**最详细的Excel模块Openpyxl教程（三）-使用公式**

在之前的推文中，我们介绍了操作Excel的模块openpyxl的cell单元格操作，相关推文可以从本公众号的底部相关菜单获取。接下来的推文我们来学习一下openpyxl这个python模块中的其他知识，想了想还是先来学习一下怎么借助openpyxl来进行Excel中公式的设置。

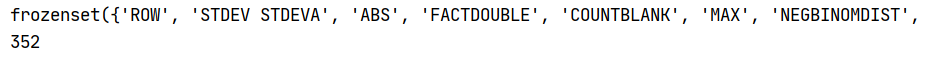
**一．公式数量和种类**

我们先看一下在openpyxl中能使用的公式有哪些，我们来看代码：

from openpyxl.utils import FORMULAE

print(len(FORMULAE))  
print(FORMULAE)

**输出的内容如下：**



可以看出我支持的公式有ROW，ABS，MAX等等。支持的公式的数量可能因为版本不同而不一样，目前我使用的openpyxl的版本是3.0.5，支持公式的数量为352。

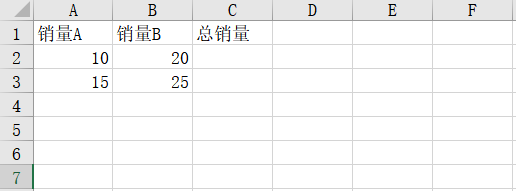
那么我们怎么判断自己要使用的公式是否支持，则可以简单的进行判断：

print(**'MID'** in FORMULAE)  
print(**'minddd'** in FORMULAE)

上述代码输出**True**和**False**，也就是说公式“MID”支持，公式‘minddd’是不支持的。

**二．在单元格中使用公式**

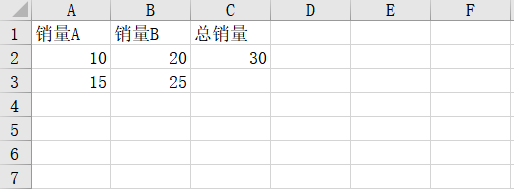
假设我们现在有一个数据表“**formulae\_1.xlsx**”，数据显示如下：



我们将使用公式来计算一个总值，代码如下：

from openpyxl import load\_workbook  
wbook = load\_workbook(filename=**'formula\_1.xlsx'**)  
wsheet = wbook[**'Sheet1'**]  
  
wsheet[**"C2"**] = **"=SUM(A2,B2)"**print(wsheet[**'C2'**].value)  
wbook.save(**"formula\_1.xlsx"**)

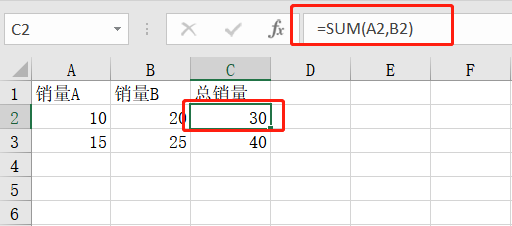
代码中的输出为：**=SUM(A2,B2)**，操作后的表格显示如下，正好是两者之和。当然了上述第四行代码我们也可以这样写：**wsheet["C2"] = "=SUM(10,20)"**，也能得到我们下图的结果：



利用for循环，我们可以将上述表格中的所有行的值：

from openpyxl import load\_workbook  
wbook = load\_workbook(filename=**'formula\_1.xlsx'**)  
wsheet = wbook[**'Sheet1'**]for j in range(2,4):  
 cell\_a = **'A'** + str(j)  
 cell\_b = **'B'** + str(j)  
 cell\_c = **'C'** + str(j)  
 wsheet[cell\_c] = **"=SUM({},{})"**.format(cell\_a,cell\_b)  
wbook.save(**"formula\_1.xlsx"**)

我们来看一下表格中的数据，选中单元格C2，可以看出这个单元格显示的是一个公式：



那我们重新加载这个表格并读取 这个单元格会出现什么呢？

from openpyxl import load\_workbook  
wbook = load\_workbook(filename=**'formula\_1.xlsx'**)  
wsheet = wbook[**'Sheet1'**]cell\_C2 = wsheet.cell(2,3).value  
print(cell\_C2)  
wbook.save(**"formula\_1.xlsx"**)

代码中的打印输出为：**=SUM(A2,B2)。**那么这里就存在一个问题，我们怎么得到这个单元格中计算的值呢？也就是如何打印出30。这里就要提及到openpyxl中的一个打开文件时的参数了，我们将第二行代码更改为：

wbook = load\_workbook(filename=**'formula\_1.xlsx'**,data\_only=True)

load\_workbook方法中涉及到较多的属性，包括read\_only，keep\_vba，guess\_types，data\_only等。**data\_only用于读取cell中的值，当单元格中的值是一个公式的时候，会返回计算到的结果。data\_only 控制具有公式的单元格是否具有公式（默认值）或上次Excel读取工作表时存储的值。**

我们运行替换后的代码的后，打印cell\_C2值的时候是为却为**None**，这是为什么呢？查询相关的资料的解释可以参考：

<https://blog.51cto.com/antidarkness/1974684>

那么这个问题该怎么解决呢？我们可以使用win32库进行打开关闭的操作之后，在使用openpyxl打开文件传入**data\_only=True**即可，使用win32打开保存的代码如下：（注意filaname需要传一个全路径）

from win32com.client import Dispatch

filename = **r'C:\Users\LEGION\Desktop\tweets\_code\formula\_1.xlsx'**  
xlApp = Dispatch(**'Excel.Application'**)  
xlApp.Visible = False  
xlBook = xlApp.Workbooks.Open()  
xlBook.Save()  
xlBook.Close()

当然了，在计算多行值的时候如A1到A5的和的时候，可以使用：**"=SUM(A1:A5)"。**

**三．转换公式位置**

将公式从一个位置转换到另一个位置在日常的工作中也是非常常见的，那么该怎么实现呢？

我们先看一个代码：

from openpyxl import load\_workbook  
from openpyxl.formula.translate import Translator  
wbook = load\_workbook(filename=**'formula\_1.xlsx'**)wsheet = wbook[**'Sheet1'**]wsheet[**'C3'**] = Translator(**"=SUM(A2,B2)"**, origin=**"C2"**).translate\_formula(**"C3"**)  
print(wsheet[**'C3'**].value)  
wbook.save(**"formula\_1.xlsx"**)

上述代码将**"=SUM(A2,B2)"**显示的公式，通过转变复制到C3单元格中了，print打印的结果为：=SUM(A3,B3)。C3单元格中的值也就变成了A3+B3的值了，是不是很有意思。

官方文档对这个Translator(object)类的作用以及参数的解释为：**Modifies a formula so that it can be translated from one cell to another.****formula:** *The str string to translate. Must include the leading '=' character.***origin:** *The cell address (in A1 notation) where this formula was defined (excluding the worksheet name).*

**这里补充上次单元格中的一个知识点：**

当我们在设置格式的时候没有自己想要的格式，那该怎么办呢？很简单，我们可以先使用Excel设置相应单元格的样式，然后使用下述代码打印格式：

cell\_style = wsheet.cell(i,j).number\_format  
 print(cell\_style)

**三．总结**

以上就是本次的推文，推文介绍的单元格中使用公式的相关操作，大家跟着学习的时候最好也跟着实践一下。后期我们将继续介绍其他方面的知识。**如进行表格的数据的筛选过滤等操作。**大家在学习的时候有什么疑问，也欢迎在评论区留言。