# 环境搭建

## python2.7.13、selenium (2.53.6)、 firefox46.0.1

小编环境windows10 64位

|  |  |
| --- | --- |
| python 2.7.12 | C:\Python27\; C:\Python27\Scripts; 加到环境变量 |
| selenium 2.53.6 | Cmd里pip install selenium==2.53.6 |
| firefox 44 |  |

Python官网<https://www.python.org/>

selenium (2.53.6) 、python2.7.13、 firefox46.0.1 完美

# 有问题看API （pydoc）

pydoc? ,这个是准确的解释：Documentation generator and online help system.

启动server

  1.打开cmd命令行，输入：python -m pydoc -p 6666

  2.-p参数：这个表示在本机上启动服务

  3.6666参数：这个是服务端口号，随意设置

打开后，界面会出现一个地址：http://localhost:6666/,在浏览器直接打开，Built-in Moudles ：这个是python自带的模块

webdriver API

 1.找到这个路径：python2.7\lib\site-packages，点开selenium

 2.打开的

selenium>webdriver>firefox>webdriver,最终路径：http://localhost:6666/selenium.webdriver.firefox.webdriver.html

 3.最终看到的这些就是selenium的webdriver API帮助文档啦

## 安装pycharm

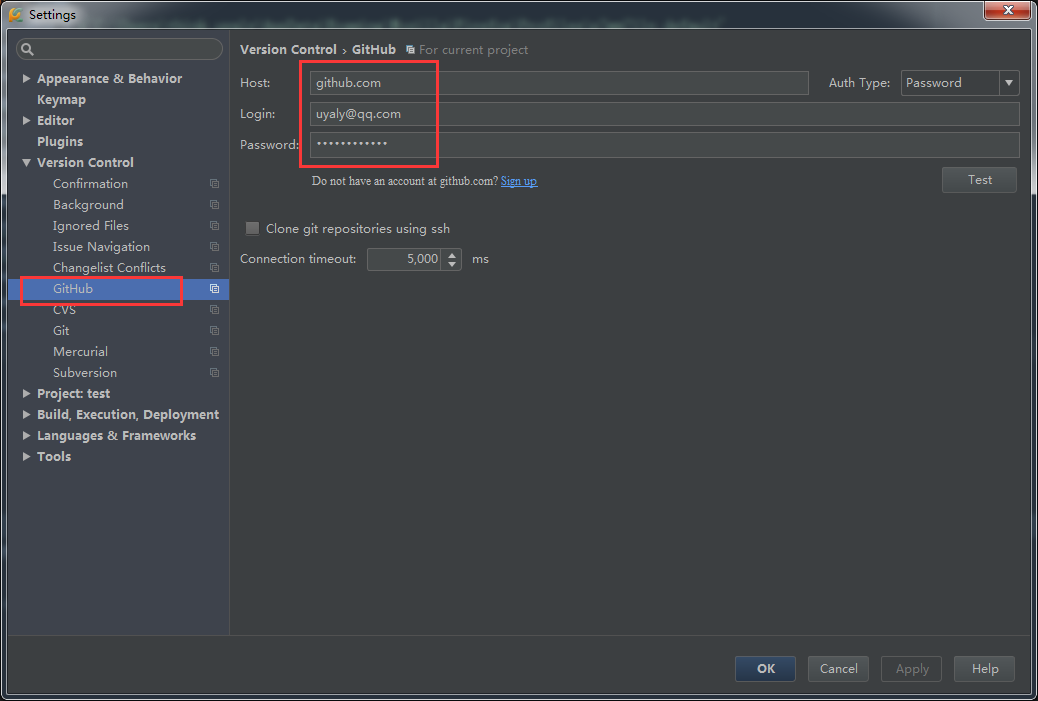
pycharm-community-5.0.1.exe

## 安装Git，Pycharm连接Github

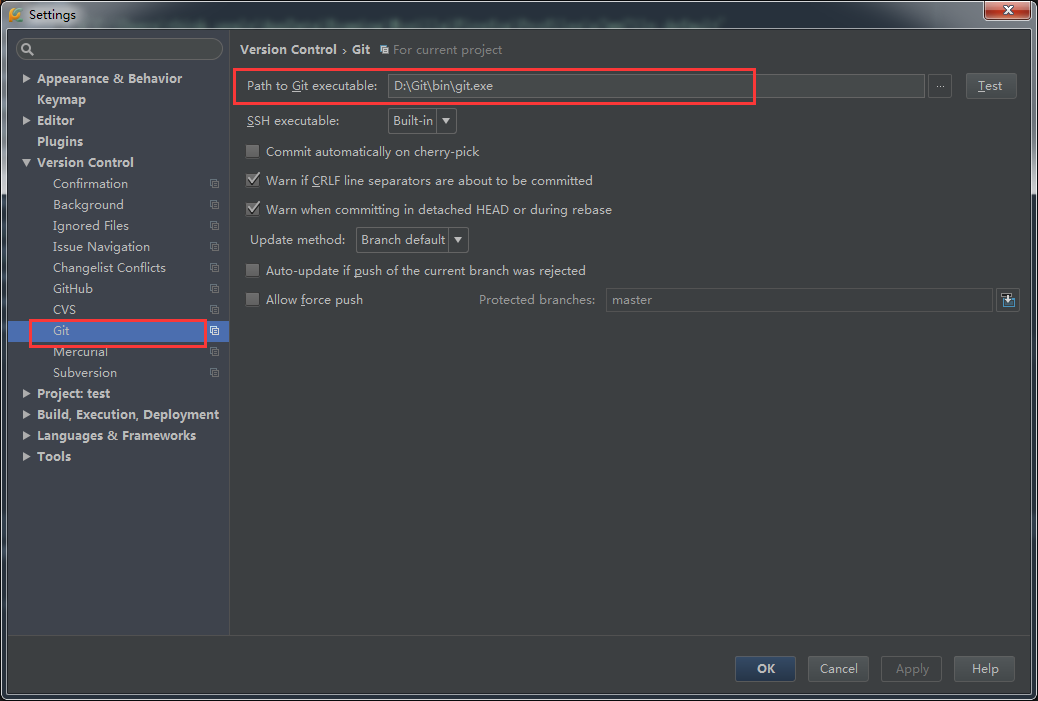
**绑定账号**

1、File->Settings->Version Control->Github

会出现github，然后在旁边输入你github的用户名和密码，可以点击”test”测试一下，如果出现：**Connection successful**,则代表连接成功了。

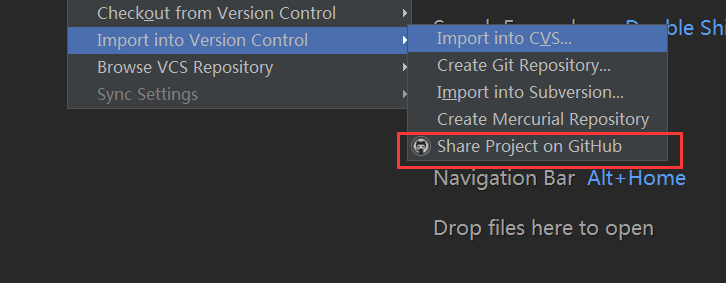


2、选择Git安装路径  
设置github后选择git，输入你git.exe的安装位置路径,下面是我的git.exe的位置：**D:\Git\bin\git.exe**,然后点击**OK**。

