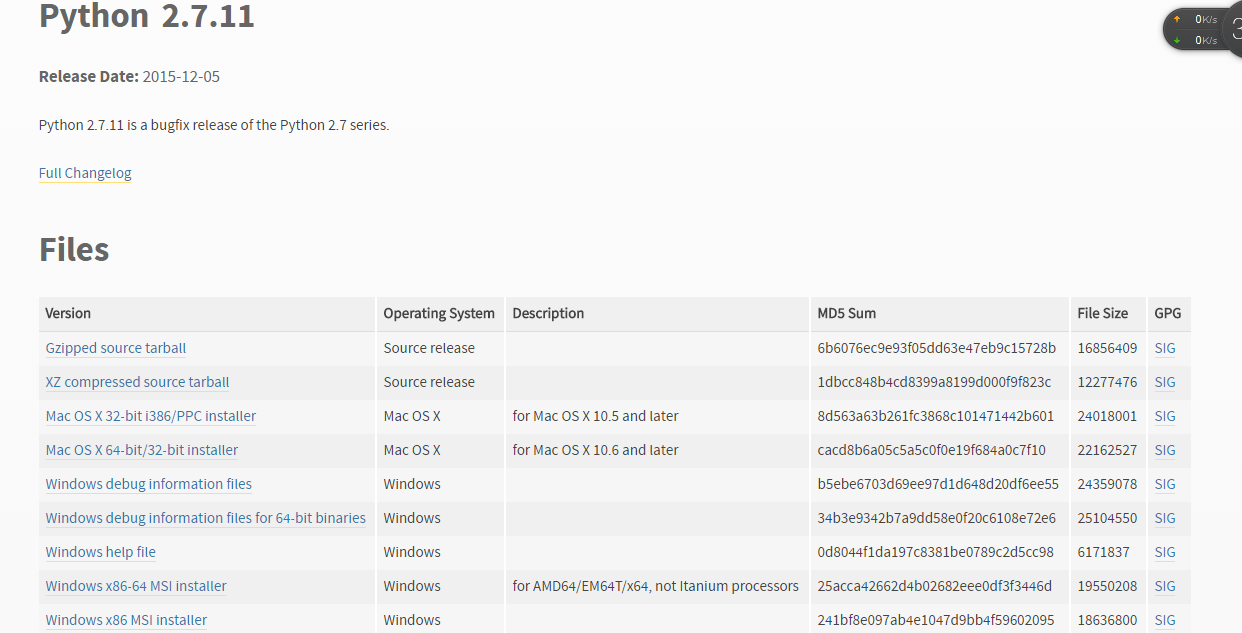
# 环境搭建

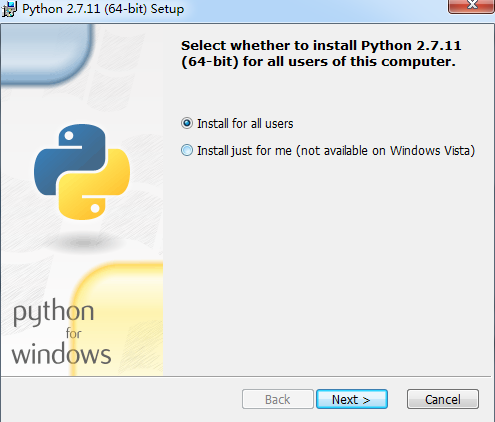
详细步骤：

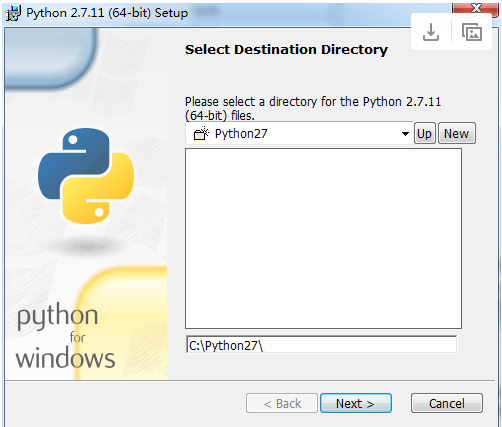
1. 安装Python

访问Python的下载地址：https://www.python.org/downloads/release/python-2711/

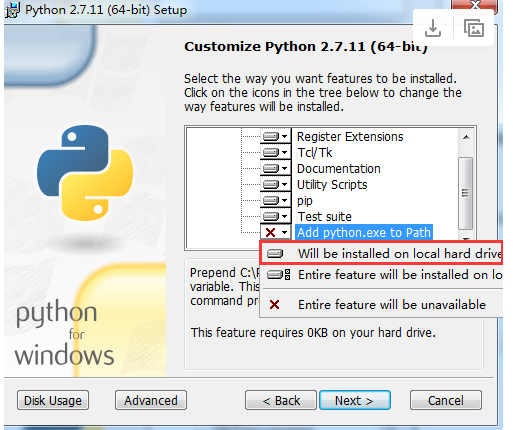


如果以后有新版可以在官网首页看一下是什么版本了，我们需要下载Python2.7系列的，现在最新的是2.7.11，所以我下载Python2.7.11.msi文件，这个是32位的。

下载后双击文件运行 

这里直接点Next就行了 

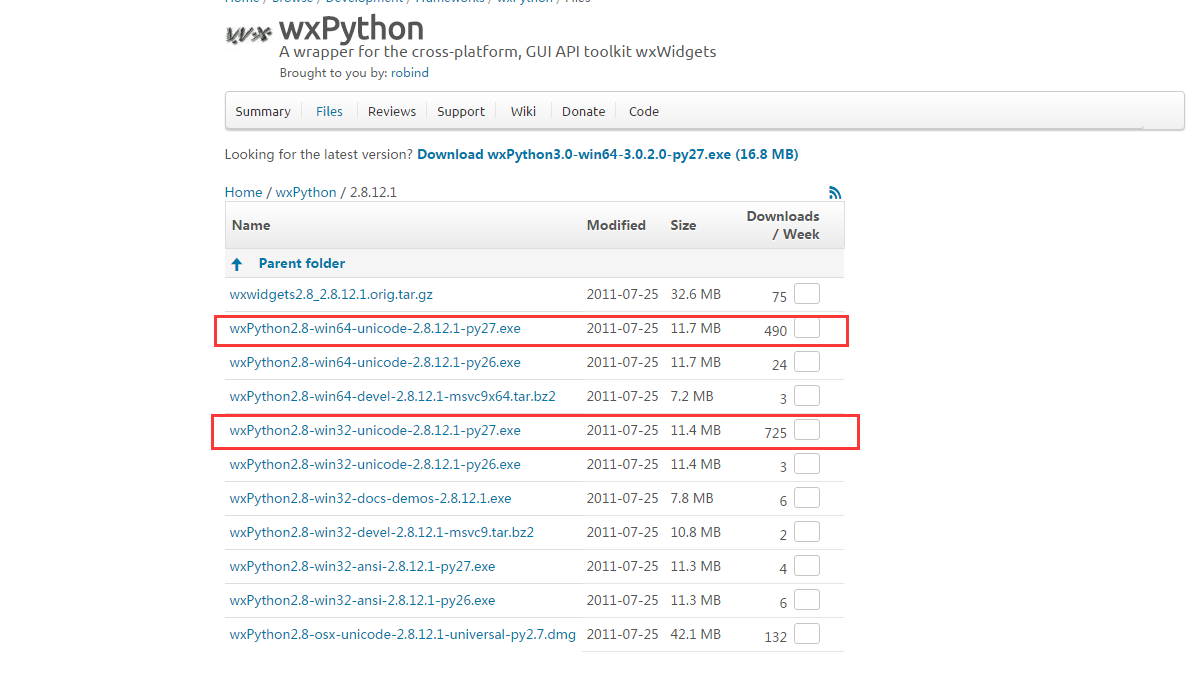
这里设置Python的目录，默认是C:\Python27\，你可以改成D盘或其他盘，但是最好不要选择有中文或空格的目录（比如Program Files），因为以后你安装其他库会遇到问题（尤其是AutoItLibrary）。设置好之后点Next继续

