



- 单元测试
- UI测试

- 测试用例



You got a new ins  
Jumps only if the  
holding is negativ  
zero.) Otherwise  
the next line.

01	inbox	
02	copyto	0
03	inbox	
04	copyto	1
05	sub	0
06	jump	if negative
07	copyfrom	
08	jump	
09	copyfrom	
10	outbox	
11	jump	

# 测试模式

- TDD（测试驱动开发）

先写测试、代码、测试、重构

- BDD（行为驱动开发）

按照它提供的几个方法，进行一些测试用例，  
如果通过则证明代码可信赖。

# Mocha

```
describe('测试套件',function() {  
  it('测试用例', function() {  
    //断言  
  })  
});
```

# 断言库

- assert     `assert.equal(a, b);`
- expect     `expect(a).to.be.equal(b)`
- should     `a.should.equal(b)`
- chai

...

可以做什么



# 单元测试

页面js测试

node服务器接口测试

# Continuous integration

持续集成

travis CI

# UI测试

- PhantomCSS：截图像素对比
- page-monitor（张云龙）：dom结构对比
- selenium：多浏览器下截图对比
- PhantomJS：用户操作测试、性能监控等

# headless browser

无界面浏览器

phantoms



# 为什么需要自动化测试

- 一个项目最终会经过快速迭代走向以维护为主的状态，在合理的时机以合理的方式引入自动化测试能有效减少人工维护成本。自动化测试的收益可以简单总结为：
- $\text{自动化的收益} = \text{迭代次数} * \text{全手动执行成本} - \text{首次自动化成本} - \text{维护次数} * \text{维护成本}$
- 对于自动化测试来说，相对于发现未知的问题，更倾向于避免可能的问题。