SPRINT 2

Cap13 – Working with PDF

Para trabalhar com PDF, no Python, é necessário fazer o download do pacote PyPDF2.

Extraindo texto de PDF

1. Download do PDF desejado
Ex: http://nostarch.com/automatestuff/

2.Após o download, deve-se digitar no interactive shell o seguinte:

```
>> import PyPDF2
>>> pdfFileObj = open('meetingminutes.pdf', 'rb')
 >>> pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
>>> pdfReader.numPages
  19
 >>> pageObj = pdfReader.getPage(0)
>>> pageObj.extractText()
  'OOFFFFIICCIIAALL BBOOAARRDD MMIINNUUTTEESS Meeting of March 7, 2015
  \n     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 global market. BOARD of ELEMENTARY and SECONDARY EDUCATION '
```

3. Importar o módulo do PyPDF2
   abrir meetingminutes.pdf em modo de leitura de binário
   guardar em pdfFileObj
   PyPDF2.PdfFileReader()
   passe pdfFileObj
   guarde o pdfFileReader em pdfReader

4. O numero total de páginas do pdf é guardado em numPages
   Para extrair texto de apenas uma página:
   Pode-se usar o getPage() método e passar a pagina desejada
   No caso, página 0
   pdfReader.getPage(0)
   Para retornar uma string do texto da página:
extractText()
Vale lembrar que a extração do texto nem sempre é perfeita

Decrypting PDF

Alguns PDFs podem exigir senhas para serem lidos.
No PDF que foi feito download a senha é rosebud

```
import PyPDF2
 pdfReader = PyPDF2.PdfFileReader(open('encrypted.pdf', 'rb'))
pdfReader.isEncrypted
  True
  pdfReader.getPage(0)
Traceback (most recent call last):
    File "<pyshell#173>", line 1, in <module>
      pdfReader.getPage()
    --snip--
    File "C:\Python34\lib\site-packages\PyPDF2\pdf.py", line 1173, in getObject
      raise utils.PdfReadError("file has not been decrypted")
  PyPDF2.utils.PdfReadError: file has not been decrypted
pdfReader.decrypt('rosebud')
  1
 pageObj = pdfReader.getPage(0)
```

Todo objeto PDF tem um atributo que é True se o PDF tem senha ou False se não tem.
Tentativas de chamar alguma função para ler o PDF antes de colocar a senha irá resultar em um erro.
Para ler basta chamar a função decrypt() e passar a senha como string.

Criando PDFs

o PdfFileWriter será usado para a criação de um novo arquivo, entretanto, o PyPDF2 apenas pode copiar páginas de outros PDFs, fazer rotação e sobrepor páginas, além de adicionar senhas ao arquivo.

Passos:
1- Abra um ou mais PDFs no PdfFileReader
2- Crie um novo PDF no PdfFileWriter
3-Copie páginas do PdfFileReader para o PdfFileWriter
4-Use o PdfFileWriter para escrever o arquivo final

Para realmente criar o arquivo em PDF é preciso acionar o PdfFileWriter’s write() method.

Copiando Páginas:

Faça o download de meetingminutes.pdf e meetingminutes2.pdf do link:
http://nostarch.com/automatestuff/ e em seguida digite o seguinte no interactive shell:

```
import PyPDF2
pdf1File = open('meetingminutes.pdf', 'rb')
pdf2File = open('meetingminutes2.pdf', 'rb')
pdf1Reader = PyPDF2.PdfFileReader(pdf1File)
pdf2Reader = PyPDF2.PdfFileReader(pdf2File)
pdfWriter = PyPDF2.PdfFileWriter()

 for pageNum in range(pdf1Reader.numPages):
 pageObj = pdf1Reader.getPage(pageNum)
 pdfWriter.addPage(pageObj)

for pageNum in range(pdf2Reader.numPages):
pageObj = pdf2Reader.getPage(pageNum)
pdfWriter.addPage(pageObj)

pdfOutputFile = open('combinedminutes.pdf', 'wb')
pdfWriter.write(pdfOutputFile)
pdfOutputFile.close()
pdf1File.close()
pdf2File.close()
```

Abra os dois arquivos em PDF no modo binary reade e salve os resultantes em pdf1File e pdf2File
acione PyPDF2.PdfFileReader() e passe meetingminutes.pdf
aciona de novo e passe meetingminutes2.pdf
agora crie um novo PdfFileWriter
copie todas as paginas dos PDFs iniciais e adiciona ao PdfFileWriter
após copiar as páginas, escreva um novo PDF nomeado combinedminutes.pdf passando o arquivo para PdfFileWriter’s write() method.

