The code you provided is a good start, but it has a few issues that need to be addressed:

1. The code checks if the input file ends with “.pdf” and the output file ends with “.doc”, but you mentioned you want to convert a PDF to a text file. So, the output file should end with “.txt”, not “.doc”.
2. The code only extracts text from the first page of the PDF. If you want to extract text from all pages, you’ll need to loop over all the pages.
3. The extracted text is printed to the console, but it’s not saved to the output file.

Here’s a revised version of your code that addresses these issues:

# -\*- coding: utf-8 -\*-

# importing required modules

try:

import PyPDF2

except ImportError:

import os

os.system('pip install PyPDF2')

import PyPDF2

def extract\_text\_from\_pdf(pdf\_filename, txt\_filename):

# creating a pdf reader object

reader = PyPDF2.PdfReader(pdf\_filename)

# printing number of pages in pdf file

print(len(reader.pages))

# extracting text from all pages and saving to txt file

with open(txt\_filename, 'w', encoding='utf-8') as f: # Open the file with UTF-8 encoding

for page in reader.pages:

text = page.extract\_text()

f.write(text + '\n')

if \_\_name\_\_ == "\_\_main\_\_":

pdf\_filename = input("Please enter the path to the PDF file: ")

txt\_filename = input("Please enter the path to the output text file: ")

if not pdf\_filename.endswith(".pdf"): #check pdf file

print("No pdf file specified")

elif not txt\_filename.endswith(".txt"): #check txt file

print("No txt file specified")

else:

# Migrating starts here....

extract\_text\_from\_pdf(pdf\_filename, txt\_filename)

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

In this modified version, the code that extracts text from the PDF and writes it to a text file has been moved into a function called extract\_text\_from\_pdf. This function takes two arguments: pdf\_filename and txt\_filename, which are the paths to the PDF file and the text file, respectively. The function is then called in the if \_\_name\_\_ == "\_\_main\_\_": block, which is executed when the script is run directly from the command line.

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

[This code will extract text from all pages of the PDF and save it to a text file1](https://www.askpython.com/python/examples/convert-pdf-to-txt). [Please ensure that the PyPDF2 library is installed in your Python environment1](https://www.askpython.com/python/examples/convert-pdf-to-txt). If it’s not, you can install it using pip:

pip install PyPDF2

[Please note that PyPDF2 may not always be able to correctly extract text if the PDF contains images, complex layouts, or non-standard fonts2](https://www.geeksforgeeks.org/extract-text-from-pdf-file-using-python/). [In such cases, you might need to use more advanced libraries or services3](https://kb.aspose.com/pdf/python/how-to-convert-pdf-to-text-in-python/)

# -\*- coding: utf-8 -\*-

# importing required modules

try:

import PyPDF2

except ImportError:

import os

os.system('pip install PyPDF2')

import PyPDF2

def extract\_text\_from\_pdf(pdf\_filename, txt\_filename):

# creating a pdf reader object

reader = PyPDF2.PdfReader(pdf\_filename)

# printing number of pages in pdf file

print(len(reader.pages))

# extracting text from all pages and saving to txt file

with open(txt\_filename, 'w', encoding='utf-8') as f: # Open the file with UTF-8 encoding

for page in reader.pages:

text = page.extract\_text()

f.write(text + '\n')

if \_\_name\_\_ == "\_\_main\_\_":

pdf\_filename = input("Please enter the path to the PDF file: ")

txt\_filename = input("Please enter the path to the output text file: ")

if not pdf\_filename.endswith(".pdf"): #check pdf file

print("No pdf file specified")

elif not txt\_filename.endswith(".txt"): #check txt file

print("No txt file specified")

else:

# Migrating starts here....

extract\_text\_from\_pdf(pdf\_filename, txt\_filename)