安装

2020年6月11日 10:03

1、找到pip3.exe所在的文件夹，复制路径

我的路径是：C:\Users\孙艺航\AppData\Local\Programs\Python\Python37\Scripts

2、按Win+R,输入CMD确定

3、进入后，先输入cd 路径 回车

4、输入 pip3 install pypdf2 回车

5、输入 pip3 install pdfplumber 回车

6、输入pip3 install pymupdf 回车

前景回顾：with语句

2020年6月15日 20:08

Python基础课程PPT笔记184页 with语句

解析PDF文本及表格的几种库：

2020年6月15日 21:08

1. **pdfminer3k：主要用于读取 pdf 中的文本，代码太复杂**
2. **pdfminer 是pdfminer3k在Python2x时代的版本，对于表格的处理非常的不友好，能提取出文字，但是没有格式**
3. **tabula-py 是专门用来提取PDF表格数据的，同时支持PDF导出为CSV、Excel格式，但是这工具是用 java 写的，依赖 java7/8。tabula-py 就是对它做了一层 python 的封装，所以也依赖 java7/8。**
4. **pypdf2 网上代码比较多，但是读出来有时是乱码**
5. **pdfplumber 是按页来处理 pdf 的，可以获得页面的所有文字，并且提供的单独的方法用于提取表格，对于合并单元格等提取也存在问题。相比前面4个稍好一点。**

01.从PDF中提取文本

2020年6月15日 20:18

一、对其中一页提取

```
import pdfplumber
路径 = r'c:/文字.pdf'
with pdfplumber.open(路径) as pdf:
    首页 = pdf.pages[0]  # 指定页码
    文本 = 页码.extract_text()  # 提取文本
    文件 = open('c:/1.txt', mode='a')  # 新建文件，追加形式写入
    文件.write(文本)  # 将文本写入到文件
```

二、对所有页面提取

```
import pdfplumber
路径 = r'c:/文字.pdf'
with pdfplumber.open(路径) as pdf:
   for 页码 in pdf.pages:
      文本 = 页码.extract_text()
      文件 = open('c:/1.txt', mode='a')
      文件.write(文本)
```

02.从PDF中提取表格

2020年6月15日 20:55

一、保存成Csv文件

```
import pdfplumber
import pandas as pd
文件 = r'c:/表1.pdf'
with pdfplumber.open(文件) as pdf:
    for 页码 in pdf.pages:
        for 表格 in 页码.extract_tables():
            数据 = pd.DataFrame(表格[1:],columns=表格[0])
            数据.to_csv('c:/1.csv',mode='a',encoding='ANSI')
```

二、保存成Excel文件

```
import pdfplumber
import pandas as pd
a = r'c:/表1.pdf'   # 混合.pdf
count = 1
with pdfplumber.open(a) as pdf:
    with pd.ExcelWriter('c:/1.xlsx') as writer:
        for 页码 in pdf.pages:
            for 表格 in 页码.extract_tables():
                数据 = pd.DataFrame(表格[1:],columns=表格[0])
                数据.to_excel(writer,sheet_name=f'sheet{count}')
                count += 1
```

2.1 去除空行

2020年6月15日　　22:19

建议在VBA里解决，直接复制代码即可：


```
Sub 删除空行()
For Each ws In Sheets
    ws.Range("A1").EntireColumn.Delete
    lngFirstRow = ws.UsedRange.Row
    lngLastRow = lngFirstRow + ws.UsedRange.Rows.Count - 1
    For a = lngLastRow To lngFirstRow Step -1
        If Application.WorksheetFunction.CountA(ws.Rows(a)) = 0 Then
            ws.Rows(a).Delete
        End If
    Next
Next
End Sub
```

2.2 合并单元格

2020年6月16日 9:50

参考VBA笔记9.3

特别提示：导入模块注意大小写

2020年6月15日 19:50

import PyPDF2

03.拆分PDF

2020年6月16日 11:32

```
from PyPDF2 import PdfFileReader,PdfFileWriter # 读和写
路径 = r'c:/表格.pdf'
读PDF = PdfFileReader(路径)
for page in range(读PDF.getNumPages()): # getNumPages()获取总页数
    写pdf = PdfFileWriter()  # 实例化对象
    写pdf.addPage(读PDF.getPage(page)) # 将遍历出的每一页添加到实例化对象中
    with open(f'c:/{page+1}.pdf', "wb") as 变量名:
        写pdf.write(变量名)
```