Rotando Páginas:

```
import PyPDF2
minutesFile = open('meetingminutes.pdf', 'rb')
pdfReader = PyPDF2.PdfFileReader(minutesFile)
page = pdfReader.getPage(0)
page.rotateClockwise(90)
  {'/Contents': [IndirectObject(961, 0), IndirectObject(962, 0),
  --snip--
  }
 pdfWriter = PyPDF2.PdfFileWriter()
 pdfWriter.addPage(page)
 resultPdfFile = open('rotatedPage.pdf', 'wb')
 pdfWriter.write(resultPdfFile)
 resultPdfFile.close()
 minutesFile.close()
```

No código acima usamos o getPage(0) para selecionar a primeira página
depois acionamos rotateClockwise(90)escrevemos um novo PDF com a página e salvamos em rotatedPage.pdf

---

Cap13P1

```
import PyPDF2

pdfFile = open('bruteForce.pdf', 'rb')
pdfReader = PyPDF2.PdfFileReader(pdfFile)
dictionaryFile = open('dictionary.txt')
passwordList = dictionaryFile.readlines()
for word in range(len(passwordList)):
   passWord = passwordList[word].strip()
   passWorkedUpper = pdfReader.decrypt(passWord.upper())
   if passWorkedUpper == 1:
      print('The password is: ' + passWord.upper())
      break
   else:
      print(passWord.upper() + ' did NOT work...')
   passWorkedLower = pdfReader.decrypt(passWord.lower())
   if passWorkedLower == 1:
      print('The password is: ' + passWord.lower())
      break
   else:
      print(passWord.lower() + ' did NOT work...')
dictionaryFile.close()
pdfFile.close()
```

Exercícios

1) A File object returned from open()

2) Read-binary ('rb') for PdfFileReader() and write-binary ('wb') for PdfFileWriter()

3) Calling getPage(4) will return a Page object for page 5, since page 0 is the first page

4) The numPages variable stores an integer of the number of pages in the PdfFileReader object.

5) Call decrypt('swordfish')

6) The rotateClockwise() and rotateCounterClockwise() methods. The degrees to rotate is passed as an integer argument.

7) docx.Document('demo.docx')

8) A document contains multiple paragraphs. A paragraph begins on a new line and contains multiple runs. Runs are contiguous groups of characters within a paragraph.

9) Use doc.paragraphs.

10) A Run object has these variables (*not* a Paragraph).

11) True always makes the Run object bolded and False makes it always not bolded, no matter what the style's bold setting is. None will make the Run object just use the style's bold setting.

12) Call the docx.Document() function.

13) doc.add_paragraph('Hello there!')

14) The integers 0, 1, 2, 3, and 4

Cap17 – Manipulating Images

Editar mais de uma imagem à mão pode ser um trabalho demorado e chato, O Pillow é um módulo Python de terceiros para interagir com arquivos de imagem. Com o poder de manipular imagens da mesma forma que você faria com software como o Microsoft Paint ou o Adobe Photoshop, o Python pode editar centenas ou milhares de imagens com facilidade.

