

高效办公第2关笔记



一、安装openpyxl库

openpyxl库是一个第三方库，由两位业内大佬开发的，所以在使用之前，得先下载openpyxl库，下载指令👉

Windows系统：

```
1 pip install openpyxl
```

MacOS系统：

```
1 pip3 install openpyxl
```

二、新模块导入语句

定义：from...import...是import语句的一个变体，可以导入库或模块中的函数、方法、类或变量。

import与from...import...差别在哪呢？

举个例子👉：

```
import time
time.sleep(1)
```

解释：导入time库，并调用.sleep()方法

```
from time import sleep
sleep(1)
```

解释：导入time库中的sleep方法，并使用.sleep()方法

两者的差别在于，使用from...import... 在代码的构写上会更简洁，方便。

三、openpyxl库

工作簿对象

获取工作簿对象的2种方式

- 方式1：打开【已有的】工作簿对象。语法如下👉

```
1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 工作簿对象 = load_workbook('填入要打开的excel表格路径')
```

你可以通过load_workbook()函数，获取到已有的工作簿,并把它创建为一个工作簿对象。

- 方式2：新建工作簿对象。语法如下👉

```
1 # 导入openpyxl库中的Workbook类
2 from openpyxl import Workbook
3
4 工作簿对象 = Workbook()
```

你可以通过Workbook()，新建一个工作簿对象，并把它赋给一个变量。

工作簿对象操作

- 操作1：保存修改好的工作簿，举个例子👉

```
1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
```

```
7 # 保存工作簿，filename指的是你要保存的路径。
8 工作簿对象.save(filename)
```

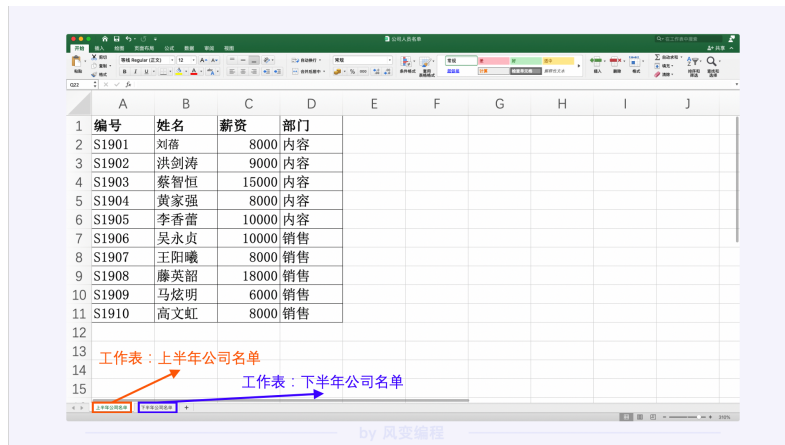
工作表对象

获取工作表对象的2种方式

- 方式1：获取默认（第一个）的工作表。语法如下👉

```
1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取工作表对象
8 工作表对象 = 工作簿对象.active
```

