

zhangdingqu的博客

博客园 | 首页 | 新随笔 | 联系 | 管理 | 订阅 | 

随笔- 81 文章- 0 评论- 1

让Excel飞起来——xlwings

最近工作需要整理将近60位同事的周报合并到一个excel中，每周都会占用的大量的时间。希望把这个工作自动化，网上看了下基本都是用vba编程实现这一功能。由于没有使用过vba，希望使用自己会的东西去实现。
我们的周报模板大致是这个样子的：

任务类型	任务主题	任务详情	任务进度	小时	开始时间	结束时间	备注
运维	统计08	统计08 数据稽核		25%	4	2015/12/13	2015/12/13 问题1

搜了下python操作Excel的包，看到最多的是xlrd和xlwt。我的操作其实很简单就是拷贝每一个excel中的内容并粘贴到一个汇总的excel，但这两个包好像并不符合我的要求。

偶然间看到xlwings这个python包，使用这个包基本可以实现你在excel中的操作，如选中B5 到J20 这些单元格中的数据。这正是我想要的，写了个简单的脚本实现周报的整理，在这里记录下。

使用中参考了[xlwings官方文档](#)

我主要使用了WorkBook、Sheet、Range这三个。

self.wb.sheets.add('Shope Info') #这句话是添加一个工作表名哦

```
import xlwings as xw
#app_visible=False 不打开excel窗口
wb = xw.Workbook(path, app_visible=False)
detail_sheet = xw.Sheet(unicode("任务详细信息", "utf-8"))
#获取有数据的最大行数，由于我们单个excel数据不会超过几十行，我使用的是B4:I40
last_row_index = xw.Range(detail_sheet, "B4").table.last_cell.row
#确定我们选取的数据范围，如：B4:I20
rg = "B4:I" + str(last_row_index)
#获取划定范围单元格中的数据
work_detail = xw.Range(detail_sheet, rg).value
#关闭打开的excel
wb.close()
```

接下来就是将work_detail 获取到的数据拷贝到我们汇总的excel中，代码几乎一致

```
target_excel_path = u"F:\经分应用产品线小组周报模板.xlsx"
wb_target = xw.Workbook(target_excel_path, app_visible=False)
cr_sheet = xw.Sheet(u"任务详细信息")
#清空内容,单不清空格式
xw.Range(cr_sheet, "B4:I200").clear_contents()
xw.Range(cr_sheet, "B4").table.value = work_detail
wb_target.save(u"F:\经分应用产品线小组周报.xlsx")
wb_target.close()
```

(PS 后面希望加入自动添加数据透视图的功能，暂时还没找到这方面的接口)

好了，以后只需要run一下python脚本，自动整理并且发送邮件给负责人。
搞定。。。。。。

昵称: zhangdingqu

园龄: 1年8个月

粉丝: 0

关注: 1

+加关注

2019年12月						
日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

我的标签

Excel(1)

fiddler(1)

opencv(1)

pygame(1)

Python(1)

python 自定义函数(1)

Python3变量编码格式化(1)

python3读取csv文件(1)

python查找重复元素(1)

Python读写Excel文件(1)

更多

随笔分类

C++(6)

crx备份(1)

GitHub(1)

javascript

linux相关(1)

MongoDB(1)

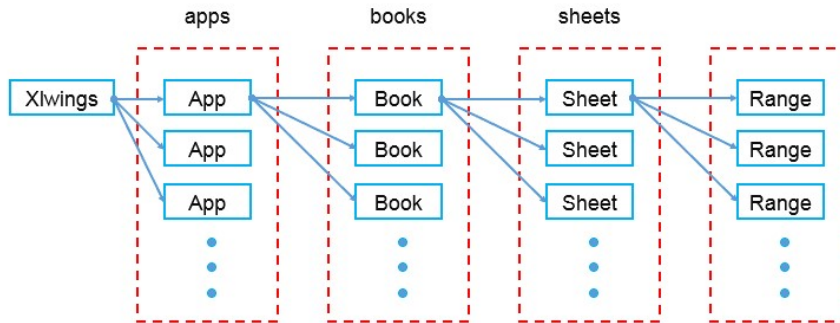
PyQT(5)

Python class模块(3)

python系统学习笔记(31)

selenium选择器(2)

1. xlwings基本对象



xlwings基本对象

App相当于Excel程序，Book相当于工作簿。N个Excel程序则由apps表示，N个工作簿由books表示。

1. 对工作簿的操作

```
# 导入xlwings模块
import xlwings as xw
# 新建工作簿，这里默认为新建了一个App，即打开Excel程序，并新建一个工作簿
xw.Book()
# 引用工作簿
xw.Book('工作簿3')
# 引用工作簿也可以用books
xw.books['工作簿3']
# 打开已有工作簿
xw.Book(r'e:\myworkbook.xlsx')
```