04.合并PDF

2020年6月16日 12:02

```
from PyPDF2 import PdfFileReader, PdfFileWriter
写PDF = PdfFileWriter()
for page in range(1,4):
    读PDF = PdfFileReader(f'c:/{page}.pdf') # 循环读取每一个PDF
    for page in range(读PDF.getNumPages()): # 从读取到的PDF中遍历每一页
        写PDF.addPage(读PDF.getPage(page))  # 写入每一页
with open("c:/合并后的.pdf", "wb") as 变量名:
    写PDF.write(变量名)
```

05.旋转PDF

2020年6月16日 12:21

```
from PyPDF2 import PdfFileReader,PdfFileWriter
读PDF = PdfFileReader("c:/合并后的.pdf")
写PDF = PdfFileWriter()
page = 读PDF.getPage(0).rotateClockwise(90) # 将读取PDF的第1页，顺时针90度
写PDF.addPage(page) # 追加写入
page = 读PDF.getPage(1).rotateCounterClockwise(90) # 将读取PDF的第2页，逆时针90度
写PDF.addPage(page)
with open("c:/旋转后的.pdf", "wb") as 变量名:
    写PDF.write(变量名)
```

06.倒序排列PDF

2020年6月16日 12:21

```
from PyPDF2 import PdfFileReader, PdfFileWriter
读PDF = PdfFileReader("c:/合并后的.pdf")
写PDF = PdfFileWriter()
for page in range(读PDF.getNumPages()-1, -1, -1):
    # print(page)
    写PDF.addPage(读PDF.getPage(page))
with open("c:/倒序.pdf", "wb") as 变量名:
    写PDF.write(变量名)
```

练习1：

2020年6月16日 12:39

要求：

（1）打开文件"笔记.pdf"

（2）分割奇数页（第1、3、5...页）

（3）倒序保存页面

（4）生成“新.pdf”文件

关于range的知识：range作为一个生成器，主要是用来生成数值数据的

```
列表= list(range(10))
print(列表)
```

返回：[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

生成倒序数值数据的两种方式

```
列表1= [ i for i in range(10,-1,-1)]
#方法1
列表2 = list(range(10,-1,-1))
#方法2
print(列表1)
print(列表2)
```

按照特定的要求生成数值数据（比如偶数位数值输出）

```
列表3 = [ i for i in range(0,10,2)]
列表4 = list(range(0,10,2))
print(列表3)
print(列表4)

列表5 = [ i for i in range(10,0,-2)]
列表6 = list(range(10,0,-2))
print(列表5)
print(列表6)
# 返回：[10, 8, 6, 4, 2]

列表7 = [ i for i in range(8,-1,-2)]
列表8 = list(range(8,-1,-2))
print(列表7)
print(列表8)
# 返回：[8, 6, 4, 2, 0]
```

01.分割奇数页

2020年6月16日 12:56

```
from PyPDF2 import PdfFileReader, PdfFileWriter
读PDF = PdfFileReader("c:/笔记.pdf")
for page in range(0, 读PDF.getNumPages(), 2): # 这里的0代表第1页
    写PDF = PdfFileWriter()
    写PDF.addPage(读PDF.getPage(page))
    with open(f"c:/新笔记{page+1}.pdf", "wb") as 变量名:
        写PDF.write(变量名)
```

02.排充并汇总

2020年6月16日 12:56

```
from PyPDF2 import PdfFileReader, PdfFileWriter

读PDF = PdfFileReader("c:/笔记.pdf")
for page in range(0, 读PDF.getNumPages(), 2): # 这里的0代表第1页
    写PDF = PdfFileWriter()
    写PDF.addPage(读PDF.getPage(page))
    with open(f"c:/新笔记{page+1}.pdf", "wb") as 变量名:
        写PDF.write(变量名)

写PDF = PdfFileWriter()
for page in range(读PDF.getNumPages(), 0, -2):
    读PDF = PdfFileReader(f'c:/新笔记{page}.pdf')
    for page in range(读PDF.getNumPages() - 1, -1, -1):
        写PDF.addPage(读PDF.getPage(page))

with open("c:/1.pdf", "wb") as 变量名:
    写PDF.write(变量名)
```

07. 添加水印

2020年6月16日 13:07

```
from PyPDF2 import PdfFileReader,PdfFileWriter
from copy import copy  # 水印就是页面，复制页面的模块
读取 = PdfFileReader("c:/水印.pdf")
水印 = 读取.getPage(0) # 指定哪页是水印
读PDF = PdfFileReader("c:/笔记.pdf") # 读要添加水印的文件
写PDF = PdfFileWriter()  # 实例化对象
# 类似PS图层的概念，
for page in range(读PDF.getNumPages()):
    要加水印的每一页 = 读PDF.getPage(page)
    新页面 = copy(水印)
    新页面.mergePage(要加水印的每一页)  # 相当于图层合成，新页面里面加水印，水印就在文字下面的图层
    写PDF.addPage(新页面)  # 写入
with open("c:/带水印的笔记.pdf", "wb") as 变量名:
    写PDF.write(变量名)
```