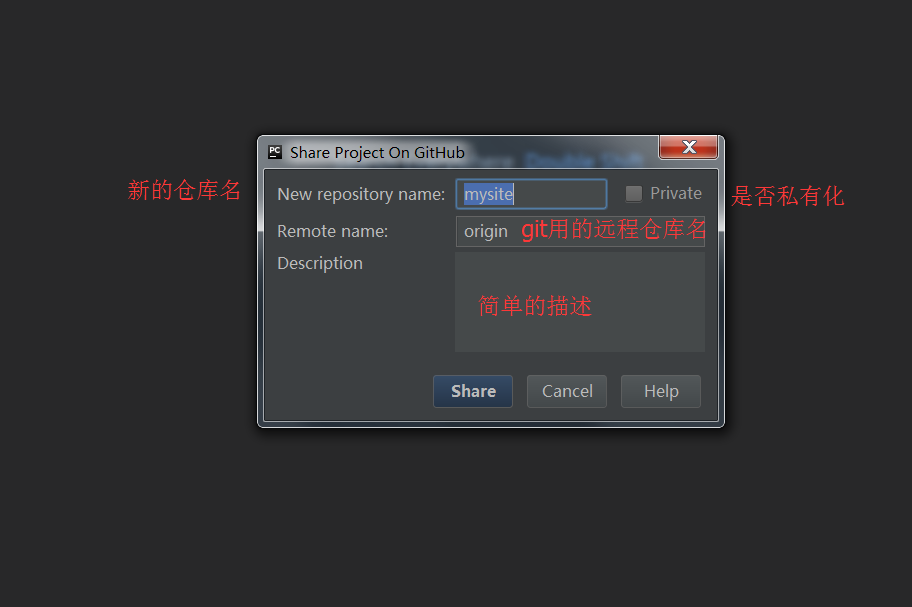


**创建Github的仓库**

1、VCS->Import Into Version Control->Share Project On Github



然后会弹出框让你输入一个仓库名（不能为中文）



然后点击share

然后会弹出让你选择哪些文件需要被同步，选好后，在下面的commit Message可以输入自己的信息，然后点OK，你的代码就提交到网上了。可以看看

**修改后的文件提交**

1、对修改后的文件或目录点击右键：Git—>Add，先把文件放入暂存区。

2、对修改后的文件或目录点击右键：Git—>Commit File，把刚才add放在暂缓区的文件提交。

3、填好commit的信息后，点击下面的commit and Push，可以直接提交到网上。

## Ff浏览器，fire path

安装插件Selenium Builder

工具-Web开发者-Launch Selenium Builder

输入测试web的url，点击record，哪里不会点哪里

## 导入HTMLTestRunner

*记录自3.5*

下载地址：<http://tungwaiyip.info/software/HTMLTestRunner.html>

2.Download下HTMLTestRunner.py文件就是我们需要下载的包。

3.下载后手动拖到python安装文件的Lib目录下

备注：（以下是python2.7的HTMLTestRunner，python3.x的HTMLTestRunner需要自己稍做修改，可以在这里下载：http://pan.baidu.com/s/1hs5OXNY）

## 安装xlrd模块（参数化，读取xls）

打开cmd，输入pip install xlrd在线安装

>>pip install xlrd

## 安装ddt模块（数据驱动）

打开cmd输入pip install ddt在线安装

>>pip install ddt

二、数据驱动原理

1.测试数据为多个字典的list类型

2.测试类前加修饰@ddt.ddt

3.case前加修饰@ddt.data()

4.运行后用例会自动加载成三个单独的用例

## 抓包工具fiddler抓包

# 各种浏览器下的页面元素xpath获取方法

工具 X Path for free！

下载地址：

<https://chrome.google.com/webstore/detail/lbghbpofdlcecfbpjgmffnkieenjkboi>

在chrome中安装好这个扩展后，可以到用户的目录下查看源码。

此工具是Chrome的插件，下载安装后，Chrome右上角会出现1个标记，点击它，然后用鼠标点选要定位的元素，右下角Element的xpath就 生成了，可以随便拷贝。

Fire Fox下有没有这样的插件呢？当然也有！

那就是

<https://addons.mozilla.org/en-US/firefox/addon/firepath/>

和

<https://addons.mozilla.org/en-US/firefox/addon/xpath-checker/>

具体的使用有同学写了，请看这篇文章：

<http://blog.sina.com.cn/s/blog_5aefba9a0100csy8.html>

IE下呢？当然也有解。不过貌似比较悲惨。得用一个比较复杂的Trick

文章在这里

<http://blog.sina.com.cn/s/blog_6067dbcb0100ndi4.html>

-----------------------------------------

update 2012.2.14

给出 xpath 教程链接：<http://www.w3school.com.cn/xpath/>

----------------------------------------------------

# \*\*\*\*\*docstring

# 编译成exe文件运行

用py2exe转成exe文件，可脱离py环境执行

# 发短信

pip install twilio

pip uninstall twilio

from twilio.rest import TwilioRestClient

client = TwilioRestClient(ACCOUNT\_SID, AUTH\_TOKEN)

recipient = '+86158\*\*\*\*\*\*\*\*' # 接收短信的手机

text = '短信内容'