看到这个界面，需要加入环境变量进入PATH 

2. 安装wxPython

为了用到后面的RIDE，我们需要先安装wxPython，虽然RIDE已经在2.0alpha版本开始用3.0的，但是貌似新版还有很多坑，先别急着踩坑了，还是用稳定的wxPython2.8.12.1吧。

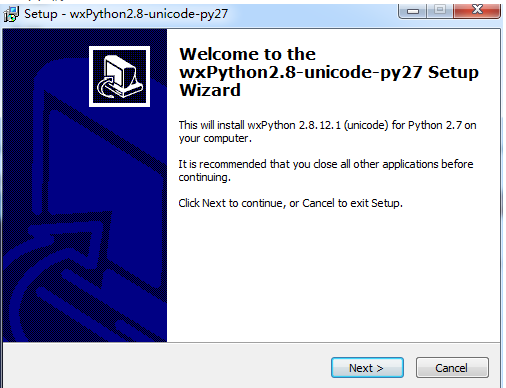
下载地址：https://sourceforge.net/projects/wxpython/files/wxPython/2.8.12.1/



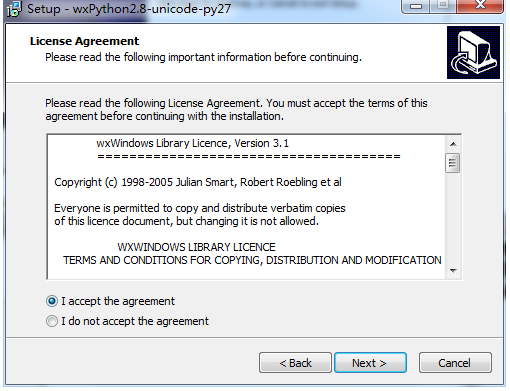
由于文件名字太长，需要把鼠标悬浮到每一个文件上看完整的文件名，确认名字是wxPython2.8-win32-unicode-2.8.12.1-py27.exe，不要有差错。

win32是因为Python是32位的，unicode是必须的，不要下载ansi版本，2.8.12.1是RIDE稳定支持的版本，py27是Python2.7系列的，不要下错2.6的。

双击下载exe文件



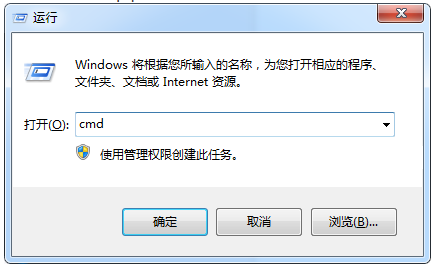
点击Next，选择I accept那一项，点击Next



这里不用管，直接一路Next安装完成就行。

3. 安装robotframework

Python2.7.11这个版本自带了pip（我不记得从哪个版本开始有pip的，印象里之前的版本都没有的），所以安装好Python后就不需要单独安装pip了，我们直接在打开运行，输入cmd



接下来我们用pip来安装Robotframework：

1、cmd页面 进入C:\Python27\Scripts目录

 pip install robotframework==3.0

一般情况下后面的 ==3.0 这部分不需要输入的，这是指定特定版本安装的语法，如果不加也没问题，默认是安装最新的版本。

看到Successfully就是安装完成了。

如果没有看到Successfully，有可能你的网络有问题，因为这种安装方式都是依赖网络的。

如果你的网络无法安装，那么只能先去下载Robotframework的源码包，这里提供两个地址：

https://github.com/robotframework/robotframework/releases

https://pypi.python.org/pypi/robotframework

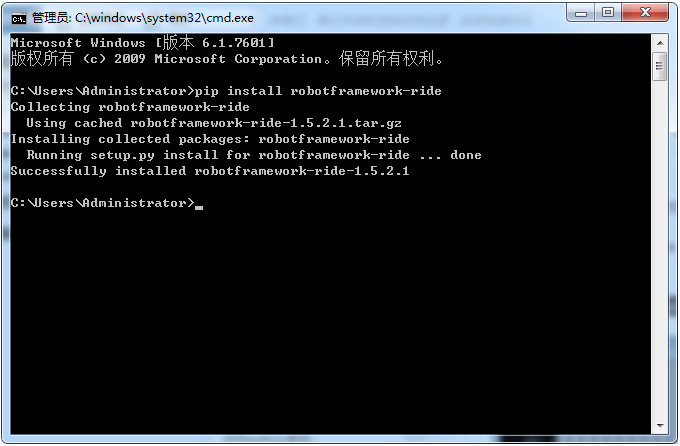
两个地方都有最新的包，以3.0版本为例，robotframework-3.0.tar.gz文件就是我们需要下载的，下载后解压缩，从命令行进入解压缩后的目录（有setup.py文件的那一层），输入命令：

python setup.py install

4. 安装robotframework-ride

输入安装命令：cmd页面 进入C:\Python27\Scripts目录

> pip install robotframework-ride



比如这次我们没有输入特定版本号，他就安装了最新的1.5.2.1版本。

同样看到Successfully就是安装成功了。

如果没有成功的话，请参考Robotframework下载源码安装的方式，下载ride的源码，同样提供两个地址：

https://pypi.python.org/pypi/robotframework-ride

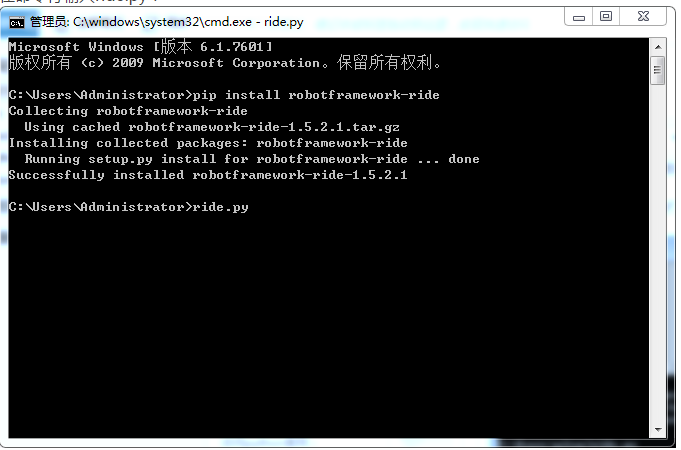
https://github.com/robotframework/RIDE/releases

