**第九章：自动化处理excel电子表格（15min）**

目标：

使用pip安装python包

使用openpyxl库来打开，修改，另存电子表格文件

使用Reference和BarChart 类来可视化电子表格数据

封装该函数然后实现批量处理待电子表格

目录

[第一节 下载openpyxl的python库并调用 1](#_Toc64556188)

[第二节 打开excel文件并输出值与一些表格的性质 2](#_Toc64556189)

[第三节 用for循环对某列或某行数据进行处理 3](#_Toc64556190)

[第四节 保存修改过部分值的excel文件 3](#_Toc64556191)

[第五节 如何加入excel的图表信息 3](#_Toc64556192)

[第六节 代码分析 4](#_Toc64556193)

## 下载openpyxl的python库并调用

##### python中pip的安装：

###### windows用户

安装地址：[https://pypi.python.org/pypi/pip#downloads](https://pypi.python.org/pypi/pip" \l "downloads)

下载并解压后找到对应文件夹复制该文件夹地址

按下win（一般为左下方第二个或第三个键）+R键打开命令行

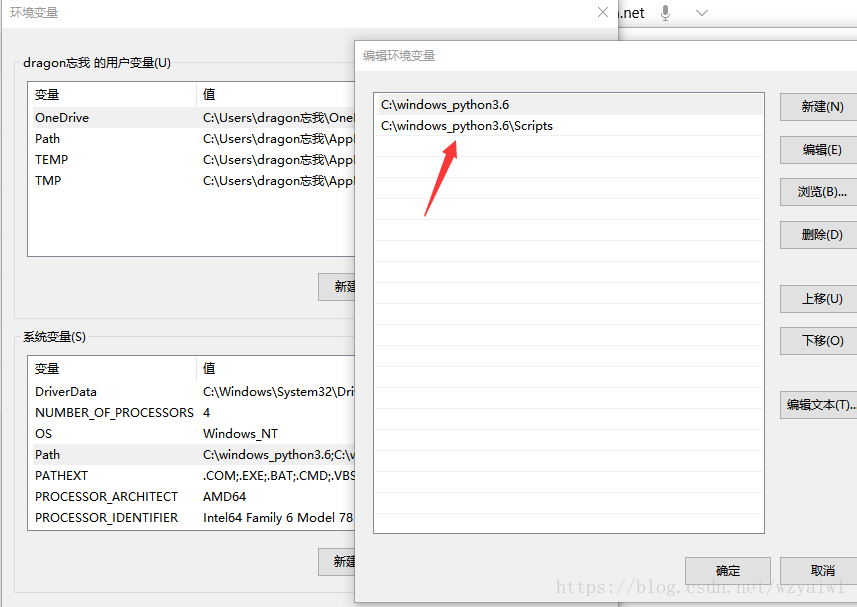
输入以下命令后按回车

cd 该文件夹地址

之后安装python setup.py install

在左下角搜索系统变量





最后为pip添加环境变量，找到python下的scrip，添加至path下，记得后面加上分号。

mac用户

打开终端（command+空格），

在终端中输入：

sudo easy\_install pip

接着根据终端提示输入密码，点击回车，系统会让你输入密码，密码是当前登录电脑的用户名的密码。

安装后在终端上输入 pip -V （V必须大写），如果显示以下代码，则代表安装成功。

代码范例：

appledeiMac:~ apple$ pip -V

pip 20.2b1 from /Library/Python/2.7/site-packages/pip-20.2b1-py2.7.egg/pip (python 2.7)

##### 用pip安装openpyxl：

如果没有anacanda等第三方python整合python附加库的话，使用pip来安装python的额外库是十分方便的，所以这里我们用pip来安装python pip库安装好后输入如下命令：（注意有的时候会提示pip需要更新，只需要根据命令行给出的指导意见把命令复制之后再输出就好）

pip install openpyxl

等待下载完成即可

##### 调用openpyxl库

openpyxl库由于名字很长，我们使用以前曾经学到过的简写进行导入

import openpyxl as xl

这样方便以后调用openpyxl库，但注意不能重名

from openpyxl.chart import BarChart, Reference

从这个库中调用两个类，方便画图时数据的选取和输出范围的确定

## 打开excel文件并输出值与一些表格的性质

将我们附加的文件放到创设这个python工程文件的文件夹下面，（注意，如果在今后想自己创建表格进行处理的时候，表格的格式应保存为xlsx而不是xls格式，现在的openpyxl库已经不支持打开过老版本（03）的电子表格文件格式xls了）使用的时pycharm的话与老师使用的相同，如果使用vs，看vs右方的工程文件管理，点击下图中的红圈图标，如果显示了test.xlsx这样的电子表格在工程文件中，证明已经引用正确。如果无法招到或者希望从别的文件夹直接引用可以使用如下命令

wb=xl.load\_workbook(r"地址\test.xlsx")