2. 对sheet以及Range的操作

```
import xlwings as xw
wb=xw.Book()
# 引用名字为"sheet2"的sheet
sht=wb.sheets['sheet2']
# 引用"sheet2"中的B1单元格
rng=sht.range('B1')
# 将B1单元格的值改为666
rng.value=666
```

数据结构

1. 一维数据

python的列表，可以和Excel中的行列进行数据交换，python中的一维列表，在Excel中默认为一行数据

```
import xlwings as xw
sht=xw.sheets.active
# 将1, 2, 3分别写入了A1, B1, C1单元格中
sht.range('A1').value=[1,2,3]
# 将A1, B1, C1单元格的值存入list1列表中
list1=sht.range('A1:C1').value
# 将1, 2, 3分别写入了A1, A2, A3单元格中
sht.range('A1').options(transpose=True).value=[1,2,3]
# 将A1, A2, A3单元格中值存入list1列表中
list1=sht.range('A1:A3').value
```

2. 二维数据

python的二维列表，可以转换为Excel中的行列。二维列表，即列表中的元素还是列表。在Excel中，二维列表中的列表元素，代表Excel表格中的一列。例如：

```
# 将a1,a2,a3输入第一列，b1,b2,b3输入第二列
list1=[['a1','a2','a3'],['b1','b2','b3']]
sht.range('A1').value=list1
```

	A	B
1	a1	b1
2	a2	b2
3	a3	b3
4		

运行结果

Windows系统相关(13)

个人代码xiapi(1)

远程桌面(2)

随笔档案

2019年9月(2)

2019年7月(1)

2019年6月(2)

2019年5月(2)

2019年4月(4)

2019年3月(2)

2019年2月(2)

2019年1月(3)

2018年11月(3)

2018年10月(19)

2018年9月(25)

2018年8月(11)

2018年7月(5)

最新评论

1. Re:识别缩略图加载

--zhangdingqu

阅读排行榜

1. 我们无法创建新的分区 也找不到

现有的分区(21950)

2. Python 向列表中添加元素(556

1)

3. python 查看文件名和文件路径(5

218)

4. pyqt5安装命令(4567)

5. windows远程以及文件共享方法

总结(3550)

评论排行榜

1. 识别缩略图加载(1)

推荐排行榜

1. Python while循环实现重试(1)

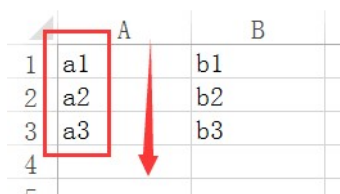
Copyright © 2019 zhangdingqu
Powered by .NET Core 3.1.0 on Linux

```
# 将A1: B3的值赋给二维列表list1
list1=sht.range('A1:B3').value
```

3. Excel中区域的选取表格

选取第一列

```
rng=sht.range('A1').expand('down')
rng.value=['a1','a2','a3']
```

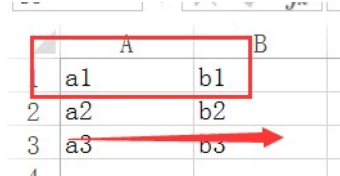


	A	B
1	a1	b1
2	a2	b2
3	a3	b3
4		

第一列

选取第一行

```
rng=sht.range('A1').expand('right')
rng.value=['a1','b1']
```

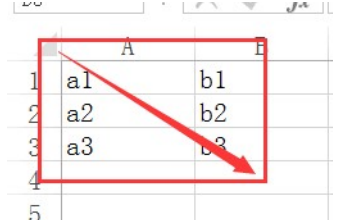


	A	B
1	a1	b1
2	a2	b2
3	a3	b3

第一行

选取表格

```
rng=sht.range('A1').expand('table')
rng.value=[['a1','a2','a3'], ['b1','b2','b3']]
```



	A	B
1	a1	b1
2	a2	b2
3	a3	b3
4		
5		

表格

结束

- 本文首先简单介绍了，通过Python运用xlwings模块，对Excel工作簿、工作表和单元格的操作。
- 再介绍了xlwings，如何对Excel表格的行和列写入和引用数据
- 通过二维或者一维列表对Excel工作表的数据进行写入或者引用，既简化了代码，也提高了程序的效率。因为python读写Excel表格的速度，原小于python程序自身的运行速度。

