

# 用Python使Excel表格合并

## 一、应用场景

在日常的工作中，会经常使用Excel表格进行内容的合并，像报销单的汇总，对于一些财务人员来说，每到月底是一个很让人头疼的事情，因为一大堆表格需要汇总，费时费力，使用python对文件进行读取和写入，快速合并我们需要汇总的表格。

## 二、项目代码

```
1 import os, openpyxl      # os模块是内置模块；openpyxl是第三方模块，需安装
2
3 # 定义walkFile()函数获取文件夹的所有文件
4 def walkFile(path):
5     for root_path, dirs_list, file_list in os.walk(path):
6         # root_path是当前目录，dirs_list是子目录，file_list是子文件
7         Read_Write_File(root_path, file_list)
8
9 # 定义Read_Write_File()函数用于读写表格
10 def Read_Write_File(root_path, file_list):
11     # 新建表格
12     wb_w = openpyxl.Workbook()
13     sheet_w = wb_w.active
14     sheet_w.title = '汇总数据' # 自定义一个工作表
15     # 遍历文件夹中的文件
16     for file in file_list:
17         file_path = os.path.join(root_path, file)
18         print(file_path)      # 打印文件中所有文件名
19
20     if '.xlsx' in file: # 判断文件的扩展名是不是xlsx
21         wb_r = openpyxl.load_workbook(file_path)
22         sheet_name = wb_r.sheetnames[0] # 获取第0个工作表的名称
23         sheet_r = wb_r[sheet_name]      # 获取工作表
24
25         # 写入汇总表格并保存
26         for row in sheet_r.values:
27             sheet_w.append(row)          # 写入新表格
28         wb_w.save('New_Excel.xlsx')      # 保存文件
```

```
29
30 # 定义main()函数，获取路径并执行合并
31 def main():
32     path_pwd = os.getcwd() # 获取当前工作路径
33     os.chdir(path_pwd)     # 切换到当前目录
34     path = path_pwd         # 使用相对路径获取 测试表2文件夹
35     walkFile(path)
36     print('合并完毕! ')
37
38 if __name__ == "__main__":
39     main() # 调用main()函数执行代码
```

注意：在main()函数中，给path 赋值自己的文件所在文件夹的路径。

## 三、项目操作及解析

### 项目准备条件：

- 安装：安装openpyxl模块，windows命令：pip install openpyxl；Mac系统命令：pip3 install openpyxl；
- 路径：要把Excel文件汇总到一个文件夹，然后在文件夹下新建一个Python文件，上述操作完成之后，在vs code中打开文件夹，复制代码到新建的python文件中，并运行；
- 格式：Excel文件里的格式保持一致，如：表头保持一致，工作表都在第0个位置（即首个）。

### 代码思路：

- 打开python文件，使用os模块获取当前路径和该文件夹下所有文件的列表，
- 然后筛选出Excel文件依次遍历读取，写入新的Excel文件中并保存。

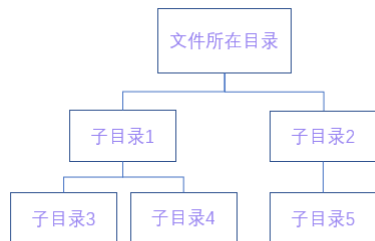
### 3.1 walkFile()函数

```
1 def walkFile(path):
2     for root_path, dirs_list, file_list in os.walk(path):
3         Read_Write_File(root_path, file_list)
```

**代码解析：**walkFile()函数，主要是通过os模块的walk()方法获取文件所在目录及该目录下的子目录和子文件，使用for循环时，会依次返回这三者，所以需要在for后面添加三个变量分别对应返回的这三个值。

当文件所在的目录下没有子目录时，for循环只循环一次。

当文件所在的目录下有子目录时，运行路径如下。



遍历路径：文件所在目录>子目录1>子目录3>子目录4>子目录2>子目录5

遍历路径是先从一个目录深入遍历完，再遍历另外一个目录。

## 3.2 Read\_Write\_File()函数

```
1 def Read_Write_File(root_path, file_list):
2     # 新建表格
3     wb_w = openpyxl.Workbook()
4     sheet_w = wb_w.active
5     sheet_w.title = '汇总数据' # 汇总表格的工作表可以自定义
6     # 遍历文件夹中的文件
7     for file in file_list:
8         file_path = os.path.join(root_path, file)
9         print(file_path) # 打印文件中所有文件名
10        if '.xlsx' in file: # 判断文件的扩展名是不是xlsx
11            wb_r = openpyxl.load_workbook(file_path)
12            sheet_name = wb_r.sheetnames[0] # 获取第0个工作表的名称，可根据工作表位置更改
13            sheet_r = wb_r[sheet_name] # 获取工作表
14
15            # 写入汇总表格并保存
16            for row in sheet_r.values:
17                sheet_w.append(row)
18            wb_w.save('New_Excel.xlsx') # 保存文件，文件名可以自定义
```

代码解析：

1. 在第12行是获取第0个工作表的名称，如果每一个要合并的文件的工作表名称都一样，也可以直接使用文件名，即sheet\_name = '[工作表名称]'，[工作表名称]为实际工作表名称。
2. 第18行是保存在和打开文件的同一个文件夹中，也可以自定义路径。

### 3.3 main()函数

```
1 def main():
2     path_pwd = os.getcwd() # 获取当前工作路径, 该路径也是Excel文件的路径
3     os.chdir(path_pwd)    # 切换到当前目录
4     walkFile(path_pwd)    # 调用walkFile()函数
5     print('合并完毕!')
```

#### 代码解析:

1. 使用os.getcwd()函数可以获取到当前运行的python文件的文件路径, 并传给walkFile()函数, 获取相关文件的内容。
2. os.chdir()是把运行的目录切换到当前运行的python文件的文件路径, 这样可以保证代码中使用的相对路径都是相对该路径去查找, 比如: wb\_w.save('New\_Excel.xlsx')使用了相对路径, 便会保存到目前python文件所在的目录里。