1. For PdfFileReader () file objects should be opened in rb -> read binary mode, Whereas for PdfFileWriter () file objects should be opened in wb -> write binary mode.
2. PdfFileReader class provides a method called get Page (page no) to get a page object.
3. getNum Pages () method of PdfFileReader class stores the no pages in a PDF document
4. If a PdfFileReader object’s PDF is encrypted with the password swordfish and you're not aware of it. first read the Pdf using the PdfFileReader Class. PdfFileReader class provides a attribute called is Encrypted to check whether a pdf is encrypted or not. the method returns true if a pdf is encrypted and vice versa.

if pdf is encrypted use the decrypt () method provided by PdfFileReader class first then try to read the contents/pages of the pdf, else PyPDF2 will raise the following error PyPDF2.utils. PdfReadError: file has not been decrypted

1. PyPDF2 Package provides 2 methods to rotate a page:

1.rotate Clockwise () -> For Clockwise rotation

2.rotateCounterClockWise () -> For Counter Clockwise rotation

The PyPDF2 package only allows you to rotate a page in increments of 90 degrees. You will receive an AssertionError otherwise.

1. The structure of a document is represented by three different data types in python-Docx. At the highest level, a Document object represents the entire document. The Document object contains a list of Paragraph objects for the paragraphs in the document. (A new paragraph begins whenever the user

presses ENTER or RETURN while typing in a Word document.) Each of these Paragraph objects contains a list of one or more Run objects.

The text in a Word document is more than just a string. It has font, size, colour, and other styling information associated with it. A style in Word is a collection of these attributes. A Run object is a contiguous run of text with the same style. A new Run object is needed whenever the text style changes.

1. # Example Program

from docx import Document

doc = Document("sample\_file.docx") # Path of the Docx file

print (doc. paragraphs) # Prints the list of Paragraph objects for a Document

for paragraph in doc. paragraphs:

print (paragraph. Text) # Prints the text in the paragraph

1. Run object has bold, underline, italic, strike, and outline variables. The text in a Word document is more than just a string. It has font, size, colour, and other styling information associated with it.

A style in Word is a collection of these attributes. A Run object is a contiguous run of text with the same style. A new Run object is needed whenever the text style changes.

1. bold = True # Style Set to Bold

bold = False # Style Not Set to Bold

bold = None # Style is Not Applicable

1. # Example Program

from docx import Document

document = Document ()

document.add\_paragraph ("I Neuron Full Stack Data Science Course")

document. Save('mydocument.docx')

1. # Example Program

from docx import Document

doc = Document ()

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

doc. saves('hello.docx')

1. The levels for a heading in a word document can be specified by using the level attribute inside the add heading method. There are a total of 5 levels starting for 0 t0 4. where level 0 makes a headline with the horizontal line below the text, whereas the heading level 1 is the main heading. Similarly, the other headings are sub-heading with their font-sizes in decreasing order.