前言

本说明主要针对工程机械报告出具挖掘机技术部分做详细讲解，探讨报告出具的核心和精髓。

制作部分：工程机械事业部

制作人：康亚卓

Part1 封面

封面格式较为固定，按统一格式插入即可

1.1 设置内联图片

纸张为A4(21mm*29.7mm)

```
document = Document()
    section = document.sections[0]
    section.page_width = Cm(21)
    section.page_height = Cm(29.7)
```

首页有插章和CNAS字样的图片，观察发现其位置浮于文字上方即浮动图片(floating picture)，但是帮助文档内找不到相关说明，本节要通过剖析xml、追踪源码，最后得到完整代码。
转换为xml型式，观察区别：
内联图片片段：

```
<w:drawing>
    <wp:inline>
        <wp:extent cx="3297600" cy="2782800"/>
        <wp:effectExtent l="0" t="0" r="0" b="0"/>
        <wp:docPr id="1" name="Picture 1"/>
        <wp:cNvGraphicFramePr>
            <a:graphicFrameLocks/>
        </wp:cNvGraphicFramePr>
        <a:graphic>
            <a:graphicData>
                <pic:pic>
                    <!-- more pic content -->
                </pic:pic>
            </a:graphicData>
        </a:graphic>
    </wp:inline>
</w:drawing>
```

浮动图片片段:

```
<w:drawing>
    <wp:anchor behindDoc="1" locked="0" layoutInCell="1" allowOverlap="1">
        <wp:simplePos x="0" y="0"/>
        <wp:positionH relativeFrom="page">
            <wp:posOffset>285750</wp:posOffset>
        </wp:positionH>
```

```
        <wp:positionV relativeFrom="page">
            <wp:posOffset>457200</wp:posOffset>
        </wp:positionV>
        <wp:extent cx="3297600" cy="2782800"/>
        <wp:effectExtent l="0" t="0" r="0" b="0"/>
        <wp:wrapNone/>
        <wp:docPr id="1" name="Picture 1"/>
        <wp:cNvGraphicFramePr>
            <a:graphicFrameLocks/>
        </wp:cNvGraphicFramePr>
        <a:graphic>
            <a:graphicData>
                <pic:pic>
                    <!-- more pic content -->
                </pic:pic>
            </a:graphicData>
        </a:graphic>
    </wp:anchor>
</w:drawing>
```

对比发现相似点：

- 两类图片都放在`<w:drawing>`节点下：内联图片`<wp:inline>`，浮动图片`<wp:anchor>`
- 具备相同的内容节点：`<wp:extent>`、`<wp:docPr>`、`<a:graphic>`等

  还有一些猜测：

  - wp:anchor节点的behindDoc属性表明图片版式为衬于文本下方
  - wp:positionH和wp:positionV节点表明水平和竖直绝对定位方式，其中：
  - ■ relativeFrom属性指定用于定位的参考对象
  - ■ 子节点`<wp:posOffset>`指定具体坐标值

从内联图片开始

```
from docx import Document
from docx.shared import Pt

document = Document()
document.add_picture('资质图片.jpg', width=Pt(200))
document.save('output.docx')
```

从`python-docx`安装文件夹`site-packages/docx`进行内容搜索`add_picture`，得到`docx.text.run.add_picture`原始定义处：

```
def add_picture(self, image_path_or_stream, width=None, height=None):
    inline = self.part.new_pic_inline(image_path_or_stream, width, height)
    self._r.add_drawing(inline)
    return InlineShape(inline)
```

继续搜索`new_pic_inline`得到`docx.parts.story.BaseStoryPart.new_pic_inline`。从注释可知这是利用`CT_Inline`类创建`<wp:inline>`元素，因此后续创建浮动图片的`<wp:anchor>`可以在此基础上修改。

```
def new_pic_inline(self, image_descriptor, width, height):
    """Return a newly-created `w:inline` element.
    The element contains the image specified by *image_descriptor* and is scaled
    based on the values of *width* and *height*.
    """
    rId, image = self.get_or_add_image(image_descriptor)
    cx, cy = image.scaled_dimensions(width, height)
    shape_id, filename = self.next_id, image.filename
    return CT_Inline.new_pic_inline(shape_id, rId, filename, cx, cy)
```

