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

Ans: When working with the PdfFileReader() and PdfFileWriter() classes from the PyPDF2 library in Python, the underlying file objects representing PDF files should be opened in the following modes:

1. **PdfFileReader():** The file object representing the PDF file should be opened in the 'rb' mode, which stands for "read in binary mode". This mode ensures that the PDF content is read as binary data, which is necessary for processing PDF files.
2. **PdfFileWriter():** The file object representing the PDF file to be written should be opened in the 'wb' mode, which stands for "write in binary mode". This mode allows the PdfFileWriter object to write PDF content as binary data to the specified file.

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

Ans: To obtain a Page object for a specific page, such as page 5, from a PdfFileReader object in the PyPDF2 library, you can use the getPage() method. This method retrieves the page based on the page index (0-based), allowing you to access and manipulate the content of that particular page.

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

Ans: The numPages attribute of a PdfFileReader object in the PyPDF2 library stores the number of pages in the associated PDF document. This attribute provides a way to access the total count of pages within the PDF file, allowing you to perform operations based on the overall page count.

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?

Ans: If a PdfFileReader object's associated PDF is encrypted with the password "swordfish" or any other password, you need to use the decrypt() method before you can obtain Page objects from it. This method decrypts the encrypted PDF file using the specified password, allowing you to access and manipulate the content of the PDF.

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

Ans: To rotate a page in a PDF using PyPDF2, you can use the rotateClockwise() or rotateCounterClockwise() methods provided by the PageObject class. These methods allow you to rotate the content of a specific page clockwise or counterclockwise by 90 degrees.

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

Ans: the key distinction between a Run object and a Paragraph object is that a Run object represents a specific formatted run of text within a paragraph, while a Paragraph object encompasses the entire paragraph, including all of its constituent runs of text and any associated formatting.

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

Ans: To obtain a list of Paragraph objects for a Document object stored in a variable named doc using the python-docx library, you can use the paragraphs attribute. This attribute provides access to the list of all Paragraph objects present in the document.

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

Ans: In the context of the python-docx library, the Run object has the attributes bold, underline, italic, strike, and outline to represent various character formatting options. These attributes allow you to apply specific formatting styles to a particular run of text within a paragraph.

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

Ans: In the context of the Font object in the python-pptx library, the bold property can take three different values, each representing a different text formatting option:

1. False: When the bold property is set to False, it indicates that the text is not formatted to appear in bold. This value means that the text does not have bold formatting applied to it.
2. True: If the bold property is set to True, it means that the text is formatted to appear in bold. This value indicates that the text has the bold formatting applied.
3. None: If the bold property is set to None, it implies that the text inherits the bold formatting from its style hierarchy. In this case, the bold attribute is not explicitly set for the text and depends on the default or inherited style.

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

Ans: To create a Document object for a new Word document using the python-docx library, you can use the Document() function without any arguments. This function initializes a new Document object, allowing you to create and manipulate Word documents programmatically.

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

Ans: To add a paragraph with the text 'Hello, there!' to a Document object stored in a variable named doc using the python-docx library, you can use the add\_paragraph() method of the Document object. This method allows you to append a new paragraph with the specified text to the document.

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

Ans: In Word documents, the integer values used to represent the levels of headings are generally in the range of 0 to 9, inclusive. However, the common heading levels used in most documents typically range from 1 to 9, where 1 represents the highest level of the main heading, and subsequent integers represent lower-level headings.