

办公自动化之-Word

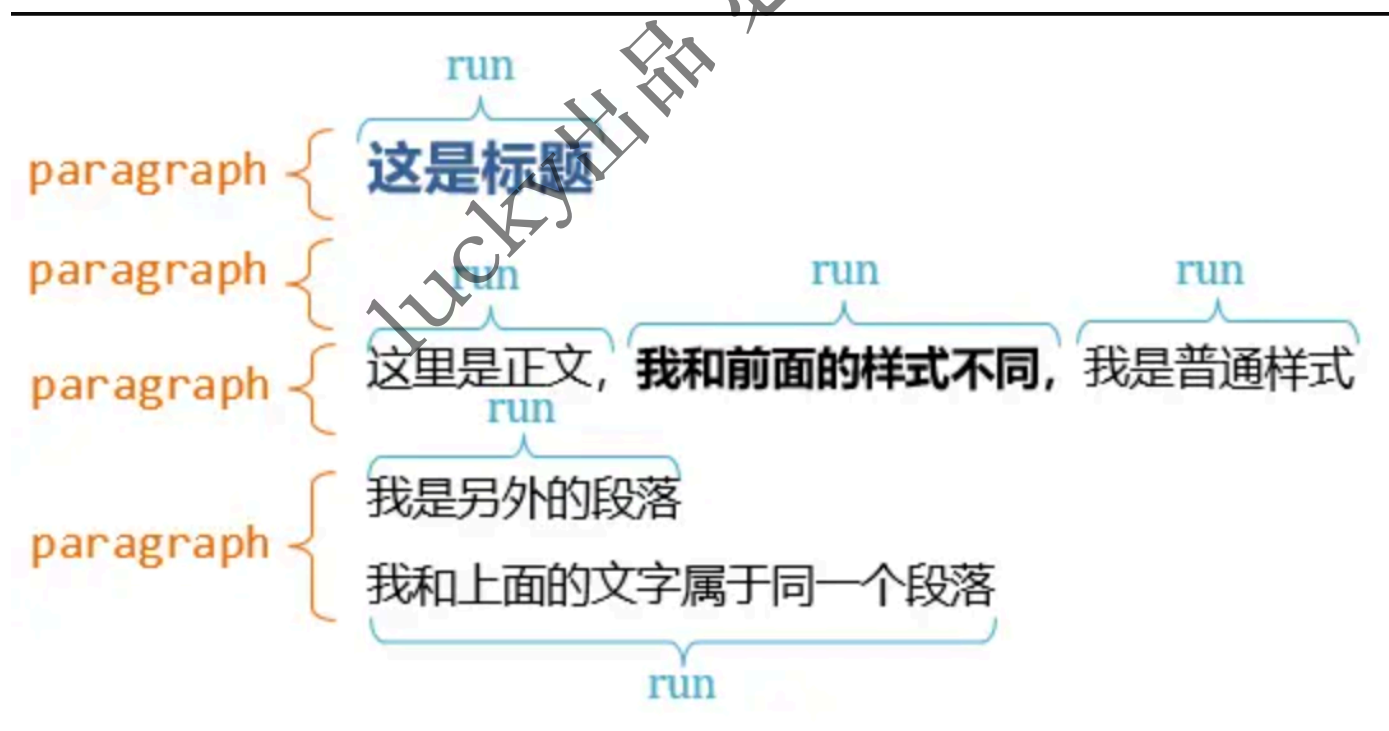
python-docx库

1、安装python-docx库

cmd窗口输入 `pip install python-docx` 按下回车等待安装完成。

说明

2、文档的结构说明



在word文档中，其主要结构如下所述：

- 每个document包含多个paragraph,每个paragraph有多个run,每个run包含有(text文本,font字体,color颜色,字号)
- 每个document包含多个tables,table中有多个rows,每个row包含多个cells,每个cell中包含多个paragraph。对于写word表格不论是 head 还是paragraph 基本操作都是先添加对象,然后再添加run就好了
- word表格的结构包含head标题、normal 正文、Caption表

3、基本操作语法

3.1 打开文档

```
1 document = Document()
```

不填默认为新建一个文档

3.3 加入不同等级的标题

```
1 document.add_heading('总标题',0)
2 document.add_heading('一级标题',1)
3 document.add_heading('二级标题',2)
```