进入`CT_Inline`类

```
class CT_Inline(BaseOxmlElement):
    """
    ``<w:inline>`` element, container for an inline shape.
    """
    @classmethod
    def new(cls, cx, cy, shape_id, pic):
        pass

    @classmethod
    def new_pic_inline(cls, shape_id, rId, filename, cx, cy):
        pass

    @classmethod
    def _inline_xml(cls):
        return (
            '<wp:inline %s>\n'
            '  <wp:extent cx="914400" cy="914400"/>\n'
            '  <wp:docPr id="666" name="unnamed"/>\n'
            '  <wp:cNvGraphicFramePr>\n'
            '    <a:graphicFrameLocks noChangeAspect="1"/>\n'
            '  </wp:cNvGraphicFramePr>\n'
            '  <a:graphic>\n'
            '    <a:graphicData uri="URI not set"/>\n'
            '  </a:graphic>\n'
            '</wp:inline>' % nsdecls('wp', 'a', 'pic', 'r')
        )
```

- `_inline_xml()`方法给出内联图片`<wp:inline>`的xml结构。
- `new()`方法调用`_inline_xml()`，并为其中的子节点例如`<wp:extent>`和`<wp:docPr>`赋值。
- `new_pic_inline()`调用`new()`，同时拼接`CT_Picture`类的结果（节点`<pic:pic>`，即图片的具体内容）到`<a:graphicData>`节点中去。

从xml结构的对比及上述`python-docx`对内联图片的实现，得到创建浮动图片的思路：

- 初始化`<wp:anchor>`结构，例如`behindDoc="1"`指定图片版式为衬于文本下方
- 使用类似的代码填充`<wp:anchor>`元素，尤其是`<wp:extent>`、`<wp:docPr>`和`<pic:pic>`
- 填充`<wp:positionH>`和`<wp:positionV>`精确定位图片