5. 完成基础环境安装，简单验证

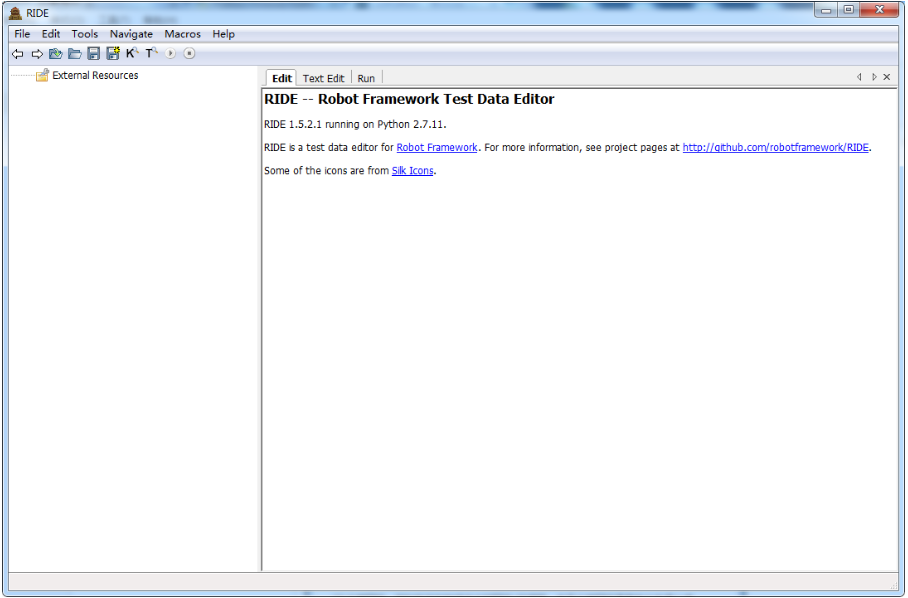
此时基础环境基本上都安装好了，我们来简单验证一下。

cmd页面 进入C:\Python27\Scripts目录

在命令行输入ride.py：



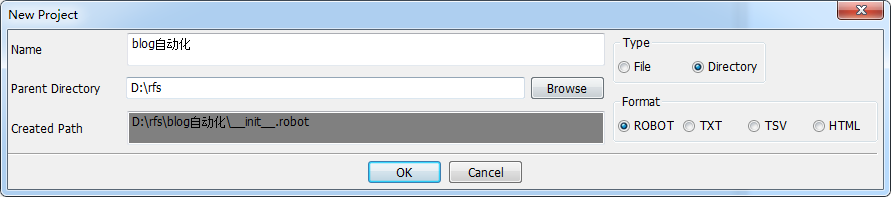
看到如下界面就是安装成功了。



# Robot Framework 入门

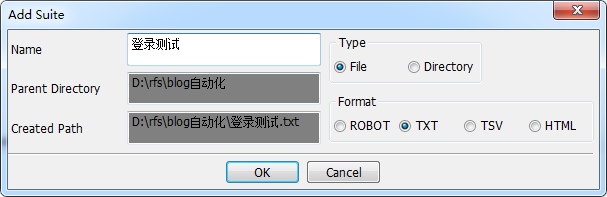
1、创建测试项目

选择菜单栏 file----->new Project

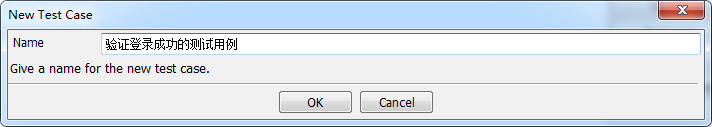


从Tpye上来说，分为文件和目录两种，区别嘛，看你的工程定，如果内容很简单，File类型的就可以，如果内容较多，从方便管理的角度来说，选Directory的比较合适。

2、创建测试套件



3、创建测试用例

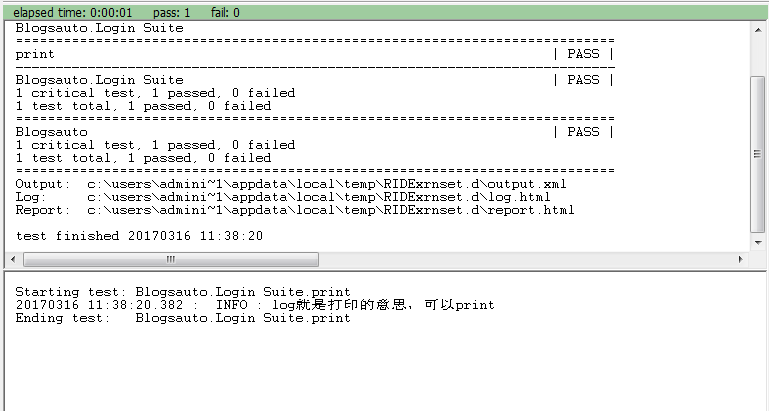


## 常用关键字介绍

1、log 就是“print”



