**[python操作Excel模块openpyxl](https://www.cnblogs.com/zeke-python-road/p/8986318.html)**

**来源：https://www.cnblogs.com/zeke-python-road/p/8986318.html**

**1、 安装**

pip install openpyxl

想要在文件中插入图片文件，需要安装pillow，安装文件：PIL-fork-1.1.7.win-amd64-py2.7.exe

· font(字体类)：字号、字体颜色、下划线等

· fill(填充类)：颜色等

· border(边框类)：设置单元格边框

· alignment(位置类)：对齐方式

· number\_format(格式类)：数据格式

· protection(保护类)：写保护

**2、 创建一个excel 文件，并写入不同类的内容**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
wb **=**Workbook()    #创建文件对象  
  
*# grab the active worksheet*ws **=**wb.active     #获取第一个sheet  
  
*# Data can be assigned directly to cells*ws['A1'] **=**42      #写入数字  
ws['B1'] **=**"你好"+"automation test" #写入中文（unicode中文也可）  
  
*# Rows can also be appended*ws.append([1, 2, 3])    #写入多个单元格  
  
*# Python types will automatically be converted***import**datetime  
**import**time  
ws['A2'] **=**datetime.datetime.now()    #写入一个当前时间

*#写入一个自定义的时间格式*  
ws['A3'] **=**time.strftime("%Y年%m月%d日 %H时%M分%S秒",time.localtime())  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**3、 创建sheet**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
wb **=**Workbook()  
  
ws1 **=**wb.create\_sheet("Mysheet")           *#创建一个sheet*ws1.title **=**"New Title"                    *#设定一个sheet的名字*ws2 **=**wb.create\_sheet("Mysheet", 0)      *#设定sheet的插入位置 默认插在后面*ws2.title **= u"**你好**"***#设定一个sheet的名字 必须是Unicode*ws1.sheet\_properties.tabColor **=**"1072BA"   *#设定sheet的标签的背景颜色  
  
#获取某个sheet对象***print**wb.get\_sheet\_by\_name(**u"**你好**"**)  
**print**wb["New Title" ]  
  
*#获取全部sheet 的名字，遍历sheet名字***print**wb.sheetnames  
**for**sheet\_name **in**wb.sheetnames**:  
    print**sheet\_name  
  
**print**"\*"**\***50  
  
**for**sheet **in**wb**:  
    print**sheet.title  
  
*#复制一个sheet*wb["New Title" ]["A1"]**=**"zeke"  
source **=**wb["New Title" ]  
target **=**wb.copy\_worksheet(source)

# w3 = wb.copy\_worksheet(wb['new title'])

# ws3.title = 'new2'

# wb.copy\_worksheet(wb['new title']).title = 'hello'  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**4、 操作单元格**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
wb **=**Workbook()  
  
ws1 **=**wb.create\_sheet("Mysheet")           *#创建一个sheet*ws1["A1"]**=**123.11  
ws1["B2"]**=**"你好"  
d **=**ws1.cell(row**=**4, column**=**2, value**=**10)  
  
**print**ws1["A1"].value  
**print**ws1["B2"].value  
**print**d.value  
  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**5、 操作批量的单元格**

无论ws.rows还是ws.iter\_rows都是一个对象

除上述两个对象外 单行，单列都是一个元祖，多行多列是二维元祖

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
wb **=**Workbook()  
  
ws1 **=**wb.create\_sheet("Mysheet")           *#创建一个sheet*ws1["A1"]**=**1  
ws1["A2"]**=**2  
ws1["A3"]**=**3  
  
ws1["B1"]**=**4  
ws1["B2"]**=**5  
ws1["B3"]**=**6  
  
ws1["C1"]**=**7  
ws1["C2"]**=**8  
ws1["C3"]**=**9  
  
*#操作单列***print**ws1["A"]  
**for**cell **in**ws1["A"]**:  
    print**cell.value  
  
*#操作多列,获取每一个值***print**ws1["A:C"]  
**for**column **in**ws1["A:C"]**:  
    for**cell **in**column**:  
        print**cell.value  
  
