



Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

匿名函数

目录



PART 01 匿名函数



匿名函数

1.匿名函数



匿名函数

前面我们定义函数的时候，发现是不能在一个函数中，再次定义一个函数的。

如果我们想在一个函数中再定义一个函数，那么就可以使用匿名函数。

所谓匿名函数就是没有名字的函数。

```
func main() {  
    var num int  
    num = 9  
    f := func() {  
        num++  
        fmt.Println("匿名函数: ", num)  
    }  
    f()  
    fmt.Println("main函数:", num)  
}
```

匿名函数，注意没有函数名字。



匿名函数

```
func main() {  
    var num int  
    num = 9  
    f := func() {  
        num++  
        fmt.Println("匿名函数: ", num)  
    }  
    f()  
    fmt.Println("main函数:", num)  
}
```

匿名函数，注意没有函数名字。

在main()函数中定义了一个匿名函数，定义的方式非常简单func(){ 函数体 }

一定要注意的是在func的后面没有函数的名字，同时在这里定义的该匿名函数也没有参数。

我们将定义好的匿名函数赋值给了变量f,那么变量f就是一个函数类型。

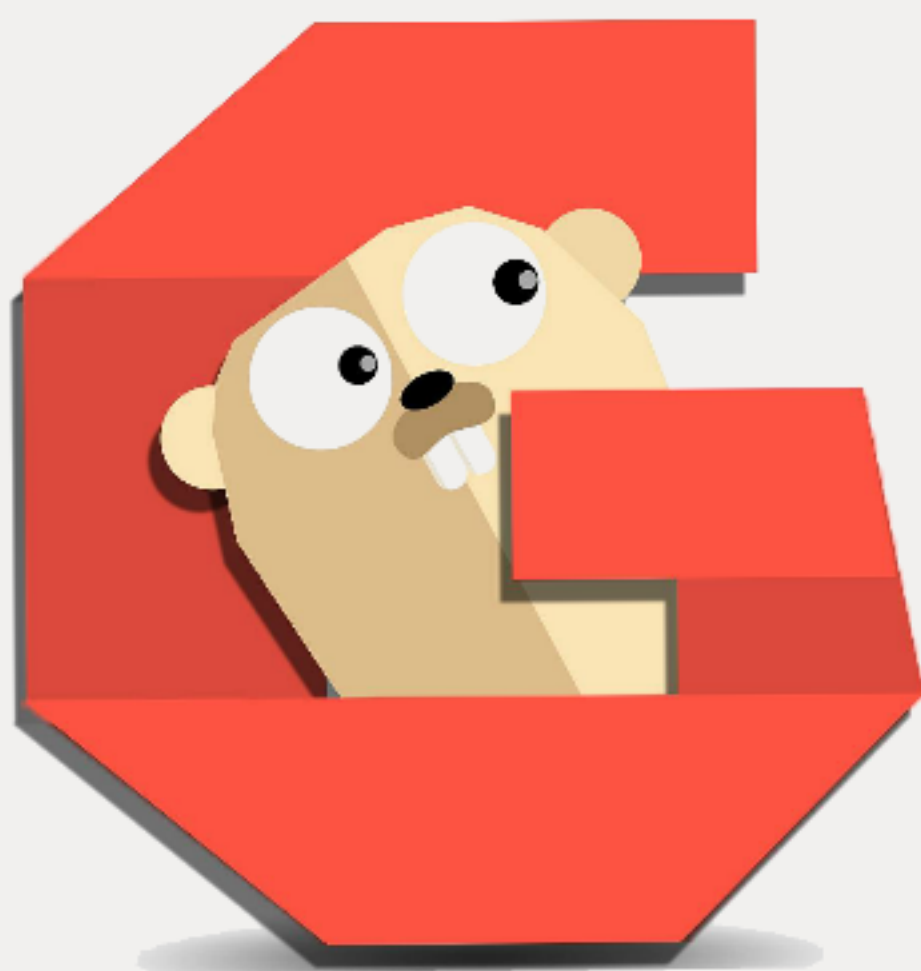
要想执行该匿名函数，就可以通过f()的方式去调用执行。

在这里，有一件非常有意思的事情，就是在匿名函数中可以直接访问main()函数中定义的局部变量，并且在匿名函数中对变量的值进行了修改，最终会影响到整个main()函数中定义的变量的值，所以上面两行输出都是10。关于这一点，一定与上一节讲解的函数作用域进行区别。



实例演示





江洲老师云课堂

—— 主讲：江洲老师 ——

感谢您的聆听和观看