目录

Python表格处理：CSV和Excel

- 最新版本： `v1.0`
- 更新时间： `20210413`

简介

整理如何用Python进行表格处理，尤其是csv和excel文件的相关操作。先概述csv和Excel操作的相关的库有哪些以及各自大概特点；再去详细介绍，用自带的csv库去读取和创建保存csv文件，以及封装出常用函数；以及处理Excel的，好用的openpyxl库，其常用的功能点、部分常用功能已封装成独立函数、通过具体例子来说明如何使用、以及常见的一些问题；介绍了xlutils的xlrd和xlwt的下载和安装，基本的操作，比如新建excel文件、追加和写入新的数据等，以及常见问题及解决办法；概述了pandas的操作csv和excel的相关函数；介绍了用win32com.client去处理Excel的图表。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

Gitbook源码

- crifan/python_process_csv_excel: Python表格处理：CSV和Excel

如何使用此Gitbook源码去生成发布为电子书

详见：crifan/gitbook_template: demo how to use crifan gitbook template and demo

在线浏览

- Python表格处理：CSV和Excel book.crifan.com
- Python表格处理：CSV和Excel crifan.github.io

离线下载阅读

- Python表格处理：CSV和Excel PDF
- Python表格处理：CSV和Excel ePub
- Python表格处理：CSV和Excel Mobi

版权说明

鸣谢

感谢我的老婆**陈雪**的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 `crifan` 还写了其他 `100+` 本电子书教程，感兴趣可移步至：

crifan/crifan_ebook_readme: Crifan的电子书的使用说明

概述

- `csv`
- `Excel`

用Python处理 `csv` 和 `Excel` 的常见库有：

- `csv` 库
  - Python内置库： `csv`
  - `pandas`
    - 读： `pandas.read_csv`
    - 写： `pandas.DataFrame.to_csv`
- `Excel` 库
  - 读写文件的库
    - `openpyxl`
      - 功能强大，支持设置背景色等样式的细节设置
      - 支持新的（ `Excel 2010` 之后的）格式： `.xlsx`
    - `xlutils` ：
      - 概述
        - 整合了 `xlrd` 和 `xlwt` ，额外加 `copy` 等辅助功能
        - 只支持Excel旧格式： `.xls`
      - 相关库
        - 读： `xlrd`
        - 写： `xlwt`
    - `pandas`
      - 读： `pandas.read_excel`
      - 写： `pandas.DataFrame.to_excel`
    - 其他
      - `xlsxwriter`
        - An alternative package for writing data, formatting information and, in particular, charts in the Excel 2010 format (ie: .xlsx)
      - `pyxlsb`
        - This package allows you to read Excel files in the xlsb format.
      - `pylightxl`
        - This package allows you to read xlsx and xlsm files and write xlsx files.
  - 自动化操作的库
    - 说明：对于Excel文件的自动化操作 =对标旧的Excel（其实是 `Microsoft` 的 `Office` 的） `VBA` 脚本 = 英文称: `Excel add-ins`

            - 注意：需要系统中已安装 `Excel` 软件
        - 库
            - `PyXLL`
                - 概述
                - PyXLL is a commercial product that enables writing Excel add-ins in Python with no VBA. Python functions can be exposed as worksheet functions (UDFs), macros, menus and ribbon tool bars.
                - 主页
                - https://www.pyxll.com/
            - `xlwings`
                - 概述
                - xlwings is an open-source library to automate Excel with Python instead of VBA and works on Windows and macOS: you can call Python from Excel and vice versa and write UDFs in Python (Windows only). xlwings PRO is a commercial add-on with additional functionality.
                - 主页
                - https://www.pyxll.com/

如何选择

- 根据不同情况，选择合适的解析 `csv` 和 `Excel` 的Python库
    - 数据量不大的 + 简单的 csv文件：Python自带的 `csv` 库
    - 数据量不大的 + Excel旧文件 `.xls` + 操作不复杂： `xlutils` (`xlrd` + `xlwt`)
    - 数据量不大的 + Excel新格式 `.xlsx` + 操作复杂 + 能设置样式： `openpyxl`
    - 数据量较大的 `csv` 或 `Excel` ，主要用于数据处理和计算的： `pandas`

处理csv

已把 `csv` 处理相关代码，整理成独立函数。

最新代码详见：

- https://github.com/crifan/crifanLibPython/blob/master/python3/crifanLib/crifanCsv.py
  - https://github.com/crifan/crifanLibPython/blob/master/crifanLib/demo/crifanCsvDemo.py

下面分别介绍：

python读取 `csv` 文件

核心代码 `loadCsvFromFile` ：

```
import codecs
import csv

with codecs.open(csvFilePath, "r", encoding="utf-8") as csv
    csvReader = csv.reader(csvFp)
    csvHeaders = next(csvReader) # <class 'list'>: ['url',
    for eachRowList in csvReader:
        print("eachRowList=%s" % eachRowList)
        # eachRowList=['https://car.autohome.com.cn/pic/se
```

已封装成函数：

```
def loadCsvFromFile(csvFilePath, fileEncoding="utf-8", isRe
    """read data from csv file

    Args:
        csvFilePath (str): full csv file path
        fileEncoding (str): file encoding, default to 'utf-
        isReturnDictList (bool): return data is row dict li
    Returns:
        isReturnDictList=True  -> csv row dict list
        isReturnDictList=False -> (csv header list, csv row
    Raises:
    """
    csvDictList = []

    csvHeaderList = []
    csvRowListList = []

    with codecs.open(csvFilePath, "r", encoding=fileEncodin
        csvReader = csv.reader(csvFp)
        csvHeaderList = next(csvReader)
        print("csvHeaderList=%s" % csvHeaderList)
        # <class 'list'>: ['url', '品牌', '子品牌', '车型', '
        # ['appName', 'pkgName', 'authorName', 'categoryNam
        for eachRowList in csvReader:
            # print("eachRowList=%s" % eachRowList)
            # eachRowList=['https://car.autohome.com.cn/pic
            # eachRowList=['传奇世界手游', 'com.tencent.cqsj'
            csvRowListList.append(eachRowList)

    if isReturnDictList:
        for eachRowList in csvRowListList:
            curRowDict = {}
            for curIdx, curHeader in enumerate(csvHeaderLis
                curRowValue = eachRowList[curIdx]
                curRowDict[curHeader] = curRowValue

            csvDictList.append(curRowDict)

        return csvDictList
    else:
        return csvHeaderList, csvRowListList
```

调用：

```
resultRowDictList = loadCsvFromFile(InputYingYongBaoSearchF
```

新建=写入 csv文件

```
rowValueList = [mongoId, invalidReason.name, sentence]
with codecs.open(gCurInvalidSentenceFullFilename, "a", enc
    csvListWriter = csv.writer(csvFp)
    csvListWriter.writerow(rowValueList)
```

或：用 `dict` 去保存值（需要指定 `header` ）

```
gInvalidSentenceCsvHeaders = ['MongoID', 'Reason', 'Senten

rowDict = {
    "MongoID": mongoId,
    # "Reason": str(invalidReason),
    "Reason": invalidReason.name,
    "Sentence": sentence
}

with codecs.open(gCurInvalidSentenceFullFilename, "a", enc
    csvDictWriter = csv.DictWriter(csvFp, fieldnames=gInva
    csvDictWriter.writerow(rowDict)
```

效果：

`saveToCsvByDictList` 和 `saveToCsvByHeaderAndList`

后来已封装成独立函数

代码：

```
import codecs
import csv

def saveToCsvByDictList(csvDictList, outputFilePath):
    # generate csv headers from dict list
    firstItemDict = csvDictList[0]
    csvHeaders = list(firstItemDict.keys())
    with codecs.open(outputFilePath, "w", "UTF-8") as outCs
        csvDictWriter = csv.DictWriter(outCsvFp, fieldnames

        # write header by inner function from fieldnames
        csvDictWriter.writeheader()

        for eachRowDict in csvDictList:
            csvDictWriter.writerow(eachRowDict)

def saveToCsvByHeaderAndList(csvHeaderList, csvRowListList,
    with codecs.open(outputFilePath, "w", "UTF-8") as outCs
        csvWriter = csv.writer(outCsvFp)

        # write header from list
        csvWriter.writerow(csvHeaderList)

        # type1: write each row
        # for eachRowList in csvRowListList:
        #   csvWriter.writerow(eachRowList)

        # type2: write all rows
        csvWriter.writerows(csvRowListList)
```

writeheader是后来新增的

`writeheader()` ：是 `python 2.7` 和 `Python 3.2` 之后才新增的

- 当然也可以自己写：

  ```
  dw.writerow( dict((f,f) for f in dr.fieldnames) )
  output.writerow(dict(zip(dr.fieldnames, dr.fieldnames))
  ```

- 其实官方库的内部也是类似的实现

  ```
  header = dict(zip(self.fieldnames, self.fieldnames))
  self.writerow(header)
  ```

调用举例：

文件： `crifanLib/demo/crifanCsvDemo.py`

```
OutputFilenameByHeaderAndList = "OutputDemoData_ByHeaderAnd
OutputFilenameByDictList = "OutputDemoData_ByDictList.csv"

OutputCsvHeader = ["单词", "重复频率", "来源列表"]

curFile = os.path.abspath(__file__)
curFolder = os.path.dirname(curFile)

DemoRowListList = [
  ["a", 0.5, ['NewConcept', 'FamilyAndFriends']],
  ["about", 0.75, ['NewConcept', 'YLE', 'EverybodyUp']],
  ["above", 0.5, ['NewConcept', 'YLE']],
  ["abroad", 0.25, ['NewConcept']]
]

DemoDictList = [
  {
    "单词": "a",
    "重复频率": 0.5,
    "来源列表": ['NewConcept', 'FamilyAndFriends'],
  },
  {
    "单词": "about",
    "重复频率": 0.75,
    "来源列表": ['NewConcept', 'YLE', 'EverybodyUp'],
  },
  {
    "单词": "above",
    "重复频率": 0.5,
    "来源列表": ['NewConcept', 'YLE'],
  },
  {
    "单词": "abroad",
    "重复频率": 0.25,
    "来源列表": ['NewConcept'],
  },
]

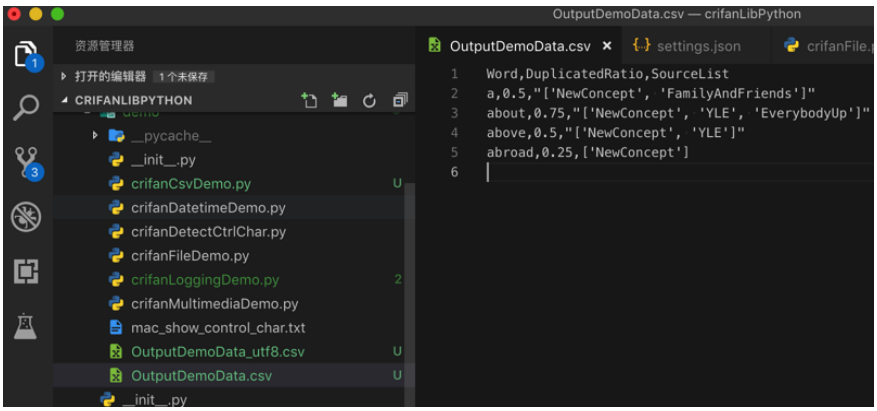
def demoCsvOutput():
  # Demo1: save by list of each row item list
  fullFilePathByHeaderAndList = os.path.join(curFolder, Out
  saveToCsvByHeaderAndList(OutputCsvHeader, DemoRowListList

  # Demo1: save by list of dict, not need assign header
  fullFilePathByDictList = os.path.join(curFolder, OutputF
  saveToCsvByDictList(DemoDictList, fullFilePathByDictList

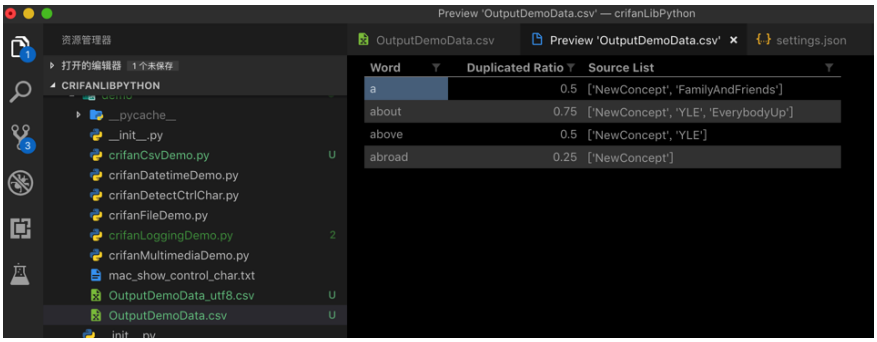
  """
  单词,重复频率,来源列表
  a,0.5,"['NewConcept', 'FamilyAndFriends']"
```

```
about,0.75,"['NewConcept', 'YLE', 'EverybodyUp']"
above,0.5,"['NewConcept', 'YLE']"
abroad,0.25,['NewConcept']
"""
```