Colors and RGBA Values Cores e valores RGBA

1. Editar mais de uma imagem à mão pode ser um trabalho demorado e chato, O Pillow é um módulo Python de terceiros para interagir com arquivos de imagem. Com o poder de manipular imagens da mesma forma que você faria com software como o Microsoft Paint ou o Adobe Photoshop, o Python pode editar centenas ou milhares de imagens com facilidade.
2. Baixar instale o módulo de Pillow
3. A instalação do Pillow tem algumas alterações. StackOverflow tem guias de instalação no Windows e Mac. Para Linux, execute sudo apt-get instale python3-tk, sudo apt-get instale python3-dev, então sudo pip3 instala Pillow.
4. Trabalhar com cores e coordenadas em Pillow.
5. Programas de computador geralmente representam uma cor em uma imagem como um valor RGBA (um grupo de números que especificam a quantidade de vermelho, verde, azul e alfa em uma cor) cada um desses valores de componente é um inteiro de 0 a 255, estes valores RGBA são atribuídos a pixels individuais. A configuração RGB de um pixel indica exatamente qual sombra de cor ela deve exibir.
6. Em Pillow a cor vermelha é representada por (255, 0, 0, 255), o verde é representado por (0, 255, 0, 255), eo azul é (0, 0, 255, 255). O branco, a combinação de todas as cores, é (255, 255, 255, 255), enquanto o preto, que não tem nenhuma cor, é (0, 0, 0, 255).
7 . O Pillow usa os nomes de cores padrão que o HTML usa.
8 . Travesseiro suporta um grande número de nomes de cores, de 'aliceblue' para 'whitesmoke'. Você pode encontrar a lista completa de mais de 100 nomes de cores padrão nos recursos em http://nostarch.com/automatestuff/.

Manipulating Images with Pillow and Working with the Image Data Type

1. Depois de ter o arquivo de imagem Zophie.png em seu diretório de trabalho atual, basta carregar a imagem do Zophie em Python, da seguinte forma:

```
 >>> de PIL importação Imagem
>>> catIm = Image.open ('zophie.png')
```

2. Se o arquivo de imagem não estiver no diretório de trabalho atual, altere o diretório de trabalho para a pasta que contém o arquivo de imagem chamando a função os.chdir ().

```
>>> importar os
>>> os.chdir ('C: \\ folder_with_image_file')
```

3. Verifique se o arquivo zophie.png está no diretório de trabalho atual para que a função Image.open () possa encontrá-lo.

4. O Pillow também fornece a função Image.new (), que retorna um objeto Image, bem como Image.open (), exceto que a imagem representada pelo objeto Image.new () ficará em branco. Os argumentos para Image.new () são os seguintes: A string 'RGBA', que define o modo de cor para RGBA. O tamanho, como uma tupla de dois inteiros da largura e altura da nova imagem.A cor de plano de fundo com a qual a imagem deve começar, como uma tupla de quatro inteiros de um valor RGBA.

5. Por exemplo, digite o seguinte no shell interativo:

```
>>> from PIL import Image
 >>> im = Image.new('RGBA', (100, 200), 'purple')
   >>> im.save('purpleImage.png')
 >>> im2 = Image.new('RGBA', (20, 20))
   >>> im2.save('transparentImage.png')
```

Foi criado um objeto Image para uma imagem com 100 pixels de largura e 200 pixels de altura, com um fundo roxo. Esta imagem é então gravada no arquivo purpleImage.png. Depois é chamado novamente Image.new () para criar outro objeto Image, passando este tempo (20, 20) para as dimensões e nada para a cor de fundo. Invisível preto, (0, 0, 0, 0), é a cor padrão usada se nenhum argumento de cor é especificado, então a segunda imagem tem um fundo transparente; foi gravado quadrado transparente de 20 × 20 em transparentImage.png

---

Cap17P1

```
import os
from PIL import Image

for foldername, subfolders, filenames in os.walk('c:\\'):
    numPhotoFiles = 0
    numNonPhotoFiles = 0
    for filename in filenames:
        if not filename.endswith('.png') or filename.endswith('.jpg'):
            numNonPhotoFiles += 1
            continue
        try: im = Image.open(os.path.join(foldername, filename))
        except: continue
        width, height = im.size
        if width >= 500 and height >= 500:
            numPhotoFiles += 1
        else:
            numNonPhotoFiles += 1
        if numPhotoFiles > numNonPhotoFiles:
            print(foldername)
```

Exercícios

1) An RGBA value is a tuple of 4 integers, each ranging from 0 to 255. The four integers correspond to the amount of red, green, blue, and alpha (transparency) in the color.

