# Combine-PDF

A Python script to combine multiple PDFs, allowing the insertion of one PDF before the last page of another. Flexible for adding additional documents. Perfect for document management tasks.

# PDF Combiner

## Description

A Python script to combine multiple PDFs, allowing the insertion of one PDF before the last page of another. This tool is ideal for managing and organizing documents, providing flexibility for adding additional PDFs.

## Requirements

- Python 3.x

- PyPDF2

## Installation

1. Clone the repository:

```bash

git clone https://github.com/ashainp/combine-PDF.git

2. Navigate to the directory

cd combine-PDF

3. Install the required package

pip install PyPDF2

##Usage

1. Update the file paths in the combine\_pdfs.py script:

- base\_pdf\_path: The path to your base PDF file.

- insert\_pdf\_path: The path to the PDF file you want to insert.

- output\_pdf\_path: The path where the combined PDF will be saved.

2. Run script

python combine.py

##Additional features

The script can be easily modified to add more PDFs in any sequence.

##Other functions

1. Merge Multiple PDFs

This function merges multiple PDFs into a single document, combining all pages from several documents in the specified order.

def merge\_pdfs(pdf\_list, output\_pdf\_path):

pdf\_writer = PyPDF2.PdfWriter()

for pdf in pdf\_list:

with open(pdf, 'rb') as pdf\_file:

pdf\_reader = PyPDF2.PdfReader(pdf\_file)

for page in pdf\_reader.pages:

pdf\_writer.add\_page(page)

with open(output\_pdf\_path, 'wb') as output\_file:

pdf\_writer.write(output\_file)

2. Extract Text from PDFs

This function extracts and saves the text content of a PDF to a .txt file.

def extract\_text(pdf\_path, output\_txt\_path):

with open(pdf\_path, 'rb') as pdf\_file:

pdf\_reader = PyPDF2.PdfReader(pdf\_file)

with open(output\_txt\_path, 'w') as txt\_file:

for page\_num in range(len(pdf\_reader.pages)):

page = pdf\_reader.pages[page\_num]

txt\_file.write(page.extract\_text())

txt\_file.write('\n')

3. Split a PDF into Separate Pages

This function splits a multi-page PDF into individual single-page PDFs.

def split\_pdf(pdf\_path, output\_folder):

with open(pdf\_path, 'rb') as pdf\_file:

pdf\_reader = PyPDF2.PdfReader(pdf\_file)

for page\_num in range(len(pdf\_reader.pages)):

pdf\_writer = PyPDF2.PdfWriter()

pdf\_writer.add\_page(pdf\_reader.pages[page\_num])

output\_path = f'{output\_folder}/page\_{page\_num + 1}.pdf'

with open(output\_path, 'wb') as output\_pdf:

pdf\_writer.write(output\_pdf)

4. Add Watermarks to PDFs

This function adds a watermark to each page of a PDF using another PDF page as the watermark.

def add\_watermark(input\_pdf, output\_pdf, watermark\_pdf):

with open(input\_pdf, 'rb') as pdf\_file, open(watermark\_pdf, 'rb') as watermark\_file:

pdf\_reader = PyPDF2.PdfReader(pdf\_file)

watermark\_reader = PyPDF2.PdfReader(watermark\_file)

watermark\_page = watermark\_reader.pages[0]

pdf\_writer = PyPDF2.PdfWriter()

for page\_num in range(len(pdf\_reader.pages)):

page = pdf\_reader.pages[page\_num]

page.merge\_page(watermark\_page)

pdf\_writer.add\_page(page)

with open(output\_pdf, 'wb') as output\_file:

pdf\_writer.write(output\_file)

5. Encrypt PDFs with a Password

This function encrypts a PDF with a password to secure its contents.

def encrypt\_pdf(input\_pdf, output\_pdf, password):

pdf\_writer = PyPDF2.PdfWriter()

with open(input\_pdf, 'rb') as pdf\_file:

pdf\_reader = PyPDF2.PdfReader(pdf\_file)

for page in pdf\_reader.pages:

pdf\_writer.add\_page(page)

pdf\_writer.encrypt(password)

with open(output\_pdf, 'wb') as output\_file:

pdf\_writer.write(output\_file)