Computer Programming Using Artificial Intelligence

Alan German

May 10, 2023

Programing Possibilities

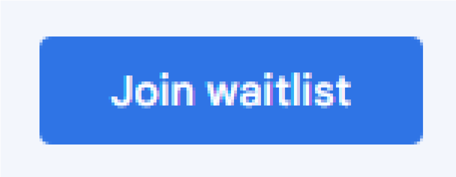
- Generate code in a number of popular languages (C++, Python, Java…)
- Scripting (Batch commands, PowerShell)
- Macros (VBA, LibreOffice Basic)
- Web page development (HTML, PHP, JavaScript)
- Code debugging and optimization
- Documentation (Add comments)
- Learning assistant

ChatGPT

- Developed by OpenAI
- During the research preview usage is free
- Try it at: https://chat.openai.com
- *ChatGPT Plus* subscription plan
  - $20/month
  - access during peak times
  - faster response
  - priority for new features and improvements

Bard

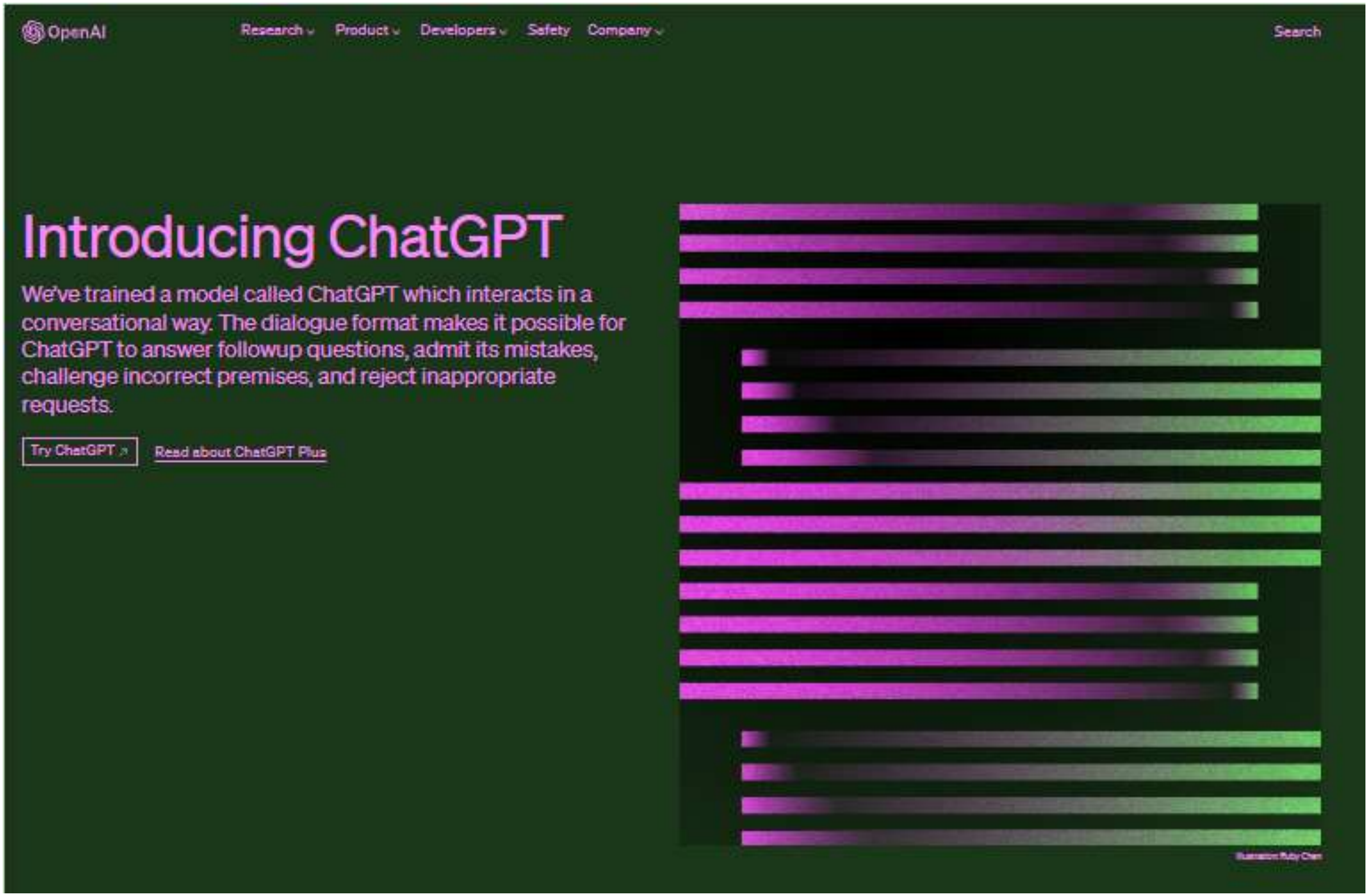
- Developed by Google
- https://bard.google.com
- Help with coding and topics about coding
- Bard is an experiment and may give inaccurate or inappropriate responses
- You can help make Bard better by leaving feedback
- Join the waitlist and try it for yourself



Bard isn’t currently supported in your country. Stay tuned!

GitHub Copilot

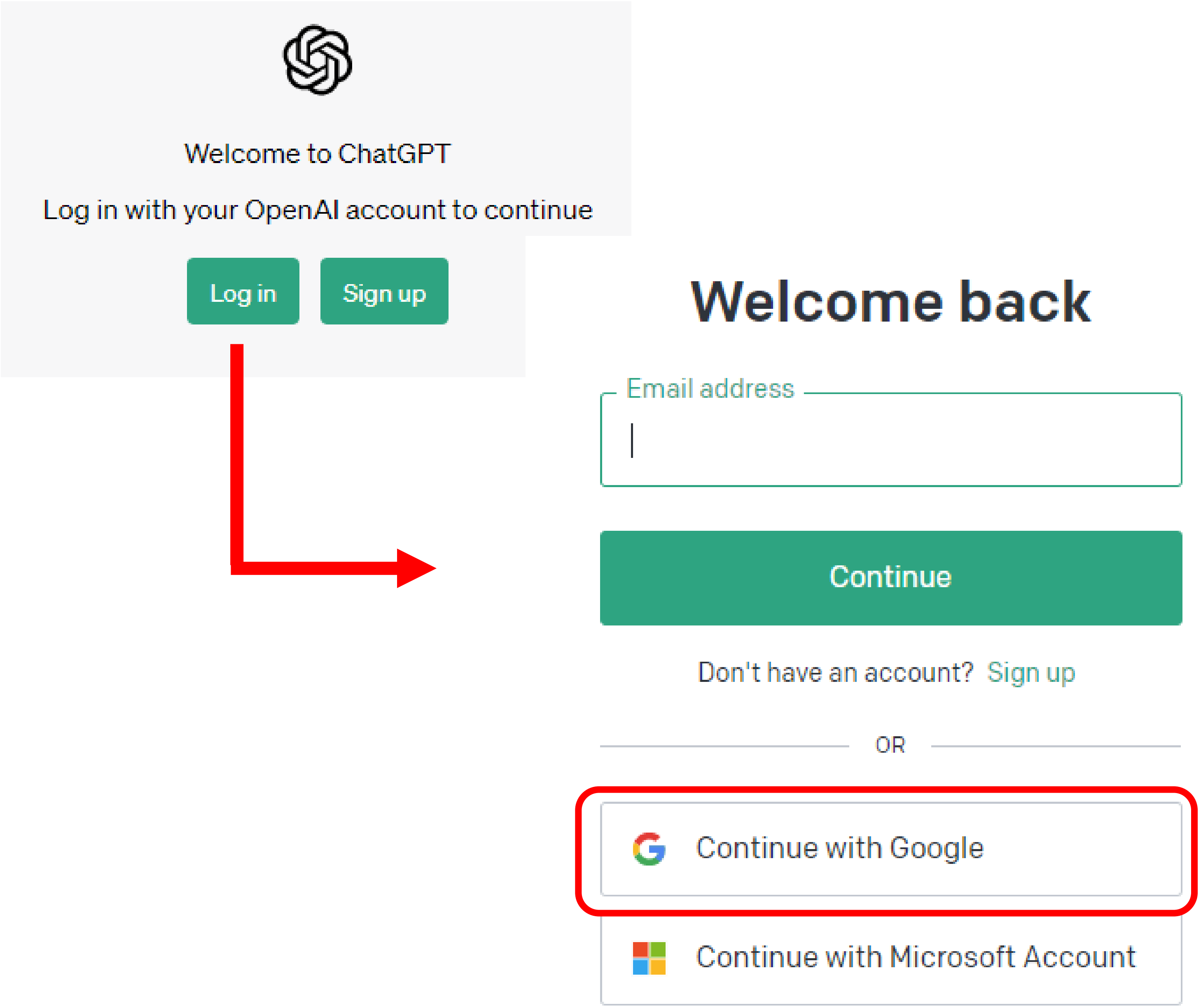
- Developed by GitHub and OpenAI
- 30-day free trial
- Need to setup a (free) GitHub account
- Plans – $10/month for individuals
  – $20/month for business



Introducing ChatGPT

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

https://openai.com/blog/chatgpt

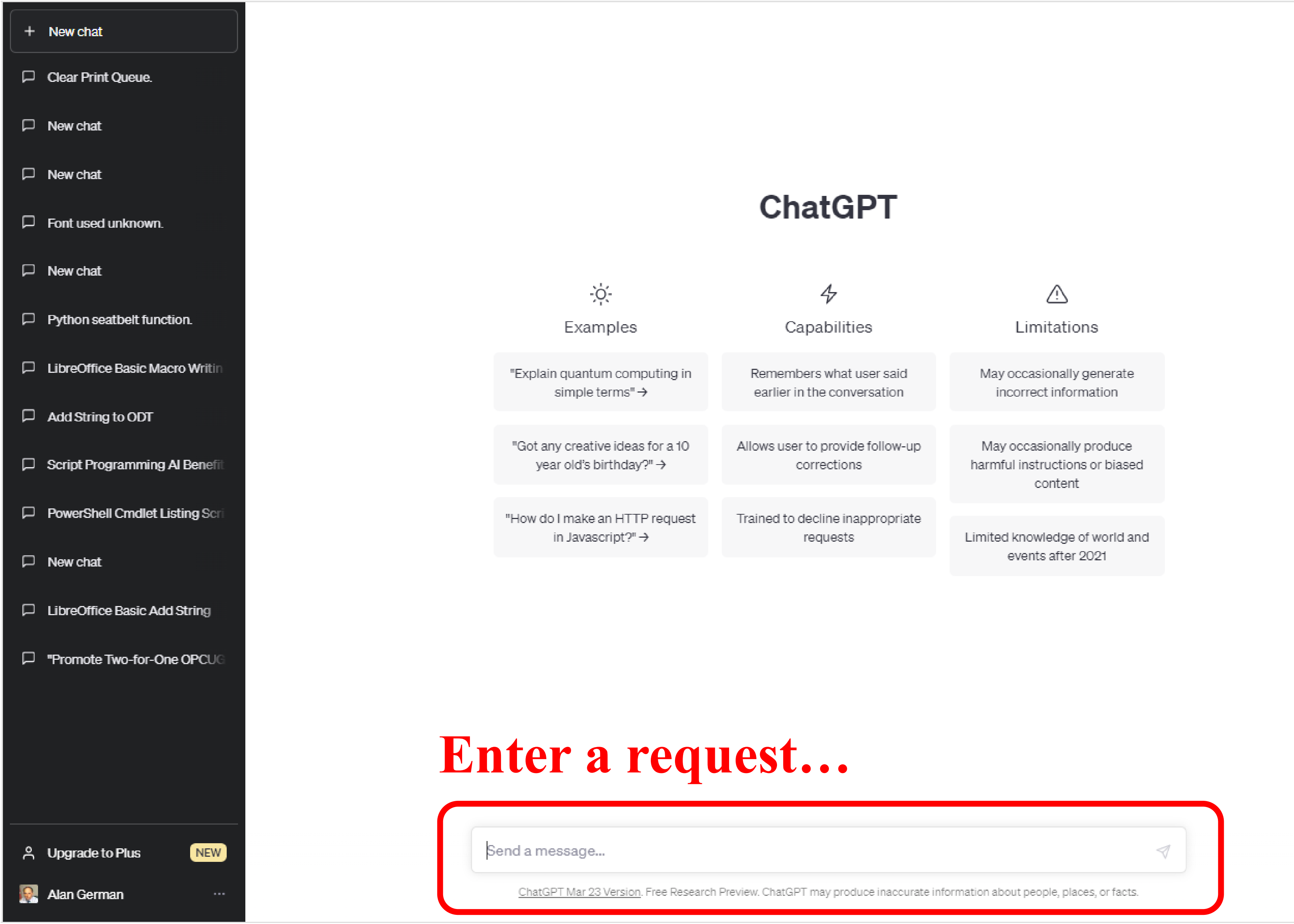


Welcome to ChatGPT

Log in with your OpenAI account to continue

Welcome back

Don't have an account? Sign up


+ New chat
Clear Print Queue.
New chat
New chat
Font used unknown.
New chat
Python seatbelt function.
LibreOffice Basic Macro Writin
Add String to ODT
Script Programming AI Benefit
PowerShell Cmdlet Listing Scri
New chat
LibreOffice Basic Add String
"Promote Two-for-One OPCUG
Upgrade to Plus NEW
Alan German ...

ChatGPT

Examples

"Explain quantum computing in simple terms" →

"Got any creative ideas for a 10 year old's birthday?" →

"How do I make an HTTP request in Javascript?" →

Capabilities

Remembers what user said earlier in the conversation

Allows user to provide follow-up corrections

Trained to decline inappropriate requests

Limitations

May occasionally generate incorrect information

May occasionally produce harmful instructions or biased content

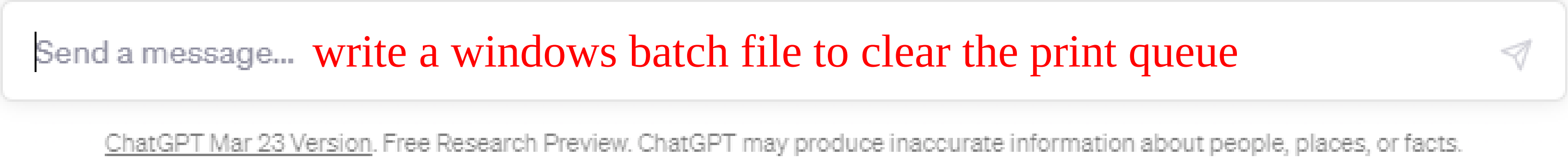
Limited knowledge of world and events after 2021

Enter a request…

Send a message...

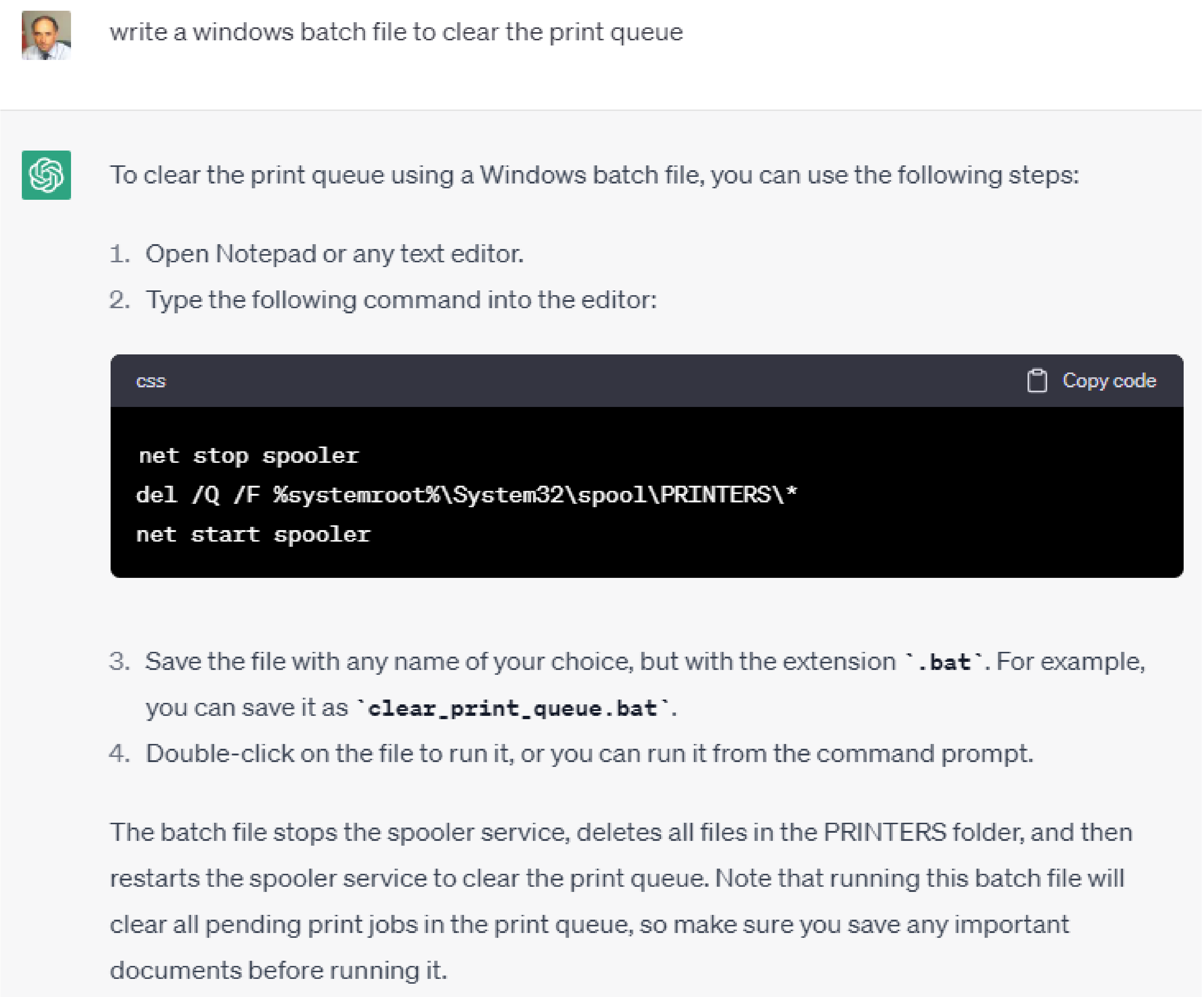
ChatGPT Mar 23 Version. Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts.

Batch File



write a windows batch file to clear the print queue

ChatGPT Mar 23 Version. Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts.



write a windows batch file to clear the print queue

To clear the print queue using a Windows batch file, you can use the following steps:

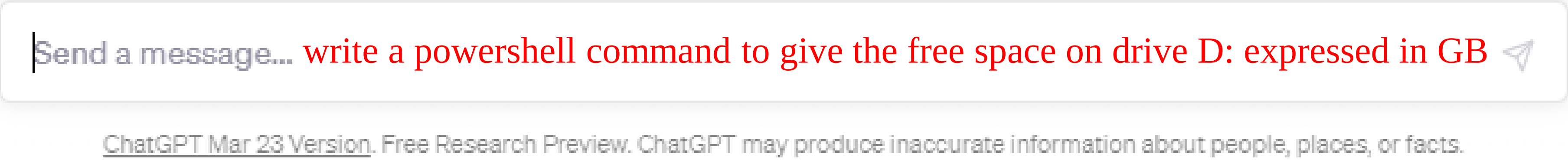
1. Open Notepad or any text editor.
2. Type the following command into the editor:

```css
net stop spooler
del /Q /F %systemroot%\System32\spool\PRINTERS\*
net start spooler
```

3. Save the file with any name of your choice, but with the extension `` `.bat` ``. For example, you can save it as `` `clear_print_queue.bat` ``.
4. Double-click on the file to run it, or you can run it from the command prompt.

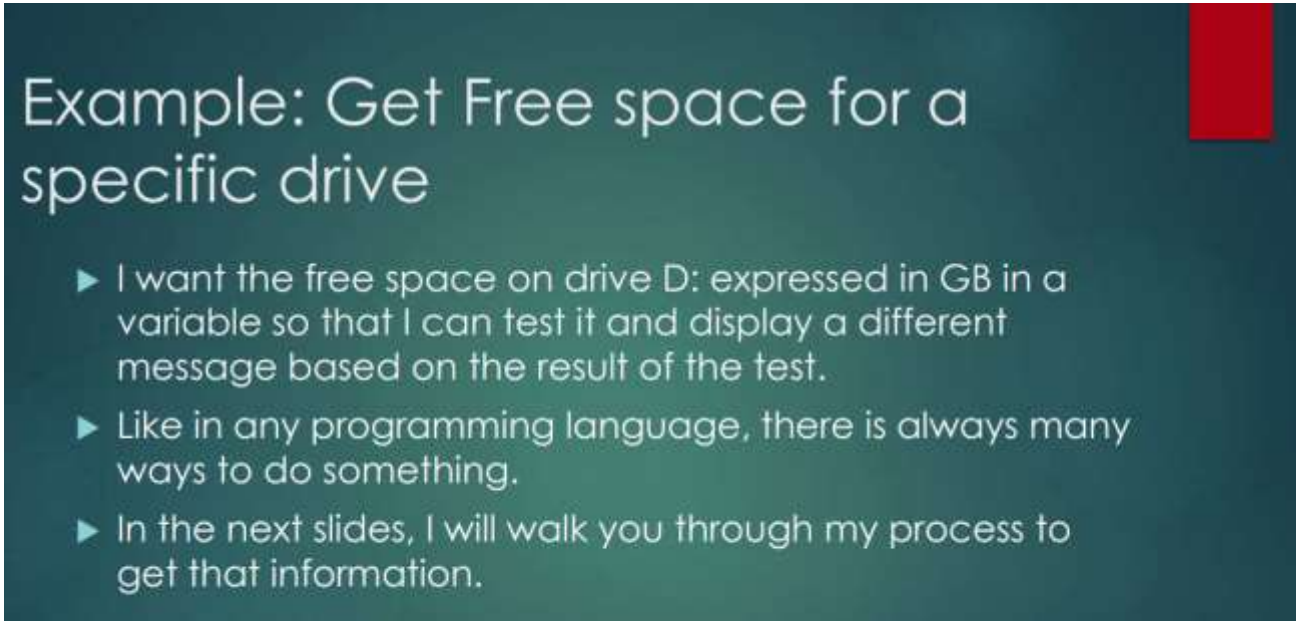
The batch file stops the spooler service, deletes all files in the PRINTERS folder, and then restarts the spooler service to clear the print queue. Note that running this batch file will clear all pending print jobs in the print queue, so make sure you save any important documents before running it.

PowerShell Script



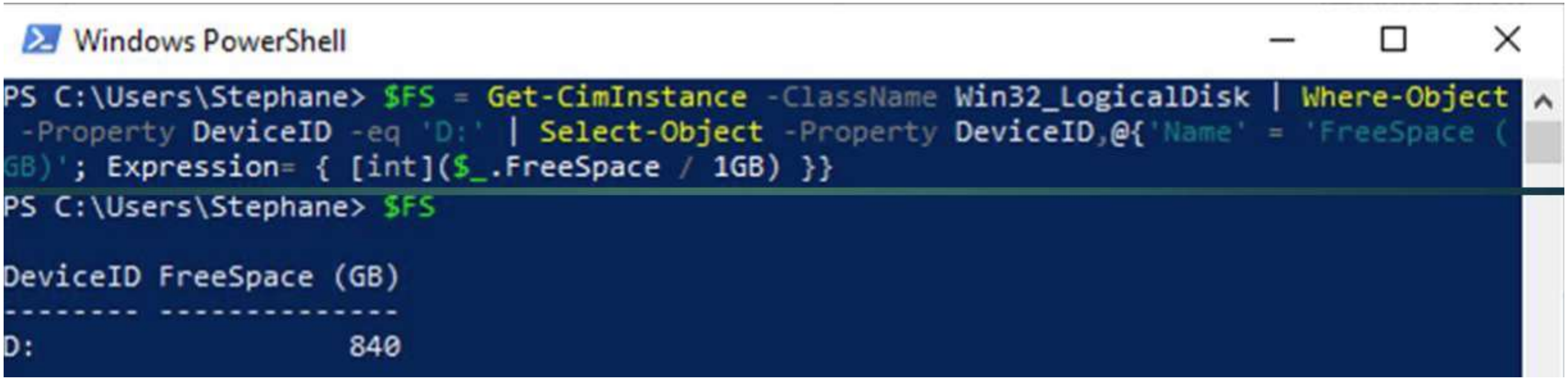
Send a message... write a powershell command to give the free space on drive D: expressed in GB

ChatGPT Mar 23 Version. Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts.