在 Run 标签页，将看到执行结果：

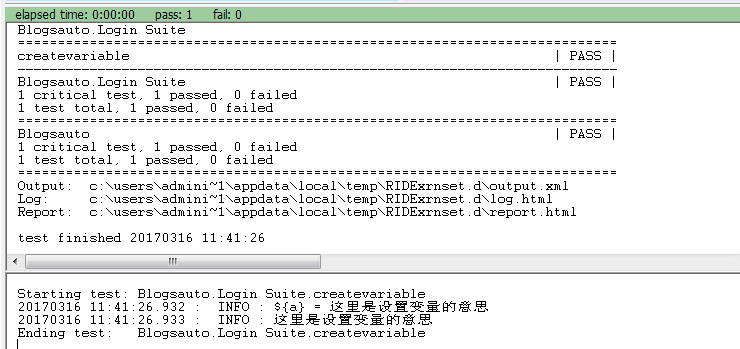


2、定义变量：

通过“Set variable”关键字来定义变量



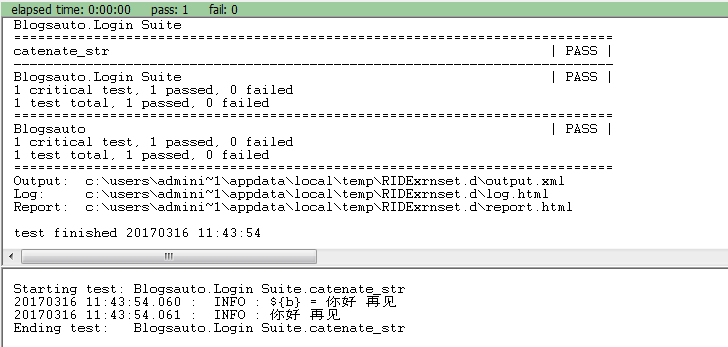
运行结果：



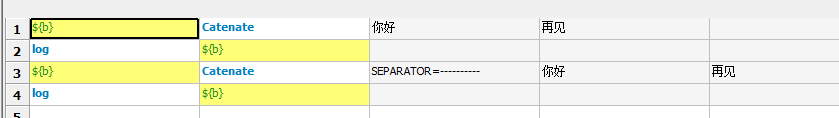
3、连接对象

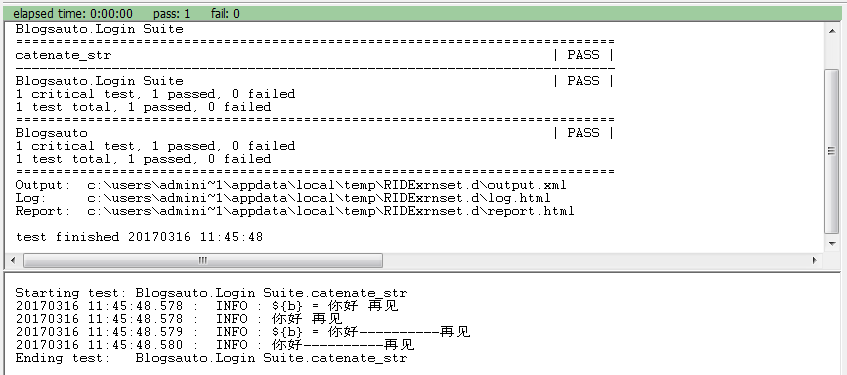
“Catenate”关键字可以连接多个信息



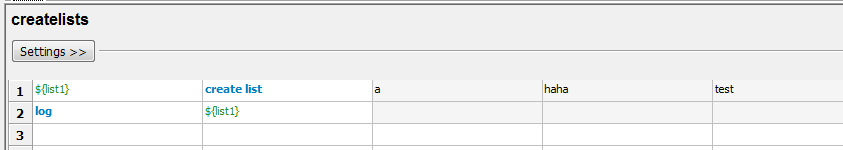


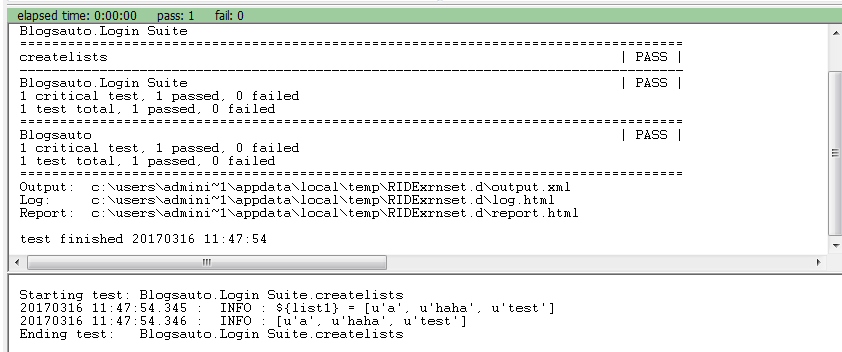
“SEPARATOR=”可以对多个连接的信息进行分割



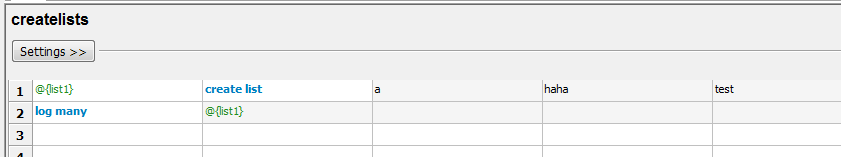


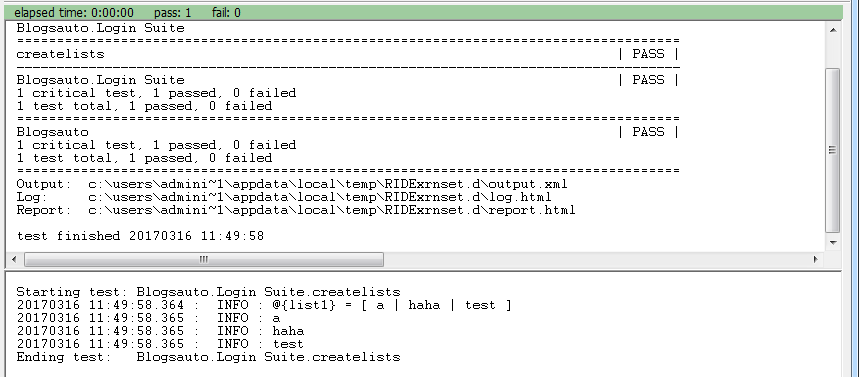
4、定义列表（通过“Create List”关键字可以定义列表）





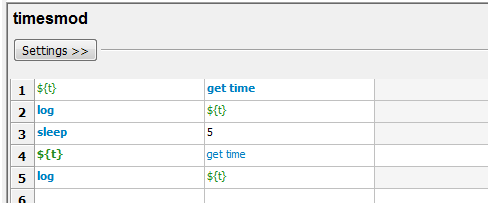
Ps（通过“@{}”去定义列表的话，可以通过“log many”关键字进行打印）

