

Pointers



Pointer

value , & and *

```
func foo(number int) {}  
func bar(number *int) {}
```

Parameters in Go are always passed by value,
and a copy of the value being passed is made.



golang : pass by copy value

create main.go in folder chapter8-1 :

```
func main() {  
    amount := 5  
    double(amount)  
    fmt.Printf("original %v\n", amount)  
}  
  
func double(number int) {  
    number *= 2  
    fmt.Println(number)  
}
```

run -> no error -> push to your git repository



golang : pass by pointer

create main.go in folder chapter8-2 :

```
func main() {  
    amount := 5  
    double(&amount)  
    fmt.Printf("original %v\n", amount)  
}  
  
func double(number *int) {  
    *number *= 2  
    fmt.Println(*number)  
}
```

When you pass a pointer, the pointer value will be copied and passed.

run -> no error -> push to your git repository



golang : pass by array value

create main.go in folder chapter8-3 :

```
func main() {  
    array := [3]int{1, 2, 3}  
    double(array)  
    fmt.Printf("origin addr %p\n", &array)  
    fmt.Printf("original %v\n", array)  
}  
  
func double(nums [3]int) {  
    fmt.Printf("double addr %p\n", &nums)  
    for i := range nums {  
        nums[i] *= 2  
    }  
    fmt.Println(nums)  
}
```

run -> no error -> push to your git repository



golang : pass by array pointer

create main.go in folder chapter8-4 :

```
func main() {  
    array := [3]int{1, 2, 3}  
    double(&array)  
    fmt.Printf("origin addr %p\n", &array)  
    fmt.Printf("original %v\n", array)  
}  
  
func double(nums *[3]int) {  
    fmt.Printf("double addr %p\n", nums)  
    fmt.Printf("double value %v\n", *nums)  
    for i := range *nums {  
        nums[i] *= 2  
    }  
    fmt.Println(*nums)  
}
```

run -> no error -> push to your git repository



golang : pass by slice value

create main.go in folder chapter8-5 :

```
func main() {  
    slice := []int{1, 2, 3}  
    double(slice)  
    fmt.Printf("origin addr %p\n", slice)  
    fmt.Printf("original %v\n", slice)  
}  
  
func double(nums []int) {  
    fmt.Printf("double addr %p\n", nums)  
    for i := 0; i < len(nums); i++ {  
        nums[i] *= 2  
    }  
    fmt.Println(nums)  
}
```

run -> no error -> push to your git repository



golang : pass by map value

create main.go in folder chapter8-6 :

```
func main() {  
    m := map[int]int{1:1, 2:2, 3:3,}  
    double(m)  
    fmt.Printf("origin addr %p\n", m)  
    fmt.Printf("original %v\n", m)  
}  
  
func double(nums map[int]int) {  
    fmt.Printf("double addr %p\n", nums)  
    for i := range nums {  
        nums[i] *= 2  
    }  
  
    fmt.Println(nums)  
}
```

run -> no error -> push to your git repository