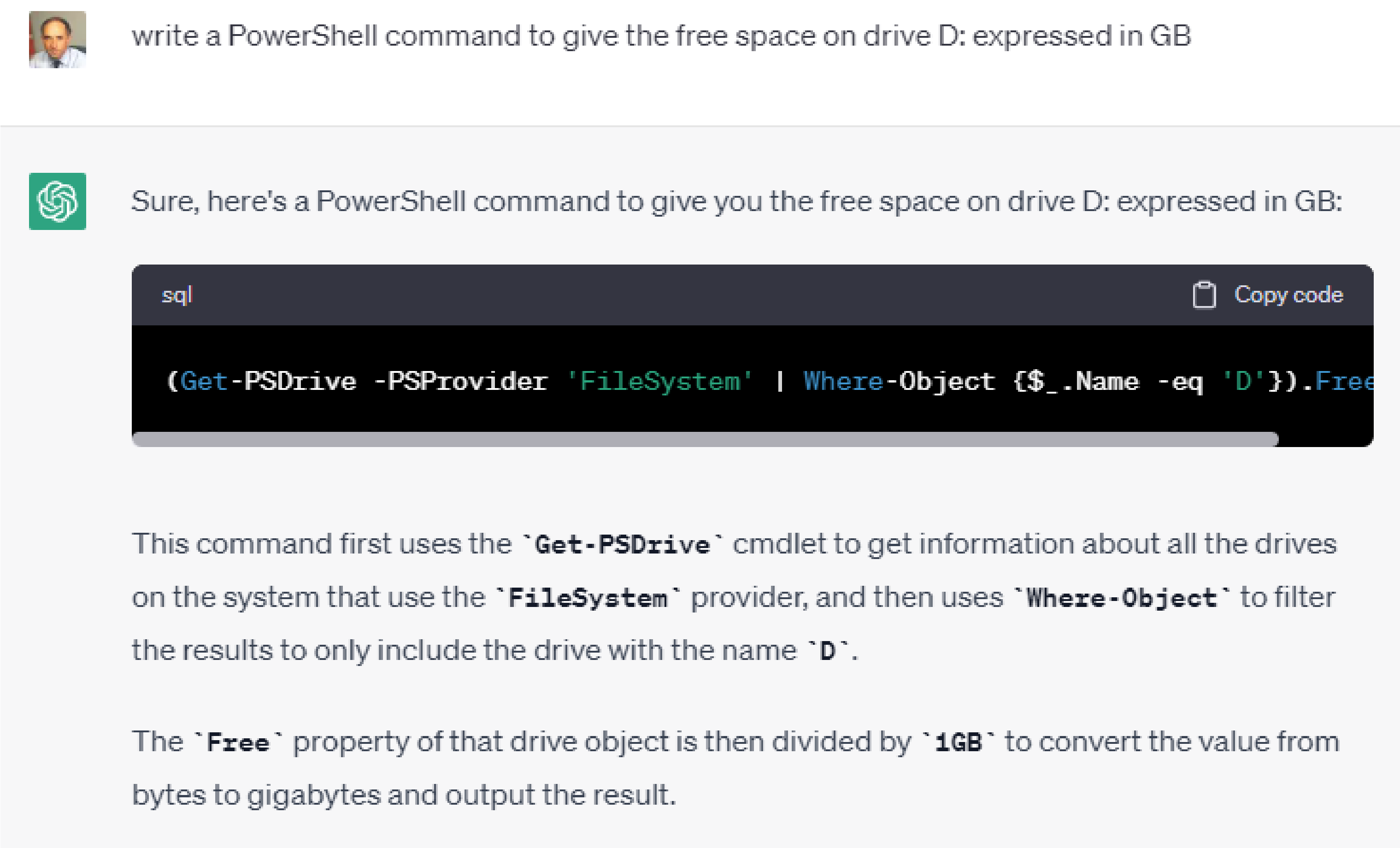
Example: Get Free space for a specific drive

- I want the free space on drive D: expressed in GB in a variable so that I can test it and display a different message based on the result of the test.
- Like in any programming language, there is always many ways to do something.
- In the next slides, I will walk you through my process to get that information.



```
PS C:\Users\Stephane> $FS = Get-CimInstance -ClassName Win32_LogicalDisk | Where-Object
 -Property DeviceID -eq 'D:' | Select-Object -Property DeviceID,@{'Name' = 'FreeSpace (
GB)'; Expression= { [int]($_.FreeSpace / 1GB) }}
PS C:\Users\Stephane> $FS

DeviceID FreeSpace (GB)
-------- --------------
D:                  840
```

OPCUG Share PowerShell Overview
Stéphane Richard
https://opcug.ca/qa_recordings/20230125_PowerShell.pdf



write a PowerShell command to give the free space on drive D: expressed in GB

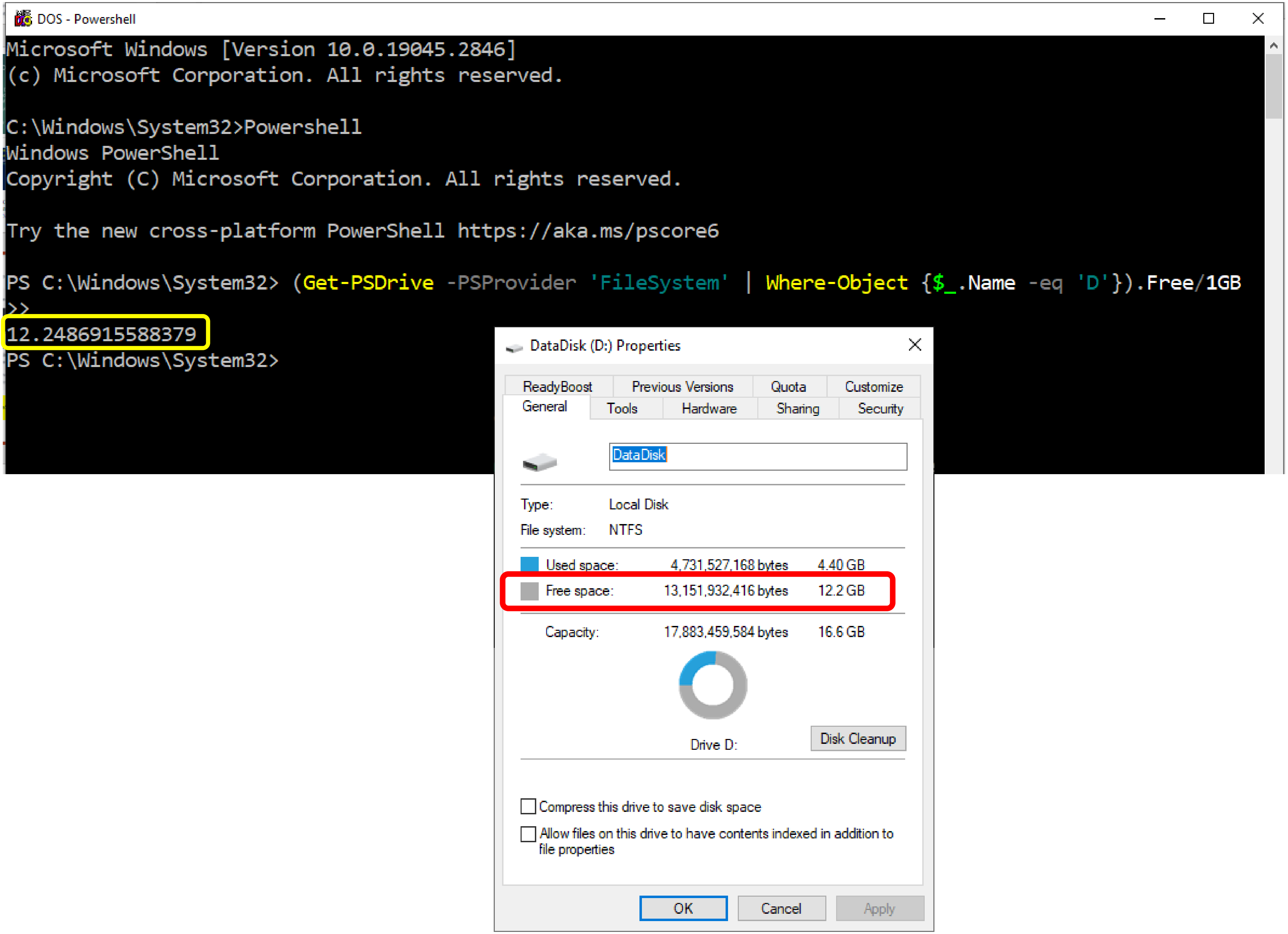
Sure, here's a PowerShell command to give you the free space on drive D: expressed in GB:

```sql
(Get-PSDrive -PSProvider 'FileSystem' | Where-Object {$_.Name -eq 'D'}).Free
```

This command first uses the `Get-PSDrive` cmdlet to get information about all the drives on the system that use the `FileSystem` provider, and then uses `Where-Object` to filter the results to only include the drive with the name `D`.

The `Free` property of that drive object is then divided by `1GB` to convert the value from bytes to gigabytes and output the result.

```
(Get-PSDrive -PSProvider 'FileSystem' | Where-Object {$_.Name -eq 'D'}).Free/1GB
```



```
Microsoft Windows [Version 10.0.19045.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>Powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\System32> (Get-PSDrive -PSProvider 'FileSystem' | Where-Object {$_.Name -eq 'D'}).Free/1GB
>>
12.2486915588379
PS C:\Windows\System32>
```

DataDisk (D:) Properties

DataDisk

Type: Local Disk

File system: NTFS

Used space:	4,731,527,168 bytes	4.40 GB
Free space:	13,151,932,416 bytes	12.2 GB
Capacity:	17,883,459,584 bytes	16.6 GB

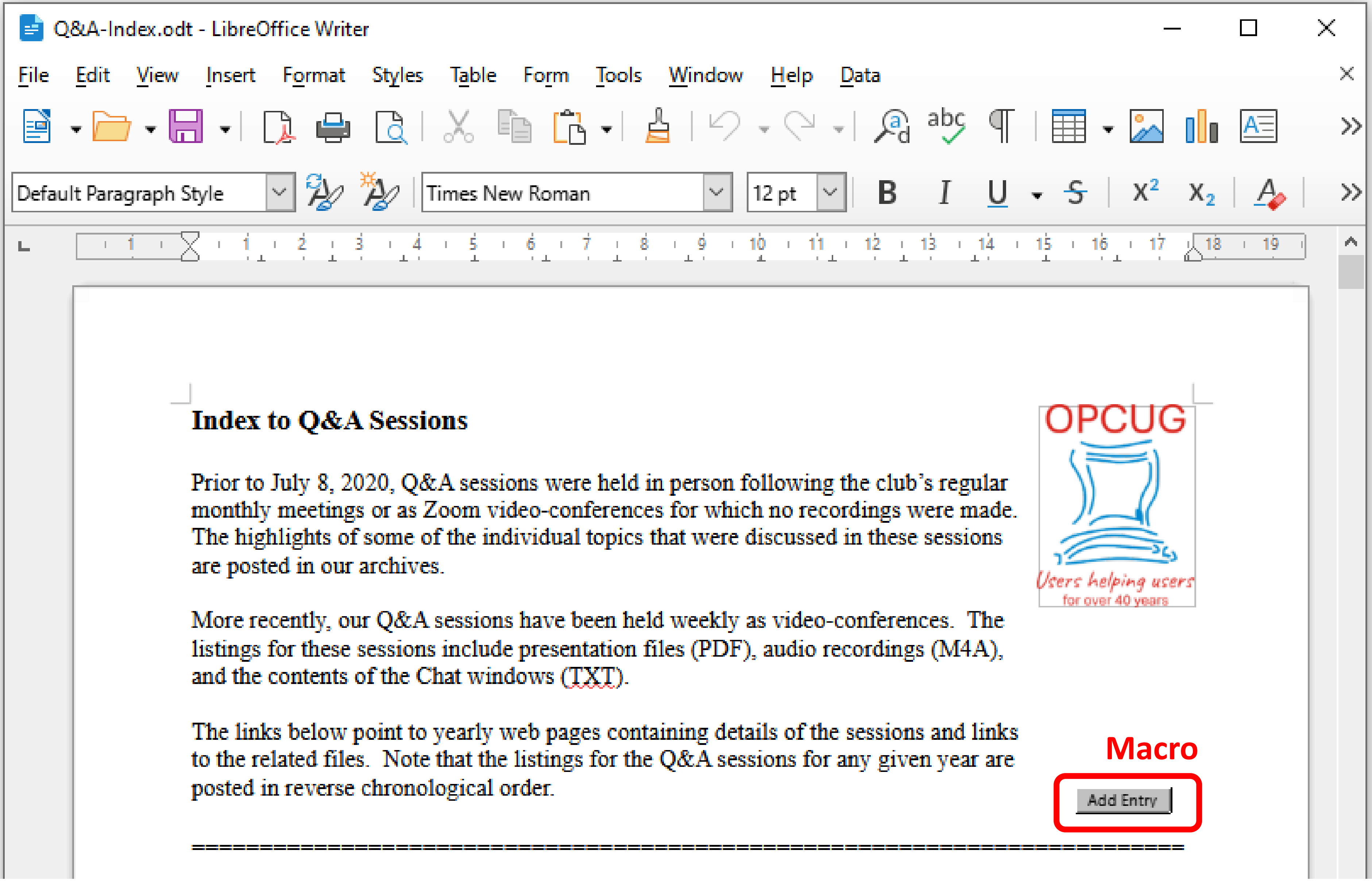
☐ Compress this drive to save disk space

☐ Allow files on this drive to have contents indexed in addition to file properties

Automatic Update to Q&A-Index.pdf

write some python code…

ChatGPT Mar 23 Version. Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts.


Q&A-Index.odt - LibreOffice Writer
File Edit View Insert Format Styles Table Form Tools Window Help Data
Default Paragraph Style
Times New Roman
12 pt
OPCUG
Users helping users
for over 40 years
Add Entry

Index to Q&A Sessions

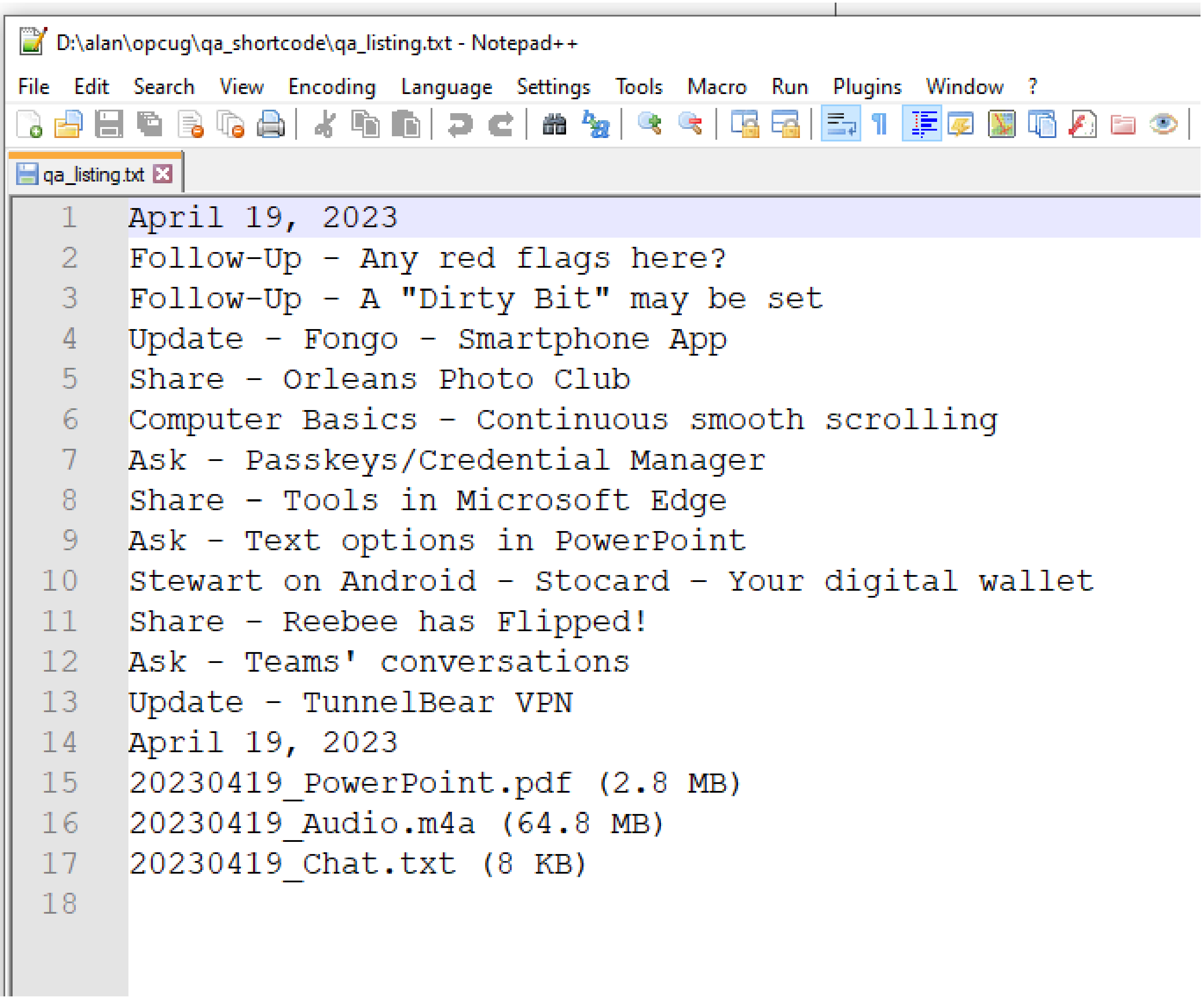
Prior to July 8, 2020, Q&A sessions were held in person following the club's regular monthly meetings or as Zoom video-conferences for which no recordings were made. The highlights of some of the individual topics that were discussed in these sessions are posted in our archives.

More recently, our Q&A sessions have been held weekly as video-conferences. The listings for these sessions include presentation files (PDF), audio recordings (M4A), and the contents of the Chat windows (TXT).

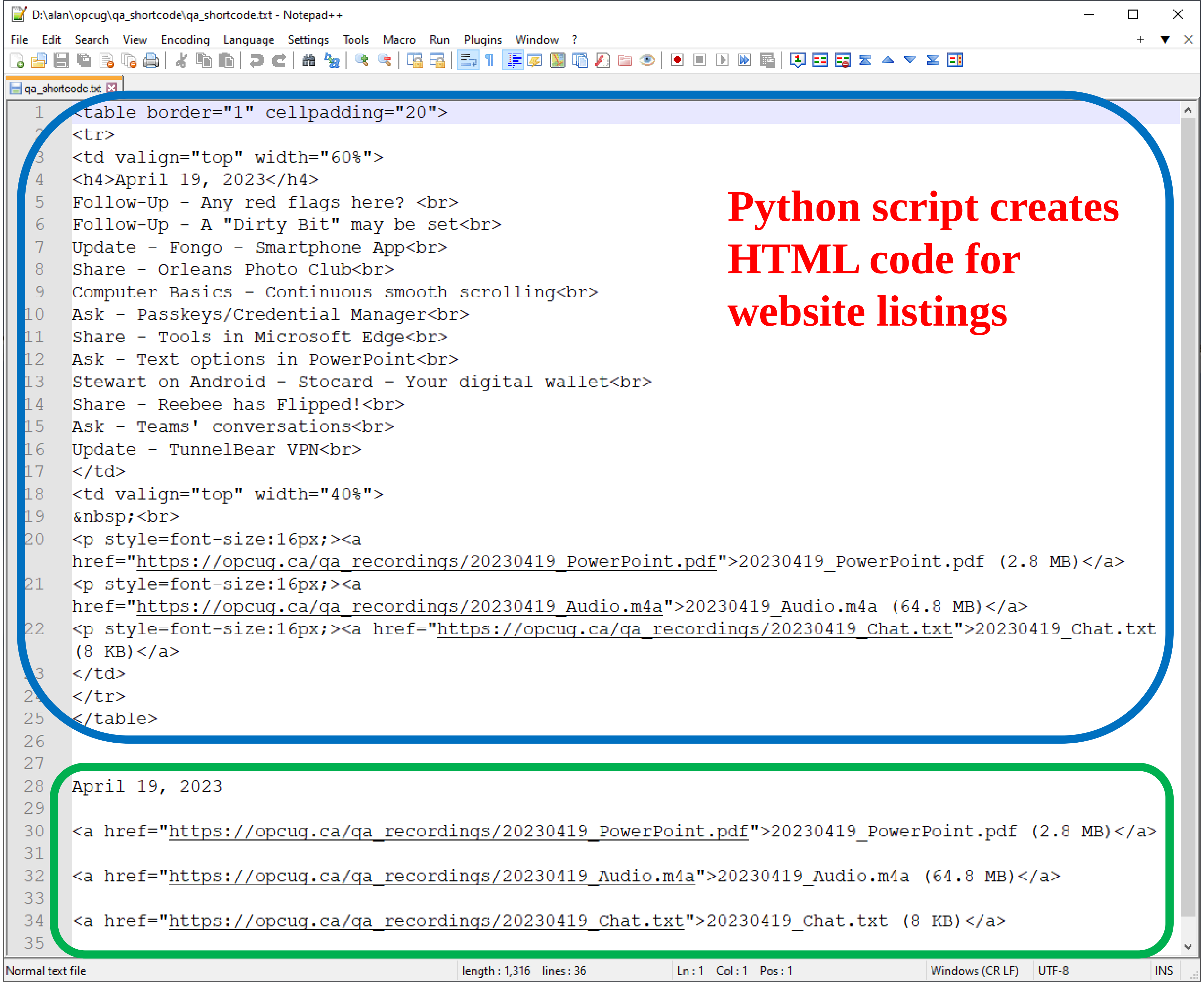
The links below point to yearly web pages containing details of the sessions and links to the related files. Note that the listings for the Q&A sessions for any given year are posted in reverse chronological order.

Macro

==

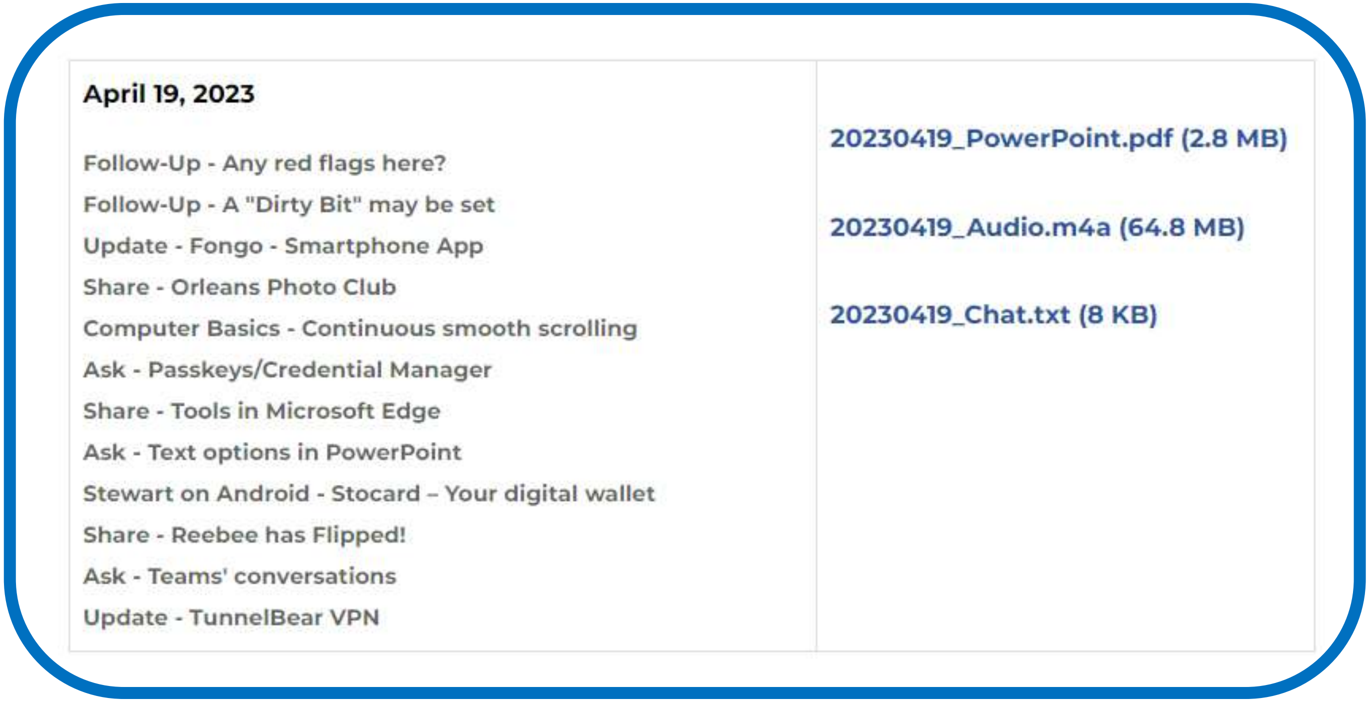


```
April 19, 2023
Follow-Up - Any red flags here?
Follow-Up - A "Dirty Bit" may be set
Update - Fongo - Smartphone App
Share - Orleans Photo Club
Computer Basics - Continuous smooth scrolling
Ask - Passkeys/Credential Manager
Share - Tools in Microsoft Edge
Ask - Text options in PowerPoint
Stewart on Android - Stocard - Your digital wallet
Share - Reebee has Flipped!
Ask - Teams' conversations
Update - TunnelBear VPN
April 19, 2023
20230419_PowerPoint.pdf (2.8 MB)
20230419_Audio.m4a (64.8 MB)
20230419_Chat.txt (8 KB)

```


D:\alan\opcug\qa_shortcode\qa_shortcode.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
qa_shortcode.txt
Normal text file
length : 1,316 lines : 36
Ln : 1 Col : 1 Pos : 1
Windows (CR LF)
UTF-8
INS