*#操作多行*row\_range **=**ws1[1**:**3]  
**print**row\_range  
**for**row **in**row\_range**:  
    for**cell **in**row**:  
        print**cell.value  
  
**print**"\*"**\***50  
**for**row **in**ws1.iter\_rows(min\_row**=**1, min\_col=1, max\_col**=**3, max\_row**=**3)**:  
    for**cell **in**row**:  
        print**cell.value  
  
*#获取所有行***print**ws1.rows  
**for**row **in**ws1.rows**:  
    print**row  
  
**print**"\*"**\***50  
*#获取所有列***print**ws1.columns  
**for**col **in**ws1.columns**:  
    print**col *# Save the file*wb.save("e:**\\**sample.xlsx")

使用百分数

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl **import**load\_workbook  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
wb.guess\_types **=**True  
ws**=**wb.active  
ws["D1"]**=**"12%"  
**print**ws["D1"].value  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

#结果会打印小数

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl **import**load\_workbook  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
wb.guess\_types **=**False  
ws**=**wb.active  
ws["D1"]**=**"12%"  
**print**ws["D1"].value  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

#结果会打印百分数

**获取所有的行对象：**

#coding=utf-8

from openpyxl import Workbook

from openpyxl import load\_workbook

wb = load\_workbook('e:\\sample.xlsx')

ws=wb.active

rows=[]

for row in ws.iter\_rows():

            rows.append(row)

print rows   #所有行

print rows[0] #获取第一行

print rows[0][0] #获取第一行第一列的单元格对象

print rows[0][0].value #获取第一行第一列的单元格对象的值

print rows[len(rows)-1] #获取最后行 print rows[-1]

print rows[len(rows)-1][len(rows[0])-1] #获取第后一行和最后一列的单元格对象

print rows[len(rows)-1][len(rows[0])-1].value #获取第后一行和最后一列的单元格对象的值

**获取所有的列对象：**

#coding=utf-8

from openpyxl import Workbook

from openpyxl import load\_workbook

wb = load\_workbook('e:\\sample.xlsx')

ws=wb.active

cols=[]

cols = []

for col in ws.iter\_cols():

    cols.append(col)

print cols   #所有列

print cols[0]   #获取第一列

print cols[0][0]   #获取第一列的第一行的单元格对象

print cols[0][0].value   #获取第一列的第一行的值

print "\*"\*30

print cols[len(cols)-1]   #获取最后一列

print cols[len(cols)-1][len(cols[0])-1]   #获取最后一列的最后一行的单元格对象

print cols[len(cols)-1][len(cols[0])-1].value   #获取最后一列的最后一行的单元格对象的值

**6、 操作已经存在的文件**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl **import**load\_workbook  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
wb.guess\_types **=**True   #猜测格式类型  
ws**=**wb.active  
ws["D1"]**=**"12%"  
**print**ws["D1"].value  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

*#注意如果原文件有一些图片或者图标，则保存的时候可能会导致图片丢失*

**7、 单元格类型**

8、 *# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl **import**load\_workbook  
**import**datetime  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
  
ws**=**wb.active  
wb.guess\_types **=**True  
  
ws["A1"]**=**datetime.datetime(2010, 7, 21)  
**print**ws["A1"].number\_format  
  
ws["A2"]**=**"12%"  
**print**ws["A2"].number\_format  
  
ws["A3"]**=**1.1  
**print**ws["A4"].number\_format  
  
ws["A4"]**=**"中国"  
**print**ws["A5"].number\_format  
*# Save the file*wb.save("e:**\\**sample.xlsx")

