w.
– Stop harming Monica
Apr 22, 2018 at 12:44

- @Goyo thanks. But I'm not able to run this commando in the terminal. It doesn't recognize lowriter as an executable command. Why is that?
  – Also
  Apr 22, 2018 at 13:17
- 3
  How could I know? It is probably related to the way you installed libreoffice. But you'd better figure it out, you can't expect python to run a program when you are unable to run it yourself.
  – Stop harming Monica
  Apr 22, 2018 at 14:20
- What is your operating system? Use [my answer from two weeks ago]() but modify the paths.
  – Jim K
  Apr 23, 2018 at 18:59
- @JimK Thank you very much. I saw your post recently, and I used it as a help for my trouble, and it worked perfectly for converting the pdf to odg. (I checked your answer for that reason). However, I was looking for the conversion to .docx, that seems to be more difficult... In any case, my operating system is Windows 7.
  – Also
  Apr 24, 2018 at 8:55

Show **2** more comments

5 Answers

Sorted by:

Highest score (default) ▾

3

I am not aware of a way to convert a pdf file into a Word file using libreoffice.
However, you can convert from a pdf to a html and then convert the html to a docx.
Firstly, get the commands running on the command line. (The following is on Linux. So you may have to fill in path names to the soffice binary and use a full path for the input file on your OS)

```
soffice --convert-to html ./my_pdf_file.pdf
```

then

```
soffice --convert-to docx:'MS Word 2007 XML' ./my_pdf_file.html
```

You should end up with:

```
my_pdf_file.pdf
my_pdf_file.html
my_pdf_file.docx
```

Now wrap the commands in your subprocess code
Share
Edit
Follow
answered May 7, 2018 at 16:02

Rolf of Saxony
20.4k33 gold badges3636 silver badges5656 bronze badges
Add a comment
3
I use this for multiple files

```
####
from pdf2docx import Converter
import os

# # # dir_path for input reading and output files & a for loop # # #

path_input = '/pdftodocx/input/'
path_output = '/pdftodocx/output/'

for file in os.listdir(path_input):
    cv = Converter(path_input+file)
    cv.convert(path_output+file+'.docx', start=0, end=None)
    cv.close()
    print(file)
```

Share
Edit
Follow
answered Feb 8, 2021 at 20:31

simon
3111 bronze badge
Add a comment
1
My approach does not follow the same methodology of using subsystems. However this one does the job of reading through all the pages of a PDF document and moving them to a docx file. Note: It only works with text; images and other objects are usually ignored.

```
#Description: This python script will allow you to fetch text information from a pdf file

#import libraries

import PyPDF2
import os
import docx

mydoc = docx.Document() # document type
pdfFileObj = open('pdf/filename.pdf', 'rb') # pdffile loction
pdfReader = PyPDF2.PdfFileReader(pdfFileObj) # define pdf reader object

# Loop through all the pages

for pageNum in range(1, pdfReader.numPages):
        pageObj = pdfReader.getPage(pageNum)
        pdfContent = pageObj.extractText()  #extracts the content from the page.
```

```
    print(pdfContent) # print statement to test output in the terminal. codeline optional.
    mydoc.add_paragraph(pdfContent) # this adds the content to the word document

mydoc.save("p
```