Python script creates HTML code for website listings

```
<table border="1" cellpadding="20">
<tr>
<td valign="top" width="60%">
<h4>April 19, 2023</h4>
Follow-Up - Any red flags here? <br>
Follow-Up - A "Dirty Bit" may be set<br>
Update - Fongo - Smartphone App<br>
Share - Orleans Photo Club<br>
Computer Basics - Continuous smooth scrolling<br>
Ask - Passkeys/Credential Manager<br>
Share - Tools in Microsoft Edge<br>
Ask - Text options in PowerPoint<br>
Stewart on Android - Stocard - Your digital wallet<br>
Share - Reebee has Flipped!<br>
Ask - Teams' conversations<br>
Update - TunnelBear VPN<br>
</td>
<td valign="top" width="40%">
&nbsp;<br>
<p style=font-size:16px;><a href="https://opcug.ca/qa_recordings/20230419_PowerPoint.pdf">20230419_PowerPoint.pdf (2.8 MB)</a>
<p style=font-size:16px;><a href="https://opcug.ca/qa_recordings/20230419_Audio.m4a">20230419_Audio.m4a (64.8 MB)</a>
<p style=font-size:16px;><a href="https://opcug.ca/qa_recordings/20230419_Chat.txt">20230419_Chat.txt (8 KB)</a>
</td>
</tr>
</table>


April 19, 2023

<a href="https://opcug.ca/qa_recordings/20230419_PowerPoint.pdf">20230419_PowerPoint.pdf (2.8 MB)</a>

<a href="https://opcug.ca/qa_recordings/20230419_Audio.m4a">20230419_Audio.m4a (64.8 MB)</a>

<a href="https://opcug.ca/qa_recordings/20230419_Chat.txt">20230419_Chat.txt (8 KB)</a>

```



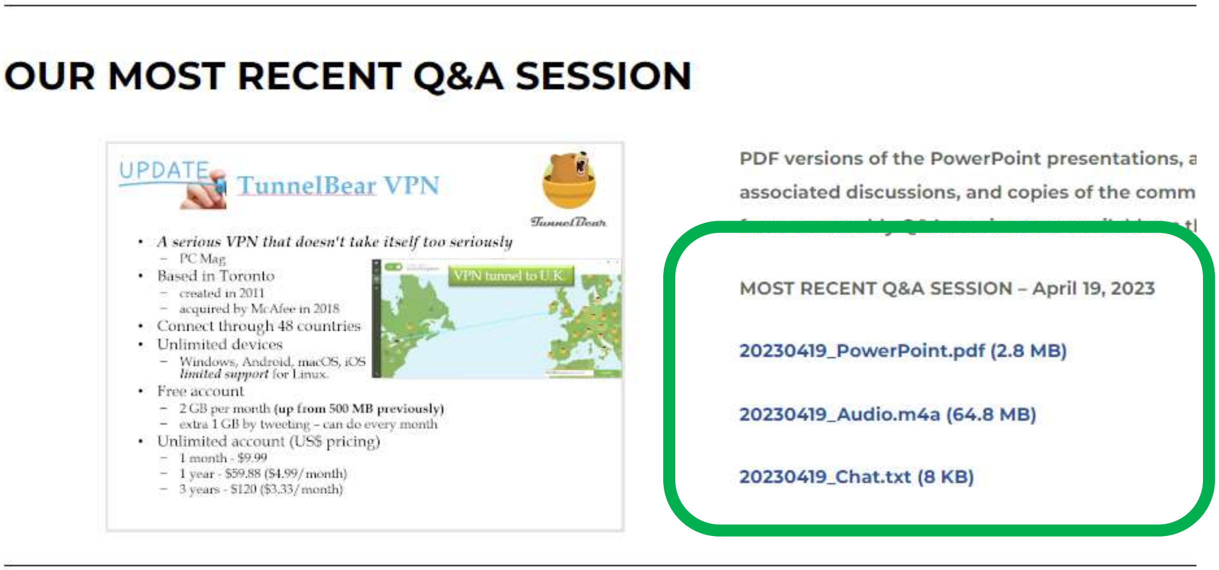
April 19, 2023

Follow-Up - Any red flags here?
Follow-Up - A "Dirty Bit" may be set
Update - Fongo - Smartphone App
Share - Orleans Photo Club
Computer Basics - Continuous smooth scrolling
Ask - Passkeys/Credential Manager
Share - Tools in Microsoft Edge
Ask - Text options in PowerPoint
Stewart on Android - Stocard – Your digital wallet
Share - Reebee has Flipped!
Ask - Teams' conversations
Update - TunnelBear VPN

20230419_PowerPoint.pdf (2.8 MB)

20230419_Audio.m4a (64.8 MB)

20230419_Chat.txt (8 KB)



OUR MOST RECENT Q&A SESSION

UPDATE TunnelBear VPN

- *A serious VPN that doesn't take itself too seriously*
  - PC Mag
- Based in Toronto
  - created in 2011
  - acquired by McAfee in 2018
- Connect through 48 countries
- Unlimited devices
  - Windows, Android, macOS, iOS
    limited support for Linux.
- Free account
  - 2 GB per month **(up from 500 MB previously)**
  - extra 1 GB by tweeting – can do every month
- Unlimited account (US$ pricing)
  - 1 month - $9.99
  - 1 year - $59.88 ($4.99/month)
  - 3 years - $120 ($3.33/month)

PDF versions of the PowerPoint presentations, a
associated discussions, and copies of the comm

MOST RECENT Q&A SESSION – April 19, 2023

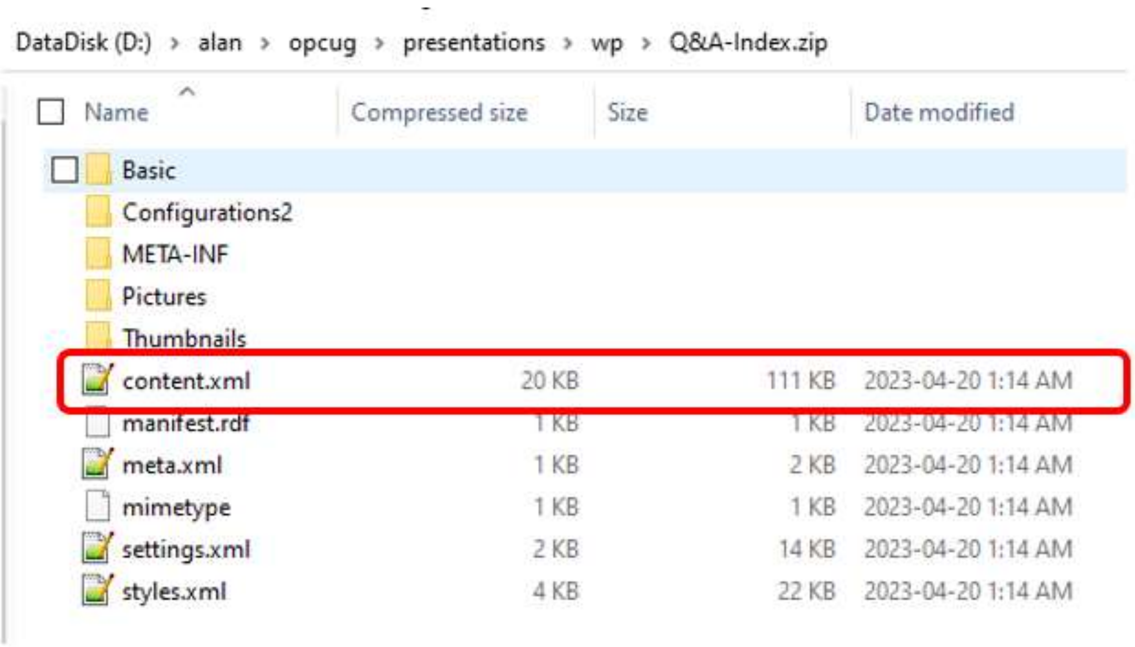
20230419_PowerPoint.pdf (2.8 MB)

20230419_Audio.m4a (64.8 MB)

20230419_Chat.txt (8 KB)

Copy the topics to Q&A-Index.odt

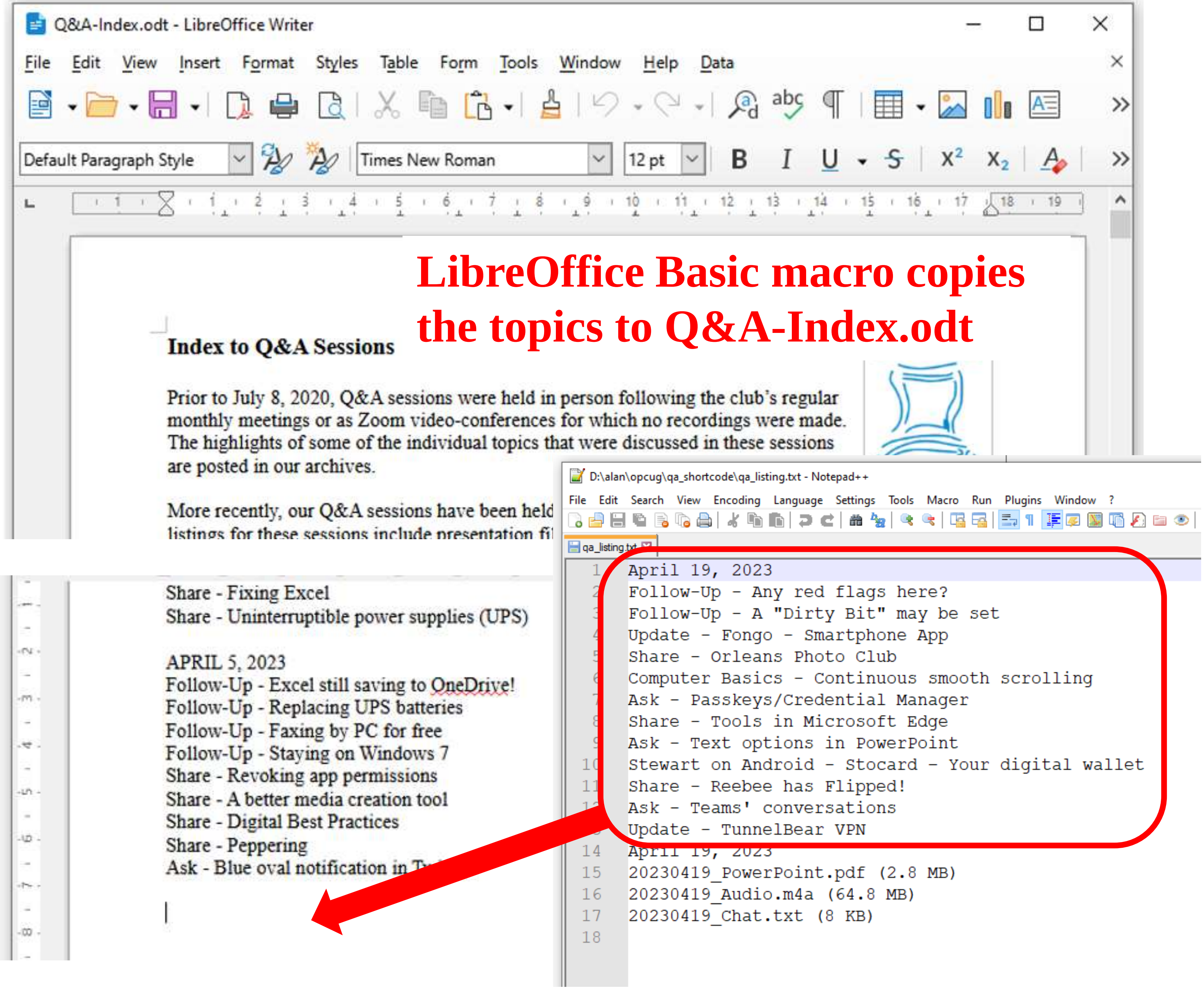
- I know how to read lines of text from a TXT file (and write these lines to a text file)
- But an ODT file is not a TXT file!
- It's a ZIP file with XML components, one of which contains the body text for the document



DataDisk (D:) › alan › opcug › presentations › wp › Q&A-Index.zip

☐ Name	Compressed size	Size	Date modified
☐ Basic			
Configurations2			
META-INF			
Pictures			
Thumbnails			
content.xml	20 KB	111 KB	2023-04-20 1:14 AM
manifest.rdf	1 KB	1 KB	2023-04-20 1:14 AM
meta.xml	1 KB	2 KB	2023-04-20 1:14 AM
mimetype	1 KB	1 KB	2023-04-20 1:14 AM
settings.xml	2 KB	14 KB	2023-04-20 1:14 AM
styles.xml	4 KB	22 KB	2023-04-20 1:14 AM

LibreOffice Basic macro copies the topics to Q&A-Index.odt


Q&A-Index.odt - LibreOffice Writer
File Edit View Insert Format Styles Table Form Tools Window Help Data
Default Paragraph Style
Times New Roman
12 pt
D:\alan\opcug\qa_shortcode\qa_listing.txt - Notepad++
File Edit Search Encoding Language Settings Tools Macro Run Plugins Window ?
qa_listing.txt

Index to Q&A Sessions

Prior to July 8, 2020, Q&A sessions were held in person following the club's regular monthly meetings or as Zoom video-conferences for which no recordings were made. The highlights of some of the individual topics that were discussed in these sessions are posted in our archives.

More recently, our Q&A sessions have been held

listings for these sessions include presentation fi

Share - Fixing Excel
Share - Uninterruptible power supplies (UPS)

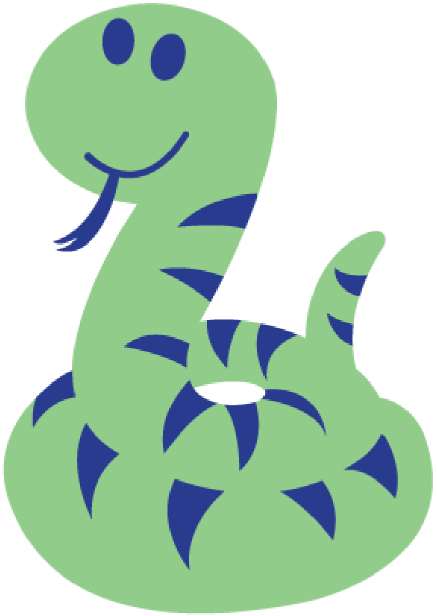
APRIL 5, 2023
Follow-Up - Excel still saving to OneDrive!
Follow-Up - Replacing UPS batteries
Follow-Up - Faxing by PC for free
Follow-Up - Staying on Windows 7
Share - Revoking app permissions
Share - A better media creation tool
Share - Digital Best Practices
Share - Peppering
Ask - Blue oval notification in T

```
April 19, 2023
Follow-Up - Any red flags here?
Follow-Up - A "Dirty Bit" may be set
Update - Fongo - Smartphone App
Share - Orleans Photo Club
Computer Basics - Continuous smooth scrolling
Ask - Passkeys/Credential Manager
Share - Tools in Microsoft Edge
Ask - Text options in PowerPoint
Stewart on Android - Stocard - Your digital wallet
Share - Reebee has Flipped!
Ask - Teams' conversations
Update - TunnelBear VPN
April 19, 2023
20230419_PowerPoint.pdf (2.8 MB)
20230419_Audio.m4a (64.8 MB)
20230419_Chat.txt (8 KB)
```

Action plan:

Create Python code to:

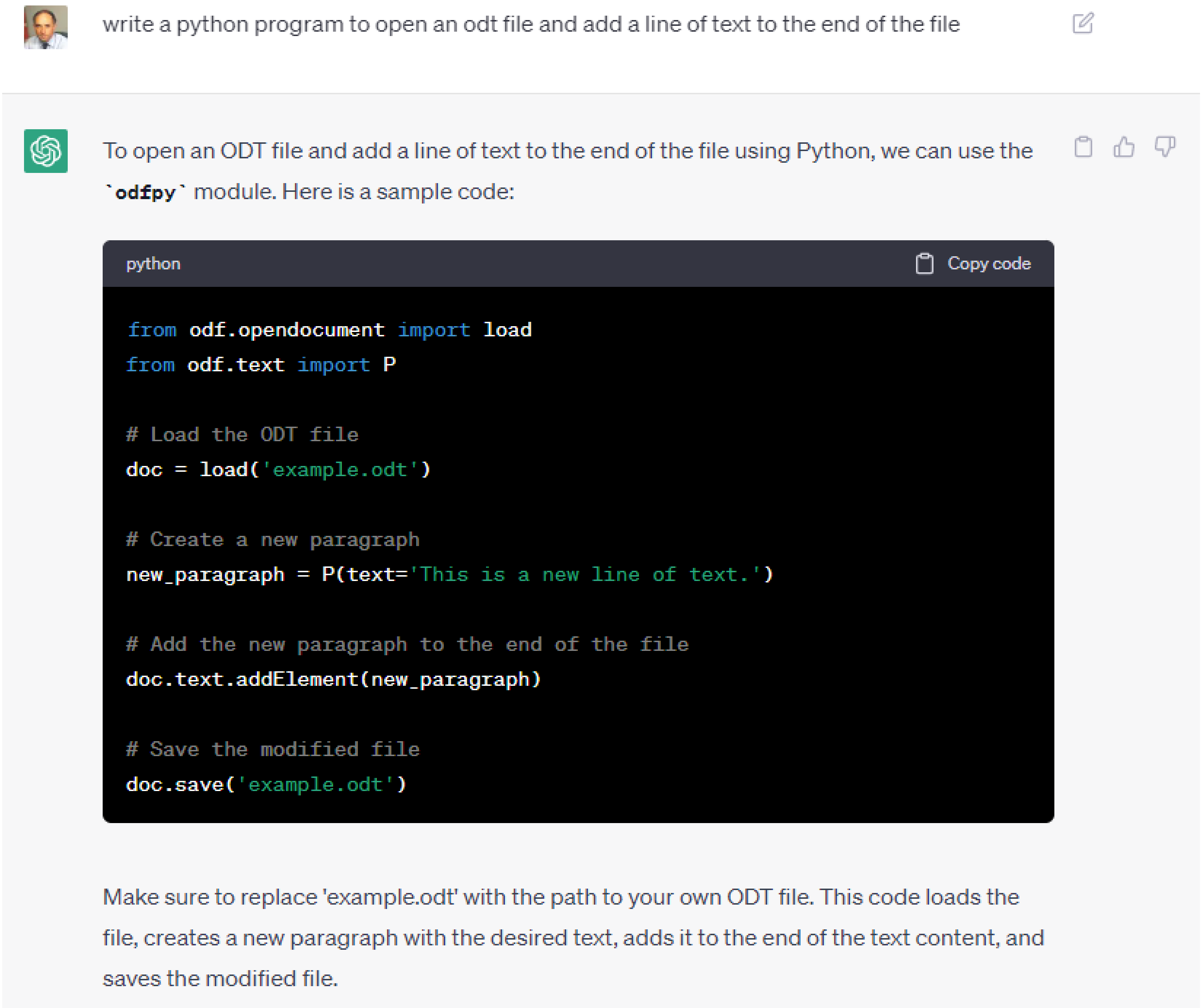
- open the ODT file
- read lines from the text file
- add these lines to the end of the ODT file
- save the ODT file
- export it to a PDF file



ChatGPT

chat.openai.com/chat

Write a python program to open an ODT file and add a line of text to the end of the file

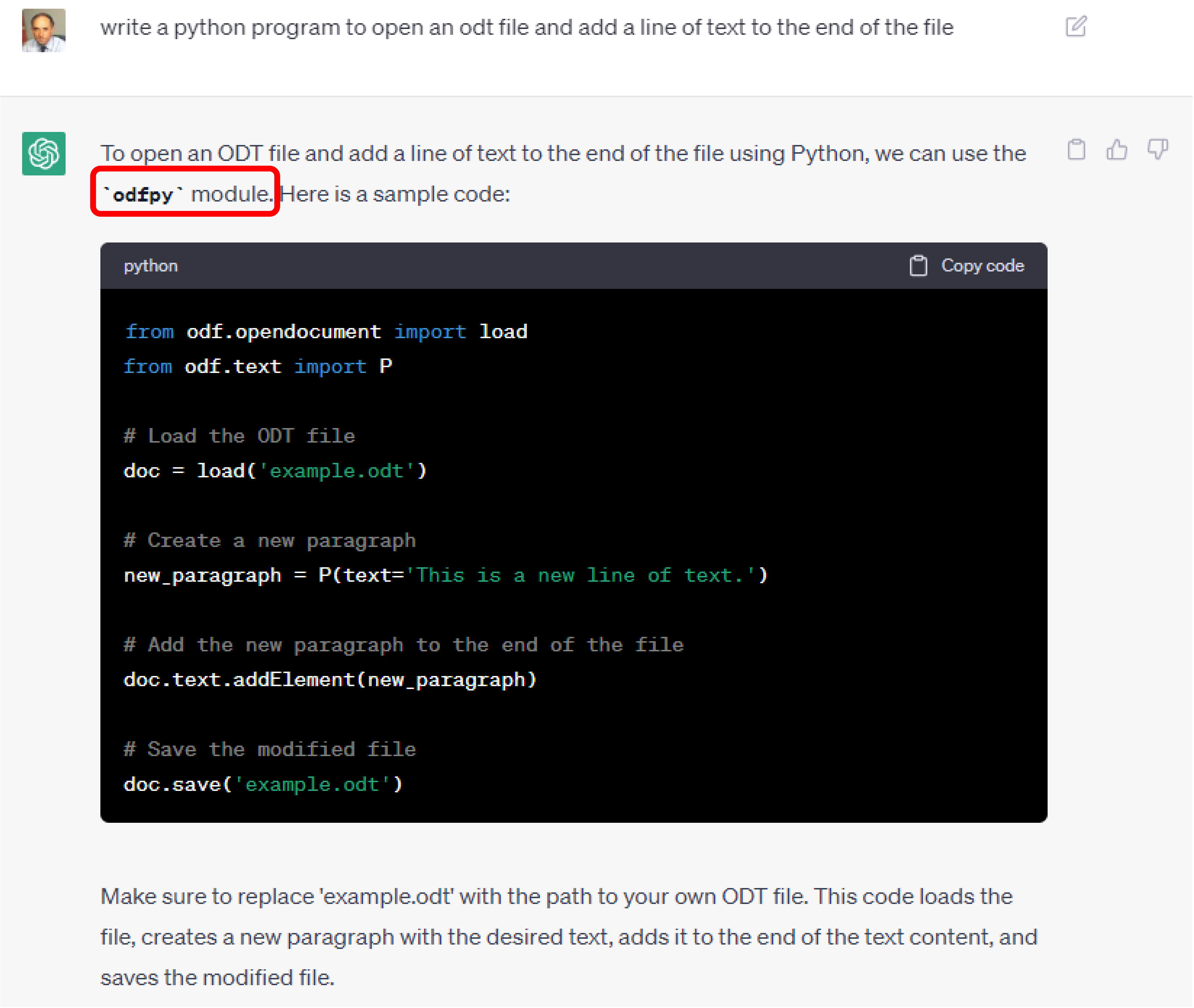


write a python program to open an odt file and add a line of text to the end of the file

To open an ODT file and add a line of text to the end of the file using Python, we can use the `odfpy` module. Here is a sample code:

```python
from odf.opendocument import load
from odf.text import P

# Load the ODT file
doc = load('example.odt')

# Create a new paragraph
new_paragraph = P(text='This is a new line of text.')

# Add the new paragraph to the end of the file
doc.text.addElement(new_paragraph)

# Save the modified file
doc.save('example.odt')
```

Make sure to replace 'example.odt' with the path to your own ODT file. This code loads the file, creates a new paragraph with the desired text, adds it to the end of the text content, and saves the modified file.

