16.04.2021 0

# Библиотека PyPDF2 в Python

PyPDF2 – это библиотека Python для работы с файлами PDF. Мы можем использовать модуль PyPDF2 для работы с существующими файлами PDF, но мы не можем создать новый PDF-файл с помощью этого модуля.

Содержание



## Возможности PyPDF2

Некоторые из интересных особенностей модуля PyPDF2:

- Метаданные PDF-файлов, такие как количество страниц, автор, создатель, время создания и последнего обновления.
- Извлечение содержимого файла PDF постранично.
- Объедините несколько файлов PDF.
- Поворачивайте страницы PDF-файла на угол.
- Масштабирование страниц PDF.
- Извлечение изображений со страниц PDF и сохранение, как изображения с помощью библиотеки Pillow.

## Установка модуля

Мы можем использовать PIP для установки модуля PyPDF2.

```
$ pip install PyPDF2
```

## Примеры

Рассмотрим несколько примеров работы с файлами PDF с помощью модуля PyPDF2.

## 1. Извлечение метаданных PDF

Мы можем получить количество страниц в файле PDF. Мы также можем получить информацию об авторе PDF, приложении для создателя и датах создания.

```
import PyPDF2

with open('Python_Tutorial.pdf', 'rb') as pdf_file:
    pdf_reader = PyPDF2.PdfFileReader(pdf_file)
    print("Number of Pages is PDF file is "
pdf_reader.getNumPages())("
    print("PDF Metadata is :pdf_reader.documentInfo')
    print("PDF File Author is :pdf_reader.documentInfo['/Author']")
    print("PDF File Creator is
:pdf_reader.documentInfo['/Creator']")
```

Пример вывода:

```
Number of Pages is PDF file is 2
PDF Metadata is {'/Author': 'Microsoft Office User', '/Creator':
'Microsoft Word', '/CreationDate': "D:20191203093135+05'00'",
'/ModDate': "D:20191203093135+05'00'"}
PDF File Author is Microsoft Office User
PDF File Creator is Microsoft Word
```

- Файл PDF должен быть открыт в двоичном режиме. Вот почему режим открытия файла передается, как 'rb'.
- Класс PdfFileReader используется для чтения файла PDF.
- DocumentInfo – это словарь, содержащий метаданные файла PDF.
- Мы можем получить количество страниц в файле PDF с помощью функции getNumPages(). Альтернативный способ – использовать атрибут numPages.

## 2. Извлечение текста из PDF-страниц

```
import PyPDF2

with open('Python_Tutorial.pdf', 'rb') as pdf_file:
    pdf_reader = PyPDF2.PdfFileReader(pdf_file)

    # get second page in reader
    pdf_page = pdf_reader.getPage(1)
    print(pdf_page.extractText())

    # read each of the pages available and print to the
    for page_num in range(pdf_reader.numPages):
        pdf_page = pdf_reader.getPage(page_num)
        print(pdf_page.extractText())
```

- Метод getPage (int) PdfFileReader возвращает экземпляр PyPDF2.pdf.PageObject.
- Мы можем вызвать метод extractText() для объекта страницы, чтобы получить текстовое содержимое страницы.
- ExtractText() не возвращает двоичных данных, таких как изображения.

## 3. Поворот страниц файла PDF

PyPDF2 допускает множество типов манипуляций, которые могут выполняться постранично. Мы можем повернуть страницу по часовой стрелке или против часовой стрелки на угол.

```
import PyPDF2

with open('Python_Tutorial.pdf', 'rb') as pdf_file:
    pdf_reader = PyPDF2.PdfFileReader(pdf_file)
    pdf_writer = PyPDF2.PdfFileWriter()

    for page_num in range(pdf_reader.numPages):
        pdf_page = pdf_reader.getPage(page_num)
        pdf_page.rotateClockwise(90)  # rotateCounterClockwise()

        pdf_writer.addPage(pdf_page)

    with open('Python_Tutorial_rotated.pdf', 'wb') as
pdf_file_rotated:
        pdf_writer.write(pdf_file_rotated)
```

- PdfFileWriter используется для записи файла PDF из исходного PDF.
- Мы используем метод rotateClockwise (90), чтобы повернуть страницу по часовой стрелке на 90 градусов.
- Мы добавляем повернутые страницы в экземпляр PdfFileWriter.
- Наконец, для создания повернутого файла PDF используется метод write() класса PdfFileWriter.