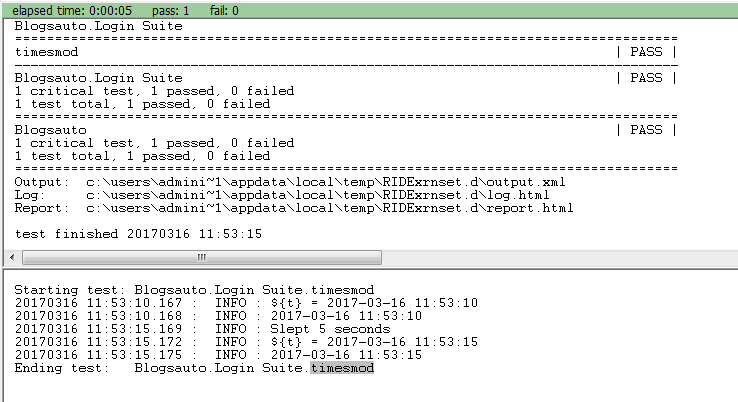




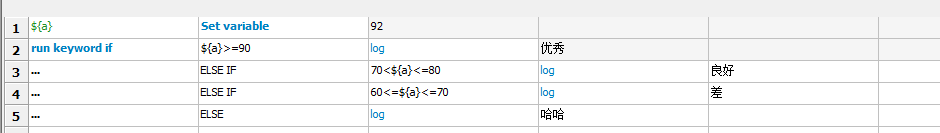
5、时间的操作

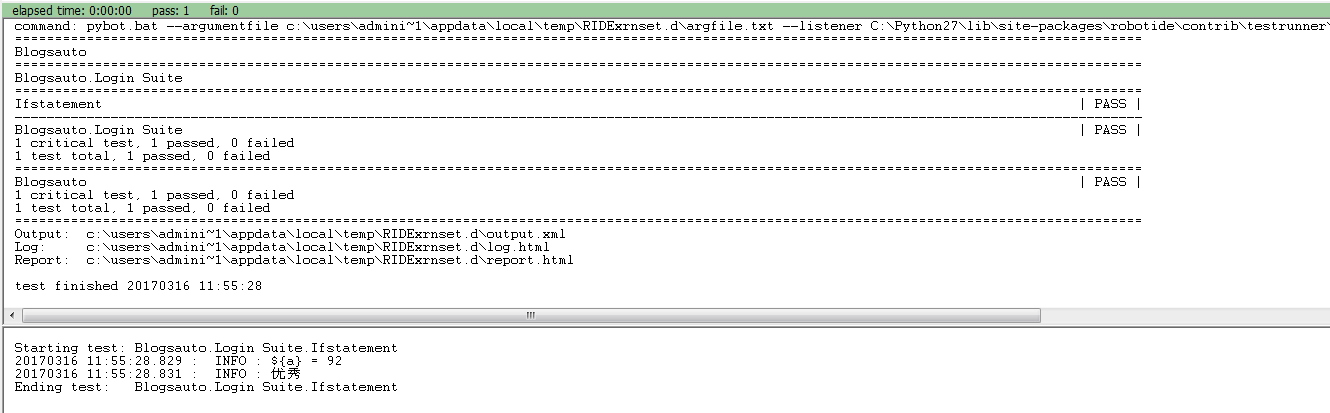
“get time”关键字用来获取当前时间。sleep表示休眠



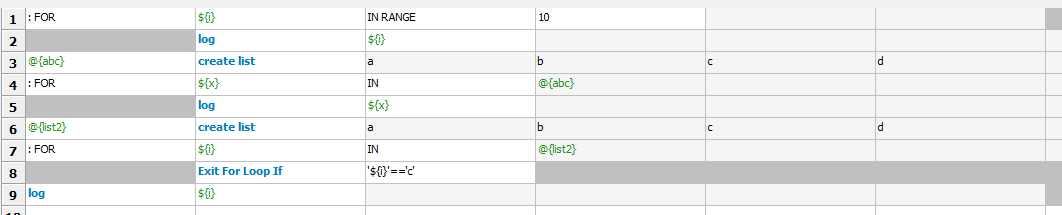


6、if 语句（“run keyword if”关键字可以编写 if 分支语句）

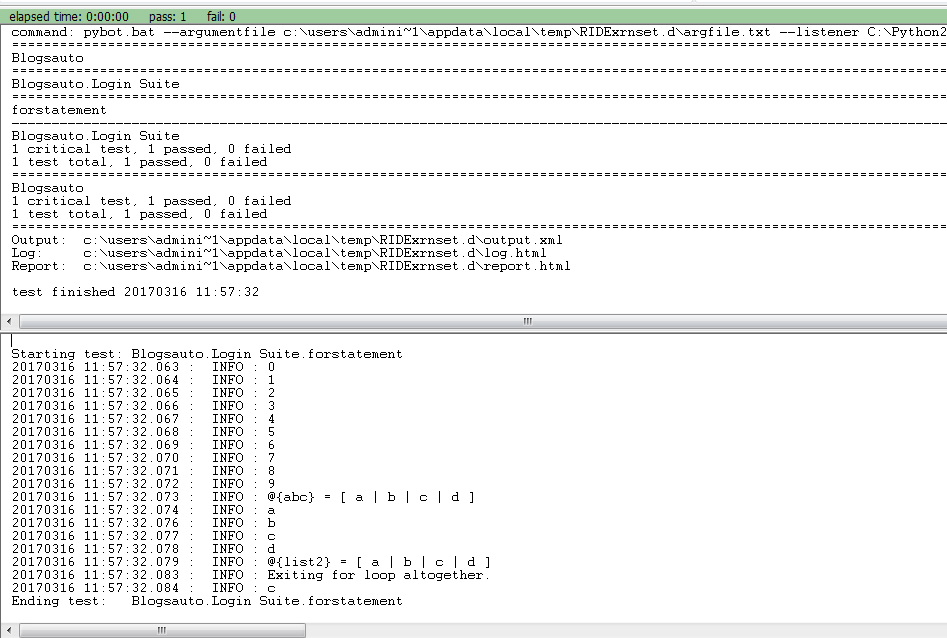




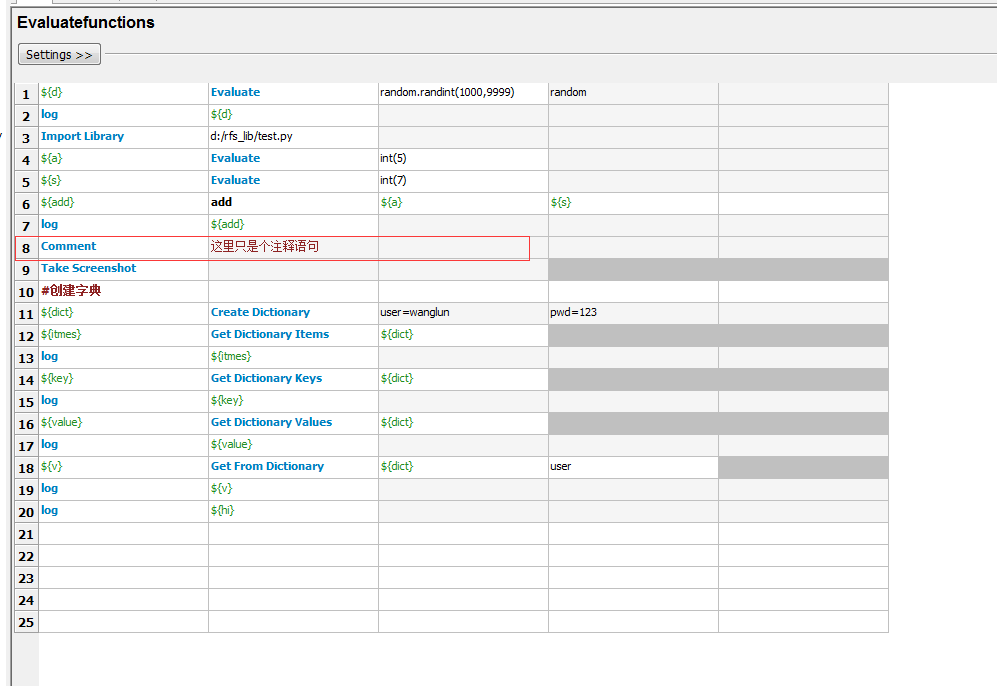
7、for 循环（通过“:for”关键字编写for循环）



通过“Exit For Loop If”关键字时行 for 循环内的判断，当满足 Exit For Loop If 条件后，循环结束。



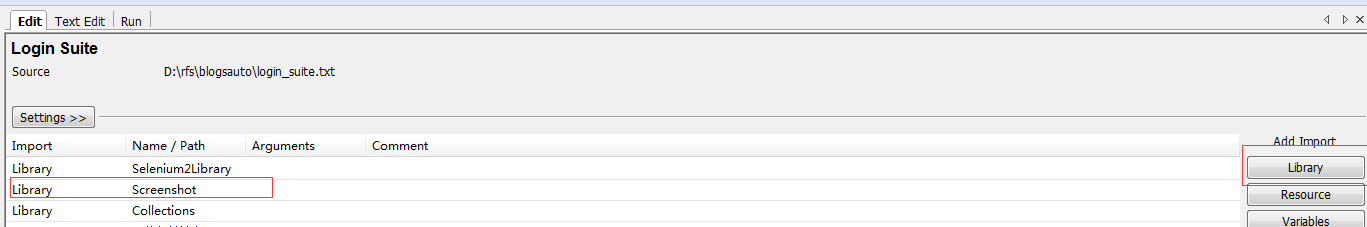
8、 Evaluate关键字，通过它可以使用 Python 语言中所提供的方法。



9、导入库（“Import Library”关键字）

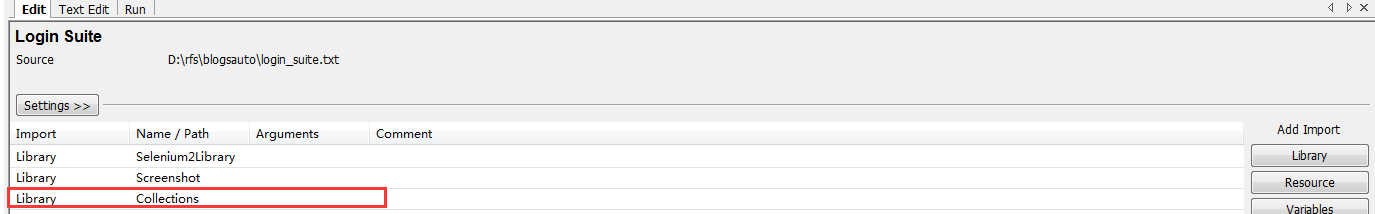
C:\Users\ADMINI~1\AppData\Local\Temp\enhtmlclip\Image(22).png

10、Screenshot 库



使用“Take Screenshot”关键字实现截取当前屏幕。

11、Collections 库（提供的关键字主要用于列表、索引、字典的 处理。）



创建字典（Create Dictionary 关键字用于创建关键字。）

操作字典（Get Dictionary Items 关键字用于读取字典的 key 和 value）

1. Get Dictionary Items 关键字获取字典中的 key 和 value。
2. Get Dictionary Keys 关键字获取字典中的 key。
3. Get Dictionary Values 关键字获取字典中的 value。
4. Get From Dictionary 关键字获取字典中的 key 对应的 value

