



# SELENIUM

## SELENIUM基础

刘挺立

Email: Liu.Tingli@outlook.com

QQ: 1819181820





## 刘挺立

---

- 8年软件开发和自动化测试经验
- 擅长敏捷测试项目管理
- 曾长期供职于外企，担任高级软件开发、数据挖掘以及测试开发、自动化测试测试主管等职位



# 学习目标

通过本课学习，您将拥有以下收获：

1. 软件测试的发展与自动化测试的场景
2. Selenium的家谱版本变迁
3. Selenium的学习重点和编程语言选择



**PART 1**  
自动化测试的基础

**PART 2**  
什么是Selenium

**CONTENTS**  
**目录**

**PART 3**  
Selenium的家谱

**PART 4**  
编程语言的选择



# 1

## PART ONE

### 自动化测试的基础

这一章我们重点讲解自动测试的场景描述以及  
自动化测试应该具备的基本技术







# 软件测试发展

自动化测试的由来





# 软件测试发展

- 程序测试是为了发现错误而执行的过程。
  - 源自G.J.Myers在其经典的著作《软件测试艺术》（ The Art of Software Testing ）



# 软件测试发展

- 软件测试的时间点
- 在编码之后？
- 在编码之前？



# 软件测试发展

- 动态测试 VS 静态测试



# 软件测试发展

- 动态测试
  - 必须运行程序，在编码之后。
  - 在代码开发出来以后，通过执行程序，像用户那样操作软件发现问题



# 软件测试发展

- 静态测试
  - 需求评审、设计评审、代码扫描、代码评审等



# 软件测试发展

- 敏捷测试
  - 对软件产品质量的持续评估
  - 软件测试贯穿到整个软件开发生命周期的全过程中，不再停留在编程之后的某个阶段。
  - 互联网行业和敏捷测试的要求
  - 持续测试和持续集成



# 软件测试发展

- 持续集成
  - Continuous Integration , 简称CI。
  - 是一种软件工程实践：项目需要经常性的集成，进行代码编译，版本构建，自动化的部署以及部署后的自动化测试。也是云开发测试模式的一种。



# 软件测试发展

- 持续集成构成了持续交付的基础。
- 自动化测试：
  - 以人为驱动测试行为转化为机器执行的一种过程





# 自动化测试场景

自动化测试应用的场景以及自动化测试的条件等



# 自动化测试的场景

- 功能测试自动化
  - 测试界面显示、测试业务逻辑、测试数据导入导出、测试接口功能
  - 冒烟测试
  - 回归测试



# 自动化测试的场景

- 性能测试的自动化：
  - 测试压力和负载、测试接口的性能等



# 自动化测试的场景

- 什么样的项目适合自动化测试？



# 自动化测试场景

1. 任务测试明确，不会频繁变动
2. 每日构建后的测试验证
3. 比较频繁的回归测试
4. 软件系统界面稳定，变动少
5. 需要在多平台上运行的相同测试案例、组合遍历型的测试、大量的重复任务
6. 软件维护周期长
7. 项目进度压力不太大
8. 被测软件系统开发比较规范，能够保证系统的可测试性
9. 测试人员具备较强的编程能力



# 自动化测试工具

自动化测试的工具以及需要掌握的技术



# 自动化测试工具

- HP QTP (UFT) :
- Selenium : 自动化Web UI 测试 python Java C# JavaScript php ruby
- Appium : 自动化App UI 测试
- HP LoadRunner : 性能测试 ( JMeter )
- Robot Framework
- ReadyAPI (SoapUI) : 接口测试工具 ( webservice , soap , restfulAPI )
- MonkeyRunner



# 自动化测试技术

- Web前端测试技术
- Web前端工具
- 自动化测试编程语言
  - Python
  - Java





# WEB前端技术

基础技术	高阶技术
HTTP协议：报文、请求、Cookie等	Ajax
Html和CSS	Html5+CSS3
JavaScript语言和框架（其中jQuery是基础框架）	Bootstrap框架
JSON和XML	AngularJS



# WEB前端工具

- Firebug
- <https://addons.mozilla.org/en-US/firefox/addon/firebug>
  - 火狐浏览器的插件



# 自动化测试技术

- 自动化测试编程语言
  - Python
  - Ruby
  - Groovy
  - Java
  - JavaScript
  - VBScript
  - C





