*My PDF Toolbox  
How to merge a bunch of PDF files*

Table of Contents

[Introduction 2](#_Toc79766425)

[Scope 2](#_Toc79766426)

[Pre-requisites 2](#_Toc79766427)

[Creating the application skeleton 2](#_Toc79766428)

[Conclusion 6](#_Toc79766429)

Introduction  
  
The primary goal of merging PDF files is for proper file management. For archiving, bulk printing or combining data sheets, e-books, reports, you need an efficient tool to merge small PDF files into a single PDF.

Scope  
   
This tutorial is intended to show you how to merge a list of PDF files into a single PDF using the Python programming language. The combined PDF may include bookmarks to improve the navigation where every bookmark is linked to the content of one of the inputted PDF files.

# Pre-requisites

As our requirements stand, the following components come into play:

1. PyPDF4 is a pure-python PDF library capable of splitting, merging together, cropping, and transforming the pages of PDF files. It can also add custom data, viewing options, and passwords to PDF files. It can retrieve text and metadata from PDFs as well as merge entire files together.

The goal of this tutorial is to develop a lightweight command line based utility in order to merge a bunch of PDF files.  
If this tutorial intrigues you and makes you want to dive into code immediately, then go to this Github repository “https://github.com/bassemmarji/MyPDFToolbox” for the code used in this article.

# Creating the application skeleton

Setting up is quite simple and straightforward. Of course you need python3 installed on your system. It is highly recommended to setup a virtual environment where we will install the needed libraries.

1. Create a virtual environment and activate it.
2. Create a file named requirements.txt and add the following lines in it.

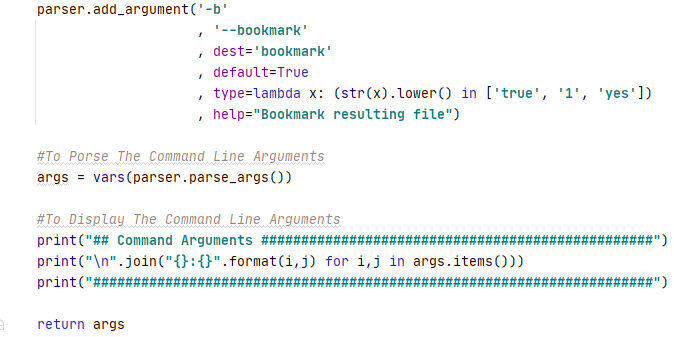
|  |
| --- |
| requirements.txt |
| PyPDF4==1.27.0 |

1. Now, let’s install these libraries to the project.  
   pip install –r requirements.txt
2. Create a folder for hosting our project called “MyPDFToolbox”.

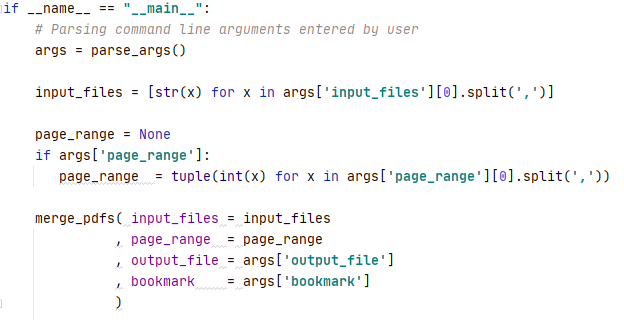
At the end, our folder structure will look like the following:  
  
  
  
  
Let’s start writing the actual code.

*#pdf\_merger.py*  


* This function constitutes the core of our program and performs the following:
  + Creates a merger object.
  + Iterates throughout the list of files to merge (input\_files).
  + For each file, defines a bookmark if required depending on the variable bookmark and add it to the merger object taking into account the page range chosen.
  + Save the merger object to the resulting file (output\_file).
  + Closes the merger object.

* The purpose of this function is to define and to set the appropriate constraints for the command line arguments to be specified by the user.
* I will lay out hereafter the arguments defined:
  + Input files: A required parameter to input the path of the files merge.
  + Page range: to specify a range of pages to be considered within the input files.
  + Output file: The path of the output file.
  + Bookmark: whether to add bookmarks to the resulting file or not.

