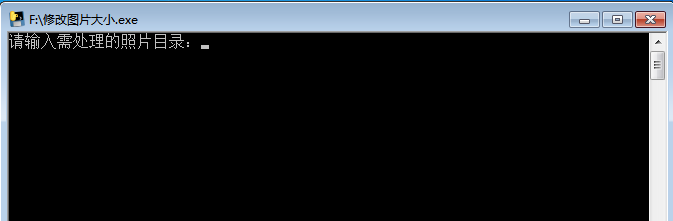
利用Python解决了一个工作问题

10月23日上午，利用Python解决了一个实际工作中的问题。

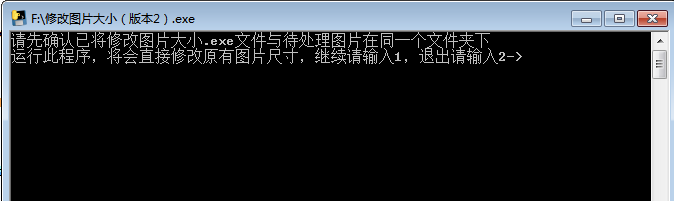
这是我学习Python以来，解决的第一个工作中的问题，写篇文章记录一下。

网上报账时，在上传附件到平台时，要求上传图片的大小不得大于1200 X 900。培训人员教的方法是利用Windows自带的画图软件修改文件大小。开学至今，由于报账不过几次，就没想过自动化的方式来进行处理。直到与部门同事聊天时，才发现这个问题修改图片大小是一个痛点，报账的附件一般是通过手机拍照上传，而手机拍摄的照片一般像素很高，必须修改。而且，必须一张张地修改。

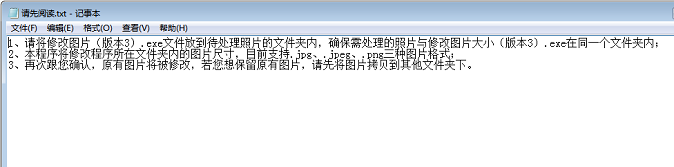
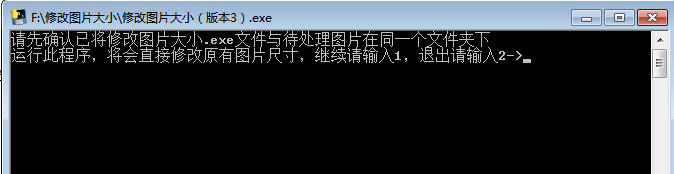
周一 ，想要用Python来解决这个问题。最初的想法是待学完GUI（学习进度较慢）之后再来编写，想了几天，今天上午，决定先干起来，代码的部分很简单，不过几十行。但是写这段程序的目的是为了减轻其他同事的工作量，而不是只是单单为自己所用。于是，通过网上搜索，找到了pyinstaller模块（利用代码pip install pyinstaller进行安装），生成了一个可执行性文件（代码：pyinstaller -F 修改图片大小.exe）。



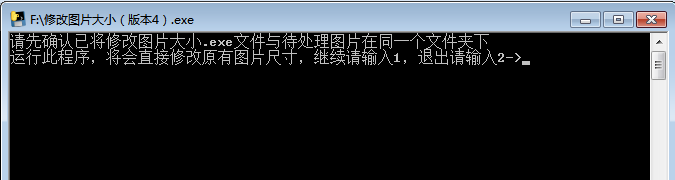
第一个版本中，考虑的是将文件夹中的图片的大小更改后，放入一个新的文件夹（也可选择原目录）。但在实际使用中，发现输入目录的方式体验不好，输入目录可能出错，而且用户的痛点是节省时间，需要的是傻瓜式的操作，提供的功能过多，反而适得其反。于是产生了第二个版本，如下：



第二个版本没有使用文档，考虑到用户可能需要保存原有图片，必须将相关问题描述清楚，而且目前只支持.jpg、.jpeg、.png三种类型的图片。所以增加了使用文档，形成了第三个版本。



满怀希望地将程序和文档打成了一个压缩包，兴冲冲地请同属帮忙测试，悲催的是，竟然发现图片大小并未变化。灰头土脸地回来检查代码，发现是if语句中的类型错误（Python默认输入的事字符串格式），重新生成EXE文件，重新测试，谢天谢地，终于通过了。



几点感悟：

1. 目前的版本在使用的过程中仍有许多限制条件，如：必须先将程序拷贝到图片所在文件夹，使用过程中被360怀疑是病毒等。
2. 行动胜于语言。先做起来，做的过程中发现了问题，再去解决它。
3. 始终以用户的角度来思考。用户体验的好坏是决定程序优劣的最好标准。