怎么去除水印呢，学完Python的人工智能的opencv模块就会了

08.添加密码

2020年6月16日 13:42

```
from PyPDF2 import PdfFileReader,PdfFileWriter
读PDF = PdfFileReader("c:/带水印的笔记.pdf")
写PDF = PdfFileWriter()
for page in range(读PDF.getNumPages()):
    写PDF.addPage(读PDF.getPage(page))
写PDF.encrypt("1234")  # 这里设置密码
with open("c:/加密的.pdf", "wb") as 变量名:
    写PDF.write(变量名)
```

09.解除密码

2020年6月16日 14:06

```
from PyPDF2 import PdfFileReader,PdfFileWriter
读PDF = PdfFileReader("c:/加密的.pdf")
读PDF.decrypt("1234") # 输入正确密码解密
写PDF = PdfFileWriter()
for page in range(读PDF.getNumPages()):
    写PDF.addPage(读PDF.getPage(page))
with open("c:/解密的.pdf", "wb") as 变量名:
    写PDF.write(变量名)
```

练习2：

2020年6月16日 14:10

（1）打开笔记.pdf

（2）加水印

（3）加密码

（4）生成新的文件 新笔记.pdf

```
from PyPDF2 import PdfFileReader,PdfFileWriter
from copy import copy
读取 = PdfFileReader("c:/水印.pdf")
水印 = 读取.getPage(0)
读PDF = PdfFileReader("c:/笔记.pdf")
写PDF = PdfFileWriter()
for page in range(读PDF.getNumPages()):
    要加水印的每一页 = 读PDF.getPage(page)
    新页面 = copy(水印)
    新页面.mergePage(要加水印的每一页)
    写PDF.addPage(新页面)
写PDF.encrypt("1234")
with open("c:/新笔记.pdf", "wb") as 变量名:
    写PDF.write(变量名)
```

10.提取PDF中的图片

2020年6月16日 14:21

```
# 导入相关库
import fitz
import re
import os
# 使用正则表达式查找PDF中的图片
def find_imag(path, img_path): # PDF的路径和图片保存的路径
    checkXO = r"/Type(?= */XObject)"
    checkIM = r"/Subtype(?= */Image)"
    pdf = fitz.open(path)
    img_count = 0
    len_XREF = pdf._getXrefLength()
    print("文件名:{}, 页数: {}, 对象: {}".format(path, len(pdf), len_XREF - 1))
    for i in range(1, len_XREF):
        text = pdf._getXrefString(i)
        isXObject = re.search(checkXO, text)
        # 使用正则表达式查看是否是图片
        isImage = re.search(checkIM, text)
        # 如果不是对象也不是图片，则continue
        if not isXObject or not isImage:
            continue
        img_count += 1
        # 根据索引生成图像
        pix = fitz.Pixmap(pdf, i)
        new_name = path.replace('\\', '_') + "_img{}.png".format(img_count)
        new_name = new_name.replace(':', '')
        # 如果pix.n<5,可以直接存为PNG
        if pix.n < 5:
            pix.writePNG(os.path.join(img_path, new_name))
        else:
            pix0 = fitz.Pixmap(fitz.csRGB, pix)
            pix0.writePNG(os.path.join(img_path, new_name))
            pix0 = None
        pix = None
        print("提取了{}张图片".format(img_count))

if __name__ == '__main__':
    pdf_path = r'笔记.pdf'
    img_path = r'图片'
    if os.path.exists(img_path):
        print("文件夹已存在，请重新创建新文件夹！")
        raise SystemExit
    else:
        os.mkdir(img_path)
    m = find_imag(pdf_path, img_path)
```