3.3 添加文本

```
1 paragraph = document.add_paragraph('文本内容')
```

3.4 设置字号

```
1 run = paragraph.add_run('设置字号、')  
2 run.font.size = Pt(34)
```

3.5 设置中文字体

```
1 run = paragraph.add_run('设置中文字体、')  
2 run.font.name = '宋体'  
3 r = run._element  
4 r.rPr.rFonts.set(qn('w: eastAsia'), '宋体')
```

3.6 设置斜体

```
1 run = paragraph.add_run('斜体、')  
2 run.italic = True
```

3.7 设置粗体

```
1 paragraph.add_run('粗体').bold = True
```

3.8、首行缩进

缩进方式	属性
左边缩进	left_indent
右边缩进	right_indent
首行缩进	first_line_indent

示例

```
1 from docx import Document
2 from docx.shared import Inches
3 doc = Document()
4 article = doc.add_paragraph ()
5 a2 = article.add_run(
6     '因_____, 特向您请事假____天。请假时间自_____年
7     _____月____日至_____年____月____日。这段时间内原计划安
8     排的课程已做好处理, 希望领导批准。' )
9 # 首行缩进2个字符
10 article2_format = article.paragraph_format
11 article2_format.first_line_indent =
12     Inches(0.3)
13 doc.save('test.docx')
```

3.9 对齐方式

名称	属性
左对齐	LEFT
居中	CENTER
右对齐	RIGHT
文本两端对齐	JUSTIFY

实例

```
1 from docx import Document
2 from docx.enum.text import
  WD_PARAGRAPH_ALIGNMENT
3 doc = Document()
4
5 title = doc.add_paragraph()
6 a2 = title.add_run('标题内容')
7 title.alignment =
  WD_PARAGRAPH_ALIGNMENT.CENTER # 居中
8 doc.save('test.docx')
```

3.10 增加列表

```

1 document.add_paragraph(
2     '无序列表元素1', style='List Bullet'
3 )
4 document.add_paragraph(
5     '无序列表元素3', style='List Bullet'
6 )
7
8 Style='List Bullet' # 为无序
9
10 Style='List Number' # 为有序

```

3.11 添加图片

```

1 document.add_picture('1.png',
2     width=Inches(1.35))

```

png为图片名称（必须与代码文件在同一个文件夹内）

3.12 增加表格

```

1 table = document.add_table(rows=1, cols=3)
2 hdr_cells = table.rows[0].cells
3 hdr_cells[0].text = 'Name'
4 hdr_cells[1].text = 'Id'
5 hdr_cells[2].text = 'Desc'

```

3.13 再增加3行表格元素

```
1 for i in range(3):
2     row_cells = table.add_row().cells
3     row_cells[0].text = 'test'+str(i)
4     row_cells[1].text = str(i)
5     row_cells[2].text = 'desc'+str(i)
```

3.14 增加分页

```
1 document.add_page_break()
```

3.15 保存文件

```
1 document.save('测试.docx')
```

3.16 完整代码

```
1 from docx import Document
2 from docx.shared import Pt
3 from docx.shared import Inches
4 from docx.xml.ns import qn
5
6
7
8 document = Document()
9 document.add_heading(u'MS WORD写入测试',0)
```

```
10 document.add_heading('一级标题',1)
11 document.add_heading('二级标题',2)
12 paragraph = document.add_paragraph('我们在做文本测试! ')
13 run = paragraph.add_run('设置字号、')
14 run.font.size = Pt(34)
15 run = paragraph.add_run('设置中文字体、')
16 run.font.name = '宋体'
17 r = run._element
18 r.rPr.rFonts.set(qn('w: eastAsia'), '宋体')
19 run = paragraph.add_run('斜体、')
20 run.italic = True
21 paragraph.add_run('粗体').bold = True
22 document.add_paragraph('Intense quote',
    style='Intense Quote')
23 document.add_paragraph(
24     '无序列表元素1', style='List Bullet'
25 )
26 document.add_paragraph(
27     '无序列表元素3', style='List Bullet'
28 )
29 document.add_paragraph(
30     '有序列表元素1', style='List Number'
31 )
32 document.add_paragraph(
33     '有序列表元素3', style='List Number'
34 )
```



```
35 document.add_picture('dlrb.jpeg',  
    width=Inches(1.35))  
36 table = document.add_table(rows=1, cols=3)  
37 hdr_cells = table.rows[0].cells  
38 hdr_cells[0].text = 'Name'  
39 hdr_cells[1].text = 'Id'  
40 hdr_cells[2].text = 'Desc'  
41 for i in range(3):  
42     row_cells = table.add_row().cells  
43     row_cells[0].text = 'test'+str(i)  
44     row_cells[1].text = str(i)  
45     row_cells[2].text = 'desc'+str(i)  
46 document.add_page_break()  
47 document.save('测试.docx')
```

