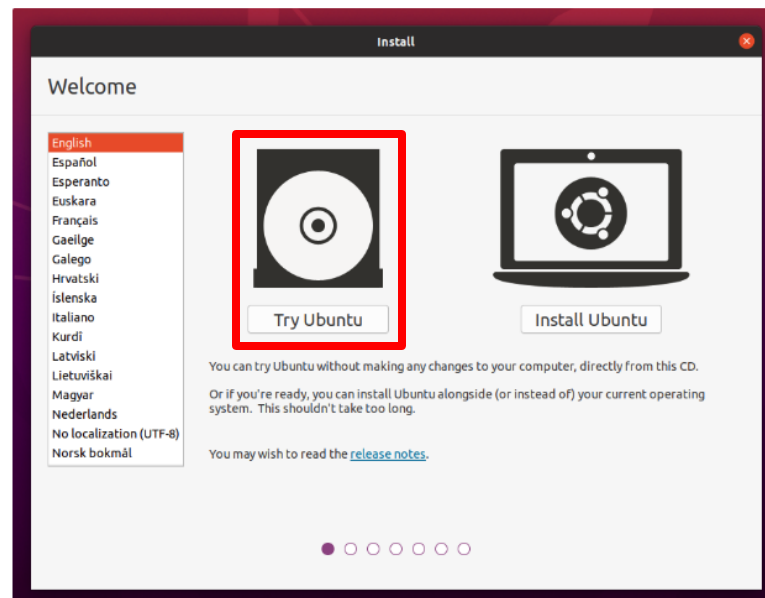


Laboratorio di Elettronica e Tecniche di Acquisizione Dati 2025-2026

Esercitazione 0 "Vocale WhatsApp"

Preliminare

- scaricare una Live di Ubuntu:
<https://ubuntu.com/tutorials/install-ubuntu-desktop#2-download-an-ubuntu-image>
- creare una penna USB "bootabile":
<https://ubuntu.com/tutorials/install-ubuntu-desktop#3-create-a-bootable-usb-stick>
- "bootare" il PC dalla penna USB:
<https://support.lenovo.com/us/en/solutions/ht500207-how-to-boot-from-usb-disk-in-the-bios-boot-menu-windows-8-windows-10-ideapadlenovo-laptops>
(caso 2)
- avviare Ubuntu Live:
<https://ubuntu.com/tutorials/install-ubuntu-desktop#4-boot-from-usb-flash-drive>
("Try Ubuntu")



Preliminare (alternativa)

- se avete già un PC con Linux basta aprire un terminale!
- se avete un Mac basta aprire un terminale!
- potete usare la Power Shell di Windows e installare python direttamente in Windows
- potete usare WLS (Windows for Linux System) e installare python lì

Preliminare

- scaricare una Live di Ubuntu:
<https://ubuntu.com/tutorials/install-ubuntu-desktop#2-download-an-ubuntu-image>
- creare una penna USB "bootabile":
<https://ubuntu.com/tutorials/install-ubuntu-desktop#3-create-a-bootable-usb-stick>
- "bootare" il PC dalla penna USB:
<https://support.lenovo.com/us/en/solutions/ht500207-how-to-boot-from-usb-disk-in-the-bios-boot-menu-windows-8-windows-10-ideapadlenovo-laptops>
(caso 2)
- avviare Ubuntu Live:
<https://ubuntu.com/tutorials/install-ubuntu-desktop#4-boot-from-usb-flash-drive>
("Try Ubuntu")
- installare pip:
<https://linuxhint.com/install-python-pip-ubuntu-22-04/>

```
$> sudo add-apt-repository universe
```
- (se necessario) installare moduli python richiesti:

```
$> pip install soundcard
```

Esercitazione

- realizzare un piccolo programma python per registrare e ri-ascoltare un segnale audio dalla scheda audio del PC

Acquisizione Audio

Fonte: <https://pypi.org/project/SoundCard/>
<https://github.com/bastibe/SoundCard>
<https://soundcard.readthedocs.io/en/latest/>

```
import soundcard as sc
import numpy as np

speakers = sc.all_speakers()
print("Speaker disponibili: ", speakers, "\n")
default_speaker = sc.default_speaker()
print("Speaker selezionato: ", default_speaker, "\n")

mics = sc.all_microphones()
print("Microfoni disponibili: ", mics, "\n")
default_mic = sc.default_microphone()
print("Microfono selezionato: ", default_mic, "\n")

data = default_mic.record(samplerate=48000, numframes=48000)
print("Array acquisito: ")
print(data)
print("\n")

default_speaker.play(data/np.max(data), samplerate=48000)

print("L'array dei dati registrati è lungo ", len(data))
```

Acquisizione Audio

```
import soundcard as sc
import numpy as np

speakers = sc.all_speakers()
print("Speaker disponibili: ", speakers, "\n")
default_speaker = sc.default_speaker()
print("Speaker selezionato: ", default_speaker, "\n")

...
```

Come eseguirlo?