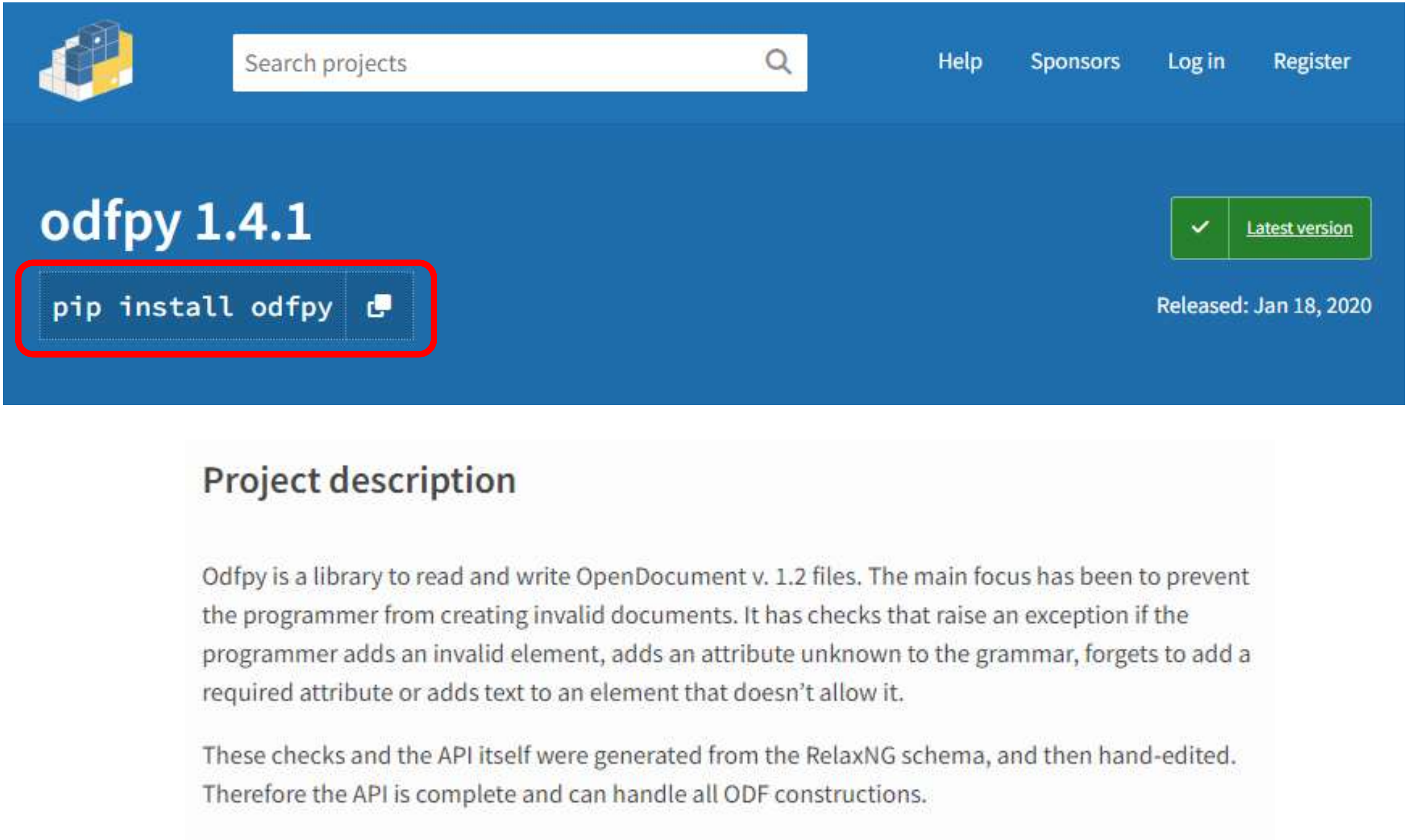


write a python program to open an odt file and add a line of text to the end of the file

To open an ODT file and add a line of text to the end of the file using Python, we can use the `odfpy` module. Here is a sample code:

```python
from odf.opendocument import load
from odf.text import P

# Load the ODT file
doc = load('example.odt')

# Create a new paragraph
new_paragraph = P(text='This is a new line of text.')

# Add the new paragraph to the end of the file
doc.text.addElement(new_paragraph)

# Save the modified file
doc.save('example.odt')
```

Make sure to replace 'example.odt' with the path to your own ODT file. This code loads the file, creates a new paragraph with the desired text, adds it to the end of the text content, and saves the modified file.



odfpy 1.4.1

```
pip install odfpy
```

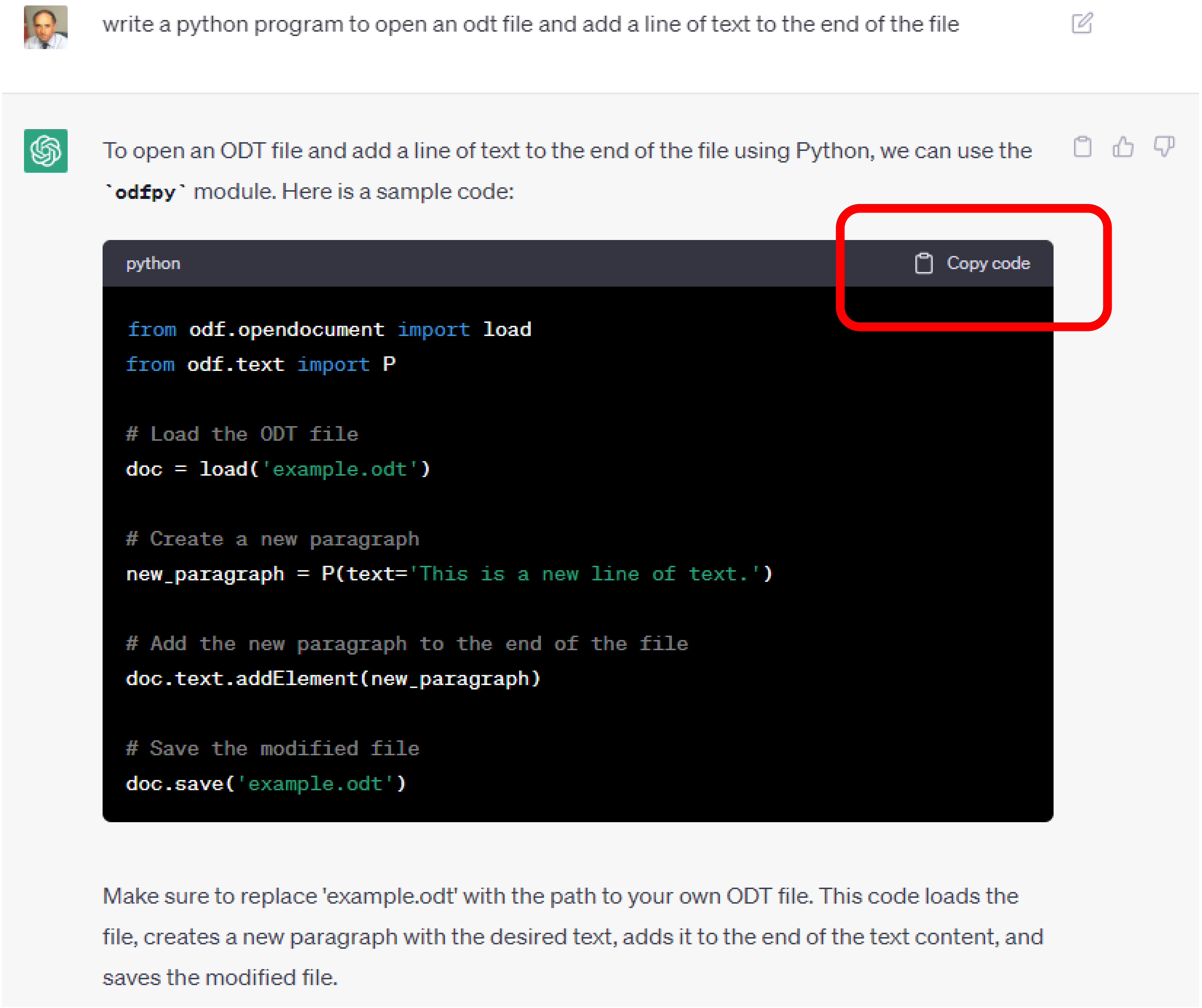
Released: Jan 18, 2020

Project description

Odfpy is a library to read and write OpenDocument v. 1.2 files. The main focus has been to prevent the programmer from creating invalid documents. It has checks that raise an exception if the programmer adds an invalid element, adds an attribute unknown to the grammar, forgets to add a required attribute or adds text to an element that doesn't allow it.

These checks and the API itself were generated from the RelaxNG schema, and then hand-edited. Therefore the API is complete and can handle all ODF constructions.

https://pypi.org/project/odfpy/



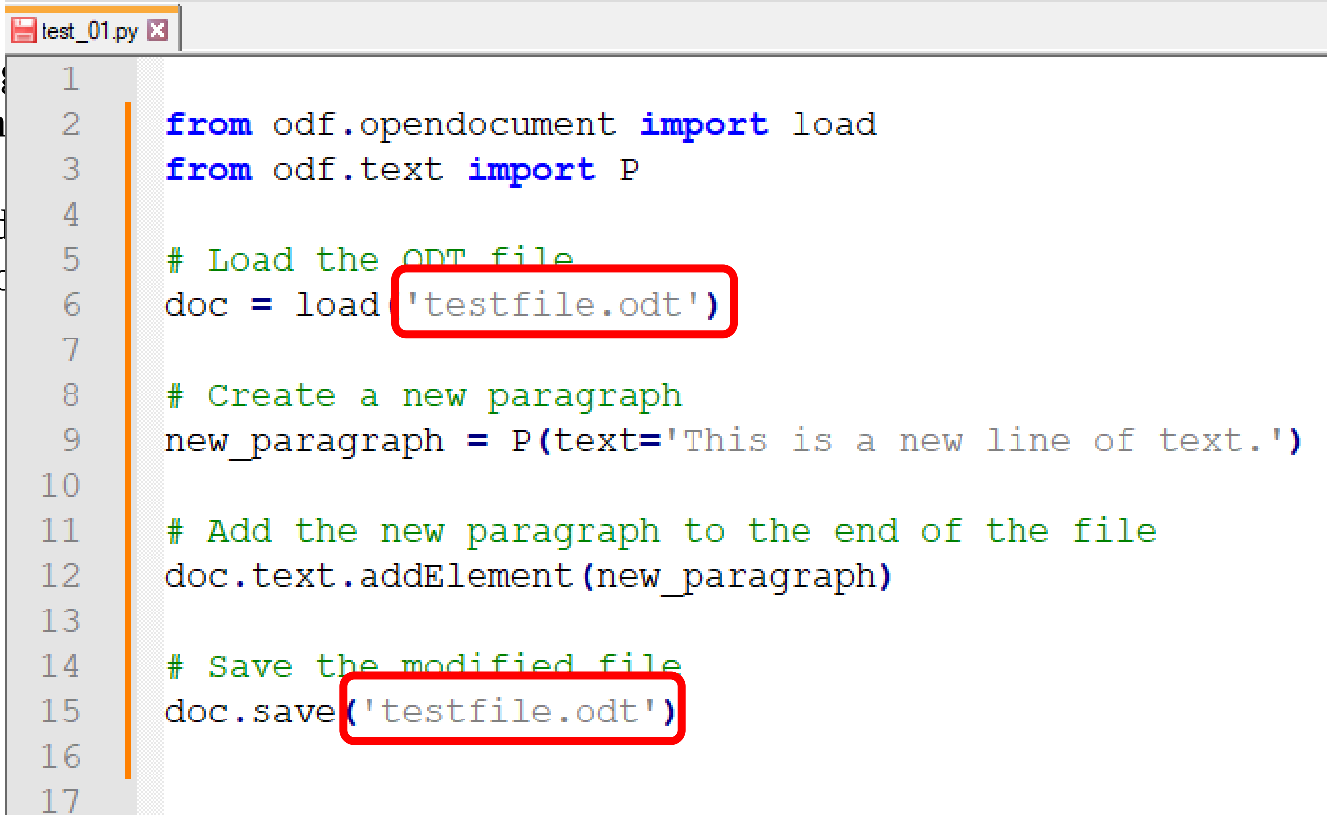
write a python program to open an odt file and add a line of text to the end of the file

To open an ODT file and add a line of text to the end of the file using Python, we can use the `odfpy` module. Here is a sample code:

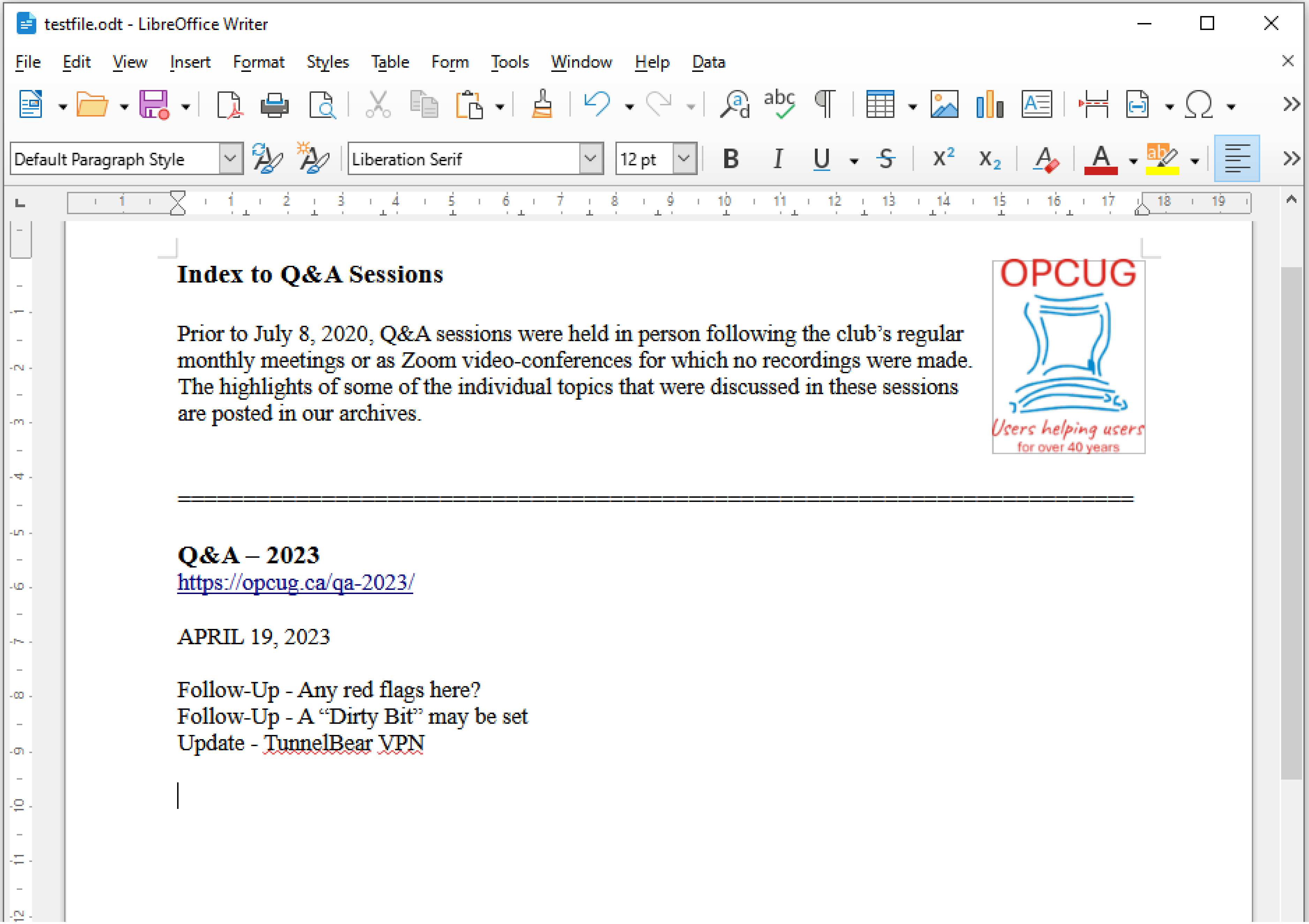
```python
from odf.opendocument import load
from odf.text import P

# Load the ODT file
doc = load('example.odt')

# Create a new paragraph
new_paragraph = P(text='This is a new line of text.')

# Add the new paragraph to the end of the file
doc.text.addElement(new_paragraph)

# Save the modified file
doc.save('example.odt')
```

Make sure to replace 'example.odt' with the path to your own ODT file. This code loads the file, creates a new paragraph with the desired text, adds it to the end of the text content, and saves the modified file.

```
from odf.opendocument import load
from odf.text import P

# Load the ODT file
doc = load('example.odt')

# Create a new paragraph
new_paragraph = P(text='This is a new line of text.')

# Add the new parag
doc.text.addElemen

# Save the modified
doc.save('example.c
```



```

from odf.opendocument import load
from odf.text import P

# Load the ODT file
doc = load('testfile.odt')

# Create a new paragraph
new_paragraph = P(text='This is a new line of text.')

# Add the new paragraph to the end of the file
doc.text.addElement(new_paragraph)

# Save the modified file
doc.save('testfile.odt')

```



Index to Q&A Sessions

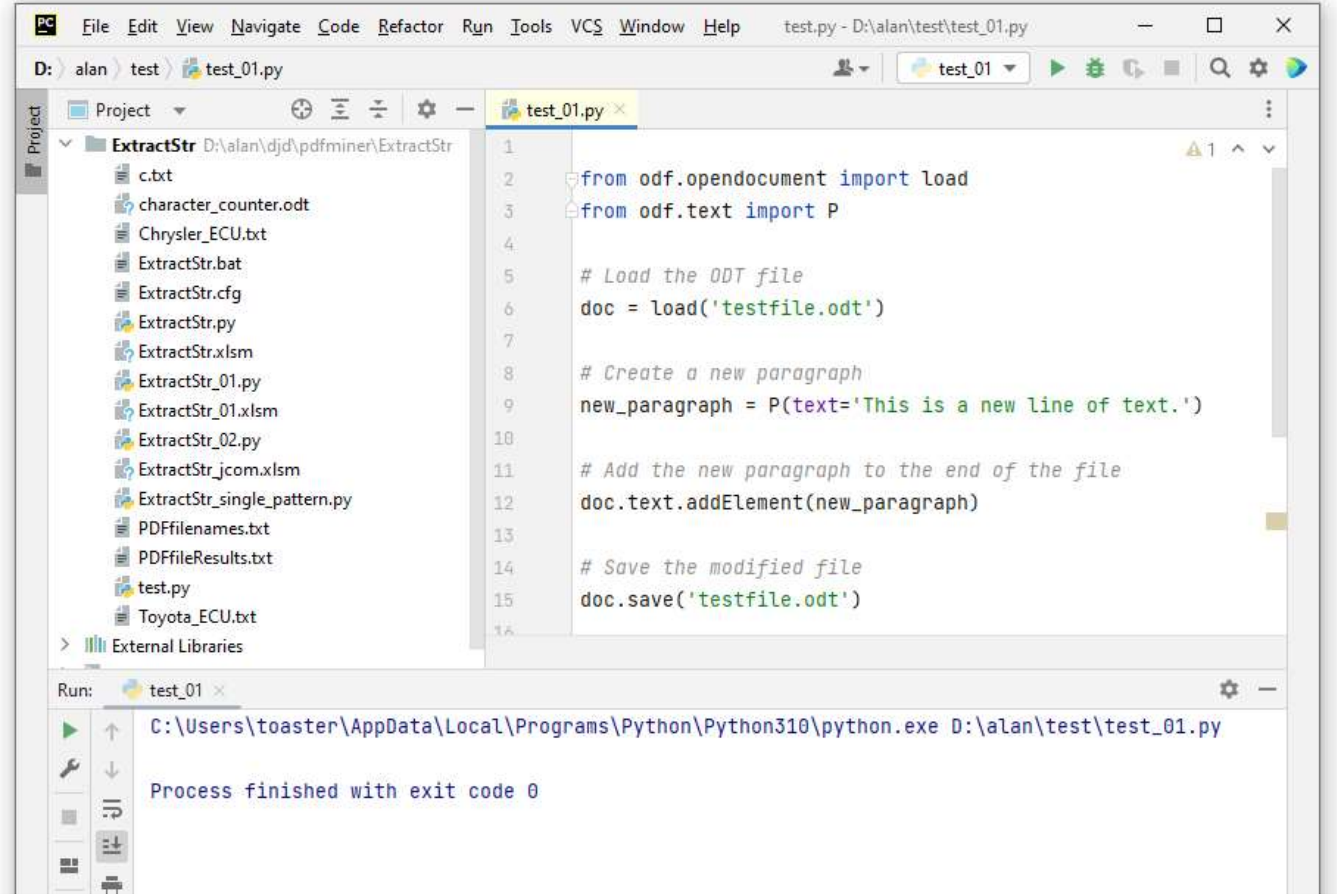
Prior to July 8, 2020, Q&A sessions were held in person following the club's regular monthly meetings or as Zoom video-conferences for which no recordings were made. The highlights of some of the individual topics that were discussed in these sessions are posted in our archives.

===

Q&A – 2023
https://opcug.ca/qa-2023/

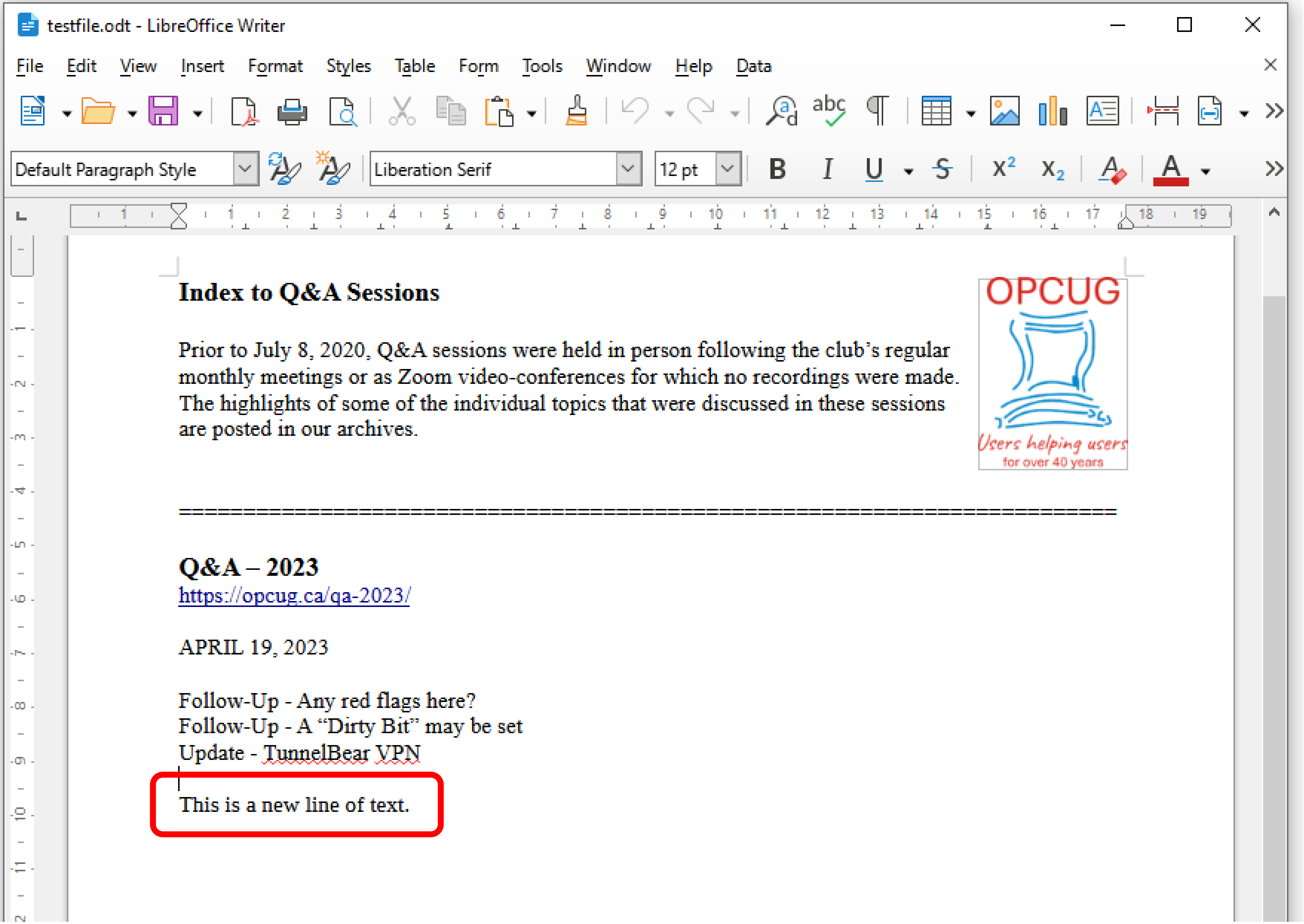
APRIL 19, 2023

Follow-Up - Any red flags here?
Follow-Up - A "Dirty Bit" may be set
Update - TunnelBear VPN



```python

from odf.opendocument import load
from odf.text import P

# Load the ODT file
doc = load('testfile.odt')

# Create a new paragraph
new_paragraph = P(text='This is a new line of text.')

# Add the new paragraph to the end of the file
doc.text.addElement(new_paragraph)

# Save the modified file
doc.save('testfile.odt')
```

```
C:\Users\toaster\AppData\Local\Programs\Python\Python310\python.exe D:\alan\test\test_01.py

Process finished with exit code 0
```



Index to Q&A Sessions

Prior to July 8, 2020, Q&A sessions were held in person following the club's regular monthly meetings or as Zoom video-conferences for which no recordings were made. The highlights of some of the individual topics that were discussed in these sessions are posted in our archives.

==

Q&A – 2023
https://opcug.ca/qa-2023/

APRIL 19, 2023

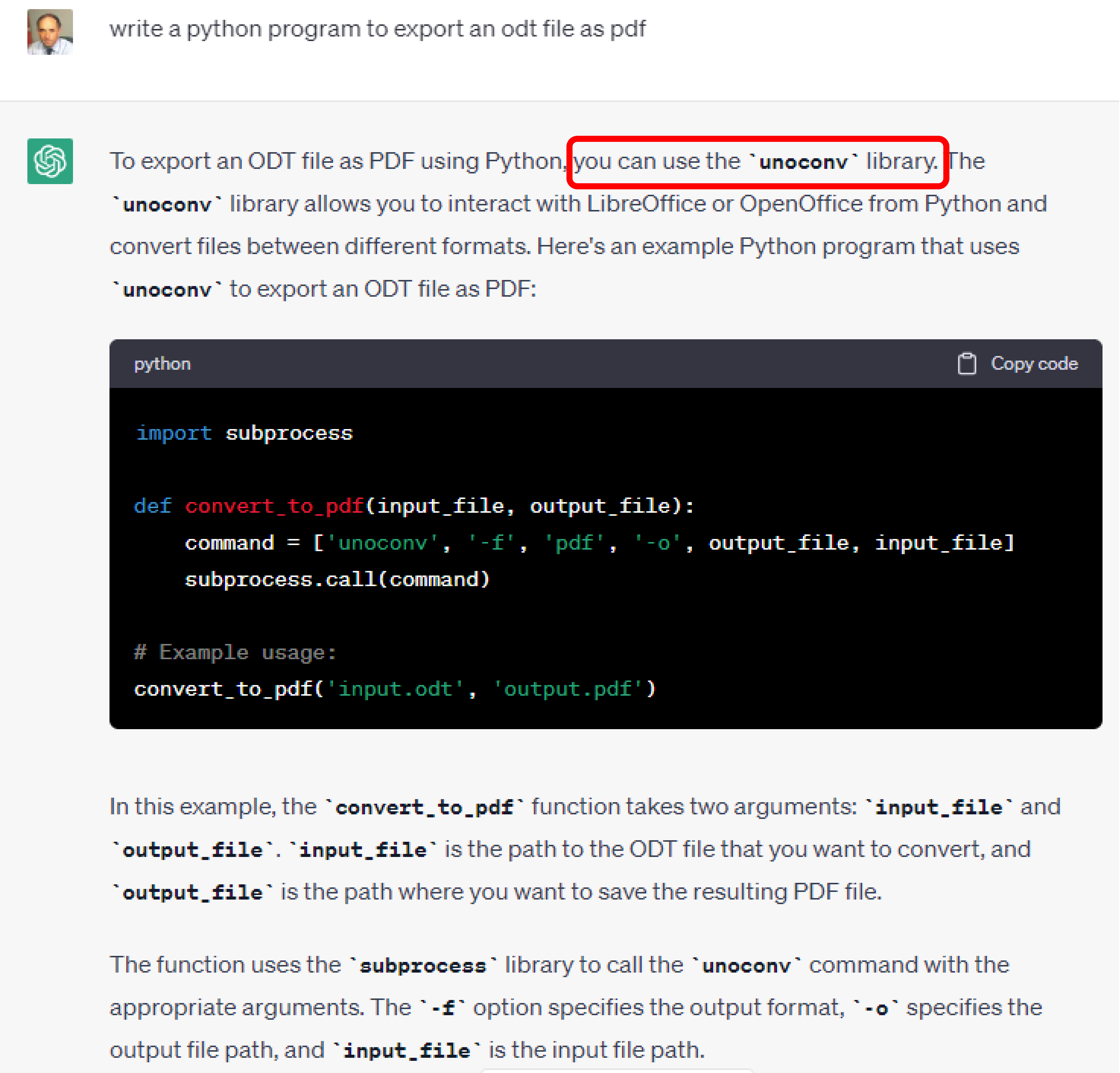
Follow-Up - Any red flags here?
Follow-Up - A "Dirty Bit" may be set
Update - TunnelBear VPN

This is a new line of text.