=====

xlwings如何设置字体、颜色等属性

最近使用python在写一个小软件，中间需要处理excel文件，所以了解了一下，使用python来操作excel文件的模块有很多，参考了^[1]中的模块对比：

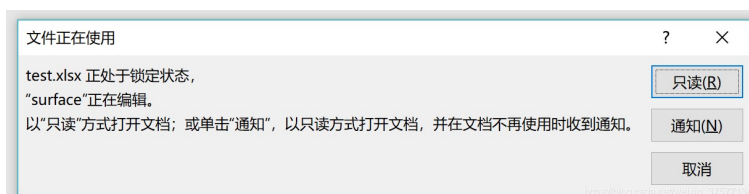


于是果断选择了“xlwings”作为我操作Excel的模块。不得不说，xlwings是一个非常好的模块，根据网上的教程以及官方文档^[2]我很快完成了数据读写部分的程序，而且测试了一下，整体读写的速度都很快。

但是在测试的过程中，遇到的第一个问题来了。

xlsx文件被锁定了怎么办？

下图就是我在debug完了以后，再次打开我的xlsx文件想去看看效果的时候弹出来的通知



查阅资料后，发现是因为我们在debug过程中结束程序后，没有执行到app.quit()语句来关闭xlwings创建的app，所以后台这个xlsx进程并没有关闭。



所以解决这个问题的方法很简单，打开任务管理器，找到上图进程，然后结束进程即可。

xlwings如何设置字体、颜色等属性

这部分是今天主要解决的问题。目前网上我所看到的所有关于xlwings的资料都没有提到设置字体的方面，甚至我在官方文档^[2]中也没有看到关于设置字体的函数，在贴吧里还看到有兄弟说xlwings不能修改字体：



正好我的软件里面需要修改xlsx文件中的字体及其颜色，而且我的软件已经使用xlwings模块写了很大一部分了，再换模块显然是不划算的，所以我决定深入研究一下。

我在使用xlwings完成按列降序排列的时候参考了@RobbenEmi 的博文^[3]，非常感谢@RobbenEmi，虽然他的博文很短，但是他一定是做过深入研究的。

受@RobbenEmi 博文^[3]的启发，我感觉突破口可能在xlwings的api方法上，于是我返回官方文档^[2]重新查找相应的函数方法，如下图所示。



看到这句话，再结合@RobbenEmi 的博文^[3]，我去微软官网查看了一下office VBA参考中的Font.Color方法

Font.Color 属性 (Excel)

2017/06/08 • 作者

返回或设置对象的主要颜色，如注释部分中的表格所示。使用 RGB 函数可创建颜色值。读/写 Variant。

语法

expression. Color

expression: 返回 Font 对象的表达式。

备注

Object	Color
Border	边框的颜色。
Borders	一个区域的所有四条边的颜色。如果四个边框不是同一种颜色，则 Color 返回的是 0（零）。
Font	字体的颜色。
Interior	单元格底纹的颜色或图形对象的填充颜色。
Tab	选项卡的颜色。

再结合xlwings官方文档中说的使用api()方法返回的是pywin32或appscript对象，我在网上查阅了pywin32如何操作xlsx文件，参考了@帅胡 的博文^[4]，我实现了使用xlwings模块设置字体、颜色、加粗等属性。

如下源码展示如何获取相应的属性。

```
# coding: utf-8

import xlwings as xw

app=xw.App(visible=False, add_book=False)
filepath = '../data/test.xlsx'
wb=app.books.open(filepath)
```

```

sht = wb.sheets('Sheet1')
font_name = sht.range('A1').api.Font.Name           # 获取字体名称
font_size = sht.range('A1').api.Font.Size           # 获取字体大小
bold = sht.range('A1').api.Font.Bold                # 获取是否加粗, True--加粗, False--未加粗
color = sht.range('A1').api.Font.Color              # 获取字体颜色
print(font_name)
print(font_size)
print(bold)
print(color)
wb.save()
wb.close()
app.quit()

```

打印的结果如下

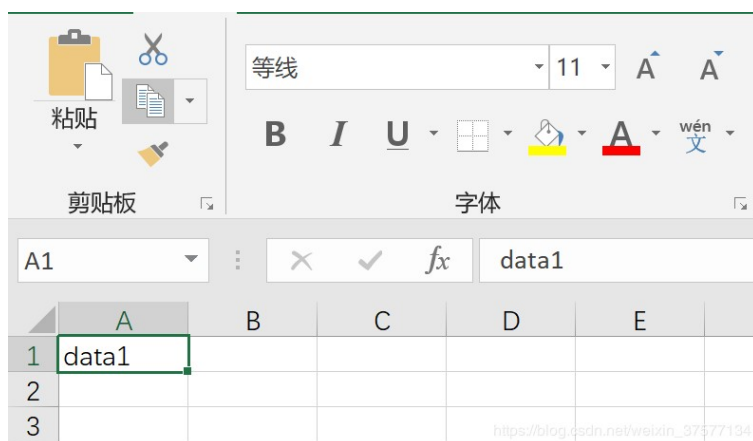
```

C:\Users\surface\AppData\Local\Programs\Python\Python36\python.exe C:/Users/surface/Desktop/FangYuanScoreSystem/src/test.py
等线
11.0
False
0.0

Process finished with exit code 0

