<https://zhuanlan.zhihu.com/p/54003662>

先说基本操作：

引入库

**import** xlwings **as** xw

打开Excel程序，默认设置：程序可见，只打开不新建工作薄

app **=** xw**.**App(visible**=**True,add\_book**=**False)

*#新建工作簿 (如果不接下一条代码的话，Excel只会一闪而过，卖个萌就走了）*

wb **=** app**.**books**.**add()

打开已有工作簿（支持绝对路径和相对路径）

wb = app.books.open('example.xlsx')

#练习的时候建议直接用下面这条

#wb = xw.Book('example.xlsx')

#这样的话就不会频繁打开新的Excel

保存工作簿

wb.save('example.xlsx')

退出工作簿（可省略）

wb.close()

退出Excel

app.quit()

引用Excel工作表，单元格

引用工作表

sht = wb.sheets[0]

#sht = wb.sheets[第一个sheet名]

引用单元格

rng = sht.range('a1')

#rng = sht['a1']

#rng = sht[0,0] 第一行的第一列即a1,相当于pandas的切片

引用区域

rng = sht.range('a1:a5')

#rng = sht['a1:a5']

#rng = sht[:5,0]

重头戏：写入数据  
(xlwings多个单元格的写入大多是以表格形式)

选择起始单元格A1,写入字符串‘Hello’

sht.range('a1').value = 'Hello'

默认按行插入：A1:D1分别写入1,2,3,4

sht.range('a1').value = [1,2,3,4]

等同于

sht.range('a1:d1').value = [1,2,3,4]

按列插入： A2:A5分别写入5,6,7,8

你可能会想

\*sht.range('a2:a5').value = [5,6,7,8]

但是你会发现xlwings还是会按行处理的，上面一行等同于

\*sht.range('a2').value = [5,6,7,8]

正确语法:

sht.range('a2').options(transpose=True).value = [5,6,7,8]

既然默认的是按行写入，我们就把它倒过来嘛（transpose），单词要打对，如果你打错单词，它不会报错，而会按默认的行来写入（别问我怎么知道的）

我们输入信息的时候往往不只是写入一行或一列，  
多行输入就要用二维列表了：

sht.range('a6').expand('table').value = [['a','b','c'],['d','e','f'],['g','h','i']]

说完了写入就该讲读取了，记住了写入，读取就简单了

读取A1:D4（直接填入单元格范围就行了）

print(sht.range('a1:d4').value)

返回的值是列表形式，多行多列为二维列表，但有一点要注意，返回的数值默认是浮点数s

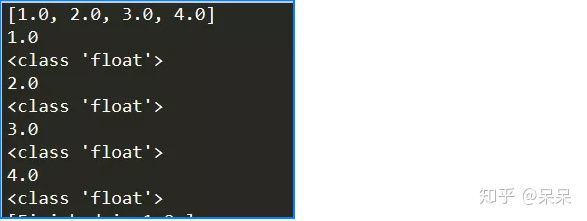
a = sht.range('a1:d1').value

print(a)

for i in a:

print(i)

print(type(i))



读取excel的第一列怎么做？

\*a = sht.range('a:a').value

print(len(a))

https://pic4.zhimg.com/80/v2-d0011c21ce362cbf15eda3c1c8a93a37_hd.png

你将会得到一个1048576个元素的列表，也就是空值也包含进去了，所以这种方法不行

思路：先计算单元格的行数(前提是连续的单元格)

rng = sht.range('a1').expand('table')

nrows = rng.rows.count

接着就可以按准确范围读取了

a **=** sht**.**range(f'a1:a{nrows}')**.**value

同理选取一行的数据也一样

ncols **=** rng**.**columns**.**count

*#用切片*

fst\_col **=** sht[0,:ncols]**.**value

好了基本操作就介绍到这里了，下次搞个实战，把xlwings好好用起来

<https://zhuanlan.zhihu.com/p/54143277>

这一期直接来实战。

比如说，我们在一个快递网站上爬取了几个快递的轨迹信息，我们需要将数据保存下来，一个常规做法是把数据保存在数据库里（Mysql，MongoDB，Redis)，另一个是用Excel的形式存下来。对于非程序员来说，后者更加普遍，毕竟Excel是世界上使用最广泛的数据分析工具（不吹不黑）。

这次我们用xlwings来实现：

1. 保存数据
2. 更新数据

假设我们爬取到了快递信息，并且已经处理成了二维列表的形式以便处理（非本文重点不细说）：

**1.保存数据**

import xlwings as xw

wb = xw.Book()

sht = wb.sheets[0]

info\_list = [['20190001','已揽收','凯撒邮局'],

['20190001','已发货','凯撒邮局'],

['20192288','已揽收','麻花镇邮局'],

['20192288','已发货','麻花镇邮局'],

['20192288','正在派送','阿里山']]

首先，写入表头，

titles = [['包裹号','状态','地点']]

sht.range('a1').value = titles

然后写入轨迹信息

sht.range('a2').value = info\_list

保存

wb.save('Track.xlsx')

