Python中的pdf文件处理

1. PyPDF2

PyPDF2是一个纯python的PDF库，能够拆分、合并、裁剪和转换PDF文件的页面。它还可以向PDF文件添加自定义数据、查看选项和密码。它可以从pdf中检索文本和元数据，还可以将整个文件合并在一起。
PyPDF2中主要提供了两个类，分别是PdfFileReader和PdfFileWriter用于读取pdf文件内容和修改pdf文件内容

PdfFileReader

使用PdfFileReader读取pdf文件前需要先创建一个PdfFileReader的对象：

```
    PyPDF2.PdfFileReader(stream, strict = True,warndest = None,overwriteWarnings =
True)
```

- **stream： File 对象或支持与 File 对象类似的标准读取和查找方法的对象**，也可以是表示 PDF 文件路径的字符串。
- **strict（bool）**： 确定是否应该警告用户所用的问题，也导致一些可纠正的问题是致命的，默认是 True
- **warndest** : 记录警告的目标(默认是 sys.stderr)
- **overwriteWarnings(bool)**：确定是否 warnings.py 用自定义实现覆盖 Python 模块（默认为 True）

PdfFileReader 对象的属性和方法

属性和方法	描述
getDestinationPageNumber(destination)	检索给定目标对象的页码
getDocumentInfo()	检索 PDF 文件的文档信息字典
getFields(tree = None,retval = None,fileObj= None)	如果此 PDF 包含交互式表单字段，则提取字段数据，
getFormTextFields()	从文档中检索带有文本数据（输入，下拉列表）的表单域
getNameDestinations(tree = None,retval= None)	检索文档中的指定目标
getNumPages()	计算此 PDF 文件中的页数
getOutlines(node = None,outline = None,)	检索文档中出现的文档大纲
getPage(pageNumber)	从这个 PDF 文件中检索指定编号的页面
getPageLayout()	获取页面布局
getPageMode()	获取页面模式
getPageNumber(pageObject)	检索给定 pageObject 处于的页码
getXmpMetadata()	从 PDF 文档根目录中检索 XMP 数据
isEncrypted	显示 PDF 文件是否加密的只读布尔属性

PdfFileWriter

属性和方法	描述
addAttachment(fname,fdata)	在 PDF 中嵌入文件
addBlankPage(width= None,height=None)	追加一个空白页面到这个 PDF 文件并返回它
addJS(javascript)	添加将在打开此 PDF 是启动的 javascript
addLink(pagenum,pagedest,rect,border=None,fit='/fit',*args)	从一个矩形区域添加一个内部链接到指定的页面
addPage(page)	添加一个页面到这个PDF 文件，该页面通常从 PdfFileReader 实例获取
getNumpages()	页数
getPage(pageNumber)	从这个 PDF 文件中检索一个编号的页面
insertBlankPage(width=None,height=None,index=0)	插入一个空白页面到这个 PDF 文件并返回它，如果没有指定页面大小，就使用最后一页的大小
insertPage(page,index=0)	在这个 PDF 文件中插入一个页面，该页面通常从 PdfFileReader 实例获取
removeLinks()	从次数出中删除连接盒注释
removeText(ignoreByteStringObject = False)	从这个输出中删除图像
write(stream)	将添加到此对象的页面集合写入 PDF 文件

PageObject

```
PageObject(pdf=None,indirectRef=None)
```

此类表示 PDF 文件中的单个页面，通常这个对象是通过访问 PdfFileReader 对象的 **getPage()** 方法来得到的，也可以使用 **createBlankPage()** 静态方法创建一个空的页面。

参数:

- pdf : 页面所属的 PDF 文件。
- indirectRef：将源对象的原始间接引用存储在其源 PDF 中。

PageObject 对象的属性和方法

属性或方法	描述
createBlankPage(pdf=None,width=None,height=None)	返回一个新的空白页面（静态方法）
extractText()	找到所有文本绘图命令，按照他们在内容流中提供的顺序，并提取文本
getContents()	访问页面内容，返回 Contents 对象或 None
rotateClockwise(angle)	顺时针旋转指定度数
scale(sx,sy)	通过向其内容应用转换矩阵并更新页面大小
mergePage(page)	页面的合并
compressContentStreams()	页面压缩
mediaBox.upperRight	将页面裁剪成指定大小
mediaBox.getUpperRight*x()/mediaBox.getUpperRight*y()	获取页面的大小

2. reportlab

reportlab模块是用python语言生成pdf文件的模块。模块默认不支持中文，如果使用中文需要注册

1. **注册字体**

```
# 注册字体
from reportlab.pdfbase import pdfmetrics
# 字体类
from reportlab.pdfbase.ttfonts import TTFont
# 生成pdf
from reportlab.pdfgen import canvas

pdfmetrics.registerFont(TTFont("SimSun", "files/dd.ttf"))

# 2. 生成文字
# 创建空的pdf文件对象（默认大小为21cm*29.7cm）
c = canvas.Canvas("files/demo1.pdf")
# 设置字体
c.setFont('SimSun', 40)
# 设置文字颜色
c.setFillColorRGB(0.5, 0.5, 0.5, 0.4)
# 文字旋转
c.rotate(45)
# 渲染文字
c.drawString(10*cm, 20, text)
# c.drawString(200, 550, "你好,PDF文件")
c.showPage()
c.save()
```