1. Write a Python program to Extract Unique values dictionary values?
2. Write a Python program to find the sum of all items in a dictionary?
3. Write a Python program to Merging two Dictionaries?
4. Write a Python program to convert key-values list to flat dictionary?
5. Write a Python program to insertion at the beginning in OrderedDict?
6. Write a Python program to check order of character in string using OrderedDict()?
7. Write a Python program to sort Python Dictionaries by Key or Value?

Solutions :

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

**Ans:** 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. From a PdfFileReader object, how do you get a Page object for page 5?**

**Ans:** **PdfFileReader** class provides a method called **getPage(page\_no)** to get a page object.

In [2]:

*# Example Code:*

**from** PyPDF2 **import** PdfFileReader

pdf\_reader **=** PdfFileReader(file\_path)

**for** page **in** pdf\_reader**.**getNumPages():

pdf\_reader**.**getPage(page)

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

**Ans:** **getNumPages()** method of **PdfFileReader** class stores the no pages in a PDF document

In [ ]:

*#Example Code:*

**from** PyPDF2 **import** PdfFileReader

pdf\_reader **=** PdfFileReader(file\_path)

print(pdf\_reader**.**getNumPages()) *# Prints the no of pages in a input 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?**

**Ans:** 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 **isEncrypted** 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**

In [ ]:

*#Example Code:*

**from** PyPDF2 **import** PdfFileReader

pdf\_reader **=** PdfFileReader(file\_path)

**if** pdf\_reader**.**isEncrypted: *# to check whether the pdf is encrypted or not*

pdf\_reader**.**decrypt("swordfish")

**for** page **in** pdf\_reader**.**pages:

print(page**.**extractText()) *# to print the text data of a page from pdf*

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

**Ans:** PyPDF2 Package provides 2 methods to rotate a page:

1. **rotateClockWise()** -> 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.

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

**Ans:** 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, color, 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.

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

In [ ]:

*# 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*

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

**Ans:** **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, color, 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.

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

In [ ]:

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?**

In [ ]:

*# Example Program*

**from** docx **import** Document

document **=** Document()

document**.**add\_paragraph("iNeuron Full Stack DataScience Course")

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?**

In [ ]:

*# Example Program*

**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?**

**Ans:** 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 statring 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's font-sizes in decreasing order.