# 认识 RIDE

Edit 标签

（1）：加载外部文件

Add Library：加载测试库，主要是[PYTHON 目录]\Lib\site-packages 里的测试库

Add Resource：加载资源，主要是你工程相关的资源文件

Add Variables：加载变量文件。

（2）：定义内部变量

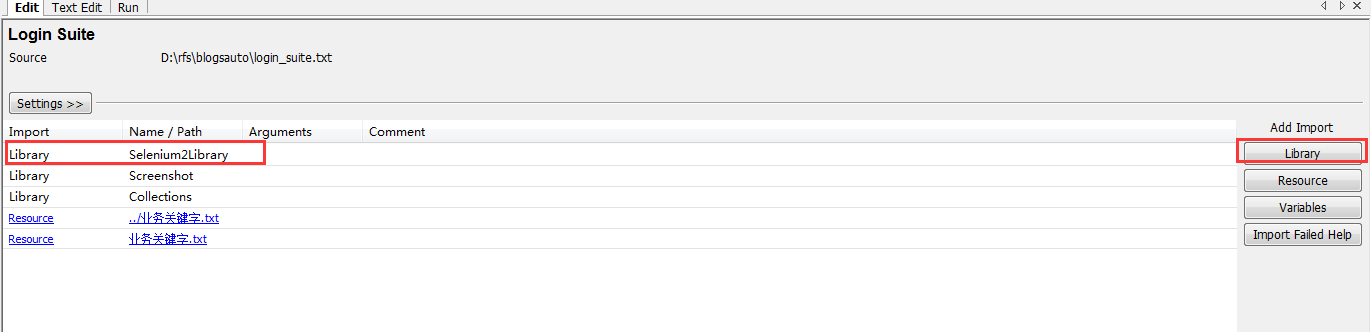
Add Scalar：定义变量

Add List：定义列表型变量

（3）：定义元数据

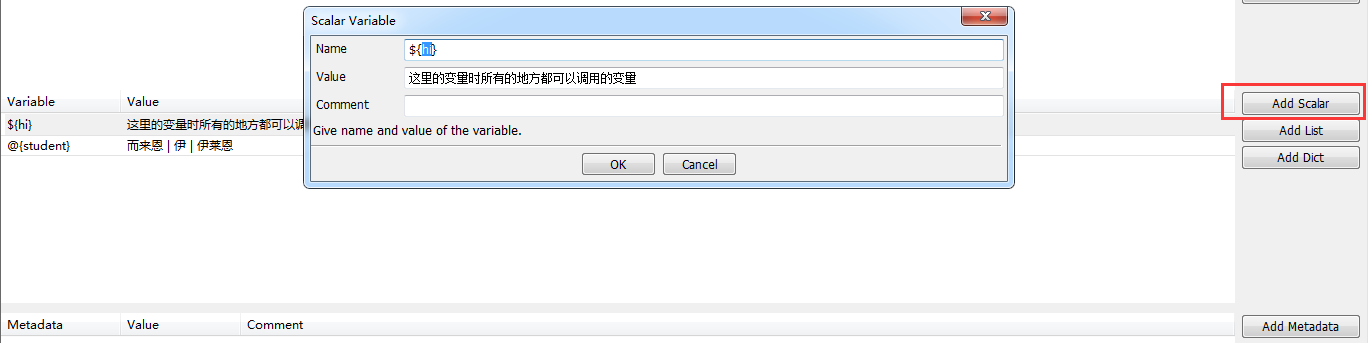
Add Metadata：定义元数据。是在 report 和 log 里显示定义好的内容，格式和 document 一样。

导入库



定义变量：

点击 Edit 标签页右侧的“Add Scalar”按钮来创建变量。这里创建的变量可以被整个测试套件中的 用例所使用。



筛选执行用例：

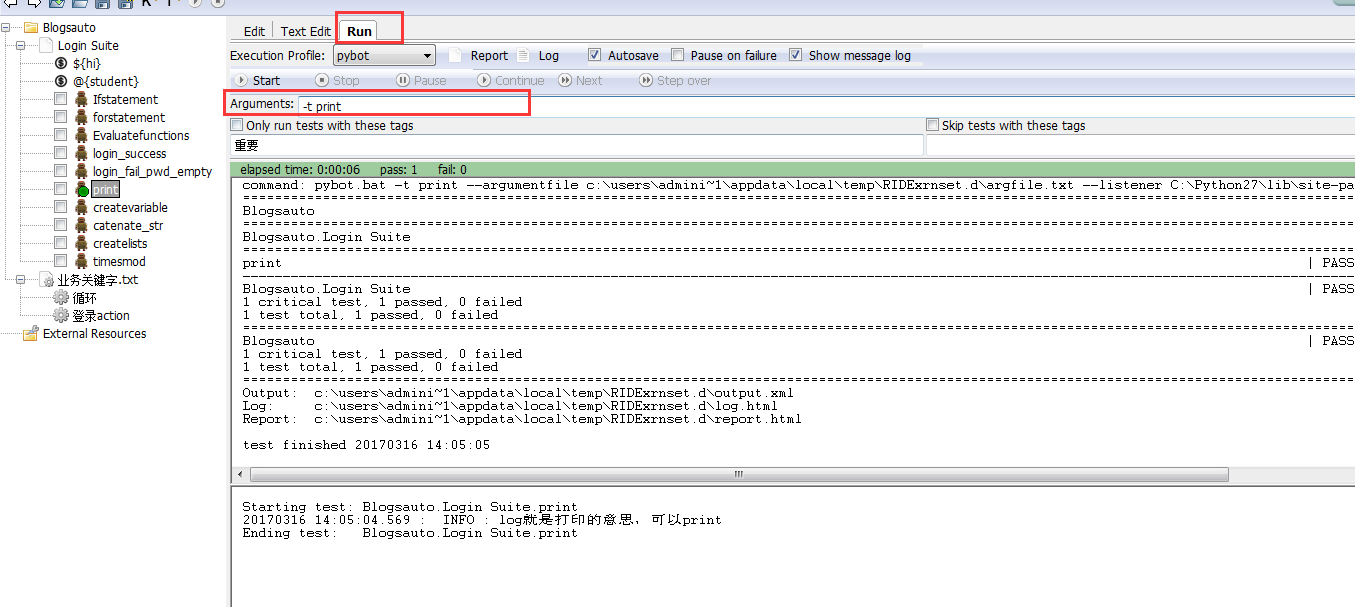
1、要运行哪条用例就勾选哪一条。如果全部不勾选，点击“运行”按钮会运行 所有用例。

2、“测试套件”上右键选择：

* Select All Test：选择当前套件的所有用例。
* Select Only Failed Test：选择当前套件下运行失败的用例。
* Select Only Passed Test：选择当前套件下运行成功的用例。

