# CasperJS是什么？



*Navigation scripting & testing for PhantomJS and SlimerJS*

1、对PhantomJS或SlimerJS的封装，作浏览或爬虫使用；

2、用于写测试。

无头浏览器headless browser：有浏览器的所有功能，除了不展示可视化界面。

浏览器工作原理：

解析DOM > CSS渲染 > JS执行 > CSS渲染……

浏览器内核组成：

排版引擎

Webkit(Safari)

Gecko(Firefox)

Trident(ie)

Edge

Presto

Blink(Chrome)

JS引擎

Chakra(IE9\edge)

[JaegerMonkey](https://baike.baidu.com/item/JaegerMonkey/10701276) (Firefox)

V8(Chrome\jdk8)

Rhino(java)

内核 无头浏览器（可脚本、可截图） 封装简易化、测试工具方法

Webkit -> PhantomJS

> CasperJS

Gecko -> SlimerJS

# 研究方向

1. 基于PhantomJS的CasperJS安装，写个基本的爬虫程序。
2. 研究CasperJS的Event和Filter，看如何做性能测试。
3. 结合业务需求，从CasperJS和PhantomJS的使用细节上设法优化，目标是能支持50个程序同时运行。

分享一

# 安装步骤

*PhantomJS 1.9.1 or greater. Please read the installation instructions for PhantomJS*

*Python 2.6 or greater for casperjs in the bin/ directory*

Windows

<http://casperjs.org/>下载zip包，解压zip包到某个目录，将此目录/bin写到环境变量。

<http://phantomjs.org/download.html> 下载windows版zip包，解压zip包到某个目录，将此目录/bin写到环境变量。

Python？我没有特意安装

CentOS

<http://casperjs.org/>下载zip包，解压zip包到某个目录。

<http://phantomjs.org/download.html> 下载linux版tar包，解压zip包到某个目录。

写环境变量，可参考：

cd /etc/profile.d/

vim phantom\_and\_casper\_env.sh

i

export PATH=$PATH:/opt/casperjs-1.1.4-1/bin:/opt/phantomjs-2.1.1-linux-x86\_64/bin

:x

cd ..

source profile.d

# 使用Demo

// casperhello.js

// casperjs casperhello.js

phantom.outputEncoding="GBK"; // windows中文编码，若是linux则注释掉

var casper = require('casper').create(); // casper对象，自定义配置

casper.start('http://www.baidu.com/'); // 第一个浏览step

casper.then(function() {

this.echo('First Page: ' + this.getTitle());

});

casper.thenOpen('http://www.msxf.com', function() { // 第二个浏览step

this.echo('Second Page: ' + this.getTitle());

});

