**懒爬取**

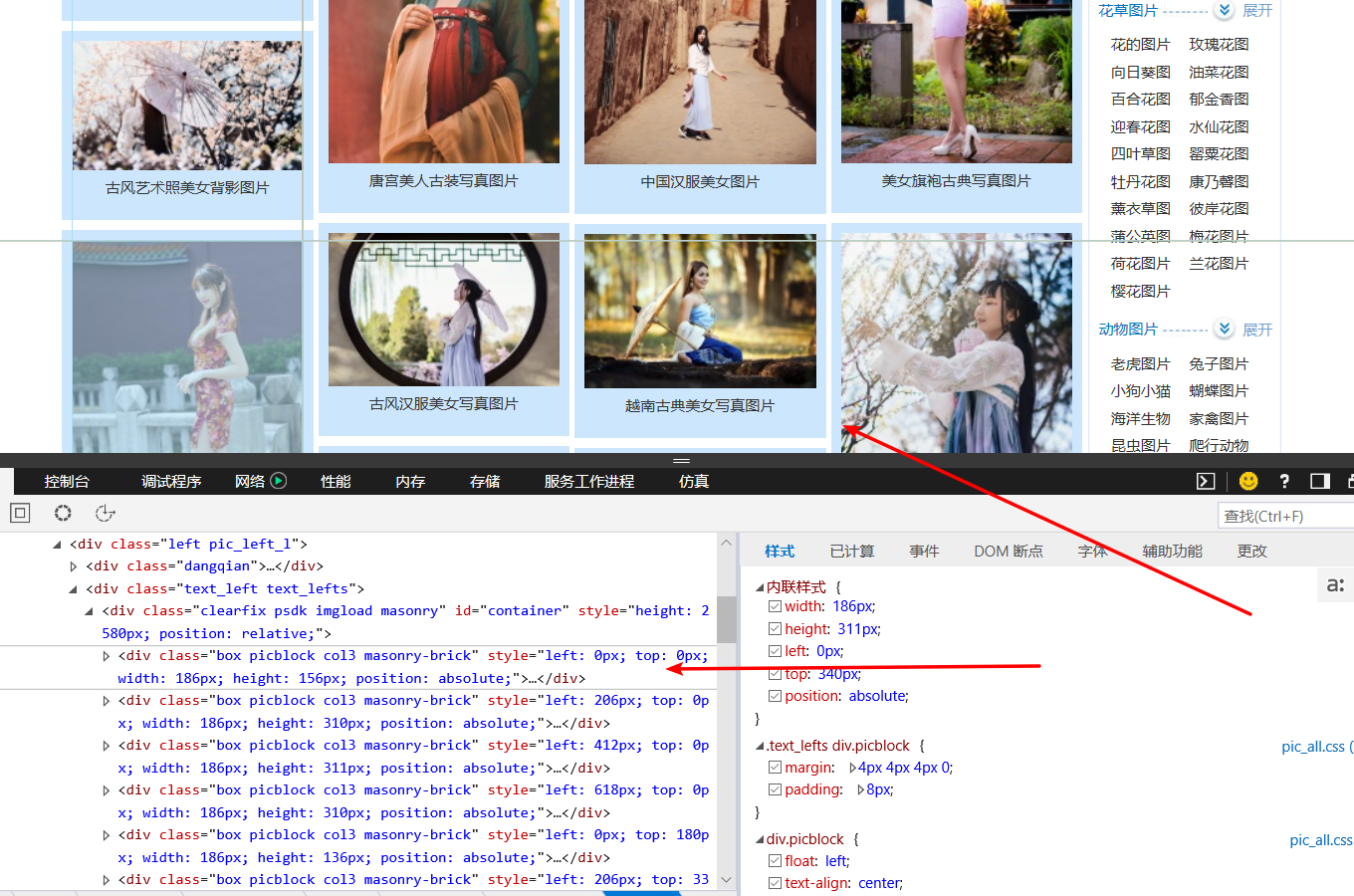
今天学习懒爬取，什么是懒爬取，这里就涉及到图片懒加载，那么什么是图片懒加载呢？

其实他就是一种网页优化技术，图片作为一种网络资源，在被请求时也与普通静态资源一样，将占用网络资源，而一次性将整个页面的所有图片加载完，将大大增加页面的首屏加载时间，为了解决这种问题，通过前后端配合，使图片仅仅在浏览器当前视窗内出现时才加载该图片，达到减少首屏图片请求数的技术就被称为图片懒加载

