Section 14

Excel, Word, and PDF Documents

Mohammed Asir Shahid

2021-08-05

Contents

1 Reading Excel Spreadsheets

The openpyxl module lets us modify Excel files using Python. It is a third party module that we'll need to install ourselves.

```
pip install openpyxl
```

The Excel document is called a workbook that is saved by .xlsx file extension. Each workbook contains sheets/worksheets. Inside each sheet there are columns (letters) and rows (numbers). The intersection of a column and row is called a cell.

```
import openpyxl,os

workbook=openpyxl.load_workbook("example.xlsx")

print(type(workbook))

print(workbook.get_sheet_names())
```

```
sheet=workbook.get_sheet_by_name("Sheet1")
print(type(sheet))

cell=sheet["A1"]

print(type(cell))
print(cell.value)

cell=sheet["B1"]

print(type(cell))
print(cell.value)

cell=sheet["C1"]

print(type(cell))
print(cell.value)


print(type(cell))
print(cell.value)

for i in range(1,8):
    print(i, sheet.cell(row=i, column=2).value)
```

```
<class 'openpyxl.workbook.workbook.Workbook'>
['Sheet1', 'Sheet2', 'Sheet3']
<class 'openpyxl.worksheet.worksheet.Worksheet'>
<class 'openpyxl.cell.cell.Cell'>
2015-04-05 13:34:02
<class 'openpyxl.cell.cell.Cell'>
Apples
<class 'openpyxl.cell.cell.Cell'>
73
<class 'openpyxl.cell.cell.Cell'>
73
1 Apples
2 Cherries
```

```
3 Pears
4 Oranges
5 Apples
6 Bananas
7 Strawberries
```

2 Editing Excel Spreadsheets

In the last lesson, we learned how to read .xlsx files. Now we will learn to create and modify them.

```
import openpyxl,os

wb=openpyxl.Workbook()

print(type(wb))

print(wb.get_sheet_names())

sheet=wb.get_sheet_by_name("Sheet")

print(sheet)
print(sheet["A1"].value)
sheet["A1"]=42
sheet["A2"]="Hello"
print(sheet["A1"].value)

wb.save("example1.xlsx")

sheet2=wb.create_sheet()

print(wb.get_sheet_names())

sheet2.title="My New Sheet Name"

print(wb.get_sheet_names())

wb.save("example2.xlsx")
```

```
wb.create_sheet(index=0, title="My Other Sheet")
# This changes the position of the new sheet

wb.save("example3.xlsx")
```

```
<class 'openpyxl.workbook.workbook.Workbook'>
['Sheet']
<Worksheet "Sheet">
None
42
['Sheet', 'Sheet1']
['Sheet', 'My New Sheet Name']
```

3 Reading and Editing PDFs

PDF files are binary files which make them far more complicated than plain text files such as .org or .py files. They store far more information than plain text files.

There are some Python modules we can use to interact with PDFs, however it isn't that straightforward. We will be looking at a third party module called PyPDF2.

```
pip install PyPDF2
```

```
import PyPDF2, os

pdfFile=open("meetingminutes1.pdf", "rb")
# The "rb" is since this is a binary file

PyPDF2.PdfFileReader(pdfFile)

reader=PyPDF2.PdfFileReader(pdfFile)

print(reader.numPages)

page=reader.getPage(0)

print(page.extractText())
```

```
#for pageNum in range(reader.numPages):
#    print(reader.getPage(pageNum).extractText())
```

```
19
OOFFFFIICCIIAALL  BBOOAARRDD  MMIINNUUTTEESS   Meeting of
March 7
, 2014

     The Board of Elementary and Secondary Education shall provide leadership and
create policies for education that expand opportunities for children, empower
families and communities, and advance Louisiana in an increasingly
competitive glob
al market.
 BOARD
 of ELEMENTARY
 and
 SECONDARY
 EDUCATION
```

Due to the complexity of PDF documents, Python can't add text arbitrarily. PDF Writer's functionality is limited to editing at the page level. So lets say we want to combine our two meeting minute fies

```
import PyPDF2, os

pdf1File=open("meetingminutes1.pdf", "rb")
pdf2File=open("meetingminutes2.pdf", "rb")
# The "rb" is since this is a binary file

reader1=PyPDF2.PdfFileReader(pdf1File)
reader2=PyPDF2.PdfFileReader(pdf2File)

writer=PyPDF2.PdfFileWriter()
```

```
for pageNum in range(reader1.numPages):
    page=reader1.getPage(pageNum)
    writer.addPage(page)

for pageNum in range(reader2.numPages):
    page=reader2.getPage(pageNum)
    writer.addPage(page)

outputFile=open("combinedminutes.pdf","wb")
writer.write(outputFile)
outputFile.close()
pdf1File.close()
pdf2File.close()
```