

# Go语言测试流程

## 测试分类:

- 单元测试: API 测试, 集成测试: 灰度测试;

→ 它又称程序员测试。顾名思义, 这就是程序员们本该做的自我检查

- Go 中的单元测试

而示例测试严格来讲也是一种功能测试, 只不过它更关注程序打印出的结果

- 功能测试 (Test)
- 基准测试 (benchmark)
- 示例测试 (example)

如: Test 前缀

如: Testing.T

Example 为前缀

benchmark ; Testing.B

- 实现:

- 每个测试源码文件都必须至少包含一个测试函数。并且, 从语法上讲每个测试的源码文件中, 都可以包含任何一类的测试

- go test:

当我们运行 go test 命令的时候, 其中的测试代码才有可能被执行

go test 命令就会针对每个被测代码包, 依次地进行构建、执行, 输出结果

- 功能测试与结果:

- go test puzzlers/article20/q2  
ok puzzlers/article20/q2 0.0008s

↓  
测试成功

↓  
测试时间

包路径  
→ 第一次

- \$ go test puzzlers/article20/q2 (cached)

→ 第二次

↓  
代码无任何改  
动的情况下

go env GOCACHE命令来查看缓存目录的路径  
go clean -cache 来清除缓存

- 性能测试:

\$ go test -bench=. -run=^\$ puzzlers/article20/q3

goos: darwin

goarch: amd64

pkg: puzzlers/article20/q3

BenchmarkGetPrimes-8 500000 2314 ns/op

PASS

ok puzzlers/article20/q3 1.192s

执行哪些性能测试函数  
(正则表达式)

执行哪些功能函数

= runtime.GOMAXPROCS