执行结果：

yyyy-mm-dd h:mm:ss

0%

General

General

#如果是常规，显示general,如果是数字，显示'0.00\_ '，如果是百分数显示0%

数字需要在Excel中设置数字类型，直接写入的数字是常规类型

**8、 使用公式**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl **import**load\_workbook  
  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
  
ws1**=**wb.active  
  
ws1["A1"]**=**1  
ws1["A2"]**=**2  
ws1["A3"]**=**3  
  
ws1["A4"] **=**"=SUM(1, 1)"  
ws1["A5"] **=**"=SUM(A1:A3)"  
  
**print**ws1["A4"].value  *#打印的是公式内容，不是公式计算后的值,程序无法取到计算后的值***print**ws1["A5"].value  *#打印的是公式内容，不是公式计算后的值,程序无法取到计算后的值  
  
# Save the file*wb.save("e:**\\**sample.xlsx")

**9、 合并单元格**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl **import**load\_workbook  
  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
  
ws1**=**wb.active  
  
**from**openpyxl.workbook **import**Workbook  
  
wb **=**Workbook()  
ws **=**wb.active  
  
ws.merge\_cells('A2:D2')  
ws.unmerge\_cells('A2:D2')  #合并后的单元格，脚本单独执行拆分操作会报错，需要重新执行合并操作再拆分  
  
*# or equivalently*ws.merge\_cells(start\_row**=**2,start\_column**=**1,end\_row**=**2,end\_column**=**4)  
ws.unmerge\_cells(start\_row**=**2,start\_column**=**1,end\_row**=**2,end\_column**=**4)  
  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**10、插入一个图片**

需要先安装Pilow,安全文件是：PIL-fork-1.1.7.win-amd64-py2.7.exe

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**load\_workbook  
**from**openpyxl.drawing.image **import**Image  
  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
  
ws1**=**wb.active  
  
img **=**Image('e:**\\**1.png')  
ws1.add\_image(img, 'A1')  
  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**11、 隐藏单元格**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**load\_workbook  
**from**openpyxl.drawing.image **import**Image  
  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
  
ws1**=**wb.active  
  
ws1.column\_dimensions.group('A', 'D', hidden**=**True)   *#隐藏a到d列范围内的列  
#ws1.row\_dimensions 无group方法  
# Save the file*wb.save("e:**\\**sample.xlsx")

**12、 画一个柱状图**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**load\_workbook  
**from**openpyxl **import**Workbook  
**from**openpyxl.chart **import**BarChart, Reference, Series  
  
wb **=**load\_workbook('e:**\\**sample.xlsx')  
  
ws1**=**wb.active  
  
wb **=**Workbook()  
ws **=**wb.active  
**for**i **in**range(10)**:**ws.append([i])  
  
  
values **=**Reference(ws, min\_col**=**1, min\_row**=**1, max\_col**=**1, max\_row**=**10)  
chart **=**BarChart()  
chart.add\_data(values)  
ws.add\_chart(chart, "E15")  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**13、 画一个饼图**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**load\_workbook  
**from**openpyxl **import**Workbook  
  
  
**from**openpyxl.chart **import**(  
    PieChart,  
    ProjectedPieChart,  
    Reference  
)  
**from**openpyxl.chart.series **import**DataPoint  
  
data **=**[  
    ['Pie', 'Sold'],  
    ['Apple', 50],  
    ['Cherry', 30],  
    ['Pumpkin', 10],  
    ['Chocolate', 40],  
]  
  
wb **=**Workbook()  
ws **=**wb.active  
  
**for**row **in**data**:**ws.append(row)  
  
pie **=**PieChart()  
labels **=**Reference(ws, min\_col**=**1, min\_row**=**2, max\_row**=**5)  
data **=**Reference(ws, min\_col**=**2, min\_row**=**1, max\_row**=**5)  
pie.add\_data(data, titles\_from\_data**=**True)  
pie.set\_categories(labels)  
pie.title **=**"Pies sold by category"  
  
