

高效办公第7关笔记



一、python-docx 库简介

1.概念

python-docx 库是专为 Python 设计的, 用于创建和处理 Microsoft Word (.docx) 文件的第三方库。

2.安装

windows安装:

```
1 pip install python-docx
```

mac安装:

```
1 pip3 install python-docx
```

3.导入

安装时输入 `python-docx`, 但写代码导入这个库时, 直接写 `docx` 即可:

```
1 import docx
2 from docx import ...
```

注意:

python-docx 库只能处理 docx 文件（Microsoft office 2007 之后的 office 版本生成的 Word 文档格式），无法处理 doc 文件。

python-docx 库在处理 Word 文件时，一般会将这个文件分为三个层级：

- 1) 最外层的 文档对象 (Document) ；
- 2) 中间的 块级元素 (Block-level) ；
- 3) 最内层的，包含在块级元素里面的 内联元素 (Inline-level) 。



二、添加文字和图片

1.生成 Document 对象

1.1 导入

一般情况下，我们会把类 Document 从 python-docx 库中直接导入：

```
1 from docx import Document
```

1.2 实例化

实例化 Document 类

```
变量名 = Document(docx)
```

by 风变编程

参数 docx 表示 Word 文件的路径。这里存在两种情况：

- 如果传入路径，会把路径代表的 Word 文件读取成一个 Document 对象。
- 如果不传入路径，就会生成一个空的 Document 对象，相当于新建了一个空白的 Word 文件。

2.添加文字和图片

2.1 块级元素

概念：块级元素（block-level）最大的特点是文本都会包含在它的左右边界之间。每当文本的长度超出右边界的范围时，就会额外增加一行。

在 Word 文件中，最常见的块级元素就是标题、表格和段落。

添加标题（Heading 对象）的方法：

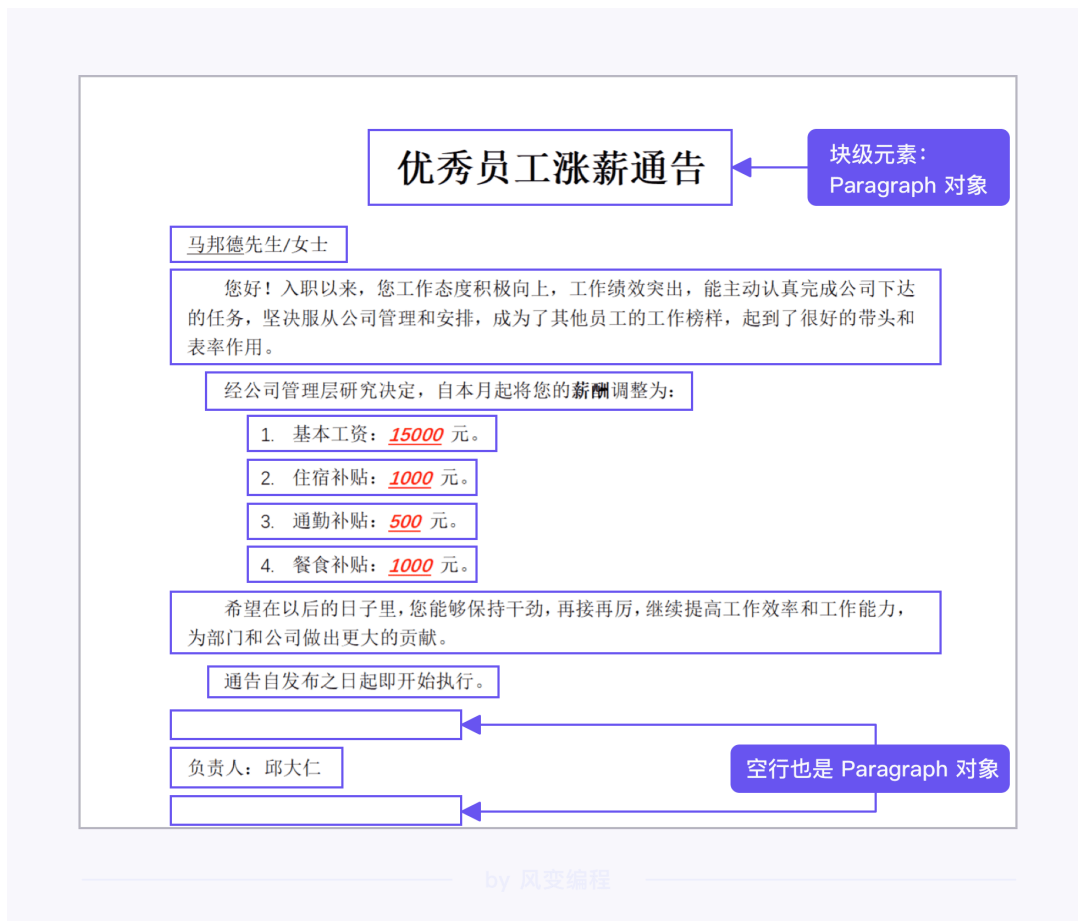
```
1 Docuement 对象.add_heading()
```

添加表格（Table 对象）的方法：

```
1 Docuement 对象.add_table()
```

2.2 Paragraph 对象

概念：Paragraph 对象是最频繁使用的块级元素，对应 Word 文件中的段落，在 Word 文件里，是将 Enter 回车键作为段落之间的分界点，每按一次回车，光标进入下一段，就意味着多了一个段落。每一个段落都对应着一个 Paragraph 对象。



