Get unlimited access to the best of Medium for less than $1/week. Become a member

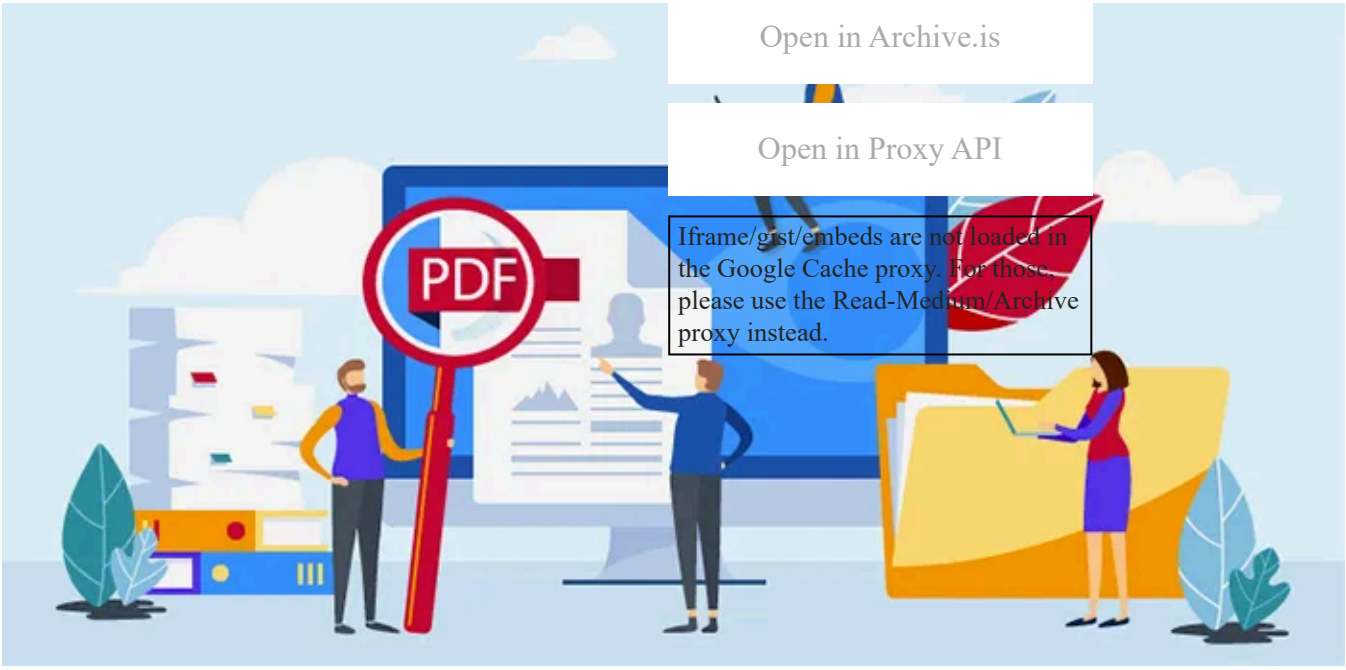
Open in Google Cache
Open in Read-Medium
Open in Freedium
Open in Archive.today
Open in Archive.is
Open in Proxy API
Iframe/gist/embeds are not loaded in the Google Cache proxy. For those, please use the Read-Medium/Archive proxy instead.

Extract all Images from PDF in Python

Ali Aref · Follow

2 min read · Aug 4, 2021

Listen Share More



In this tutorial, we will write a Python code to extract images from PDF files and save them in the local disk using **PyMuPDF** and **Pillow** libraries.

With **PyMuPDF**, you are able to access PDF, XPS, OpenXPS, epub and many other extensions. It should run on all platforms including Windows, Mac OSX and Linux.

Let's get started!