生成效果

MS WORD 写入测试

- 一级标题
- 二级标题

我们在做文本测试！设置字号、Set Font,设置中文字体、斜体、粗体

Intense quote

- 无序列表元素 1
 - 无序列表元素 2
- 有序列表元素 1
 - 有序列表元素 2



Name	Id	Desc
test0	0	desc0
test1	1	desc1
test2	2	desc2

□

3.17 获取文档操作

```
1 from docx import Document
2
3 docu = Document(r'test.docx')
4
5 paragraphs = docu.paragraphs
6 for paragraph in paragraphs:
7     print(paragraph.text)
```

3.18 获取段落数量

```
1 # 段落数量
2 print(len(doc.paragraphs))
```

4、案例

4.1 请假条

```
1 from docx.enum.text import
   WD_PARAGRAPH_ALIGNMENT
2 from docx import Document
3 from docx.shared import Pt, Inches
4 from docx.oxml.ns import qn
5
6 doc = Document()
7 # 全局字体设置
```

```
8 doc.styles['Normal'].font.name = u'宋体'
9 doc.styles['Normal']._element.rPr.rFonts.set(
    qn('w:eastAsia'), u'宋体')
10
11 # 1、标题
12 title = doc.add_paragraph ()
13 title1 = title.add_run ('公司员工请假条') # 内容
14 title1.font.size = Pt(20) # 设置字体大小
15 title1.bold = True # 加粗
16 title.alignment =
    WD_PARAGRAPH_ALIGNMENT.CENTER
17
18 # 2、正文
19 article1 = doc.add_paragraph ()
20 a1 = article1.add_run( '___部:')
21
22 article2 = doc.add_paragraph()
23 a2 = article2.add_run(
24 '因____，特向您请事假天。请假时间自_年月日
    至_年月日。这段时间内原计 划安排的课程已做好处理，希
    望领导批准。 '
25
26 )
27
28 # 首行缩进 负值表示悬挂缩进
29 article2_format = article2.paragraph_format
```

```
30 article2_format .first_line_indent =  
    Inches(0.3)  
31  
32 article3 = doc.add_paragraph ()  
33 a3 = article3.add_run('请假人:') # 内容  
34 article3.alignment =  
    WD_PARAGRAPH_ALIGNMENT.RIGHT  
35 article3_format = article3.paragraph_format  
36 article3_format.right_indent = Inches(0.9)  
37  
38 nowData = doc.add_paragraph ()  
39 n3 = nowData.add_run('年 月 日') # 内容  
40 nowData.alignment =  
    WD_PARAGRAPH_ALIGNMENT.RIGHT  
41 nowData_format = nowData.paragraph_format  
42 nowData_format .right_indent = Inches(0.3)  
43  
44 # 这一步调整文件格式为居中  
45 doc.save('test.docx' )
```

以上实例输出结果如下：

公司员工请假条

__部:

因_____，特向您请事假天。请假时间自_年月日至_年月日。这段时间内原计划安排的课程已做好处理，希望领导批准。

请假人:

年 月 日

4.2 Excel转换为word

Excel

猿来教育
Lucky出品 必属精品

Office 更新 若要保持使用最新安全更新、修复和改进，请选择“检查更新”。

J15

✕

✓

fx

	A	B	C	D	E	F	G	H
1	模板类别	模板名称	流程说明	节点名称	节点ID	审批方式	审批人员	
2	模板类别1	模板名称1	流程说明1	节点名称1	节点ID1	审批方式1	审批人员1	
3	模板类别2	模板名称2	流程说明2	节点名称2	节点ID2	审批方式2	审批人员2	
4	模板类别3	模板名称3	流程说明3	节点名称3	节点ID3	审批方式3	审批人员3	
5	模板类别4	模板名称4	流程说明4	节点名称4	节点ID4	审批方式4	审批人员4	
6	模板类别5	模板名称5	流程说明5	节点名称5	节点ID5	审批方式5	审批人员5	
7	模板类别6	模板名称6	流程说明6	节点名称6	节点ID6	审批方式6	审批人员6	
8	模板类别7	模板名称7	流程说明7	节点名称7	节点ID7	审批方式7	审批人员7	
9	模板类别8	模板名称8	流程说明8	节点名称8	节点ID8	审批方式8	审批人员8	
10	模板类别9	模板名称9	流程说明9	节点名称9	节点ID9	审批方式9	审批人员9	
11	模板类别10	模板名称10	流程说明10	节点名称10	节点ID10	审批方式10	审批人员10	
12	模板类别11	模板名称11	流程说明11	节点名称11	节点ID11	审批方式11	审批人员11	
13	模板类别12	模板名称12	流程说明12	节点名称12	节点ID12	审批方式12	审批人员12	
14	模板类别13	模板名称13	流程说明13	节点名称13	节点ID13	审批方式13	审批人员13	
15	模板类别14	模板名称14	流程说明14	节点名称14	节点ID14	审批方式14	审批人员14	
16	模板类别15	模板名称15	流程说明15	节点名称15	节点ID15	审批方式15	审批人员15	
17								
18								
19								
20								