- salvare tutto in un file, ad esempio `registra_audio.py`, e poi:

```
$> python registra_audio.py
```

```
Speaker disponibili: [<Speaker PHL 243S7 (2 channels)>, <Speaker USB Audio Device (2 channels)>, <Speaker Background Music (2 channels)>, <Speaker Background Music (UI Sounds) (2 channels)>, <Speaker Altoparlanti MacBook Pro (2 channels)>, <Speaker Microsoft Teams Audio (2 channels)>, <Speaker ZoomAudioDevice (2 channels)>]
```

```
Speaker selezionato: <Speaker USB Audio Device (2 channels)>
```

```
Microfoni disponibili: [<Microphone GENERAL WEBCAM (1 channels)>, <Microphone USB Audio Device (1 channels)>, <Microphone Background Music (2 channels)>, <Microphone Background Music (UI Sounds) (2 channels)>, <Microphone Microfono MacBook Pro (1 channels)>, <Microphone Microsoft Teams Audio (2 channels)>, <Microphone ZoomAudioDevice (2 channels)>, <Microphone VOX+MicInterno (1 channels)>]
```

```
Microfono selezionato: <Microphone Microfono MacBook Pro (1 channels)>
```

Acquisizione Audio

```
import soundcard as sc
import numpy as np

speakers = sc.all_speakers()
print("Speaker disponibili: ", speakers, "\n")
default_speaker = sc.default_speaker()
print("Speaker selezionato: ", default_speaker, "\n")

...
```

Come eseguirlo?

- salvare tutto in un file, ad esempio `registra_audio.py`, e poi:
- aprire un interprete python e inserire i comandi uno per uno:

```
$> python3
```

```
Python 3.10.7 (main, Sep 14 2022, 22:38:23) [Clang 14.0.0 (clang-1400.0.29.102)] on
```

```
darwin
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import soundcard as sc
```

```
>>> import numpy as np
```

```
>>> speakers = sc.all_speakers()
```

```
>>> print(speakers)
```

```
[<Speaker PHL 243S7 (2 channels)>, <Speaker USB Audio Device (2 channels)>, <Speaker Background Music (2 channels)>,
<Speaker Background Music (UI Sounds) (2 channels)>, <Speaker Altoparlanti MacBook Pro (2 channels)>, <Speaker Microsoft Teams Audio (2
channels)>, <Speaker ZoomAudioDevice (2 channels)>]
```


Acquisizione Audio

```
import soundcard as sc
import numpy as np

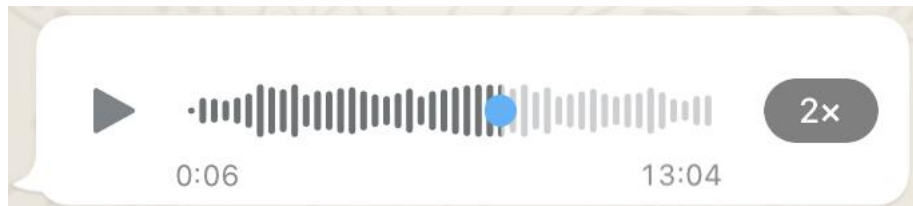
speakers = sc.all_speakers()
print("Speaker disponibili: ", speakers, "\n")
default_speaker = sc.default_speaker()
print("Speaker selezionato: ", default_speaker, "\n")

mics = sc.all_microphones()
print("Microfoni disponibili: ", mics, "\n")
default_mic = sc.default_microphone()
print("Microfono selezionato: ", default_mic, "\n")

data = default_mic.record(samplerate=48000, numframes=48000)

...
```

- quanti secondi stiamo acquisendo?
- provare a registrare 10 secondi
- provare a ri-ascoltare la registrazione al doppio e alla metà della velocità



Esercitazione

- realizzare un piccolo programma python per registrare e ri-ascoltare un segnale audio dalla scheda audio del PC
- realizzare un piccolo programma python che plotti un semplice grafico

Plot grafico / array

Fonte: <https://www.tutorialspoint.com/how-to-plot-an-array-in-python-using-matplotlib>

