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

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

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

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?

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

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

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

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

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

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

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

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

**Answers**

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

- \*\*`PdfFileReader()`\*\*: The file should be opened in binary read mode (`'rb'`).

- \*\*`PdfFileWriter()`\*\*: The file should be opened in binary write mode (`'wb'`).

\*\*Example:\*\*

```python

from PyPDF2 import PdfFileReader, PdfFileWriter

# Reading a PDF

pdf\_file = open('example.pdf', 'rb')

pdf\_reader = PdfFileReader(pdf\_file)

# Writing a PDF

output\_pdf = open('output.pdf', 'wb')

pdf\_writer = PdfFileWriter()

# Use pdf\_writer methods to add pages or content

pdf\_writer.write(output\_pdf)

# Close the files

pdf\_file.close()

output\_pdf.close()

```

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

You can get a `Page` object using the `getPage()` method, and since page numbers are zero-indexed, page 5 is at index 4.

\*\*Example:\*\*

```python

from PyPDF2 import PdfFileReader

pdf\_file = open('example.pdf', 'rb')

pdf\_reader = PdfFileReader(pdf\_file)

# Getting the Page object for page 5

page\_five = pdf\_reader.getPage(4)

pdf\_file.close()

```

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

The `numPages` attribute of a `PdfFileReader` object stores the number of pages in the PDF.

\*\*Example:\*\*

```python

from PyPDF2 import PdfFileReader

pdf\_file = open('example.pdf', 'rb')

pdf\_reader = PdfFileReader(pdf\_file)

# Getting the number of pages

total\_pages = pdf\_reader.numPages

print(total\_pages)

pdf\_file.close()

```

### 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?

You must decrypt the PDF using the `decrypt()` method before obtaining `Page` objects.

\*\*Example:\*\*

```python

from PyPDF2 import PdfFileReader

pdf\_file = open('encrypted\_example.pdf', 'rb')

pdf\_reader = PdfFileReader(pdf\_file)

# Decrypting the PDF

pdf\_reader.decrypt('swordfish')

# Now you can get Page objects

page = pdf\_reader.getPage(0)

pdf\_file.close()

```

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

You can use the following methods on a `Page` object to rotate it:

- `rotateClockwise(degrees)`

- `rotateCounterClockwise(degrees)`

\*\*Example:\*\*

```python

from PyPDF2 import PdfFileReader, PdfFileWriter

pdf\_file = open('example.pdf', 'rb')

pdf\_reader = PdfFileReader(pdf\_file)

pdf\_writer = PdfFileWriter()

# Rotate the first page clockwise by 90 degrees

page = pdf\_reader.getPage(0)

page.rotateClockwise(90)

# Add the rotated page to a new PDF

pdf\_writer.addPage(page)

# Write the rotated page to a new file

output\_pdf = open('rotated\_output.pdf', 'wb')

pdf\_writer.write(output\_pdf)

pdf\_file.close()

output\_pdf.close()

```

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

- \*\*`Paragraph` object\*\*: Represents a single paragraph of text in a Word document. A paragraph can contain multiple runs.

- \*\*`Run` object\*\*: Represents a contiguous run of text within a paragraph that has the same formatting.

\*\*Example:\*\*

```python

from docx import Document

doc = Document('example.docx')

paragraph = doc.paragraphs[0] # First paragraph

run = paragraph.runs[0] # First run in the paragraph

```

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

You can obtain a list of `Paragraph` objects using the `paragraphs` attribute.

\*\*Example:\*\*

```python

from docx import Document

doc = Document('example.docx')

paragraphs = doc.paragraphs

for para in paragraphs:

print(para.text)

```

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

A `Run` object has these formatting attributes (`bold`, `underline`, `italic`, `strike`, and `outline`).

\*\*Example:\*\*

```python

from docx import Document

doc = Document('example.docx')

run = doc.paragraphs[0].runs[0]

# Setting text formatting

run.bold = True

run.italic = True

run.underline = True

```

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

- \*\*`False`\*\*: Explicitly sets the text to not be bold.

- \*\*`True`\*\*: Explicitly sets the text to be bold.

- \*\*`None`\*\*: Inherits the bold setting from the surrounding text or style.

\*\*Example:\*\*

```python

from docx import Document

doc = Document()

p = doc.add\_paragraph()

run = p.add\_run('Hello, world!')

run.bold = True # Text will be bold

run.bold = False # Text will not be bold

run.bold = None # Text will inherit bold setting from the style

```

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

You can create a new `Document` object by simply calling `Document()`.

\*\*Example:\*\*

```python

from docx import Document

doc = Document() # Creates a new Word document

doc.save('new\_document.docx')

```

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

You can use the `add\_paragraph()` method.

\*\*Example:\*\*

```python

from docx import Document

doc = Document()

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

doc.save('hello\_document.docx')

```

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

Heading levels in Word documents are represented by integers from 0 to 9:

- \*\*Heading 0\*\*: Title style.

- \*\*Heading 1\*\*: Top-level heading.

- \*\*Heading 2\*\*: Subheading, etc., down to \*\*Heading 9\*\*.

\*\*Example:\*\*

```python

from docx import Document

doc = Document()

doc.add\_heading('Main Title', level=0)

doc.add\_heading('Chapter 1', level=1)

doc.add\_heading('Section 1.1', level=2)

doc.save('headings\_document.docx')

```