# 这里的from\_参数是一个手机号, 网站免费提供给你的

client.messages.create(

to=recipient,

from\_='+15852864161',

body=body

)

您的试用帐户还有15.50美元。

在这些国家，试用帐户只能向验证过的号码发送信息。

发送消息将开始审判“派从Twilio试用帐号”

当你有一个试用帐号，你仅限于一个Twilio数量

（336）803-7446

不喜欢这个吗？搜索不同的号码

这个美国电话号码具有以下功能：

这个号码可以接收来电和拨打电话。

SMS：这个号码可以发送和接收手机号码的短信。

彩信：这个号码可以发送和接收多媒体信息，从手机号码。

你的新电话号码是+ 13368037446

帮助建立您的Twilio的应用程序，查看资源在开始页。

一旦构建了应用程序，就可以配置这个电话号码来发送和接收呼叫和消息。

# ===========================

# 笔记

## 定位方式

1.id定位：find\_element\_by\_id(self, id\_)

2.name定位：find\_element\_by\_name(self, name)

3.class定位：find\_element\_by\_class\_name(self, name)

4.tag定位：find\_element\_by\_tag\_name(self, name)

5.link定位：find\_element\_by\_link\_text(self, link\_text)

6.partial\_link定位find\_element\_by\_partial\_link\_text(self, link\_text)

7.xpath定位：find\_element\_by\_xpath(self, xpath)

8.css定位：find\_element\_by\_css\_selector(self, css\_selector）

## select下拉框

*记录自2.20*

select\_by\_index()  :通过索引定位

select\_by\_value()  :通过value值定位

select\_by\_visible\_text() :通过文本值定位

deselect\_all()          ：取消所有选项

deselect\_by\_index()     ：取消对应index选项

deselect\_by\_value()      ：取消对应value选项

deselect\_by\_visible\_text() ：取消对应文本选项

first\_selected\_option()  ：返回第一个选项

all\_selected\_options()   ：返回所有的选项

## Js处理滚动条

*记录自2.21*

--scrollHeight 获取对象的滚动高度。

--scrollLeft 设置或获取位于对象左边界和窗口中目前可见内容的最左端之间的距离。

--scrollTop 设置或获取位于对象最顶端和窗口中可见内容的最顶端之间的距离。

--scrollWidth 获取对象的滚动宽度。

scrollTo函数不存在兼容性问题，直接用这个函数就可以了

#滚动到底部

js = "window.scrollTo(0,document.body.scrollHeight)"

driver.execute\_script(js)

#滚动到顶部

js = "window.scrollTo(0,0)"

driver.execute\_script(js)

2.Chrome浏览器解决办法：

js = "var q=document.body.scrollTop=0"

driver.execute\_script(js)

## alert\confirm\prompt

*记录自2.21*

弹出框操作主要方法有：

text：获取文本值

accept() ：点击"确认"

dismiss() ：点击"取消"或者叉掉对话框

send\_keys() ：输入文本值 --仅限于prompt,在alert和confirm上没有输入框

## Jquery定位

*记录自2.34*

*一、jquery搜索元素*

*1.按F12进控制台*

*2.点全部按钮*

*3.右侧如果没出现输入框，就点下小箭头按钮*

*4.输入框输入jquery定位语法，如：$("#input1")*

*5.点运行按钮*

*6.左边会出现定位到的元素，如果有多个会以list列表的形式展示出*

jquery语法可以学下w3school的教程：<http://www.w3school.com.cn/jquery/jquery_syntax.asp>

三、jquery行为

1.发送文本语法：$(selector).val(输入文本的值)

2.清空文本语法：$(selector).val('')   # 空字符串，两个单引号

3.点击按钮：$(selector).click()

## Python 学习视频

*记录自3.1* unittest简介

（python基础比较弱的，建议大家多花点时间把基础语法学好，这里有套视频，可以照着练习下：http://pan.baidu.com/s/1i44jZdb 密码：92fs）

