Python 操作 word文档

python-docx官方文档

安装python-docx库

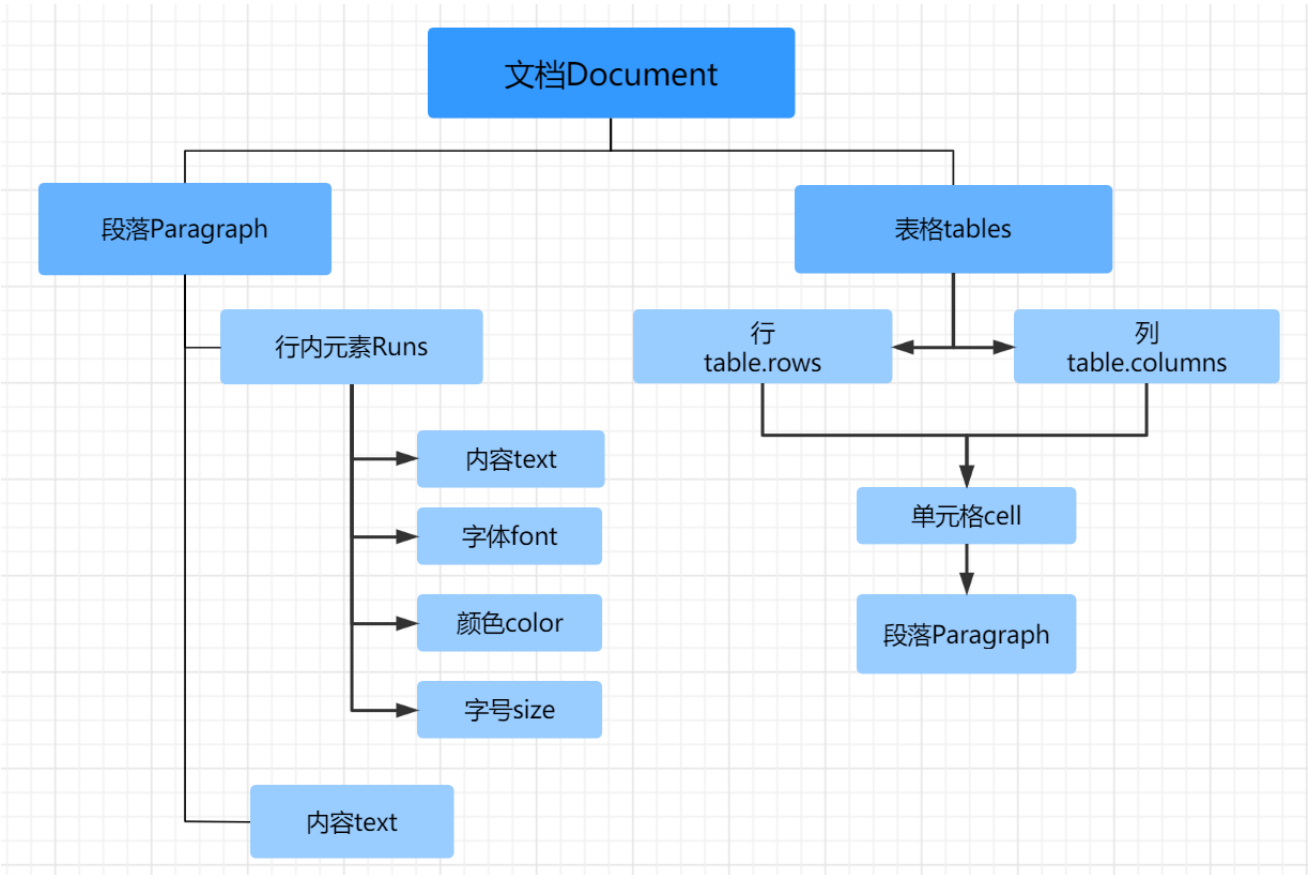
```
pip install python-docx
```

注意，docx库只能操作docx文件，不能操作doc文件，如果是doc文件，先修改成docx文件。

常用模块

```
from docx import Document
from docx.enum.text import WD_ALIGN_PARAGRAPH #设置对象居中、对齐等
from docx.enum.text import WD_TAB_ALIGNMENT,WD_TAB_LEADER #设置制表符等
from docx.shared import Pt #设置像素、缩进等
from docx.shared import RGBColor #设置字体颜色
```

