**selentium+phantomjs**

今天学的是selentium+phantomjs，也就是控制浏览器，之前有一节课已经学过了，今天加上了phantomjs

之前已经说了selentium，是一个Web的自动化测试工具，最初是为网站自动化测试而开发的，最初是为网站自动化测试而开发的，类型像我们玩游戏用的按键精灵，可以按指定的命令自动化操作，不同是Selenium可以直接运行在浏览器上，它支持所有主流的浏览器(包括PhantomJS这些无界面的浏览器)。

PhantomJS是一个基于Webkit的"无界面"(headless)浏览器，它会把网站加载到内存并执行页面上的JavaScript，因为不会展示图形界面，所以运行起来比完整的浏览器更高效。

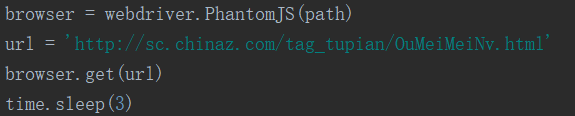
如果我们把Selenium和PhantomJS结合在一起，就可以运行一个非常强大的网络爬虫了，这个爬虫可以处理JavaScript、Cookie、headers，以及任何我们真实用户需要做的事情。

在我认为，他俩的组合可以和Scrapy媲美，当让了，Scrapy后面会学到

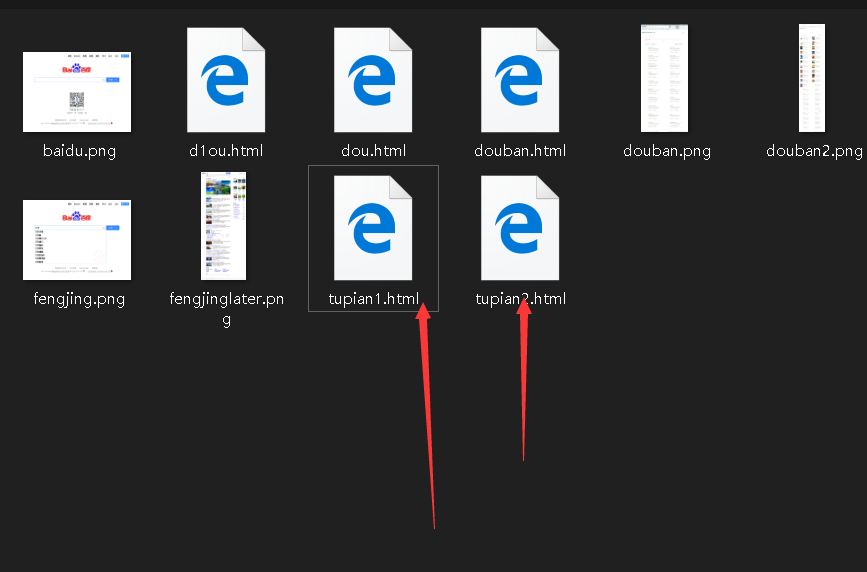
这两个其实都是一个软件，都需要下载，我已经在之前准备好了，直接用就可以，我们先看代码



分析代码：

* 我们这次爬取的是一个图片网站
* 我们先把phantomjs的路径导入
* 然后船舰浏览器对象，使用selenium里面的webdriver，
* 顺便把url写上，然后用刚才创建的浏览器对象对这个地址发送请求，然后等待3秒，方便我们截图
* 然后就是把我们获取的内容保存到本地