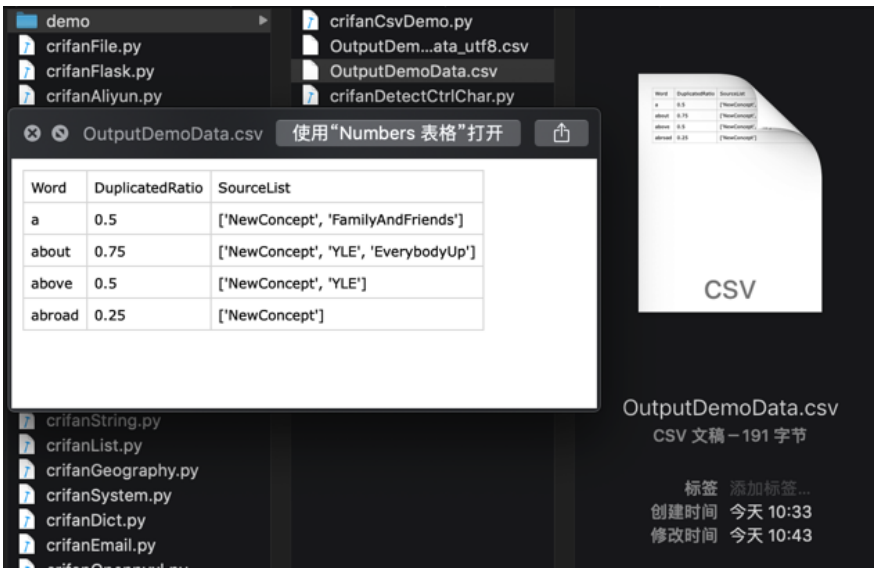
输出csv文件效果：



csv（作为表格文件的）预览效果：



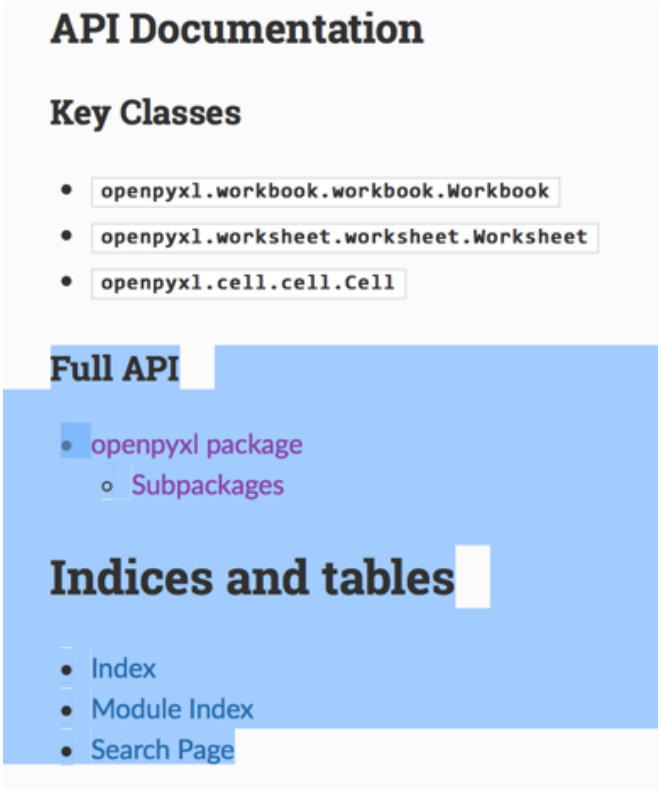
Mac中自带的预览的csv的效果：



处理excel

openpyxl

- `openpyxl`
  - 一句话描述：`read/write Excel 2010 xlsx/xlsm files`
- 文档
  - 首页
    - https://openpyxl.readthedocs.io/en/stable/
  - 入门
    - Manipulating a workbook in memory — openpyxl 2.5.3 documentation
      - http://openpyxl.readthedocs.io/en/stable/tutorial.html
  - API
    - 说明：开发期间如果遇到一些 `对象` ，需要查询其属性、函数等细节，可以去参考这些API文档

      -
    - API入口
      - openpyxl package — openpyxl 3.0.7 documentation
        - http://openpyxl.readthedocs.io/en/stable/api/openpyxl.html
    - 核心的类(API)
      - 最顶层的：`openpyxl.workbook.workbook.Workbook` = 对应excel文件

            - https://openpyxl.readthedocs.io/en/stable/api/openpyxl.workbook.workbook.html#openpyxl.workbook.workbook.Workbook
        - 第二层的：`openpyxl.worksheet.worksheet.Worksheet` = 对应excel内单个的tab页
            - https://openpyxl.readthedocs.io/en/stable/api/openpyxl.worksheet.worksheet.html#openpyxl.worksheet.worksheet.Worksheet
        - 最底层的：`openpyxl.cell.cell.Cell` = 对应 `单元格`
            - https://openpyxl.readthedocs.io/en/stable/api/openpyxl.cell.cell.html#openpyxl.cell.cell.Cell
    - 相关的：样式styles (`颜色` ， `对齐方式` 等等)
        - openpyxl.styles package — openpyxl 3.0.7 documentation
            - http://openpyxl.readthedocs.io/en/stable/api/openpyxl.styles.html

安装

```
pip3 install openpyxl
```

确认已安装成功

去python的shell中导入，没问题，且能输出版本信息，即说明安装没问题

```
➜  ~ python3
Python 3.6.1 (default, May  6 2017, 14:32:57)
[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.38)] on
Type "help", "copyright", "credits" or "license" for more
>>> import openpyxl
>>> openpyxl.__version__
'2.4.7'
>>>
```

附带：安装日志

```
➜  ~ pip3 install openpyxl
Collecting openpyxl
  Downloading openpyxl-2.4.7.tar.gz (156kB)
    100% |████████████████████████████████| 163kB 64kB/s
Collecting jdcal (from openpyxl)
  Downloading jdcal-1.3.tar.gz
Collecting et_xmlfile (from openpyxl)
  Downloading et_xmlfile-1.0.1.tar.gz
Building wheels for collected packages: openpyxl, jdcal, et
  Running setup.py bdist_wheel for openpyxl … done
  Stored in directory: /Users/crifan/Library/Caches/pip/whe
  Running setup.py bdist_wheel for jdcal … done
  Stored in directory: /Users/crifan/Library/Caches/pip/whe
  Running setup.py bdist_wheel for et-xmlfile … done
  Stored in directory: /Users/crifan/Library/Caches/pip/whe
Successfully built openpyxl jdcal et-xmlfile
Installing collected packages: jdcal, et-xmlfile, openpyxl
Successfully installed et-xmlfile-1.0.1 jdcal-1.3 openpyxl-
```

常用功能点

此处整理出 `openpyxl` 中常用的**功能点**。

基本操作

读取excel文件

```
from openpyxl import Workbook, load_workbook

wb = load_workbook(ExcelFullFilename) # work book
```

获取 `workbook` 的 `sheet`

- 获取sheet列表

  ```
  sheetNameList = wb.get_sheet_names()
  ```

- 获取某个sheet

  ```
  ws = wb["yourSheetName"] # work sheet
  ```

获取 `sheet` 中的 `cell`

```
curCell = ws.cell(row=rowIdx, column=columnIdx)
curCellValue = curCell.value
```

常见操作

对每个row循环处理

之前去循环rows是用

```
ws.iter_rows("A1:L2")
```

这种写法现在已废弃，建议换成：

```
ws["A1:L2"]
```

且看了官网:

Manipulating a workbook in memory — openpyxl 2.5.3 documentation

发现其实两种写法都可以:

- `range string` 的写法: `["XM:YN"]` ，其中 `X` ， `Y` 是字母， `M` ， `N` 是数字

```
cell_range = ws['A1':'C2']
```

- `ws` 调用 `iter_rows` 时，传递 `row` 和 `col` 的参数

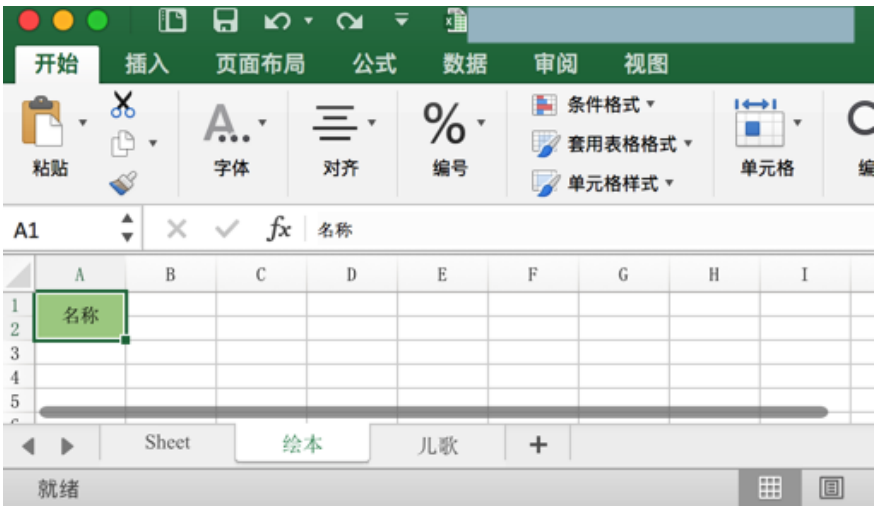
```
for row in ws.iter_rows(min_row=1, max_col=3, max_row
    for cell in row:
        print(cell)
```

设置背景色

用openpyxl设置**背景色**，用 `start_color` + `end_color` :

```
nameCell.fill = PatternFill(start_color="AACF91", end_colo
```

效果:

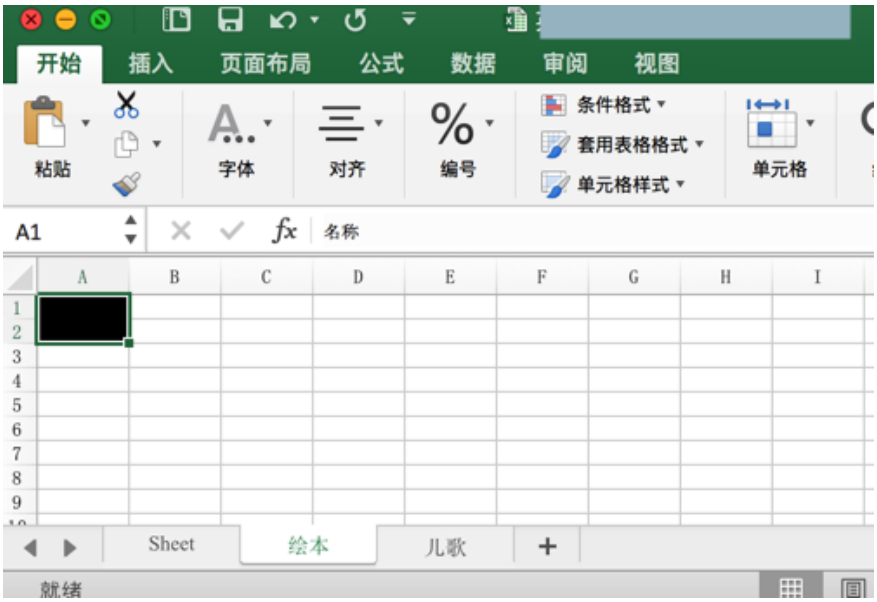


不要用PatternFill的bgColor

如果用（看起来以为可以工作的） `PatternFill` 的 `bgColor` :

```
```python
nameCell.fill = PatternFill(bgColor=Color("AACF91"), fill_
```
```

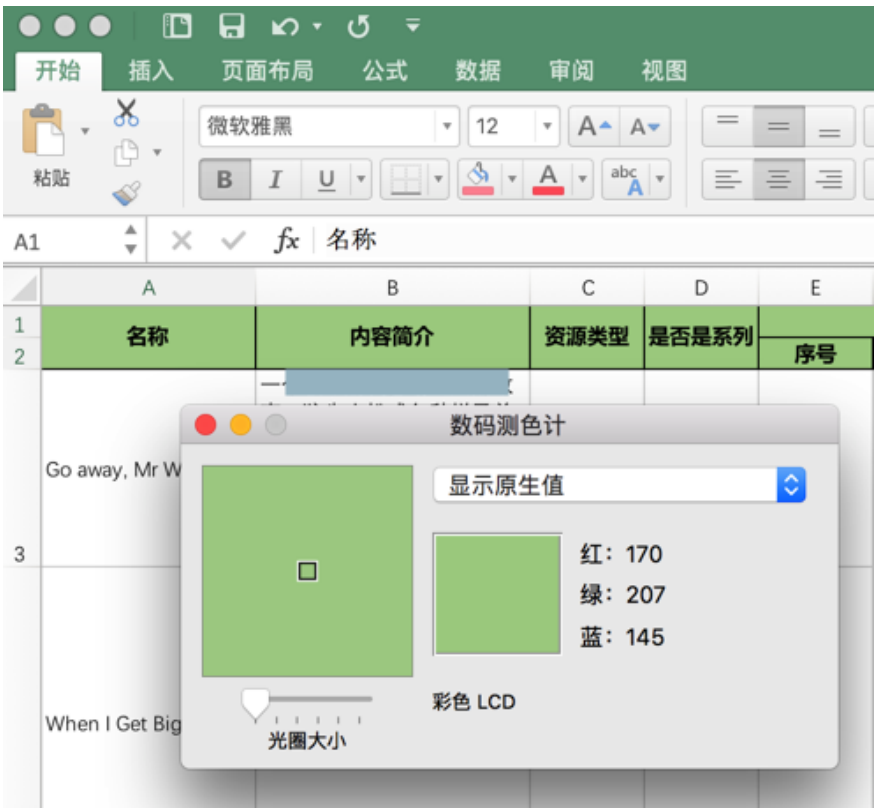
结果会导致黑色背景



如何查看颜色和RGB转换HEX值

对于上述颜色设置的值 `AACF91` 是如何得到的?

先去查看本身excel的背景色的RGB值:



此处背景色绿色的RGB值是: `170, 207, 145`

去把RGB转成HEX值:

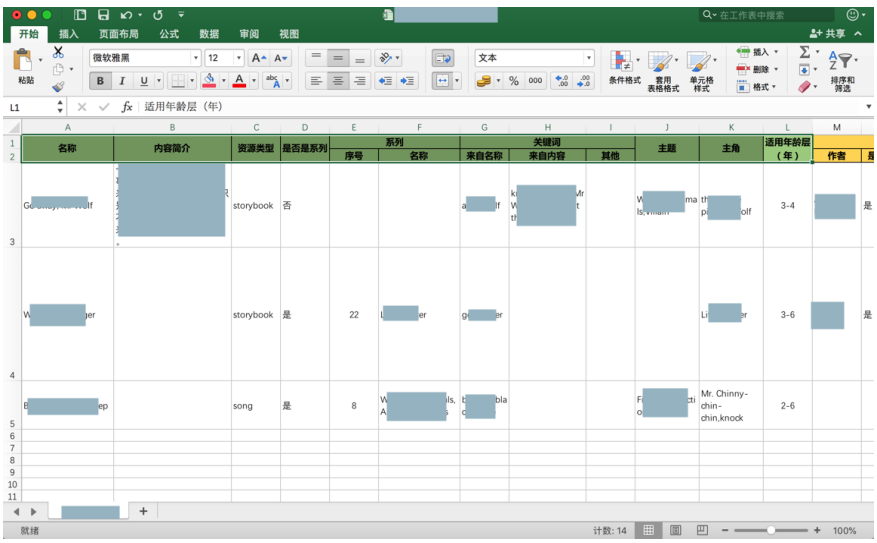
hex color

即可得到HEX值：

- RGB: `170, 207, 145` -> HEX: `#AACF91`

批量设置样式（背景色、对齐方式等）

背景：想要针对表头的所有的cell单元格（包括合并后的），批量去设置其样式



- 背景色：绿色
- 对齐方式
  - 左右：水平居中对齐

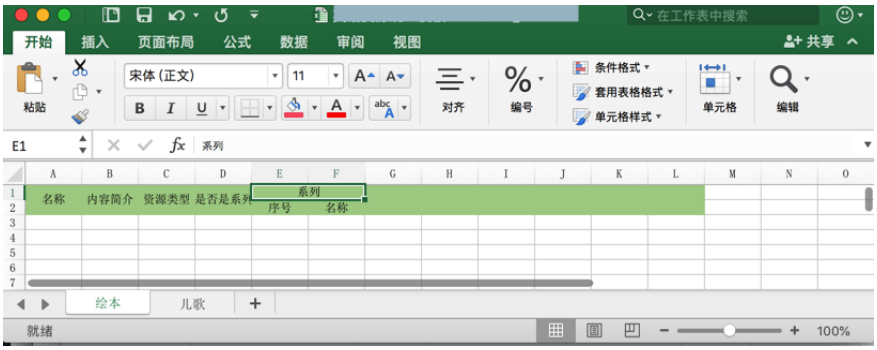
- 上下：垂直居中对齐

代码：

每个单元格循环去设置

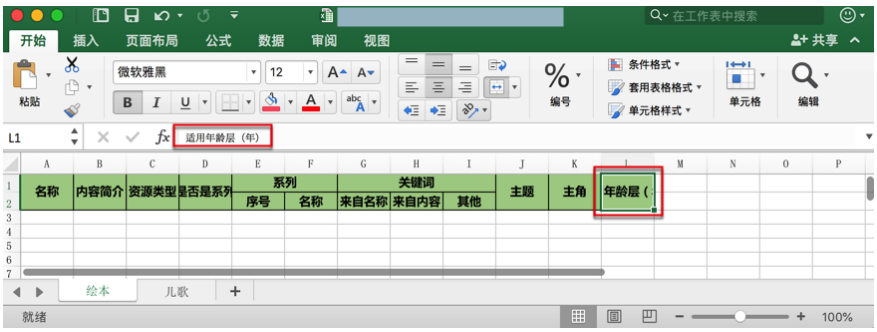
```
aligmentCenter = Alignment(horizontal='center', vertical='c
commonBackgroundColorHex = "AACF91"
commonFill = PatternFill(start_color=commonBackgroundColorH
for eachCommonRow in ws.iter_rows("A1:L2"):
    logging.info("eachCommonRow=%s", eachCommonRow)
    for eachCellInRow in eachCommonRow:
        logging.info("eachCellInRow=%s", eachCellInRow)
        eachCellInRow.alignment = aligmentCenter
        eachCellInRow.fill = commonFill
```

效果

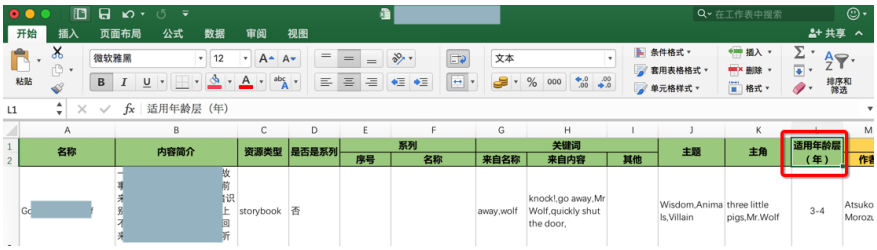


设置自适应列宽

背景：`openpyxl` 可以新建excel，但是表头内容显示不全：



希望表头的宽度，像另外一个手动创建并设置后的，去自动适应宽度：



解决办法：

由于 `openpyxl` 的 `sheet` 中的 `column_dimensions` 的 `auto_size` ， `bestFit` ， `collapsed` ，都无法启到此处想要的效果：自动设置列宽

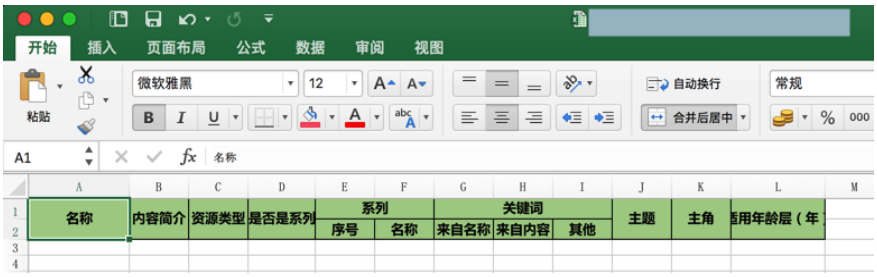
最后只能单独设置宽度：

```
nameCell = ws["A1"]
nameCell.value = "名称"
ws.merge_cells('A1:A2')
ws.column_dimensions["A"].width = 15

isSeriesCell = ws["D1"]
isSeriesCell.value = "是否是系列"
ws.merge_cells('D1:D2')
# ws.column_dimensions["D"].auto_size = True
# ws.column_dimensions["D"].bestFit = False
# ws.column_dimensions["D"].collapsed = True
ws.column_dimensions["D"].width = 10

fitAgeRangeCell = ws["L1"]
fitAgeRangeCell.value = "适用年龄层（年）"
ws.merge_cells('L1:L2')
# ws.column_dimensions["L"].auto_size = True
# ws.column_dimensions["L"].bestFit = True
# ws.column_dimensions["L"].collapsed = False
ws.column_dimensions["L"].width = 14
# ws["L3"].value = "很长的值很长的值很长的值测试列宽"
```

间接实现自动设置列宽的效果：



设置行高

背景：表头的部分内容由于行高太低而不显示了

解决办法：增加行高

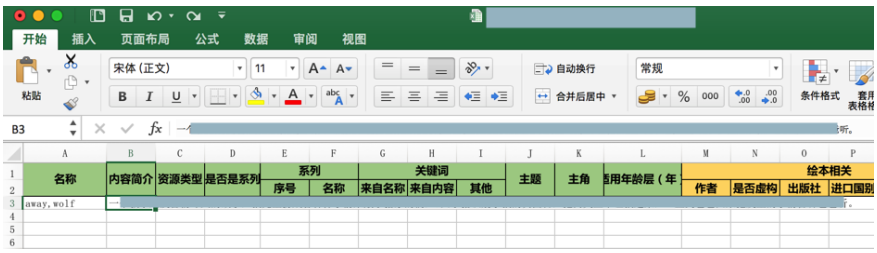
代码：

```
ws.row_dimensions[1].height = 30
ws.row_dimensions[2].height = 20
```

效果：增加了行高，内容完全显示

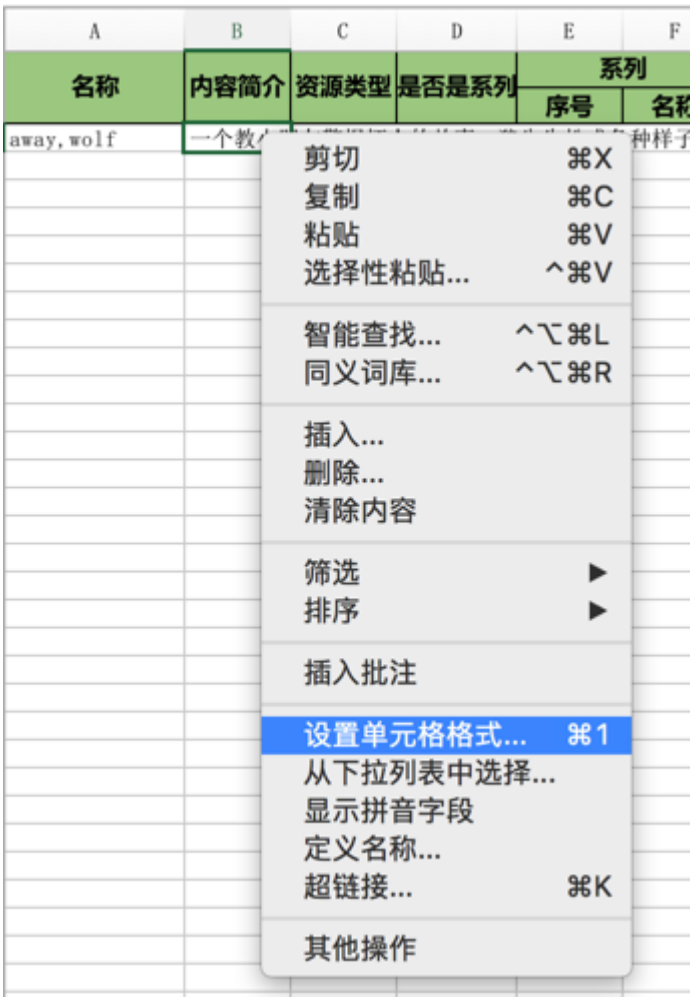
设置单元格自动换行

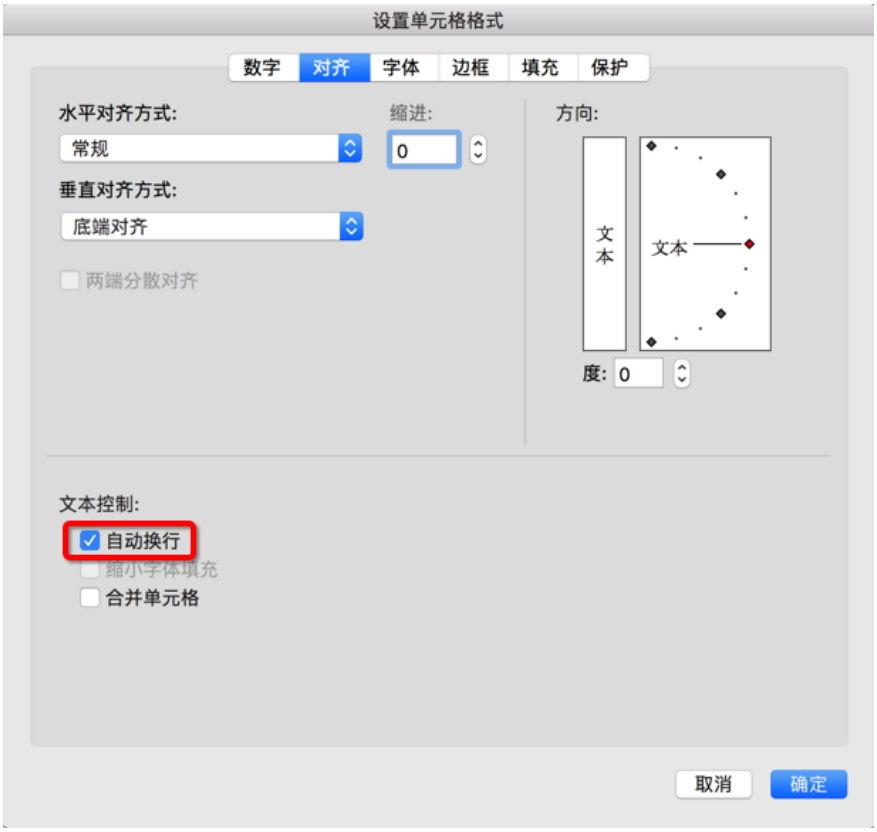
背景：之前已设置了 列表宽度（间接实现宽度自适应）+ 行高，导致现在显示效果很不好，有些内容没法完全显示



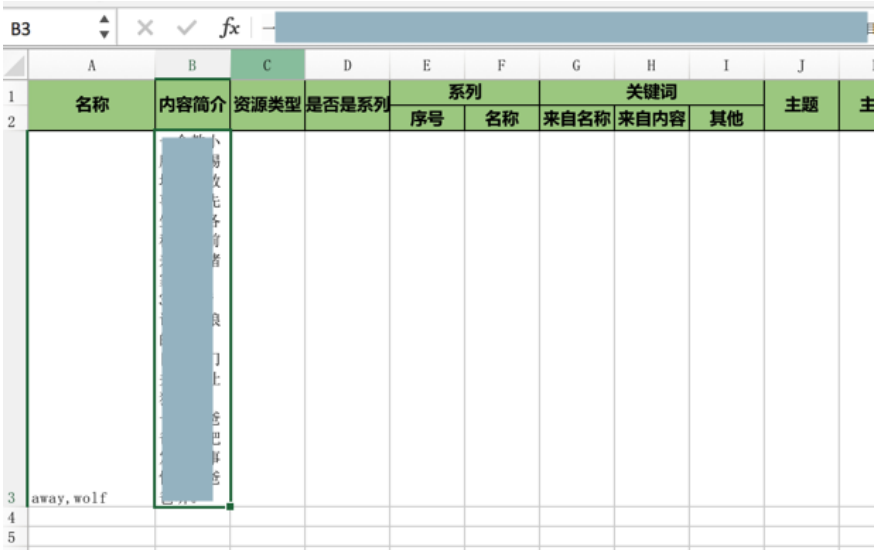
希望cell单元格的内容的自动换行。

希望效果：想要实现，类似于手动设置的：`设置单元格格式` -> `对齐` -> `文本控制` ->勾选：`自动对齐`


A
B
C
D
E
F
名称
内容简介
资源类型
是否是系列
系列
序号
名称
away, wolf
剪切 ⌘X
复制 ⌘C
粘贴 ⌘V
选择性粘贴... ^⌘V
智能查找... ^⌥⌘L
同义词库... ^⌥⌘R
插入...
删除...
清除内容
筛选
排序
插入批注
设置单元格格式... ⌘1
从下拉列表中选择...
显示拼音字段
定义名称...
超链接... ⌘K
其他操作


设置单元格格式
数字
对齐
字体
边框
填充
保护
水平对齐方式:
常规
缩进:
0
方向:
垂直对齐方式:
底端对齐
两端分散对齐
文本
文本
度: 0
文本控制:
自动换行
缩小字体填充
合并单元格
取消
确定

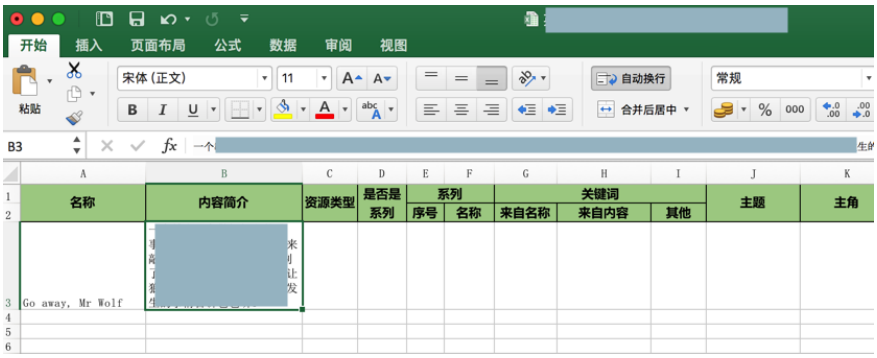
的效果：内容可以完全显示



代码：

```
# curCell.style.alignment.wrap_text = True #AttributeError:
curCell.alignment = Alignment(wrapText=True)
```