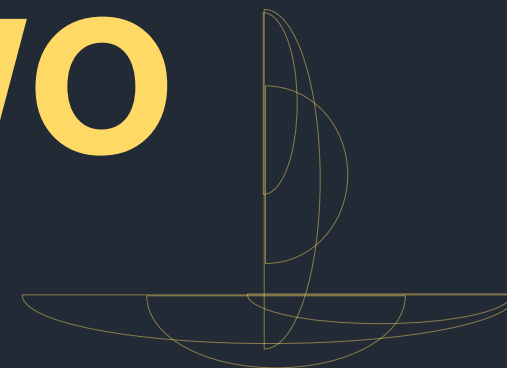
# 章节小结



# 2 PART TWO

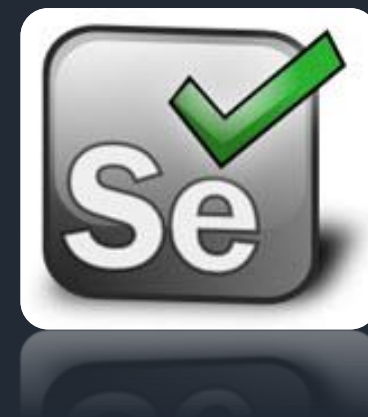
## 什么是Selenium

这里开始我们将会一步步接触到Selenium的世界，为自动化测试开启一段新的旅程



# 什么是SELENIUM

- Selenium
- /sə'liniəm/
  - 英文原意：硒
  - ThoughtWorks公司的Jason Huggins发布
  - Selenium官方网站：<http://docs.seleniumhq.org>



# 什么是SELENIUM

- Selenium 是用于测试 Web 应用程序用户界面 (UI) 的常用框架。
- Selenium is a suite of tools to automate web browsers across many platforms.



# 什么是 SELENIUM

- Jason Huggins
- 2006
- ThoughtWorks
- Google
- <http://seleniumhq.org>





# 什么是SELENIUM

- 它是一款用于运行端到端功能测试的超强工具。您可以使用多个编程语言编写测试，并且 Selenium 能够在一个或多个浏览器中执行这些测试。
- 强大的
- 跨平台的
- Web功能自动化测试系列工具





身处ThoughtWorks公司的Jason Huggins开发了一个JavaScript的类库来驱动浏览器页面的行为

## 2004年 Selenium1.0

## WebDriver 2006年

Google的一位有胆识的年轻人Simon Stewart发起了一个叫WebDriver的项目

## Selenium2.0 2008年

Selenium和webdriver结合了，  
**Selenium2.0 = Selenium1.0 + WebDriver**



# 什么是SELENIUM

- Selenium 1 和 WebDriver 合并成一款性能更佳的产品 Selenium 2（或 Selenium WebDriver），该款产品发行于 2011 年。
  - Selenium 2 具有来自 WebDriver 的清晰面向对象 API，并能以最佳的方式与浏览器进行交互。Selenium 2 不使用 JavaScript 沙盒，它支持多种浏览器和多语言绑定。



# 为什么选择SELENIUM

- 开源，免费
- 多浏览器支持：Firefox、Chrome、IE、Opera、Safari
- 多平台支持：Linux、windows、Mac
- 多语言支持：Java、python、ruby、PHP、C#、JavaScript





