

# 方法和接收者

## 指针接收者和值接收者

```
1 type dog struct {
2     name string
3 }
4
5 // 构造函数
6 func newdog(name string) dog {
7     return dog{name}
8 }
9
10 // 方法是作用于特定类型的函数
11 func (d dog) wang() { //方法名前面的括号中就是接收者
12     fmt.Printf("%s汪汪汪", d.name)
13 }
14 func main() {
15     d1 := newdog("zhoulin")
16     d1.wang() //使用接收者进行调用
17 }
```

## 指针接收者和值接收者

```
1 //值接收者
2 func (d dog) wang() {
3     fmt.Printf("%s汪汪汪", d.name)
4 }
5 //指针接收者
6 func (d *dog) guonian() {
7     d.age++
8 }
```

当其中一种方法使用指针接收者，则其他方法尽量也使用指针接收者，以保持一致性。

值接收者：传拷贝

指针接收者：接收地址

添加方法智能给自己添加的类型添加方法，程序中内置数据类型无法添加方法。