ChatGPT

chat.openai.com/chat

Write a python program to export an ODT file as PDF



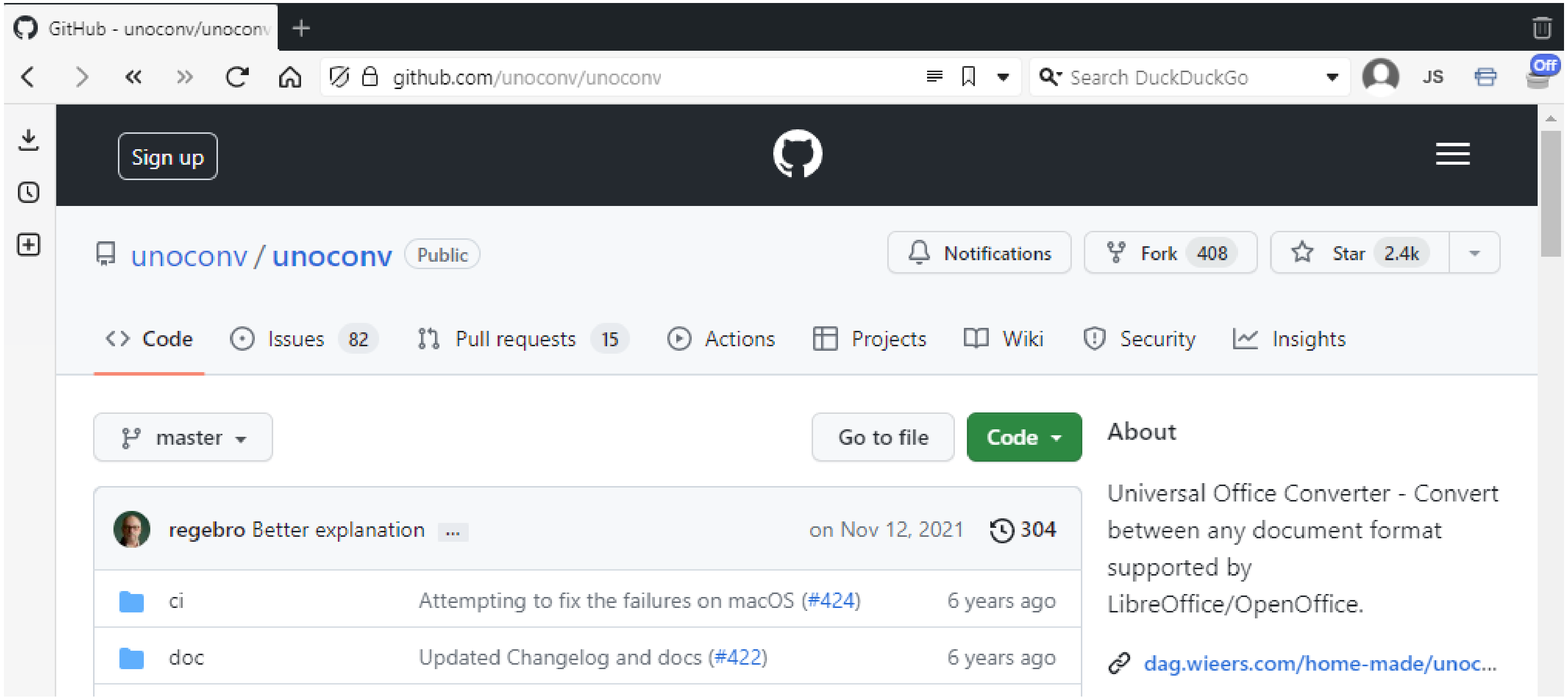
write a python program to export an odt file as pdf

To export an ODT file as PDF using Python, you can use the `unoconv` library. The `unoconv` library allows you to interact with LibreOffice or OpenOffice from Python and convert files between different formats. Here's an example Python program that uses `unoconv` to export an ODT file as PDF:

```python
import subprocess

def convert_to_pdf(input_file, output_file):
    command = ['unoconv', '-f', 'pdf', '-o', output_file, input_file]
    subprocess.call(command)

# Example usage:
convert_to_pdf('input.odt', 'output.pdf')
```

In this example, the `convert_to_pdf` function takes two arguments: `input_file` and `output_file`. `input_file` is the path to the ODT file that you want to convert, and `output_file` is the path where you want to save the resulting PDF file.

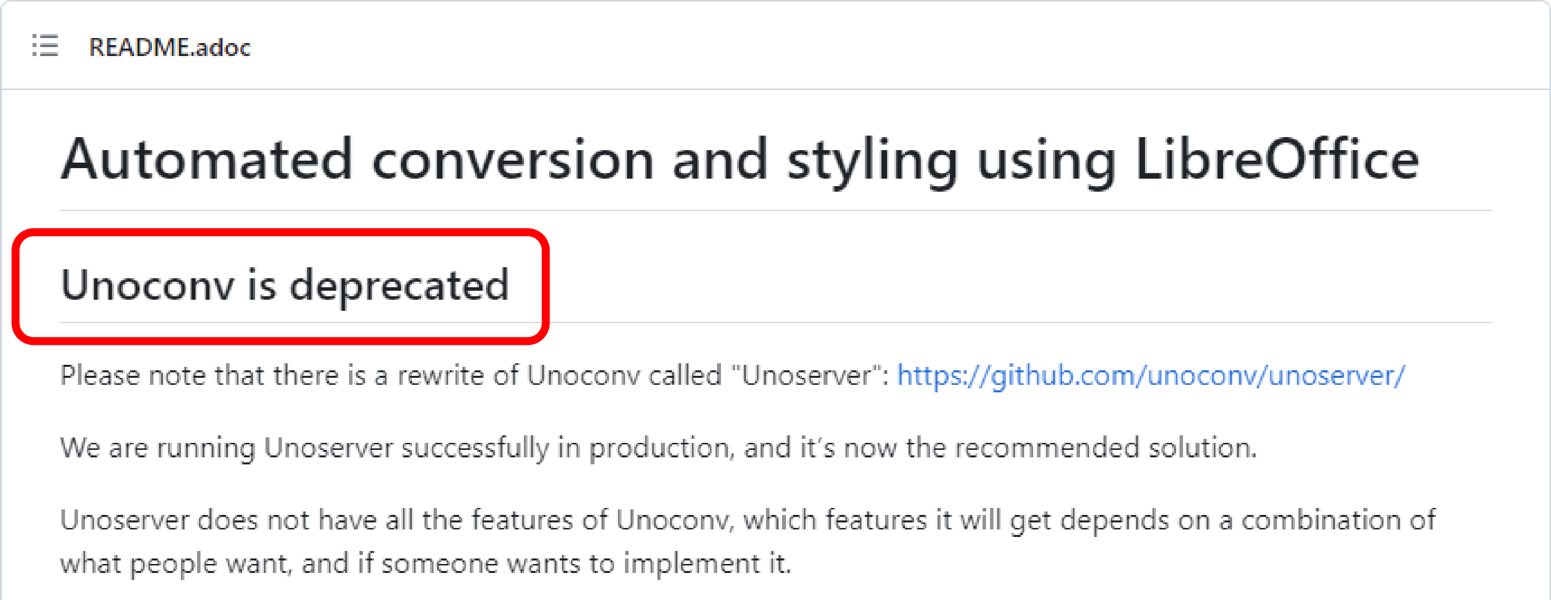
The function uses the `subprocess` library to call the `unoconv` command with the appropriate arguments. The `-f` option specifies the output format, `-o` specifies the output file path, and `input_file` is the input file path.



regebro Better explanation ...		on Nov 12, 2021 · 304
ci	Attempting to fix the failures on macOS (#424)	6 years ago
doc	Updated Changelog and docs (#422)	6 years ago

Universal Office Converter - Convert between any document format supported by LibreOffice/OpenOffice.

dag.wieers.com/home-made/unoc...



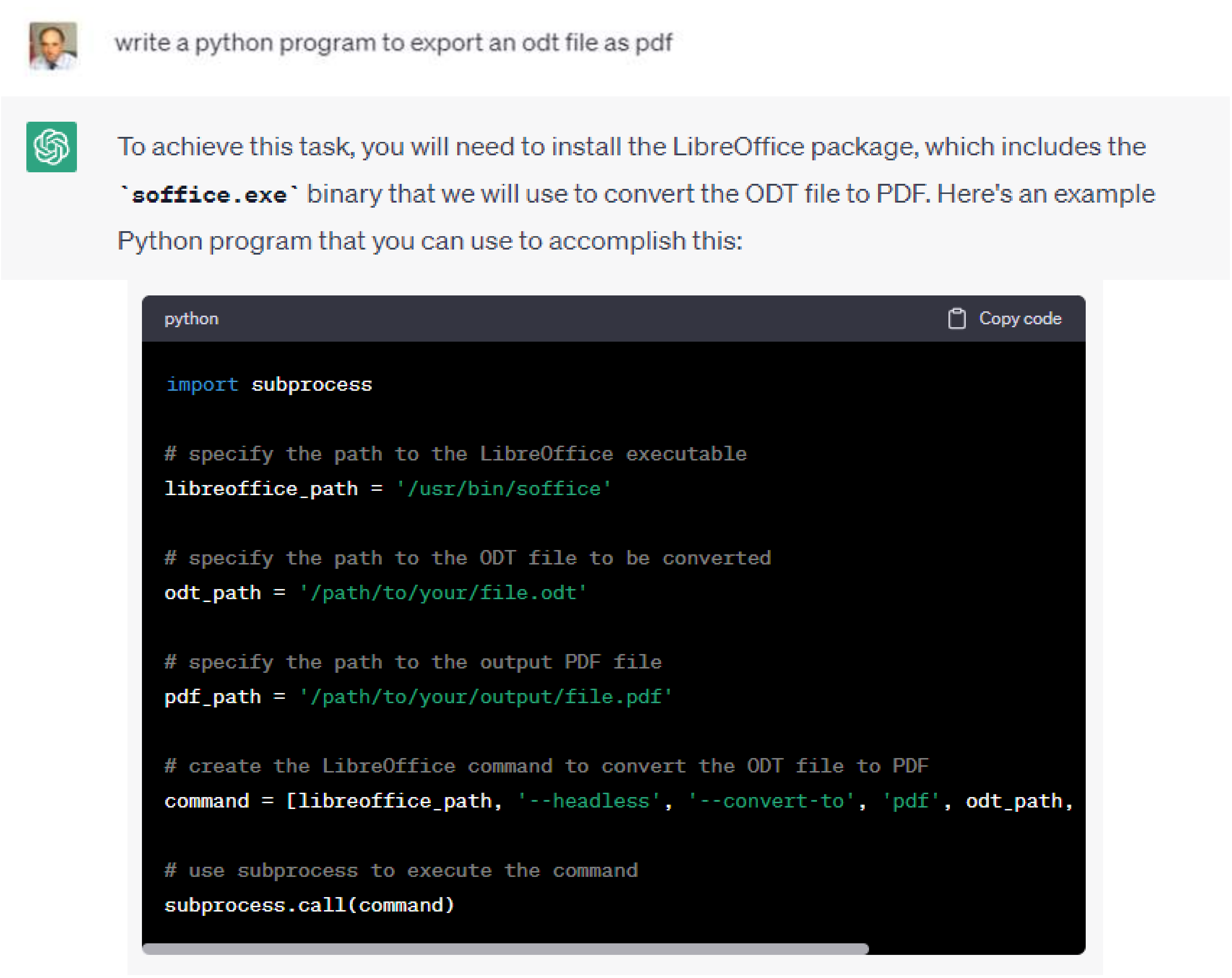
Automated conversion and styling using LibreOffice

Unoconv is deprecated

Please note that there is a rewrite of Unoconv called "Unoserver": https://github.com/unoconv/unoserver/

We are running Unoserver successfully in production, and it's now the recommended solution.

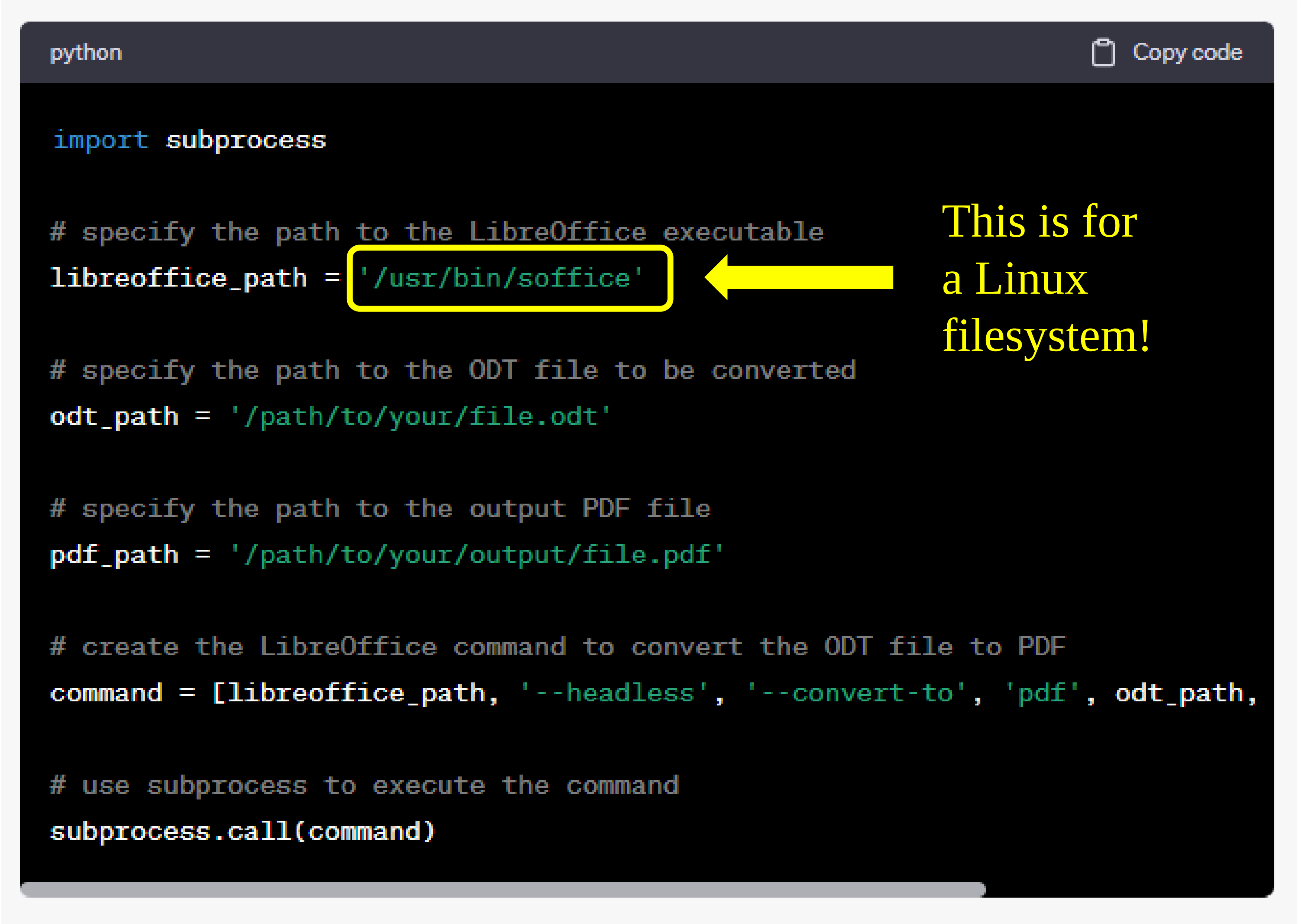
Unoserver does not have all the features of Unoconv, which features it will get depends on a combination of what people want, and if someone wants to implement it.



write a python program to export an odt file as pdf

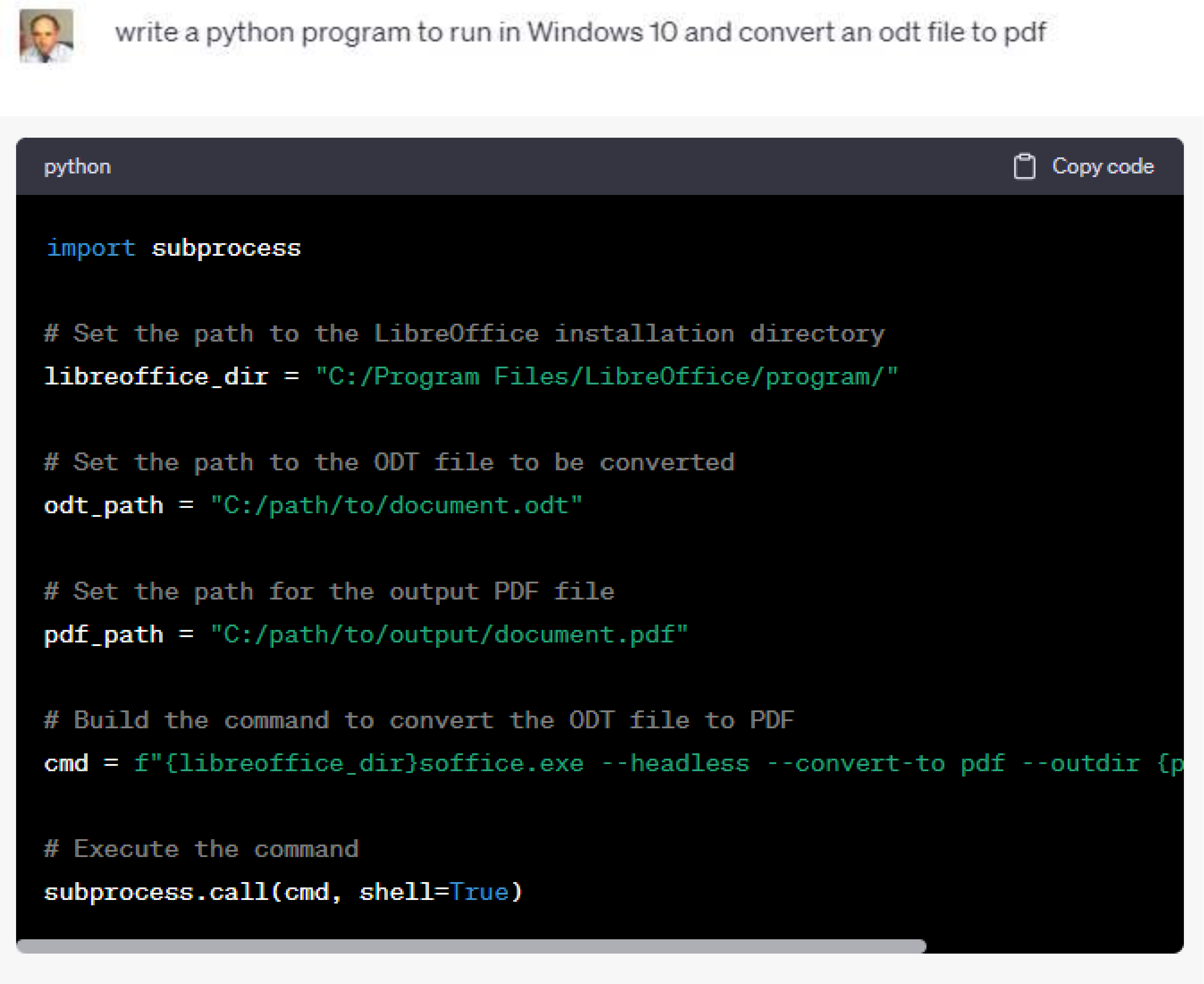
To achieve this task, you will need to install the LibreOffice package, which includes the `soffice.exe` binary that we will use to convert the ODT file to PDF. Here's an example Python program that you can use to accomplish this:

```python
import subprocess

# specify the path to the LibreOffice executable
libreoffice_path = '/usr/bin/soffice'

# specify the path to the ODT file to be converted
odt_path = '/path/to/your/file.odt'

# specify the path to the output PDF file
pdf_path = '/path/to/your/output/file.pdf'

# create the LibreOffice command to convert the ODT file to PDF
command = [libreoffice_path, '--headless', '--convert-to', 'pdf', odt_path,

# use subprocess to execute the command
subprocess.call(command)
```



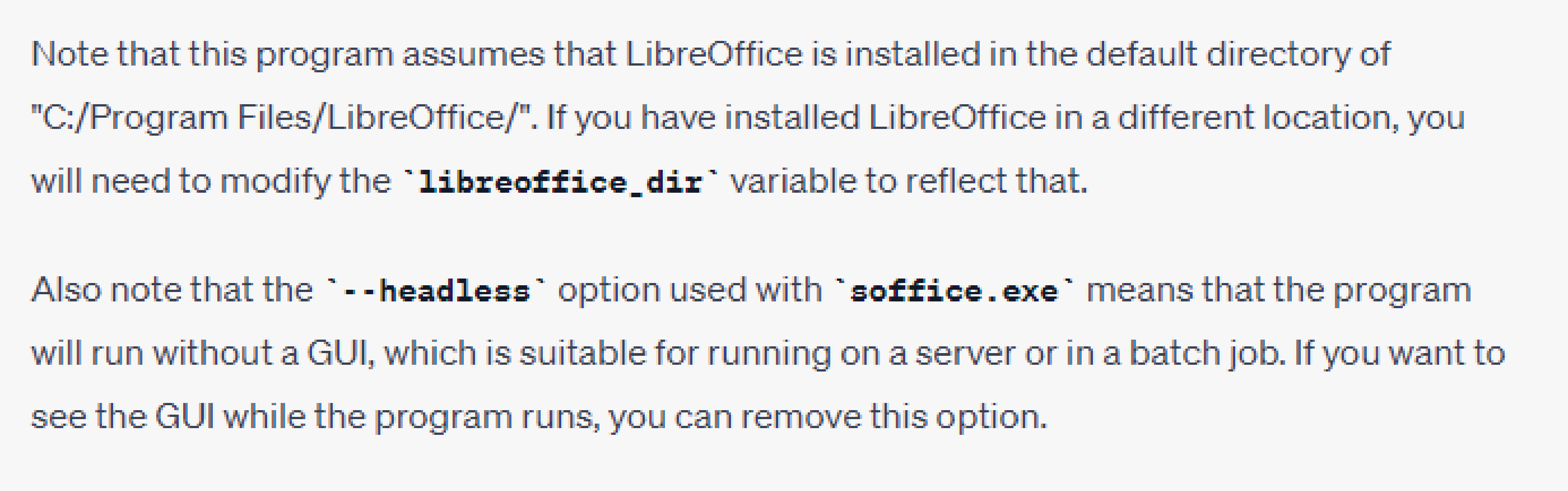
```python
import subprocess

# specify the path to the LibreOffice executable
libreoffice_path = '/usr/bin/soffice'

# specify the path to the ODT file to be converted
odt_path = '/path/to/your/file.odt'

# specify the path to the output PDF file
pdf_path = '/path/to/your/output/file.pdf'

# create the LibreOffice command to convert the ODT file to PDF
command = [libreoffice_path, '--headless', '--convert-to', 'pdf', odt_path,

# use subprocess to execute the command
subprocess.call(command)
```

This is for
a Linux
filesystem!