# 章节小结



# 3

## PART THREE

### Selenium的家谱

这一节我们将会了解Selenium的变迁以及方方面面。为以后的使用打下基础。



# SELENIUM的家谱

这里我们重点探讨Selenium1.0到Selenium2.0的变迁过程



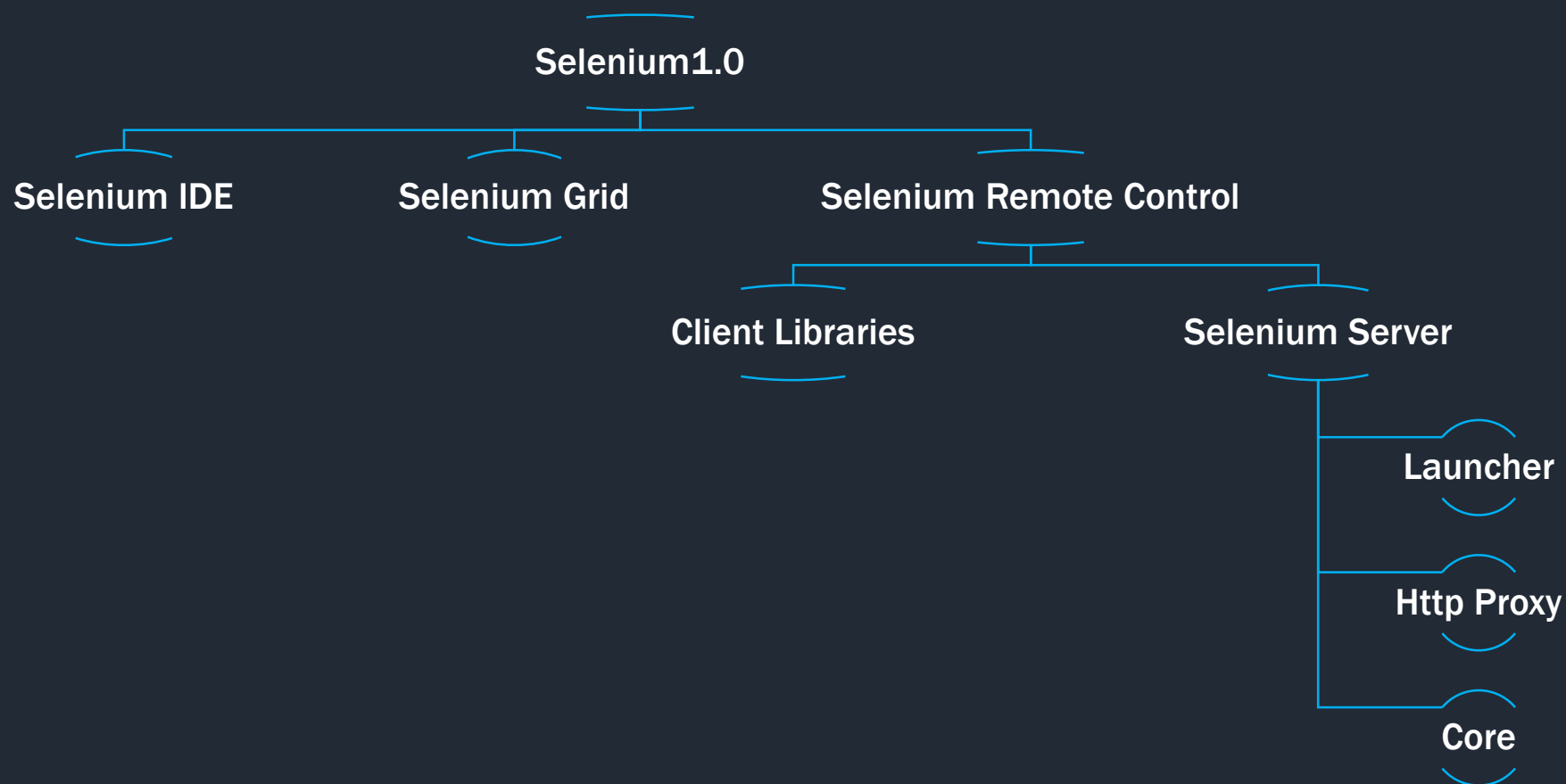
# SELENIUM的家谱

- Selenium经历了两个版本，Selenium1.0 和Selenium2.0，Selenium 也不是简单一个工具，而是由几个工具组成，每个工具都有其特点和应用场景。





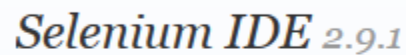
# SELENIUM1.0



# SELENIUM IDE

- Selenium IDE 是嵌入到Firefox 浏览器中的一个插件，实现简单的浏览器操作的录制与回放功能。
- IDE 录制的脚本可以转换成多种语言，从而帮助我们快速的开发脚本
  - Selenium IDE 下载地址：
  - <https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/>