效果：很长的多行内容，可以根据宽度，自动换行，内容得以完全显示



封装成函数

目前已把 `openpyxl` 常用的功能，封装成**独立函数**了。

最新完整代码详见：

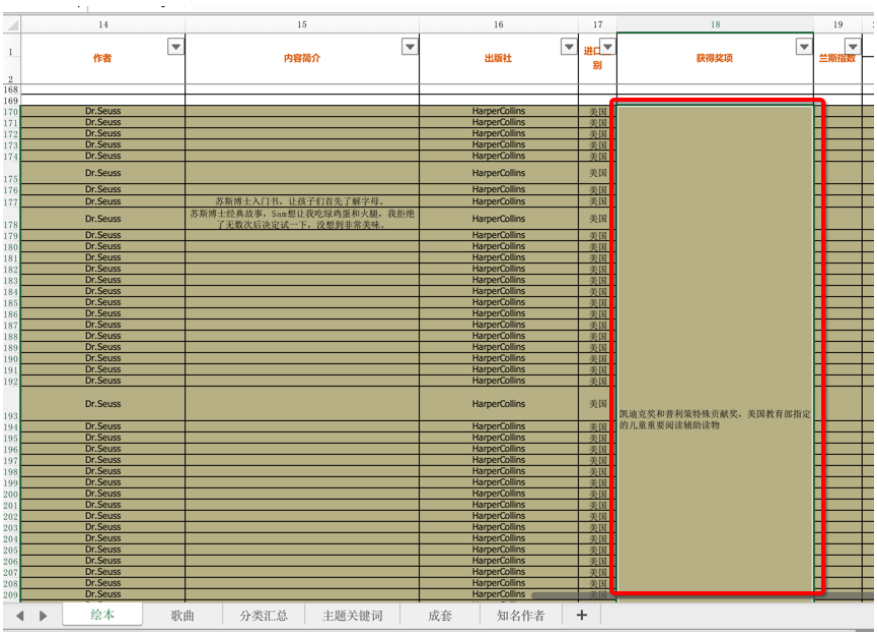
https://github.com/crifan/crifanLibPython/blob/master/python3/crifanLib/thirdParty/crifanOpenpyxl.py

下面贴出部分代码供参考：

获取合并后单元格的原始值：`isInCellRange` + `getCellRangeValue` + `getRealCellValue`

背景：

用代码自动去判断，对于合并后的单元格中的内容：



（之前调试时发现的）自动检测出来后，对于后续的，单元格值是空：

	14	15	16	17	18	19
1	作者	内容简介	出版社	进口别	获得奖项	兰斯指数
2						
168						
169						
170	Dr.Seuss		HarperCollins	美国		
171	Dr.Seuss		HarperCollins	美国		
172	Dr.Seuss		HarperCollins	美国		
173	Dr.Seuss		HarperCollins	美国		
174	Dr.Seuss		HarperCollins	美国		
175	Dr.Seuss		HarperCollins	美国		
176	Dr.Seuss		HarperCollins	美国		
177	Dr.Seuss	苏斯博士入门书，让孩子们首先了解字母。	HarperCollins	美国		
178	Dr.Seuss	苏斯博士经典故事，Sam想让我吃绿鸡蛋和火腿，我拒绝了无数次后决定试一下，没想到非常美味。	HarperCollins	美国		
179	Dr.Seuss		HarperCollins	美国		
180	Dr.Seuss		HarperCollins	美国		
181	Dr.Seuss		HarperCollins	美国		
182	Dr.Seuss		HarperCollins	美国		
183	Dr.Seuss		HarperCollins	美国		
184	Dr.Seuss		HarperCollins	美国		
185	Dr.Seuss		HarperCollins	美国		
186	Dr.Seuss		HarperCollins	美国		
187	Dr.Seuss		HarperCollins	美国		
188	Dr.Seuss		HarperCollins	美国		
189	Dr.Seuss		HarperCollins	美国		
190	Dr.Seuss		HarperCollins	美国		
191	Dr.Seuss		HarperCollins	美国		
192	Dr.Seuss		HarperCollins	美国		
193	Dr.Seuss		HarperCollins	美国	凯迪克奖和普利策特殊贡献奖，美国教育部指定的儿童重要阅读辅助读物	
194	Dr.Seuss		HarperCollins	美国		
195	Dr.Seuss		HarperCollins	美国		
196	Dr.Seuss		HarperCollins	美国		
197	Dr.Seuss		HarperCollins	美国		
198	Dr.Seuss		HarperCollins	美国		
199	Dr.Seuss		HarperCollins	美国		
200	Dr.Seuss		HarperCollins	美国		
201	Dr.Seuss		HarperCollins	美国		
202	Dr.Seuss		HarperCollins	美国		
203	Dr.Seuss		HarperCollins	美国		
204	Dr.Seuss		HarperCollins	美国		
205	Dr.Seuss		HarperCollins	美国		
206	Dr.Seuss		HarperCollins	美国		
207	Dr.Seuss		HarperCollins	美国		
208	Dr.Seuss		HarperCollins	美国		
209	Dr.Seuss		HarperCollins	美国		

绘本 | 歌曲 | 分类汇总 | 主题关键词 | 成套 | 知名作者 | +

但是属于同一系列的话：

R171C2 | Dr. Seuss

	1	2	3	4	5	6	7	8	9	10
1	绘本序列号	关键词						适合年龄段	Is Fiction or not	是否有续集
2		1绘本系列	2绘本名称	3绘本名称关键词	4主角	5主题	6内容关键词摘录			
168	5	Dr. Seuss The Big Orange Book of Be	Hooper Humperdink...?NOT HIM!							
169	6	Dr. Seuss The Big Orange Book of Be	In a people house							
170	1	Dr. Seuss	Fox in Socks	fox,socks	Fox	Animals,			Yes	否
171	2	Dr. Seuss	Hop on Pop	hop,pop					Yes	否
172	3	Dr. Seuss	I can read with my eYes shut!		The Cat in the hat				Yes	否
173	4	Dr. Seuss	Oh say can you say?						Yes	否
174	5	Dr. Seuss	Oh. The thinks you can think!						Yes	否
175	6	Dr. Seuss	One fish,two fish, red fish, blue fish		Fish and other animals				Yes	否
176	7	Dr. Seuss	There's a wocket in my pocket						Yes	否
177	8	Dr. Seuss	Dr.Suess's ABC	ABC,suess	Alphabet,	Alphabet,			Yes	否
178	9	Dr. Seuss	Green Eggs and Ham	green,eggs,ham	Sam and I	Kindhearted			Yes	否
179	10	Dr. Seuss	The cat in the hat	cat in the hat	The cat in the hat				Yes	否
180	11	Dr. Seuss	The cat in the hat comes back		The cat in the hat				Yes	否
181	12	Dr. Seuss	Mr. Brown can MooCan you?		Mr.Brown				Yes	否
182	13	Dr. Seuss	How the Grinch stole Christmas!		Grinch				Yes	否
183	14	Dr. Seuss	The Lorax		The Lorax				Yes	否
184	15	Dr. Seuss	The foot book						Yes	否
185	16	Dr. Seuss	Horton hears a who!		Horton				Yes	否
186	17	Dr. Seuss	The big brag		Bear and rabbit				Yes	否
187	18	Dr. Seuss	The Bippolo seed						Yes	否
188	19	Dr. Seuss	Gertrude McFuzz		Gertrude Mcfuzz				Yes	否
189	20	Dr. Seuss	The Glunk that got thunk		Glunk				Yes	否
190	21	Dr. Seuss	The great Henry Mcbride		Henry Mcbride				Yes	否
191	22	Dr. Seuss	Gustav,the Goldfish		Gustav				Yes	否
192	23	Dr. Seuss	King Looie Katz		King Looie Katz				Yes	否
193	24	Dr. Seuss	The Rabbit, the bear, and the Zinniga-zanniga		The Rabbit, the bear and the Zinniga-zanniga				Yes	否
194	25	Dr. Seuss	Steak for supper						Yes	否
195	26	Dr. Seuss	The strange shirt spot						Yes	否
196	27	Dr. Seuss	Tadd and Todd						Yes	否
197	28	Dr. Seuss	Too many Daves						Yes	否
198	29	Dr. Seuss	What was I scared of?						Yes	否
199	30	Dr. Seuss	The Zax						Yes	否
200	31	Dr. Seuss	And to think that I saw it on Mulberry street						Yes	否
201	32	Dr. Seuss	Bartholomew and the Oobleck						Yes	否
202	33	Dr. Seuss	The butter battle book						Yes	否
203	34	Dr. Seuss	Daisy-head Mayzie						Yes	否
204	35	Dr. Seuss	Did I ever tell you how lucky you are?						Yes	否
205	36	Dr. Seuss	Dr Suess's Sleep book						Yes	否
206	37	Dr. Seuss	The 500 hats of Bartholomew cubins						Yes	否
207	38	Dr. Seuss	Happy birthday to you!						Yes	否
208	39	Dr. Seuss	Horton hatches the egg						Yes	否
209	40	Dr. Seuss	Scrambles in Russian						Yes	否

绘本 | 歌曲 | 分类汇总 | 主题关键词 | 成套 | 知名作者 | +

那么后续单元格的值，在代码赋值时，也使用前面的值。

换句话说：

被合并后的单元的值，默认是空的

但是希望获取到被合并之前的值，即此处的合并后的左上角的第一个单元格的值

看看openpyxl是否能检测出来excel的单元格，是否是被合并的，且知道，具体合并了多少行（多少列）

这样才有利于后续单元格值为空时，使用前面的同系列的值

代码：

```
from openpyxl import Workbook, load_workbook

def isInCellRange(cellToCheck, cellRange):
    """
    to check a cell whether in a cell range
    :param cellToCheck:
    :param cellRange:
    :return:
        True : if cell in range
        False: if cell not in range
    """
    # logging.debug("cellToCheck=[%d:%d]", cellToCheck.row,
    # logging.debug("cellRange: row=[%d:%d] col=[%d:%d]",
    #               cellRange.min_row, cellRange.max_row, ce
    if (cellToCheck.row &gt;= cellRange.min_row) and \
        (cellToCheck.row &lt;= cellRange.max_row) and \
        (cellToCheck.col_idx &gt;= cellRange.min_col) and \
        (cellToCheck.col_idx &lt;= cellRange.max_col):
        logging.info("cell[%d:%d] with in cell range: row=
                     cellToCheck.row, cellToCheck.col_idx,
                     cellRange.min_row, cellRange.max_row,
        return True
    else:
        return False

def getCellRangeValue(cellRange):
    """
    get cell range value -&gt; the top left cell value
    :param cellRange:
    :return:
    """
    topLeftCell = ws.cell(row=cellRange.min_row, column=ce
    topLeftCellValue = topLeftCell.value
    return topLeftCellValue

def getRealCellValue(ws, curCell):
    """
    for openpyxl, to get real value from row and column
    expecially for merged cell, will get its (same) value

    :param row:
    :param column:
    :return:
    """

    realCellValue = curCell.value

    mergedCellsRangesList = ws.merged_cells.ranges
    # logging.info("mergedCellsRangesList=%s", mergedCellsR
```

```
    # Note:
    # to efficiency , we only check cell in range or not wh
    # for all merged cell value is None
    if not realCellValue:
        for eachCellRange in mergedCellsRangesList:
            if isInCellRange(curCell, eachCellRange):
                cellRangeValue = getCellRangeValue(eachCel
                realCellValue =  cellRangeValue
                break

    return realCellValue

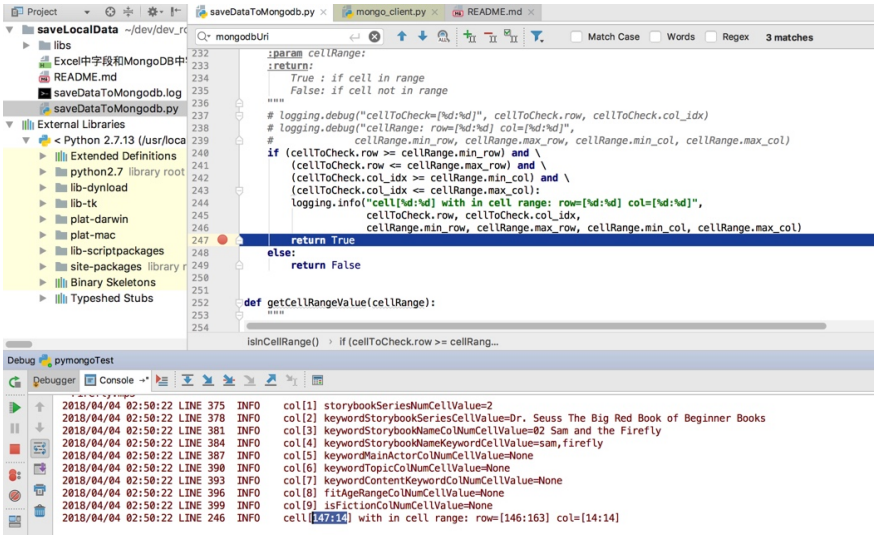
# parse excel file
wb = load_workbook(ExcelFullFilename)
logging.info("wb=%s", wb)
# sheetNameList = wb.get_sheet_names()
# logging.info("sheetNameList=%s", sheetNameList)
ws = wb[StorybookSheetTitle]
logging.info("ws=%s", ws)

# process each row in excel
# for curRowNum in range(realContentRowStartNum, ws.max_row
for curRowNum, eachRow in enumerate(ws.iter_rows(min_row=re
    curRowNum += realContentRowStartNum
    logging.info("-"*30 + " row[%d] " + "-"*30, curRowNum)

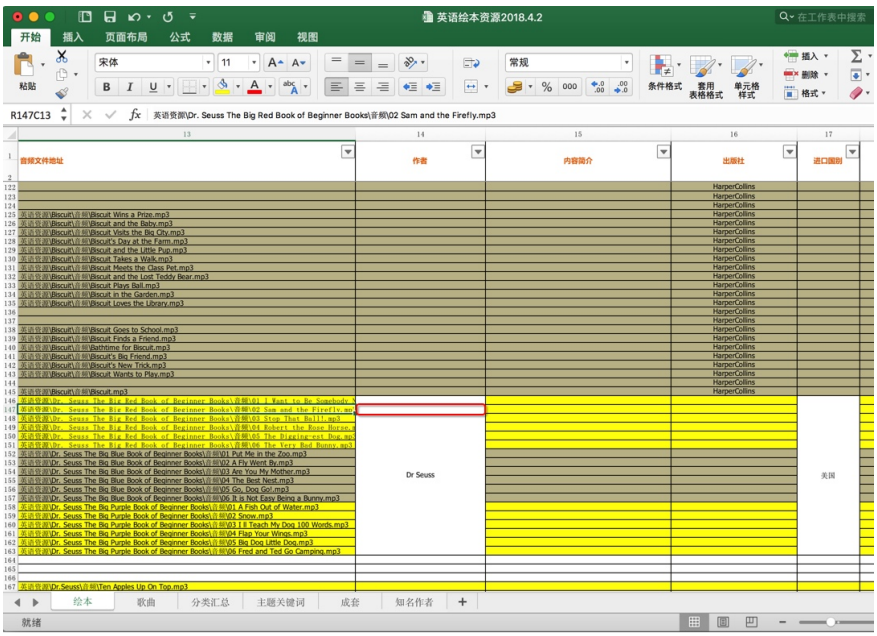
    ...
    # authorColNumCellValue = ws.cell(row=curRowNum, colum
    authorColNumCellValue = getRealCellValue(ws, eachRow[Au
    logging.info("col[%d] authorColNumCellValue=%s", Autho
    # contentAbstractColNumCellValue = ws.cell(row=curRowNu
    contentAbstractColNumCellValue = getRealCellValue(ws, e
    logging.info("col[%d] contentAbstractColNumCellValue=%s
    # publisherColNumCellValue = ws.cell(row=curRowNum, co
    publisherColNumCellValue = getRealCellValue(ws, eachRow
    logging.info("col[%d] publisherColNumCellValue=%s", Pub
    ...
```