综上，利用`python-docx`插入浮动图片（衬于文本下方、页面定位）的完整代码如下：

(在目录下添加这么一行)

```
# -*- coding: utf-8 -*-
```

```python
# filename: add_float_picture.py

'''
Implement floating image based on python-docx.
- Text wrapping style: BEHIND TEXT <wp:anchor behindDoc="1">
- Picture position: top-
left corner of PAGE `<wp:positionH relativeFrom="page">`.
Create a docx sample (Layout | Positions | More Layout Options) and explore the
source xml (Open as a zip | word | document.xml) to implement other text wrappin
g
styles and position modes per `CT_Anchor._anchor_xml()`.
'''

from docx.oxml import parse_xml, register_element_cls
from docx.oxml.ns import nsdecls
from docx.oxml.shape import CT_Picture
from docx.oxml.xmlchemy import BaseOxmlElement, OneAndOnlyOne

# refer to docx.oxml.shape.CT_Inline
class CT_Anchor(BaseOxmlElement):
    """
    ``<w:anchor>`` element, container for a floating image.
    """
    extent = OneAndOnlyOne('wp:extent')
    docPr = OneAndOnlyOne('wp:docPr')
    graphic = OneAndOnlyOne('a:graphic')

    @classmethod
    def new(cls, cx, cy, shape_id, pic, pos_x, pos_y):
        """
        Return a new ``<wp:anchor>`` element populated with the values passed
        as parameters.
        """
        anchor = parse_xml(cls._anchor_xml(pos_x, pos_y))
        anchor.extent.cx = cx
        anchor.extent.cy = cy
        anchor.docPr.id = shape_id
        anchor.docPr.name = 'Picture %d' % shape_id
        anchor.graphic.graphicData.uri = (
            'http://schemas.openxmlformats.org/drawingml/2006/picture'
        )
        anchor.graphic.graphicData._insert_pic(pic)
        return anchor

    @classmethod
    def new_pic_anchor(cls, shape_id, rId, filename, cx, cy, pos_x, pos_y):
        """
        Return a new `wp:anchor` element containing the `pic:pic` element
        specified by the argument values.
        """
        pic_id = 0  # Word doesn't seem to use this, but does not omit it
        pic = CT_Picture.new(pic_id, filename, rId, cx, cy)
        anchor = cls.new(cx, cy, shape_id, pic, pos_x, pos_y)
        anchor.graphic.graphicData._insert_pic(pic)
        return anchor
    @classmethod
    def _anchor_xml(cls, pos_x, pos_y):
```

```
        return (
            '<wp:anchor distT="0" distB="0" distL="0" distR="0" simplePos="0" re
lativeHeight="0" \n'
            '            behindDoc="1" locked="0" layoutInCell="1" allowOverlap="
1" \n'
            '            %s>\n'
            '  <wp:simplePos x="0" y="0"/>\n'
            '  <wp:positionH relativeFrom="page">\n'
            '    <wp:posOffset>%d</wp:posOffset>\n'
            '  </wp:positionH>\n'
            '  <wp:positionV relativeFrom="page">\n'
            '    <wp:posOffset>%d</wp:posOffset>\n'
            '  </wp:positionV>\n'
            '  <wp:extent cx="914400" cy="914400"/>\n'
            '  <wp:wrapNone/>\n'
            '  <wp:docPr id="666" name="unnamed"/>\n'
            '  <wp:cNvGraphicFramePr>\n'
            '    <a:graphicFrameLocks noChangeAspect="1"/>\n'
            '  </wp:cNvGraphicFramePr>\n'
            '  <a:graphic>\n'
            '    <a:graphicData uri="URI not set"/>\n'
            '  </a:graphic>\n'
            '</wp:anchor>' % ( nsdecls('wp', 'a', 'pic', 'r'), int(pos_x), int(p
os_y) )
        )
# refer to docx.parts.story.BaseStoryPart.new_pic_inline
def new_pic_anchor(part, image_descriptor, width, height, pos_x, pos_y):
    """Return a newly-created `w:anchor` element.
    The element contains the image specified by *image_descriptor* and is scaled
    based on the values of *width* and *height*.
    """
    rId, image = part.get_or_add_image(image_descriptor)
    cx, cy = image.scaled_dimensions(width, height)
    shape_id, filename = part.next_id, image.filename
    return CT_Anchor.new_pic_anchor(shape_id, rId, filename, cx, cy, pos_x, pos_
y)
# refer to docx.text.run.add_picture
def add_float_picture(p, image_path_or_stream, width=None, height=None, pos_x=0,
 pos_y=0):
    """Add float picture at fixed position `pos_x` and `pos_y` to the top-
left point of page.
    """
    run = p.add_run()
    anchor = new_pic_anchor(run.part, image_path_or_stream, width, height, pos_x
, pos_y)
    run._r.add_drawing(anchor)
# refer to docx.oxml.__init__.py
register_element_cls('wp:anchor', CT_Anchor)
```

观察到 GGJ02.2020.MY03.01.514报告封面资质章宽度为18.69cm，高度为3.7cm,公司章宽度为8.25厘米，高度为4.41cm即

$$\text{资质章尺寸：} 7.36 \times 1.46in$$

$$\text{公司章尺寸：} 3.25 \times 1.74in$$

我已经准备好高宽比的图片，且锁定横纵比

A4纸的大小为21mm*29.7mm，要居中的话，资质章距左端为

$$Pox = \frac{21 \div 2.54 - 7.36}{2} = 0.454$$

公司章距左端为

$$pox = \frac{21 \div 2.54 - 3.25}{2} = 2.51$$

另一部分:

```
from docx import Document
from docx.shared import Cm,Inches, Pt
from add_float_picture import add_float_picture
if __name__ == '__main__':
    document = Document()
    section = document.sections[0]
    section.page_width = Cm(21)
    section.page_height = Cm(29.7)
    p = document.add_paragraph()
    add_float_picture(p, 'D:/newfile/python/资质图片.png', width=Inches(7.36),
pos_x=Inches(0.454), pos_y=Pt(30))
    add_float_picture(p, 'D:/newfile/python/印章.png', width=Inches(3.25),
pos_x=Inches(2.51), pos_y=Pt(660))

    document.save('D:/newfile/python/output.docx')
```