*# Cut the first slice out of the pie*slice **=**DataPoint(idx**=**0, explosion**=**20)  
pie.series[0].data\_points **=**[slice]  
  
ws.add\_chart(pie, "D1")  
  
  
ws **=**wb.create\_sheet(title**=**"Projection")  
  
data **=**[  
    ['Page', 'Views'],  
    ['Search', 95],  
    ['Products', 4],  
    ['Offers', 0.5],  
    ['Sales', 0.5],  
]  
  
**for**row **in**data**:**ws.append(row)  
  
projected\_pie **=**ProjectedPieChart()  
projected\_pie.type **=**"pie"  
projected\_pie.splitType **=**"val" *# split by value*labels **=**Reference(ws, min\_col**=**1, min\_row**=**2, max\_row**=**5)  
data **=**Reference(ws, min\_col**=**2, min\_row**=**1, max\_row**=**5)  
projected\_pie.add\_data(data, titles\_from\_data**=**True)  
projected\_pie.set\_categories(labels)  
  
ws.add\_chart(projected\_pie, "A10")  
  
**from**copy **import**deepcopy  
projected\_bar **=**deepcopy(projected\_pie)  
projected\_bar.type **=**"bar"  
projected\_bar.splitType **=**'pos' *# split by position*ws.add\_chart(projected\_bar, "A27")  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**14、 设定一个表格区域，并设定表格的格式**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**load\_workbook  
**from**openpyxl **import**Workbook  
**from**openpyxl.worksheet.table **import**Table, TableStyleInfo  
  
wb **=**Workbook()  
ws **=**wb.active  
  
data **=**[  
    ['Apples', 10000, 5000, 8000, 6000],  
    ['Pears',   2000, 3000, 4000, 5000],  
    ['Bananas', 6000, 6000, 6500, 6000],  
    ['Oranges',  500,  300,  200,  700],  
]  
  
*# add column headings. NB. these must be strings*ws.append(["Fruit", "2011", "2012", "2013", "2014"])  
**for**row **in**data**:**ws.append(row)  
  
tab **=**Table(displayName**=**"Table1", ref**=**"A1:E5")  
  
*# Add a default style with striped rows and banded columns*style **=**TableStyleInfo(name**=**"TableStyleMedium9", showFirstColumn**=**True,  
                       showLastColumn**=**True, showRowStripes**=**True, showColumnStripes**=**True)

#第一列是否和样式第一行颜色一行，第二列是否···

#是否隔行换色，是否隔列换色  
tab.tableStyleInfo **=**style  
ws.add\_table(tab)  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**15、给单元格设定字体颜色**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl.styles **import**colors  
**from**openpyxl.styles **import**Font  
  
  
wb **=**Workbook()  
ws **=**wb.active  
  
a1 **=**ws['A1']  
d4 **=**ws['D4']  
ft **=**Font(color**=**colors.RED)  # color**=**"FFBB00"，颜色编码也可以设定颜色  
a1.font **=**ft  
d4.font **=**ft  
  
*# If you want to change the color of a Font, you need to reassign it::  
#italic 倾斜字体*a1.font **=**Font(color**=**colors.RED, italic**=**True) *# the change only affects A1*a1.value **=**"abc"  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**16、设定字体和大小**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl.styles **import**colors  
**from**openpyxl.styles **import**Font  
  
  
wb **=**Workbook()  
ws **=**wb.active  
  
a1 **=**ws['A1']  
d4 **=**ws['D4']  
a1.value **=**"abc"  
  
**from**openpyxl.styles **import**Font  
**from**copy **import**copy  
  
ft1 **=**Font(name**=u'宋体'**, size**=**14)  
ft2 **=**copy(ft1)   *#复制字体对象*ft2.name **=**"Tahoma"  
  