可以检测到后续的，值是None的，处于被合并区域内的Cell单元格：

```
2018/04/04 02:50:22 LINE 246  INFO    cell[147:14] with in
```

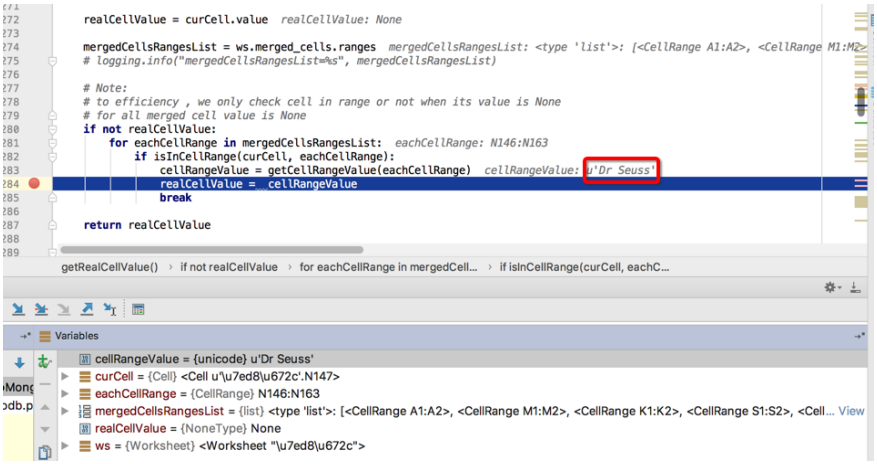


对应着147行，14列的：



然后就可以通过：

`getCellRangeValue` 去得到合并区域的值==合并区域的最左上角（top left）的那个单元格的值：

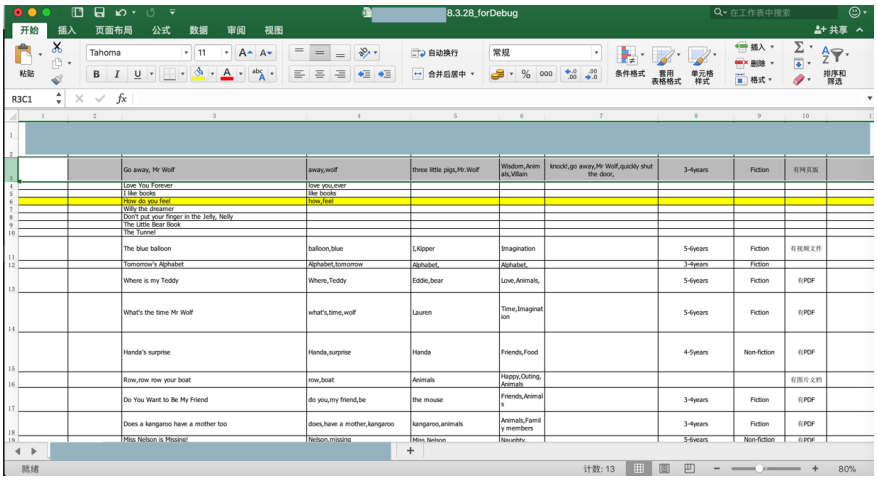


举例

此处列出一些用 `openpyxl` 操作 `excel` 的实际的例子，供参考。

用openpyxl读取excel文件

对于excel文件：



		Go away, Mr Wolf	away,wolf	three little pigs,Mr.Wolf	Wisdom,Animals,Villain	knock!,go away,Mr Wolf,quickly shut the door,	3-4years	Fiction	有网页版
		Love You Forever	love you,ever						
		I like books	like books						
		How do you feel	how,feel						
		Willy the dreamer							
		Don't put your finger in the Jelly, Nelly							
		The Little Bear Book							
		The Tunnel							
		The blue balloon	balloon,blue	I,Kipper	Imagination		5-6years	Fiction	有视频文件
		Tomorrow's Alphabet	Alphabet,tomorrow	Alphabet,	Alphabet,		3-4years	Fiction	
		Where is my Teddy	Where,Teddy	Eddie,bear	Love,Animals,		5-6years	Fiction	有PDF
		What's the time Mr Wolf	what's,time,wolf	Lauren	Time,Imagination		5-6years	Fiction	有PDF
		Handa's surprise	Handa,surprise	Handa	Friends,Food		4-5years	Non-fiction	有PDF
		Row,row row your boat	row,boat	Animals	Happy,Outing,Animals				有图片文档
		Do You Want to Be My Friend	do you,my friend,be	the mouse	Friends,Animals		3-4years	Fiction	有PDF
		Does a kangaroo have a mother too	does,have a mother,kangaroo	kangaroo,animals	Animals,Family members		3-4years	Fiction	有PDF
		Miss Nelson is Missing!	Nelson,missing	Miss Nelson	Naughty		5-6years	Non-fiction	有PDF

代码：

```
from openpyxl import Workbook, load_workbook

StorybookSheetTitle = u"绘本"
EnglishStorybookRootPath = "/Users/crifan/dev/dev_root/xxx"
ExcelFilename = "xxx资源2018.3.28_forDebug.xlsx"
ExcelFullFilename = os.path.join(EnglishStorybookRootPath,
AudioFilePathPrefix = EnglishStorybookRootPath

# parse excel file
wb = load_workbook(ExcelFullFilename) # work book
logging.info("wb=%s", wb)
# sheetNameList = wb.get_sheet_names()
# logging.info("sheetNameList=%s", sheetNameList)
ws = wb[StorybookSheetTitle] # work sheet
logging.info("ws=%s", ws)

# process each row in excel
for curRowNum in range(realContentRowStartNum, ws.max_row -
    logging.info("-"*30 + " row[%d] " + "-"*30, curRowNum)

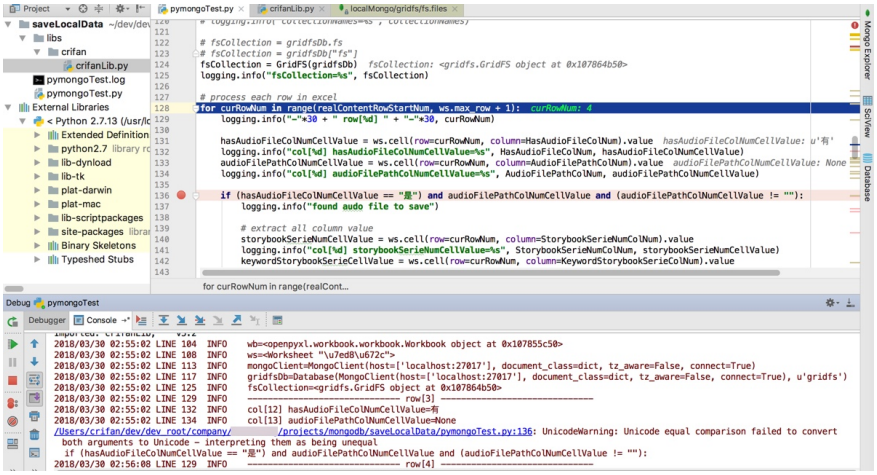
    hasAudioFileColNumCellValue = ws.cell(row=curRowNum, co
    logging.info("col[%d] hasAudioFileColNumCellValue=%s",
    audioFilePathColNumCellValue = ws.cell(row=curRowNum, c
    logging.info("col[%d] audioFilePathColNumCellValue=%s",
```

输出：

```
2018/03/30 02:55:02 LINE 104  INFO    wb=<openpyxl.workbook
2018/03/30 02:55:02 LINE 108  INFO    ws=<Worksheet "\u7ed8
2018/03/30 02:55:02 LINE 113  INFO    mongoClient=MongoClie
2018/03/30 02:55:02 LINE 117  INFO    gridfsDb=Database(Mon
2018/03/30 02:55:02 LINE 125  INFO    fsCollection=<gridfs.

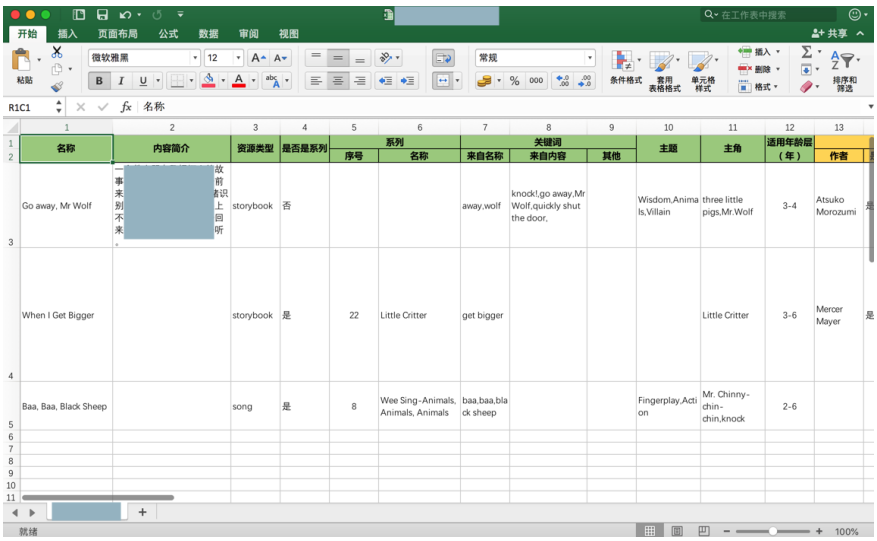
2018/03/30 02:55:02 LINE 129  INFO    ---------- row[3] ---
2018/03/30 02:55:02 LINE 132  INFO    col[12] hasAudioFileC
2018/03/30 02:55:02 LINE 134  INFO    col[13] audioFilePath
```

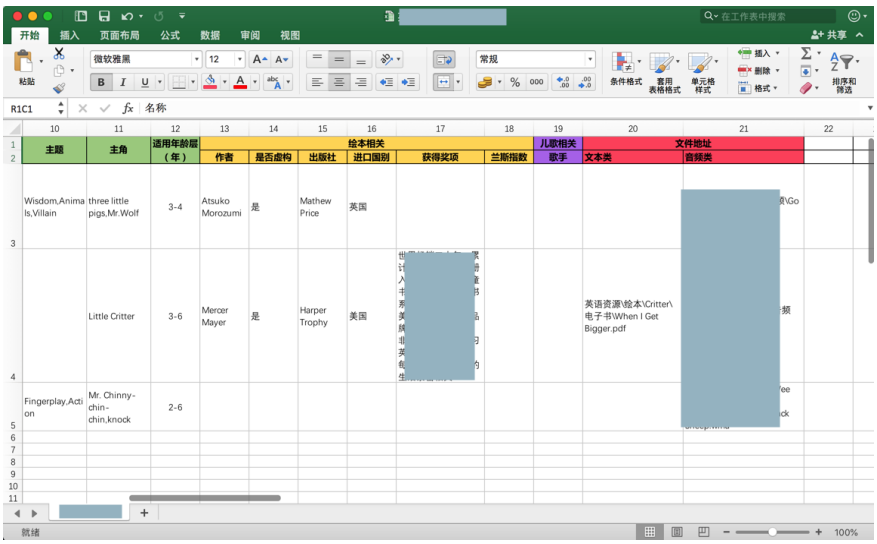
（用PyCharm）调试输出的效果：



新建excel文件并保存数据和设置单元格样式

背景：希望新建这种效果的excel

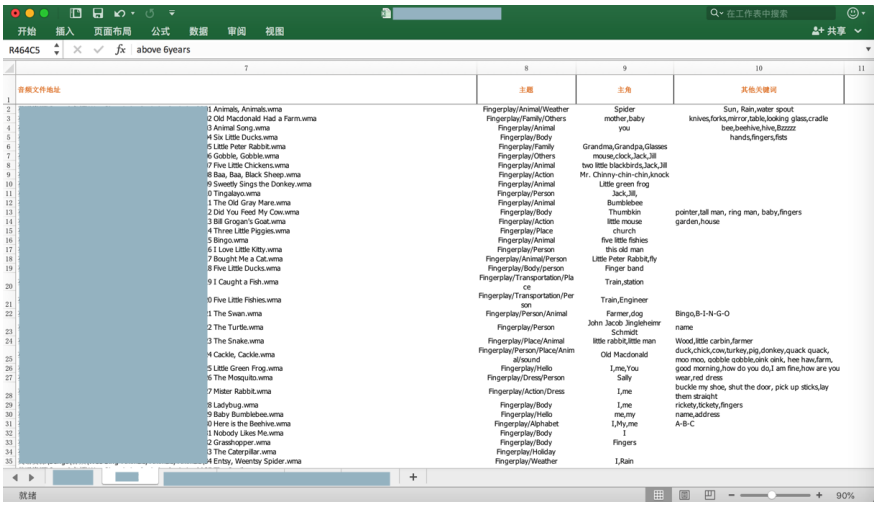




并且创建对应的第一行和第二行的表头，且部分单元格是合并的

然后再一点点把之前已有的excel中的内容：

						6内容关键词摘录	适合年龄段
		Color zoo	color zoo				
		We're going on a bear hunt	bear hunt				
		Is your mama a llama	mama,llama				2-6years
		The carrot seed					3-6years
		Mouse paint	mouse,paint	mouse	Color,Animals		4-5years
1	Madeline	Madeline	Madeline	Madeline	Family members,Sick		4-5years
2	Madeline	Madeline in London	Madeline, London	Madeline			
3	Madeline	Madeline and the Gypsies	Madeline,Gypsies	Madeline			
4	Madeline	Madeline and the Bad Hat	Madeline,bad hat	Madeline	Naughty,Friends,Animals		5-6years
5	Madeline	Madeline Paper dolls	Madeline,paper dolls	Madeline			
6	Madeline	Madeline's Rescue	Madeline,rescue	Madeline			3-8years
7	Madeline	Madeline's Christmas	Madeline,Christmas	Madeline			3-8years
		Chicka Chicka Boom Boom	chicka,boom	Alphabet,	Alphabet,		5-6years
		If you take a mouse to school	mouse,school,take	a mouse	School,Animals		5-6years
		Knock! Knock! Open the Door					3-6years
		A Good Day					2-6years
		The Napping House	napping house				3-6years
		Maisy Goes Camping					3-6years
		Busy zoo					3-6years
		Little Blue And Little Yellow	little blue,little yellow				
		An extraordinary egg	egg				
		Swimmy	swimmy				