下面是运行结果：



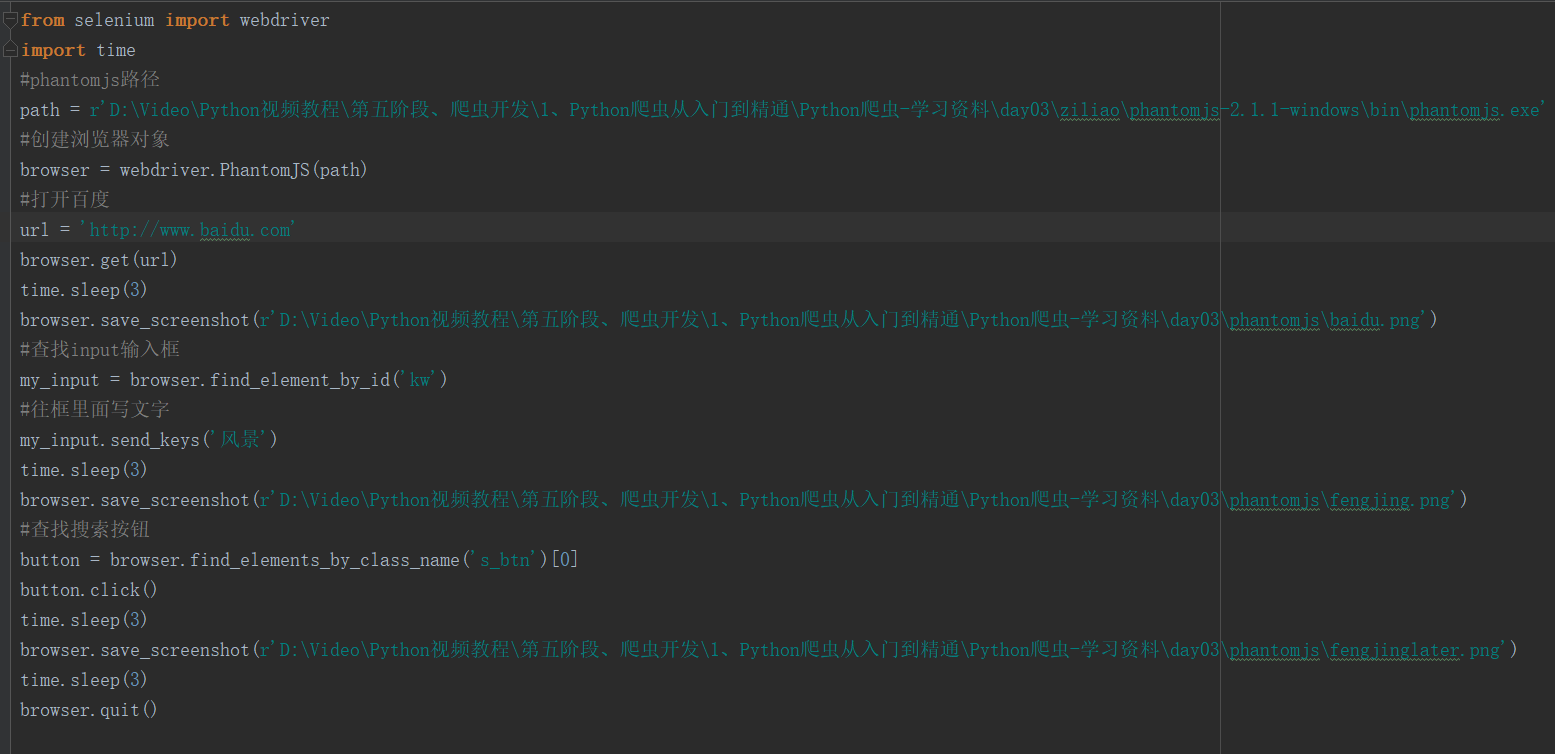
我们已经成果获取到了内容，我么打开看看



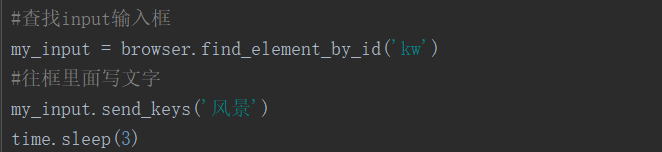
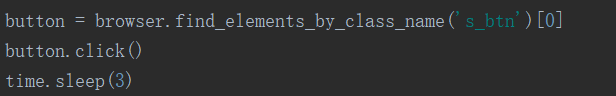
这是html格式的，看来完全没问题，这个多方便，直接控制浏览器执行代码

接下来我们再看一个示例，这次我们获取的不是html，我们这次给获取到内容之后的页面，拍摄一个快照，保存到本地，其实过程差不多，但是我们这次爬取的是百度图片的内容，这个内容需要我们自己来输入，想查询什么都可以

