In [ ]:
```
import PyPDF2
```

In [ ]:
```
# importing required modules
import PyPDF2

# creating a pdf file object
pdfFileObj = open('combinedminutes.pdf', 'rb')

# creating a pdf file reader object, pdfReader
pdfReader = PyPDF2.PdfFileReader(pdfFileObj)

# printing number of pages in pdf file
# extract author name, file name, file size
print(pdfReader.numPages)

# creating an object of page object class
pageObj = pdfReader.getPage(1)

# extracting text from page
print(pageObj.extractText())

# closing the pdf file object
pdfFileObj.close()
```

In [ ]:
```
import os
os.getcwd()
```

In [ ]:
```
#Program to decrypt PDF file

import PyPDF2

# creating a pdf file reader object, pdfReader
pdfFile=open('encrypted.pdf', 'rb')
pdfReader = PyPDF2.PdfFileReader(pdfFile)

#checking whether the pdfReaderFile object is encrypted,
#if yes return True, else False
pdfReader.isEncrypted
```

In [ ]:
```
# creating an object of page object clas
pageObj=pdfReader.getPage(0)
```

In [ ]:
```
#usage of decrpyt method, it returns 1 if the password is correct,
#else 0 if incorrect
pdfReader.decrypt('rosebud')
```

In [ ]:
```
# creating an object of page object class
pageObj = pdfReader.getPage(0)
# extracting text from page
print(pageObj.extractText())
```

In [ ]:

```
#Program to create new PDF file by copying pages of existing PDF's

import PyPDF2

#opening files in read-binary mode
pdf1File = open('meetingminutes.pdf', 'rb')
pdf2File = open('meetingminutes2.pdf', 'rb')

#creating PdfFileReader objects of two pdf's
pdf1Reader = PyPDF2.PdfFileReader(pdf1File)
pdf2Reader = PyPDF2.PdfFileReader(pdf2File)

#creating object of class PdfFileWriter
pdfWriter = PyPDF2.PdfFileWriter()


for pageNum in range(pdf1Reader.numPages): # 19 pages-- 0 to 18
    pageObj = pdf1Reader.getPage(pageNum) # page index starts from 0 to 
    pdfWriter.addPage(pageObj) #adds page at the end to the pdfWriter

for pageNum in range(pdf2Reader.numPages): #21 pages --- 0 to 20
    pageObj = pdf2Reader.getPage(pageNum)
    pdfWriter.addPage(pageObj)   #19+21 = 40 pages

pdfOutputFile = open('combinedminutes.pdf', 'wb')

#use write() method
pdfWriter.write(pdfOutputFile)

pdfOutputFile.close()
pdf1File.close()
pdf2File.close()
```

In [ ]:

```
#Program to create new PDF file by rotating page of existing PDF's

import PyPDF2

minutesFile = open('meetingminutes.pdf', 'rb')
#creating object of class PdfFileReader
pdfReader = PyPDF2.PdfFileReader(minutesFile)

#extracting page number  -- 1(First Page)
page = pdfReader.getPage(0)

#rotating the page clockwise by 90 degree
page.rotateCounterClockwise(90)

#creating PdfFileWriter object
pdfWriter = PyPDF2.PdfFileWriter()

#add the rotated page to it using addPage()
pdfWriter.addPage(page)

#The resulting PDF will have one page, rotated 90 degrees clock-wise
resultPdfFile = open('rotatedPage.pdf', 'wb')
pdfWriter.write(resultPdfFile)

#close the files
resultPdfFile.close()
minutesFile.close()
```

In [ ]:
```
#Program to create new PDF file by overlaying  page of existing PDF's

import PyPDF2

#opening the file and extracting the page
minutesFile = open('meetingminutes.pdf', 'rb')
pdfReader = PyPDF2.PdfFileReader(minutesFile)
minutesFirstPage = pdfReader.getPage(0)

#opening the watermark file
pdfWatermarkReader = PyPDF2.PdfFileReader(open('watermark.pdf', 'rb'))

#adding wateramrk to the specified page
minutesFirstPage.mergePage(pdfWatermarkReader.getPage(0))

#creating the output PdfFileWriter object
pdfWriter = PyPDF2.PdfFileWriter()
#adding the watermarked first page
pdfWriter.addPage(minutesFirstPage)

#copying the remaining pages
for pageNum in range(1, pdfReader.numPages):
    pageObj = pdfReader.getPage(pageNum)
    pdfWriter.addPage(pageObj)

#creating the final PDF file
resultPdfFile = open('watermarkedCover.pdf', 'wb')
pdfWriter.write(resultPdfFile)

#closing the files
minutesFile.close()
resultPdfFile.close()
```

In [ ]:
```
#Program to create new encrypted PDF file

import PyPDF2
pdfFile = open('meetingminutes.pdf', 'rb')
pdfReader = PyPDF2.PdfFileReader(pdfFile)

pdfWriter = PyPDF2.PdfFileWriter()

for pageNum in range(pdfReader.numPages):
    pdfWriter.addPage(pdfReader.getPage(pageNum))

pdfWriter.encrypt('swordfish')
resultPdf = open('encryptedminutes.pdf', 'wb')

pdfWriter.write(resultPdf)
resultPdf.close()
```

Working with Word Documents

In [138]:
```
import docx
doc = docx.Document('demo.docx')
len(doc.paragraphs)
```

Out[138]: 7

In [139]:
```
doc.paragraphs[0].text
```

Out[139]: 'Document Title'

In [140]:
```
doc.paragraphs[1].text
```

Out[140]: 'A plain paragraph with some bold and some italic'

In [141]:
```
doc.paragraphs[2].text
```

Out[141]: 'Heading, level 1'

In [144]:
```
doc.paragraphs[3].text
```

Out[144]: 'Intense quote'

