

理论盘点：文件类型与对应的 Python 函数库

目录

- 1 不同类型文件的差异
- 2 Word 格式与 Python 函数库
- 3 Excel 格式与 Python 函数库

不同类型文件的差异

- 文件类型表示文件存储的格式，Windows 根据扩展名区分文件类型

- 📌 常见的音频格式：WAV、MP3

WAV 格式没有压缩，音频无损，但文件大；MP3 格式压缩但文件小

- 📌 常见的 Word 格式：docx、doc

- 📌 常见的 Excel 格式：xlsx、xls

Word 格式与 Python 函数库

- Python 操作 Word 格式的库是 `python-docx` 库
- 安装方法是 `pip install python-docx`
- 导入方法是 `import docx`

Excel 格式与 Python 函数库

- xlsx 格式一般采用 `openpyxl` 进行读写
- 如果是 xls 格式，对 Word 文件读取时采用 `xlrd` 库，写入则需要使用 `xlwt` 库

总结

- 1 不同文件格式代表着不同存储方式，对不同文件格式进行读写需采用不同的软件，相应的也需要采用不同的 Python 库来进行文件操作
- 2 xlsx 格式需要采用 openpyxl 进行文件读写，docx 格式需要采用 python-docx 进行文件读写

课后作业

请你编写程序，将 Excel 文件的第一张表格中的内容存储到 docx 文件中。

如何批量读取 Excel 文件？

目录

1 Excel 批量合并

2 Excel 批量拆分

Excel 批量合并

- 通过遍历取得将要合并的所有 Excel 文件:

```
files = [x for x in p.iterdir() if PurePath(x).match('*.xlsx')]
```