确认工作表有很多种写法，.active 会默认获取工作簿里的活动工作表（也就是第一张表）

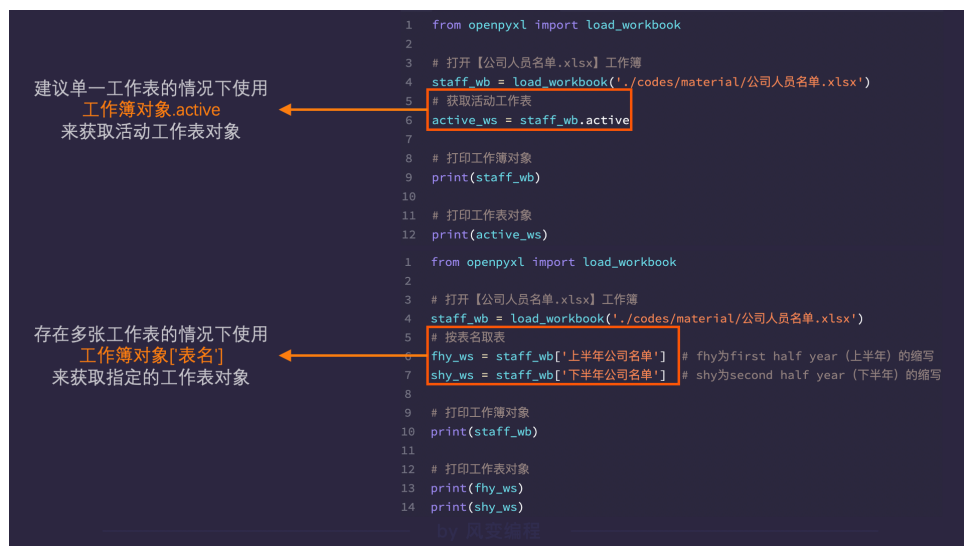


- 方式2：获取任意的工作表。语法如下👉

```
1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取任意一个工作表
8 工作表对象 = 工作簿对象['表名']
```

第2种方式，就比较简单粗暴一些，直接通过获取到的工作簿对象['具体的表名']也可获取到相应的工作表对象。

什么时候用方式1、方式2？



工作表对象的操作

- 操作1：获取单行或单列

在Excel表格中，使用数字表示行数，用英文字母表示列名。语法如下👉

```

1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取工作表对象
8 工作表对象 = 工作簿对象.active
9
10 # 获取相对应行中的内容
11 行的数据=工作表对象[行数]
12
13 # 获取相对应列中的内容
14 列的数据==工作表对象['列名']

```

使用英文字母表示列名

使用数字表示行数

	A	B	C	D	E	F	G	H
1	编号	姓名	薪资	部门				
2	S1901	刘蓓	8000	内容				
3	S1902	洪剑涛	9000	内容				
4	S1903	蔡智恒	15000	内容				
5	S1904	黄家强	8000	内容				
6	S1905	李香蕾	10000	内容				
7	S1906	吴永贞	10000	销售				
8	S1907	王阳曦	8000	销售				
9	S1908	藤英韶	18000	销售				
10	S1909	马炫明	6000	销售				
11	S1910	高文虹	8000	销售				
12								

by 风变编程

以上图举个例子👉

```

1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取工作表对象
8 工作表对象 = 工作簿对象.active
9
10 # 获取工作1行的内容
11 num = 工作表对象[1]
12
13 # 打印num的值
14 print(num)
15
16 # 结果：
17 # (<Cell '下半年公司名单'.A1>, <Cell '下半年公司名单'.B1>, <Cell '下半年公司名单'.C1>, <Cell '下半年公司名单'.D1>)

```

```

1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取工作表对象
8 工作表对象 = 工作簿对象.active
9
10 # 获取工作A列的内容
11 num = 工作表对象['A']
12
13 # 打印num的值
14 print(num)
15
16 # 结果
17 # (<Cell '下半年公司名单'.A1>, <Cell '下半年公司名单'.A2>, <Cell '下半年公司名单'.A3>, <Cell '下半年公司名单'.A4>, <Cell '下半年公司名单'.A5>, <Cell '下半年公司名单'.A6>, <Cell '下半年公司名单'.A7>, <Cell '下半年公司名单'.A8>, <Cell '下半年公司名单'.A9>, <Cell '下半年公司名单'.A10>)

```

为什么没有内容呢？别忘记了，通过这种方式会获取到一个元组，这个元组中包含了指定行或列中的所有数据，想要获取到内容还要加入一个for循环、.value

```

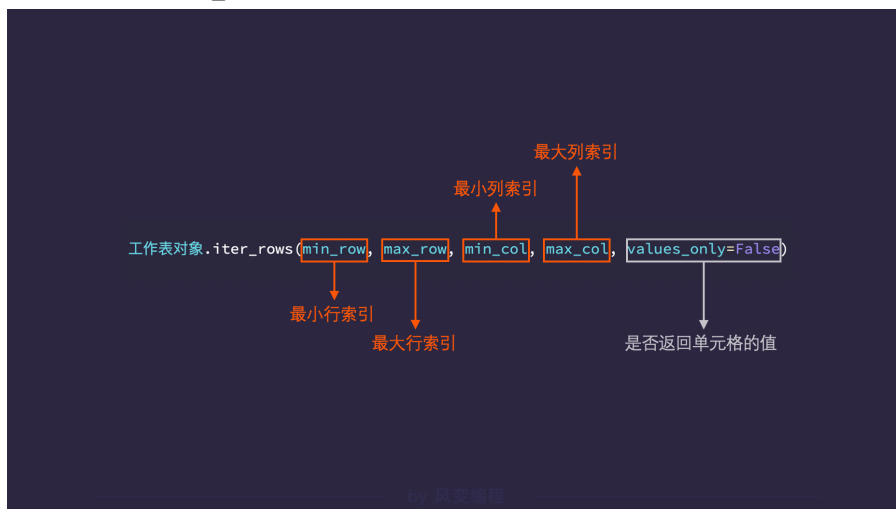
1 for i in 工作表对象[1]或者是工作表对象['A']:
2     print(i.value)

