

包 (package)

init函数

我们在test路径下创建了一个calc的文件夹，里面是自己定义的包

test	D:\Go\src\test	28
calc		29
calc.go		30
hello.go		31

```
调用包

1 package main
2
3 import (
4     "fmt"
5     "test/calc"
6 )
7
8 func main() {
9     a := calc.Add(1, 2)
10    fmt.Println(a)
11 }
```

注意:

```
// 包中的标识符(变量名\函数名\结构体\接口等)如果首字母是小写的,表示私有(只能在当前这个包中使用)
// 首字母大写的标识符可以被外部的包调用
```

博客:https://www.liwenzhou.com/posts/Go/11_package/

- 包的路径从 GOPATH/src 后面的路径开始写起,路径分隔符用 /
- 想被别的包调用的标识符都要首字母大写!
- 单行导入和多行导入
- 导入包的时候可以指定别名
- 导入包不想使用包内部的标识符,需要使用匿名导入
- 每个包导入的时候会执行一个名为 init() 的函数,它没有参数也没有返回值也不能手动调用
- 多个包中都定义了 init() 函数,则它们的执行顺序见下图:

init函数

init函数是初始化函数（我感觉有点像C++中的构造函数，但是C++中构造函数写在类的内部，并且是面向对象编程的一种工具）

init函数回自己自动调用

```
1  package main
2
3  import (
4      "fmt"
5      "test/calc"
6  )
7
8  func init() {
9      fmt.Println("init")
10 }
11
12 func main() {
13     a := calc.Add(1, 2)
14     fmt.Println(a)
```

```
> <4 go 设置调用>
init
3
```

主函数中并没有调用，但是编译器会自动调用这个函数。