至此，首页浮动图片插入完毕。

1.2 普通文本

此项较为简单，按照流程:

报告编号字样出现单倍行距，四号，第六行

产品名称1.5倍行距，缩进3.8厘米，三号

公司名三号剧中1.5倍行距

```
coverReportnumber=document.add_paragraph() #报告编号
coverReportnumber.alignment=WD_ALIGN_PARAGRAPH.RIGHT
coverReportNumberrun=coverReportnumber.add_run('\n\n\n\n\n报告编号:
GGJ02.2020.MY03.01.514\n\n\n\n')
coverReportNumberrun.font.size=Pt(14)
coverReportNumberrun.font.name='Arial'
coverReportNumberrun._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
coverReportnumber.paragraph_format.line_spacing=1.0
coverReportnumber.paragraph_format.space_after=Pt(0)  #失效
coverReportnumber.paragraph_format.space_before=Pt(0)
def inspectionReport():
    inspectionreport=document.add_paragraph() #检验报告
    inspectionreport.alignment=WD_ALIGN_PARAGRAPH.CENTER
    inspectionReportrun=inspectionreport.add_run('检验报告')
    inspectionReportrun.font.size=Pt(42)
    inspectionReportrun.font.name='Arial'
    inspectionReportrun._element.rPr.rFonts.set(qn('w:eastAsia'),'黑体')
```

```
    inspectionreport.paragraph_format.line_spacing=1.0
    inspectionreport.paragraph_format.space_after=Pt(0)
    inspectionreport.paragraph_format.space_before=Pt(0)
inspectionReport()
def maincontent():
    maincontent=document.add_paragraph() #主要内容
    maincontent.alignment=WD_ALIGN_PARAGRAPH.LEFT
    mainContentrun= maincontent.add_run('\n\n\n\n产品名称:液压挖掘机\n产品型
号:CLG926EHD\n委托方:柳州柳工挖掘机有限公司\n检验类型：型式检验\n\n\n\n')
    mainContentrun.font.size=Pt(16)
    mainContentrun.font.name='Arial'
    mainContentrun._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
    maincontent.paragraph_format.line_spacing=1.5
    maincontent.paragraph_format.space_after=Pt(0)
    maincontent.paragraph_format.space_after=Pt(0)
    maincontent.paragraph_format.left_indent=Cm(3.8)
maincontent()
downcontent=document.add_paragraph() #下方内容
downcontent.alignment=WD_ALIGN_PARAGRAPH.CENTER
downContentrun=downcontent.add_run('中机科（北京）车辆检测工程研究院有限公司\n国家工程机
械质量监督检验中心\n二零二一年四月\n')
downContentrun.font.size=Pt(16)
downContentrun.font.name='Arial'
downContentrun._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
downcontent.paragraph_format.line_spacing=1.5
downcontent.paragraph_format.space_after=Pt(0)
downcontent.paragraph_format.space_after=Pt(0)
document.add_page_break() #分页符
flyleaf1=document.add_paragraph()
flyleaf1.alignment=WD_ALIGN_PARAGRAPH.CENTER
flyleaf1run=flyleaf1.add_run('注意事项\n')
flyleaf1run.font.size=Pt(16)
flyleaf1run.font.name='Arial'
flyleaf1run._element.rPr.rFonts.set(qn('w:eastAsia'),'黑体')