音频文件地址	主题	主角	其他关键词
1 Animals, Animals.wma	Fingerplay/Animal/Weather	Spider	Sun, Rain,water spout
2 Old Macdonald Had a Farm.wma	Fingerplay/Family/Others	mother,baby	knives,forks,mirror,table,looking glass,cradle
3 Animal Song.wma	Fingerplay/Animal	you	bee,beehive,hive,Bzzzzz
4 Six Little Ducks.wma	Fingerplay/Body		hands,fingers,fists
5 Little Peter Rabbit.wma	Fingerplay/Family	Grandma,Grandpa,Glasses	
6 Gobble, Gobble.wma	Fingerplay/Others	mouse,clock,Jack,Jill	
7 Five Little Chickens.wma	Fingerplay/Animal	two little blackbirds,Jack,Jill	
8 Baa, Baa, Black Sheep.wma	Fingerplay/Action	Mr. Chinny-chin-chin,knock	
9 Sweetly Sings the Donkey.wma	Fingerplay/Animal	Little green frog	
0 Tingalayo.wma	Fingerplay/Person	Jack,Jill,	
1 The Old Gray Mare.wma	Fingerplay/Animal	Bumblebee	
2 Did You Feed My Cow.wma	Fingerplay/Body	Thumbkin	pointer,tall man, ring man, baby,fingers
3 Bill Grogan's Goat.wma	Fingerplay/Action	little mouse	garden,house
4 Three Little Piggies.wma	Fingerplay/Place	church	
5 Bingo.wma	Fingerplay/Animal	five little fishies	
6 I Love Little Kitty.wma	Fingerplay/Person	this old man	
7 Bought Me a Cat.wma	Fingerplay/Animal/Person	Little Peter Rabbit,fly	
8 Five Little Ducks.wma	Fingerplay/Body/person	Finger band	
9 I Caught a Fish.wma	Fingerplay/Transportation/Place	Train,station	
0 Five Little Fishies.wma	Fingerplay/Transportation/Person	Train,Engineer	
1 The Swan.wma	Fingerplay/Person/Animal	Farmer,dog	Bingo,B-I-N-G-O
2 The Turtle.wma	Fingerplay/Person	John Jacob Jingleheimr Schmidt	name
3 The Snake.wma	Fingerplay/Place/Animal	little rabbit,little man	Wood,little carbin,farmer
4 Cackle, Cackle.wma	Fingerplay/Person/Place/Animal/sound	Old Macdonald	duck,chick,cow,turkey,pig,donkey,quack quack, moo moo, gobble gobble,oink oink, hee haw,farm,
5 Little Green Frog.wma	Fingerplay/Hello	I,me,You	good morning,how do you do,I am fine,how are you
6 The Mosquito.wma	Fingerplay/Dress/Person	Sally	wear,red dress
7 Mister Rabbit.wma	Fingerplay/Action/Dress	I,me	buckle my shoe, shut the door, pick up sticks,lay them straight
8 Ladybug.wma	Fingerplay/Body	I,me	rickety,tickety,fingers
9 Baby Bumblebee.wma	Fingerplay/Hello	me,my	name,address
0 Here is the Beehive.wma	Fingerplay/Alphabet	I,My,me	A-B-C
1 Nobody Likes Me.wma	Fingerplay/Body	I	
2 Grasshopper.wma	Fingerplay/Body	Fingers	
3 The Caterpillar.wma	Fingerplay/Holiday		
4 Entsy, Weentsy Spider.wma	Fingerplay/Weather	I,Rain	

保存进来。

代码：

```
from openpyxl import Workbook, load_workbook
from openpyxl.styles import Alignment
from openpyxl.styles import PatternFill, colors, Color
from openpyxl.styles import NamedStyle, Font, Border, Side

# create new output unified format excel file
wbOut = Workbook()
logging.info("wbOut=%s", wbOut)
# wsStorybook = wbOut.create_sheet(StorybookSheetTitle)
wsStorybook = wbOut.active
wsStorybook.title = StorybookSheetTitle
wsSong = wbOut.create_sheet(SongSheetTitle)
logging.info("wsStorybook=%s, wsSong=%s", wsStorybook, wsS

# common style
alignmentCenter = Alignment(horizontal='center', vertical=
fontYahei12 = Font(name=u'微软雅黑', bold=True, size=12)
thinSide = Side(style='thin', color=colors.BLACK)
allBlackThinBorder = Border(left=thinSide, top=thinSide, r

# several type of headers style
NamedStyleCommonHeader      = "CommonHeader"
NamedStyleStorybookHeader   = "StorybookHeader"
NamedStyleSongHeader        = "SongHeader"
NamedStyleFileHeader        = "FileHeader"

commonBkgColorHex = "AACF91"
commonFill = PatternFill(start_color=commonBkgColorHex, end

...

commonStyleHeader = NamedStyle(name=NamedStyleCommonHeader
commonStyleHeader.font = fontYahei12
commonStyleHeader.border = allBlackThinBorder
commonStyleHeader.alignment = alignmentCenter
commonStyleHeader.fill = commonFill
wbOut.add_named_style(commonStyleHeader)

...

# init headers
def initOutputExcelHeaders(ws):
    logging.info("ws=%s", ws)

    nameCell = ws["A1"]
    nameCell.value = "名称"
    ws.merge_cells('A1:A2')

    contentAbstractCell = ws["B1"]
    contentAbstractCell.value = "内容简介"
```

```
    ws.merge_cells('B1:B2')

    resTypeCell = ws["C1"]
    resTypeCell.value = "资源类型"
    ws.merge_cells('C1:C2')

    isSeriesCell = ws["D1"]
    isSeriesCell.value = "是否是系列"
    ws.merge_cells('D1:D2')

    seriesCell = ws["E1"]
    seriesCell.value = "系列"
    ws.merge_cells('E1:F1')

    seriesNumCell = ws["E2"]
    seriesNumCell.value = "序号"

    seriesNameCell = ws["F2"]
    seriesNameCell.value = "名称"

    for eachCommonRow in ws.iter_rows("A1:L2"):
        logging.info("eachCommonRow=%s", eachCommonRow)
        for eachCellInRow in eachCommonRow:
            logging.info("eachCellInRow=%s", eachCellInRow
            eachCellInRow.style = NamedStyleCommonHeader

initOutputExcelHeaders(wsStorybook)
initOutputExcelHeaders(wsSong)
wbOut.save(OutputUnifiedFormatExcelFilename)
```

效果:

确保指定的字体系统中是存在的

设置的 `Font` 确保是系统中存在的字体才可以

此处用的是 `Python2` ，系统中当前的字体名是 `微软雅黑`

对应设置字体为雅黑的代码是

```
fontYahei12 = Font(name=u'微软雅黑', bold=True, size=12)
```

其中重点是： `font` 的 `name` 是 `unicode` 才可以

估计内容原因是：如果name不是unicode，则name找不到，所以会报错，会导致excel打不开

此处 `微软雅黑` 字体的效果是：

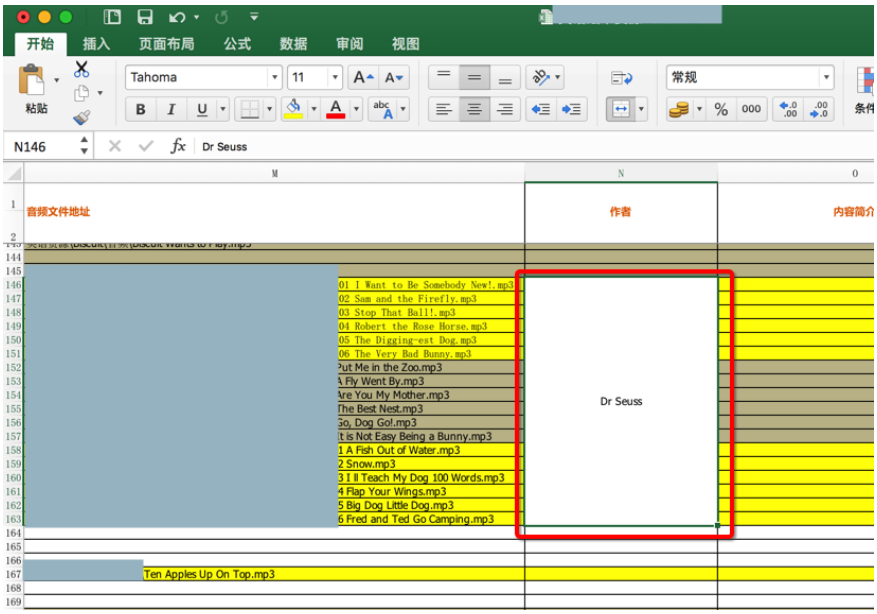
常见问题

警告：发现xlsx中的部分内容有问题。是否让我们尽量尝试恢复？如果信任此工作簿的源，请单击"是"

背景：

用openpyxl去创建新excel文件，并从某原有excel中拷贝数据到新excel中

其中包括保留原先的合并后的单元格，比如：



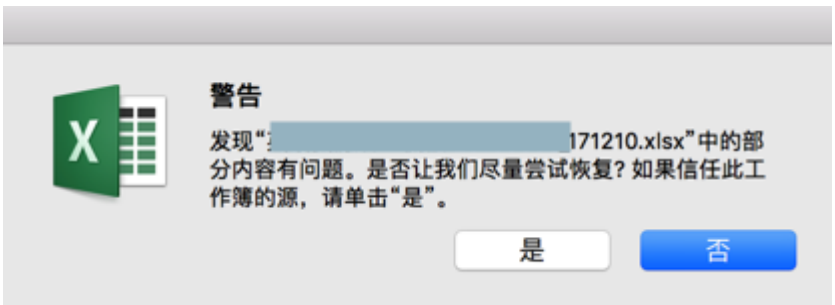
相关代码是：

```
def keepPrevMergedCells(wsIn, wsInCurCellRange, wsInCurRowN
    """
        keep original excel's merged cells
    """
    logging.info("keepPrevMergedCells: wsIn=%s, wsInCurCel
                 wsIn, wsInCurCellRange, wsInCurRowNum, ws(

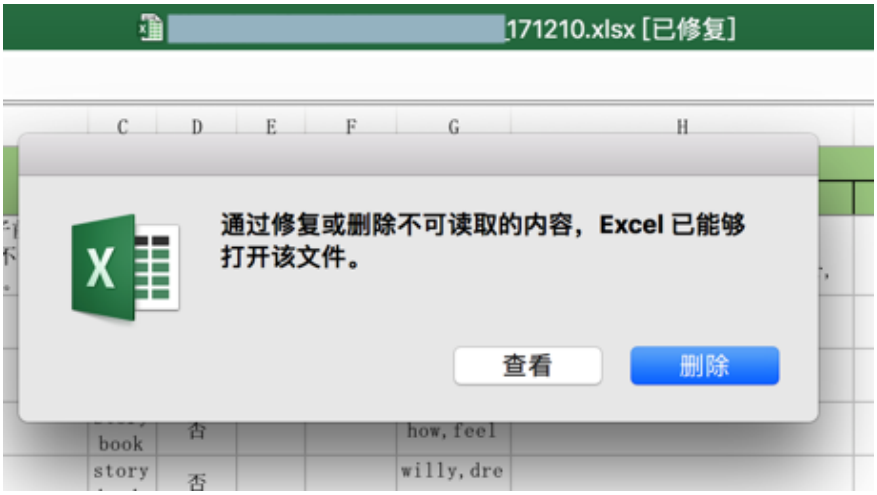
    cellRangeValue = crifanLib.getCellRangeValue(wsIn, wsIn
    rangeMinRow = wsInCurCellRange.min_row
    rangeMinCol = wsInCurCellRange.min_col
    rangeMaxRow = wsInCurCellRange.max_row
    rangeMaxCol = wsInCurCellRange.max_col
    logging.info("curCellRange: min_row=%s, min_col=%s, max
                 rangeMinRow, rangeMinCol, rangeMaxRow, ran
    rowNumDiff = wsInCurRowNum - rangeMinRow
    logging.info("rowNumDiff=%s", rowNumDiff)
    curStartCol = wsOutColNum
    curEndCol = wsOutColNum + (rangeMaxCol - rangeMinCol)
    logging.info("curStartCol=%s, curEndCol=%s", curStartCo
    if rowNumDiff > 1:
        wsOut.unmerge_cells(start_row=rangeMinRow,
                            start_column=curStartC
                            end_row=wsInCurRowNum
                            end_column=curEndCol)
    wsOut.merge_cells(start_row=rangeMinRow,
                      start_column=curStartCol,
                      end_row=wsInCurRowNum,
                      end_column=curEndCol)
```

打开保存后的excel文件后报错：

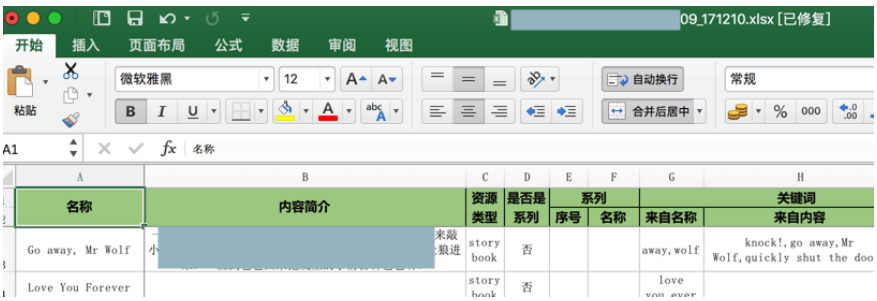
> 发现 xlsx 中的部分内容有问题。是否让我们尽量尝试恢复？如果信任此工作簿的源，请单击“是”。