转换后word

模板类别 1

模板名称 1

流程名称:	模板名称 1			
使用人:	模板类别 1			
流程说明:	流程说明 1			
节点	节点名	处理人员	处理方式	跳转信息
1	节点名称 1	审批人员 1	审批方式 1	

模板类别 2

模板名称 2

流程名称:	模板名称 2			
使用人:	模板类别 2			
流程说明:	流程说明 2			
节点	节点名	处理人员	处理方式	跳转信息
1	节点名称 2	审批人员 2	审批方式 2	

代码实现

```
1 import xlrd
2 from docx import Document
3 from docx.enum.section import WD_ORIENTATION
4 from docx.enum.text import
  WD_PARAGRAPH_ALIGNMENT
5 from docx.shared import Pt, Cm, RGBColor
6 from docx.xml.ns import qn
7 from docx.enum.table import
  WD_CELL_VERTICAL_ALIGNMENT
```

8

9


```

10 def read_excel():
11     """读取Excel"""
12     book = xlrd.open_workbook('test.xlsx')
13     sheet =
book.sheet_by_name(book.sheet_names()[0])
14     nrows = sheet.nrows # 行数
15     ncols = sheet.ncols # 列数
16     datas = [] # 存放数据
17     # 第一列 标题
18     keys = sheet.row_values(0)
19     for row in range(1, nrows):
20         data = {} # 每一行数据
21         for col in range(0, ncols):
22             value = sheet.cell_value(row, col) #
取出每一个单元格的数据
23             # 替换到特殊字符
24             value = value.replace('<',
''.replace('>', '').replace('$', ''))
25             data[keys[col]] = value
26             # 截取第一列元素
27             if col == 0:
28                 first = '' # 截取元素 第1
29                 second = '' # 截取元素 第2
30                 third = '' # 截取元素 第3
31                 arrs =
value.lstrip('/').split('/') # 去掉第一个/ 然
后再以/分组

