**Python-Excel 模块哪家强？**

[zx576](https://www.jianshu.com/u/787b549ea956) 关注

2017.04.21 11:23\* 字数 1387 阅读 20711评论 2喜欢 40

**前言**

从网页爬下来的大量数据需要excel清洗  
成堆的科学实验数据需要导入excel进行分析  
作为一名面向逼格的Python程序员  
该如何合理而又优雅的选择生产力工具呢？

得益于辛勤劳作的python大神们，处理excel已经有大量python包，主流代表有：

* [**xlwings**](https://link.jianshu.com/?t=http://docs.xlwings.org/en/stable/)：简单强大，可替代VBA
* [**openpyxl**](https://link.jianshu.com/?t=https://openpyxl.readthedocs.io/en/latest/)：简单易用，功能广泛
* [**pandas**](https://link.jianshu.com/?t=http://pandas.pydata.org/)：使用需要结合其他库，数据处理是pandas立身之本
* [**win32com**](https://link.jianshu.com/?t=http://pythonexcels.com/python-excel-mini-cookbook/)：不仅仅是excel，可以处理office;
* [**Xlsxwriter**](https://link.jianshu.com/?t=https://xlsxwriter.readthedocs.io/)：丰富多样的特性，直接创造一份美观大方的excel，代码即一切；
* [**DataNitro**](https://link.jianshu.com/?t=https://datanitro.com/)：作为插件内嵌到excel中，可替代VBA，在excel中优雅的使用python
* **xlutils**：结合xlrd/xlwt，老牌python包，需要注意的是你必须同时安装这三个库

面对形形色色的扩展包，有时候会感到困惑，到底哪个包才是最适合自己的呢?  
本文将从配置环境、文档操作、基本功能等方面比较以上扩展，让您能结合自己的生产环境，选择最适合自己的Excel操作库，同时自信的对其他库：你是个好扩展，可我们不合适！  
下面，我们通过多方面的比较，让您对这些扩展有一个基础的了解。

**1.环境配置**

再好的模块，也需要在正确的 Python 版本以及 Excel 版本才可运行。



Paste\_Image.png

所有库都支持Python2和python3。  
需要注意的是Xlutils仅支持xls文件，即2003以下版本。同时win32com与DataNitro仅支持windows

**2.文档操作**

由于设计模式的不同，导致基本的新建文件、修改文件、保存文件等功能在不同的库中存在着一定差异，比如xlsxwriter并不支持打开或修改现有文件，xlwings不支持对新建文件的命名等等，DataNitro作为excel插件依托于excel本身的操作。详见下图



Paste\_Image.png

**3.基本功能**

由于设计目的不同，每个模块通常着重于某一方面功能，各有所长。

xlwings

可结合 VBA 实现对 Excel 编程，强大的数据输入分析能力，同时拥有丰富的接口，结合 pandas/numpy/matplotlib 轻松应对 Excel 数据处理工作。

openpyxl

简单易用，功能广泛，单元格格式/图片/表格/公式/筛选/批注/文件保护等等功能应有尽有，图表功能是其一大亮点，缺点是对 VBA 支持的不够好。

pandas

数据处理是 pandas 的立身之本，Excel 作为 pandas 输入/输出数据的容器。

win32com

从命名上就可以看出，这是一个处理 windows 应用的扩展，Excel 只是该库能实现的一小部分功能。该库还支持 office 的众多操作。需要注意的是，该库不单独存在，可通过安装 pypiwin32 或者 pywin32 获取。

xlsxwriter

拥有丰富的特性，支持图片/表格/图表/筛选/格式/公式等，功能与openpyxl相似，优点是相比 openpyxl 还支持 VBA 文件导入，迷你图等功能，缺点是不能打开/修改已有文件，意味着使用 xlsxwriter 需要从零开始。