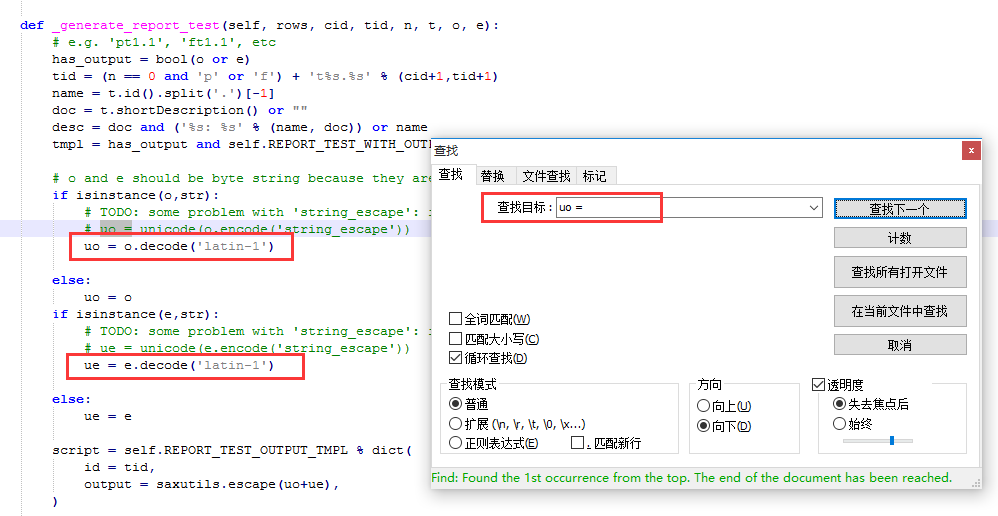
## Html报告乱码问题优化

*记录自3.6*

修改编码

1.找到HTMLTestRunner.py文件，搜索:uo =

2.找到红色区域设置编码的两个地方

3.注释掉红色区域这两个设置，重新添加编码格式为：uo = o.decode('utf-8')   ue = e.decode('utf-8')

4.修改好之后记得保存，重新运行，乱码问题就解决了

三、python3报告问题

1.python3的小伙伴直接用这个下载地址：http://tungwaiyip.info/software/HTMLTestRunner.html的文件，是不能直接生成报告的，需要稍做修改。

2.修改后的源文件已经打包放到QQ群文件：372471871

## unittest常用的断言方法

*记录自3.7*

1）.assertEqual(self, first, second,msg=None)

--判断两个参数相等：first == second

2）.assertNotEqual(self, first, second,msg=None)

--判断两个参数不相等：first ！= second

3）.assertIn(self, member, container,msg=None)

--判断是字符串是否包含：member in container

4）.assertNotIn(self, member,container, msg=None)

--判断是字符串是否不包含：member not in container

5）.assertTrue(self, expr, msg=None)

--判断是否为真：expr is True

6）.assertFalse(self, expr, msg=None)

--判断是否为假：expr is False

7）.assertIsNone(self, obj, msg=None)

--判断是否为None：objis None

8）.assertIsNotNone(self, obj,msg=None)

--判断是否不为None：obj is not None

## selenium常见异常

*记录自3.11*

1.NoSuchElementException：没有找到元素

2.NoSuchFrameException：没有找到iframe

3.NoSuchWindowException:没找到窗口句柄handle

4.NoSuchAttributeException:属性错误

5.NoAlertPresentException：没找到alert弹出框

6.ElmentNotVisibleException：元素不可见

7.ElementNotSelectableException：元素没有被选中

8.TimeoutException：查找元素超时

备注：其它异常与源码在Lib目录下：selenium/common/exceptions有兴趣的可以看看

1.get\_screenshot\_as\_file(self, filename)

--这个方法是获取当前window的截图，出现IOError时候返回False,截图成功返回True。

filename参数是保存文件的路径。

   Usage:

       driver.get\_screenshot\_as\_file('/Screenshots/foo.png')

2.get\_screenshot\_as\_base64(self)

--这个方法也是获取屏幕截图，保存的是base64的编码格式，在HTML界面输出截图的时候，会用到。

比如，想把截图放到html测试报告里。

   Usage:

       driver.get\_screenshot\_as\_base64()

## 等待

*记录自4.1*

显示等待WebDriverWait

Sleep

隐式等待implicitly\_wait()