那么我们的网站都是如何实现图片懒加载技术呢？

我们在分析网站源码，检查我们爬取内容的位置的时候，我们都会发现，有的资源保存在img标签里，这里涉及到一个伪属性，在这个伪属性里面，存放真正的图片链接而并非直接存放到src属性中，当图片出现到页面的可视化区域中，会动态将伪属性替换成为src属性，完成图片的加载。

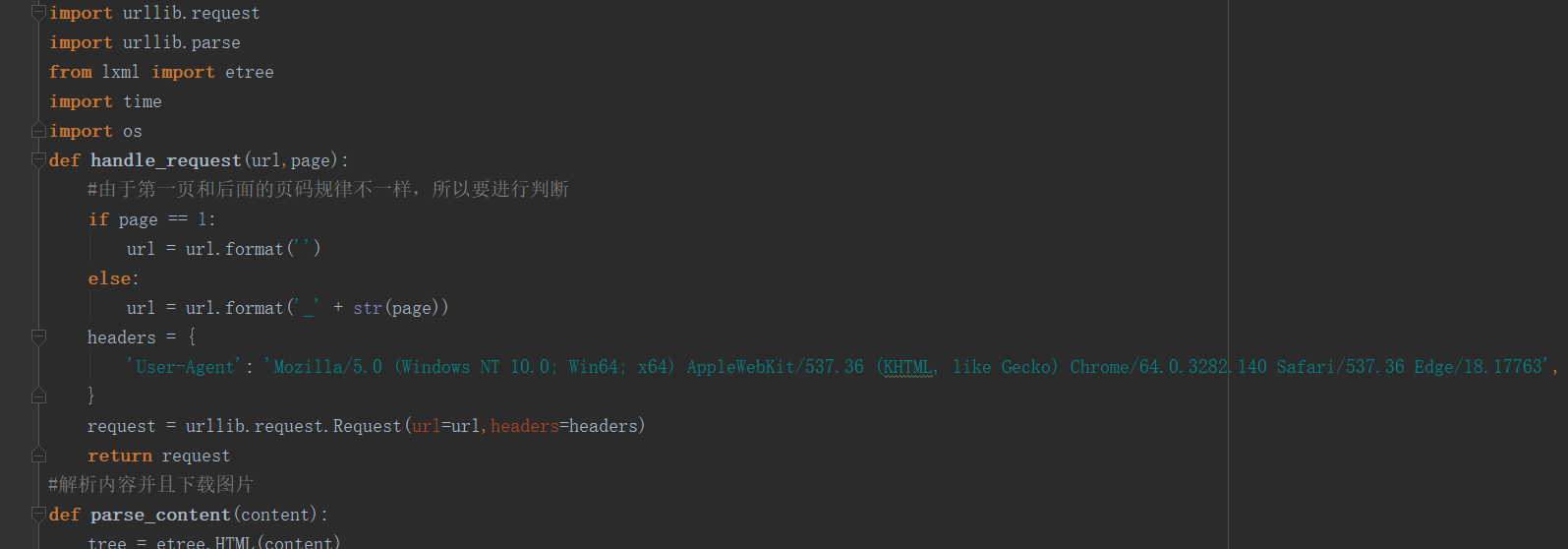
我们今天爬取的网站时站长素材，这个就用到了图片懒加载，我们先看网站