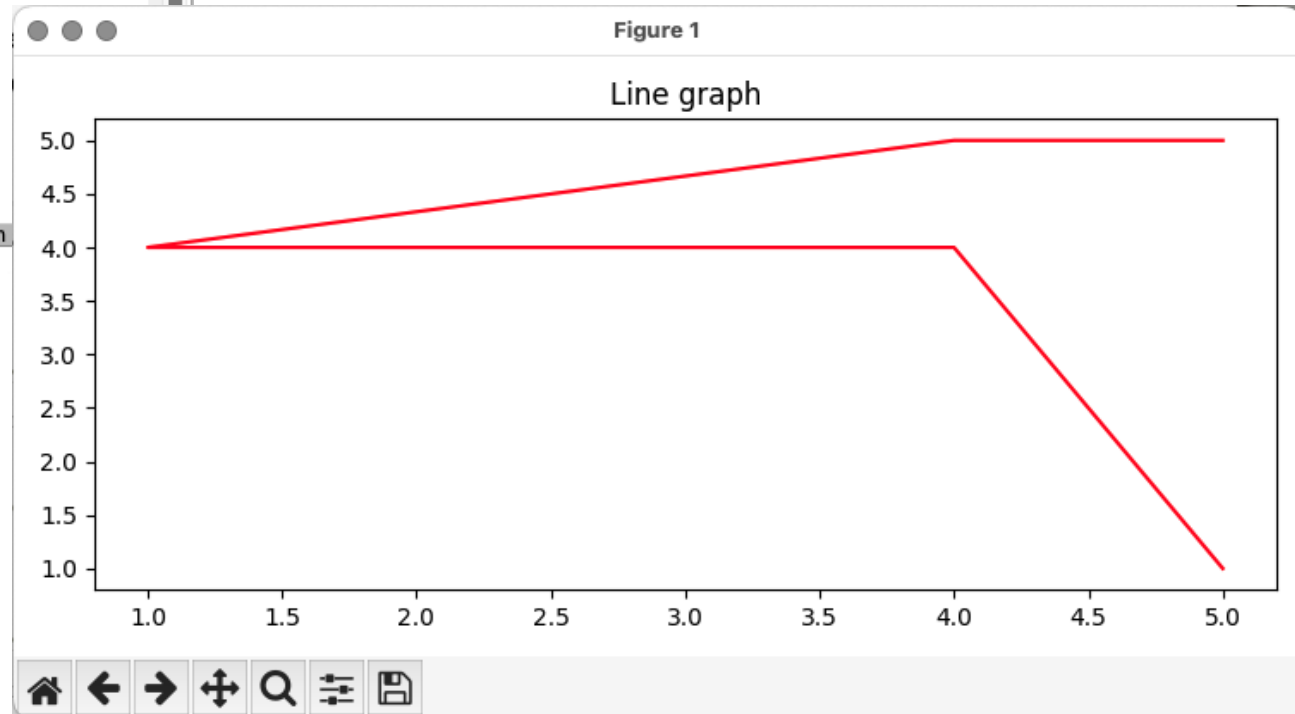
```
plotta_grafico.py
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

x = np.array([5, 4, 1, 4, 5])
y = np.sort(x)

plt.title("Line graph")
plt.plot(x, y, color="red")

plt.show()
```



Plot grafico / array

Fonte: <https://www.tutorialspoint.com/how-to-plot-an-array-in-python-using-matplotlib>
<https://www.delftstack.com/it/tutorial/matplotlib/axis-label/>

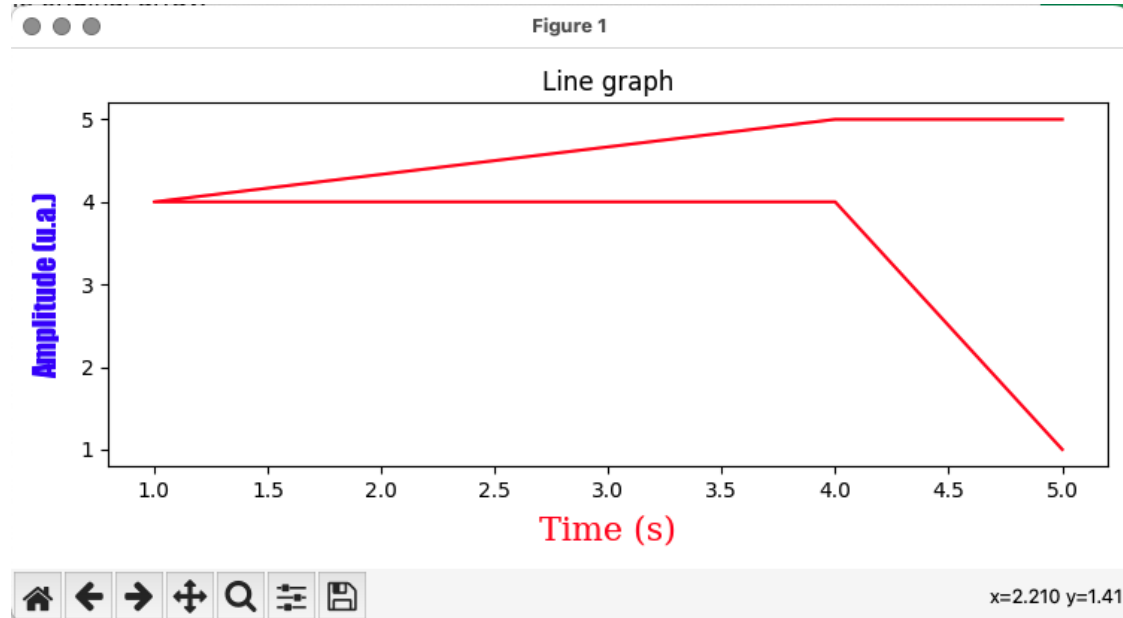
```
plotta_grafico.py
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

x = np.array([5, 4, 1, 4, 5])
y = np.sort(x)

plt.title("Line graph")
plt.plot(x, y, color="red")

plt.show()
```



