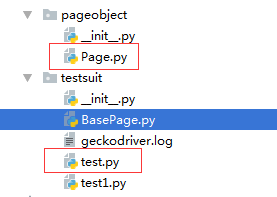
**Python中Page Object的简单实现**

1. 什么是页面对象

所谓的页面对象模型，是一种思路，不是框架，在web测试领域的一种测试思想。就是要把业务逻辑操作和测试脚本分离，而且页面发生改变，只需要去修改变化的页面，不会影响脚本的执行。

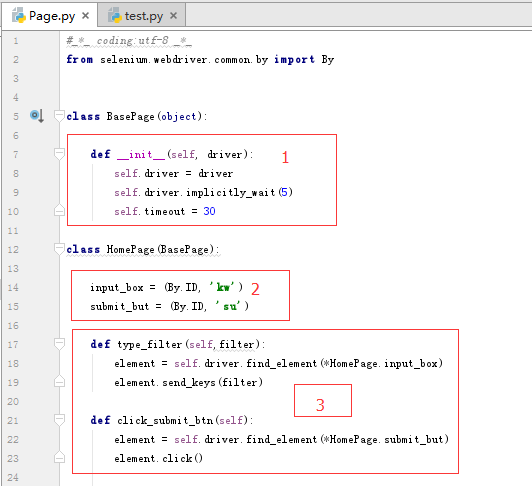
由于之前在基于java+selenium自动化测试框架中实现了这个POM，所以，也想在python中去实现，结果一动手，遇到了很多问题，所以这里，先记录下来。

1. 如何去实现
   1. 项目中的结构图



解释下，pageobject是一个包，里面新建一个Page.py 这个是一个基本页面类，当然在这里，演示中的HomePage我也写到Page.py里面。Page是一个模块，这个模块下有两个class， 一个BasePage, 一个 HomePage, 其中BasePage是所有页面的父类，HomePage继承了BasePage

* 1. Page.py代码如下



解释下：

方块1：定义了基类，有一个driver参数，后面两行关于时间的可以不写

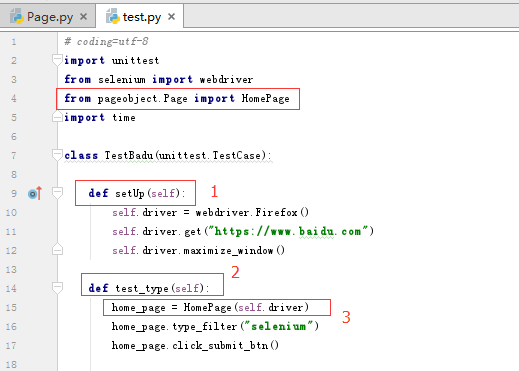
方块2：定义了百度首页输入框和搜索按钮这两个元素

方块3：两个方法，在输入框输入字符和点击搜索按钮

注意：这里element = self.driver.find\_element(\*HomePage.input\_box)

如果你想通过self.driver. 然后根据点后面提示选择find\_element方法是不行的，因为你在BasePage没有实例化一个webdriver对象，这个时候中的driver是没有这些方法提示，你直接写上这些方法就可以。还有一个注意到\* 这个符号没有，python是支持这个来引入元素或者参数的。

* 1. 测试用例test.py 代码如下



解释下：

Def setUp() 这个是unittest工具的方法，就是初始化的公共步骤放到这里执行

红框2：所有的测试脚本都的函数名称都需要test开头，看到test就告诉unittest这个需要执行的测试用例

红框3：这个是页面对象的实例化，注意到参数self.driver没，意思是把当前的在初始化中的driver传递给页面，页面才能接下去找元素，执行页面方法。

接下来我们会慢慢去修改这个POM，进行一些封装的其他方法调用。