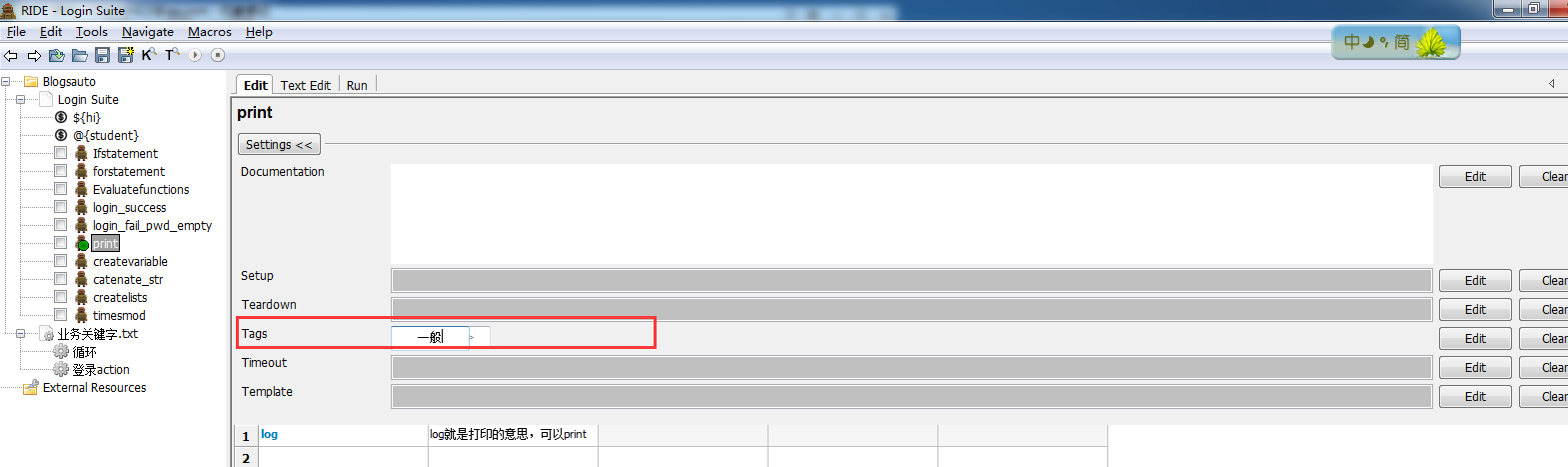
3、用命令：

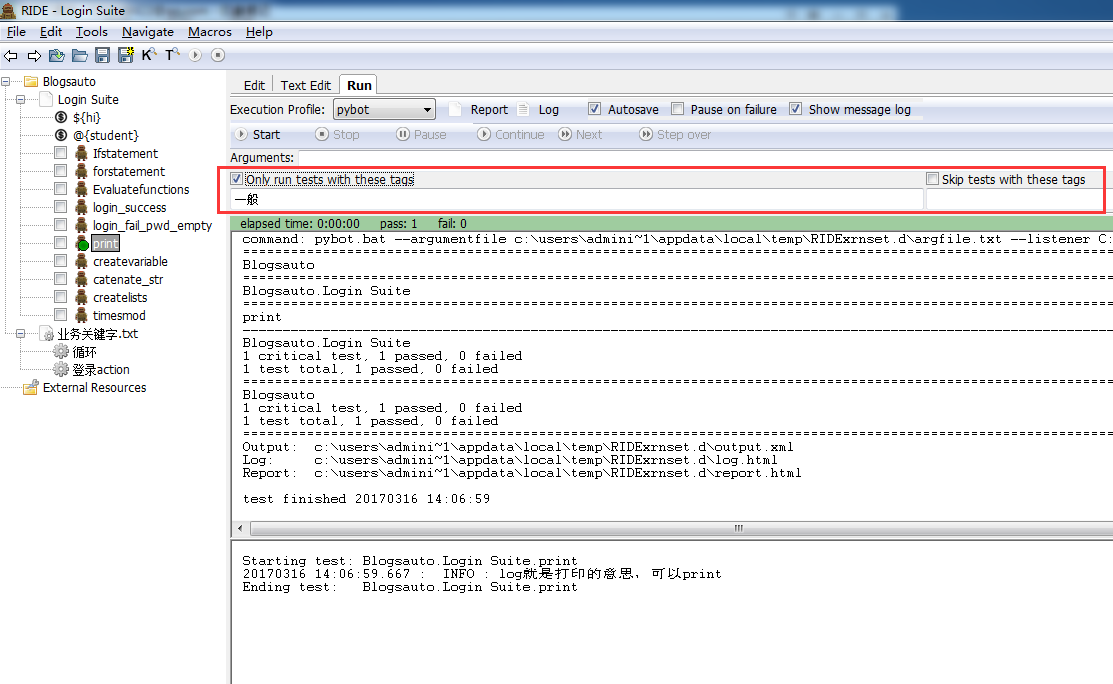
a---->在 Arguments 的输入框内输入“-t 用例名”。点击“Start”按钮，只执行了 这一条用例。



b---->筛选标记

点击某个用例，你会看到“Setting>>”的按钮，点击按钮展开：

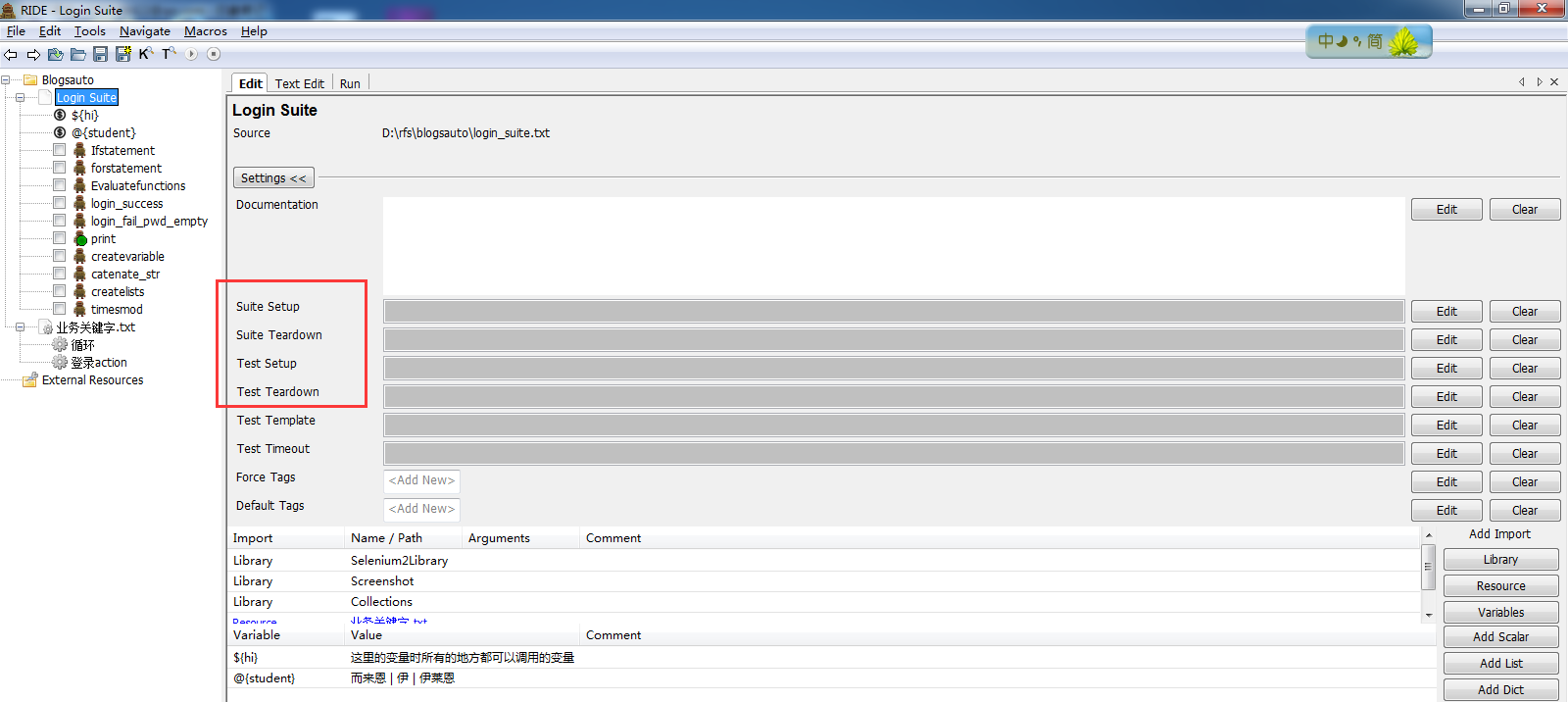




* Only Run Tests with these Tags：只运行这些标记的测试案例。
* Skip Tests with these Tags： 跳过这些标记的测试案例。

测试套件的 Settings

点击测试用例上的“Settings>>”按钮，会看到下面的选项。



Setup 和 Teardown：如果你了解 unittest 单元测试框架的话，一定对这两个单词不陌生。

setUp 用于设置初始化工作，在每一个测试用例前先被执行

tearDown 方法在每个测试方法执行后调用，这个方法用于完成测试用例执行后的清理工作，如执行 “close browser”关闭浏览器等。

TimeOut：用于设置用例的超时时间。如“1 min 10s”、“ 2 hours”等。

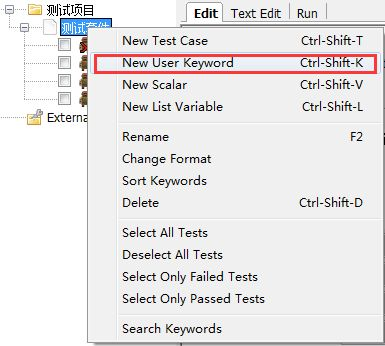
Template：指定模板使用的关键字。

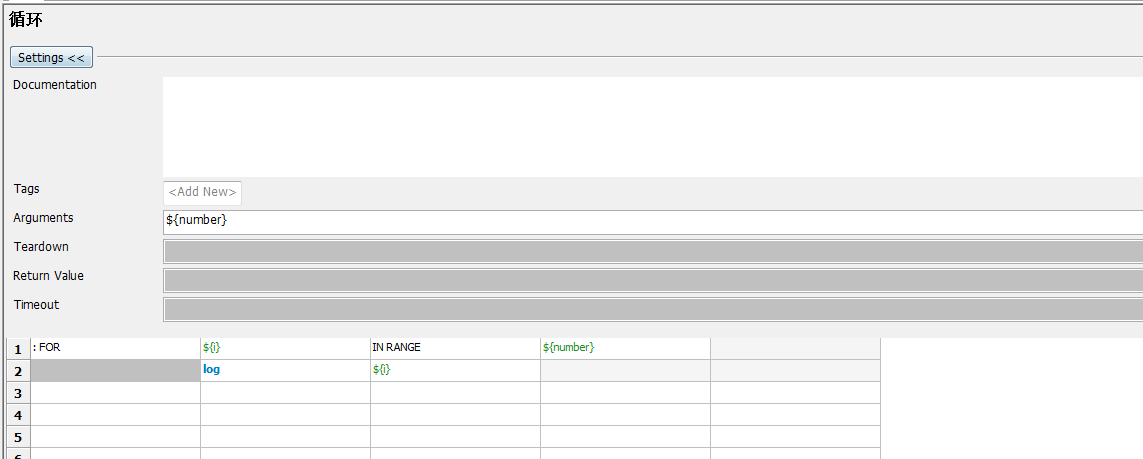
Tags ：用于给用例添加标记。在上一小节中有用到这个功能。

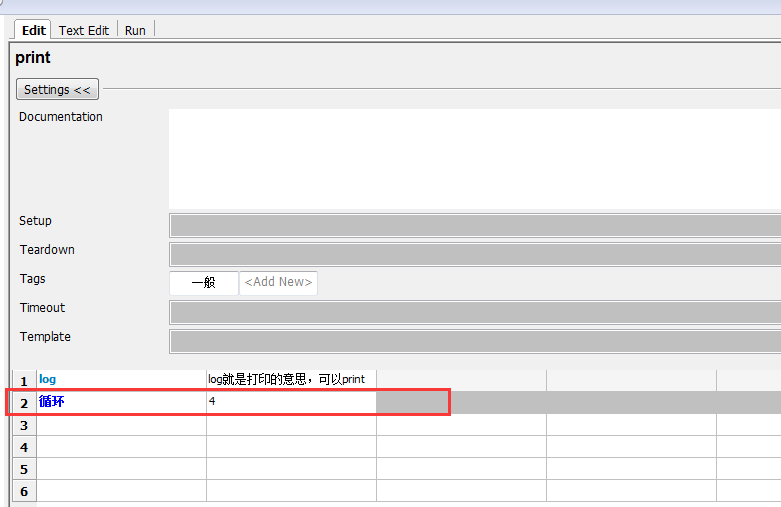
Force Tags：表示当前测试套件下测试用例强制的标记。

Default Tags：表示当前测试套件下测试用例默认的标记。

用户关键字（主要利用现有的 系统关键字，根据不同的业务，把多个重复的步骤集合在一起组成用户关键字。）



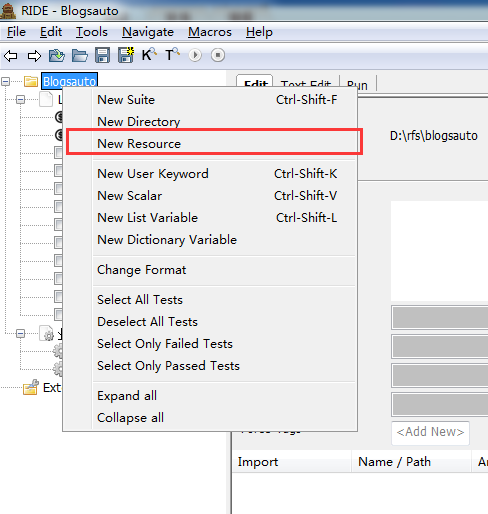




创建资源

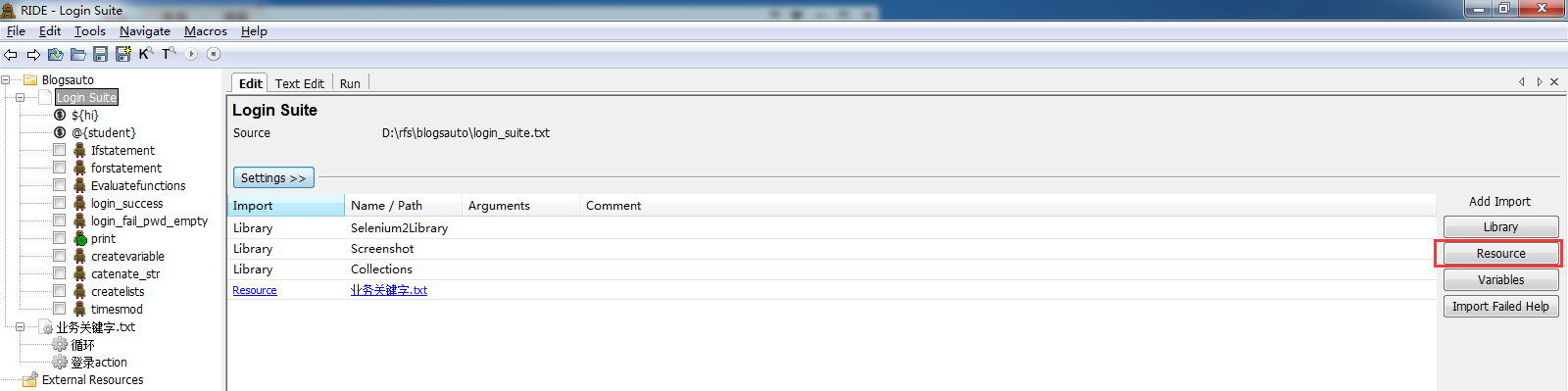
1、创建资源

右键“测试项目”选择“New Resource”创建资源。



2、导入资源

因为“业务关键字.txt”和“测试套件”属于并列关系。测试套件要想使用业务关键字下的 关键字，需要导入资源。

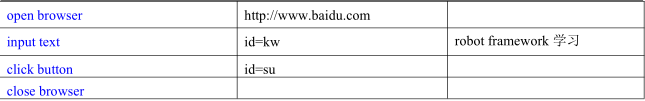


# Selenium2Library

1、安装 Selenium2Library

     pip install robotframework-selenium2library

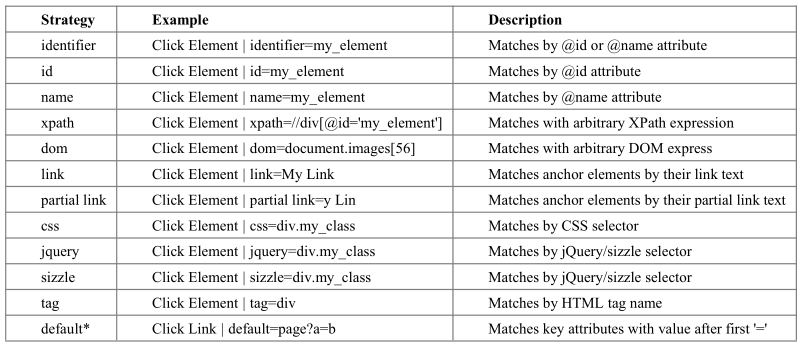
2、将“Selenium2Library”库添加到相应的测试套件中。





3、元素定位

Selenium2Library 提供了非常丰富的定位器：



4、浏览器驱动

通过不同的浏览器执行脚本

C:\Users\ADMINI~1\AppData\Local\Temp\enhtmlclip\Image(39).png



常用方法：

|  |  |
| --- | --- |
| close browser | 关闭当前的浏览器 |
| close all browser | 关闭所有打开的浏览器和缓存重置 |
| Maximize Browser Window | 关键字使当前打开的浏览器全屏 |
| get windows size | 关键字用于打设置打开浏览器的宽度和高度 |
| input text | 关键字用于向文本框内输入内容 |
| Click Element | 关键字用于点击页面上的元素 |
| Click Button | 用于点击页面上的按钮 |
| Wait Until Page Contains Element | 关键字用于等待页面上的元素显示出来 |
| get title | 关键字用于获得当前浏览器窗口的 title 信息 |
| get text | 关键字用于获取元素的文本信息 |
| Get Element Attribute | 获取元素属性值 |
| Should Contain  A B | 比较B是否inA |
| Select Frame | 进入表单 |
| Unselect Frame | 退出表单 |
| Unselect From List By Value | 下拉框选择 |
| Execute Javascript | 关键字用于使用 JavaScript 代码 |