2) A function call to ImageColor.getcolor('CornflowerBlue', 'RGBA') will return (100, 149, 237, 255), the RGBA value for that color.

3) A box tuple is a tuple value of four integers: the left edge x-coordinate, the top edge y-coordinate, the width, and the height, respectively.

4) Image.open('zophie.png')

5) imageObj.size is a tuple of two integers, the width and the height.

6) imageObj.crop((0, 50, 50, 50)). Notice that you are passing a box tuple to crop(), not four separate integer arguments.

7) Call the imageObj.save('new_filename.png') method of the Image object.

8) The ImageDraw module contains code to draw on images.

9) ImageDraw objects have shape-drawing methods such as point(), line(), or rectangle(). They are returned by passing the Image object to the ImageDraw.Draw() function.

Cap9 – Organizing Files

Cap9P1

```
import os

def deleteUnneeded(folder):
   folder = os.path.abspath(folder)
   for foldername, subfolders, filenames in os.walk(folder):
      for filename in filenames:
         fileSize = os.path.getsize(foldername + '/' + filename)
         if int(fileSize) < 10000000:
            continue
         print('Deleting ' + filename + '...') #Print only to verify working correctly
deleteUnneeded('/Users/username/Documents')
print('Done')
```

Cap9P2

```
import os
import shutil
def selectiveCopy(folder):
   folder = os.path.abspath(folder)
   for foldername, subfolders, filenames in os.walk(folder):
      for filename in filenames:
         if not filename.endswith('.pdf'):
            continue
         print('Copying ' + filename + '...') #Print only to verify working correctly
selectiveCopy(r'C:\Users\username\pdffolder')
print('Done')
```

Exercícios

1) The shutil.copy() function will copy a single file, while shutil.copytree() will copy an entire folder, along with all its contents.

2) The shutil.move() function is used for renaming files, as well as moving them.

3) The send2trash functions will move a file or folder to the recycle bin, while shutil functions will permanently delete files and folders.

4.)The zipfile.ZipFile() function is equivalent to the open() function; the first argument is the filename, and the second argument is the mode to open the ZIP file in (read, write, or append).

Cap12P1

```
import openpyxl

wb = openpyxl.load_workbook('produceSales.xlsx')
sheet = wb.get_active_sheet()
for columnNum in range(1, sheet.get_highest_column()):
   spreadText = open('Column ' + str(columnNum) +'.txt', 'a')
   for rowNum in range(2, sheet.get_highest_row()):
      rowData = sheet.cell(row=rowNum, column=columnNum).value
      spreadText.write(str(rowData) + '\n')
   spreadText.close()
```

Exercícios

1) The openpyxl.load_workbook() function returns a Workbook object.

2) The get_sheet_names() method returns a Worksheet object.

3) Call wb.get_sheet_by_name('Sheet1').

4) Call wb.get_active_sheet().

5) sheet['C5'].value or sheet.cell(row=5, column=3).value

6) sheet['C5'] = 'Hello' or sheet.cell(row=5, column=3).value = 'Hello'

7) cell.row and cell.column

8) They return the highest column and row with values in the sheet, respectively, as integer values.

9) openpyxl.cell.column_index_from_string('M')

10) openpyxl.cell.get_column_letter(14)

11) sheet['A1':'F1']

12) wb.save('example.xlsx')

13) A formula is set the same way as any value. Set the cell's value attribute to a string of the formula text. Remember that formulas begin with the = sign.

14) When calling load_workbook(), pass True for the data_only keyword argument.

15) sheet.row_dimensions[5].height = 100

16) sheet.column_dimensions['C'].hidden = True

17) OpenPyXL 2.0.5 does not load freeze panes, print titles, images, or charts.

18) Freeze panes are rows and columns that will always appear on the screen. They are useful for headers.

19) openpyxl.charts.Reference(), openpyxl.charts.Series(), openpyxl.charts. BarChart(), chartObj.append(seriesObj), and add_chart()