First of all install the required modules.

```
python -m pip install PyMuPDF Pillow
```

Now Open/Create your python file and import the **libraries.**

```
import io
import fitz
from PIL import Image
```

For testing a pdf file we gonna use this file. Feel free to choose any file and make sure you put the file in your working directory, or you have the correct path to pdf file.

```
# file path you want to extract images from
file = "1770.521236.pdf"

# open the file
pdf_file = fitz.open(file)
```

Since we want to extract images from all pages, we need to iterate over all the pages available, and get all image objects on each page, the following code does that:

```
# iterate over pdf pages
for page_index in range(len(pdf_file)):
    # get the page itself
    page = pdf_file[page_index]
    image_list = page.getImageList()
    # printing number of images found in this page
    if image_list:
        print(f"[+] Found a total of {len(image_list)} images in
page {page_index}")
    else:
        print("[!] No images found on page", page_index)
    for image_index, img in enumerate(page.getImageList(), start=1):
        # get the XREF of the image
        xref = img[0]
        # extract the image bytes
        base_image = pdf_file.extractImage(xref)
        image_bytes = base_image["image"]
        # get the image extension
        image_ext = base_image["ext"]
        # load it to PIL
        image = Image.open(io.BytesIO(image_bytes))
        # save it to local disk
        image.save(open(f"image{page_index+1}_{image_index}.
{image_ext}", "wb"))
```

We're using `getImageList()` method to list all available image objects as a list of tuples in that particular page. To get the image object index, we simply get the first element of the tuple returned.

After that, we use the `extractImage()` method that returns the image in bytes along with additional information such as the image extension.

Finally, we convert the image bytes to a **PIL image instance** and save it to the local disk using the `save()` method, which accepts a file pointer as an argument, then we're simply naming the images with their corresponding page and image indices.

That was it!

After running the script you will get the following output:

```
[!] No images found on page 0
[+] Found a total of 3 images in page 1
[+] Found a total of 3 images in page 2
[!] No images found on page 3
[!] No images found on page 4
```

And the images are saved as well, in the current directory.

Conclusion

Alright, we have successfully extracted images from that PDF file without loosing image quality. For more information on how the library works, I suggest you take a look at the documentation.

