

Go Training

Session 4

Slices

- Copy Function

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var nilSlice []int
8     emptySlice := make([]int, 2, 5)
9
10    firstSlice := []int{11,12,13}
11    secondSlice := []int{21,22,23,24}
12    thirdSlice := []int{31,32}
13
14    n1 := copy(nilSlice,firstSlice)
15    fmt.Println(n1, nilSlice) // n1=0, nilSlice=[]
16
17    n2 := copy(emptySlice,firstSlice)
18    fmt.Println(n2, emptySlice) // n2=2, emptySlice=[11,12]
19
20    n3 := copy(firstSlice,secondSlice)
21    fmt.Println(n3, firstSlice) // n3=3, firstSlice=[21,22,23]
22
23    n4 := copy(secondSlice,thirdSlice)
24    fmt.Println(n4, secondSlice) // n4=2, secondSlice=[31,32,23,24]
25
26 }
27
```

Slices

- Spread Operator

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     firstSlice := []int{11,12,13}
8     secondSlice := []int{21,22,23,24}
9
10    firstSlice = append(firstSlice,secondSlice...)
11    fmt.Println(firstSlice) // [11,12,13,21,22,23,24]
12
13 }
```

Slices

- Extract Operator

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     firstSlice := []int{11,12,13,14,15,16}
8     s1 := firstSlice[0:3]
9     fmt.Println(s1) // [11,12,13]
10
11     s2 := firstSlice[:3]
12     fmt.Println(s2) // [11,12,13]
13
14     s3 := firstSlice[2:6]
15     fmt.Println(s3) // [13,14,15,16]
16
17     s4 := firstSlice[2:]
18     fmt.Println(s4) // [13,14,15,16]
19
20     s5 := firstSlice[:]
21     fmt.Println(s5) // [11,12,13,14,15,16]
22
23 }
```

Slices

- Iteration over elements of a slice
 - Same as array
- Comparing two slices
 - A slice can only be compared to nil
- Multi-dimensional Slices
 - Similar to multi-dimensional arrays

- Deleting an element of a slice

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     firstSlice := []int{11,12,13,14,15}
8
9     //delete third element of firstslice
10    firstSlice = append(firstSlice[:2], firstSlice[3:]...)
11    fmt.Println(firstSlice) // [11,12,14,15]
12
13 }
14
```

Slices

- Slice is passed by reference to the function
- Array is passed by value to the function

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     firstArray := [5]int{11,12,13,14,15}
8
9     newValues(firstArray)
10
11     fmt.Println(firstArray) // [11,12,13,14,15]
12
13 }
14
15 func newValues(arr [5]int) {
16     for i,v := range arr{
17         arr[i] = v - 10
18     }
19
20     fmt.Println("exiting newValues functiona call")
21 }
22
```

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     firstSlice := []int{11,12,13,14,15}
8
9     newValues(firstSlice)
10
11     fmt.Println(firstSlice) // [1,2,3,4,5]
12
13 }
14
15 func newValues(slc []int) {
16     for i,v := range slc{
17         slc[i] = v - 10
18     }
19
20     fmt.Println("exiting newValues functiona call")
21 }
22
```

Maps

- A map holds data represented by *key:value* pairs

```
1 package main
2
3 import "fmt"
4
5 ▼ func main() {
6
7     var nilMap map[string]int
8
9     fmt.Println(nilMap) //map[]
10
11     nilMap["Axar"] = 22 //will throw an error
12
13     emptyMap := make(map[string]int)
14     fmt.Println(emptyMap) //map[]
15     emptyMap["Axar"] = 22
16     emptyMap["Rishabh"] = 23
17     fmt.Println(emptyMap) //map[Axar:22 Rishabh:23]
18     fmt.Println(emptyMap["Axar"]) //22
19     fmt.Println(emptyMap["Dhoni"]) //0
20     value, ok := emptyMap["Dhoni"]
21     fmt.Println(value, ok) //0, false
22
23     delete(emptyMap, "Axar")
24 ▼ newMap := map[bool]string{
25     true : "Yes",
26     false : "No",
27 }
28
29     fmt.Println(newMap, len(newMap)) //map[false:No true:Yes], 2
30
31 }
32
```

Maps

- Maps are referenced type
- Iterating over a Map

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     newMap := map[string]string{
8         "Tamil Nadu" : "Chennai",
9         "Karnataka" : "Bangalore",
10        "MP" : "Bhopal",
11        "UP" : "Lucknow",
12    }
13    for key, val := range newMap{
14        fmt.Println (key, "->", val)
15    }
16
17 }
```

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var copiedMap map[string]string
8
9     newMap := map[string]string{
10        "Tamil Nadu" : "Chennai",
11        "Karnataka" : "Bangalore",
12        "MP" : "Bhopal",
13        "UP" : "Lucknow",
14    }
15
16    copiedMap = newMap
17
18    delete(copiedMap, "MP")
19
20    fmt.Println(newMap)
21    fmt.Println(copiedMap)
22
23 }
24
```


Assignment

1. Write a function to print alternate elements of an integer array starting with its first element. Call this function on arrays [11,12,13,14,15,16,17] and [-1,1, 2, -2,-3,3]
2. Write a function to print frequency of each element of a string array. Call this function on arrays ["0","0", "3"," 0", "5", "3", "3", "0", "5", "2", "4", "3"], ["hi", "hello", "Hi", "hi"]

Thank You