注意：把PDF放到Py文件的目录下，图片目录也建在Py文件的目录下

11.批量转word 和 提取图片程序

2020年6月22日 22:26

1、找到pip3.exe所在的文件夹，复制路径

我的路径是： C:\Users\孙艺航\AppData\Local\Programs\Python\Python37\Scripts

2、按Win+R,输入CMD确定

3、进入后，先输入cd 路径 回车

4、输入 pip3 install pdfminer3k 回车

11.1 附送代码

2020年6月22日 22:27

```
from pdfminer.pdfparser import PDFParser, PDFDocument
from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
from pdfminer.layout import LAParams
from pdfminer.converter import PDFPageAggregator
from docx import Document
import warnings
import os
import time
import glob
import fitz
import re

J = True
while J:
    print('1.PDF_转_Word\n2.PDF图片提取\n3.退出系统\n')
    print("""温馨提示:选择功能之前请务必将你所要处理的文件与该程序放在同一文件夹中!\n""")
    choice = int(input('请输入数字选择对应的功能:'))
    if choice == 1:
        print('小可爱，你好！欢迎使用PDF_转_Word程序！')
        time.sleep(1.5)
        print(
            '——————————————————————————————————————Welcome to the program！——————————————————————————————————————')
        time.sleep(1.5)
        pdf_list = glob.glob('c:/pdf/*.pdf') # 查看同文件夹下的csv文件数
        print(u'共发现%s个pdf文件' % len(pdf_list))
        print(u'正在处理.....')
        print(pdf_list)
        for l in iter(pdf_list):
            file_name = os.open(l, os.O_RDWR)
            document = Document()
            warnings.filterwarnings("ignore")


            def pdf2word():
                fn = open(file_name, 'rb')
                parser = PDFParser(fn)
                doc = PDFDocument()
                parser.set_document(doc)
                doc.set_parser(parser)
                resource = PDFResourceManager()
                laparams = LAParams()
                device = PDFPageAggregator(resource, laparams=laparams)
                interpreter = PDFPageInterpreter(resource, device)
                for i in doc.get_pages():
                    interpreter.process_page(i)
                    layout = device.get_result()
                    for out in layout:
                        if hasattr(out, "get_text"):
                            content = out.get_text().replace(u'\xa0', u' ')
                            document.add_paragraph(
                                content, style='ListBullet'
                            )
                    document.save(l + '.docx')


            pdf2word()
        print('处理完成')
        break
    elif choice == 2:
        print('小可爱，你好！欢迎使用PDF图片提取系统！')
        time.sleep(1.5)
        print(
            '——————————————————————————————————————Welcome to the program！——————————————————————————————————————')
        time.sleep(1.5)


        def pdf2pic(path, pic_path):
            t0 = time.clock()  # 生成图片初始时间
            checkXO = r"/Type(?= */XObject)"  # 使用正则表达式来查找图片
            checkIM = r"/Subtype(?= */Image)"
            doc = fitz.open(path)  # 打开pdf文件
            imgcount = 0  # 图片计数
            lenXREF = doc._getXrefLength()  # 获取对象数量长度
            # 打印PDF的信息
            print("文件名:{}, 页数: {}, 对象: {}".format(path, len(doc), lenXREF - 1))
            # 遍历每一个对象
            for i in range(1, lenXREF):
                text = doc._getXrefString(i)  # 定义对象字符串
                isXObject = re.search(checkXO, text)  # 使用正则表达式查看是否是对象
                isImage = re.search(checkIM, text)  # 使用正则表达式查看是否是图片
                if not isXObject or not isImage:  # 如果不是对象也不是图片，则continue
                    continue
                imgcount += 1
                pix = fitz.Pixmap(doc, i)  # 生成图像对象
                new_name = "图片{}.png".format(imgcount)  # 生成图片的名称
                if pix.n < 5:  # 如果pix.n<5,可以直接存为PNG
                    pix.writePNG(os.path.join(pic_path, new_name))
                else:  # 否则先转换CMYK
                    pix0 = fitz.Pixmap(fitz.csRGB, pix)
                    pix0.writePNG(os.path.join(pic_path, new_name))
                    pix0 = None
                pix = None  # 释放资源
                t1 = time.clock()  # 图片完成时间
                print("运行时间:{}s".format(t1 - t0))
```

```
            print("提取了{}张图片".format(imgcount))


    if __name__ == '__main__':
        path = input('请输入需提取PDF文件路径:')
        pic_path = input('请输入提取图片保存路径:')
        # 创建保存图片的文件夹
        if os.path.exists(pic_path):
            print("文件夹已存在，不必重新创建！")
            pass
        else:
            os.mkdir(pic_path)
        pdf2pic(path, pic_path)
    break
elif choice == 3:
    print('拜拜~欢迎下次光临！')
    break
else:
    print('小伙子\小姐姐不要开车哦！我可不是傻子，请按照正确的流程输入！')
    print('——————————————————————————❁——————————————————————————❁——————————————————————————❁——————————————————————————')
    time.sleep(1.5)
    continue
```