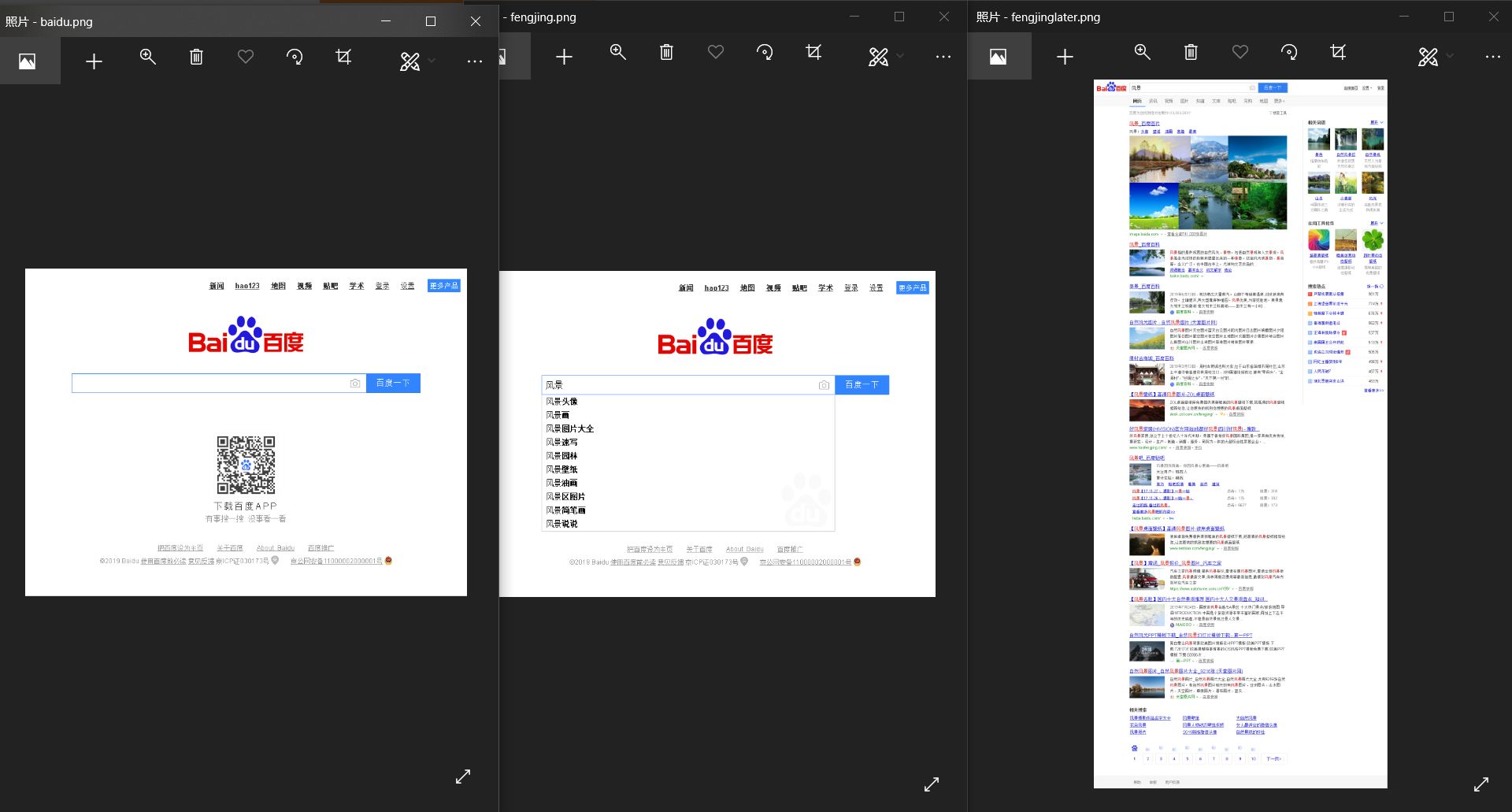
下面是代码：



分析代码：

* 首先还是导入库，然后把phantomjs的路径写上，还是创建浏览器对象，输入我们要爬取的url，用刚才的浏览器对象，对这个url发送请求
* 发送完请求，我们已经来到了百度页面，这个时候我们需要输入我们想要搜索的图片的内容，另外说一句，这些都是在后台自己执行的，是不可见的，所以我们要设置快照，
* 
* 输入完内容之后，我们需要控制浏览器点击搜索按钮
* 
* 然后不出意外我们已经到了图片的页面，接下来我们就再设置一个快照