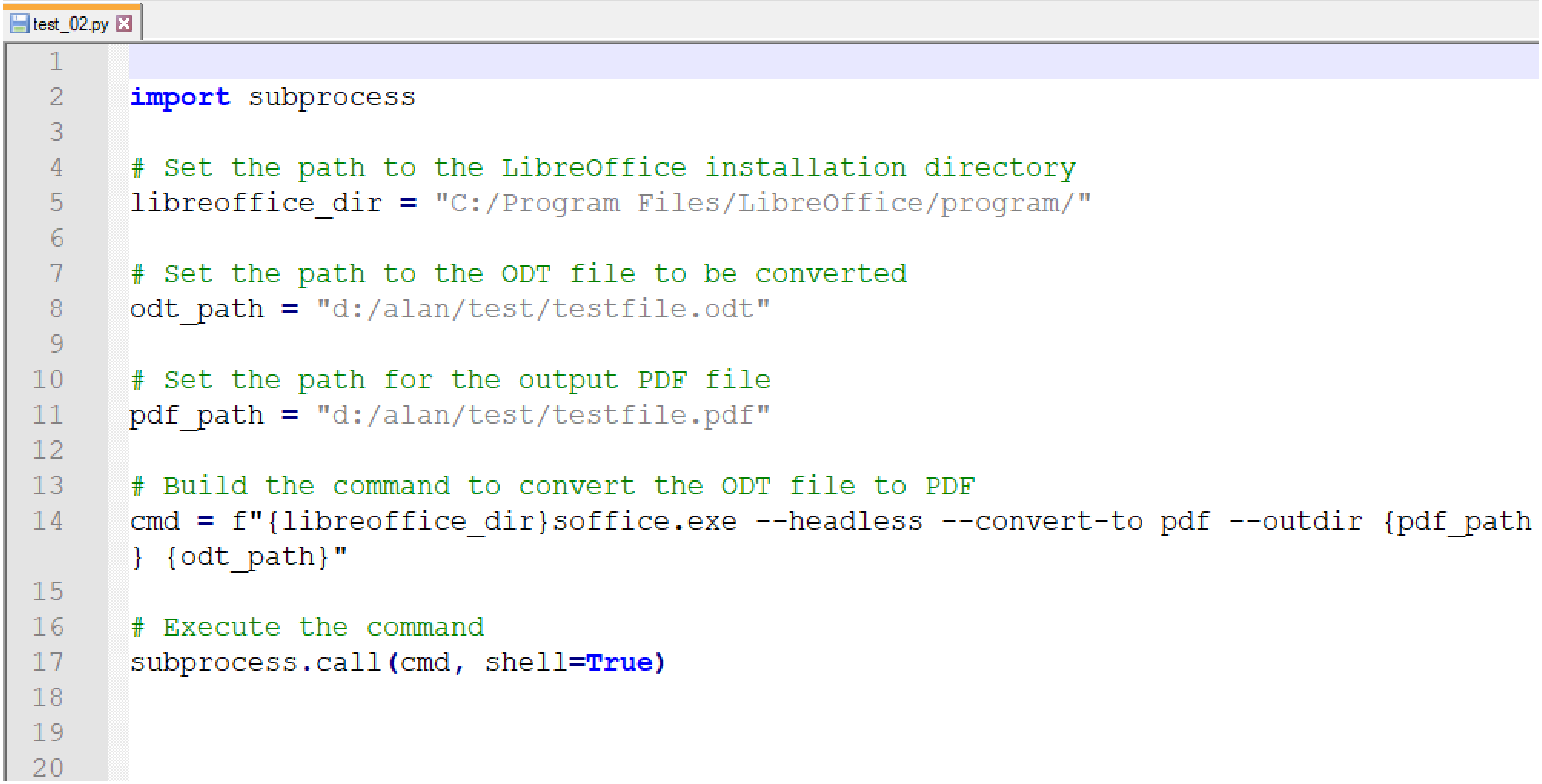
write a python program to run in Windows 10 and convert an odt file to pdf

```python
import subprocess

# Set the path to the LibreOffice installation directory
libreoffice_dir = "C:/Program Files/LibreOffice/program/"

# Set the path to the ODT file to be converted
odt_path = "C:/path/to/document.odt"

# Set the path for the output PDF file
pdf_path = "C:/path/to/output/document.pdf"

# Build the command to convert the ODT file to PDF
cmd = f"{libreoffice_dir}soffice.exe --headless --convert-to pdf --outdir {p

# Execute the command
subprocess.call(cmd, shell=True)
```



Note that this program assumes that LibreOffice is installed in the default directory of "C:/Program Files/LibreOffice/". If you have installed LibreOffice in a different location, you will need to modify the **`libreoffice_dir`** variable to reflect that.

Also note that the **`--headless`** option used with **`soffice.exe`** means that the program will run without a GUI, which is suitable for running on a server or in a batch job. If you want to see the GUI while the program runs, you can remove this option.



```python

import subprocess

# Set the path to the LibreOffice installation directory
libreoffice_dir = "C:/Program Files/LibreOffice/program/"

# Set the path to the ODT file to be converted
odt_path = "d:/alan/test/testfile.odt"

# Set the path for the output PDF file
pdf_path = "d:/alan/test/testfile.pdf"

# Build the command to convert the ODT file to PDF
cmd = f"{libreoffice_dir}soffice.exe --headless --convert-to pdf --outdir {pdf_path} {odt_path}"

# Execute the command
subprocess.call(cmd, shell=True)
```

Build the command to run LibreOffice:

```
{libreoffice_dir}soffice.exe --headless --convert-to pdf
 --outdir {pdf_path} {odt_path}
```

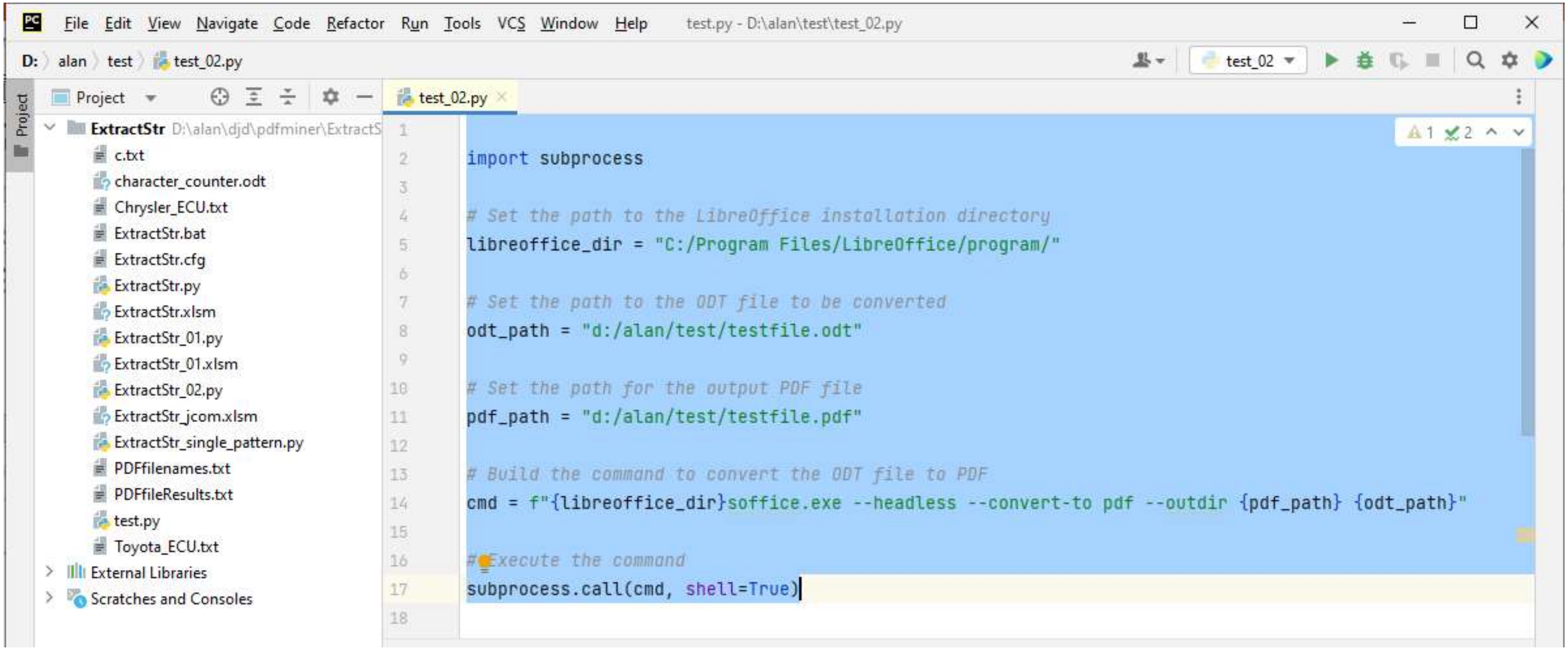
{libreoffice_dir} = Path to soffice.exe

--headless = Run LibreOffice with no user interface

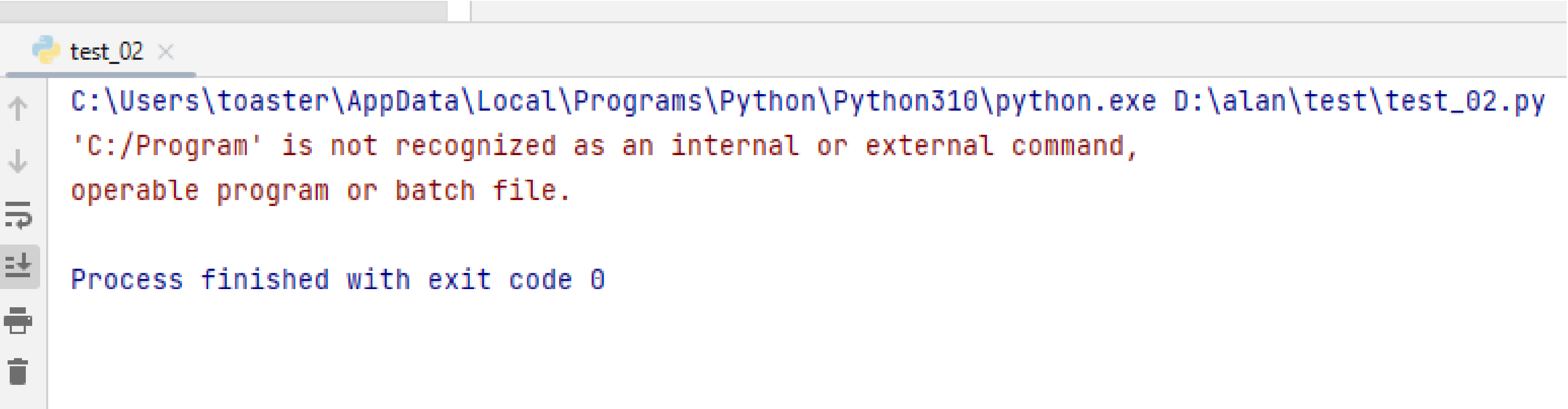
--convert-to pdf = Convert source file to PDF

--outdir {pdf_path} = (*Optional*) Path to save PDF file

{odt_path} = Path to source ODT file

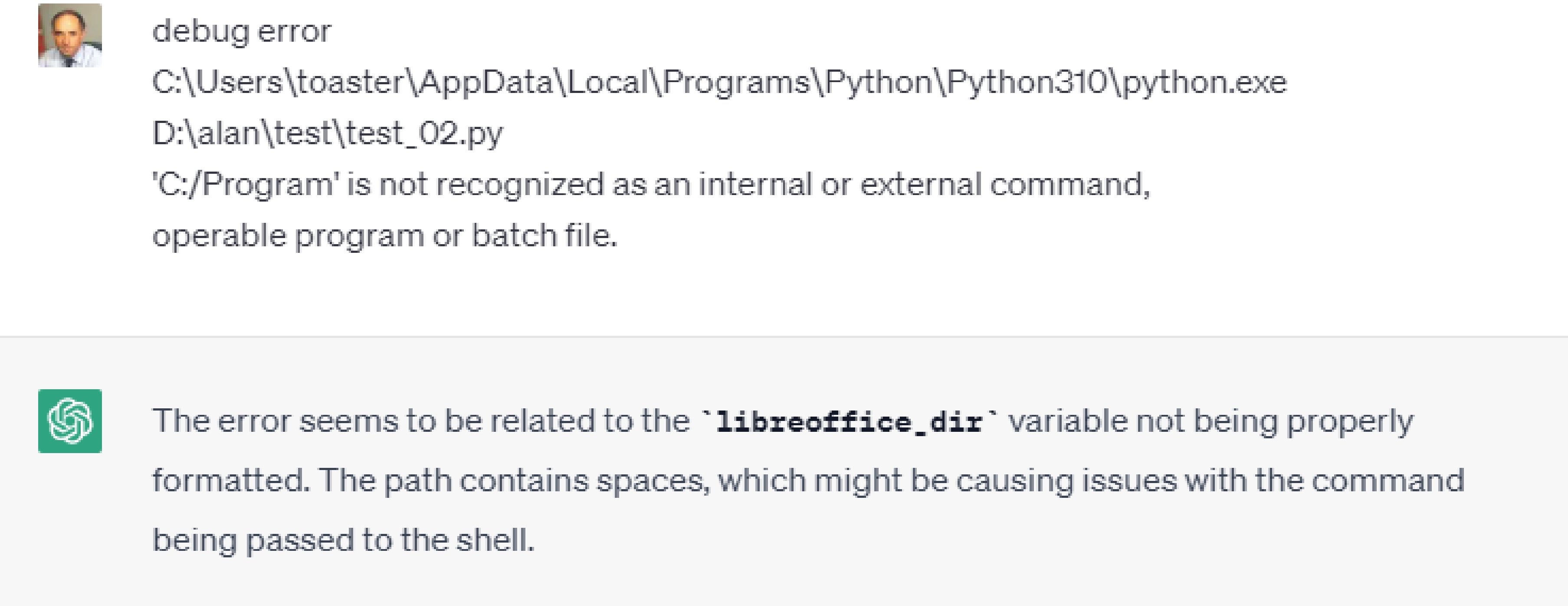


```python

import subprocess

# Set the path to the LibreOffice installation directory
libreoffice_dir = "C:/Program Files/LibreOffice/program/"

# Set the path to the ODT file to be converted
odt_path = "d:/alan/test/testfile.odt"

# Set the path for the output PDF file
pdf_path = "d:/alan/test/testfile.pdf"

# Build the command to convert the ODT file to PDF
cmd = f"{libreoffice_dir}soffice.exe --headless --convert-to pdf --outdir {pdf_path} {odt_path}"

# Execute the command
subprocess.call(cmd, shell=True)
```



```
C:\Users\toaster\AppData\Local\Programs\Python\Python310\python.exe D:\alan\test\test_02.py
'C:/Program' is not recognized as an internal or external command,
operable program or batch file.

Process finished with exit code 0
```

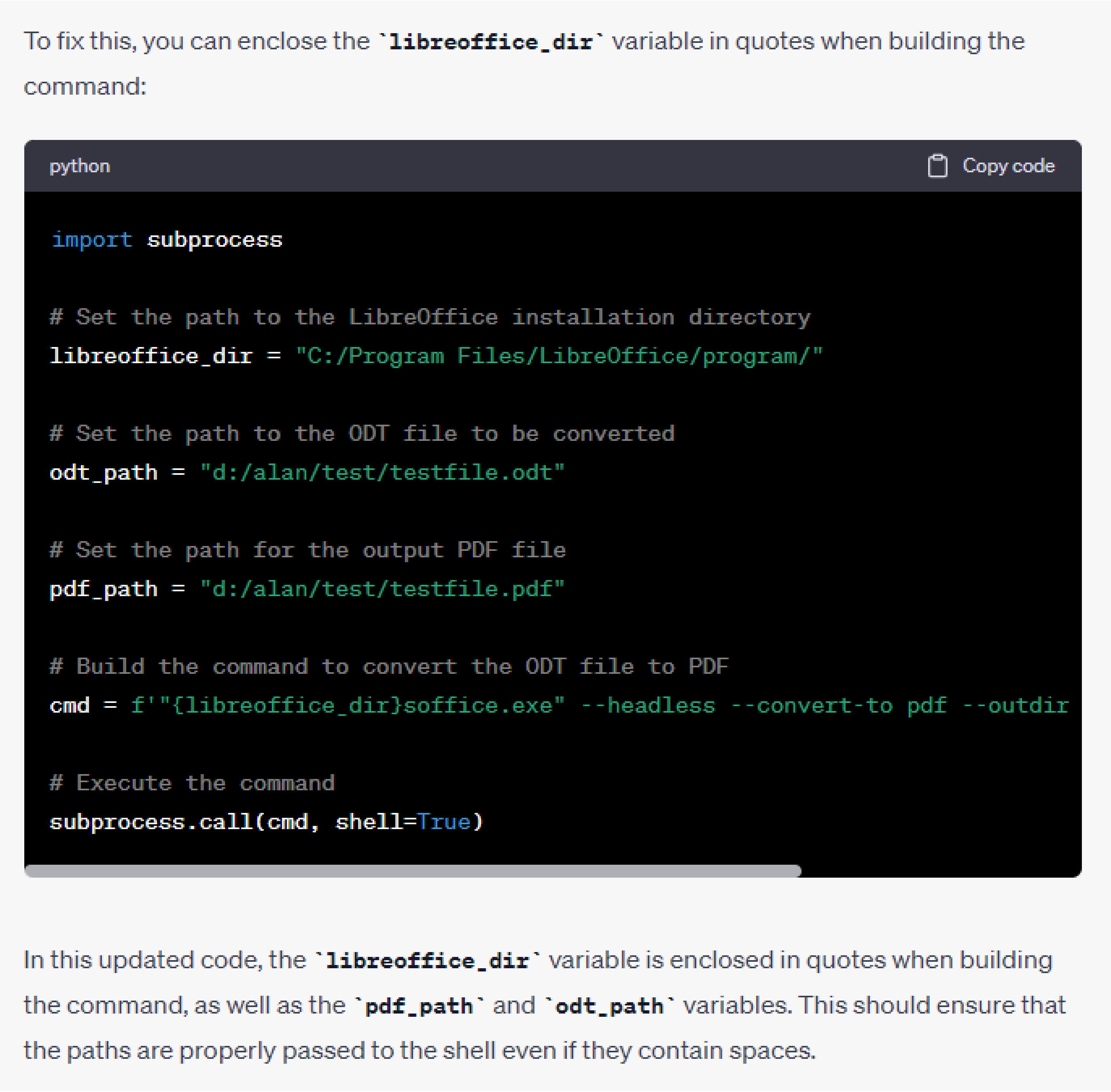
Another Use for ChatGPT – Debug!



debug error

```
C:\Users\toaster\AppData\Local\Programs\Python\Python310\python.exe D:\alan\test\test_02.py
'C:/Program' is not recognized as an internal or external command,
operable program or batch file.
```

The error seems to be related to the `` `libreoffice_dir` `` variable not being properly formatted. The path contains spaces, which might be causing issues with the command being passed to the shell.



To fix this, you can enclose the `libreoffice_dir` variable in quotes when building the command:

```python
import subprocess

# Set the path to the LibreOffice installation directory
libreoffice_dir = "C:/Program Files/LibreOffice/program/"

# Set the path to the ODT file to be converted
odt_path = "d:/alan/test/testfile.odt"

# Set the path for the output PDF file
pdf_path = "d:/alan/test/testfile.pdf"

# Build the command to convert the ODT file to PDF
cmd = f'"{libreoffice_dir}soffice.exe" --headless --convert-to pdf --outdir

# Execute the command
subprocess.call(cmd, shell=True)
```

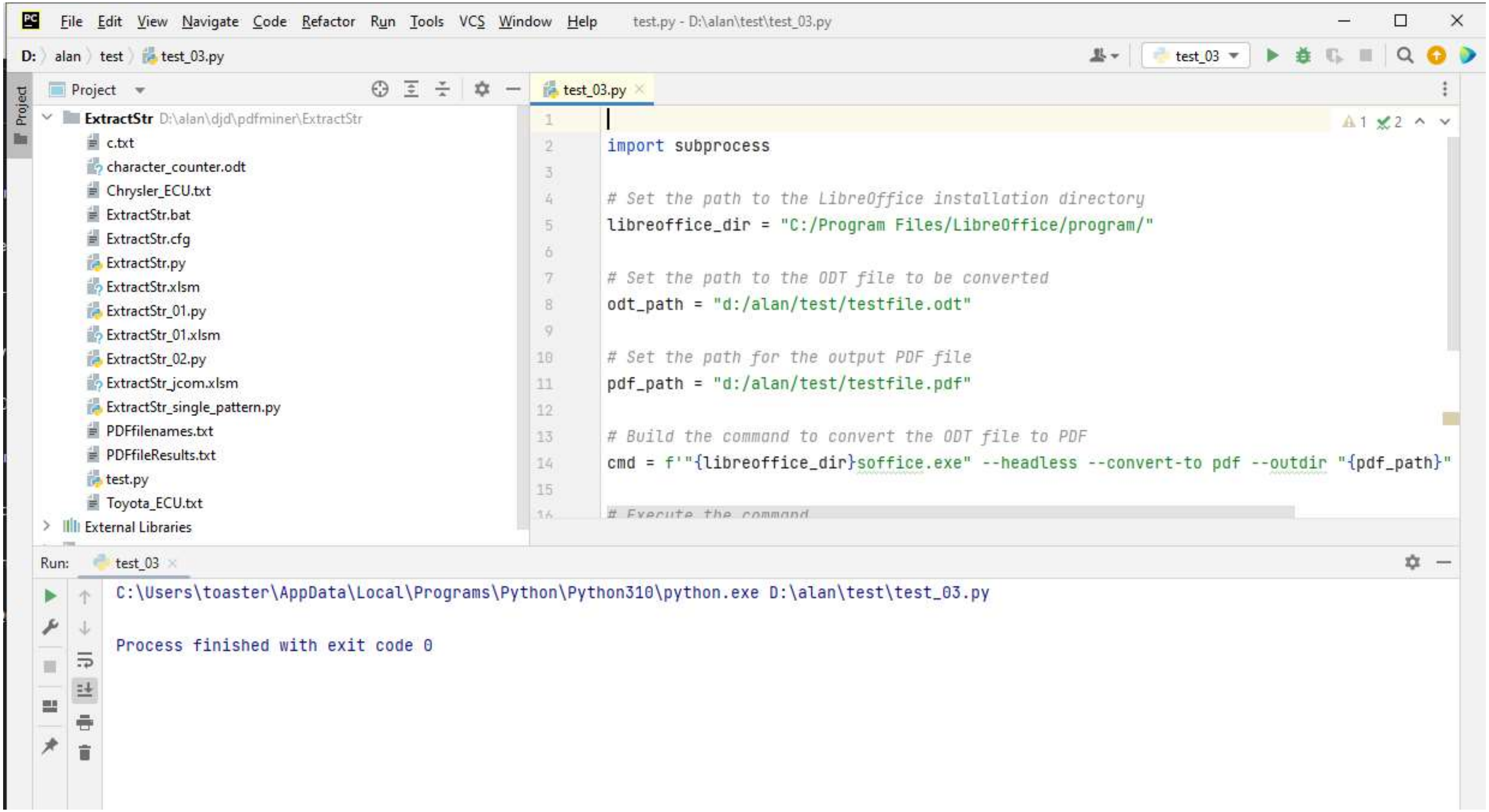
In this updated code, the `libreoffice_dir` variable is enclosed in quotes when building the command, as well as the `pdf_path` and `odt_path` variables. This should ensure that the paths are properly passed to the shell even if they contain spaces.