In [145]:
```
doc.paragraphs[5].text
```

Out[145]: 'first item in ordered list'

In [146]:
```
for para in doc.paragraphs:
    print(para.text)
    print("-------")
```

```
Document Title
-------
A plain paragraph with some bold and some italic
-------
Heading, level 1
-------
Intense quote
-------
first item in unordered list
-------
first item in ordered list
-------


-------
```

In [147]:
```
len(doc.paragraphs[1].runs)
```

Out[147]: 4

In [148]:
```
doc.paragraphs[1].runs[0].text
```

Out[148]: 'A plain paragraph with some '

In [149]:
```
doc.paragraphs[1].runs[1].text
```

Out[149]: 'bold'

In [150]:
```
doc.paragraphs[1].runs[2].text
```

Out[150]:
```
' and some '
```

In [151]:
```
doc.paragraphs[1].runs[3].text
```

Out[151]:
```
'italic'
```

In [152]:
```
for run in doc.paragraphs[1].runs:
    print(run.text)
    print("-------")
```

```
A plain paragraph with some
-------
bold
-------
 and some
-------
italic
-------
```

In [154]:
```
#to get the entire document as a single string
import docx

def getText(filename):
    doc = docx.Document(filename)
    fullText = []
    for para in doc.paragraphs:
        fullText.append(para.text)
    return '\n'.join(fullText)

doc_str=getText('demo.docx')
print(doc_str)
```

```
Document Title
A plain paragraph with some bold and some italic
Heading, level 1
Intense quote
first item in unordered list
first item in ordered list
```

In [ ]:
```
#writing word documents in python
import docx
doc = docx.Document() #new blank Document object
doc.add_paragraph('Hello world!') #add paragraph
```

In [ ]:
```
paraObj1 = doc.add_paragraph('This is a second paragraph.')
paraObj2 = doc.add_paragraph('This is a yet another paragraph.')
paraObj2.alignment=1 #0-left,1-center,2-right
paraObj1.add_run(' This text is being added to the second paragraph.')
doc.save('helloworld.docx')
```

In [124]:
```
#t retrieve the style info and adding new styles
doc=docx.Document("demo.docx")
doc.paragraphs[0].text
```

Out[124]:
```
'Document Title'
```

In [125]:
```
doc.paragraphs[0].style
```

Out[125]:
```
_ParagraphStyle('Title') id: 2437389766344
```

In [126]:
```
doc.paragraphs[0].style = 'Mamatha'
doc.paragraphs[0].alignment=1  #0 for left, 1 for center, 2 for right
```

In [127]:
```
doc.paragraphs[1].text
```

Out[127]:
```
'A plain paragraph with some bold and some italic'
```

In [128]:
```
(doc.paragraphs[1].runs[0].text, doc.paragraphs[1].runs[1].text, doc.
paragraphs[1].runs[2].text, doc.paragraphs[1].runs[3].text)
```

Out[128]:
```
('A plain paragraph with some ', 'bold', ' and some ', 'italic')
```

In [130]:
```
doc.paragraphs[1].runs[0].style = 'QuoteChar'
doc.paragraphs[1].runs[1].underline = True
doc.paragraphs[1].runs[3].underline = True
doc.paragraphs[1].alignment = 2
doc.add_paragraph("Heelo World","Heading 1")
doc.save('restyled.docx')
```

In [ ]:
```
#To add Headings
import docx
doc = docx.Document()
doc.add_heading('Header 0', 0)
obj=doc.add_paragraph("SAI VIDYA INSTITUTE OF TECHNOLOGY")
obj.add_run("V SEMESTER")
doc.add_heading('Header 1', 1)
doc.add_heading('Header 2', 2)
doc.add_heading('Header 3', 3)
doc.add_heading('Header 4', 4)
doc.save('headings.docx')
```

In [ ]:
```
#To add pictures
import docx
doc = docx.Document()
doc.add_picture('photo.jpg')
doc.save('headings.docx')
```

In [ ]:
```
#To add linebreak/pagebreak
import docx
doc = docx.Document()
doc.add_paragraph('This is on the first page!')
doc.paragraphs[0].runs[0].add_break(docx.enum.text.WD_BREAK.PAGE) #Inserts page break
doc.add_paragraph('This is on the second page!')
doc.save('twoPage1.docx') #no of pages=2
```

In [ ]:
```
#To add linebreak/pagebreak
import docx
doc = docx.Document("twopage.docx")
paraObj=doc.add_paragraph('This is on the first page!')
paraObj.style="Mamatha"
paraObj.add_run('This is on the second page!')
doc.save('twoPage2.docx') #no of pages=2
```

In [134]:
```
#To create invitation on each page for every guest

import docx

# Open the guests.txt,pick up the name of the guests.
guestsTxt = open('guests.txt','r')

INTRO = "It would be a pleasure to have the company of"
ADDRESS = "at 11010 Memory Lane on the Evening of"
DATE = "April 1st"
TIME = "at 7 o'clock"

docInvi = docx.Document("Doc1.docx")

for guestname in guestsTxt: #iterating over every line in the file guestsTxt
    #guestname="Mamatha\n"
    guestname=guestname.strip('\n')
    #guestname="Mamatha"
    docInvi.add_paragraph(INTRO,"Mamatha")
    docInvi.add_paragraph(guestname,"NameStyle")
    docInvi.add_paragraph(ADDRESS,"Mamatha")
    docInvi.add_paragraph(DATE,"Mamatha")
    obj=docInvi.add_paragraph(TIME,"Mamatha")
    obj.runs[0].add_break(docx.enum.text.WD_BREAK.PAGE) #page break

docInvi.save('Invitations.docx')
guestsTxt.close()
print('Done')
```

```
Done
```

In [ ]: