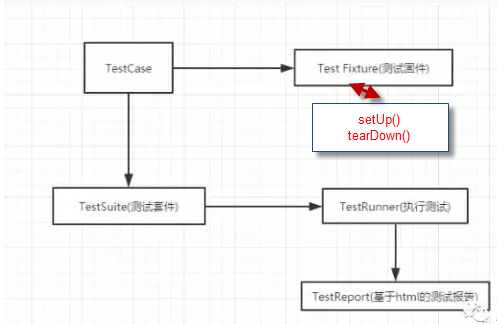
Python中采用unittest管理自动化测试用例使用笔记

1. 什么是unittest

不论对于开发还是测试，都离不开单元测试框架，对于开发而言使用单元测试框架，可以编写测试代码来验证验证自己编写的功能是否正确，对于测试而言，使用单元测试框架，可以编写自动化的测试用例，在Python中单元测试框架是Pyunit，即unittest,它是python的标准库，官方详细的地址是：https://docs.python.org/2/library/unittest.html。单元测试支持测试自动化、 共享的安装程序和关闭代码测试、 聚合成集合，测试和报告框架从测试的独立性。单元测试模块提供可以很容易地支持这些素质的一组测试的类。

1. Unittest的运行原理

unittest各个模块的关系为：



从上面结构图可以看出，最小测试单元是TestCase, 然后是TestSuite。在unittest框架内部，大概用例执行顺序是：

1. 找到TestCase(一个或者多个测试脚本文件)
2. 添加到TestSuite(如果没有，默认新建一个测试套件)
3. 调用框架内TestRunner模块去执行
4. 调用TestReport模块生成基于html的测试报告

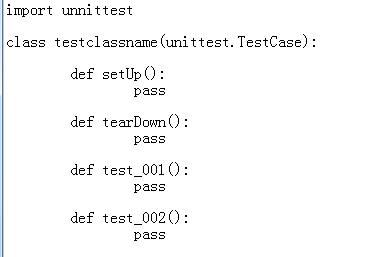
我们知道TestCase是unittest框架中最小组成部分，一个testcase一般，有三部分组成：

1. 用例执行前提操作
2. 具体用例
3. 用例执行后操作

一般我们习惯把 用例执行前提操作放在方法 setup()中，用例执行后操作放到tearDown()方法中去，具体的一个测试用例，方法名称必须要小写的test开头，unittest才能自动识别这是需要执行的用例，否则不会被执行。

Unittest把setUp()和tearDown()部分叫TestFixture，即测试构件。

在面向对象编程的自动化测试脚本中，一个最基本的测试类文件如下：



解释：

1. **unittest.TestCase**：TestCase类，所有测试用例类继承的基本类。
2. **setup()和tearDown()**如果成对出现，先执行setUp,然后执行test开头的用例，最后一定会执行tearDown部分。
3. **pass** 是指具体业务代码块，pass在python是一个保留字，意思是跳过执行，什么也不做，如果看到pass

TestCase包含的属性方法：

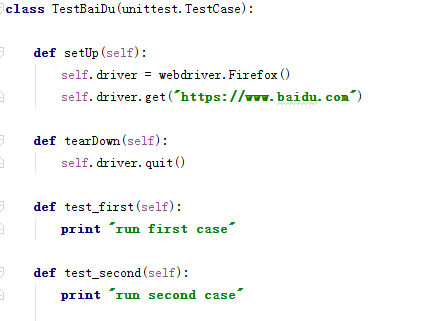


最常用的就是setUp() 和tearDown()方法，还有就是assert开头的一些断言方法。

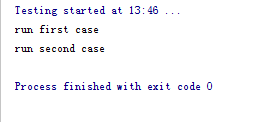
注意两个相似的方法 setUpClass() 和 tearDownClass() 这个下面文章会介绍。

1. 实际项目中举例
   1. **同一个类文件，setUp() 和tearDown() 会执行多次，如何避免**

具体看一个web测试用例，比较形象，下面是一个打开百度，执行两条用例的例子。

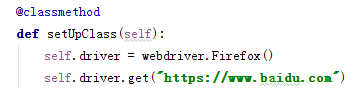


执行结果：

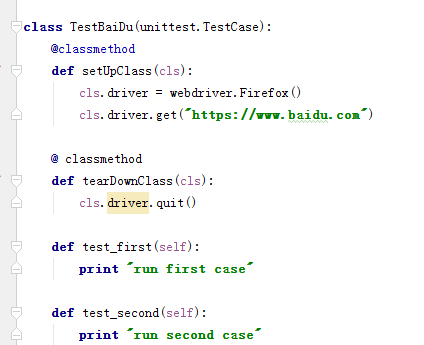


如果你运行，你可以发现，setUp()和tearDown()执行了两次，也就是说，打开和关闭火狐浏览器都操作了两次。那么可不可以让测试固件只执行一次了，也就是说在一个测试类中，有N个测试用例，执行这个测试类中的测试用例后，测试固件只执行一次。当然是可以的，unittest提供了这样的解决方案，在这里钩子方法使用的是类方法：setUpClass() 和tearDownClass()派上用场。

我们利用类方法的构件来重构这个代码，看看是不是构件只会执行一次。



需要添加@classmethod 指明是一个类方法，看到self下面的波浪线没有，类方法默认参数是cls, 鼠标悬停在波浪线上面，会出现这个提示，所以我们修改后的代码如下：



Note:

这里有一个警告，换色区域的driver，报错是说这个当期类没有定义driver这个属性，但是在tearDownClass()中引用，发生警告。幸运的是，不会影响脚本的执行，最后也会关闭当前浏览器。这个问题，暂时没有更好的办法，所以，先放这里。一个类文件，下面有N个测试用例方法，以后我们都采取这样组织结构。