Python Pdf Extraction

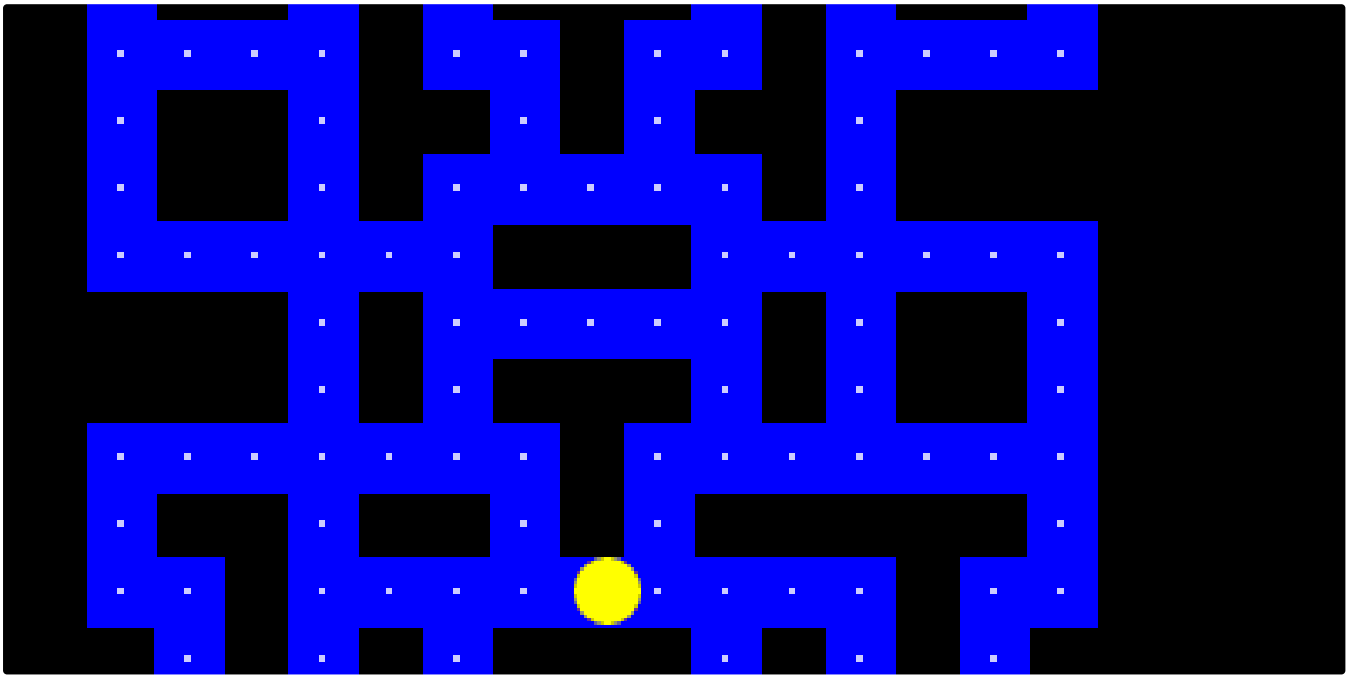
Follow

Written by Ali Aref

103 Followers

https://aliaref.dev

More from Ali Aref



Ali Aref

Build Simple Python Games

Python developers never never get tired

8 min read · Aug 16, 2021

22

Ali Aref

What is Vim?

Why should I use it ? What are the pros and cons ?