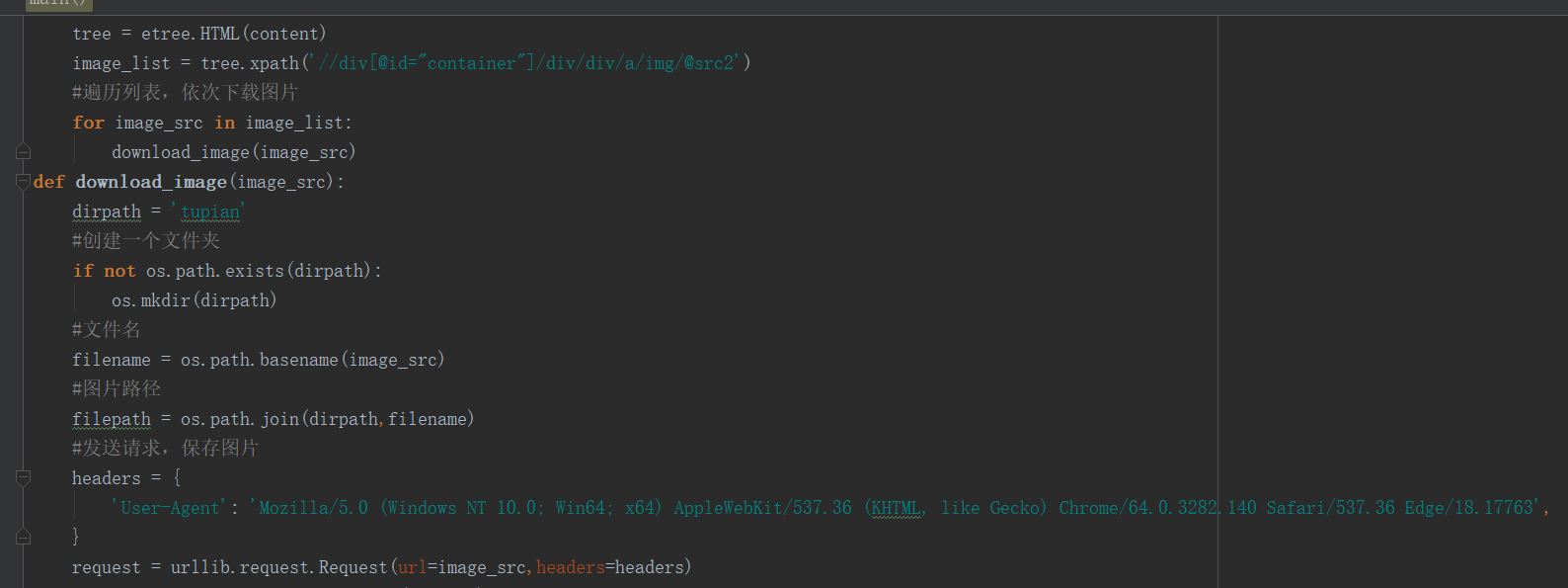


我们可以看到这些图片现在都在这里，当我们下拉菜单



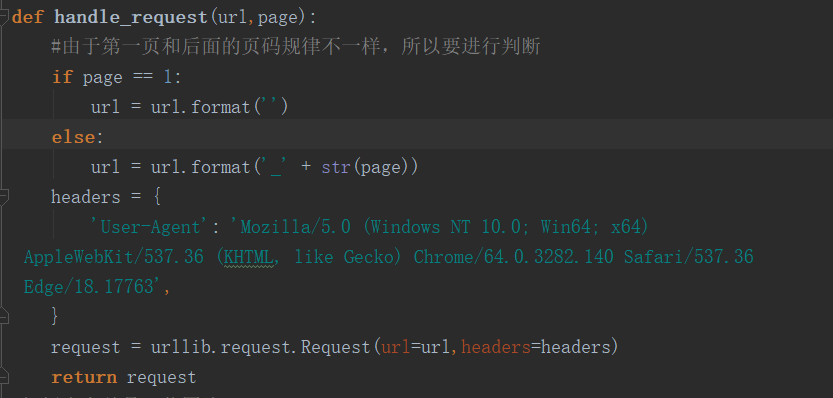
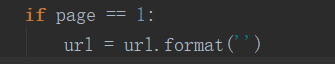
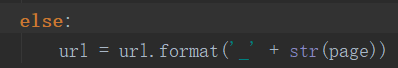
真正的src在这里