casper.run(); // 运行结果



// caspertest.js

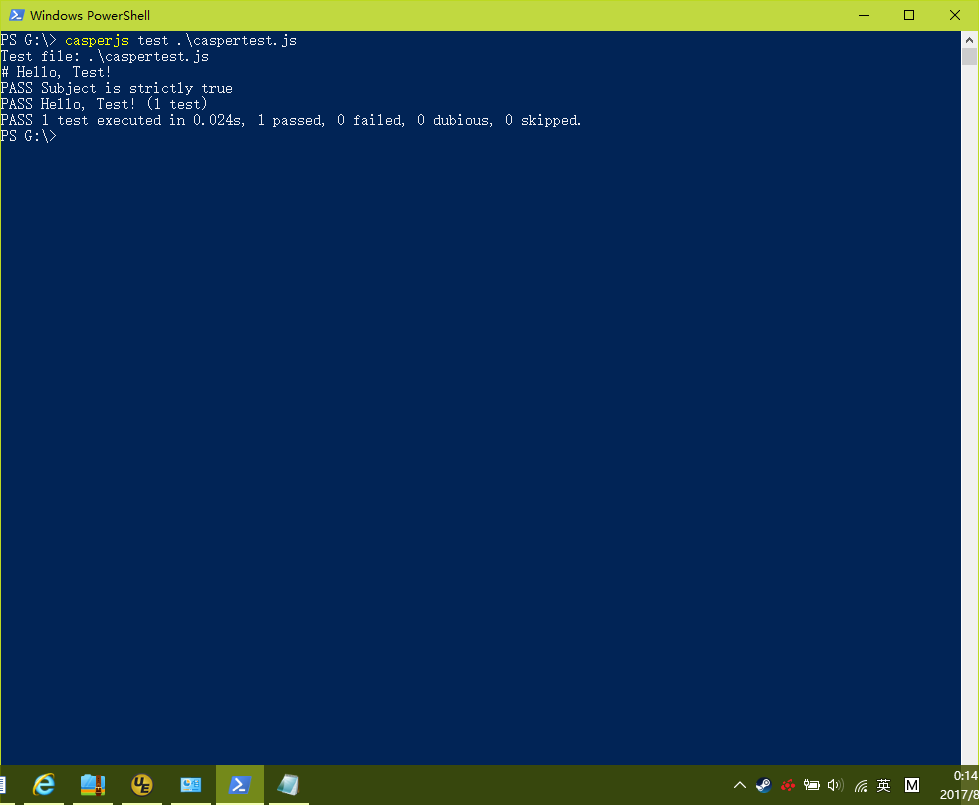
// casperjs test caspertest.js

casper.test.begin('Hello, Test!', 1, function(test) {

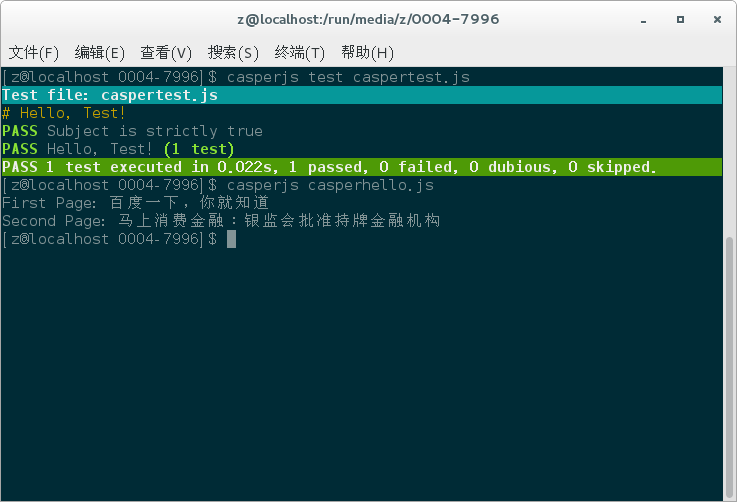
test.assert(true);

test.done();

}); // 运行结果



linux下的运行结果



# 自动录制casperjs的chrome插件

<https://github.com/casperjs/resurrectio>

针对test功能，暂不支持鼠标动作，对浏览器内容变化不够敏感。

分享二

phantomJs：浏览器、文件系统、服务器

casperJs：执行步骤定义、丰富的waitFor

# 只能ie浏览的网页

举例，交通银行 <https://pbank.95559.com.cn/personbank/logon.jsp>



程序：src/bcm.js



ActiveX控件检测原理

isIE:function(){

if(!!window.ActiveXObject || "ActiveXObject" in window){

return true;

}else{

return false;

}

},

可以打开，和chrome的展示效果相同。

# 日志文件写入

无log文件配置，可实现文件写入

程序：[src](https://github.com/BasZhang/headless-learning-plan/tree/master/src)/log-file.js

# casperjs是否自动结束

casper.run方法有个onComplete参数，默认是退出phantomjs，另外参考参数 exitOnError

# 跨进程外部动态控制

Webserver传参，但是webserver性能不行。

# 只开1个浏览器，实现同时执行10段js代码，是否像浏览器一样待机可供后续使用

当时说的是。我感觉可以，但先驱者没有研究出来

目前结论：？

# 在执行js时，如果中途把js改了，或者删掉，看有什么效果

程序：[src](https://github.com/BasZhang/headless-learning-plan/tree/master/src)/injectjs.js

