# **golang 函数以及函数和方法的区别**

原创 2016年12月11日 20:12:54

* 5979

在接触到go之前，我认为函数和方法只是同一个东西的两个名字而已（在我熟悉的c/c++，python，java中没有明显的区别），但是在golang中者完全是两个不同的东西。官方的解释是，方法是包含了接收者的函数。到底什么意思呢。

首先函数的格式是固定的，func＋函数名＋ 参数 ＋ 返回值（可选） ＋ 函数体。例

func main（）

{

fmt.Println("Hello go")

}

在golang中有两个特殊的函数，main函数和init函数，main函数不用介绍在所有语言中都一样，它作为一个程序的入口，只能有一个。init函数在每个package是可选的，可有可无，甚至可以有多个(但是强烈建议一个package中一个init函数)，init函数在你导入该package时程序会自动调用init函数，所以init函数不用我们手动调用,l另外它只会被调用一次，因为当一个package被多次引用时，它只会被导入一次。

**[plain]** [view plain](https://blog.csdn.net/d_guco/article/details/53575067%22%20%5Co%20%22view%20plain%22%20%5Ct%20%22https%3A//blog.csdn.net/d_guco/article/details/_self) [copy](https://blog.csdn.net/d_guco/article/details/53575067%22%20%5Co%20%22copy%22%20%5Ct%20%22https%3A//blog.csdn.net/d_guco/article/details/_self)

1. package mypackage
2.
3. import (
4. "fmt"
5. )
6.
7. var I int
8.
9. func init() {
10. I = 0
11. fmt.Println("Call mypackage init1")
12. }
13.
14. func init() {
15. I = 1
16. fmt.Println("Call mypackage init2")
17. }

**[plain]** [view plain](https://blog.csdn.net/d_guco/article/details/53575067%22%20%5Co%20%22view%20plain%22%20%5Ct%20%22https%3A//blog.csdn.net/d_guco/article/details/_self) [copy](https://blog.csdn.net/d_guco/article/details/53575067%22%20%5Co%20%22copy%22%20%5Ct%20%22https%3A//blog.csdn.net/d_guco/article/details/_self)

1. package main
2.
3. import (
4. "demo/mypackage"
5. "fmt"
6. )
7.
8. func main() {
9. fmt.Println("Hello go.... I = ", mypackage.I)
10. }

运行结果：



我们可以看到，程序为我们自动调用了两个init函数，并且是按照顺序调用的。

下面来看方法。

**[plain]** [view plain](https://blog.csdn.net/d_guco/article/details/53575067%22%20%5Co%20%22view%20plain%22%20%5Ct%20%22https%3A//blog.csdn.net/d_guco/article/details/_self) [copy](https://blog.csdn.net/d_guco/article/details/53575067%22%20%5Co%20%22copy%22%20%5Ct%20%22https%3A//blog.csdn.net/d_guco/article/details/_self)

1. package main
2.
3. import "fmt"
4.
5. type myint int
6.
7. //乘2
8. func (p \*myint) mydouble() int {
9. \*p = \*p \* 2
10. return 0
11. }
12.
13. //平方
14. func (p myint) mysquare() int {
15. p = p \* p
16. fmt.Println("mysquare p = ", p)
17. return 0
18. }
19.
20. func main() {
21. var i myint = 2
22. i.mydouble()
23. fmt.Println("i = ", i)
24. i.mysquare()
25. fmt.Println("i = ", i)
26. }

运行结果：



我们可以看到方法和函数的区别，方法在func关键字后是接收者而不是函数名，接收者可以是自己定义的一个类型，这个类型可以是struct，interface，甚至我们可以重定义基本数据类型。我们可以给他一些我们想要的方法来满足我们的实际工程中的需求，就像上面一样我重定义了int并给了它一个乘2和平法的方法，这里我们要注意一个细节，接收者是指针和非指针的区别，我们可以看到当接收者为指针式，我们可以通过方法改变该接收者的属性，但是非指针类型缺做不到。

这里的接收者和c++中的this指针有一些相似，我们可以把接受者当作一个class，而这些方法就是类的成员函数，当接收者为指针类型是就是c++中的非const成员函数，为非指针时就是const成员函数，不能通过此方法改变累的成员变量。