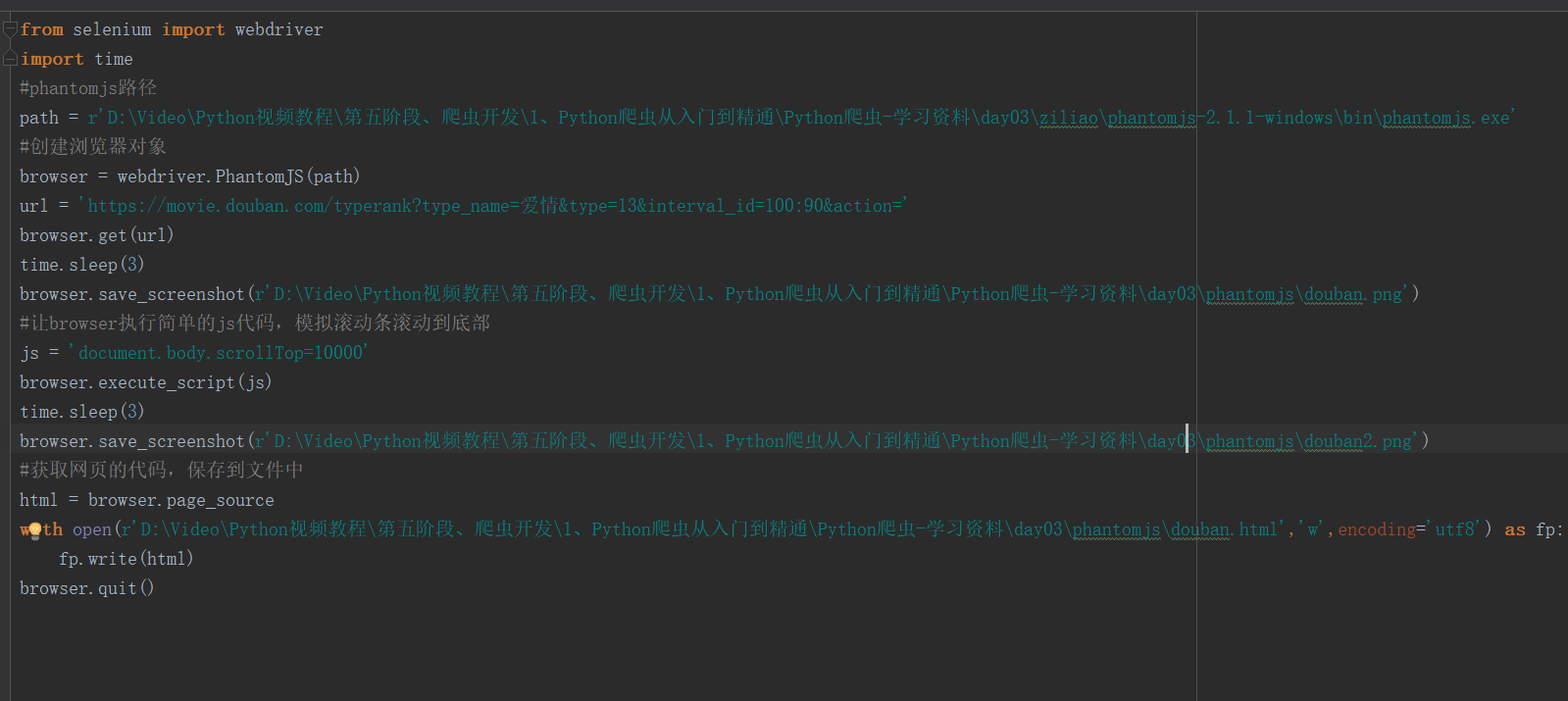
下面是代码运行结果：



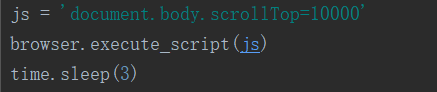
我们成功了，保存到了本地，yeah！

现在我们发现一个问题，虽然我们成果获取到了照片，但是我们只有这一页啊，还有下拉选项对不对，所以我们下一个实例就是实现下拉，这次我们爬取豆瓣电影