flyleaf1run.bold=True
flyleaf1.paragraph_format.line_spacing=1
flyleaf1.paragraph_format.space_after=Pt(0)
flyleaf1.paragraph_format.space_after=Pt(0)

flyleaf2=document.add_paragraph()
flyleaf2.alignment=WD_ALIGN_PARAGRAPH.LEFT
flyleaf2run=flyleaf2.add_run('1.报告无“检验报告专用章”或型式检验检测机构公章无效；报告无加
盖骑缝章无效。\n2.复制报告未重新加盖“检验报告专用章”或型式试验检验检测机构公章无效；复制报告未重
新加盖骑缝章无效。\n3.报告无主检、审核、签发人签字无效。\n4.报告涂改无效。\n5.报告是对设备型式
的确认，对样品本身的合格与否负责，且仅对符合送样样品的产品有效。')
flyleaf2run.font.size=Pt(14)
flyleaf2run.font.name='Arial'
flyleaf2run._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
flyleaf2.paragraph_format.line_spacing=1.5
flyleaf2.paragraph_format.space_after=Pt(0)
flyleaf2.paragraph_format.space_after=Pt(0)
flyleaftable=document.add_table(rows=1,cols=1)
flyleaftable.cell(0,0).width=Cm(15.5)
flyleaftable.cell(0,0).height=Cm(0.04) ##

flyleaf3=document.add_paragraph()
flyleaf3.alignment=WD_ALIGN_PARAGRAPH.CENTER
flyleaf3run=flyleaf3.add_run('报告附加说明')
```

```
flyleaf3run.font.size=Pt(16)
flyleaf3run.font.name='Arial'
flyleaf3run._element.rPr.rFonts.set(qn('w:eastAsia'),'黑体')
flyleaf3run.bold=True
flyleaf3.paragraph_format.line_spacing=1
flyleaf3.paragraph_format.space_after=Pt(0)
flyleaf3.paragraph_format.space_after=Pt(0)
flyleaf4=document.add_paragraph()
flyleaf4.alignment=WD_ALIGN_PARAGRAPH.LEFT
def flyleaf40(part1,part2,part3):
    flyleaf401run=flyleaf4.add_run(part1)
    flyleaf401run.font.size=Pt(14)
    flyleaf401run.font.name=u'宋体'
    flyleaf401run._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
    flyleaf402run=flyleaf4.add_run(part2)
    flyleaf402run.font.color.rgb= RGBColor(255,255,255)
    flyleaf402run.font.size=Pt(14)
    flyleaf402run.font.name=u'宋体'
    flyleaf402run._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
    flyleaf403run=flyleaf4.add_run(part3)
    flyleaf403run.font.size=Pt(14)
    flyleaf403run.font.name=u'宋体'
    flyleaf403run._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
flyleaf40('1.','-','委托方地址：广西柳州市柳江县拉堡镇双拥路\n')
flyleaf40('2.','-','委托方电话：0772-7257200\n')
flyleaf40('3.','-','委托方法定代表人：黄敏\n')
flyleaf40('4.','-','制造商地址：广西柳州市柳江县拉堡镇双拥路\n')
flyleaf40('5.','-','制造商电话：0772-7257200\n')
flyleaf40('6.','-','制造商法定代表人：黄敏\n')
flyleaf40('7.','-','性能试验与安全要求验证用样机接受日期：2020年11月09日\n')
flyleaf40('8.','-','可靠性试验用样机接受日期：2020年04月01日\n')
flyleaf40('9.','-','试验项目有无外包：无\n')
flyleaf40('10.','-','检验检测机构地址：北京市延庆区东外大街55号\n')
flyleaf40('11.','-','检验检测机构电话：010-69141870\n')
flyleaf40('12.','-','投诉电话：010-69145748\n')
document.add_page_break() #分页符
```