点击 `是` ，然后再点击 `删除` ：



是可以打开的，且貌似数据也是对的：

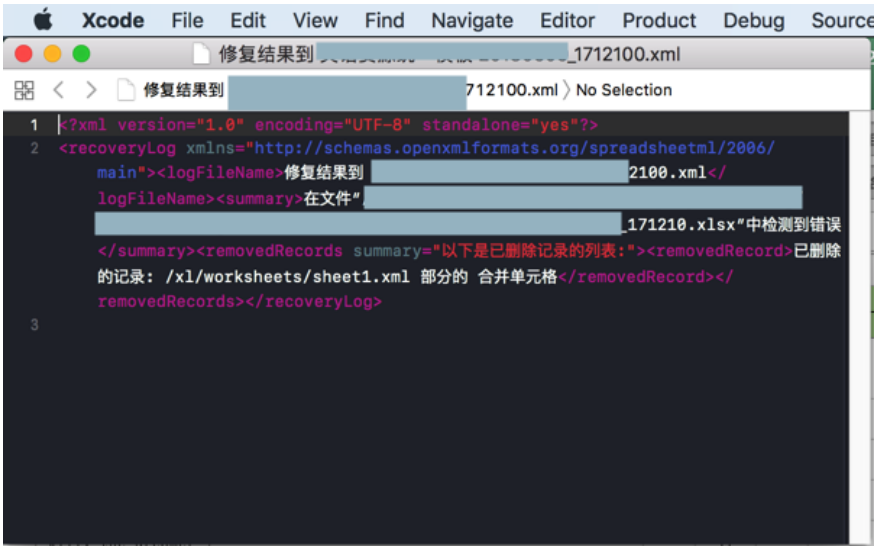


调试过程：

点击 `查看` ，是打开了 `xml` 文件：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<recoveryLog xmlns="http://schemas.openxmlformats.org/sprea
```



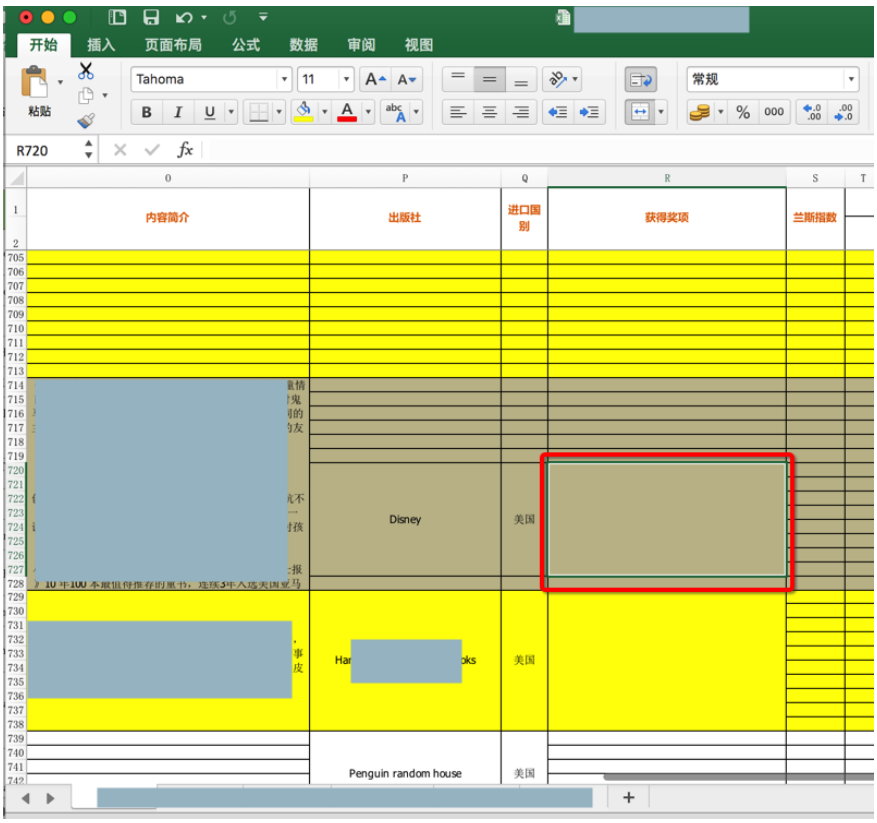
最终调试到的现象是：

- 720行没问题
- 但是750行出问题

然后去原先文件看看是否有什么异常的情况：

最后找到了是原先的excel中的 `702 – 727` 行 `R` 列中：

是合并的单元格，但是内容却是**空=** `None`



根本原因：代码生成的excel中，某些单元格的值是无效的 `None` ，无法被Excel软件所识别

解决办法

- （不太推荐）办法1：忽略之
  - 根据提示，正常打开excel，点击删除，会删除对应无效的值
    - 后续即可正常使用excel文件
- （推荐）方法2：代码生成时，不要设置（Excel软件不支持的） `None` 等值
  - 即可彻底规避掉，不会产生此问题
  - 修复后的相关代码是

```
def keepPrevMergedCells(wsIn, wsInCurCellRange, wsI
    """
        keep original excel's merged cells
        if merged cell range itself value is None,
        otherwise will cause generated excel file c
    """
    logging.info("keepPrevMergedCells: wsIn=%s, wsI
                 wsIn, wsInCurCellRange, wsInCurRowN

    cellRangeValue = crifanLib.getCellRangeValue(ws
    if not cellRangeValue:
        logging.warning("Not keep previous None or
        return
    # then wsOut.unmerge_cells and wsOut.merge_cell
```

xlrd和xlwt

下载和安装

xlrd

从

http://pypi.python.org/pypi/xlrd

下载，最新版本的

xlrd-0.8.0.tar.gz

得到 `130KB` 的 `xlrd-0.8.0.tar.gz`

解压后，打开cmd切换到对应目录，去执行

```
setup.py install
```

xlwt

从主页：

http://pypi.python.org/pypi/xlwt

下载 `xlwt-0.7.4.tar.gz` 后，解压

切换到对应目录后，去安装：

```
setup.py install
```

xlutils

从主页：

http://pypi.python.org/pypi/xlutils

下载

xlutils-1.5.2.tar.gz

得到47KB的 `xlutils-1.5.2.tar.gz`

解压后，切换到对应目录后，去安装：

```
setup.py install
```

基本操作

概述：

EXCEL自动化常用库速查表

《用 PYTHON 自动办公，做职场高手》课程配套资料

扫码查看课程

xlrd: 用来读Excel的库

可以用来读取 Excel 的第三方库很多，然而 xlrd 是其中更方便的那个。简单易用，功能全面，对于.xls 和 .xlsx 格式都可以很好的支持。

import xlrd	导入 xlrd 模块
xlsx = xlrd.open_workbook('文件地址', formatting_info=True/False)	打开 Excel 工作簿，是否带格式
sheet = xlsx.sheet_by_index(索引号)	打开 Excel 工作表（方法1）
sheet = xlsx.sheet_by_name("工作表名")	打开 Excel 工作表（方法2）
sheet.cell_value(行, 列)	获取单元格的值（方法1）
sheet.cell(行, 列).value	获取单元格的值（方法2）
sheet.row(行)[列].value	获取单元格的值（方法3）
xlsx.sheet_names()	获取所有工作表的名字
xlsx.nsheets	获取工作表数量
sheet.nrows	获取所有行数
sheet.ncols	获取所有列数

xlwt：用来写 Excel 的库

看起来像是 xlrd 库的孪生兄弟，实际上确实是同一个作者开发的作品。同样的容易上手，兼容.xls 和 .xlsx 格式。

import xlwt	导入 xlwt 模块
workbook = xlwt.Workbook()	新建工作簿
sheet = workbook.add_sheet('表名')	新建工作表
sheet.write(行, 列, '内容')	向某行某列写入内容
workbook.save('文件地址')	保存工作簿
style = xlwt.XFStyle()	新建样式
font = xlwt.Font()	新建字体样式
font.name = '字体名'	设置字体名
font.bold = True/False	设置是否加粗
font.height = 字号	设置字号
style.font = font	将字体样式添加进样式中
borders = xlwt.Borders()	新建边框样式
borders.top = xlwt.Borders.边框样式	设置上边框样式
borders.bottom = xlwt.Borders.边框样式	设置下边框样式
borders.left = xlwt.Borders.边框样式	设置左边框样式
borders.right = xlwt.Borders.边框样式	设置右边框样式
style.borders = borders	将边框样式添加进样式中
alignment = xlwt.Alignment()	新建对齐样式
alignment.horz = xlwt.Alignment.水平对齐样式	设置水平对齐样式
alignment.vert = xlwt.Alignment.垂直对齐样式	设置垂直对齐样式
style.alignment = alignment	将对齐样式添加进样式中
sheet.write(行, 列, 内容, style)	以特定格式写入内容

xlutils：用来复制Excel的库

xlutils 借鉴了 xlrd 和 xlwt, 然而并没有它们设计的好用，最可取的地方就是复制功能做的不错，我们在课程中也有用到。

from xlutils.copy import copy	导入模块的复制功能
new_excel = xlutils.copy(tem_excel)	复制已打开工作簿

新建excel文件

代码

```
import xlwt
from datetime import datetime

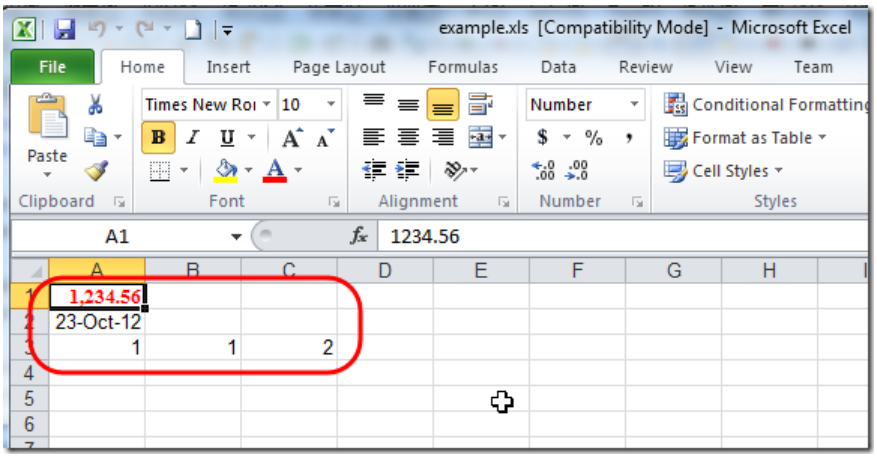
style0 = xlwt.easyxf('font: name Times New Roman, color-inc
style1 = xlwt.easyxf(num_format_str='D-MMM-YY')

wb = xlwt.Workbook()
ws = wb.add_sheet('A Test Sheet')

ws.write(0, 0, 1234.56, style0)
ws.write(1, 0, datetime.now(), style1)
ws.write(2, 0, 1)
ws.write(2, 1, 1)
ws.write(2, 2, xlwt.Formula("A3+B3"))

wb.save('example.xls')
```

效果



追加写入新数据

背景：想要添加写入数据到已经存在的 `Excel` 的 `xls` 文件，即打开excel文件，写入新数据

解决办法

想要往已经存在的xls文件中，写入新的行，新的数据，对应的逻辑为：

1. 用 `xlrd.open_workbook` 打开已有的xsl文件
    - 注意：添加参数 `formatting_info=True` ，得以保存之前数据的格式
2. 然后用 `from xlutils.copy import copy` 拷贝出原有数据
    - `copy` 去从打开的 `xlrd` 的 `Book` 变量中，拷贝出一份，成为新的 `xlwt` 的 `Workbook` 变量
3. 对xlwt的Workbook变量的正常的操作
    i. 通过get_sheet去获得对应的sheet
    ii. 拿到sheet变量后，就可以往sheet中，写入新的数据
    iii. 写完新数据后，最终save保存

代码

```
import xlwt
import xlrd
#import xlutils
from xlutils.copy import copy

styleBoldRed   = xlwt.easyxf('font: color-index red, bold c
headerStyle = styleBoldRed
wb = xlwt.Workbook()
ws = wb.add_sheet(gConst['xls']['sheetName'])
ws.write(0, 0, "Header",        headerStyle)
ws.write(0, 1, "CatalogNumber", headerStyle)
ws.write(0, 2, "PartNumber",    headerStyle)
wb.save(gConst['xls']['fileName'])

#open existed xls file
#newWb = xlutils.copy(gConst['xls']['fileName'])
#newWb = copy(gConst['xls']['fileName'])
oldWb = xlrd.open_workbook(gConst['xls']['fileName'], forma
print oldWb #<xlrd.book.Book object at 0x000000000315C940>
newWb = copy(oldWb)
print newWb #<xlwt.Workbook.Workbook object at 0x000000000
newWs = newWb.get_sheet(0)
newWs.write(1, 0, "value1")
newWs.write(1, 1, "value2")
newWs.write(1, 2, "value3")
print "write new values ok"
newWb.save(gConst['xls']['fileName'])
print "save with same name ok"
```

常见问题

`AttributeError: 'module' object has no attribute 'copy'`

背景

代码：

```
import xlutils

newWb = xlutils.copy(gConst['xls']['fileName'])
```

报错：

```
AttributeError: 'module' object has no attribute 'copy'
```

原因：未知。可能是库本身的bug。

解决办法

改为：

```
from xlutils.copy import copy

newWb = copy(gConst['xls']['fileName'])
```

即可（规避此问题）

`AttributeError: 'str' object has no attribute 'datemode'`

背景

代码：

```
from xlutils.copy import copy;

newWb = copy(gConst['xls']['fileName'])
```

出错：

```
    newWb = copy(gConst['xls']['fileName']);
  File "D:\tmp\dev_install_root\Python27_x64\lib\site-packa
    w
  File "D:\tmp\dev_install_root\Python27_x64\lib\site-packa
    reader(chain[0])
  File "D:\tmp\dev_install_root\Python27_x64\lib\site-packa
    filter.workbook(workbook,filename)
  File "D:\tmp\dev_install_root\Python27_x64\lib\site-packa
    self.wtbook.dates_1904 = rdbook.datemode
AttributeError: 'str' object has no attribute 'datemode'
```

原因：参考官网的资料 xlutils copy，才知道 `copy` 的参数，是对应的 `workbook` ，而不是 `xls` 的 `filename`

解决办法：先从excel文件中（通过 `xlrd` ）读取得到 `workbook` ，再去用 `copy`

代码