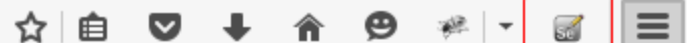
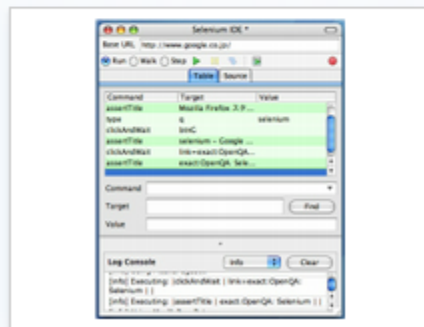


by Jason Huggins, Adam Goucher, Shinya Kasatani, Dave Hunt, Samit Badle

~~Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests.~~

**+ Add to Firefox**

This add-on has been preliminarily reviewed by Mozilla. [Learn more](#)



[Register](#) or [Log in](#)

Other Applications ▼

 search for add-ons

## WEB DEVELOPER

~~Toggle Tools~~

Inspector

Ctrl+Shift+C

## Web Console

Ctrl+Shift+K

## Debugger

Ctrl+Shift+S

## Style Editor

Shift+F7

## Performance

Shift+F5

## Network

Ctrl+Shift+Q

## Developer Toolbar

Shift+F2

WebIDE

Shift+F8

## Browser Console

Ctrl+Shift+J

### Responsive Design View

Ctrl+Shift+M

Eyedropper

Scratchpad

Shift+F4

Page Source

Ctrl+U

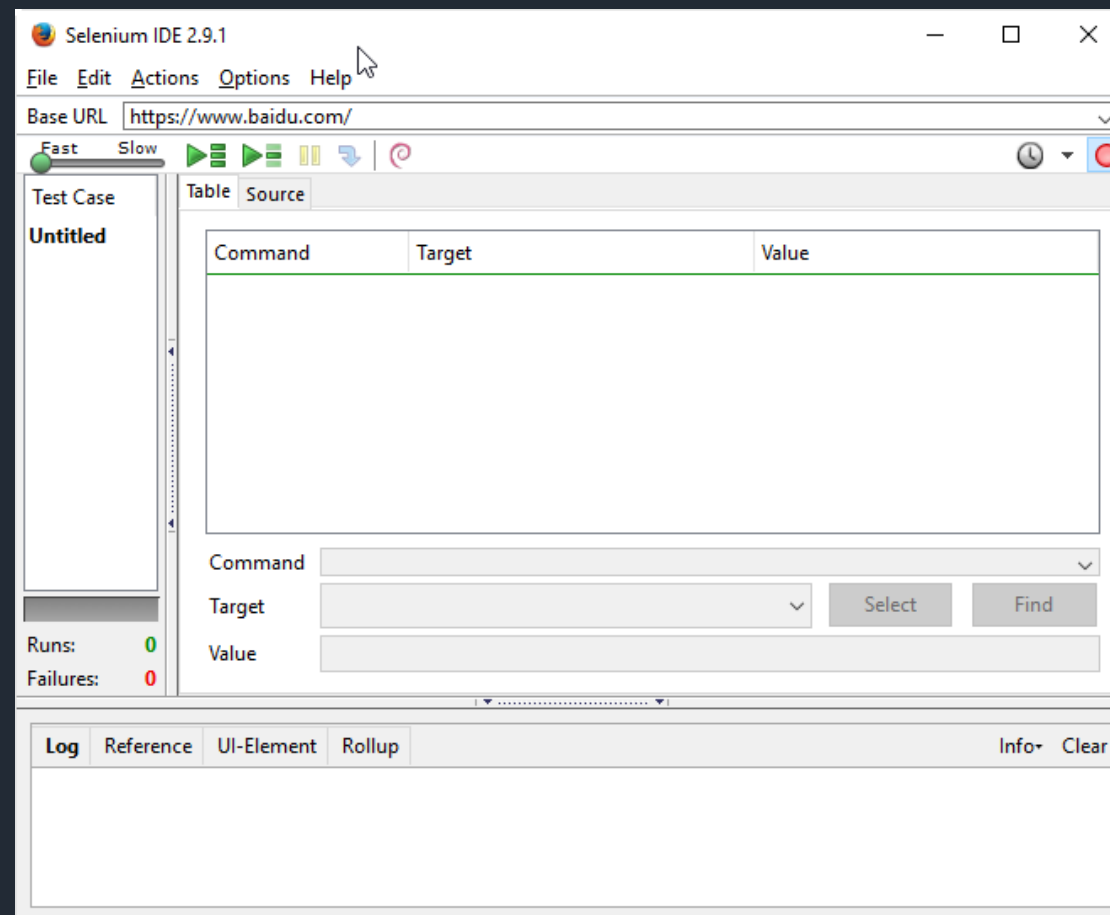
[Get More Tools](#)