7 min read · Sep 20, 2021

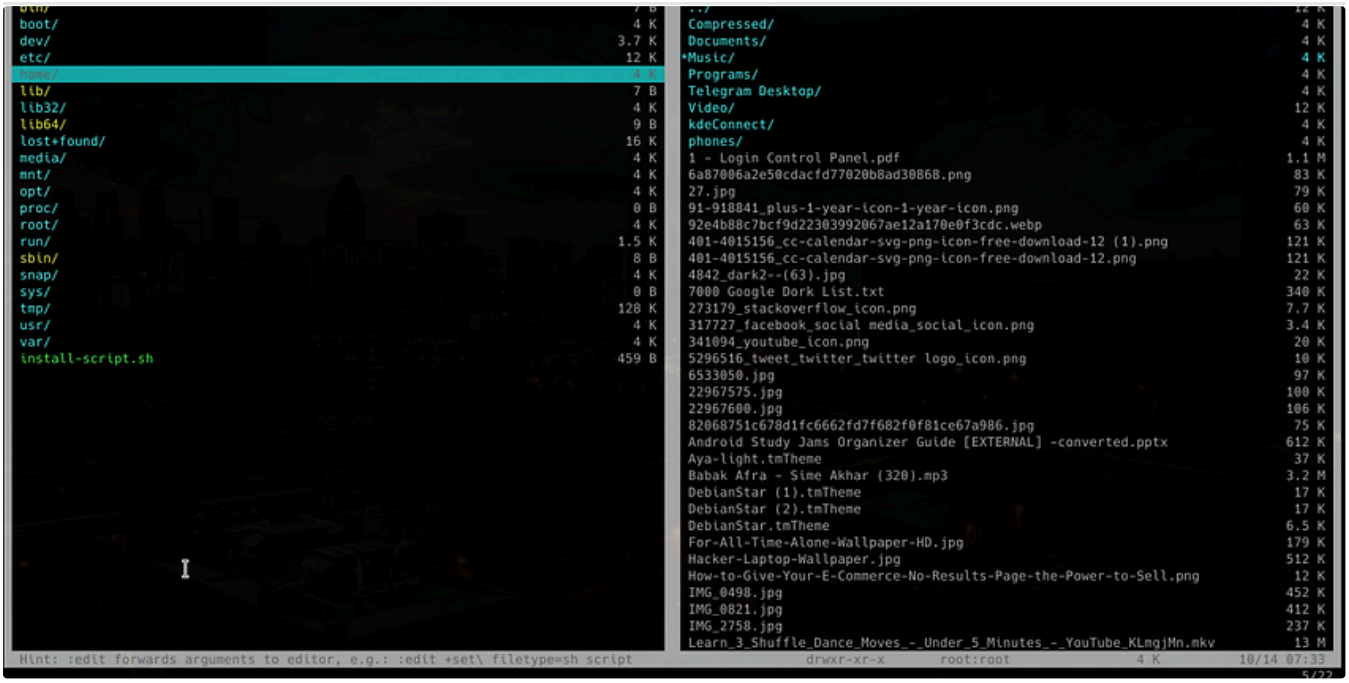
8

Ali Aref

Arch Linux vs Ubuntu: Which One Is The Best Choice For Me?

Making the decision between two popular operating systems can be difficult, but with this article, you can find out if Arch Linux or Ubuntu...

5 min read · Dec 25, 2022



Ali Aref

Vifm—Powerful command line file manager

Vifm is an ncurses based file manager with vi(m) like keybindings. If you use vi(m), then vifm gives you complete keyboard control over...

6 min read · Oct 2, 2021

3 1

See all from Ali Aref

Recommended from Medium

Open in app ↗

Search

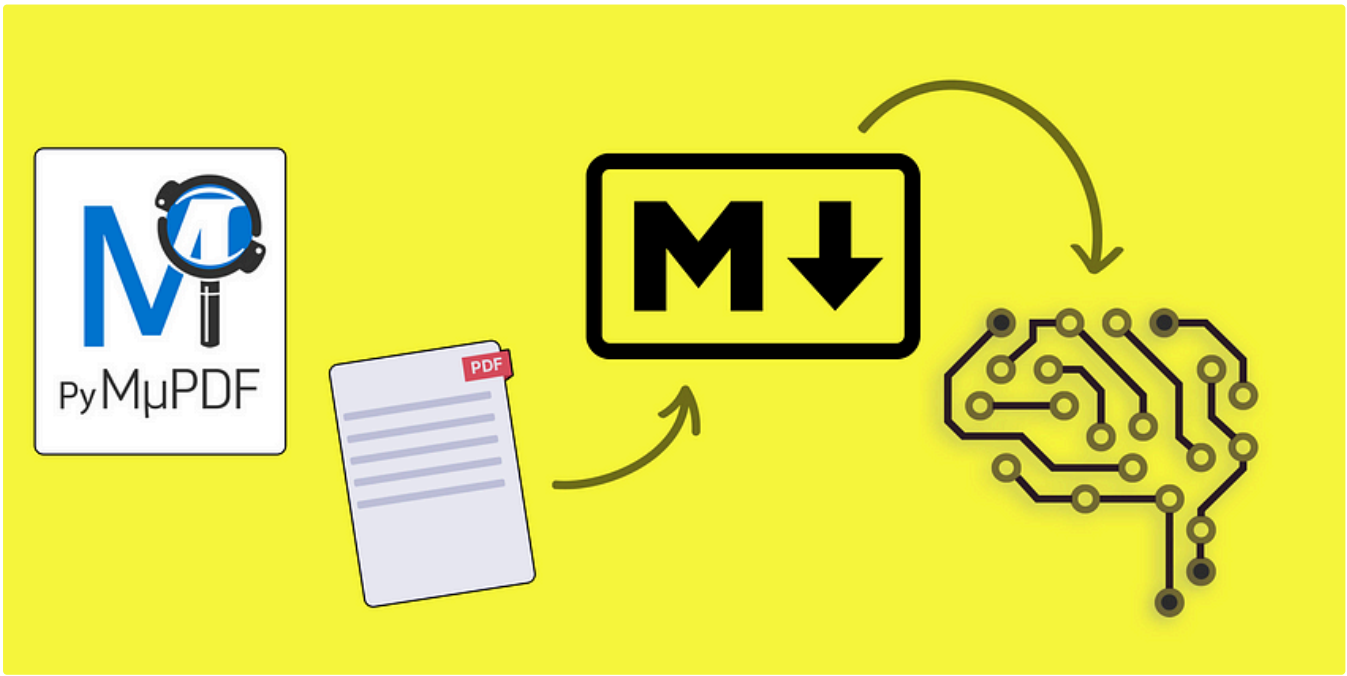
Benoit Pothier

Generating structured data from an image with GPT vision and Langchain

In today's world, where visual data is abundant, the ability to extract meaningful information from images is becoming increasingly...