```
import xlwt
import xlrd
#import xlutils
from xlutils.copy import copy

oldWb = xlrd.open_workbook(gConst['xls']['fileName'])
print oldWb; #<xlrd.book.Book object at 0x000000000315C940>
newWb = copy(oldWb)
print newWb; #<xlwt.Workbook.Workbook object at 0x00000000
```

才真正可以正常打开旧的xls，拷贝出一份新的xls

pandas

Python的科学计算方面的库，主要用于数据处理的 `pandas` ，也可以操作 `excel` 和 `csv`

- `pandas`
  - 官网API文档
    - Excel
      - 读： `pandas.read_excel`
        - pandas.read_excel — pandas 1.2.4 documentation
          - https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_excel.html
            - 语法
              - `pandas.read_excel(io, sheet_name=0, header=0, names=None, index_col=None, usecols=None, squeeze=False, dtype=None, engine=None, converters=None, true_values=None, false_values=None, skiprows=None, nrows=None, na_values=None, keep_default_na=True, na_filter=True, verbose=False, parse_dates=False, date_parser=None, thousands=None, comment=None, skipfooter=0, convert_float=True, mangle_dupe_cols=True, storage_options=None)`
      - 写： `pandas.DataFrame.to_excel`
        - pandas.DataFrame.to_excel — pandas 1.2.4 documentation
          - https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_excel.html
            - 语法
              - `DataFrame.to_excel(excel_writer, sheet_name='Sheet1', na_rep='', float_format=None,`

`columns=None, header=True, index=True, index_label=None, startrow=0, startcol=0, engine=None, merge_cells=True, encoding=None, inf_rep='inf', verbose=True, freeze_panes=None, storage_options=None)[source]`

- csv
  - 读：`pandas.read_csv`
    - pandas.read_csv — pandas 1.2.4 documentation
      - https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html
        - 语法
          - `pandas.read_csv(filepath_or_buffer, sep=<object object>, delimiter=None, header='infer', names=None, index_col=None, usecols=None, squeeze=False, prefix=None, mangle_dupe_cols=True, dtype=None, engine=None, converters=None, true_values=None, false_values=None, skipinitialspace=False, skiprows=None, skipfooter=0, nrows=None, na_values=None, keep_default_na=True, na_filter=True, verbose=False, skip_blank_lines=True, parse_dates=False, infer_datetime_format=False, keep_date_col=False, date_parser=None, dayfirst=False, cache_dates=True, iterator=False, chunksize=None, compression='infer', thousands=None, decimal='.', lineterminator=None, quotechar='"', quoting=0,`

`doublequote=True, escapechar=None, comment=None, encoding=None, dialect=None, error_bad_lines=True, warn_bad_lines=True, delim_whitespace=False, low_memory=True, memory_map=False, float_precision=None, storage_options=None)[source]`

- 写： `pandas.DataFrame.to_csv`
  - pandas.DataFrame.to_csv — pandas 1.2.4 documentation
    - https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html
      - 语法
        - `DataFrame.to_csv(path_or_buf=None, sep=',', na_rep='', float_format=None, columns=None, header=True, index=True, index_label=None, mode='w', encoding=None, compression='infer', quoting=None, quotechar='"', line_terminator=None, chunksize=None, date_format=None, doublequote=True, escapechar=None, decimal='.', errors='strict', storage_options=None)`

win32com.client

此处用 `win32com.client` 去处理 `Excel` 中的**图表**，比如 `Chart` 、 `Graph` 等。

处理Excel的图表

在当前文件夹中新建一个空的 `xls` 文件： `chart_demo.xls`

用（ `Python 2` 的）代码：

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
"""
Function:
【已解决】Python中处理操作Excel中的图表（Chart, Graph）
https://www.crifan.com/python_process_excel_chart_graph

Author:     Crifan Li
Version:    2012-12-25
Contact:    admin at crifan dot com
"""

import os
from win32com.client import Dispatch
#from win32com.client import *

def excelChart():
    xl = Dispatch("Excel.Application")
    #xl = win32com.client.Dispatch("Excel.Application")
    print "xl=",xl

    #[1] Fail
    # xlsPath = "chart_demo.xls"
    # wb = xl.Workbooks.open(xlsPath) #pywintypes.com_erro

    #[2] Fail
    # xlsPath = "D:\tmp\tmp_dev_root\python\excel_chart\cha
    # absPath = os.path.abspath(xlsPath)
    # print "absPath=",absPath #absPath= D:\tmp\tmp_dev_roo
    # wb = xl.Workbooks.open(absPath) #pywintypes.com_erro

    #[3] Fail
    # xlsPath = "D:\tmp\tmp_dev_root\python\excel_chart\cha
    # normalPath = os.path.normpath(xlsPath)
    # print "normalPath=",normalPath #normalPath= D:  mp
    # wb = xl.Workbooks.open(normalPath) #pywintypes.com_er

    #[4] Fail
    # rawPath = r"chart_demo.xls"
    # wb = xl.Workbooks.open(rawPath) #pywintypes.com_erro

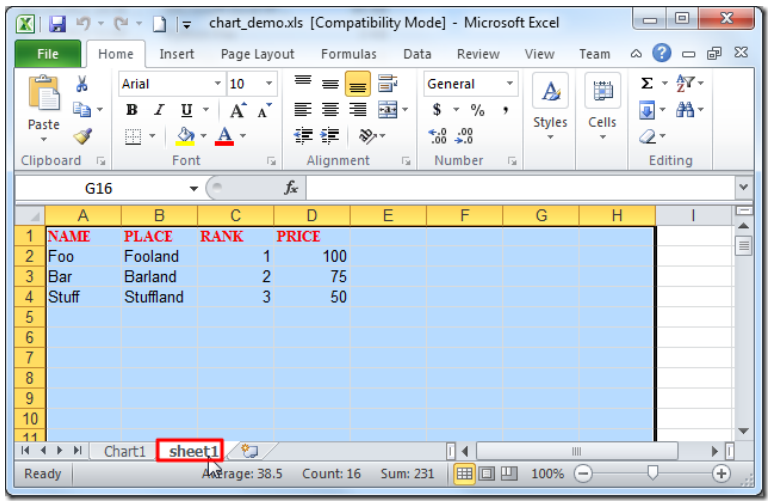
    #[5] OK
    xlsPath = "chart_demo.xls"
    absPath = os.path.abspath(xlsPath)
    print "absPath=",absPath #absPath= D:\tmp\tmp_dev_root\
    wb = xl.Workbooks.open(absPath) #OK

    #[6] OK
    # rawPath = r"D:\tmp\tmp_dev_root\python\excel_chart\ch
    # wb = xl.Workbooks.open(rawPath) # OK
```

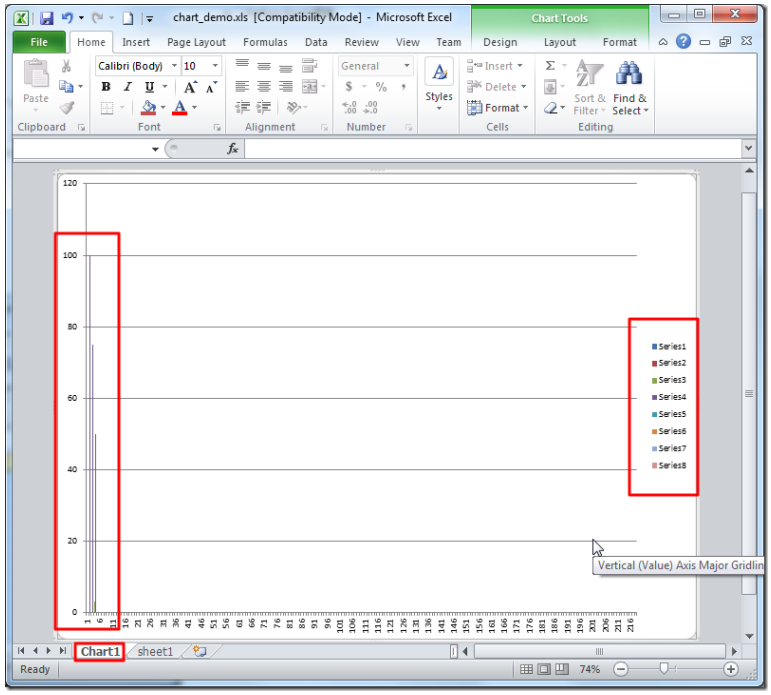
```
    xl.Visible = 1
    ws = wb.Worksheets(1)
    ws.Range('$A1:$D1').Value = ['NAME', 'PLACE', 'RANK',
    ws.Range('$A2:$D2').Value = ['Foo', 'Fooland', 1, 100]
    ws.Range('$A3:$D3').Value = ['Bar', 'Barland', 2, 75]
    ws.Range('$A4:$D4').Value = ['Stuff', 'Stuffland', 3, 5
    wb.Save()
    wb.Charts.Add()
    wc1 = wb.Charts(1)

if __name__ == "__main__":
    excelChart()
```

效果：

- Excel单元格内容
  - 

- Excel对应图表效果

- 

常见问题

`ImportError: No module named win32com.client`

代码：

```
from win32com.client import Dispatch
```

会报错：

```
D:\tmp\tmp_dev_root\python\excel_chart>excel_chart.py
Traceback (most recent call last):
  File "D:\tmp\tmp_dev_root\python\excel_chart\excel_chart.
    from win32com.client import Dispatch
ImportError: No module named win32com.client
```

解决办法：下载和安装 `pywin32`

具体步骤

去

http://sourceforge.net/projects/pywin32/files/pywin32/

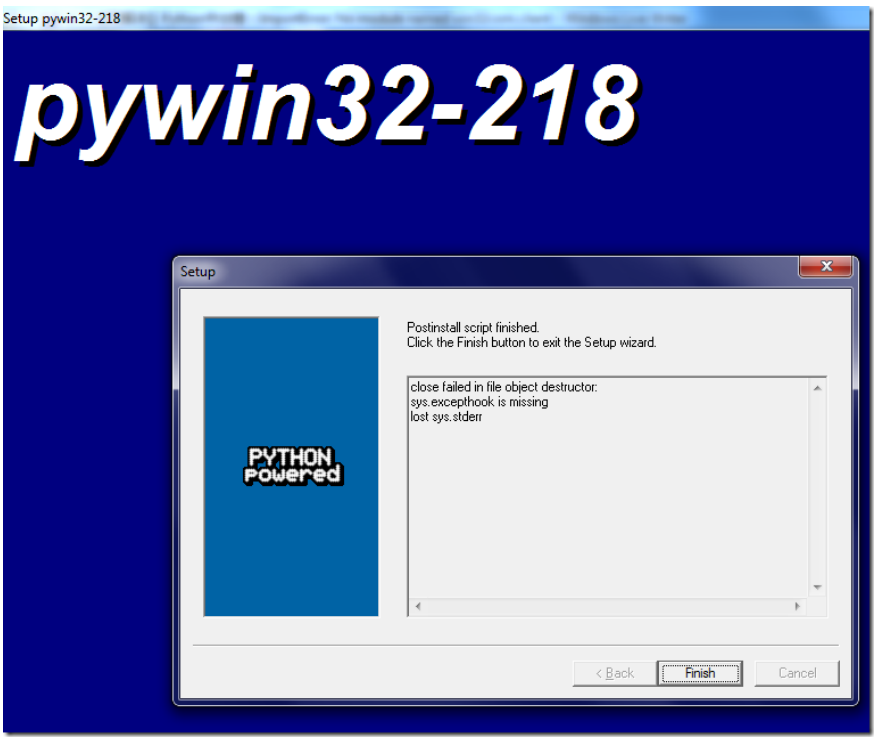
下载

http://sourceforge.net/projects/pywin32/files/pywin32/Build%20218/

此处 `Win7 64位 + Python 2.7.3` 对应的版本：

pywin32-218.win-amd64-py2.7.exe

然后去双击安装，一路默认设置：



即可。

附录

下面列出相关参考资料。

参考资料

- 【已解决】Python中如何读取csv文件中每一行的数据
- 【整理】用Python把列表或字典数据输出写入csv文件
- 【已解决】python的输出列表数据到csv但没有表头
-
- 【已解决】Python中创建和保存数据到csv文件中
- 【已解决】python解析excel文件并读取其中的sheet和row和column的值
- 【已解决】openpyxl中给单元格设置背景色
- 【部分解决】openpyxl的excel设置列宽自适应+设置行高 – 在路上
- 【已解决】openpyxl中设置单元格自动换行
- 【已解决】openpyxl中给一个范围内的单元格批量设置样式：居中对齐和背景色
- 【已解决】用openpyxl去新建excel文件并保存数据和设置单元格样式
- 【已解决】openpyxl代码警告：Using a range string with iter_rows is deprecated. Use ws[range_string]
- 【已解决】openpyxl生成excel文件打开提示：发现xlsx中的部分内容有问题 是否让我们尽量尝试恢复
- 【已解决】Python中openpyxl处理excel去判断单元格是合并后的以及合并的范围
- 【整理】Python中，添加写入数据到已经存在的Excel的xls文件，即打开excel文件，写入新数据 – 在路上
- 【记录】Python中生成（写入数据到）Excel文件中
- 【记录】Python中安装xlrd模块
- 【记录】Python中安装可以读写excel的xls文件的xlutils模块（需依赖于xlrd和xlwt）
- 【已解决】Python中处理操作Excel中的图表（Chart，Graph） – 在路上
- 【已解决】Python中出错：ImportError: No module named win32com.client – 在路上
- 【记录】Mac中用Python去解析excel文件 – 在路上
- 【已解决】Mac中安装Python的第三方库：openpyxl
- 【整理】python中处理excel的库xlrd和xlwt的python 3.x版本是：xlrd3和xlwt3 – 在路上
- 【已解决】Python中使用xlutils.copy出错：AttributeError: ‘module’ object has no attribute ‘copy’ – 在路上
- 【已解决】Python中使用xlutils的copy出错：AttributeError: ‘str’ object has no attribute ‘datemode’ – 在路上
-
- 第 6 章 Python中的CSV,Excel

-
- 【python】pandas库pd.read_excel操作读取excel文件参数整理与实例_brucewong0516的博客-CSDN博客
- 最新Pandas.read_excel()全参数详解（案例实操，如何利用python导入excel） - 知乎
- pandas.read_excel 详细介绍 | Pandas 中文教程 - 盖若
- Python Resources for working with Excel - Working with Excel Files in Python
- 请教下 Python 高手，如何用 Python 自动化操作 Excel? - 知乎
- python - How to write header row with csv.DictWriter? - Stack Overflow
- Openpyxl tutorial - read, write Excel xlsx files in Python
-