

# Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

## 方法值与表达式

# 目录



PART 01  
方法值



PART 02  
方法表达式



# 方法值

## 1.方法值



## 方法值

我们调用结构体(类)中的方法，一般都是通过如下的方式：

```
var dog Dog
dog.Bark()
var cat Cat
cat.Brak()
```

或者是指针变量，现在，给大家补充另外一种方式。

```
func main() {
    var dog Dog
    dFunc := dog.Bark
    dFunc()
}
```

保存方法地址

以上调用的方式称为方法值，这种方式隐藏了接收者。



# 方法表达式

## 1.方法表达式



## 方法表达式

除了方法值，还有一种调用的方式是通过方法表达式。如下所示：

```
type Person struct {  
    name string //名字  
    sex  byte   //性别, 字符类型  
    age  int    //年龄  
}  
  
func (p Person) SetInfoValue() {  
    fmt.Printf("SetInfoValue: %p, %v\n", &p, p)  
}  
  
func (p *Person) SetInfoPointer() {  
    fmt.Printf("SetInfoPointer: %p, %v\n", p, p)  
}  
  
func main() {  
    p := Person{"mike", 'm', 18}  
    fmt.Printf("main: %p, %v\n", &p, p)  
  
    //方法值    f := p.SetInfoPointer //隐藏了接收者  
    //方法表达式  
    f := (*Person).SetInfoPointer  
    f(&p) //显式把接收者传递过去 ===> p.SetInfoPointer()  
  
    f2 := (Person).SetInfoValue  
    f2(p) //显式把接收者传递过去 ===> p.SetInfoValue()
```

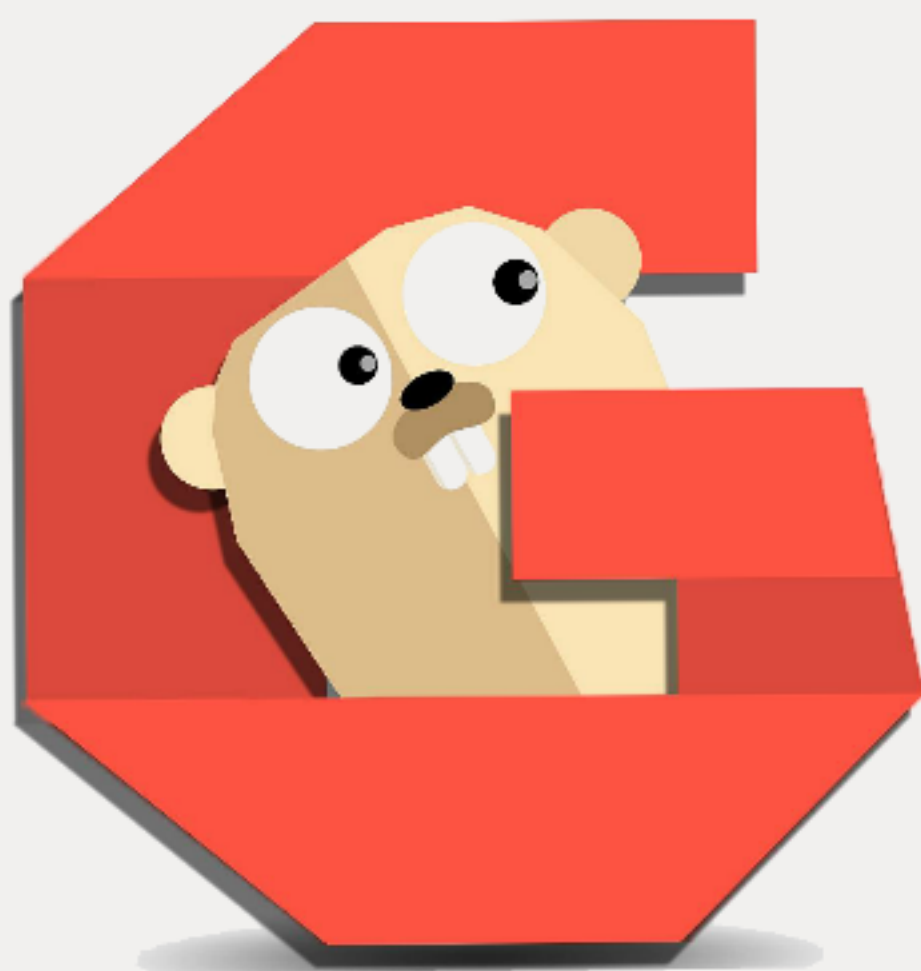
方法表达式需要显示把接收者传递过去。



## 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