- provare a cambiare la dimensione della finestra
- aumentare il numero di punti
- cambiare il colore della linea
- cambiare il titolo del grafico
- aggiungere le label sugli assi

Esercitazione

- realizzare un piccolo programma python per registrare e ri-ascoltare un segnale audio dalla scheda audio del PC
- realizzare un piccolo programma python che plotti un semplice grafico
- realizzare un programma che registri un segnale audio, lo faccia ri-ascoltare e ne plotti la "waveform"

Acquisizione Audio + Plot

```
$> python3 registra_e_plotta_audio.py
```

```
[<Speaker PHL 243S7 (2 channels)>, <Speaker USB Audio Device (2 channels)>, <Speaker Background Music (2 channels)>, <Speaker Background Music (UI Sounds) (2 channels)>, <Speaker Altoparlanti MacBook Pro (2 channels)>, <Speaker Microsoft Teams Audio (2 channels)>, <Speaker ZoomAudioDevice (2 channels)>]
```

```
<Speaker USB Audio Device (2 channels)>
```

```
[<Microphone GENERAL WEBCAM (1 channels)>, <Microphone USB Audio Device (1 channels)>, <Microphone Background Music (2 channels)>, <Microphone Background Music (UI Sounds) (2 channels)>, <Microphone Microfono MacBook Pro (1 channels)>, <Microphone Microsoft Teams Audio (2 channels)>, <Microphone ZoomAudioDevice (2 channels)>, <Microphone VOX+MicInterno (1 channels)>]
```

```
<Microphone Microfono MacBook Pro (1 channels)>
```

```
[[-8.24038580e-08]
```

```
[-9.52531991e-08]
```

```
[ 1.71060674e-06]
```

```
...
```

```
[ 5.29479177e-04]
```

```
[ 4.57531773e-04]
```

```
[ 7.15787406e-04]]
```

```
480000
```

```
[[0.00000000e+00]
```

```
[2.08333333e-05]
```

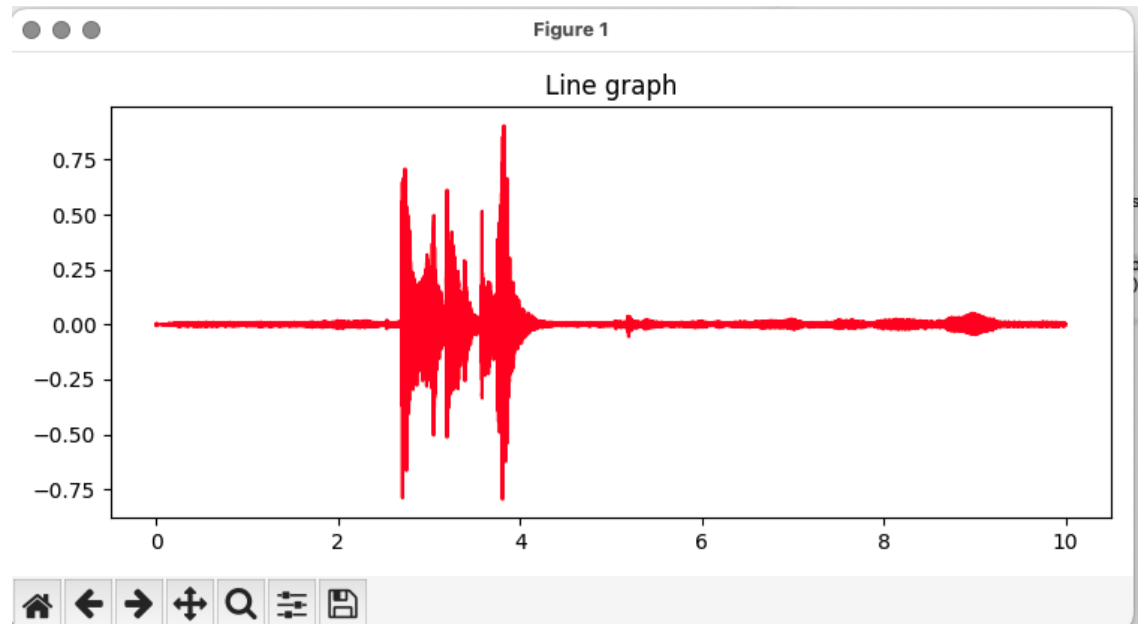
```
[4.16666667e-05]
```

```
...
```

```
[9.99993750e+00]
```

```
[9.99995833e+00]
```

```
[9.99997917e+00]]
```