Excel 批量拆分

- 批量拆分的关键是遍历所有的单元格
- 和遍历元组一样，可以采用 `for...in` 的结构遍历单元格

总结

- 1 Excel 批量读取是实现批量合并和拆分的基础
- 2 Excel 批量合并和拆分都需要依赖于 for 循环实现

课后作业

请你编写程序，将 Excel 的每一行拆分成一个文件，并以每行第一个单元格作为文件名进行保存。

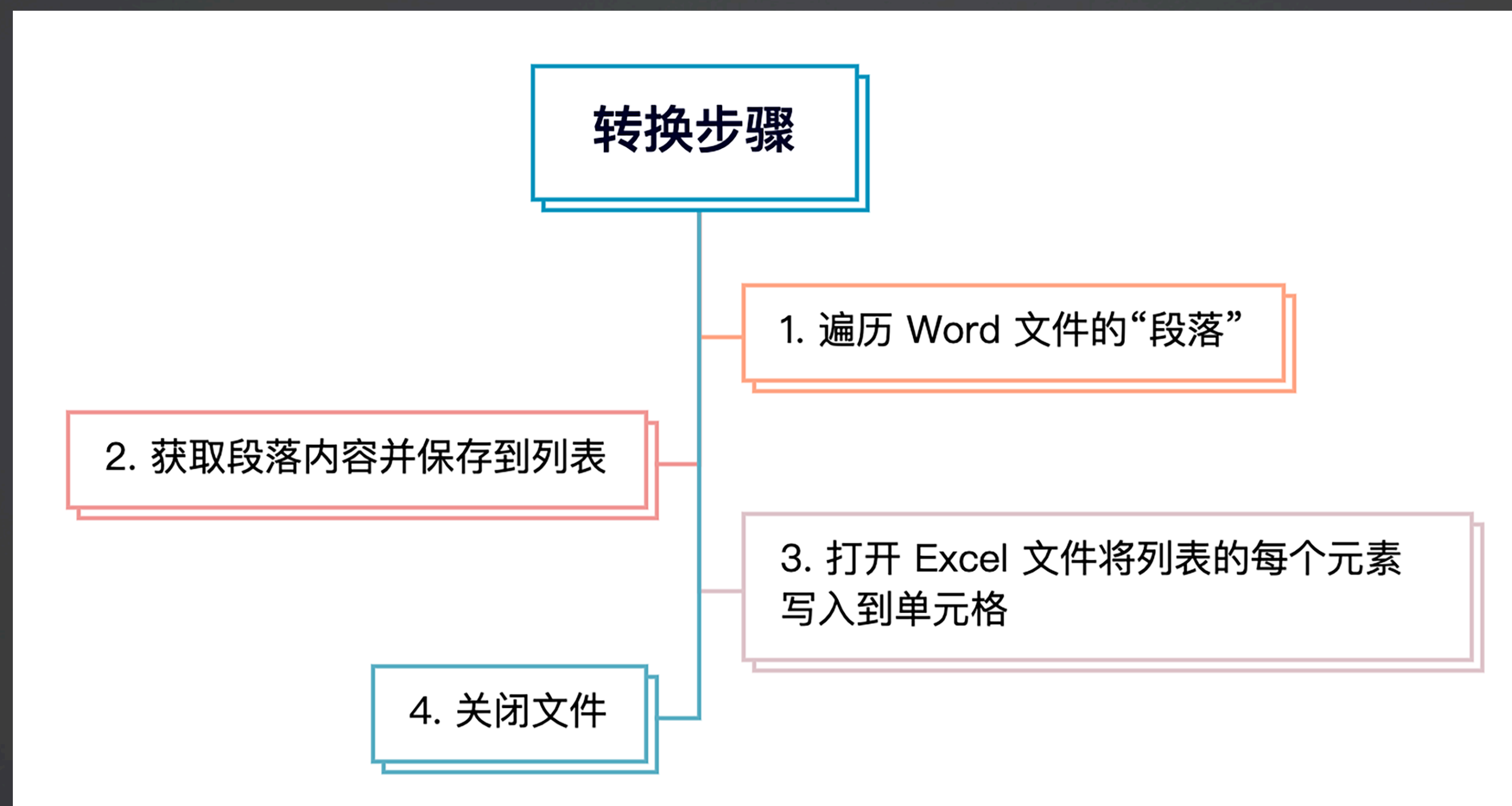
如何实现数据的批量格式转换？

目录

- 1 将 Word 文件转换为 Excel 文件
- 2 将 Excel 文件转换为 Word 文件

将 Word 文件转换为 Excel 文件

转换步骤



将 Excel 文件转换为 Word 文件

- Excel 文件有多张表，也比 Word 文件的宽度更宽，因此要提前确定转换前后的格式

总结

- 1 不同格式转换需要使用不同的库
- 2 格式转换一般需要打开多个文件，操作完成后需要将文件关闭，否则会导致内存占用过高

课后作业

请你编写程序，将指定 Word 文件中的文字内容，转换为同名的 Excel 文件。

如何扩展数据类型？

目录

1 命名元组

2 双端队列

3 计数器

4 字典和列表子类化

命名元组

- `namedtuple()` 是命名元组的工厂函数
- 命名元组使用前需要导入 `collections` 库
- `Point = namedtuple('Point', ['x', 'y'])`，定义一个“点”，包含了 `x` 和 `y` 两个坐标

双端队列

- `deque` 对象是实现双向队列的对象
- 双向队列能够支持从左右两端实现元素的添加和移除
- `deque` 比传统的列表多了 `appendleft()`、`popleft()` 方法

计数器

- 通过 Counter 对象，可实现计数器工具
- 计数器工具可以方便统计字典中元素的数量，也可以用来统计元素的出现次数

字典和列表子类化

- ◎ UserDict 类用于字典对象的二次开发
- ◎ UserList 类用于列表对象的二次开发
- * 当你需要字典、列表的功能，而他们又不能完全满足需要时，
可通过继承 UserDict 和 UserList 实现增强功能的字典和列表

总结

- 1 collections 模块提供了一系列的扩展数据类型
- 2 当传统的列表、字典无法满足需要时，可以利用计数器、双端队列、命名元组来丰富数据类型

课后作业

请你编写程序，统计一篇文章中出现频率在前五的单词，并将单词和出现次数一起输出到终端。

THANKS