所以我们分析完网页看代码：  
 

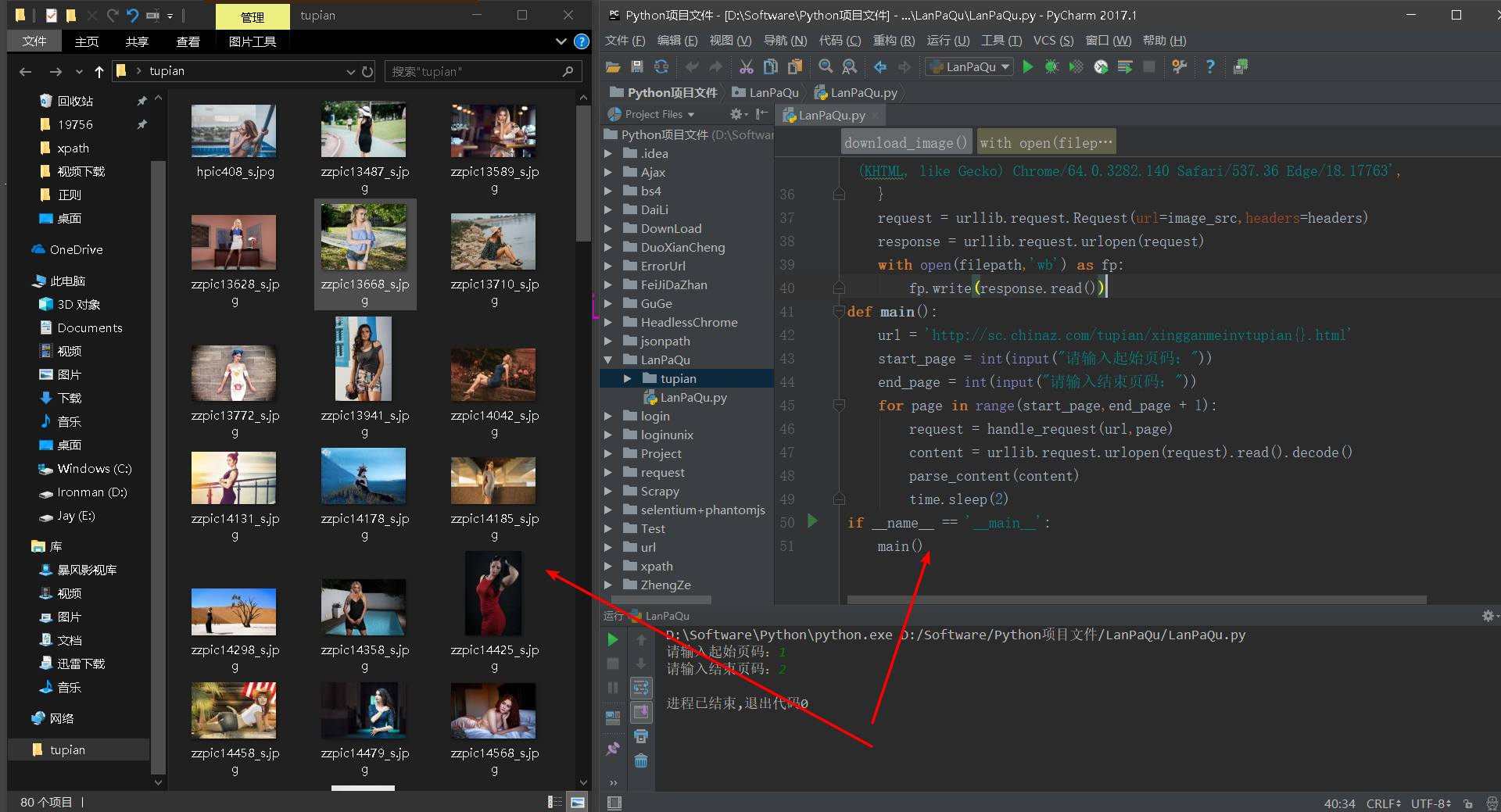




分析代码：

* 先定义主函数，在这里是url和页码，写了好多遍了，然后遍历页码
* 在这个循环里面，第一件事还是写发送请求的函数
* 
* 在这里要注意，第一页的url是这样的
* 
* 之后的是这样的
* 
* 
* 所以第一页和后面的页码规律不一样，所以要进行判断
* 当页码是第一页的时候我们url拼接为空
* 
* 之后的页码需要拼接一个下划线加上page
* 
* 这就是和之前不一样的
* 然后我们写请求头，发送请求，返回请求到主函数
* 这时我们获取响应，然后写一个专门的函数来解析响应
* 在这里我们使用xpath来获取图片的src，注意，我们这里获取的src是真正的src
* 
* 然后我们遍历列表，依次下载图片，最后保存到本地

我们来看看运行结果：



大功告成！