笔记2

The master has failed more times than the beginner has tried

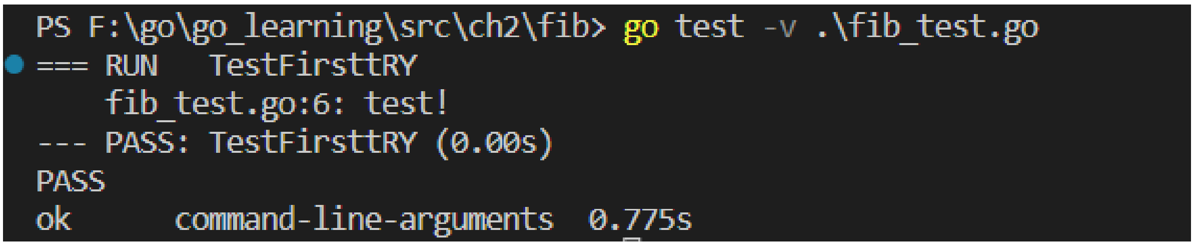
编写测试文件

- 源码文件以 `_test` 结尾：`xxx_test.go`
- 测试方法名以Test 开头：`func TestXXX(t testing.T) {...}`

```
package try_test

import "testing"

func TestFirsttRY(t *testing.T) {
    t.Log("test!")
}
```



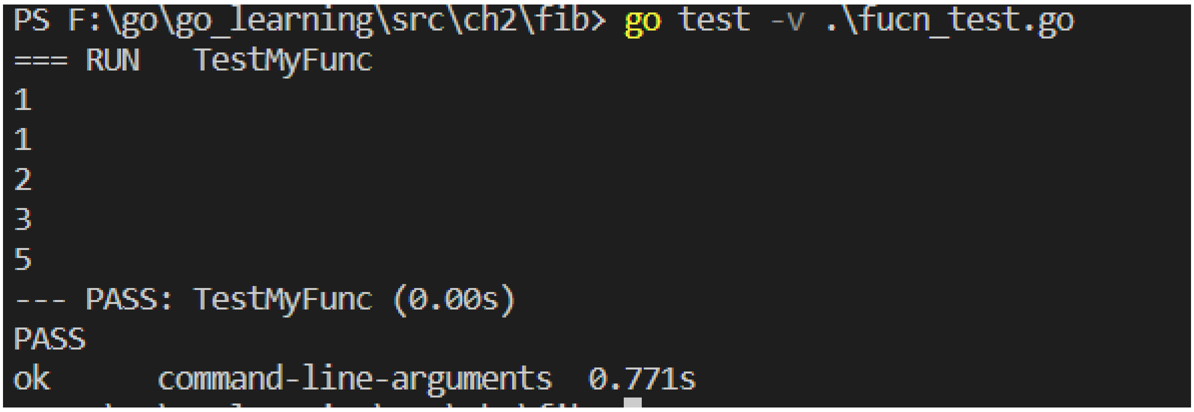
```
PS F:\go\go_learning\src\ch2\fib> go test -v .\fib_test.go
=== RUN   TestFirsttRY
    fib_test.go:6: test!
--- PASS: TestFirsttRY (0.00s)
PASS
ok      command-line-arguments  0.775s
```

```
package my_test

import (
    "fmt"
    "testing"
)

func TestMyFunc(t *testing.T) {

    a := 1
    b := 1
    fmt.Println(a)
    for i := 1; i < 5; i++ {
        fmt.Println(b)
        temp := b
        b = a + b
        a = temp
    }
}
```



```
PS F:\go\go_learning\src\ch2\fib> go test -v .\fucn_test.go
=== RUN   TestMyFunc
1
1
2
3
5
--- PASS: TestMyFunc (0.00s)
PASS
ok      command-line-arguments  0.771s
```

变量赋值

赋值可以进行自动类型推断

在一个赋值语句中可以对多个变量进行同时赋值

`a,b = b,a`

常量

```
const (
    a = 100
    b       //100
    c       //100
    d = "haha"
    e       //"haha"
)
```

`iota[ai'əutə]`

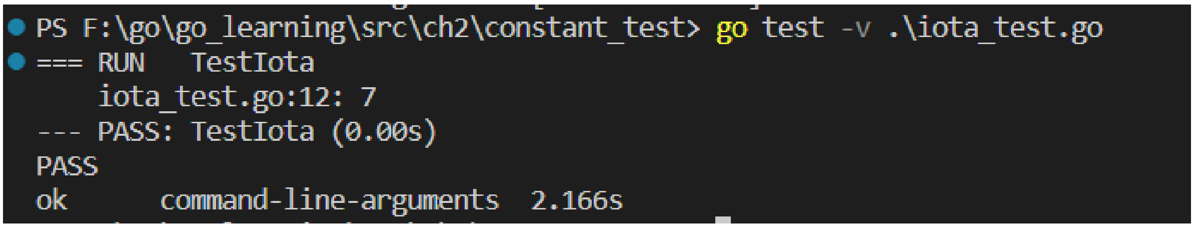
```
package iota_test

import "testing"

const (
    Readable = 1 << iota
    Writable
    Executabe
)

func TestIota(t *testing.T) {
    t.Log(Readable | Writable | Executabe) //

}
```



```
PS F:\go\go_learning\src\ch2\constant_test> go test -v .\iota_test.go
=== RUN   TestIota
    iota_test.go:12: 7
--- PASS: TestIota (0.00s)
PASS
ok      command-line-arguments  2.166s
```