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

Ans1

The PdfFileReader() and PdfFileWriter() File objects in PyPDF2 should be opened in binary mode using the 'rb' and 'wb' modes

PdfFileReader() is used to read an existing PDF file and should be opened in binary read mode 'rb'.

PdfFileWriter() is used to create a new PDF file or modify an existing one, and should be opened in binary write mode 'wb'.

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

Ans2

Use the getPage() method of the PdfFileReader object, passing in the index of the page.

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

Ans3

In PyPDF2, the number of pages in a PDF document can be obtained from the numPages attribute of a PdfFileReader object.

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

Ans4

If a PdfFileReader object's PDF is encrypted with the password "swordfish" must provide the password to the PdfFileReader object before obtain Page objects from it.

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

Ans5

rotateClockwise() and rotateCounterClockwise() methods do not modify the original Page object. Instead, they return a new Page object that has been rotated.

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

Ans6

A Paragraph object contains one or more Run objects. When you add text to a Paragraph object, the library creates a new Run object to represent that text. If change in any of the character properties in the middle of a paragraph, the library will create a new Run object to represent the text with the new properties.

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

Ans7

Use the paragraphs attribute of the Document object. This attribute returns a list-like object that contains one Paragraph object for each paragraph in the document.

from docx import Document

doc = Document('example.docx')

paragraphs = doc.paragraphs

for paragraph in paragraphs:

print(paragraph.text)

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

Ans8

bold, underline, italic, strike, and outline attribute of a Paragraph object in a python-docx library.

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

Ans9

In python-docx, the bold variable for a Run object can be set to three different values: True, False, or None.

If bold is set to True, the text in the Run object will be formatted as bold.

If bold is set to False, the text in the Run object will not be formatted as bold.

If bold is set to None, the text in the Run object will inherit the bold formatting of the surrounding text. If the surrounding text is bold, the Run object will be bold.

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

Ans10

from docx import Document

from docx.shared import Inches

doc = Document()

doc.add\_heading('Document Title', 0)

doc.add\_paragraph('This is the first paragraph.')

doc.add\_picture('image.png', width=Inches(1.25))

doc.add\_paragraph('This is the second paragraph.')

doc.save('example.docx')

11. 11. How do you add a paragraph with the text ‘Hello, there!’ to a Document object stored in a

variable named doc?

Ans11

from docx import Document

doc = Document()

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

doc.save('example.docx')

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

Ans12

In Word documents, the following integers represent the levels of headings

Level 1: wdHeadingLevel1 - integer value of 1

Level 2: wdHeadingLevel2 - integer value of 2

Level 3: wdHeadingLevel3 - integer value of 3

Level 4: wdHeadingLevel4 - integer value of 4

Level 5: wdHeadingLevel5 - integer value of 5

Level 6: wdHeadingLevel6 - integer value of 6

Level 7: wdHeadingLevel7 - integer value of 7

Level 8: wdHeadingLevel8 - integer value of 8

Level 9: wdHeadingLevel9 - integer value of 9