重要的一点是通过 key 来快速检索数据，key 类似于索引，指向数据的值。

Map 是一种集合，所以我们可以像迭代数组和切片那样迭代它。不过，Map 是无序的，我们无法决定它的返回顺序，

这是因为 Map 是使用 hash 表来实现的。

定义 Map

可以使用内建函数 make 也可以使用 map 关键字来定义 Map:

```
/* 声明变量，默认 map 是 nil */
var map_variable map[key_data_type]value_data_type

/* 使用 make 函数 */
map_variable := make(map[key_data_type]value_data_type)
```

如果不初始化 map，那么就会创建一个 nil map。nil map 不能用来存放键值对

实例

下面实例演示了创建和使用 map:

```
package main

import "fmt"

func main() {
    var countryCapitalMap map[string]string /*创建集合 */
    countryCapitalMap = make(map[string]string)

    /* map 插入 key - value 对,各个国家对应的首都 */
    countryCapitalMap [ "France" ] = "Paris"
    countryCapitalMap [ "Italy" ] = "罗马"
    countryCapitalMap [ "Japan" ] = "东京"
    countryCapitalMap [ "India " ] = "新德里"

    /*使用键输出地图值 */ for country := range countryCapitalMap {
        fmt.Println(country, "首都是", countryCapitalMap [country])
    }

    /*查看元素在集合中是否存在 */
    captial, ok := countryCapitalMap [ "美国" ] /*如果确定是真实的,则存在,否则不存在 */
    /*fmt.Println(captial) */
    /*fmt.Println(ok) */
    if (ok) {
        fmt.Println("美国的首都是", captial)
    } else {
        fmt.Println("美国的首都不存在")
    }
}
```

以上实例运行结果为：

```
France 首都是 Paris
Italy 首都是 罗马
Japan 首都是 东京
India  首都是 新德里
美国的首都不存在
```

delete() 函数

delete() 函数用于删除集合的元素，参数为 map 和其对应的 key。实例如下：

```
package main

import "fmt"

func main() {
    /* 创建 map */
    countryCapitalMap := map[string]string{"France": "Paris", "Italy": "Rome", "Japan": "Tokyo", "Ind
ia": "New delhi"}

    fmt.Println("原始地图")

    /* 打印地图 */
    for country := range countryCapitalMap {
        fmt.Println(country, "首都是", countryCapitalMap [ country ])
    }

    /*删除元素*/ delete(countryCapitalMap, "France")
    fmt.Println("法国条目被删除")

    fmt.Println("删除元素后地图")

    /*打印地图*/
    for country := range countryCapitalMap {
        fmt.Println(country, "首都是", countryCapitalMap [ country ])
    }
}
```

以上实例运行结果为：

```
原始地图
India 首都是 New delhi
France 首都是 Paris
Italy 首都是 Rome
Japan 首都是 Tokyo
法国条目被删除
删除元素后地图
Italy 首都是 Rome
Japan 首都是 Tokyo
India 首都是 New delhi
```