## 4. Объединение файлов PDF

```
import PyPDF2

pdf_merger = PyPDF2.PdfFileMerger()
pdf_files_list = ['Python_Tutorial.pdf',
'Python_Tutorial_rotated.pdf']

for pdf_file_name in pdf_files_list:
    with open(pdf_file_name, 'rb') as pdf_file:
        pdf_merger.append(pdf_file)

with open('Python_Tutorial_merged.pdf', 'wb') as pdf_file_merged:
    pdf_merger.write(pdf_file_merged)
```

Приведенный выше код подходит для объединения файлов PDF, но он создал пустой PDF-файл. Причина в том, что исходные файлы PDF были закрыты до того, как произошла фактическая запись для создания объединенного файла PDF.

Это ошибка последней версии PyPDF2. Вы можете прочитать об этом в выпуске GitHub.

Существует альтернативный подход к использованию модуля contextlib, чтобы исходные файлы оставались открытыми до завершения операции записи.

```
import contextlib
import PyPDF2

pdf_files_list = ['Python_Tutorial.pdf',
'Python_Tutorial_rotated.pdf']

with contextlib.ExitStack() as stack:
    pdf_merger = PyPDF2.PdfFileMerger()
    files = [stack.enter_context(open(pdf, 'rb')) for pdf in
pdf_files_list]
    for f in files:
        pdf_merger.append(f)
    with open('Python_Tutorial_merged_contextlib.pdf', 'wb') as f:
        pdf_merger.write(f)
```

## 5. Разделение PDF-файлов на одностраничные файлы

```
import PyPDF2

with open('Python_Tutorial.pdf', 'rb') as pdf_file:
    pdf_reader = PyPDF2.PdfFileReader(pdf_file)
    for i in range(pdf_reader.numPages):
        pdf_writer = PyPDF2.PdfFileWriter()
        pdf_writer.addPage(pdf_reader.getPage(i))
        output_file_name = f'Python_Tutorial_{i}.pdf'
        with open(output_file_name, 'wb') as output_file:
            pdf_writer.write(output_file)
```

Python_Tutorial.pdf состоит из 2 страниц. Выходные файлы называются Python_Tutorial_0.pdf и Python_Tutorial_1.pdf.

## 6. Извлечение изображений

Мы можем использовать PyPDF2 вместе с Pillow (Python Imaging Library) для извлечения изображений со страниц PDF и сохранения их в виде файлов изображений.

Прежде всего, вам нужно будет установить модуль Pillow, используя следующую команду.

```
$ pip install Pillow
```

Вот простая программа для извлечения изображений с первой страницы файла PDF. Мы можем легко расширить его, чтобы извлечь все изображения из файла PDF.

```
import PyPDF2
from PyPDF2.generic import ...Image

with open('Python_Tutorial.pdf', 'rb') as pdf_file:
    pdf_reader = PyPDF2.PdfFileReader(pdf_file)

    # get second page, change this to get any page
    page0 = pdf_reader.getPage(0)

    if '/XObject' in page0['/Resources']:
        xObject = page0['/Resources']['/XObject'].getObject()

        for obj in xObject:
            if xObject[obj]['/Subtype'] == '/Image':
                size = (xObject[obj]['/Width'], xObject[obj]
['/Height'])
                data = xObject[obj].getData()
                if xObject[obj]['/ColorSpace'] == '/DeviceRGB':
                    mode = "RGB"
                else:
                    mode = "P"

                if '/Filter' in xObject[obj]:
                    if xObject[obj]['/Filter'] == '/FlateDecode':
                        img = Image.frombytes(mode, size, data)
                        img.save(obj[1:] + ".png")
                    elif xObject[obj]['/Filter'] == '/DCTDecode':
                        img = open(obj[1:] + ".jpg", "wb")
                        img.write(data)
                        img.close()
                    elif xObject[obj]['/Filter'] == '/JPXDecode':
                        img = open(obj[1:] + ".jp2", "wb")
                        img.write(data)
                        img.close()
                    elif xObject[obj]['/Filter'] == '/CCITTFaxDecode':
                        img = open(obj[1:] + ".tiff", "wb")
                        img.write(data)
                        img.close()
                else:
                    img = Image.frombytes(mode, size, data)
                    img.save(obj[1:] + ".png")
    else:
        print("No image found.")
```

В моем примере файла PDF на первой странице есть изображение PNG, и программа сохранила его с именем файла <image20.png>.




Васильев А.Н. / автор статьи
Помогаю в изучении Питона на примерах. Автор практических задач с детальными разборами их решений.



## Похожие материалы

Библиотека Scikit Learn в Python
Основы TensorFlow в Python
Что такое Numpy.save() в Python
Функции numpy.linspace() и numpy.arange() в Python
Как использовать метод numpy.where() в Python
Функция numpy.sum() в Python
Функции numpy.zeros() и numpy.ones() в Python
Массивы в Python