PDF Processing in Python

PDF processing in Python involves extracting content from PDFs, creating and manipulating PDF files, and handling advanced PDF features like forms and encryption. Several libraries can help with these tasks, each offering different capabilities and features.

**Extracting Text from PDFs**

**Using PyPDF2 for Text Extraction**

**PyPDF2** is a popular library for manipulating PDFs in Python, including text extraction. It is straightforward but may struggle with complex PDF layouts.

* **Basic Text Extraction**:

**Example**:

import PyPDF2

# Open a PDF file

with open('example.pdf', 'rb') as file:

reader = PyPDF2.PdfFileReader(file)

num\_pages = reader.numPages

# Extract text from each page

for page\_num in range(num\_pages):

page = reader.getPage(page\_num)

text = page.extract\_text()

print(f"Page {page\_num+1} text:\n{text}")

* **Handling Complex PDF Structures**:

PyPDF2 can struggle with PDFs that have complex layouts or non-standard encoding. For such cases, combining PyPDF2 with other libraries or using more advanced tools might be necessary.

**Using pdfminer.six for Detailed Extraction**

**pdfminer.six** is a more advanced library that provides detailed text extraction and can handle complex PDF structures better than PyPDF2.

* **Extracting Text and Metadata**:

**Example**:

from pdfminer.high\_level import extract\_text

from pdfminer.pdfparser import PDFParser

from pdfminer.pdfdocument import PDFDocument

from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter

from pdfminer.layout import LAParams

from pdfminer.pdfpage import PDFPage

# Extract text from a PDF

text = extract\_text('example.pdf')

print(text)

# Extract metadata

with open('example.pdf', 'rb') as file:

parser = PDFParser(file)

document = PDFDocument(parser)

metadata = document.info[0]

print(metadata)

* **Handling Encrypted PDFs**:

**Example**:

from pdfminer.pdfparser import PDFParser

from pdfminer.pdfdocument import PDFDocument

from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter

from pdfminer.layout import LAParams

from pdfminer.pdfpage import PDFPage

# Handle encrypted PDF

password = 'your\_password'

with open('encrypted.pdf', 'rb') as file:

parser = PDFParser(file)

document = PDFDocument(parser, password)

if not document.is\_encrypted:

print("Document is not encrypted.")

else:

print("Document is encrypted.")

**Creating and Manipulating PDFs**

**Creating PDFs with ReportLab**

**ReportLab** is a powerful library for creating and manipulating PDFs from scratch.

* **Basic PDF Creation**:

**Example**:

from reportlab.lib.pagesizes import letter

from reportlab.pdfgen import canvas

# Create a PDF file

c = canvas.Canvas("example.pdf", pagesize=letter)

width, height = letter

# Add text

c.drawString(100, height - 100, "Hello, PDF World!")

# Save the PDF

c.save()

* **Adding Text, Images, and Graphics**:

**Example**:

from reportlab.lib.pagesizes import letter

from reportlab.pdfgen import canvas

from reportlab.lib.units import inch

c = canvas.Canvas("example\_with\_image.pdf", pagesize=letter)

width, height = letter

# Add text

c.drawString(100, height - 100, "Hello, PDF World!")

# Add an image

c.drawImage("example\_image.png", 100, height - 300, width=2\*inch, height=2\*inch)

# Draw a rectangle

c.rect(50, height - 350, width-100, height-200, stroke=1, fill=0)

c.save()

**Manipulating PDFs with PyPDF2**

**PyPDF2** can also be used to manipulate existing PDFs, such as merging or splitting them.

* **Merging and Splitting PDFs**:

**Example**:

import PyPDF2

# Merge PDFs

pdf\_writer = PyPDF2.PdfFileWriter()

for pdf in ['file1.pdf', 'file2.pdf']:

pdf\_reader = PyPDF2.PdfFileReader(pdf)

for page in range(pdf\_reader.numPages):

pdf\_writer.addPage(pdf\_reader.getPage(page))

with open('merged.pdf', 'wb') as output\_file:

pdf\_writer.write(output\_file)

# Split PDFs

pdf\_reader = PyPDF2.PdfFileReader('example.pdf')

pdf\_writer = PyPDF2.PdfFileWriter()

pdf\_writer.addPage(pdf\_reader.getPage(0)) # Add only the first page

with open('split\_page.pdf', 'wb') as output\_file:

pdf\_writer.write(output\_file)

* **Adding Annotations and Forms**:

**Example**:

import PyPDF2

# Load PDF and prepare to write

pdf\_reader = PyPDF2.PdfFileReader('example.pdf')

pdf\_writer = PyPDF2.PdfFileWriter()

# Add annotations (e.g., watermarks)

watermark = PyPDF2.PdfFileReader('watermark.pdf').getPage(0)

for page\_num in range(pdf\_reader.numPages):

page = pdf\_reader.getPage(page\_num)

page.merge\_page(watermark)

pdf\_writer.addPage(page)

with open('annotated.pdf', 'wb') as output\_file:

pdf\_writer.write(output\_file)

**Advanced PDF Processing**

**Working with PDF Forms**

* **Reading and Filling PDF Forms**:

**Example**:

from PyPDF2 import PdfFileReader, PdfFileWriter

# Read a PDF form

pdf\_reader = PdfFileReader('form.pdf')

pdf\_writer = PdfFileWriter()

# Fill the form

page = pdf\_reader.getPage(0)

page.updatePageFormFieldValues({'name': 'John Doe', 'email': 'john.doe@example.com'})

pdf\_writer.addPage(page)

with open('filled\_form.pdf', 'wb') as output\_file:

pdf\_writer.write(output\_file)

**Encrypting and Decrypting PDFs**

* **Adding Encryption**:

**Example**:

from PyPDF2 import PdfFileReader, PdfFileWriter

pdf\_reader = PdfFileReader('example.pdf')

pdf\_writer = PdfFileWriter()

# Add encryption

pdf\_writer.add\_page(pdf\_reader.getPage(0))

pdf\_writer.encrypt('password123')

with open('encrypted.pdf', 'wb') as output\_file:

pdf\_writer.write(output\_file)

* **Decrypting Protected PDFs**:

**Example**:

from PyPDF2 import PdfFileReader

pdf\_reader = PdfFileReader('encrypted.pdf')

pdf\_reader.decrypt('password123')

with open('decrypted.pdf', 'wb') as output\_file:

pdf\_writer = PdfFileWriter()

pdf\_writer.addPage(pdf\_reader.getPage(0))

pdf\_writer.write(output\_file)

**Summary**:

* **Text Extraction**: Use PyPDF2 for basic tasks and pdfminer.six for more detailed extraction.
* **PDF Creation and Manipulation**: Use ReportLab for creating PDFs and PyPDF2 for manipulating them.
* **Advanced Processing**: Includes working with PDF forms and handling encryption/decryption.