**ADITYA AMIN** **ASSIGN : 12**

1. In what modes should the PdfFileReader() and PdfFileWriter() File objects will be opened?

PdfFileReader() needs to be opened in read-binary mode by passing 'rb' as the second argument to open(). Likewise, the File object passed to PyPDF2. PdfFileWriter() needs to be opened in write-binary mode with 'wb'.

1. From a PdfFileReader object, how do you get a Page object for page 5?

from PyPDF2 import PdfReader

# Create a PdfReader object

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

# Get the Page object for page 5 (page numbers are zero-indexed)

page\_number = 4 # page 5 is at index 4

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

# You can now access various properties and methods of the Page object

print("Page Number:", page\_number + 1) # Add 1 to page\_number to get the actual page number

print("Number of Objects on Page:", page['/Count'])

print("Page Media Box:", page.mediaBox)

1. What PdfFileReader variable stores the number of pages in the PDF document?

In PyPDF2, which is a popular Python library for reading and manipulating PDF files, you can use the numPages attribute of a PdfFileReader object to get the number of pages in a PDF document.

1. If a PdfFileReader object’s PDF is encrypted with the password swordfish, what must you do before you can obtain Page objects from it?

If a PdfFileReader object's PDF is encrypted with a password (e.g., "swordfish"), you need to provide the correct password before you can obtain Page objects from it. To do so in PyPDF2, you can use the decrypt() method of the PdfFileReader object to provide the password and decrypt the PDF file.

1. What methods do you use to rotate a page?

In PyPDF2, which is a popular Python library for reading and manipulating PDF files, you can use the rotateClockwise() and rotateCounterClockwise() methods of the Page object to rotate a page in a PDF document. These methods allow you to rotate a page clockwise or counterclockwise by a specified angle.

1. What is the difference between a Run object and a Paragraph object?

a "Run" object is used to represent a contiguous run of text with the same formatting within a "Paragraph" object, while a "Paragraph" object represents a single paragraph of text within a document and can contain multiple "Run" objects with different formatting.

1. How do you obtain a list of Paragraph objects for a Document object that’s stored in a variable named doc?

from docx import Document

# Load the document from a file or create a new Document object

doc = Document('example.docx')

# Access the list of Paragraph objects in the Document object

paragraphs = doc.paragraphs

# Iterate through the list of Paragraph objects and perform desired operations

for paragraph in paragraphs:

# Access the text content of the paragraph

print(paragraph.text)

# Access other properties of the paragraph, such as style, indentation, etc.

print(paragraph.style.name)

print(paragraph.left\_indent)

print(paragraph.right\_indent)

print(paragraph.first\_line\_indent)

# ... other properties

# Perform desired operations on the paragraph

# ...

1. What type of object has bold, underline, italic, strike, and outline variables?

The bold, underline, italic, strike, and outline are properties or attributes that can be found in a Font object in Python libraries such as python-docx for working with Microsoft Word documents.

1. What is the difference between False, True, and None for the bold variable?

False: When bold is set to False, it means that the text should not be formatted as bold.

True: When bold is set to True, it means that the text should be formatted as bold.

None: When bold is set to None, it means that the text should inherit the bold formatting from the style applied to the text.

1. How do you create a Document object for a new Word document?

from docx import Document

# Create a new Document object

doc = Document()

# Add a paragraph with some text

paragraph = doc.add\_paragraph("Hello, world!")

# Save the document to a file

doc.save("my\_document.docx")

1. How do you add a paragraph with the text 'Hello, there!' to a Document object stored in a variable named doc?

from docx import Document

# Create a Document object

doc = Document()

# Add a paragraph with the text 'Hello, there!'

paragraph = doc.add\_paragraph('Hello, there!')

1. What integers represent the levels of headings available in Word documents?

Here is the mapping of integer values to the corresponding heading levels in Word documents:

Heading 1: Level 1

Heading 2: Level 2

Heading 3: Level 3

Heading 4: Level 4

Heading 5: Level 5

Heading 6: Level 6

Heading 7: Level 7

Heading 8: Level 8

Heading 9: Level 9

These heading levels can be used to apply different styles, formatting, and numbering to different levels of headings in a Word document, allowing for consistent and organized document structure.