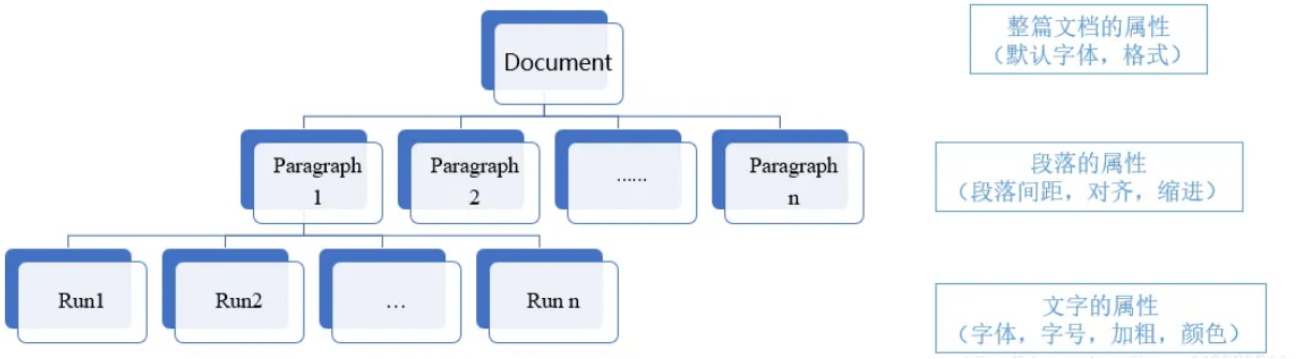
doc文件的完整结构



`Document`：是一个 Word 文档 对象，不同于 VBA 中 Worksheet 的概念，Document 是独立的，打开不同的 Word 文档，就会有不同的 Document 对象，相互之间没有影响

`Paragraph`：是段落，一个 Word 文档由多个段落组成，当在文档中输入一个回车键，就会成为新的段落，输入 shift + 回车，不会分段

`Run` 表示一个节段，每个段落由多个 节段组成，一个段落中具有相同样式的连续文本，组成一个节段，所以一个段落 对象有个 Run 列表



如下是一个word结构的划分:

python-docx库的使用

读取段落文字

一、文件对象.paragraphs 得到的是一个列表，包含了每个段落的实例，可以索引、切片、遍历

```
# 导入Document功能类
from docx import Document

# 加载docx文件内容 返回一个实例化文件对象 doc,
doc = Document('c:/练习.docx')

# 返回文档中每个段落集合，是一个列表，可以通过索引获取
# doc.paragraphs
print(doc.paragraphs)
print(doc.paragraphs[0])
print(doc.paragraphs[0:2])
```

二、段落.text 得到该段落的文字内容

```
doc = Document('./练习.docx')
paras = doc.paragraphs
for para in paras:
    print(para.text)
```

三、块与文字

parapraph.runs 得到一个列表，包含了每个文字块，可索引、切片、遍历

块.text 得到该文字块的文字内容

```
from docx import Document
doc = Document('./练习2.docx')
paras = doc.paragraphs
runs = paras[0].runs
for run in runs:
    print(run.text)
```

四、读取表格

```
doc = Document('./练习2.docx')
tables = doc.tables
print(tables)

table = tables[0]
# 按行读取
for row in table.rows:
    for cell in row.cells:
        print(cell.text, cell.paragraphs)

# 按列读取
for col in table.columns:
    for cell in col.cells:
        print(cell.text, cell.paragraphs)
```

练习：在练习2.docx中搜索“路飞学城”出现的次数

```
doc = Document('./练习2.docx')
paras = doc.paragraphs
total = 0
for para in paras:
    count = para.text.count('路飞学城')
    total += count

table = tables[0]
for col in table.columns:
    for cell in col.cells:

        count = cell.text.count('路飞学城')
```

```
        total += count

print(total)
```

按规则读取文字

通过para.style.name可以读取所有paragraph的风格名称，根据这个名称可以有辨识的读取相应内容

```
from docx import Document
# 1. 取Word所有内容并打印风格名称
doc = Document('./练习3.docx')
for para in doc.paragraphs:
    print("StyleName:",para.style.name)
    print(para.text)

# 2. 读取一级标题
doc = Document('./练习3.docx')
for para in doc.paragraphs:
    if para.style.name == 'Heading 1':
        print(para.text)

# 3.读取二级标题
doc = Document('./练习3.docx')
for para in doc.paragraphs:
    if para.style.name == 'Heading 2':
        print(para.text)

# 4.读取所有标题 【使用正则】
import re
doc = Document('./练习3.docx')
for para in doc.paragraphs:
    temp = re.match('Heading \d+', para.style.name)
    if temp:
        print(para.text)

# 5.读取正文
doc = Document('./练习3.docx')
for para in doc.paragraphs:
    if para.style.name == 'Normal':
        print(para.text)
```

