

Python Basic- Assignment- 12

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

Answer:

The PdfFileReader() and PdfFileWriter() File objects should be opened in binary mode, with the mode argument set to "rb" (read binary) and "wb" (write binary), respectively.

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

Answer:

You can use the getPage() method of the PdfFileReader object to get a Page object for page 5, like this:

```
page5 = pdfFileReader.getPage(4) # Page numbering starts from 0
```

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

Answer:

The PdfFileReader variable "numPages" 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?

Answer:

We must first call the PdfFileReader object's setPassword() method, passing in the password "swordfish" as an argument. Then we can call the getPage() method to obtain Page objects from it.

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

Answer:

The pages of a PDF can be rotated in 90-degree increments with the rotateClockwise() and rotateCounterClockwise() methods. More can be used as below:

- Use the Image.rotate() function from the Pillow library.
- Use the rotate() function from the OpenCV library.
- Use the transform() function from the scikit-image library.
- Use the rotate() function from the NumPy library.
- Use the ImageOps.mirror() function from the Pillow library.

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

Answer:

A Run object is a collection of characters within a Paragraph object. It can apply formatting to a set of characters within a Paragraph, such as font, size, color, bold, italic, etc.

A Paragraph object is a block of text that contains multiple runs, each of which can have different formatting.

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

Answer:

We can obtain a list of Paragraph objects for a Document object stored in a variable named doc by using the Document object's built-in .paragraphs method, like so. For Example:

```
from docx import Document
doc = Document("sample.docx")
print(doc.paragraphs)
for paragraph in doc.paragraphs:
    print(paragraph.text)
```

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

Answer:

In Python, an object of type str (string) can have bold, underline, italic, strike, and outline variables. To use these variables, the string must be enclosed in the built-in function f-string (formatted string literal).

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

Answer:

`bold = True # Style Set to Bold`

`bold = False # Style Not Set to Bold`

`bold = None # Style is Not Applicable`

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

Answer:

You can use the python-docx library to create a Document object for a new Word document. The code would look something like this: For Example:

```
from docx import Document
document = Document()
document.add_paragraph("This is Mahmud")
document.save('mydocument.docx')
```

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

Answer:

```
from docx import Document
doc = Document()
doc.add_paragraph('Hello, there!')
doc.save('hello.docx')
```

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

Answer:

The level attribute found inside the add_heading method of a word document can be used to specify the levels for a heading. There are a total of five levels, ranging from 0 to 4. where level 0 creates a headline by adding a horizontal line beneath the text, and level 1 is the primary heading. Similar to this, the other headers are sub-headings with diminishing font sizes.