



链滴

Go 函数

作者: [feiwo](#)

原文链接: <https://ld246.com/article/1609944951232>

来源网站: [链滴](#)

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



如果把人比作一个程序，那么人的五官就是这个程序的属性，而人的行为就是这个程序的函数，函数述的是一种行为。

函数

几乎所有的编程语言都会有函数类型，用以将零散的代码组织起来，封装成一个独立的函数，表述一行为方式(处理过程)。

在Java中函数的表示法，通常在类中进行定义，称为方法，Java中没有独立的函数，都需要定义在类。

```
public class Person {  
    public static void main(String[] args) {  
        Person p = new Person();  
        p.say("feiwo") // 调用方法  
    }  
  
    // 定义方法  
    public void say(String msg) {  
        System.out.println(msg)  
    }  
}
```

在Java中，函数的定义是没有关键字的，而有些语言中，则需要使用函数关键字来标识这是一个函数譬如：在Python中有def,在PHP和JavaScript中有function，在Go中则是使用func来进行定义。

Python

```
def say(msg):  
    print(msg)
```

PHP

```
function say(msg) {  
    echo msg;  
}
```

JavaScript

```
function say(msg) {  
    console.log(msg)  
}
```

Go

```
func Say(msg string) {  
    fmt.Println(msg)  
}
```

上面列举的语言中，除了 **Java**，其他的语言都将函数作为一等公民，函数可以独立存在，直接被主函数调用。

Go函数定义

在**Go**中，有两种函数类型，一种的是有名函数，一种是匿名函数，顾名思义，一种是有名字的，一种是没有名字，都使用 **func** 关键字来定义。

定义格式为：**func (InputTypeList)OutPutTypeList**

```
// 命名函数  
func Say(msg string) {  
    fmt.Println(msg)  
}  
  
func main() {  
    // 函数调用  
    Say("feiwo")  
  
    // 匿名函数  
    f := func(msg string) {  
        fmt.Println(msg)  
    }  
    f("feiwo")  
}
```

在**Go**中，函数可以作为类型参数或返回值参数传递，也可以使用 **type** 关键字定义新的类型

函数作为参数

```
// ResponseCallback 为函数类型  
type ResponseCallback func(resp string)  
  
// 将ResponseCallback函数类型作为参数传递  
func Req(url string, callback ResponseCallback) {  
    callback(url + "response...")  
}
```

```

}

func main() {
    Req("https://www.feiwo.xyz",func (resp string) {
        fmt.Println(resp)
    })
}

```

函数作为返回值

// 定义一个结果返回的函数类型， 参数为convert函数参数

```

type FResult func(convert func(s interface{}) string) interface{}

```

// handle函数返回值为FResult的函数类型

```

func handle(origin interface{}) FResult {
    return func(convert func(s interface{}) string) interface{} {
        return convert(origin)
    }
}

```

```

func main() {
    // 调用函数， handle因为返回的是一个函数类型， 写作为f()， 调用handle时立即执行FResult函数， 因为FResult函数的参数是一个函数类型，
    // 这里使用了匿名函数方式实现， 也可以定义一个命名函数传入
    result := handle("feiwo")(func(s interface{}) string {
        return "my name is "+s.(string)
    })
    fmt.Println(result)
}

```

函数签名

在平时的开发中， 会编写大量的函数， 那么系统系统是怎么识别是这个函数的呢？ 答案是通过函数签名来确定的。

Go的函数签名func (InputTypeList)OutPutTypeList

在Go中， 有名函数和匿名函数， 只要函数签名就可以相互赋值

// 定义一个结果返回的函数类型， 参数为convert函数参数

```

type FResult func(convert func(s interface{}) string) interface{}

```

// handle函数返回值为FResult的函数类型

```

func handle(origin interface{}) FResult {
    // 将匿名函数赋值给FResult， 匿名函数与FResult函数类型的函数签名是一样的
    return func(convert func(s interface{}) string) interface{} {
        return convert(origin)
    }
}

```

函数方法

在Go中有一种函数的用法是很多语言所不具备的， 那就是给函数类型添加方法和实现接口

```
// 定一个有名函数类型HandlerFunc
type HandlerFunc func(ResponseWriter, *Request)

// 为有名函数类型HandlerFunc添加方法
func (f HandlerFunc) ServeHTTP(w ResponseWriter, r *Request) {
    f(w,r)
}
```

函数在程序中，是一种功能复用的技法，每个函数代表着一种功能的实现，在**Go**中，函数也是一种型，和基础类型一样，可以作为参数和返回值。