

Go Training

Session 5

Variadic Functions

- Can take variable number of arguments

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     newSlice := []int{1,2,3}
8     result1 := multiplier(4,10,20,30,40)
9     result2 := multiplier(4,newSlice...)
10
11     fmt.Println(result1,result2)
12 }
13
14 func multiplier(m int, args ...int) []int {
15     multipliedSlice := make([]int, len(args))
16     for i,v := range args{
17         multipliedSlice[i] = m*v
18     }
19
20     return multipliedSlice
21 }
22
```

Pointers

- Pointer is a variable that stores the memory address referenced by other variable

```
1 package main
2
3 import "fmt"
4
5 ▼ func main() {
6
7     x := 5
8     var newPointer *int
9
10    fmt.Printf("Type of newPointer is %T and value is %v \n", newPointer, newPointer)
11
12    newPointer = &x
13
14    fmt.Printf("Type of newPointer is %T and value is %v \n", newPointer, newPointer)
15
16    fmt.Println(*newPointer)
17    *newPointer = 10
18    fmt.Println(x)
19
20 }
21
```

Pointers

- new function

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     newPointer := new(int)
8
9     fmt.Printf("Type of newPointer is %T and value is %v \n", newPointer, newPointer)
10
11     fmt.Println(*newPointer)
12
13 }
14
```

- Passing a pointer to a function

Reading Assignment

1. Go through all the 27 pages of <https://tour.golang.org/moretypes>
2. Go through first 14 pages of <https://tour.golang.org/methods>

Thank You