Part2 目录

目录并非自动生成，格式固定占一页，为了方便程序阅读，每一个章都采用单独的模块

```
################################################
def content(indent,part1,part2,part3):
    cont=document.add_paragraph()
    cont.alignment=WD_ALIGN_PARAGRAPH.DISTRIBUTE
    cont.paragraph_format.line_spacing=1
    cont.paragraph_format.space_after=Pt(0)
    cont.paragraph_format.space_after=Pt(0)
    cont.paragraph_format.left_indent=Inches(indent)
    run=cont.add_run(part1)
    run.font.size=Pt(12)
    run.font.name=u'宋体'
    run._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
    run1=cont.add_run(part2)
    run1.font.size=Pt(12)
    run1.font.name=u'宋体'
```

```
    run1._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
    run2=cont.add_run(part3)
    run2.font.size=Pt(12)
    run2.font.name=u'宋体'
    run2._element.rPr.rFonts.set(qn('w:eastAsia'),'宋体')
        #
content (0,'检验结论','.....','1')
content (0,'附录A 试验对象','.....','2')
content (0,'附录B 试验依据','.....','7')
content (0,'附录C 试验条件','.....','8')
content (0,'附录D 试验结果','.....','9')
content (0,'D1试验前样机验收检查结果','.....','9')
content (0,'D2性能试验结果','.....','9')
content (0.25,'D2.1外形尺寸测定结果','.....','9')
content (0.25,'D2.2作业参数测定结果','.....','9')
content (0.25,'D2.3质量测定结果','.....','10')
content (0.25,'D2.4接地比压测定结果','.....','10')
content (0.25,'D2.5最大挖掘力测定结果','.....','10')
content (0.25,'D2.6行驶速度试验结果','.....','10')
content (0.25,'D2.7爬坡能力试验结果','.....','10')
content (0.25,'D2.8履带式挖掘机行驶直线性测定结果','.....','11')
content (0.25,'D2.9作业试验结果','.....','11')
content (0.25,'D2.10强度试验结果','.....','11')
content (0.25,'D2.11液压系统试验结果','.....','13')
content (0,'D3安全要求和防护措施的验证结果','.....','14')
content (0.25,'D3.1通道检查结果','.....','14')
content (0.25,'D3.2司机操作位置检查结果','.....','16')
content (0.25,'D3.3座椅全检查结果','.....','19')
content (0.25,'D3.4司机的操纵装置和指示装置检查结果','.....','20')
content (0.25,'D3.5转向系统检查结果','.....','24')
content (0.25,'D3.6制动系统检查结果','.....','24')
content (0.25,'D3.7可视性检查结果','.....','26')
content (0.25,'D3.8报警装置和安全标志检查结果','.....','26')
content (0.25,'D3.9稳定性测试结果','.....','31')
content (0.25,'D3.10噪声测试结果','.....','31')
content (0.25,'D3.11保护措施及装置检查结果','.....','32')
content (0.25,'D3.12救助、捆系、起吊、牵引和运输检查结果','.....','34')
content (0.25,'D3.13电气和电子系统检查结果','.....','35')
content (0.25,'D3.14压力系统检查结果','.....','37')
content (0.25,'D3.15燃油箱和液压油箱检查结果','.....','38')
content (0.25,'D3.16防火检查结果','.....','38')
content (0.25,'D3.17维修检查结果','.....','38')
content (0.25,'D3.18使用信息检查结果','.....','39')
content (0,'D4可靠性试验结果','.....','39')
content (0,'附录E  参试人员','.....','40')
content (0,'附录F  试验照片','.....','41')
document.add_page_break() #分页符
```

part3 检验结论

此页格式固定，占一页，表格单行高度1.4厘米，前四行行高值是固定值（无论如何此表格要占满且仅占一页所以格式固定），第5-7行行高是最小值，第5行行高1.2Cm，第6行行高9.65Cm,第7行行高2.19Cm

设置行高规则

```
from docx.enum.table import WD_ROW_HEIGHT_RULE
#########################################################
    for rr in range(4):
        table.rows[rr].height_rule = WD_ROW_HEIGHT_RULE.EXACTLY
        table.rows[rr].height=Cm(1.4)
    table.rows[4].height_rule = WD_ROW_HEIGHT_RULE.AT_LEAST
    table.rows[4].height=Cm(1.2)
    table.rows[5].height_rule = WD_ROW_HEIGHT_RULE.AT_LEAST
    table.rows[5].height=Cm(9.65)
    table.rows[6].height_rule = WD_ROW_HEIGHT_RULE.AT_LEAST
    table.rows[6].height=Cm(2.19)
```

插签字页

关于表格中文字属性的设定，可以逐一设定单元格:

```
from docx.enum.text import WD_PARAGRAPH_ALIGNMENT
    p=table1.cell(0,0).paragraphs[0]
    run=p.add_run('主检：')
    run.font.color.rgb = RGBColor(255, 0, 0)
    run.font.size = Pt(15)
    table1.cell(0,0).paragraphs[0].paragraph_format.alignment =
WD_PARAGRAPH_ALIGNMENT.CENTER
```