```

相应xlsx文件中的信息如下



xlwings模块获取结果与实际xlsx文件字体属性吻合，因为xlwings模块相同的语句既可以获取property又可以set property，所以设置相应属性的源码如下。

```

# coding: utf-8

import xlwings as xw

app=xw.App(visible=False,add_book=False)
filepath = '../data/test.xlsx'
wb=app.books.open(filepath)
sht = wb.sheets('Sheet1')
font_name = sht.range('A1').api.Font.Name           # 获取字体名称
font_size = sht.range('A1').api.Font.Size           # 获取字号
bold = sht.range('A1').api.Font.Bold                # 获取是否加粗, True--加粗, False--未加粗
color = sht.range('A1').api.Font.Color              # 获取字体颜色
print(font_name)
print(font_size)
print(bold)
print(color)
print('-----设置-----')
sht.range('A1').api.Font.Name = 'Times New Roman'   # 设置字体为Times New Roman
sht.range('A1').api.Font.Size = 15                  # 设置字号为15
sht.range('A1').api.Font.Bold = True                # 加粗
sht.range('A1').api.Font.Color = 0x0000ff           # 设置为红色RGB(255,0,0)
font_name = sht.range('A1').api.Font.Name           # 获取字体名称
font_size = sht.range('A1').api.Font.Size           # 获取字体大小
bold = sht.range('A1').api.Font.Bold                # 获取是否加粗, True--加粗, False--未加粗
color = sht.range('A1').api.Font.Color              # 获取字体颜色
print(font_name)
print(font_size)
print(bold)
print(color)

```

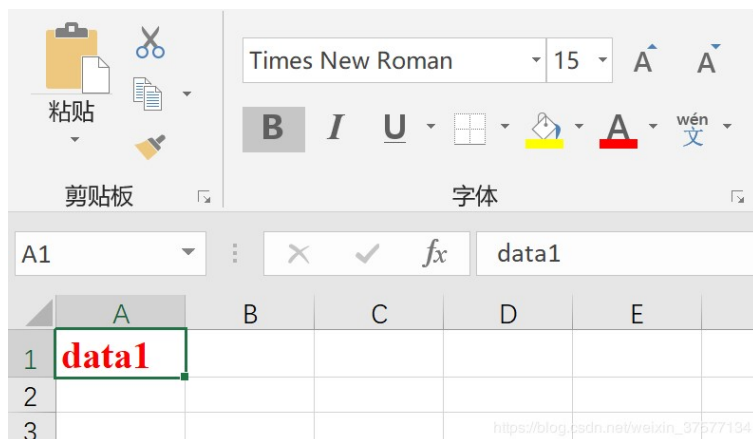
```
wb.save()
wb.close()
app.quit()
```

打印的结果如下

```
C:\Users\surface\AppData\Local\Programs\Python\Python36\python.exe C:/Users/surface/Desktop/FangYuanScoreSystem/src/test.py
等线
11.0
False
0.0
-----设置-----
Times New Roman
15.0
True
255.0

Process finished with exit code 0
```

相应xlsx文件中的信息如下



现在你可以使用xlwings自由地设置你想要的任何字体属性了！

细心的你一定会观察到，我这段设置字体属性的源码中，设置字体颜色的那句代码的赋值，并不是红色RGB(255, 0, 0)的十六进制6位颜色码，这是怎么回事呢？

在下一篇博文（关于颜色码的几种常用表示及其相互转换）中，我会讨论RGB颜色码的一些问题，相信你会在下篇博文找到答案。

补充资料

获取工作表的总行数和总列数

```
workbook=xw.Book(r'path')
workbook.sheets[1].range(1, 1).expand().shape
返回: (25087, 3)
```

```
In[53]: workbook.sheets[1].range(1, 1).expand().shape
Out[53]: (25087, 3)
```

posted @ 2019-05-13 16:47 zhangdingqu 阅读(1167) 评论(0) 编辑 收藏
刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)。

【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【推荐】腾讯云热门云产品限时秒杀，爆款1核2G云服务器99元/年！

【推荐】阿里云双11返场来袭，热门产品低至一折等你来抢！

【推荐】百度智能云岁末感恩季，明星产品低至1元新老用户畅享

【活动】京东云服务器_云主机低于1折，低价高性能产品备战双11

【活动】ECUG For Future 技术者的年度盛会（杭州，1月4-5日）

