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

Answer:

PdfFileReader() – will be opened in ‘rb’ read binary mode.

PdfFileWriter() – will be opened in ‘wb’ write binary mode

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

**pdfFileReader.getPage(4).extractText()** – gives the content in page 5 (since the index starts at 0) and returns the value.

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

pdfFileReader.numPages – returns the total number of pages in the PDF file

--------------------------------------------------------------------------------------------------------------------------------------

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?

Answer:

We must decrypt the file before obtaining the Page objects from it.

For example,

**password = “swordfish”**

**pdfFileReader.decrypt(password)**

#This will unlock the PDF file and makes it ready to be read.

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

The **pdfFileReader .rotateClockwise()** and **pdfFileReader rotateCounterClockwise()** are used to rotate the PDF clockwise and anti-clockwise direction respectively. The degree of rotation is given as an integer argument.

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

Run object: It is a contiguous run of string/text with the same style. A new Run object is needed whenever the text style changes.

Paragraph object: It is a list of Run objects. A new paragraph begins whenever the user presses ENTER or RETURN while typing a document.

Note: A list of paragraph represents a Document object.

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

**doc.paragraphs** can be used to obtains the list of Paragraph objects.

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

A Run object has the bold, underline, italic, strike, and outline variables.

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

False – The attribute is always disabled.

True – The attribute is always enabled, no matter what other styles are applied to run.

None – defaults to whatever the run’s style is set to.

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

pip install python-docx

import docx

doc = docx.Document(“sampleDoc.docx”)

#doc is the Document object

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

doc.add\_paragraph(‘Hello, there!’)

--------------------------------------------------------------------------------------------------------------------------------------

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

Answer:

The level of headings is - 0, 1, 2, 3 & 4. The integer 0 makes the heading the Title style, 1 makes in the main heading all the way down, 4 makes the lowest subheading.

--------------------------------------------------------------------------------------------------------------------------------------