 Selenium IDE

Alt+Ctrl+S

### Work Offline

# SELENIUM IDE



# SELENIUM IDE

- 编辑一行命令或者注释
- 插入一行注释
- 插入一行命令
- 导出脚本



# SELENIUM GRID

- **Selenium Grid** 是一种自动化的测试辅助工具，可以很方便地同时多台机器上和异构环境中并行运行多个测试事例。
  - 并行执行
  - 通过一个主机统一控制用例在不同环境、不同浏览器下运行。
  - 灵活添加变动测试机



# SELENIUM REMOTE CONTROL

- Selenium RC ( Remote Control ) 是Selenium1.0 家族的核心工具，  
Selenium RC 支持多种不同的语言编写自动化测试脚本，通过selenium RC  
的服务器作为代理服务器去访问应用从而达到测试的目的。



# SELENIUM2.0

这里我们将会进入面向未来和面向主流的Selenium2.0





# SELENIUM2.0

- Selenium2.0 = Selenium1.0 + WebDriver



# SELENIUM2.0

- 其实在Selenium2.0中主推的就是WebDriver这个模块。WebDriver是Selenium RC的替代品。
  - 因为Selenium需要保留向下兼容性的原因，Selenium RC才没有被彻底的抛弃，如果使用Selenium开发一个新的自动化测试项目，那么我们强烈推荐使用Selenium2.0的WebDriver进行编码。



# SELENIUM1.0的弱点

- Selenium RC的实质
- 在浏览器中运行一个JavaScript应用程序，使用每个浏览器内置的JavaScript解释器来解释和执行Selenium的命令集合。



# SELENIUM2.0的优点

- WebDriver针对不同的浏览器进行开发，取代了RC那样嵌入到被测试的WebApp程序中的JavaScript，使用了与浏览器更加集成的方式进行测试，避免了RC使用JavaScript而可能遇到的安全模型限制导致的问题等。



# SELENIUM2.0

- Selenium1.0代表的是经典，Selenium2.0代表的是未来。





# 章节小结



# 4

## PART FOUR

### 编程语言的选择

All master artwork can be obtained from the marketing department



# 编程语言的选择

- 我们在此重点关注Selenium2.0的WebDriver。而WebDriver面临的第一个选择就是编程语言。
- WebDriver支持众多的编程语言，例如，C#、Ruby、Python、Java、Javascript、Objective-C、PHP。
- 我们在使用之前必须要确定一种。





# PYTHON

- Python
- 我们选择的是Python
  - /'paɪθən/
  - 优雅
  - 明确
  - 简单



# PYTHON 编程工具

- Sublime Text



# PYTHON 编程工具

- PyCharm



# 选择PYTHON的理由

- 测试人员的编程能力普遍不是很强，而Python作为一种脚本语言，不仅功能强大，而且语法优美，支持多种自动化测试工具，而且学习上手比较容易。



# 选择PYTHON的理由

- Python的社区发展比较好，有着非常多的文档和支持库，另外Python也可以在Web开发、数据处理、科学计算等纵多领域有着非常好的应用前景。



# 选择PYTHON的理由

- 对于有一定编程基础的人员，使用Python作为自动化测试的语言可以非常顺畅的转换，几乎没有学习成本。
  - 同时Python是标准的面向对象的编程语言，对于C#、Java等面向对象的语言有着非常好的示例作用，通过Python的示例可以非常轻松的触类旁通，使用其他语言进行Selenium2.0的WebDriver的使用。



# PYTHON

- /'paɪθən/
- Python的创始人为Guido Van Rossum。1989年圣诞节期间，在阿姆斯特丹，Guido为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，做为ABC语言的一种继承。之所以选中Python（大蟒蛇的意思）作为程序的名字，是因为他是一个叫Monty Python的喜剧团体的爱好者。



# PYTHON

- Python 语言除了在自动化测试领域有出色的表现外，在系统编程，网络编程，web 开发，GUI开发，科学计算，游戏开发等多个领域应用非常广泛，而且具有非常良好的社区支持。也就是说学习和掌握python 编程，其实是为你打开了一道更广阔的大门。





脚本文件一般用 .py 后缀

coffeeghost-q-in-py.py x

中文用户一定得先用这行来声明编码,同时文件本身也得存储成UTF-8编码!