Original command:

```
cmd = f"{libreoffice_dir}soffice.exe"
--headless --convert-to pdf "{odt_path}"
```

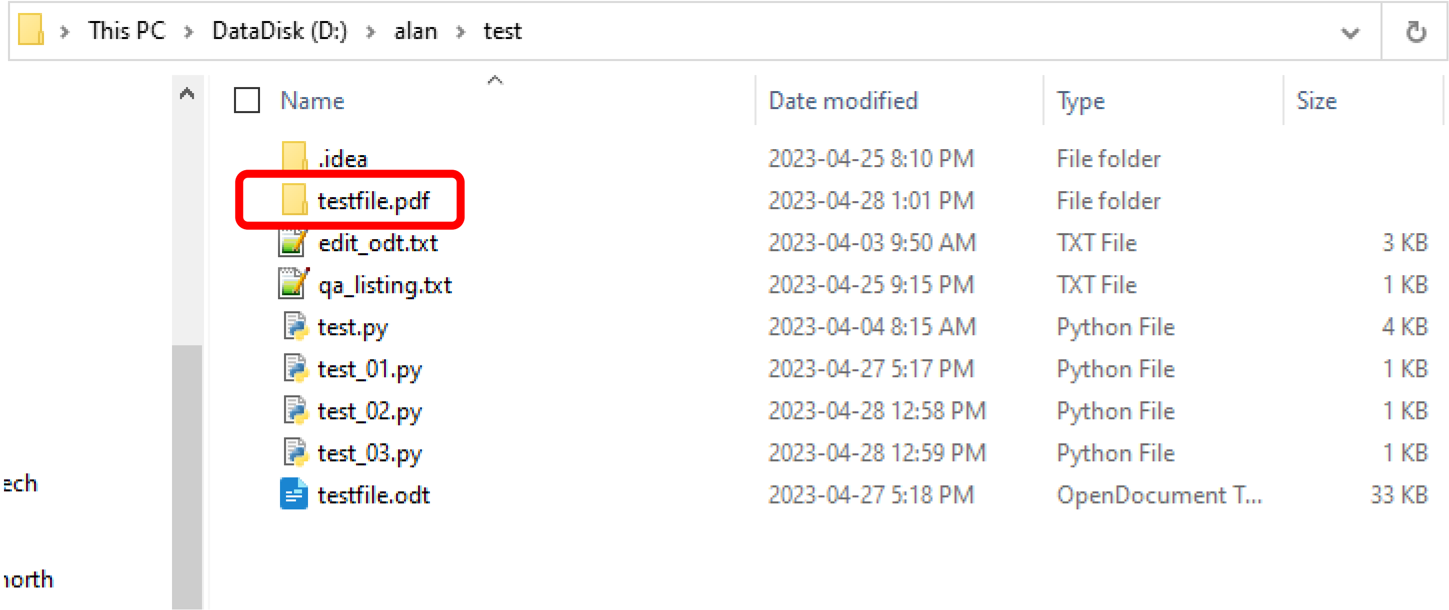
Revised command:

```
cmd = f'"{libreoffice_dir}soffice.exe"
--headless --convert-to pdf "{odt_path}"'
```



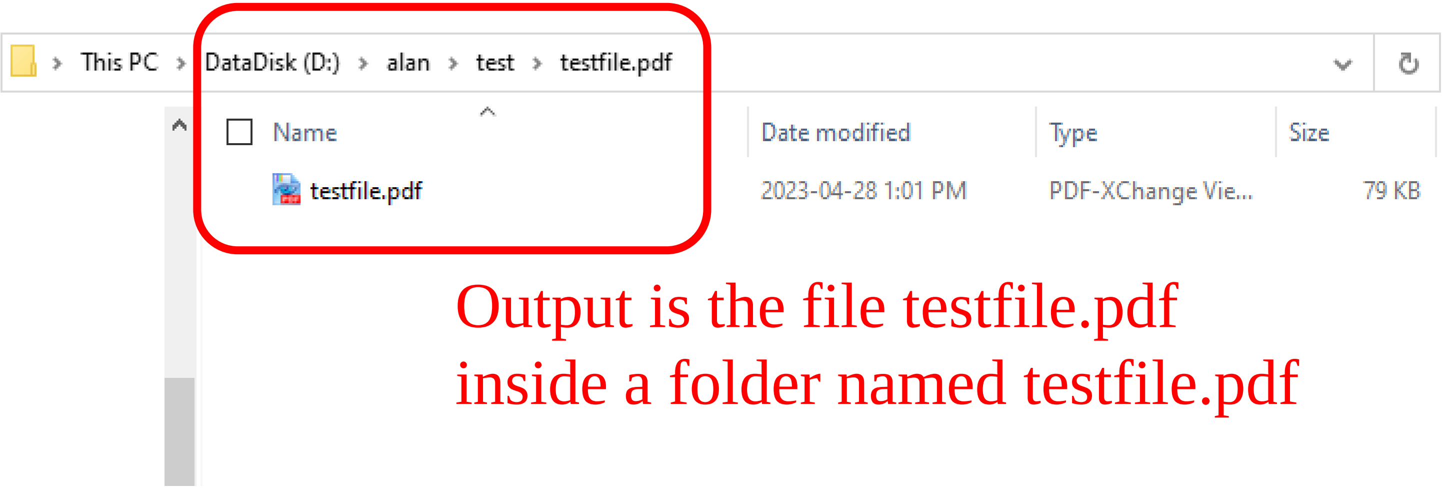
```python

import subprocess

# Set the path to the LibreOffice installation directory
libreoffice_dir = "C:/Program Files/LibreOffice/program/"

# Set the path to the ODT file to be converted
odt_path = "d:/alan/test/testfile.odt"

# Set the path for the output PDF file
pdf_path = "d:/alan/test/testfile.pdf"

# Build the command to convert the ODT file to PDF
cmd = f'"{libreoffice_dir}soffice.exe" --headless --convert-to pdf --outdir "{pdf_path}"

# Execute the command
```

```
C:\Users\toaster\AppData\Local\Programs\Python\Python310\python.exe D:\alan\test\test_03.py

Process finished with exit code 0
```



This PC › DataDisk (D:) › alan › test

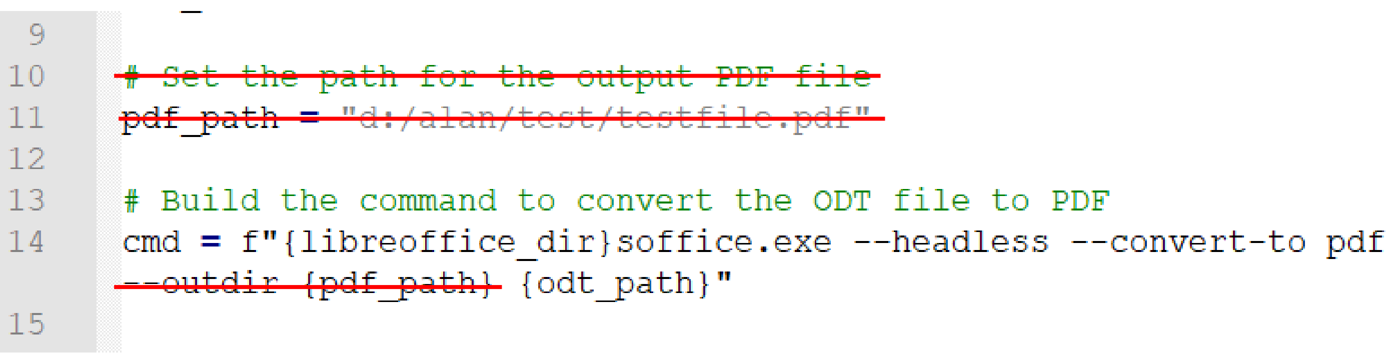
Name	Date modified	Type	Size
.idea	2023-04-25 8:10 PM	File folder	
testfile.pdf	2023-04-28 1:01 PM	File folder	
edit_odt.txt	2023-04-03 9:50 AM	TXT File	3 KB
qa_listing.txt	2023-04-25 9:15 PM	TXT File	1 KB
test.py	2023-04-04 8:15 AM	Python File	4 KB
test_01.py	2023-04-27 5:17 PM	Python File	1 KB
test_02.py	2023-04-28 12:58 PM	Python File	1 KB
test_03.py	2023-04-28 12:59 PM	Python File	1 KB
testfile.odt	2023-04-27 5:18 PM	OpenDocument T...	33 KB



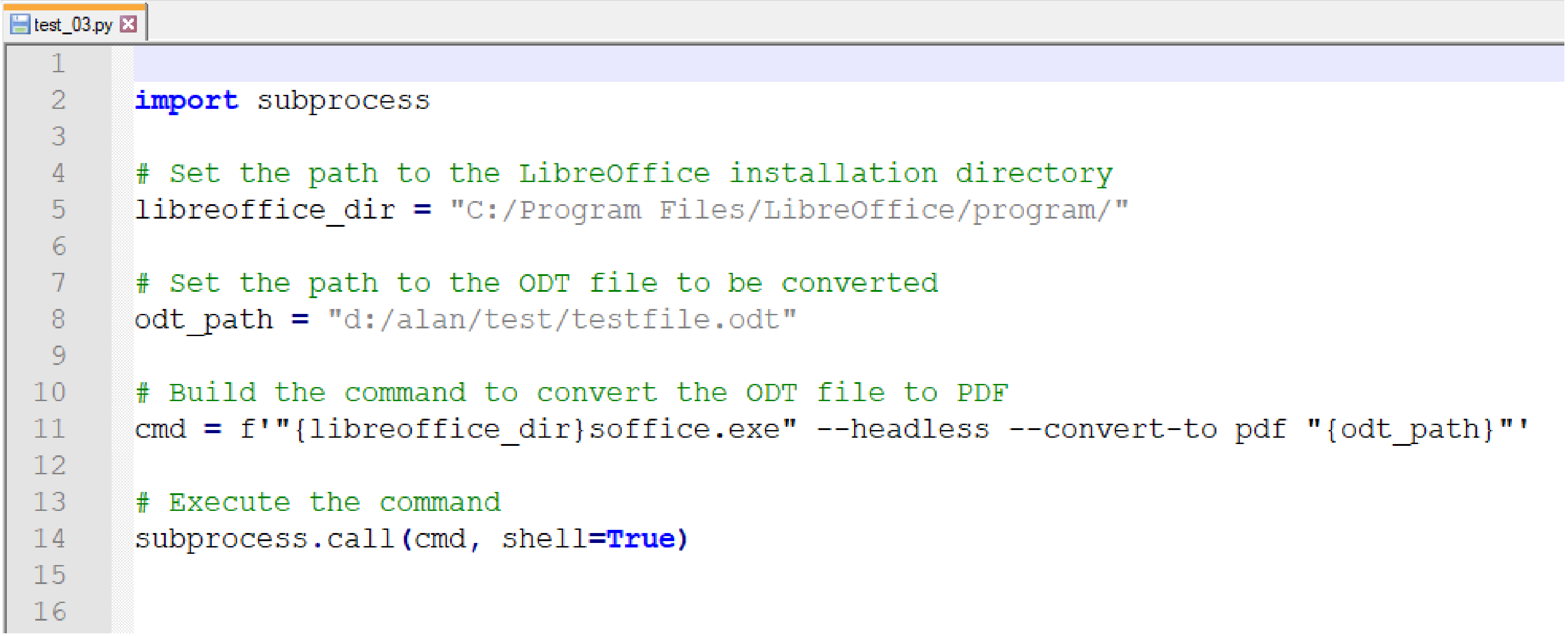
This PC › DataDisk (D:) › alan › test › testfile.pdf

Name	Date modified	Type	Size
testfile.pdf	2023-04-28 1:01 PM	PDF-XChange Vie...	79 KB

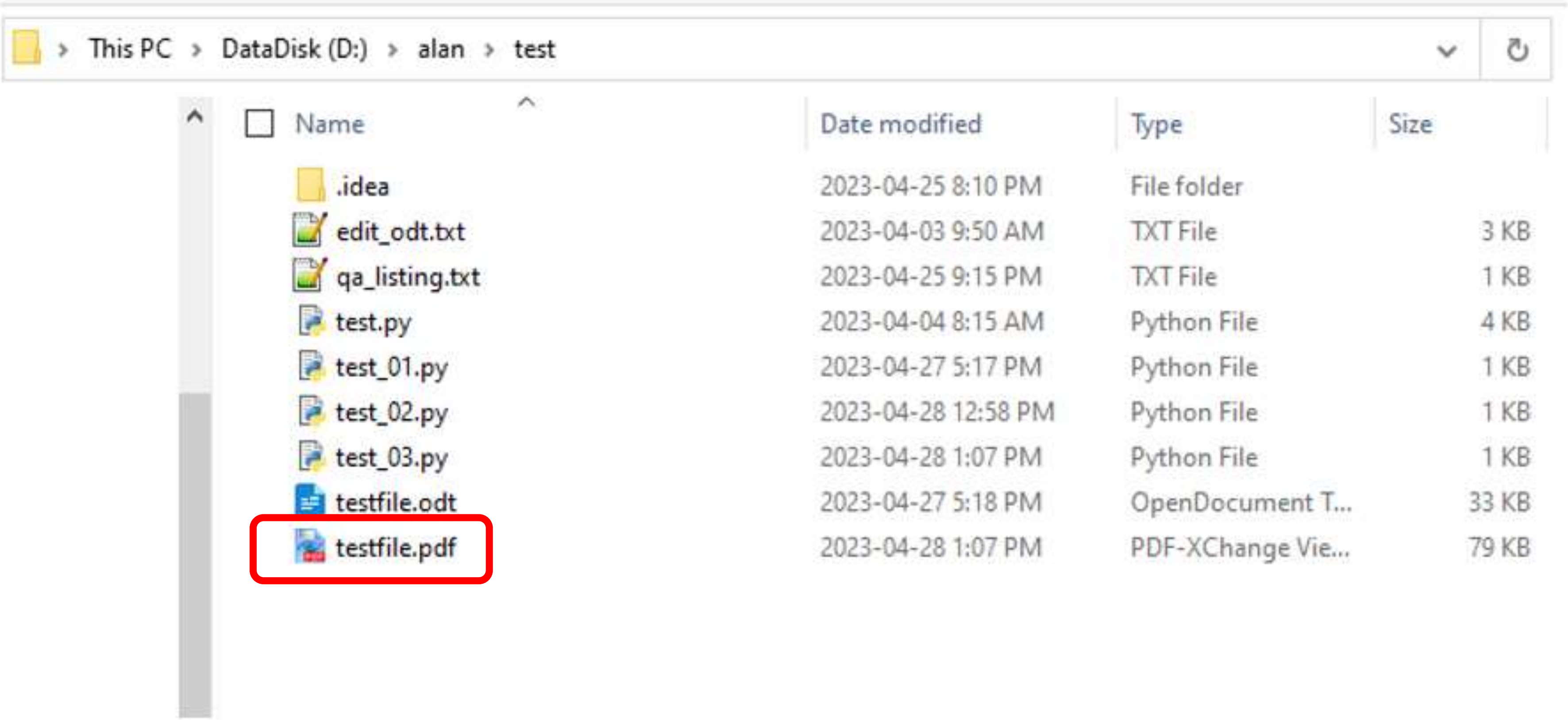
Output is the file testfile.pdf
inside a folder named testfile.pdf



```
~~# Set the path for the output PDF file~~
~~pdf_path = "d:/alan/test/testfile.pdf"~~

# Build the command to convert the ODT file to PDF
cmd = f"{libreoffice_dir}soffice.exe --headless --convert-to pdf
~~--outdir {pdf_path}~~ {odt_path}"

```

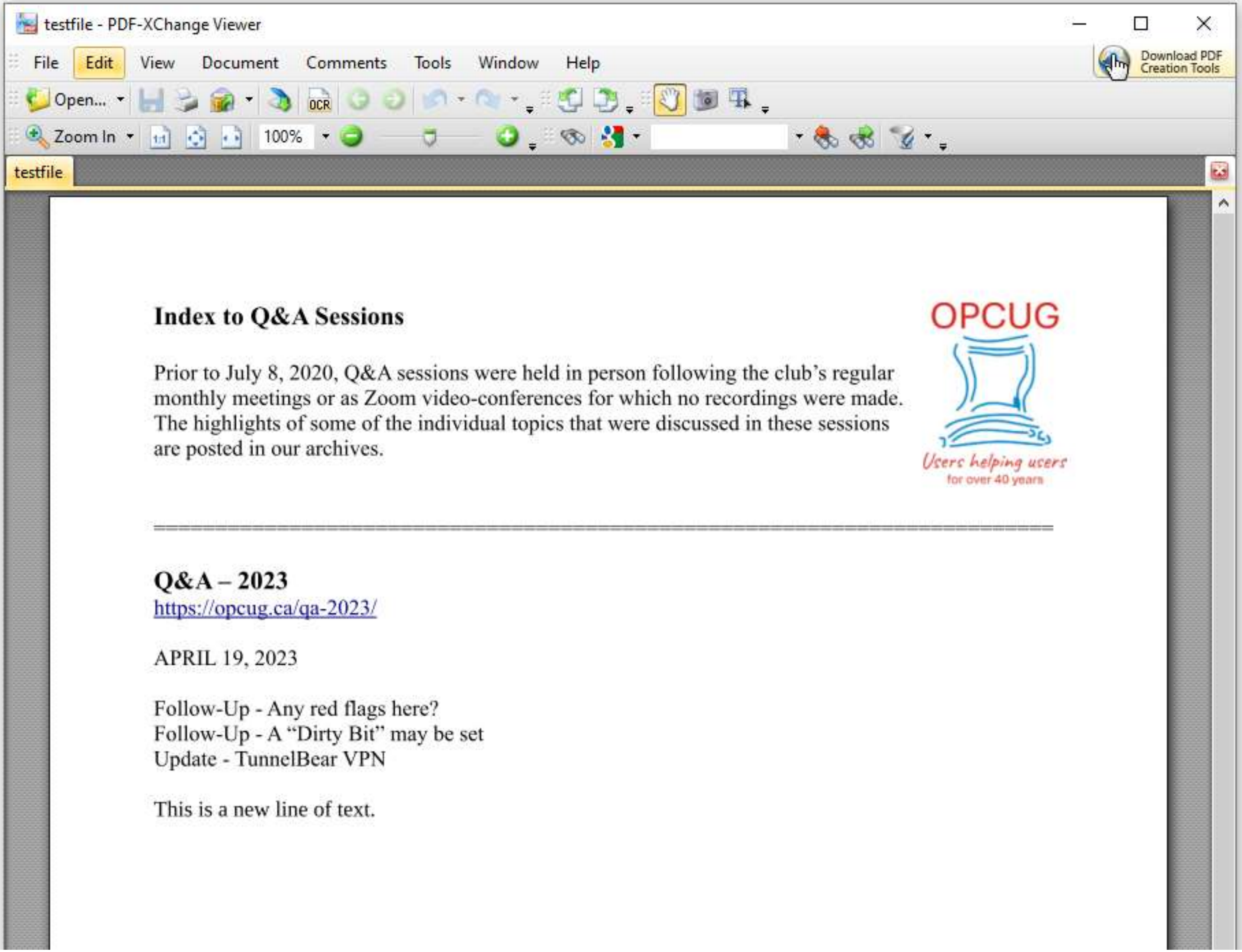


```

import subprocess

# Set the path to the LibreOffice installation directory
libreoffice_dir = "C:/Program Files/LibreOffice/program/"

# Set the path to the ODT file to be converted
odt_path = "d:/alan/test/testfile.odt"

# Build the command to convert the ODT file to PDF
cmd = f'"{libreoffice_dir}soffice.exe" --headless --convert-to pdf "{odt_path}"'

# Execute the command
subprocess.call(cmd, shell=True)


```



This PC › DataDisk (D:) › alan › test

Name	Date modified	Type	Size
.idea	2023-04-25 8:10 PM	File folder	
edit_odt.txt	2023-04-03 9:50 AM	TXT File	3 KB
qa_listing.txt	2023-04-25 9:15 PM	TXT File	1 KB
test.py	2023-04-04 8:15 AM	Python File	4 KB
test_01.py	2023-04-27 5:17 PM	Python File	1 KB
test_02.py	2023-04-28 12:58 PM	Python File	1 KB
test_03.py	2023-04-28 1:07 PM	Python File	1 KB
testfile.odt	2023-04-27 5:18 PM	OpenDocument T...	33 KB
testfile.pdf	2023-04-28 1:07 PM	PDF-XChange Vie...	79 KB



Index to Q&A Sessions

Prior to July 8, 2020, Q&A sessions were held in person following the club’s regular monthly meetings or as Zoom video-conferences for which no recordings were made. The highlights of some of the individual topics that were discussed in these sessions are posted in our archives.

==

Q&A – 2023

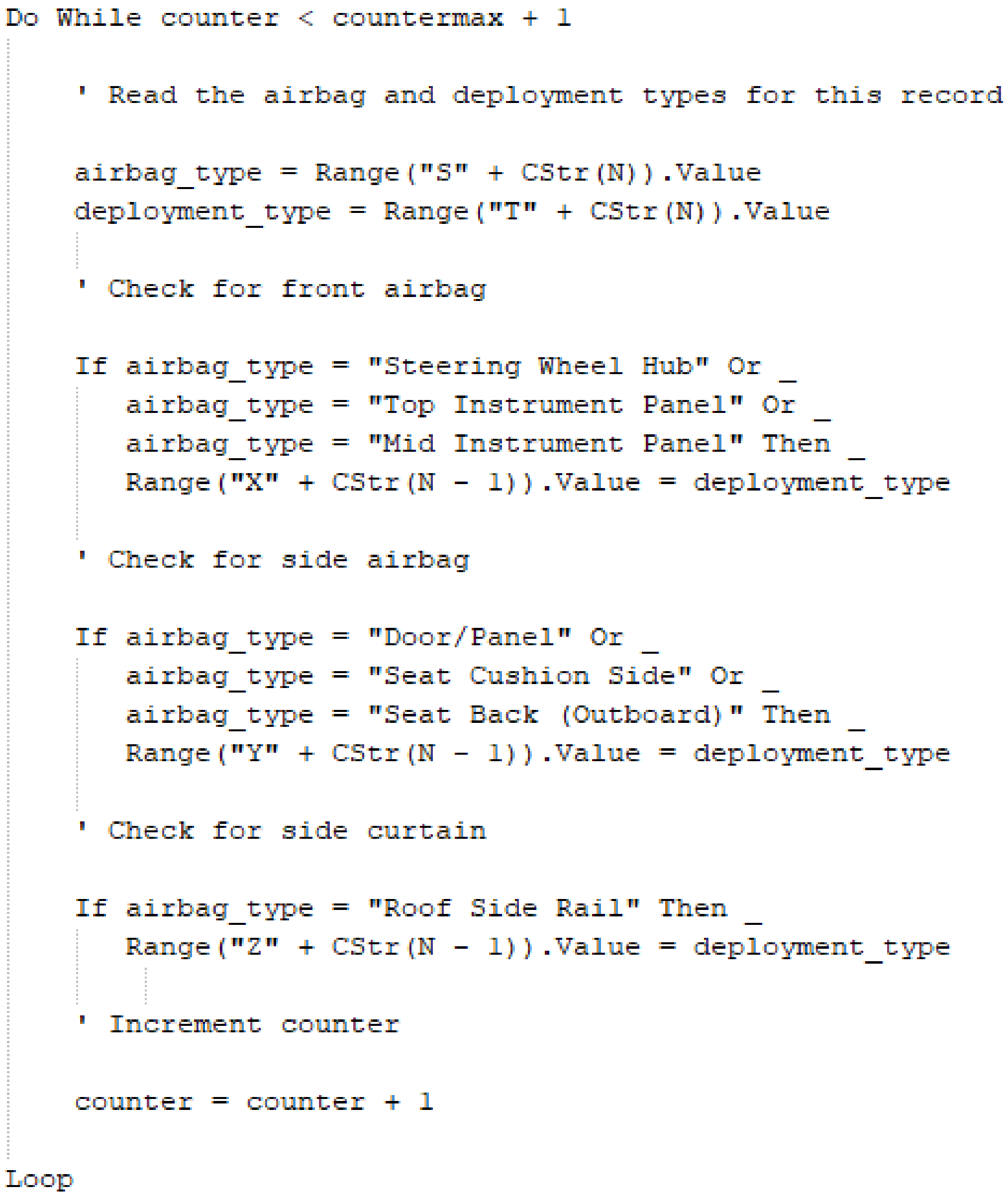
https://opcug.ca/qa-2023/

APRIL 19, 2023

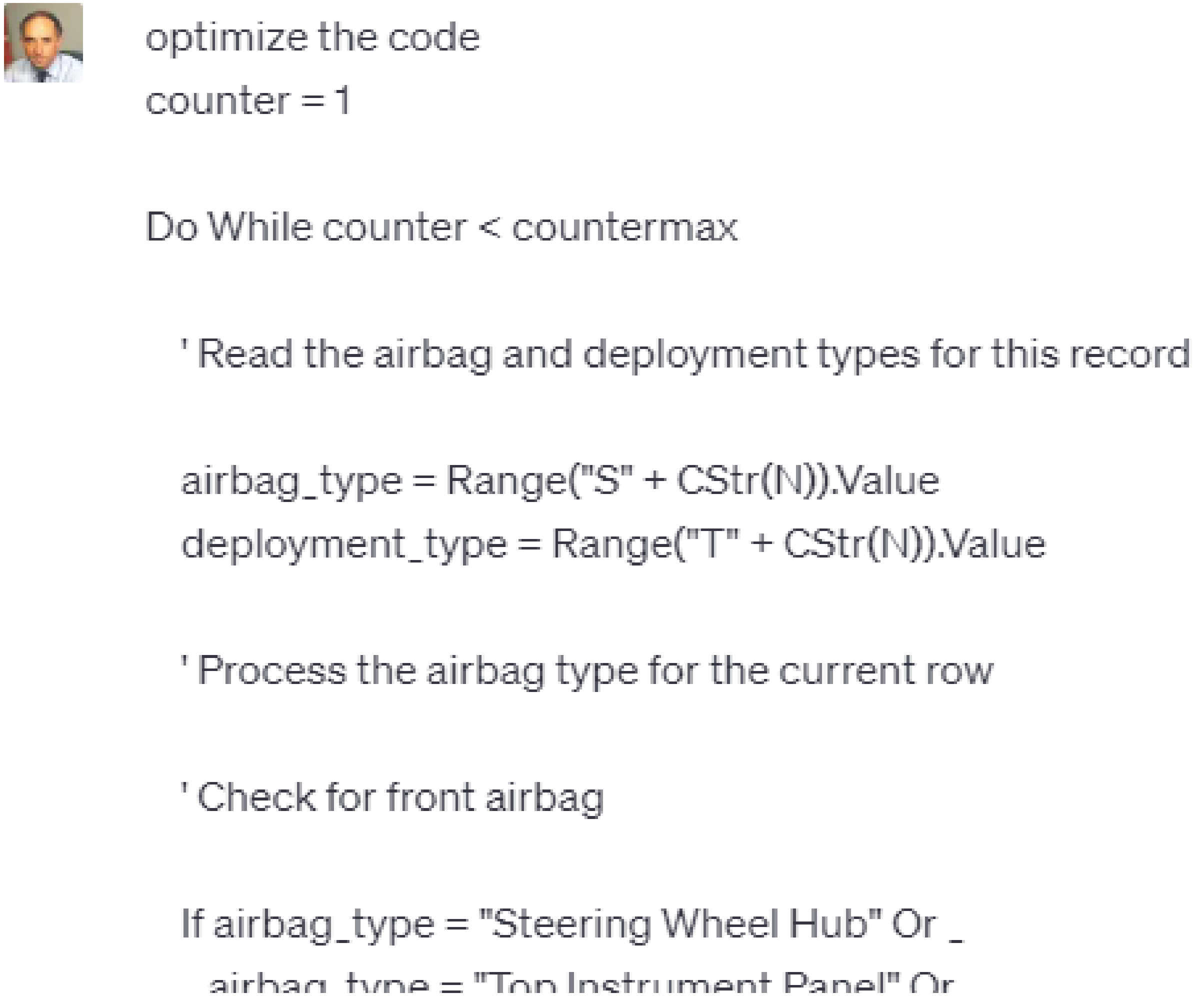
Follow-Up - Any red flags here?
Follow-Up - A “Dirty Bit” may be set
Update - TunnelBear VPN

This is a new line of text.