注意斜杠的方向（windows用户使用）

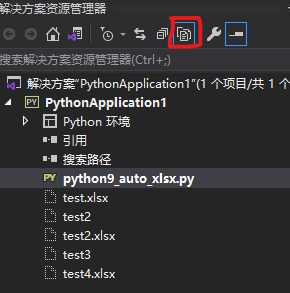
用下列代码验证是否成功引用：（把加载的xlsx文档的第一个表单命名给sheet方便引用，请注意这里的大小写敏感必须是Sheet1）

sheet=wb['Sheet1']

cell=sheet['A1']

cell=sheet.cell(1,1)

print(cell.value)



## 用for循环对某列或某行数据进行处理

批量修改表格的核心思想就在于以前学过的for in循环当中，注意这里的的范围值和“：”就可以了

## 保存修改过部分值的excel文件

wb.save("test4.xlsx")

后缀名一定要正确，这样才能方便日后打开，如果没有后缀名出现的问题将是处理的文档无法打开，必须手动指定电子表格文件进行打开

## 如何加入excel的图表信息

先使用类Reference定义可视化数据的范围

之后调用BarChart类来进行画图

values=Reference(sheet,

min\_row=2,

max\_row=sheet.max\_row,

min\_col=4,

max\_col=4)

chart= BarChart()

chart.add\_data(values)

sheet.add\_chart(chart,'e2')

BarChart提供了非常强大的画图功能，可以修改图例，颜色坐标轴等，您可以通过BarChart的说明手册进行进一步学习

## 代码分析

import openpyxl as xl

#需要使用pip安装命令行具体做法是win键+r之后输入cmd后回车，在里面使用命令语句：pip install openpyxl

#import \*\* as \*方便以后引用，但是也要注意不能与这个项目下其他库的名称重合从而引起歧义

from openpyxl.chart import BarChart, Reference

#从库中引用类

# def process\_workbook(filename):

# 补充知识：缩排快捷键（在vs和pycharm里面都使用整体选中+tab缩进一格，整体选中后+shift+tab整体反缩进一格）

# 注意缩进，在您想封装该函数的时候要把下方统一选中按下tab之后，修改文件夹名字

wb=xl.load\_workbook("test.xlsx")

# 注意原视频中老师使用的目录是该文件夹下面的

# 如果最终也无法找到该excel文件可以以下语句使用自动分配定向地址

# wb=xl.load\_workbook(r"地址\test.xlsx")

# 注意，如果您使用的是linux系统，其内部的文件加的分类符号不是"\"而是"/"请注意区分

# 无论是使用wps还是windows office都是可以的但是注意excel文件的后缀必须是xlsx

# 如果希望复用该函数到不同的地方可以使用wb=xl.load\_workbook

sheet=wb['Sheet1']

#把加载的xlsx文档的第一个表单命名给sheet方便引用

#请注意这里的大小写敏感必须是Sheet1

# cell=sheet['A1']

# cell=sheet.cell(1,1)

# 以上两个语句的意思是相同的都是把一个（1，1）单元格给了cell这个元素

# print(cell.value)

# print(sheet.max\_row)

#上面语句可以作为测试看是否成功引用了这个xlsx表格，正确输出结果是：number（第一列第一行） 12

for row in range(2,sheet.max\_row+1):#注意这里的范围是sheet.max\_row+1否则将输出不全，因为是小于该值，这个右边范围值在结果中是不输出的

# print(row) 方便测试使用

cell=sheet.cell(row,4)

#将第四列每个循环一个值的横排都把该单元格元素给cell这个变量

corrected\_price=cell.value\*0.9

corrected\_price\_cell=sheet.cell(row,4)

corrected\_price\_cell.value=corrected\_price

# 这三个变量的赋值精简为两个也可，但是命名三个变量的优点在于可以清晰地直接修改需要更正的价格的计算式子

# print("%.1f" % cell.value)

#注意由于计算机对于浮点运算的定义如果不加以控制对于小数点位数的确定会出现小数点位数不统一以及小数点位数过多的情况，这里使用格式进行限制

values=Reference(sheet,

min\_row=2,

max\_row=sheet.max\_row,

min\_col=4,

max\_col=4)

#Refercence（注意大小写）,确定了引用参数的范围。行数是从非字符的第二行开始到最后一行数字，列数是第四列

chart= BarChart()

chart.add\_data(values)

sheet.add\_chart(chart,'e2')

#BarChart提供了非常强大的画图功能，可以修改图例，颜色坐标轴等，您可以通过BarChart的说明手册进行进一步学习

#这个命令最基本的使用方法就是先将需要画图的范围值给指定画图的命令chart之后用add\_chart(图表,'元格')将图标放到e2这个单元格子内部

wb.save("test4.xlsx")

# 为了不影响以前的xlsx表格一般我们再重新找个文件地址进行保存，同样的如果你不想保存到该同样的目录文件下你可以使用这个命令

#wb.save(r"目录/test2.xlsx")

#注意这里的后缀名一定要加上，这样才能保存为指定格式的excel文件，否则需要自行选取打开方式，十分麻烦

#通过加def之后改filename可以封装这个函数只需要找到文件夹名字就可以进行统一快速处理了

#wb.save(filename)

#之前我们就已经学习过如何找到目录中的所有文件，这样我们就可以在一个目录中找到每个文件进而快速处理大批量excel文件