1.Map定义：

* map[K]V,复合map：map[K1]map[K2]V

```
func mapDemo() {
   m := map[string]string {
      "name"    :  "ccmounse",
      "course"  :  "goloang",
      "site"    :  "kenrou",
      "quality" :  "good",
   }
   println("map = ", m)
}
```

* 结果：
```
map =  map[course:goloang name:ccmounse quality:good site:kenrou]
```

2.定义空map：

```
func mapDemo() {
    // 定义空map
    m2 := make(map[string]int)  // m2 == empty map
    var m3 map[string]int       // m3 == nil
    fmt.Println("map2 = ", m2)
    fmt.Println("map3 = ", m3)
}
```

* 结果：
```
map2 =  map[]
map3 =  map[]
```

3.map的遍历：

* 底层实现是：hashMap(无须的)

```
func traversingMap() {
   m := map[string]string {
      "name"    :  "ccmounse",
      "course"  :  "goloang",
      "site"    :  "kenrou",
      "quality" :  "good",
   }
   for k, v := range m {
      fmt.Printf("k = %s, v = %s", k, v)
      fmt.Println()
   }
}
```

* 结果：
```
k = name, v = ccmounse
k = course, v = goloang
k = site, v = kenrou
k = quality, v = good
```

4.map操作

```
func operateMap() {
   m := map[string]string {
      "name"    :  "ccmounse",
      "course"  :  "goloang",
      "site"    :  "kenrou",
      "quality" :  "good",
   }
   courseName := m["course"]
   fmt.Println("course name = ", courseName)
   cseName := m["cse"]                       // key值填写错误
   fmt.Println("cse name = ", cseName)

    // 判断key是否存在
    courseName1, ok := m["course"]
    fmt.Println("course name1 = ", courseName1, ok)
    cseName1, ok := m["cse"]                       // key值填写错误
    fmt.Println("cse name1 = ", cseName1, ok)

    // 一般写法
    if cseName2, ok := m["cse"]; ok {
       fmt.Println("cse name2 = ", cseName2)
    } else {
       fmt.Println("key does not exist!")
    }
}
```

* 结果：

```
course name =  goloang
cse name =
course name1 =  goloang true
cse name1 =   false
key does not exist!
```

5.删除元素

```
func deletingElements() {
   m := map[string]string {
      "name"    :  "ccmounse",
      "course"  :  "goloang",
      "site"    :  "kenrou",
      "quality" :  "good",
   }


   fmt.Println("Deleting elements")
   name, ok := m["name"]
   fmt.Printf("name = %s, ok = %v", name, ok)


   fmt.Println();


   delete(m, "name")
   name, ok = m["name"]
   fmt.Printf("name = %s, ok = %v", name, ok)
}
```

* 结果：

```
Deleting elements
name = ccmounse, ok = true
name = , ok = false
```

6. map的key

```
* map使用哈希表，必须可以比较相等
* 除了slice， map， function的内建类型都可以作为key
* Struct类型不包含上述类型，也能作为key
```