# Unnittest

# 场景判断与封装

## 判断元素16种方法expected\_conditions

selenium的expected\_conditions模块收集了一系列的场景判断方法，这些方法是逢面试必考的！！！

expected\_conditions一般也简称EC，本篇先介绍下有哪些功能，后续更新中会单个去介绍。

一、功能介绍和翻译

title\_is： 判断当前页面的title是否完全等于（==）预期字符串，返回布尔值

title\_contains : 判断当前页面的title是否包含预期字符串，返回布尔值

presence\_of\_element\_located : 判断某个元素是否被加到了dom树里，并不代表该元素一定可见

visibility\_of\_element\_located : 判断某个元素是否可见. 可见代表元素非隐藏，并且元素的宽和高都不等于0

visibility\_of : 跟上面的方法做一样的事情，只是上面的方法要传入locator，这个方法直接传定位到的element就好了

presence\_of\_all\_elements\_located : 判断是否至少有1个元素存在于dom树中。举个例子，如果页面上有n个元素的class都是'column-md-3'，那么只要有1个元素存在，这个方法就返回True

text\_to\_be\_present\_in\_element : 判断某个元素中的text是否 包含 了预期的字符串

text\_to\_be\_present\_in\_element\_value : 判断某个元素中的value属性是否 包含 了预期的字符串

frame\_to\_be\_available\_and\_switch\_to\_it : 判断该frame是否可以switch进去，如果可以的话，返回True并且switch进去，否则返回False

invisibility\_of\_element\_located : 判断某个元素中是否不存在于dom树或不可见

element\_to\_be\_clickable : 判断某个元素中是否可见并且是enable的，这样的话才叫clickable

staleness\_of : 等某个元素从dom树中移除，注意，这个方法也是返回True或False

element\_to\_be\_selected : 判断某个元素是否被选中了,一般用在下拉列表

element\_selection\_state\_to\_be : 判断某个元素的选中状态是否符合预期

element\_located\_selection\_state\_to\_be : 跟上面的方法作用一样，只是上面的方法传入定位到的element，而这个方法传入locator

alert\_is\_present : 判断页面上是否存在alert

1.打开python里这个目录l可以找到：Lib\site-packages\selenium\webdriver\support\expected\_conditions.py

### 判断title方法title\_is

# 环境失败，总报错

## 安装Python3.6

C:\Users\think\_uyaly\AppData\Local\Programs\Python\Python36

因为Python是跨平台的，它可以运行在Windows、Mac和各种Linux/Unix系统上。在Windows上写Python程序，放到Linux上也是能够运行的。

要开始学习Python编程，首先就得把Python安装到你的电脑里。安装后，你会得到Python解释器（就是负责运行Python程序的），一个命令行交互环境，还有一个简单的集成开发环境。

### 安装Python 3.5

目前，Python有两个版本，一个是2.x版，一个是3.x版，这两个版本是不兼容的。由于3.x版越来越普及，我们的教程将以最新的Python 3.5版本为基础。请确保你的电脑上安装的Python版本是最新的3.5.x，这样，你才能无痛学习这个教程。

### 在Mac上安装Python

如果你正在使用Mac，系统是OS X 10.8~10.10，那么系统自带的Python版本是2.7。要安装最新的Python 3.5，有两个方法：

方法一：从Python官网下载Python 3.5的[安装程序](https://www.python.org/ftp/python/3.5.2/python-3.5.2-macosx10.6.pkg)（网速慢的同学请移步[国内镜像](https://pan.baidu.com/s/1kU5OCOB#list/path=%2Fpub%2Fpython)），双击运行并安装；

方法二：如果安装了Homebrew，直接通过命令brew install python3安装即可。

### 在Linux上安装Python

如果你正在使用Linux，那我可以假定你有Linux系统管理经验，自行安装Python 3应该没有问题，否则，请换回Windows系统。

对于大量的目前仍在使用Windows的同学，如果短期内没有打算换Mac，就可以继续阅读以下内容。

### 在Windows上安装Python

首先，根据你的Windows版本（64位还是32位）从Python的官方网站下载Python 3.5对应的[64位安装程序](https://www.python.org/ftp/python/3.5.2/python-3.5.2-amd64.exe)或[32位安装程序](https://www.python.org/ftp/python/3.5.2/python-3.5.2.exe)（网速慢的同学请移步[国内镜像](https://pan.baidu.com/s/1kU5OCOB#list/path=%2Fpub%2Fpython)），然后，运行下载的EXE安装包：