写入文字

```
# Document不写路径，会创建一个空的Document对象，可以使用doc.save('路径')写入本地
doc = Document()

# 添加标题
doc.add_heading('一级标题',level=1)

# 添加正文
doc.add_paragraph('这里是正文',style='Normal')
```

```
# 添加分页（另起一个新页）
# doc.add_page_break()

# 添加文字块
para = doc.add_paragraph('我是一个正文，我后面的文字会被设置格式')
para.add_run('被设置加粗的文字块').bold = True
para.add_run('被设置普通的文字块')
para.add_run('被设置斜体的文字块').italic = True
from docx.shared import RGBColor
para.add_run('红色字体').font.color.rgb = RGBColor(255,0,0)
# 指定位置插入一个段落
doc = Document('dancer.docx')
para2 = doc.paragraphs[1]
# 在第二个段落处插入
para2.insert_paragraph_before('这是添加的新的第二个段落')

# 保存文件
doc.save('dancer.docx')
```

样式

样式可以针对整体文档（document）、段落（paragraph）、节段（run），越具体，样式优先级越高

python-docx 样式功能配置多样，功能丰富，可以通过如下方式获取

```
from docx.enum.style import WD_STYLE_TYPE

# WD_STYLE_TYPE 定义了4种风格分类
# WD_STYLE_TYPE.CHARACTER  性格风格（字体风格）
# WD_STYLE_TYPE.LIST 列表样式
# WD_STYLE_TYPE.PARAGRAPH 段落样式
# WD_STYLE_TYPE.TABLE 表格样式

doc = Document()

# 读取文档支持的所有风格，并查看所有风格所属分类
for style in doc.styles:
    if style.type == WD_STYLE_TYPE.CHARACTER:
        print('CHARACTER:',style.name)
    if style.type == WD_STYLE_TYPE.LIST:
        print('LIST:',style.name)
    if style.type == WD_STYLE_TYPE.PARAGRAPH:
        print('PARAGRAPH:',style.name)
    if style.type == WD_STYLE_TYPE.TABLE:
        print('TABLE:',style.name)
```

段落样式

段落样式包括：对齐、列表样式、行间距、缩进、背景色等，可以在添加段落时设定，也可以在添加之后设置：

```
# 添加一个段落，设置为无序列表样式
document.add_paragraph('我是个无序列表段落', style='List Bullet')

# 添加段落后，通过 style 属性设置样式
paragraph = document.add_paragraph('我也是个无序列表段落')
paragraph.style = 'List Bullet'
```

文字样式

在前面 python-docx 文档结构图可以看到，段落中，不同样式的内容，被划分成多个 节段（Run），文字样式是通过 节段（Run）来设置的

设置加粗/斜体/颜色

```
from docx import Document
from docx.shared import Pt, RGBColor # 字号，颜色

paragraph = document.add_paragraph('添加一个段落')
# 设置 节段文字为加粗
run = paragraph.add_run('添加一个节段')
run.bold = True
run.font.bold = True # 加粗
run.font.italic = True # 斜体
run.font.underline = True # 下划线
run.font.strike = True # 删除线
run.font.shadow = True # 阴影
run.font.size = Pt(24)
run.font.color.rgb = RGBColor(255,0,0) # 颜色
run.font.name = 'Arial' # 字体设置
```

写入表格数据

```
# eg1.添加一个 2×2 表格
doc = Document()
table = doc.add_table(rows=2, cols=2, style='Light Shading Accent 6')

cell = table.cell(1, 1)
cell.text = '我是单元格文字'

# 通过表格的行访问cell
row0 = table.rows[0]
row0.cells[0].text = 'Hello'
row0.cells[1].text = '路飞学城'

# 通过表格的列访问cell
col0 = table.columns[0]
col0.cells[1].text = '第一列第二行'

# 增加行
row = table.add_row()
```

```
doc.save('表格数据.docx')
```

```
# eg2. 批量添加数据
# 数据源
items = (
    (6, '1024', '爬虫'),
    (3, '2042', '数分'),
    (1, '1068', '全栈'),
)
document = Document()
# 添加一个表格
table = document.add_table(1, 3)

# 设置表格标题
heading_cells = table.rows[0].cells
heading_cells[0].text = '数量'
heading_cells[1].text = '编码'
heading_cells[2].text = '课程'

# 将数据填入表格
for item in items:
    cells = table.add_row().cells
    cells[0].text = str(item[0])
    cells[1].text = item[1]
    cells[2].text = item[2]

# 添加第二个表格
table2 = document.add_table(rows=2, cols=2, style='Light Shading Accent 3')
table2.rows[0].cells[0].text = '新增的表格'

document.save('表格数据.docx')
```