



链滴

一个 demo 理解 golang defer

作者: [xhaoxiong](#)

原文链接: <https://ld246.com/article/1541068614415>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

eg1:
func main() {
    var a = make(map[string]interface{})
    defer func() {
        fmt.Println(a)
    }()

    m:= make(map[string]interface{})
    m["1"] =1
    m["2"] =2

    a= m
}

```

```

eg2:
func main() {
    fmt.Println(test2())
}

```

```

func test2() map[string]interface{} {
    var a = make(map[string]interface{})
    defer func() {
        m:= make(map[string]interface{})
        m["1"] =1
        m["2"] =2

        a= m

    }()

    return a
}

```

```

eg3:
func main() {
    var a = make(map[string]interface{})
    defer fmt.Println(a)

    m:= make(map[string]interface{})
    m["1"] =1
    m["2"] =2

    a= m
}

```

eg1 eg2

1. 将 `func () {fmt.Println}()` 压入栈
2. 对a进行赋值

3. return
4. 出栈执行func()
5. func外变量重新赋值给里面的变量
6. 返回对应的值 (如果为指针则defer中将修改变量的值,否则不修改)

eg1:map[1:1 2:2]

eg2:map[](从func()(e) e中传值也是返回这个值,返回的是字面量)

eg3:map[]