特别要注意勾上Add Python 3.5 to PATH，然后点“Install Now”即可完成安装。

视频演示：

### 运行Python

安装成功后，打开命令提示符窗口，敲入python后，会出现两种情况：

情况一：



看到上面的画面，就说明Python安装成功！

你看到提示符>>>就表示我们已经在Python交互式环境中了，可以输入任何Python代码，回车后会立刻得到执行结果。现在，输入exit()并回车，就可以退出Python交互式环境（直接关掉命令行窗口也可以）。

情况二：得到一个错误：

‘python’ 不是内部或外部命令，也不是可运行的程序或批处理文件。



这是因为Windows会根据一个Path的环境变量设定的路径去查找python.exe，如果没找到，就会报错。如果在安装时漏掉了勾选Add Python 3.5 to PATH，那就要手动把python.exe所在的路径添加到Path中。

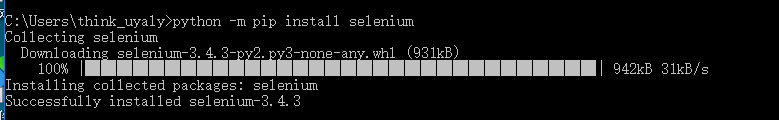
如果你不知道怎么修改环境变量，建议把Python安装程序重新运行一遍，务必记得勾上Add Python 3.5 to PATH。

视频演示：

## 安装selenium

在cmd里面执行

C:\Users\think\_uyaly>python -m pip install selenium



1. 方法一：pip命令行安装，运行 | cmd，打开命令行，-U其实就是--upgrade，升级安装。

pip install -U selenium

1. 方法二：源码解压安装，前往[https://pypi.python.org/pypi/selenium下载最新版的PyPI版本的Selenium，解压后执行](https://pypi.python.org/pypi/selenium%E4%B8%8B%E8%BD%BD%E6%9C%80%E6%96%B0%E7%89%88%E7%9A%84PyPI%E7%89%88%E6%9C%AC%E7%9A%84Selenium%EF%BC%8C%E8%A7%A3%E5%8E%8B%E5%90%8E%E6%89%A7%E8%A1%8C)

python setup.py install

uninstall

E:\ly\other\python\selenium-3.4.3\selenium-3.4.3

## 开始使用WebDriver，运行一段代码

## 引入WebDriver包

from selenium import webdriver

## 引入WebDriver Keys包

from selenium.webdriver.common.keys import Keys

## 创建浏览器对象

browser = webdriver.Firefox()

## 导航到百度主页

browser.get('https://www.baidu.com')

## 检查标题是否为‘百度一下，你就知道’

assert '百度一下，你就知道' in browser.title

## 找到名字为wd的元素，赋值给elem

elem = browser.find\_element\_by\_name('wd') # 找到搜索框

elem.send\_keys('seleniumhq' + Keys.RETURN) # 搜索seleniumhq

## 关闭浏览器

browser.quit()

## Selenium 报错解决办法

1. Geckodriver executable needs to be in PATH。Geckodirver的下载地址：<https://github.com/mozilla/geckodriver/releases>  
   报错内容:

WebDriverException:Message:'geckodriver'executable needs to be in Path

geckodriver是一原生态的第三方浏览器，对于selenium3.x版本都会使用geckodriver来驱动firefox，所以需要下载geckodriver.exe。放置在Path 环境变量可以访问到的地方。例如 C:\python34

C:\Users\think\_uyaly\AppData\Local\Programs\Python\Python36 C:\Users\Administrator\AppData\Local\Programs\Python\Python36

1. 需要将火狐的安装路径放到path，然后重启（必须重启电脑）  
   报错内容：

selenium.common.exceptions.WebDriverException: Message: 'geckodriver' executable needs to be in PATH.

参考地址：<http://stackoverflow.com/questions/40208051/selenium-using-python-geckodriver-executable-needs-to-be-in-path/40208762>

3. selenium.common.exceptions.WebDriverException: Message: Unable to find a matching set of capabilities

Message: Unable to find a matching set of capabilities

解决办法：Update Firefox to version >= 52.0.3 升级firefox

然后ff可以了