6 min read · Mar 6, 2024

81 2



PyMuPDF

RAG/LLM and PDF: Conversion to Markdown Text with PyMuPDF

Data feeding in markdown text format increases generated text quality

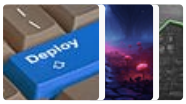
5 min read · Apr 11, 2024

67 2

Lists

Coding & Development

11 stories · 617 saves



Predictive Modeling w/ Python

20 stories · 1205 saves

Practical Guides to Machine Learning

10 stories · 1455 saves

ChatGPT

21 stories · 642 saves

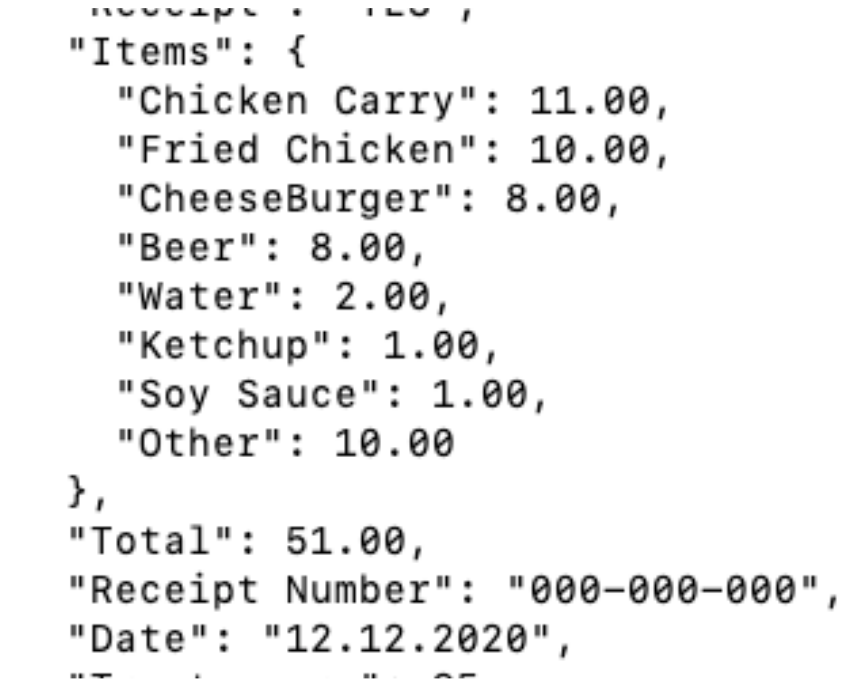
Evenword

6 Python Packages for Working with PDF Files

Here are the top 6 Python packages for working on PDF files. These packages offer comprehensive support for various PDF operations...

2 min read · Nov 28, 2023

107



Dr.Pixel

Document Image Understanding with OpenAI's GPT4-Vision

As I work on several document understanding projects, I wanted to test document reading capabilities of GPT-4-Vision model from OpenAI.

· 3 min read · Feb 20, 2024

1

Alice Yang

Extract Images from Word Documents with Python

Images are often an integral part of Word documents, providing visual context and enhancing the overall presentation. Extracting these...

6 min read · Mar 28, 2024

50

Netra Prasad Neupane

Invoice Extraction Bot using Streamlit and Gemini

The large language model has been a very hot topic since the release of ChatGPT in late 2022. There is more than one LLM release every...

7 min read · Feb 16, 2024

 103  2 

See more recommendations