也可以设置整个表格：

```
table1.style.font.size=Pt(15)
table1.style.font.color.rgb=RGBColor(255, 0, 0)
table1.style.paragraph_format.alignment=WD_PARAGRAPH_ALIGNMENT.CENTER
```

表格1中设置如下

3.1表格、单元格对齐方式

表格的对齐方式

在docx.enum.table import WD_TABLE_ALIGNMENT中，共定义了LEFT、CENTER和RIGHT三个常量。含义如下
WD_TABLE_ALIGNMENT.LEFT ：表格为左对齐WD_TABLE_ALIGNMENT.CENTER：表格为居中对齐
WD_TABLE_ALIGNMENT.RIGHT：表格为右对齐
WD_TABLE_ALIGNMENT.RIGHT：表格为右对齐

单元格对齐方式

在对单元格对齐方式设置的时候，将单元格视为一个整体，要使用单元格中的垂直对齐（cell.vertical_alignment）和单元格中的段落的对齐（paragraph.alignment）等2种对齐方式配合使用。在docx.enum.table .WD_ALIGN_VERTICAL定义了TOP、CENTER和BOTTOM等3种类型，含义如下：
WD_CELL_VERTICAL_ALIGNMENT.TOP：单元格内容靠上对齐
WD_CELL_VERTICAL_ALIGNMENT.CENTER：单元格内容居中对齐
WD_CELL_VERTICAL_ALIGNMENT.BOTTOM：单元格内容靠下对齐
在WD_PARAGRAPH_ALIGNMENT中定义了4中类型，分别是LEFT、CENTER、RIGHT和JUSTIFY等4中类型，含义如下：

WD_PARAGRAPH_ALIGNMENT.LEFT：段落左对齐
WD_PARAGRAPH_ALIGNMENT.CENTER：段落居中对齐
WD_PARAGRAPH_ALIGNMENT.RIGHT：段落右对齐
WD_PARAGRAPH_ALIGNMENT.JUSTIFY：段落两端对齐
对齐方式共有12种可能

```
from docx.enum.table import WD_CELL_VERTICAL_ALIGNMENT #单元格对齐方式
from docx.enum.text import WD_PARAGRAPH_ALIGNMENT      #段落对齐方式
```

```
cell = table.cell(0,0)
#1 靠上两端对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.TOP
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.JUSTIFY
#2 靠上居中对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.TOP
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.CENTER
#3 靠上右对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.TOP
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.RIGHT
#4 中部两端对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.CENTER
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.JUSTIFY
#5 中部居中对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.CENTER
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.CENTER
#6 中部右对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.CENTER
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.RIGHT
#7 靠下两端对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.BOTTOM
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.JUSTIFY
#8 靠下中部对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.BOTTOM
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.CENTER
#9 靠下右对齐
cell.vertical_alignment = WD_CELL_VERTICAL_ALIGNMENT.BOTTOM
cell.paragraphs[0].alignment = WD_PARAGRAPH_ALIGNMENT.RIGHT
#10 左对齐略
#11 左对齐略
#12 左对齐略
```

代码中，表格编号如下：

<table>
<tr><td>[0, 0]</td><td>[0, 1]</td><td>[0, 2]</td><td>[0, 3]</td></tr>
<tr><td>(1, 0)</td><td>[1, 1]</td><td>[1, 2]</td><td>[1, 3]</td></tr>
<tr><td>[2, 0]</td><td>[2, 1]</td><td rowspan="2">A</td><td rowspan="2">C</td></tr>
<tr><td>[3, 2]</td><td>[3, 3]</td></tr>
<tr><td>[4, 0]</td><td>[4, 1]</td><td>[4, 2]</td><td>[4, 3]</td></tr>
<tr><td>[5, 0]</td><td colspan="3">B</td></tr>
<tr><td>[6, 0]</td><td colspan="3">D</td></tr>
</table>