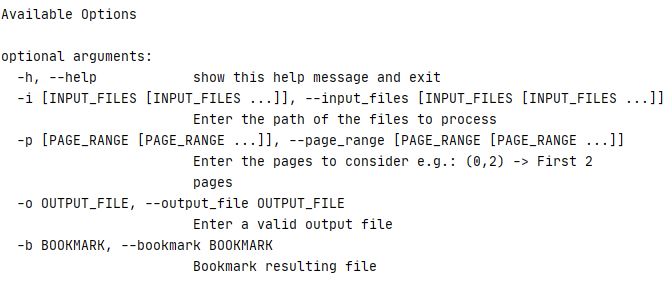


* The above represents the main function of our program.

Let’s test our program:  
  
Please proceed as per the following steps:

1. Open up a terminal window and type the following in it:

**python pdf\_merger –h**

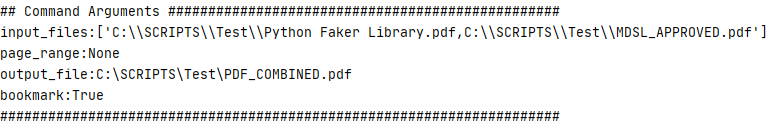
The list of available options will be displayed:

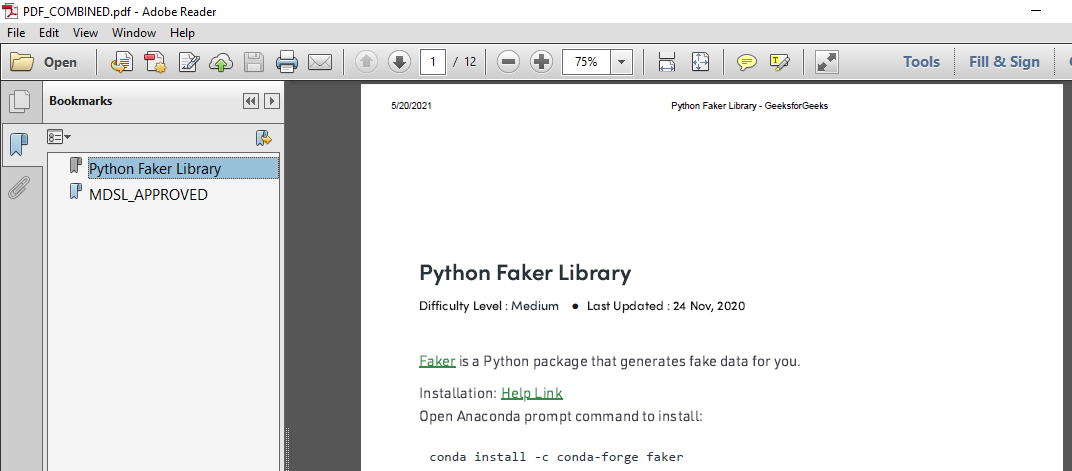
Beware, to avoid encountering the error “Permission error”, please close the PDF files to merge and the resulting PDF file before running this utility.

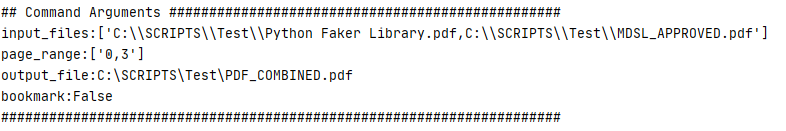
1. Let’s merge the following PDF files:  
     
   **pdf\_merger** **-i** "C:\SCRIPTS\Test\Python Faker Library.pdf","C:\SCRIPTS\Test\MDSL\_APPROVED

.pdf" **-o** "C:\SCRIPTS\Test\PDF\_COMBINED.pdf"

The following information will be displayed on the terminal window showing the command line arguments:



The above 2 files were combined and the resulting file “PDF\_COMBINED.pdf” contains bookmarks targeting each of these files as shown hereafter:  
  


Now let’s try to select a range of pages (i.e.: the first 3 pages of the chosen PDF files) and without adding bookmarks:  
  
The following information will be displayed:  


Conclusion

Hoping you enjoyed this cool feature.

|  |  |
| --- | --- |
|  | ***Bassem Marji*** *is a project implementation manager at BLOM Bank with a proven track record of success.  He managed the implementation of over 50 projects and propelled the digital transformation of mission critical applications. He spends his free time discovering the latest technology trends in the IT field.* |