```

```
32         if len(arrs) > 0:
33             if len(arrs) == 1:
34                 first = arrs[0]
35                 second = first
36                 third = second
37             elif len(arrs) == 2:
38                 first = arrs[0]
39                 second = arrs[1]
40                 third = second
41             elif len(arrs) == 3:
42                 first = arrs[0]
43                 second = arrs[1]
44                 third = arrs[2]
45             else:
46                 first = arrs[0]
47                 second = arrs[1]
48                 third = arrs[2]
49         else:
50             first = value.lstrip('/')
51             second = first
52             third = second
53         data['first'] = first
54         data['second'] = second
55         data['third'] = third
56         # 截取第一列结束
57         datas.append(data)
58     return datas
```

```
59
60
61
62 def write_word(datas):
63     """生成word文件"""
64     if len(datas) < 1:
65         print('Excel没有内容')
66         return
67     # 定义word文档对象
68     doc = Document()
69     # 添加横向
70     section = doc.sections[0] #
71     doc.add_section(start_type=WD_SECTION_START.
72     CONTINUOUS) # 添加横向页的连续节
73     section.orientation =
74     WD_ORIENTATION.LANDSCAPE
75     page_h, page_w = section.page_width,
76     section.page_height
77     section.page_width = page_w # 设置横向纸的
78     宽度
79     section.page_height = page_h # 设置横向纸
80     的高度
81     # 设置字体
82     doc.styles['Normal'].font.name = u'宋体'
83
84     doc.styles['Normal']._element.rPr.rFonts.set(
85     qn('w: eastAsia'), u'宋体')
```

```
78     # 获取第3部分(部门) 并去重
79     data_third = []
80     for data in datas:
81         third = data['third']
82         if data_third.count(third) == 0:
83             data_third.append(third)
84     for third in data_third:
85         h2 = doc.add_heading(third, level=2) #
写入部门,二级标题
86         run = h2.runs[0] # 可以通过add_run来设置
文字,也可以通过数组来获取
87         run.font.color.rgb = RGBColor(0, 0, 0)
88         run.font.name = u'宋体'
89         doc.add_paragraph(text='', style=None)
# 增加空白行 换行
90         # 开始获取模板
91         data_template = []
92         for data in datas:
93             if data['third'] == third:
94                 template = {'first':
data['first'], '模板名称': data['模板名称']}
95                 if data_template.count(template)
== 0:
96                     data_template.append(template)
97                 # 获取模板完成
98                 # 遍历模板
99                 for template in data_template:
```

```
100     h3 = doc.add_heading(template['模板名
101 称'], level=3) # 插入模板名称,三级标题
102     run = h3.runs[0] # 可以通过add_run来设
103 置文字,也可以通过数组来获取
104     run.font.color.rgb = RGBColor(0, 0,
105 0)
106     run.font.name = u'宋体'
107     doc.add_paragraph(text='',
108 style=None) # 换行
109     data_table = filter(
110         lambda data: data['third'] ==
111 third and data['模板名称'] == template['模板名
112 称'] and data['first'] ==
113 template['first'], datas)
114     data_table = list(data_table)
115     # 新增表格 4行5列
116     doc_table = doc.add_table(rows=4,
117 cols=5)
118     doc_table.style = "Table Grid"
119     doc_table.style.font.size = Pt(9)
120     doc_table.style.font.name = '宋体'
121
122     # 合并单元格 赋值
123
124     doc_table.rows[0].cells[1].merge(doc_table.
125 rows[0].cells[4])
```

```
117     doc_table.rows[1].cells[1].merge(doc_table.
rows[1].cells[4])
118
    doc_table.rows[2].cells[1].merge(doc_table.
rows[2].cells[4])
119     doc_table.rows[0].cells[0].text =
'流程名称: '
120     doc_table.rows[0].cells[1].text =
data_table[0]['模板名称']
121     doc_table.rows[1].cells[0].text =
'使用人: '
122     doc_table.rows[1].cells[1].text =
data_table[0]['first']
123     doc_table.rows[2].cells[0].text =
'流程说明: '
124     doc_table.rows[2].cells[1].text =
data_table[0]['流程说明']
125
    # 设置标题
126     head_cells = doc_table.rows[3].cells
# 前面还有三行, 特殊处理
127
    head_cells[0].text = '节点'
128     head_cells[1].text = '节点名'
129     head_cells[2].text = '处理人员'
130     head_cells[3].text = '处理方式'
131     head_cells[4].text = '跳转信息'
132
```

```
133         # 设置列宽
134         head_cells[0].width = Cm(1.9)
135         head_cells[1].width = Cm(4.83)
136         head_cells[2].width = Cm(8.25)
137         head_cells[3].width = Cm(2.54)
138         head_cells[4].width = Cm(5.64)
139         # 第1 列水平居中，并设置行高，所有单元格内
容垂直居中
140         for i in range(0, 4):
141             # 水平居中
142             p =
doc_table.rows[i].cells[0].paragraphs[0]
143             p.alignment =
WD_PARAGRAPH_ALIGNMENT.CENTER
144             doc_table.rows[i].height = Cm(0.6)
# 行高
145             # 垂直居中
146             for j in range(0, 5):
147
doc_table.rows[i].cells[j].vertical_alignme
nt = WD_CELL_VERTICAL_ALIGNMENT.CENTER
148
149         # 生成表格并填充内容
150         row_num = 0
151         for data in data_table:
152             row = doc_table.add_row()
153             row_cells = row.cells
```

```
154         row_cells[0].text = str(row_num +
1) # 序号, 需要转换成字符串
155         row_cells[1].text = data['节点名
称']
156         row_cells[2].text = data['审批人
员']
157         row_cells[3].text = data['审批方
式']
158         row_cells[4].text = ''
159         # 水平居中
160         p = row_cells[0].paragraphs[0]
161         p.alignment =
WD_PARAGRAPH_ALIGNMENT.CENTER
162         row.height = Cm(0.6) # 行高
163         # 垂直居中
164         for j in range(0, 5):
165             row_cells[j].vertical_alignment
= WD_CELL_VERTICAL_ALIGNMENT.CENTER
166         row_num = row_num + 1
167
168         doc.add_paragraph(text='',
style=None) # 换行
169         doc.save('test.docx')
170
171
172 print(write_word(read_excel()))
```


猿来教育

lucky出品 必属精品