**一：workbook操作**

1. 新建一个xlsx文件（同时系统会默认新建一个名为’Sheet’的表）

wb = openpyxl.Workbook() # 需要写入大量数据时可加参数write\_only=True

1. 加载一个已有的xlsx文件

wb = openpyxl.load\_workbook(filename, read\_only=False) # 只读模式有利于加# 载超大文件

**二：sheet操作**

1. 新建一个sheet

ws = wb.create\_sheet(‘sheetName’)

1. 修改sheet名字

ws.title = ‘sheetNameModified’

1. 修改sheet样式（这里修改sheet标签颜色为例）

ws.sheet\_properties.tabColor = ‘RRGGBB’

1. 查看workbook中所有的sheet

Sheetnames\_all = wb.sheetnames

1. 选取某一个sheet

sheet\_selected = wb[‘sheetname’]

sheet0 = wb.active # 默认情况选中第一个sheet

**三：cell（单元格）操作**

1. 获取某个单元格数据(Cell.value)

ws[‘a1’].value

ws.cell(1, 1).value # ws[‘a1’]和ws.cell(1, 1)都是快速得到Cell对象的方法

1. 设置某个单元格的数据(Cell.value = xxx)

ws[‘a1’] = xxx # 这里可直接对该对象赋值

ws.cell(1, 1).value = xxx

1. 获取某一矩形区域的Cell对象

cell\_area = ws[‘a1’:’c4’] # 这里返回的是一个 ((Cell, Cell, … ),

# (Cell, Cell, …),(Cell, Cell, … )….) tuple对象

1. 获取sheet中所有单元格

cell\_all = ws.rows # 这两者都可获取所有单元格，

cell\_all = ws.colums # 但返回的tuple组织顺序不同

1. 合并或拆分单元格

ws.merge\_cells(‘a1:b2’) # 注意这里的写法和获取一块Cell对象时是不同的！

ws.unmerge\_cells(‘a2:b2’)

四：保存文件（或另存为）

wb.save(filename)

五：单元格样式控制

https://blog.csdn.net/aishenghuomeidaoli/article/details/52165305