```

因为通过工作表对象[行数]、工作表对象['列名']，所取出来的值，是以元组的方式进行包装的，所以若是想要依次读取出来，就需要for循环，并搭配.value(单元格对象显示结果的关键词)。

- 操作2：获取获取多行数据

借助工作表对象的方法iter_rows()来得到表格中指定范围内的多行数据。语法如下📌



操作2的iter_rows方法，对于操作1来说更人性化，你可以通过内置的values_only参数设置，选择返回值是相对应的单元格内容，还是单元格对象。

- 操作3：添加数据

使用工作表对象的append()来添加一行数据，该方法可将部分可迭代对象（常见的如列表、元组）添加到工作表对象中，即给表格的末尾追加一行数据，与列表的append()相似。语法如下📌

```
1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取工作表对象
8 工作表对象 = 工作簿对象.active
9
10 工作表对象.append(列表、元组)
```

单元格对象

获取单元格对象的3种方式

- 方式1：通过for循环、iter_rows()方法

第一种方式你刚才学过，就是通过 `for row in 工作表对象.iter_rows()` 来获取指定范围的行，当参数 `values_only` 为默认的 `False` 时。**语法如下**👉

```
1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取工作表对象
8 工作表对象 = 工作簿对象.active
9
10 for row in 工作表对象.iter_rows(min_row="值", max_row="值", min_col="值",
11                                 max_col="值", values_only=False)
12     print(row)
```

输出的 `row` 值就是一个个由单元格对象组成的元组。

- **方式2：通过for循环、工作表对象[行数]/工作表对象['列名']方法**

通过工作表对象[行数]、工作表对象['列名']的方式能取出，单元格对象元祖的形式，而后通过for循环的方式，将元祖进行遍历，即可获取单个单元格对象。**语法如下**👉

```
1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取工作表对象
8 工作表对象 = 工作簿对象.active
9
10 for cell in 工作表对象[行数]:
11     print(cell)
12 for cell in 工作表对象['列名']:
13     print(cell)
```

若是想要得到具体的内容。我们还得要借助 `.value` 属性，与工作表操作1方法一致。

- **方式3：工作表对象['单元格坐标']，与方式2一致**

单元格对象的基本操作

- **操作1：取值**

借助单元格对象的属性value，我们就可以得到具体的数据，同时，我们也可以通过这个属性给单元格对象赋值（修改单元格的值或给单元格添加值）。语法如下👉

```
1 # 导入openpyxl库中的load_workbook函数
2 from openpyxl import load_workbook
3
4 # 获取工作簿对象
5 工作簿对象 = load_workbook()
6
7 # 获取工作表对象
8 工作表对象 = 工作簿对象.active
9
10 # 获取单元格对象
11 单元格对象 = 工作表对象['单元格坐标']
12
13 # 获取单元格的值
14 单元格对象.value
15
16 # 给单元格对象赋值
17 单元格对象.value = 值
```

四、总结

工作表对象与单元格对象的内容有很多是重合，但是不用担心会混乱，不妨这么联想一个**工作簿(excel表格)**由多个**工作表**汇总，一个**工作表**由多个**单元格**组成，通过这么联想你就能轻松的记住了openpyxl库的语法规则了，就好比一个洋葱，一层一层往下剥。

```
1 # 工作簿对象获取方式：
2 工作簿对象 = load_workbook('填入要打开的excel表格路径')
3 工作簿对象 = Workbook()
4
5 # 工作表对象获取方式：
6 工作表对象 = 工作簿对象.active
7 工作表对象 = 工作簿对象['表名']
8
9 # 单元格对象获取方式：
10 for row in 工作表对象.iter_rows():
11
12     for cell in 工作表对象[行数]:
13
14         for cell in 工作表对象['列名']:
15
16             工作表对象['单元格坐标']
```