加载后无影响

# 执行完了，是否自动释放资源

程序：[src](https://github.com/BasZhang/headless-learning-plan/tree/master/src)/loop.js

Cpu和内存未见起伏

Webserver的情况

程序：src/ws-wrong.js

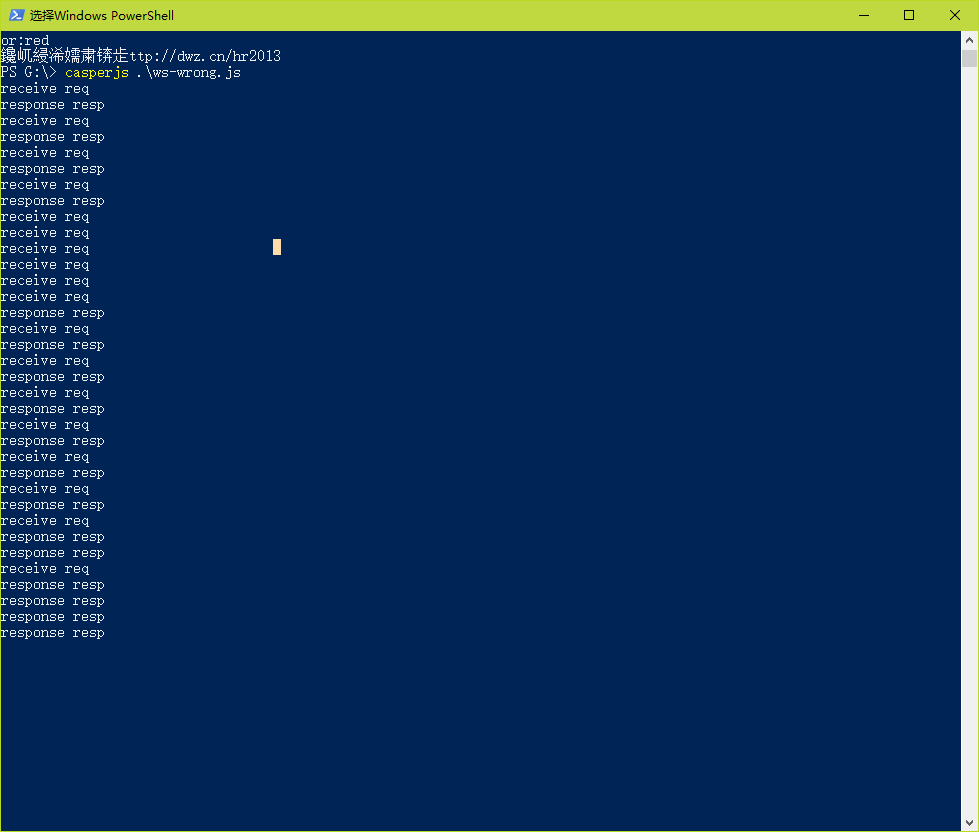
监听127.0.0.1：9999，开4个浏览器发请求，表现差，最差时如图：

16g内存占用20.7%，cpu跑满一个核心

同时还觉得，我的程序的写法，新的req对老的resp有影响



开3个chrome网页，发现有18个请求；开eclipse浏览器，请求2次，接到16个请求，直接把服务器轰掉了，之后cpu和内存非空闲状态



# waitFor到底在wait什么

Casper.prototype.waitFor = function waitFor(testFx, then, onTimeout, timeout, details) {}

定义了一个step，每this.options.retryTimeout（20）检查一次条件

# 反爬虫

针对phantomJs的检测

https://github.com/ikarienator/phantomjs\_hide\_and\_seek