**openpyxl模块介绍**

　　openpyxl模块是一个读写Excel 2010文档的Python库，如果要处理更早格式的Excel文档，需要用到额外的库，openpyxl是一个比较综合的工具，能够同时读取和修改Excel文档。其他很多的与Excel相关的项目基本只支持读或者写Excel一种功能。

**安装openpyxl模块**

　　openpyxl是一个开源项目，这里使用如下命令安装openpyxl模块

pip3 install openpyxl

**openpyxl基本用法**

　　想要操作Excel首先要了解Excel 基本概念，Excel中列以字幕命名，行以数字命名，比如左上角第一个单元格的坐标为A1，下面的为A2，右边的B1。

　　openpyxl中有三个不同层次的类，Workbook是对工作簿的抽象，Worksheet是对表格的抽象，Cell是对单元格的抽象，每一个类都包含了许多属性和方法。

**操作Excel的一般场景:**

1. 打开或者创建一个Excel需要创建一个Workbook对象
2. 获取一个表则需要先创建一个Workbook对象，然后使用该对象的方法来得到一个Worksheet对象
3. 如果要获取表中的数据，那么得到Worksheet对象以后再从中获取代表单元格的Cell对象

**Workbook对象**

　　一个Workbook对象代表一个Excel文档，因此在操作Excel之前，都应该先创建一个Workbook对象。对于创建一个新的Excel文档，直接进行Workbook类的调用即可，对于一个已经存在的Excel文档，可以使用openpyxl模块的load\_workbook函数进行读取，该函数包涵多个参数，但只有filename参数为必传参数。filename 是一个文件名，也可以是一个打开的文件对象。

>>> import openpyxl

>>> excel = openpyxl.Workbook(‘hello.xlxs‘)

>>> excel1 = openpyxl.load\_workbook(‘abc.xlsx‘)

>>>

**PS：Workbook和load\_workbook相同，返回的都是一个Workbook对象。**

**Workbook对象提供了很多属性和方法，其中，大部分方法都与sheet有关，部分属性如下：**

* active：获取当前活跃的Worksheet
* worksheets：以列表的形式返回所有的Worksheet(表格)
* read\_only：判断是否以read\_only模式打开Excel文档
* encoding：获取文档的字符集编码
* properties：获取文档的元数据，如标题，创建者，创建日期等
* sheetnames：获取工作簿中的表（列表）

>>> import openpyxl

>>> excel2 = openpyxl.load\_workbook(‘abc.xlsx‘)

>>> excel2.active

<Worksheet "abc">

>>> excel.read\_only

False

>>> excel2.worksheets

[<Worksheet "abc">, <Worksheet "def">]

>>> excel2.properties

<openpyxl.packaging.core.DocumentProperties object>

Parameters:

creator=‘openpyxl‘, title=None, description=None, subject=None, identifier=None, language=None, created=datetime.datetime(2006, 9, 16, 0, 0), modified=datetime.datetime(2018, 2, 5, 7, 25, 18), lastModifiedBy=‘Are you SuperMan‘, category=None, contentStatus=None, version=None, revision=None, keywords=None, lastPrinted=None

>>> excel2.encoding

‘utf-8‘

>>>

**Workbook提供的方法如下：**

* get\_sheet\_names：获取所有表格的名称(新版已经不建议使用，通过Workbook的sheetnames属性即可获取)
* get\_sheet\_by\_name：通过表格名称获取Worksheet对象(新版也不建议使用，通过Worksheet[‘表名‘]获取)
* get\_active\_sheet：获取活跃的表格(新版建议通过active属性获取)
* remove\_sheet：删除一个表格
* create\_sheet：创建一个空的表格
* copy\_worksheet：在Workbook内拷贝表格

>>> excel2.get\_sheet\_names()

Warning (from warnings module):

File "\_\_main\_\_", line 1

DeprecationWarning: Call to deprecated function get\_sheet\_names (Use wb.sheetnames).