单行注释

导入其它代码模块

注意!Python最好也最个性的语法:  
使用缩进来代替其它语句块声明;  
一般建议每个层级用4个空格来缩进。

变量得先实例化  
才可进一步计算

单行的语句块,其实可以不换行的,  
但是,建议清晰起见,规范点:  
- 另起一行  
- 缩进一级

函数声明,  
注意使用冒号结束声明

多行注释的内容不用遵守当前缩进  
只要开始的''' 缩进正确就成!

每级语法块不用}之类的括号引领!  
直接回车+4空格  
(当然,要在当前缩进基础上)

```
1 # -*- coding: utf-8 -*-
2 # Quick Python Script Explanation for Progeammers
3 # 给程序员们的超快速Py脚本解说
4 import os
5
6 def main():
7     print 'Hello World!'
8     print "这是Alice\的问候."
9     print '这是Bob\的问候.'
10
11     foo(5, 10)
12
13     print '=' * 10
14     print '这将直接执行'+os.getcwd()
15
16     counter = 0
17     counter += 1
18
19     food = ['苹果', '杏子', '李子', '梨']
20     for i in food:
21         print '俺就爱整只:'+i
22
23     print '数到10'
24     for i in range(10):
25         print i
26
27 def foo(param1, secondParam):
28     res = param1+secondParam
29     print '%s 加 %s 等于 %s'%(param1, secondParam, res)
30     if res < 50:
31         print '这个'
32     elif (res>=50) and ((param1==42) or (secondParam==24)):
33         print '那个'
34     else:
35         print '嗯...'
36     return res
37
38 '''这是多
39 行注释.....'''
40
41 if __name__ == '__main__':
42     main()
```

模块名,其实导入了 os.py

函数名"main"在这儿并不是必须的,调用在这段脚本的最后部分;

声明单行字符串,使用双/单引号都成,  
注意对字符串中的引号进行逃逸处理!

函数调用,声明在后述代码;

字符串可乘,等于:'====='

调用了os 模块中的函数

连接字符串

内置的列表类型对象,其实可以包含不同类型数据,  
甚至可以包含其它列表对象;

在循环中,i 指代了列表中按顺序的每个"food"

range()内置函数,返回类似  
[0,1,2,3,4,5,6,7,8,9]  
的数字列表,注意 for in 循环语句使用冒号结束声明!

字符串的格式化输出基本类似C语言的

判定式也基本和C语言的相同

逻辑运算符,不使用 && 和 ||,使用直观的E文单词

这都是合法注释

一般在脚本最后调用主函数 main();而且使用 内置的运行脚本名来判定;  
当且仅当我们直接运行当前脚本时, \_\_name\_\_ 才为 \_\_main\_\_  
这样当脚本被当作模块进行 import 导入时,并不运行 main()  
所以,一般这里是进行测试代码安置的...

在M\$中很好的支持UTF-8的编辑器不多,  
跨平台又支持Py特性的更少;  
好在我们有 Limodou 贡献的 UliPad  
这一编辑器本身就是Py实现的!  
(基于wxPython)

关于

UliPad 4.0

作者: Limodou (limodou@gmail.com)

如果你有任何问题请与我联系。

[The UliPad project homepage](#)

[The UliPad maillist](#)

[The UliPad Snippets Site](#)

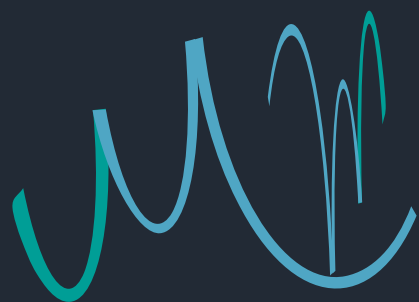
[My Blog](#)

[Contact me](#)

确定

用冒号来结束判断句,  
在 if elif else 行最后





# 课程总结



# THANKS!

刘挺立

Email: Liu.Tingli@outlook.com

QQ: 1819181820