**print**ft1.name  
  
**print**ft2.name  
  
**print**ft2.size *# copied from the*a1.font **=**ft1  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**17、设定行和列的字体**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl.styles **import**Font  
wb **=**Workbook()  
ws **=**wb.active  
  
col **=**ws.column\_dimensions['A']  
col.font **=**Font(bold**=**True)   *#将A列设定为粗体*row **=**ws.row\_dimensions[1]  
row.font **=**Font(underline**=**"single")  *#将第一行设定为下划线格式  
  
  
# Save the file*wb.save("e:**\\**sample.xlsx")

**18、设定单元格的边框、字体、颜色、大小和边框背景色**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl.styles **import**Font  
**from**openpyxl.styles **import**NamedStyle, Font, Border, Side,PatternFill  
  
wb **=**Workbook()  
ws **=**wb.active  
  
highlight **=**NamedStyle(name**=**"highlight")  
highlight.font **=**Font(bold**=**True, size**=**20,color**=**"ff0100")  
highlight.fill **=**PatternFill("solid", fgColor**=**"DDDDDD")#背景填充  
bd **=**Side(style**=**'thick', color**=**"000000")  
highlight.border **=**Border(left**=**bd, top**=**bd, right**=**bd, bottom**=**bd)  
  
**print**dir(ws["A1"])  
ws["A1"].style **=**highlight  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")

**19、常用的样式和属性设置**

*# -\*- coding: utf-8 -\*-***from**openpyxl **import**Workbook  
**from**openpyxl.styles **import**Font  
**from**openpyxl.styles **import**NamedStyle, Font, Border, Side,PatternFill  
**from**openpyxl.styles **import**PatternFill, Border, Side, Alignment, Protection, Font  
  
wb **=**Workbook()  
ws **=**wb.active  
  
ft **=**Font(name**=u'微软雅黑'**,  
    size**=**11,  
    bold**=**False,  
    italic**=**False,  
    vertAlign**=**None,  
    underline**=**'none',  
    strike**=**False,  
    color**=**'FF000000')  
  
fill **=**PatternFill(fill\_type**=**"solid",  
    start\_color**=**'FFEEFFFF',  
    end\_color**=**'FF001100')  
  
*#边框可以选择的值为：'hair', 'medium', 'dashDot', 'dotted', 'mediumDashDot', 'dashed', 'mediumDashed', 'mediumDashDotDot', 'dashDotDot', 'slantDashDot', 'double', 'thick', 'thin']  
#diagonal 表示对角线*bd **=**Border(left**=**Side(border\_style**=**"thin",  
              color**=**'FF001000'),  
    right**=**Side(border\_style**=**"thin",  
               color**=**'FF110000'),  
    top**=**Side(border\_style**=**"thin",  
             color**=**'FF110000'),  
    bottom**=**Side(border\_style**=**"thin",  
                color**=**'FF110000'),  
    diagonal**=**Side(border\_style**=**None,  
                  color**=**'FF000000'),  
    diagonal\_direction**=**0,  
    outline**=**Side(border\_style**=**None,  
                 color**=**'FF000000'),  
    vertical**=**Side(border\_style**=**None,  
                  color**=**'FF000000'),  
    horizontal**=**Side(border\_style**=**None,  
                   color**=**'FF110000')  
                )  
  
alignment**=**Alignment(horizontal**=**'general',  
        vertical**=**'bottom',  
        text\_rotation**=**0,  
        wrap\_text**=**False,  
        shrink\_to\_fit**=**False,  
        indent**=**0)  
  
number\_format **=**'General'  
  
protection **=**Protection(locked**=**True,  
            hidden**=**False)  
  
ws["B5"].font **=**ft  
ws["B5"].fill **=**fill  
ws["B5"].border **=**bd  
ws["B5"].alignment **=**alignment  
ws["B5"].number\_format **=**number\_format  
  
ws["B5"].value **=**"zeke"  
  
*# Save the file*wb.save("e:**\\**sample.xlsx")