A further Use for ChatGPT –Optimize

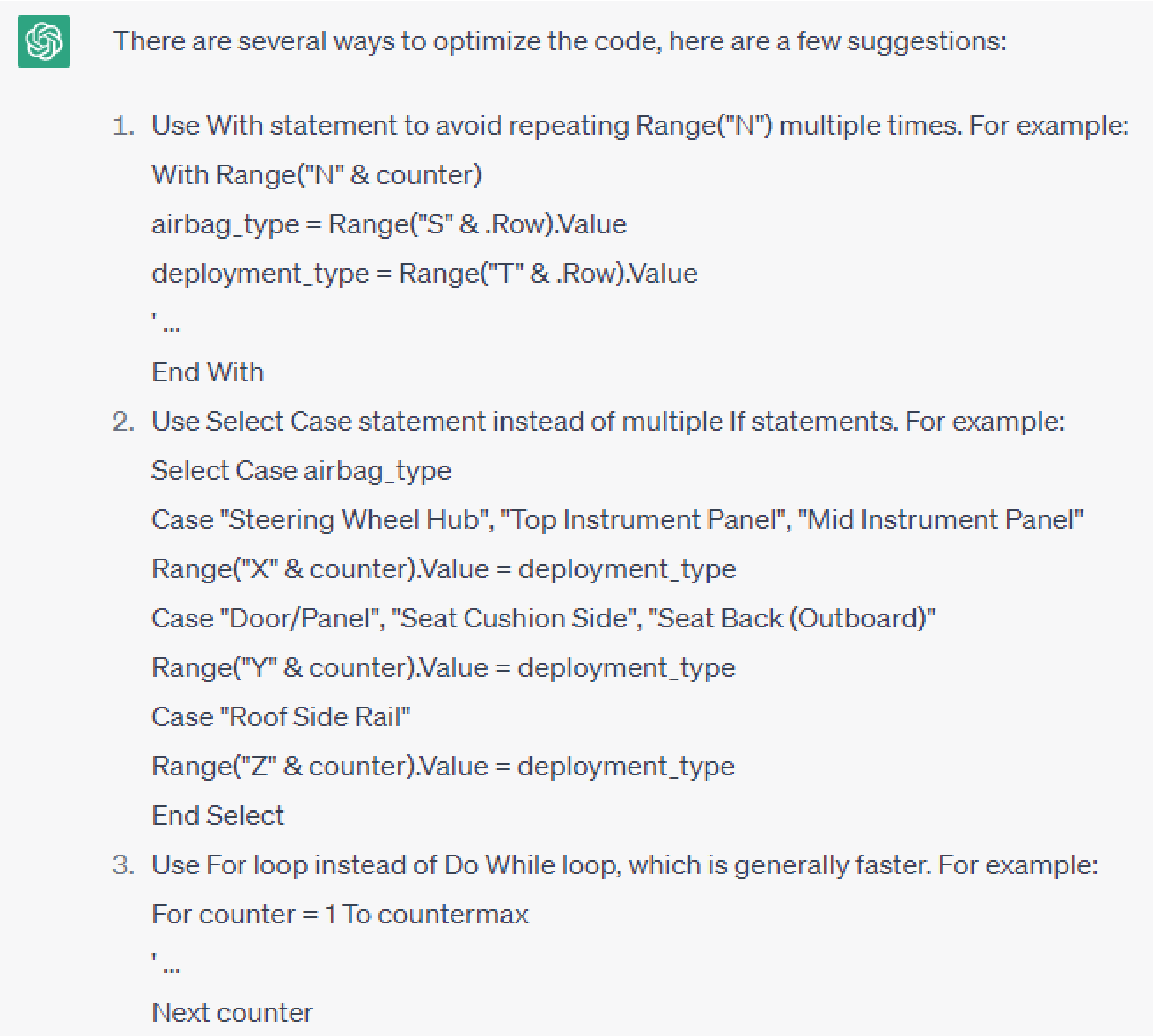


```
Do While counter < countermax + 1

    ' Read the airbag and deployment types for this record

    airbag_type = Range("S" + CStr(N)).Value
    deployment_type = Range("T" + CStr(N)).Value

    ' Check for front airbag

    If airbag_type = "Steering Wheel Hub" Or _
       airbag_type = "Top Instrument Panel" Or _
       airbag_type = "Mid Instrument Panel" Then _
       Range("X" + CStr(N - 1)).Value = deployment_type

    ' Check for side airbag

    If airbag_type = "Door/Panel" Or _
       airbag_type = "Seat Cushion Side" Or _
       airbag_type = "Seat Back (Outboard)" Then _
       Range("Y" + CStr(N - 1)).Value = deployment_type

    ' Check for side curtain

    If airbag_type = "Roof Side Rail" Then _
       Range("Z" + CStr(N - 1)).Value = deployment_type

    ' Increment counter

    counter = counter + 1

Loop
```



optimize the code

```
counter = 1

Do While counter < countermax

   ' Read the airbag and deployment types for this record

   airbag_type = Range("S" + CStr(N)).Value
   deployment_type = Range("T" + CStr(N)).Value

   ' Process the airbag type for the current row

   ' Check for front airbag

   If airbag_type = "Steering Wheel Hub" Or _
     airbag_type = "Top Instrument Panel" Or
```



There are several ways to optimize the code, here are a few suggestions:

1. Use With statement to avoid repeating Range("N") multiple times. For example:

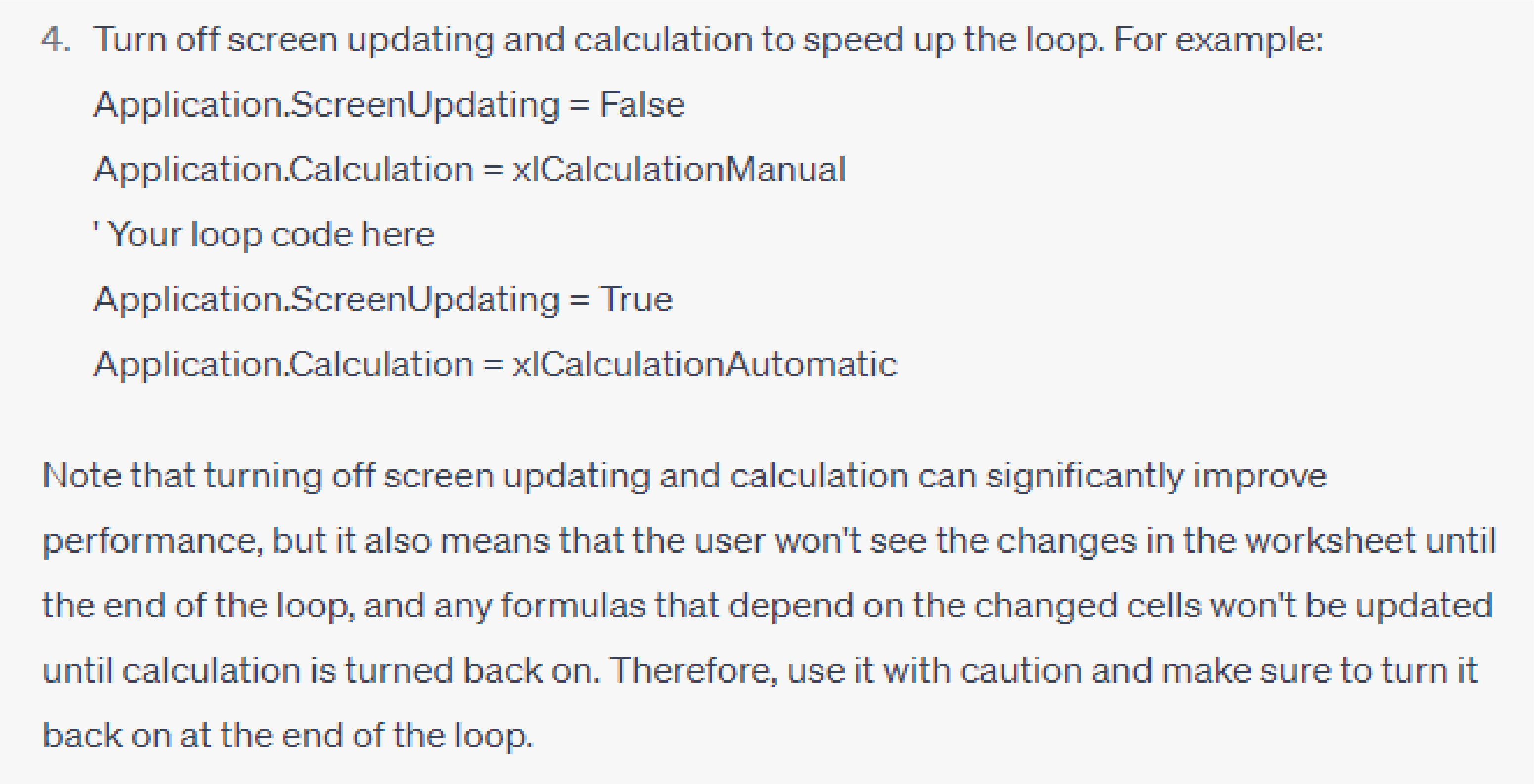
   ```
   With Range("N" & counter)
   airbag_type = Range("S" & .Row).Value
   deployment_type = Range("T" & .Row).Value
   ' ...
   End With
   ```

2. Use Select Case statement instead of multiple If statements. For example:

   ```
   Select Case airbag_type
   Case "Steering Wheel Hub", "Top Instrument Panel", "Mid Instrument Panel"
   Range("X" & counter).Value = deployment_type
   Case "Door/Panel", "Seat Cushion Side", "Seat Back (Outboard)"
   Range("Y" & counter).Value = deployment_type
   Case "Roof Side Rail"
   Range("Z" & counter).Value = deployment_type
   End Select
   ```

3. Use For loop instead of Do While loop, which is generally faster. For example:

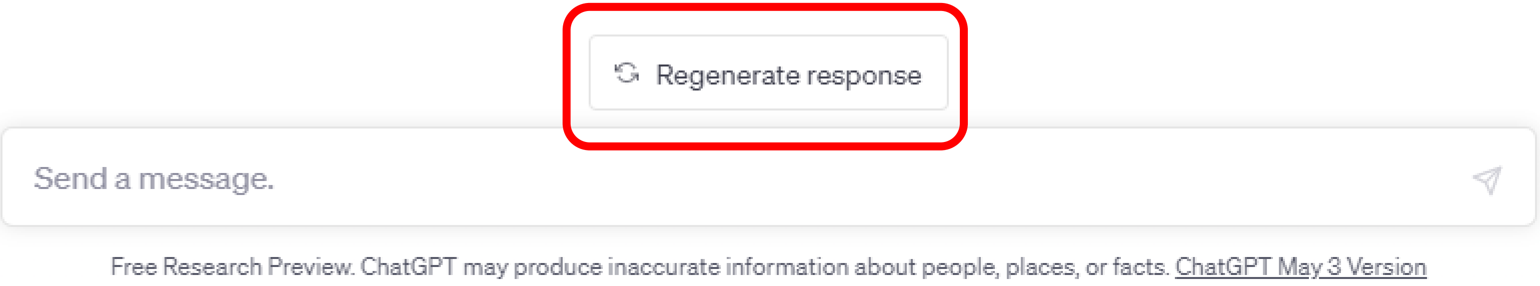
   ```
   For counter = 1 To countermax
   ' ...
   Next counter
   ```



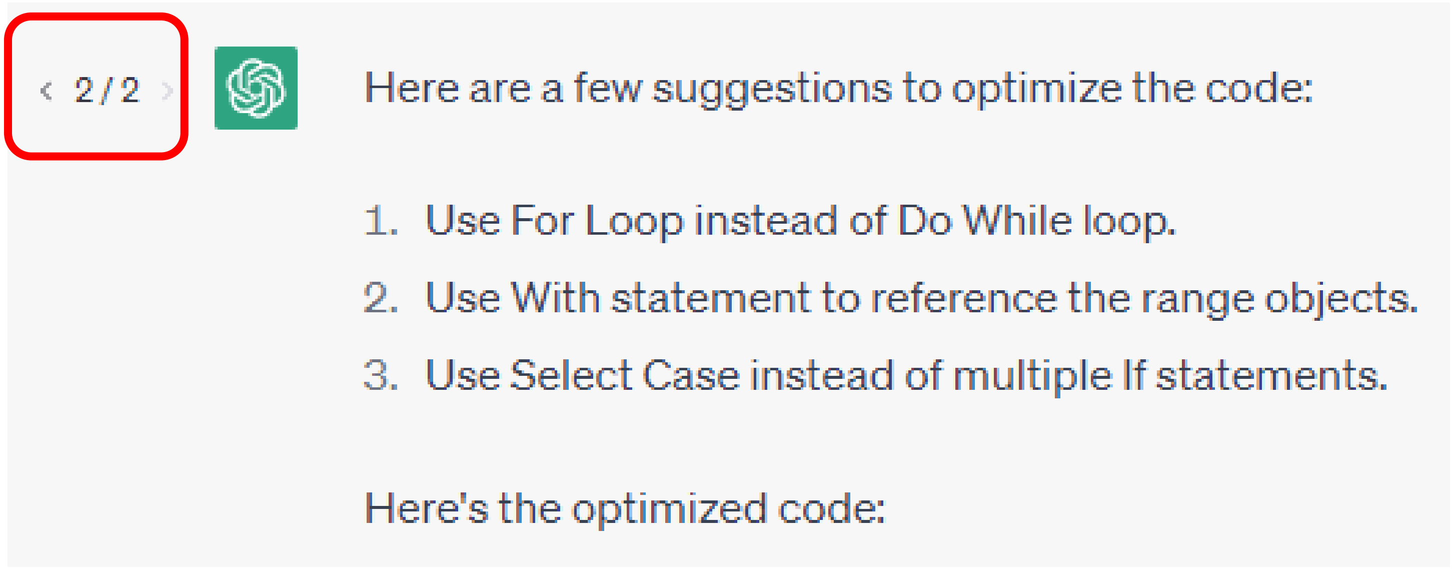
4. Turn off screen updating and calculation to speed up the loop. For example:

   ```
   Application.ScreenUpdating = False
   Application.Calculation = xlCalculationManual
   ' Your loop code here
   Application.ScreenUpdating = True
   Application.Calculation = xlCalculationAutomatic
   ```

Note that turning off screen updating and calculation can significantly improve performance, but it also means that the user won't see the changes in the worksheet until the end of the loop, and any formulas that depend on the changed cells won't be updated until calculation is turned back on. Therefore, use it with caution and make sure to turn it back on at the end of the loop.



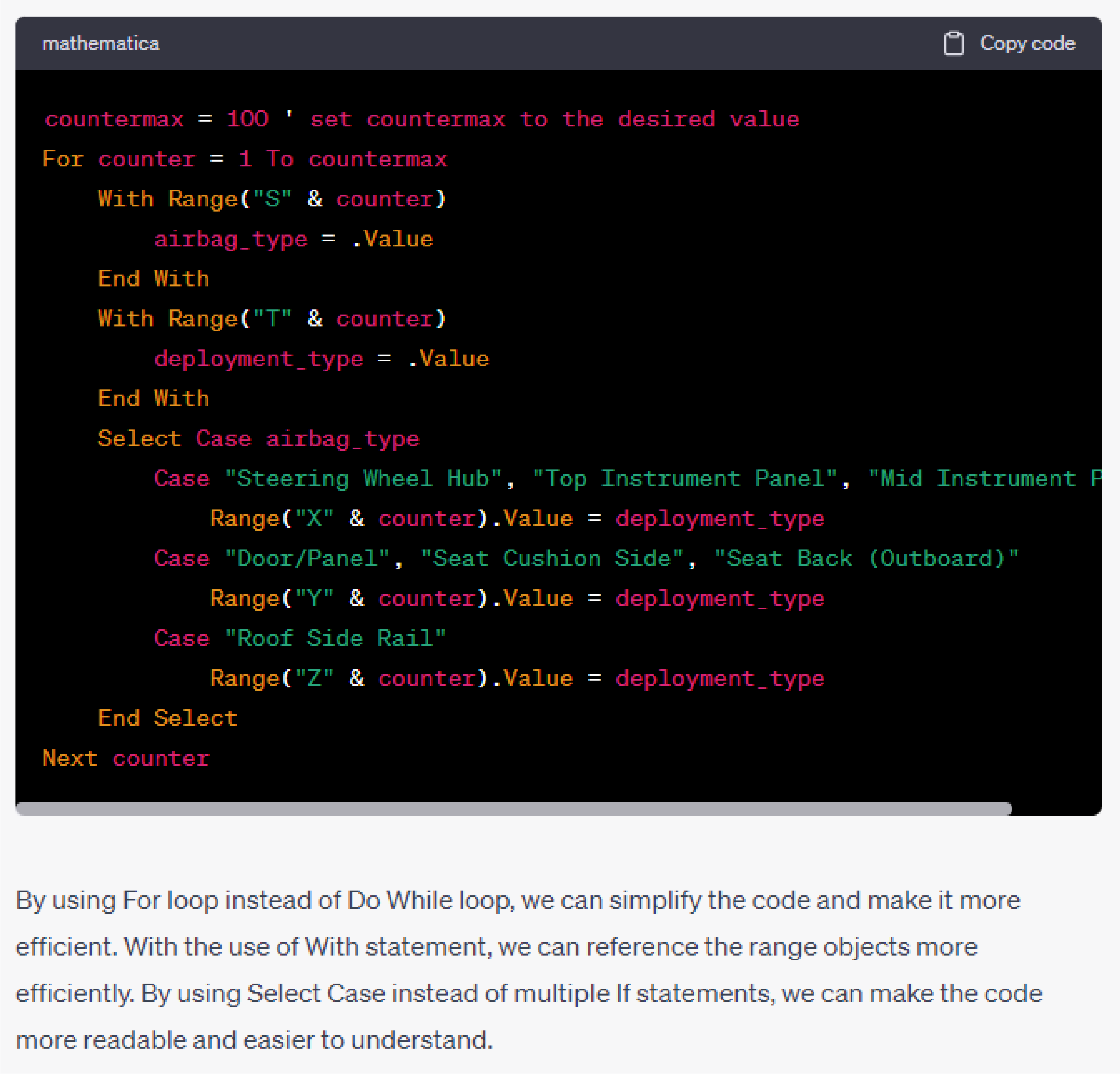
Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT May 3 Version



Here are a few suggestions to optimize the code:

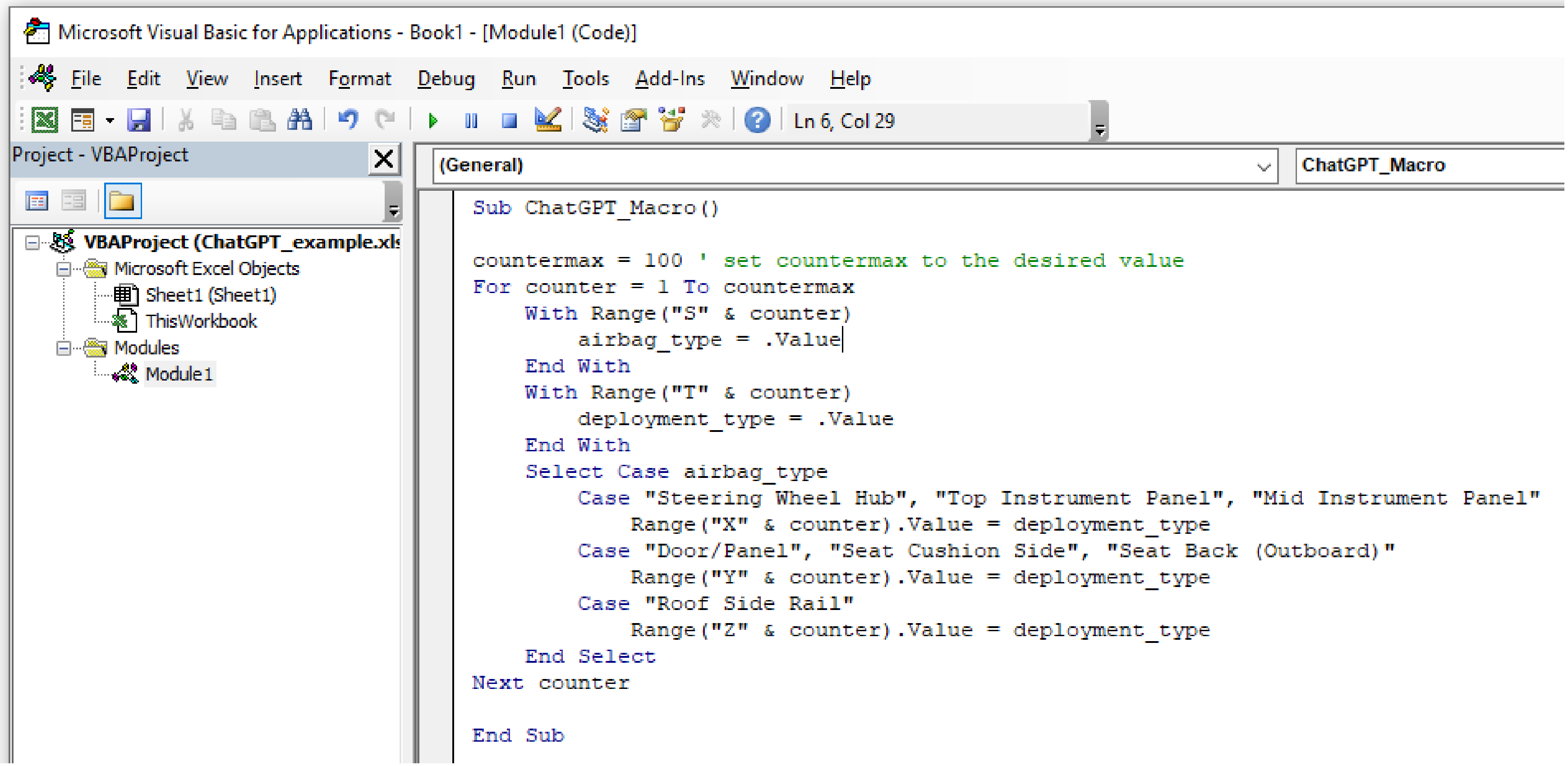
1. Use For Loop instead of Do While loop.
2. Use With statement to reference the range objects.
3. Use Select Case instead of multiple If statements.

Here's the optimized code:

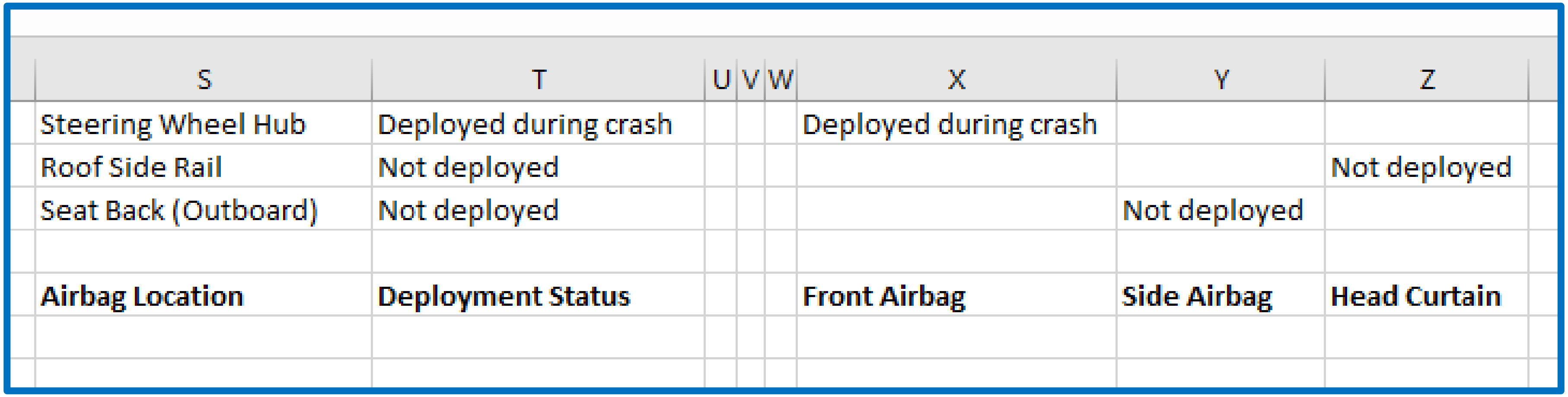


```
countermax = 100 ' set countermax to the desired value
For counter = 1 To countermax
    With Range("S" & counter)
        airbag_type = .Value
    End With
    With Range("T" & counter)
        deployment_type = .Value
    End With
    Select Case airbag_type
        Case "Steering Wheel Hub", "Top Instrument Panel", "Mid Instrument P
            Range("X" & counter).Value = deployment_type
        Case "Door/Panel", "Seat Cushion Side", "Seat Back (Outboard)"
            Range("Y" & counter).Value = deployment_type
        Case "Roof Side Rail"
            Range("Z" & counter).Value = deployment_type
    End Select
Next counter
```

By using For loop instead of Do While loop, we can simplify the code and make it more efficient. With the use of With statement, we can reference the range objects more efficiently. By using Select Case instead of multiple If statements, we can make the code more readable and easier to understand.



```
Sub ChatGPT_Macro()

countermax = 100 ' set countermax to the desired value
For counter = 1 To countermax
    With Range("S" & counter)
        airbag_type = .Value
    End With
    With Range("T" & counter)
        deployment_type = .Value
    End With
    Select Case airbag_type
        Case "Steering Wheel Hub", "Top Instrument Panel", "Mid Instrument Panel"
            Range("X" & counter).Value = deployment_type
        Case "Door/Panel", "Seat Cushion Side", "Seat Back (Outboard)"
            Range("Y" & counter).Value = deployment_type
        Case "Roof Side Rail"
            Range("Z" & counter).Value = deployment_type
    End Select
Next counter

End Sub
```



S	T	U	V	W	X	Y	Z
Steering Wheel Hub	Deployed during crash				Deployed during crash		
Roof Side Rail	Not deployed						Not deployed
Seat Back (Outboard)	Not deployed					Not deployed	
Airbag Location	**Deployment Status**				**Front Airbag**	**Side Airbag**	**Head Curtain**

Now I know how to write lines of text to an ODT file, save the ODT file as PDF, and how to debug and optimize my code in Python!