这样，第一步保存数据就完成了



**2.更新数据**

我们第二天又爬取了一次信息，发现信息更新了：

[

['20190001','已揽收','凯撒邮局'],

['20190001','已发货','凯撒邮局'],

***['20190001','正在派送','王村村口'],***

***['20190001','已签收','老王家'],***

['20192288','已揽收','麻花镇邮局'],

['20192288','已发货','麻花镇邮局'],

['20192288','正在派送','阿里山'],

***['20192288','已发货','小马家']***

]

更新数据其实没什么难度，直接**覆盖写入**就好了

但是如果我想知道更新了多少条记录怎么办呢？

**将数据去重，剩下的就是更新的**

首先读取之前写入的信息：

import xlwings as xw

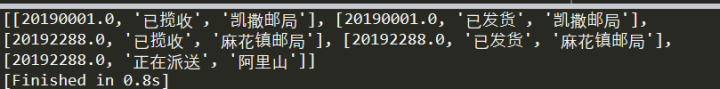
wb = xw.Book('Track.xlsx')

sht = wb.sheets[0]

first = sht.range('a2').expand('table').value

print(first)

结果如下



乍一看没什么问题，仔细一看，包裹号都成了浮点数！写入的时候是字符串，读取出来就成了浮点数，所以这时候去重，由于数据类型不一致，无法真正去重。

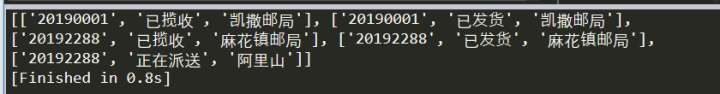
思路一：直接转化数据类型，将每个列表的第一个元素转为整数，再转为字符串

for i in first:

i[0] = str(round(i[0]))

first\_str.append(i)

print(first\_str)



思路二：如果大家对Excel熟悉的话，就会知道，在数字前面加一个英文字符的单引号（'），数字就变成文本格式了，所以我们可以在写入信息的时候加上一个单引号，这样Excel就不会乱改格式了。（以后可能会写一些Excel方面的东西）

import xlwings as xw

wb = xw.Book('Track.xlsx')

sht = wb.sheets[0]

info\_list = [["'20190001","已揽收","凯撒邮局"],

["'20190001","已发货","凯撒邮局"],

["'20192288","已揽收","麻花镇邮局"],

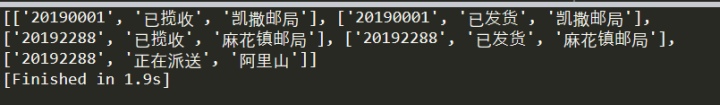
["'20192288","已发货","麻花镇邮局"],

["'20192288","正在派送","阿里山"]]

sht.range('a2').value = info\_list

values = sht.range('a2').expand('table').value

print(values)



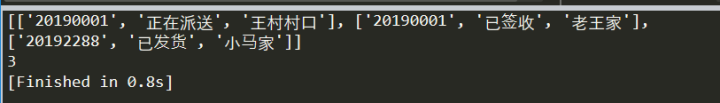
然后开始真正的去重

extra = [i for i in second if i not in first\_str]

print(extra)

print(len(extra))

结果没问题，多出三个轨迹信息



为了介绍xlwings的插入功能，我们再来设想这样一种情况：

已经有了两个包裹的轨迹情况



但是我们得到了20190001包裹的最新情况，需要更新这**一个**包裹的信息:

[

["20190001","已揽收","凯撒邮局"],

["20190001","已发货","凯撒邮局"],

["20190001","正在派送","王村村口"],

***["20190001","已签收","老王家"]***

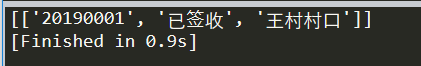
]

首先，去重

extra = [i for i in second if i not in first\_str]

print(extra)

显示要更新的就一条



读取第一列的包裹号

rng = sht.range('a1').expand('table')

nrows = rng.rows.count

row\_a = sht.range(f'a1:a{nrows}').value

找到要更新的包裹号

for i in extra:

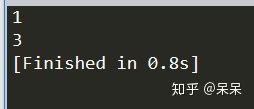
pkg = i[0]

position = row\_a.index(pkg)

print(position)

times = row\_a.count(pkg)

print(times)



position = 1 是指在第二行出现，times = 3 是指一共有3个此包裹号的信息

所以要在第五行插入

rows = position+times+1

sht.range(f'{rows}:{rows}').api.Insert()



然后再写入更新的信息

sht.range(f'a{rows}').value = extra

大功告成！保存

wb.save()

其实折腾了半天都是为了这个插入语句

sht**.**range('单元格或行列')**.**api**.**Insert()

#插入列

sht.range('a:a').api.Insert()

#插入行

sht.range('2:2').api.Insert()

#插入单元格

sht.range('b4').api.Insert()

xlwings就说到这里了，讲得还算清楚吗？