DataNitro

作为插件内嵌到 Excel 中，可完全替代 VBA，在 Excel 中使用 python 脚本。既然被称为 Excel 中的 python，协同其他 python 库亦是小事一桩。然而，这是付费插件...

xlutils

基于 xlrd/xlwt，老牌 python 包，算是该领域的先驱，功能特点中规中矩，比较大的缺点是仅支持 xls 文件。

**4.性能**

我们对几个库做了最基本的写入和读取测试，分别使用不同库进行添加及读取 1000行 \* 700列 数据操作，得到所用时间，重复操作取平均值。另外在不同的电脑配置，不同的环境下结果肯定会有出入，数据仅供参考。



Paste\_Image.png

**5.小结**

通过以上的分析，相信大家对几个库都有了简单的了解。在编写文章的过程中，笔者也在思考各个库最适合的应用场景。

1. 不想使用 GUI 而又希望赋予 Excel 更多的功能，openpyxl 与 xlsxwriter，你可二者选其一；
2. 需要进行科学计算，处理大量数据，建议 pandas+xlsxwriter 或者 pandas+openpyxl；
3. 想要写 Excel 脚本，会 Python 但不会 VBA 的同学，可考虑 xlwings 或 DataNitro；
4. 至于 win32com，不管是功能还是性能都很强大，有 windows 编程经验的同学可以使用。不过它相当于是 windows COM 的封装，自身并没有很完善的文档，新手使用起来略有些痛苦。

你可根据自己的需求和生产环境，选择合适的 Python-Excel 模块。

**6.代码**

**6.1 xlwings基本代码**

import xlwings as xw

*#连接到excel*

workbook = xw.Book(r'path/myexcel.xlsx')*#连接excel文件*

*#连接到指定单元格*

data\_range = workbook.sheets('Sheet1').range('A1')

*#写入数据*

data\_range.value = [1,2,3]

*#保存*

workbook.save()

**6.2 xlsxwriter基本代码**

import xlsxwriter as xw

*#新建excel*

workbook = xw.Workbook('myexcel.xlsx')

*#新建工作薄*

worksheet = workbook.add\_worksheet()

*#写入数据*

worksheet.wirte('A1',1)

*#关闭保存*

workbook.close()

**6.3 xlutils基本代码**

import xlrd *#读取数据*

import xlwt *#写入数据*

import xlutils *#操作excel*

-----*#xlrd库*

*#打开excel文件*

workbook = xlrd.open\_workbook('myexcel.xls')

*#获取表单*

worksheet = workbook.sheet\_by\_index(0)

*#读取数据*

data = worksheet.cell\_value(0,0)

----*#xlwt库*

*#新建excel*

wb = xlwt.Workbook()

*#添加工作薄*

sh = wb.add\_sheet('Sheet1')

*#写入数据*

sh.write(0,0,'data')

*#保存文件*

wb.save('myexcel.xls')

-----*#xlutils库*

*#打开excel文件*

book = xlrd.open\_workbook('myexcel.xls')

*#复制一份*

new\_book = xlutils.copy(book)

*#拿到工作薄*

worksheet = new\_book.getsheet(0)

*#写入数据*

worksheet.write(0,0,'new data')

*#保存*

new\_book.save()

**6.4 win32com基本代码**

import win32com.client as wc

*#启动Excel应用*

excel\_app = wc.Dispatch('Excel.Application')

*#连接excel*

workbook = excel\_app.Workbooks.Open(r'e:/myexcel.xlsx' )

*#写入数据*

workbook.Worksheets('Sheet1').Cells(1,1).Value = 'data'

*#关闭并保存*

workbook.SaveAs('newexcel.xlsx')

excel\_app.Application.Quit()

**6.5 openpyxl基本代码**

import openpyxl

*# 新建文件*

workbook = openpyxl.Workbook()

*# 写入文件*

sheet = workbook.activesheet['A1']='A1'

*# 保存文件*

workbook.save('test.xlsx')