查看 Paragraph 对象：

```
1 Document 对象.paragraphs
```

该方式会返回一个列表，列表中的每一个元素都是一个 Paragraph 对象。还可以通过 Paragraph 对象的属性text来查看段落中的文字。

添加 Paragraph 对象：

```
1 Document 对象.add_paragraph()
```

这个方法会在 Document 对象末尾添加一个 Paragraph 对象，即添加一个段落。空文档时，在一开始的空白处添加段落，但当段落中有内容时，在最后的内容后另起一行添加新的段落。

2.3 内联元素

概念：内联元素组成了块级元素，它是块级元素中的一部分。比如，它可以是一个粗体显示的单词或一个全大写的句子。最常见的内联元素是 **Run 对象**。

2.4 Run对象

概念：在 python-docx 中，Run 是最基本的单位，Run 对象内的文本样式都是一致的，也就是说，在读取 docx 文件生成文档对象时，不同样式的文本会变成一个个独立的 Run 对象。

在读取出的 Document 对象中，Run 对象是 Paragraph 对象中具有相似样式（如相似的字体大小，字体形状和字体样式）的文字，也可能是一张图片。

查看 Run 对象：

```
1 Paragraph 对象.runs
```

该方法会返回一个列表，列表中的每一个元素都是一个 Run 对象。与 Paragraph 对象类似，Run 对象也有 text 属性，所以可以使用该属性查看其中的文字。

添加 Run 对象：

```
1 Paragraph 对象.add_run()
```

总结：



2.5 添加文字

```
1 Run 对象.add_text(text)
```

语法


Run 对象.add_text(text)

参数 text 表示 Run 对象中需要添加的文字，一般为字符串类型

示例代码

```
from docx import Document
doc = Document()
para = doc.add_paragraph()
run = para.add_run()
run.add_text('使用 Run 对象.add_text(text) 添加文字内容')
doc.save('./添加文字.docx')
```

代码实现效果



by 风变编程

2.6 添加图片

1 Run 对象.add_picture(path)

语法

Run 对象.add_picture(path)

参数 path 表示需要添加图片的路径

示例代码

```
from docx import Document
doc = Document()
para = doc.add_paragraph()
run1 = para.add_run()
run1.add_text("使用 Run 对象.add_picture(text) 添加图片")
run1.add_picture('./代码/Shining.png')
doc.save('./添加图片.docx')
```

代码实现效果



by 风变编程

直接添加文字的方法：



Document 对象.add_paragraph() 和 Paragraph 对象.add_run() 这两个方法都可以传入参数 text, 直接在创建 Paragraph 对象和 Run 对象的时候就添加好文字, 可以节省很多时间。

三、添加带样式的文字

1.设置段落样式

使用 Paragraph 对象的属性 paragraph_format 来控制段落的样式, 即段落在页面的位置, 例如对齐方式、缩进和段落前后的间距。

如果要具体指定段落是左对齐、右对齐、居中对齐还是两端对齐, 就需要在一开始导入 WD_ALIGN_PARAGRAPH, 导入方式:

```
1 from docx.enum.text import WD_ALIGN_PARAGRAPH
```

四种对齐方式:

4 种对齐方式	
from docx.enum.text import WD_ALIGN_PARAGRAPH	
左对齐 (默认)	Paragraph 对象.paragraph_format.alignment = WD_ALIGN_PARAGRAPH.LEFT
右对齐	Paragraph 对象.paragraph_format.alignment = WD_ALIGN_PARAGRAPH.RIGHT
居中对齐	Paragraph 对象.paragraph_format.alignment = WD_ALIGN_PARAGRAPH.CENTER
两边对齐	Paragraph 对象.paragraph_format.alignment = WD_ALIGN_PARAGRAPH.JUSTIFY
by 风变编程	

2.设置字体样式

通常, 可以使用 Run 对象的属性 font 来设置字体的样式。该属性可以设置字体相关功能, 包括斜体、加粗、大小、颜色等。

2.1 设置字号

导入语法：

```
1 from docx.shared import Pt
```

Pt 是一种表示文字大小的绝对单位，但是熟悉 Word 的人都知道，中文文本习惯使用字号作为大小单位，比如五号字、一号字。他们之间的对应关系是：

字号对应关系	
中文字号	Pt 字号
初号字	42pt
小初号字	36pt
一号字	26pt
小一号字	24pt
二号字	22pt
小二号字	18pt
三号字	16pt
小三号字	15pt
四号字	14pt
小四号字	12pt
五号字	10.5pt
小五号字	9pt
六号字	7.5pt
小六号字	6.5pt

by 风变编程

然后再让属性 font 调用 size，根据上面的表格，将导入的 Pt 设置好后赋值给 size 就可以设置字体的大小了。

```
1 from docx.shared import Pt
2 # 设置字体大小
3 Run 对象.font.size = Pt(14) # 字体大小
```

2.3 设置加粗

可以通过属性 font 调用 bold，然后将值设置为 True 即可：

```
1 # 设置字体加粗
2 Run 对象.font.bold = True # 字体加粗
```


font 补充

from docx.shared import RGBColor, Pt

参数	说明	示例
指定字体	Run 对象.font.name	font.name = 'Calibri'
设置斜体	Run 对象.font.italic	font.italic = True
设置下划线	Run 对象.font.underline	font.underline = True
设置字体颜色	Run 对象.font.color.rgb	font.color.rgb = RGBColor(255, 0, 0)

by 风变编程