[‘abc‘, ‘def‘]

>>> excel2.sheetnames

[‘abc‘, ‘def‘]

>>> excel2.get\_sheet\_by\_name(‘def‘)

Warning (from warnings module):

File "\_\_main\_\_", line 1

DeprecationWarning: Call to deprecated function get\_sheet\_by\_name (Use wb[sheetname]).

<Worksheet "def">

>>> excel2[‘def‘]

<Worksheet "def">

>>> excel2.get\_active\_sheet()

Warning (from warnings module):

File "\_\_main\_\_", line 1

DeprecationWarning: Call to deprecated function get\_active\_sheet (Use the .active property).

<Worksheet "abc">

>>> excel2.create\_sheet(‘ghk‘)

<Worksheet "ghk">

**Worksheet对象**

　　有了Worksheet对象以后，我们可以通过这个Worksheet对象获取表格的属性，得到单元格中的数据，修改表格中的内容。openpyxl提供了非常灵活的方式来访问表格中的单元格和数据，**常用的Worksheet属性如下：**

* title：表格的标题
* dimensions：表格的大小，这里的大小是指含有数据的表格的大小，即：左上角的坐标:右下角的坐标
* max\_row：表格的最大行
* min\_row：表格的最小行
* max\_column：表格的最大列
* min\_column：表格的最小列
* rows：按行获取单元格(Cell对象) - 生成器
* columns：按列获取单元格(Cell对象) - 生成器
* freeze\_panes：冻结窗格
* values：按行获取表格的内容(数据)  - 生成器

PS：freeze\_panes，参数比较特别，主要用于在表格较大时冻结顶部的行或左边的行。对于冻结的行，在用户滚动时，是始终可见的，可以设置为一个Cell对象或一个端元个坐标的字符串，单元格上面的行和左边的列将会冻结(单元格所在的行和列不会被冻结)。例如我们要冻结第一行那么设置A2为freeze\_panes，如果要冻结第一列，freeze\_panes取值为B1，如果要同时冻结第一行和第一列，那么需要设置B2为freeze\_panes，freeze\_panes值为none时 表示 不冻结任何列。

**常用的Worksheet方法如下:**

* iter\_rows：按行获取所有单元格，内置属性有(min\_row,max\_row,min\_col,max\_col)
* iter\_columns：按列获取所有的单元格
* append：在表格末尾添加数据
* merged\_cells：合并多个单元格
* unmerged\_cells：移除合并的单元格

>>> for row in excel2[‘金融‘].iter\_rows(min\_row=2,max\_row=4,min\_col=2,max\_col=4):

print(row)

(<Cell ‘abc‘.B2>, <Cell ‘abc‘.C2>, <Cell ‘abc‘.D2>)

(<Cell ‘abc‘.B3>, <Cell ‘abc‘.C3>, <Cell ‘abc‘.D3>)

(<Cell ‘abc‘.B4>, <Cell ‘abc‘.C4>, <Cell ‘abc‘.D4>)

PS：从Worksheet对象的属性和方法可以看到，大部分都是返回的是一个Cell对象，一个Cell对象代表一个单元格，我们可以使用Excel坐标的方式来获取Cell对象，也可以使用Worksheet的cell方法获取Cell对象。

>>> excel2[‘abc‘][‘A1‘]

<Cell ‘abc‘.A1>

>>> excel2[‘abc‘].cell(row=1,column=2)

<Cell ‘abc‘.B1>

>>>

**Cell对象**

　　Cell对象比较简单，常用的属性如下:

* row：单元格所在的行
* column：单元格坐在的列
* value：单元格的值
* coordinate：单元格的坐标

>>> excel2[‘abc‘].cell(row=1,column=2).coordinate

‘B1‘

>>> excel2[‘abc‘].cell(row=1,column=2).value

‘test‘

>>> excel2[‘abc‘].cell(row=1,column=2).row

1

>>> excel2[‘abc‘].cell(row=1,column=2).column

‘B‘

**打印表中数据的几种方式**

# ---------- 方式1 ----------

>>> for row in excel2[‘abc‘].rows:

print( \*[ cell.value for cell in row ])

# ---------- 方式2 ----------

>>> for row in excel2[‘abc‘].values:

print(\*row)