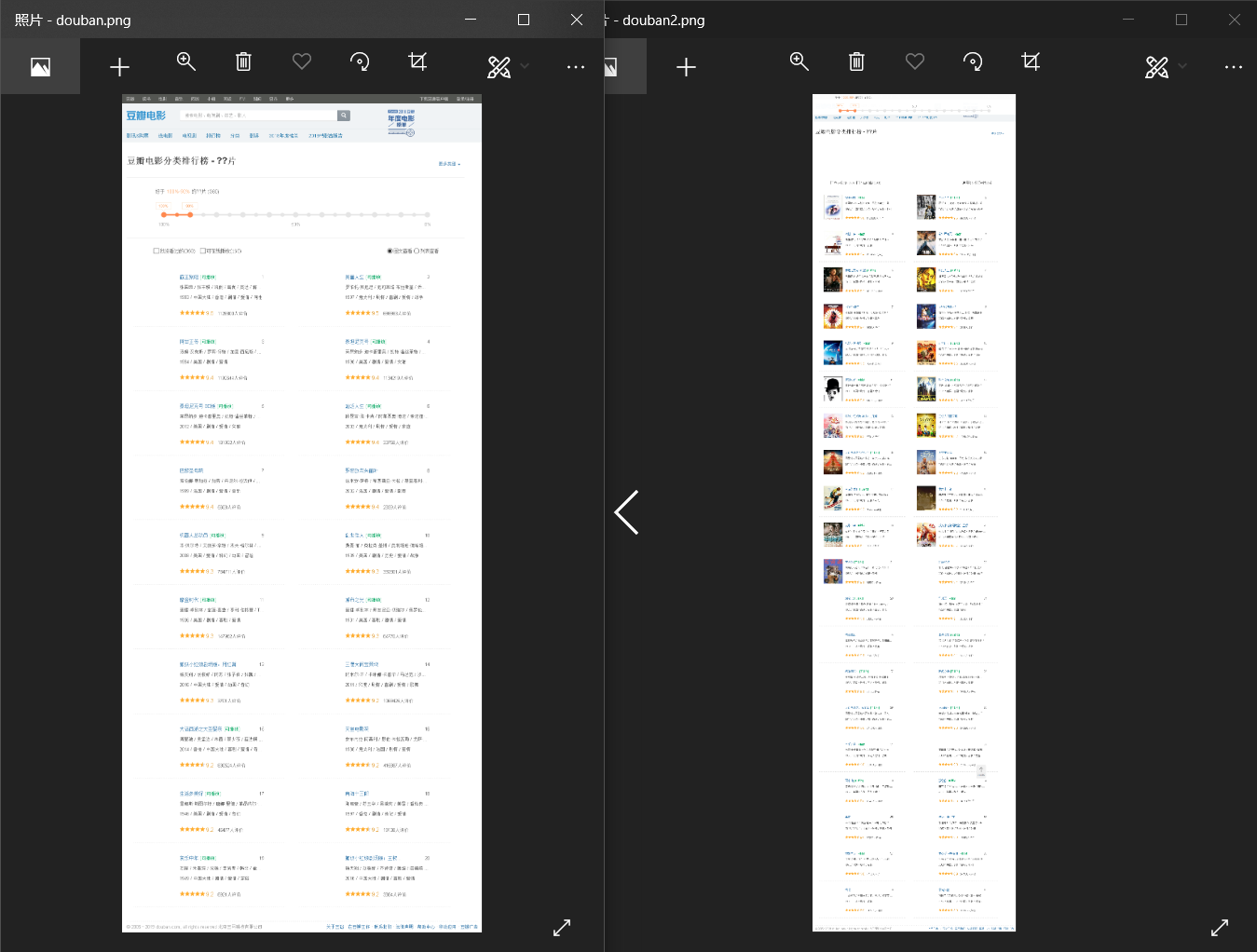
下面是代码：

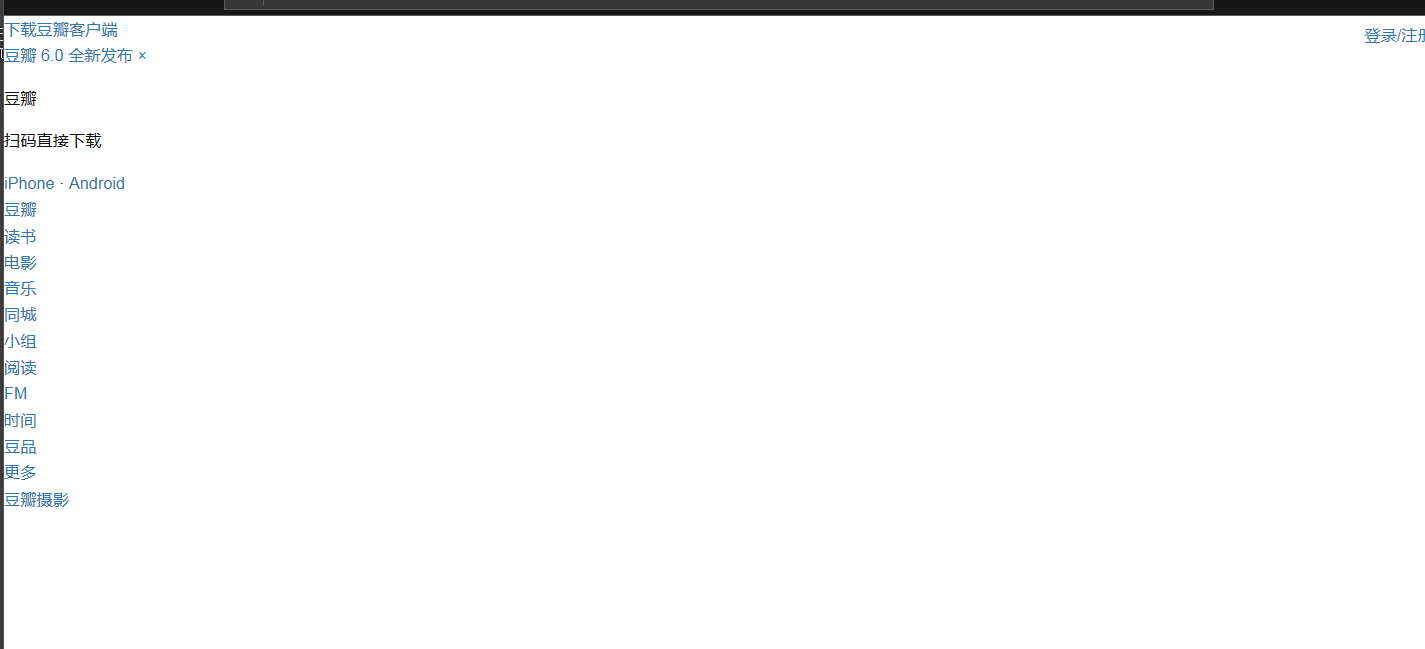


分析代码：

* 还是老规矩，写入phantomjs路径和创建浏览器对象，对这个url发送请求，先设置一个快照，其实这个东西就是个套路，和之前一样，我们框架写好，只需要把里面的东西改一改
* 然后我们已经来到了豆瓣电影这个页面，我们要实现下拉功能，所以我们让browser执行简单的js代码，模拟滚动条滚动到底部
* 
* 然后在这个时候我们需要设置一个快照，记录下这美好的一刻，我们再加一个吧，获取一下网页的代码
* 
* 我们分别把这些内容保存到本地

我们来看看运行结果





大功告成！