

Python与Excel

create by Dcount

- Python与Excel
 - create by Dcount
- 一、简介
 - 1. 使用模块: openpyxl
 - 2. 功能
- 二、打开和读取表格
 - 1. 打开并获取表格名称
 - 2. 通过sheet名称获取表格内容
 - 3. 获取表格内某个格子的数据
- 三、修改表格内容
 - 1. 向某个格子写入内容并保存
 - 2. 列表数据插入一行
 - 3. 插入/删除列行
 - 4. 插入公式
 - 5. 移动格子
 - 6. 创建/删除新的sheet
 - 7. 创建新的表格文件
 - 8. 冻结窗格
 - 9. 筛选
- 四、修改样式
 - 1. 修改/获取字体样式
 - 2. 对齐样式
 - 3. 边框
 - 4. 填充样式
 - 5. 设置行高和列宽
 - 6. 合并单元格

一、简介

1. 使用模块: openpyxl

直接在anaconda里安装即可

2. 功能

- 要在一堆部门预算表格里找到合计那个格子并统计
- 找到所有加粗字体的表格并将格子背景色标红?
- 将某些数据从一堆表格复制到另一堆表格中?
- 每天将同事发来邮件中的数据整理为表格?

- 每天监测几个网站的数据并整理到表格中进行统计和记录?
- ...

二、打开和读取表格

column:列, 字母表示 row:行, 数字表示, Excel从1开始 cell:单元格 sheet: 表, 一个xlr可能有多个sheet

1. 打开并获取表格名称

`load_workbook(filename)`, 参数为文件名 注意只能打开存在的表格, 不能用来创建新表格!

方法:

`workbook.sheetnames`:获取表格文件内sheet的名称, 注意会得到所有的sheet,就算其他的sheet是空的也会返回名字。返回的类型为字符串列表。

2. 通过sheet名称获取表格内容

```
import openpyxl as x
workbook = x.load_workbook(filename='d.xlsx')
sheet = workbook[workbook.sheetnames[0]]
```

- 获取表格尺寸 `sheet.dimensions`

3. 获取表格内某个格子的数据

三步

1. 选中表

`sheet=work.active` 如果可用表只有一张 `sheet=workbook.sheetnames[0]` 选中第几张表

2. 获取格子 `cell = sheet['A1']` 在这一步可以获取行数和列数

`cell.row`: 行数 `cell.column`:列数 `cell.coordinate`: 坐标

3. 获取值: `cell.value`

4. 获取一系列格子

`sheet.iter_rows(min_row=最低行数,max_row=最高行数,min_col=最低列数,max_col=最高列数)`

5. 其他选取格子方法

`cell=sheet.cell(row,column)`:用纯数字表示, 不用A1, B5这类表示 片
选: "A1:A5", "A", "1"

三、修改表格内容

1. 向某个格子写入内容并保存

1. 导入文件，选中表格，上面讲过了
2. 选中格子 `sheet['A1']='你好'`
3. 保存文件 `workbook.save(filename='d.xlsx')`
4. 或者可用

```
cell = sheet['A1']
cell.value = '你好'
workbook.save(filename='.xlsx')
```

2. 列表数据插入一行

`sheet.append(一维列表)`，会接在表格内已有的数据后面

```
data = [['a',1],['b',2],['c',3]]
for row in data:
    sheet.append(row)
workbook.save(filename)
```

3. 插入/删除列行

`sheet.insert_cols(idx,amount)`在idx列左边插入amount列 `sheet.insert_rows(idx,amount)`在行上边插入amount行 删除行insert换成delete

4. 插入公式

直接写Excel的公式就行，虽然不好用。例： `sheet['F1']='=SUM(F2:F9)'`

5. 移动格子

`sheet.move_range('C1:D4',rows=2,cols=-2)`正表示向下或向右，负表示向左或向上

6. 创建/删除新的sheet

创建 `workbook.create_sheet(sheetname)` 选中sheet后删除 `workbook.remove(sheet)` 复制一个sheet `workbook.copy_worksheet(sheet)`,复制后名字变成原本的+Copy，修改sheet名称 `sheet.title='表格1'`

7. 创建新的表格文件

```
import openpyxl as x
workbook = x.Workbook()
```

```
sheet = workbook.active
sheet.title = "表格1"
workbook.save(filename = "这是一个新表格.xlsx")
```

8. 冻结窗格

冻结B2 `sheet.freeze_panes = "B2"`

9. 筛选

激活筛选 `sheet.auto_filter.ref = sheet.dimensions`, 对多大的范围去应用筛选

四、修改样式

1. 修改/获取字体样式

`bold`:加粗 `italic`:斜体

```
cell = sheet['A1']
font = x.styles.Font(name='楷体', size=14, bold = True,
                    italic=True, color='ff0000')
cell.font = font
workbook.save(filename)
```

- 获取: `font = cell.font`
- 字体: `font.name`
- 字号: `font.size`
- 是否加粗: `font.bold`
- 是否倾斜: `font.italic`

2. 对齐样式

`x.styles.Alignment(horizontal,vertical,text_rotation,wrap_text)`

- `horizontal`:水平对齐模式
- `vertical`:垂直对齐模式
- `text_rotation`:旋转角度
- `wrap_text`: 是否自动换行

```
import openpyxl as x
workbook = load_workbook(filename)
sheet = workbook.active
cell = sheet['A1']
alignment = x.styles.Alignment(horizontal='center',
```

```
vertical='center',  
text_rotation=45)  
  
cell.alignment = alignment  
workbook.save(filename)
```

3. 边框

四条边单独设定

1. `x.styles.Side(style=边线样式, color=边线颜色)`

style可选thin,thick等等, 都是excel里的英文

2. `x.styles.Border(left=左边线样式, right=右边线样式, top=上边线样式, bottom=下边线样式)`

样式由side控制

4. 填充样式

1. 简单填颜色 `fill=x.styles.PatternFill(fill_type, fgColor)`

fill_type:填充样式, fgColor:填充颜色

2. 渐变色 `fill=x.styles.GradientFill(stop=(1,2,...))`

stop=(渐变颜色1, 渐变颜色2, ...) 色阶!

3. 赋值 `cell.fill=fill`

5. 设置行高和列宽

行高: `sheet.row_dimensions[index].height=5` 列宽: `sheet.column_dimensions[index].width=5`

6. 合并单元格

`sheet.merge_cells('C1:D2')` 或者

`sheet.merge_cells(start_row=7,start_column=1,end_row=8,end_column=4)` 取消合并

`sheet.unmerge_cells('C1:D2')` 或者

`sheet.unmerge_cells(start_row=7,start_column=1,end_row=8,end_column=4)`