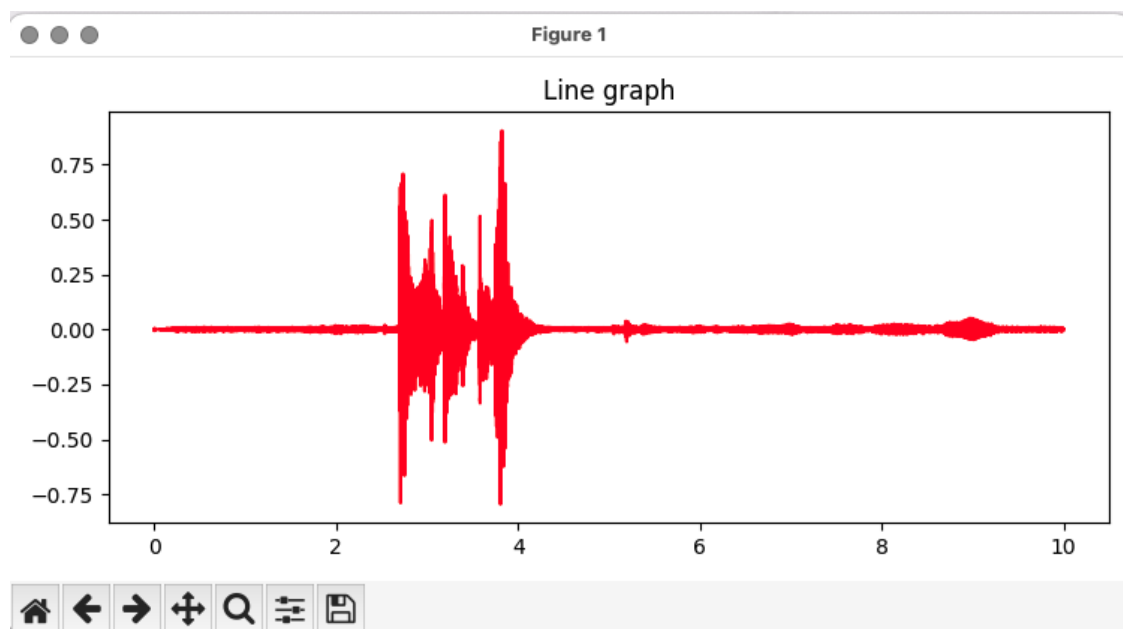
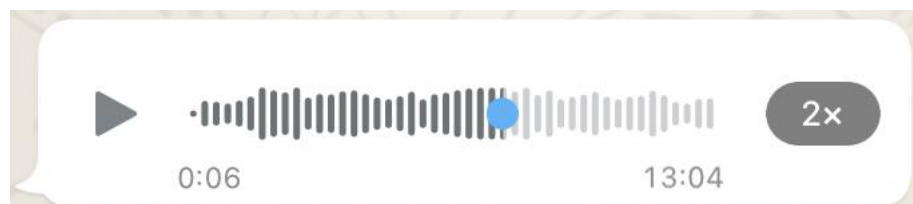
Acquisizione Audio + Plot

- cosa mettete sulle x?

- Fonti utili:

<https://appdividend.com/2022/10/02/how-to-copy-an-array-in-python/>

<https://www.askpython.com/python/list/iterate-through-list-in-python>



Esercitazione

- realizzare un piccolo programma python per registrare e ri-ascoltare un segnale audio dalla scheda audio del PC
- realizzare un piccolo programma python che plotti un semplice grafico
- realizzare un programma python che registri un segnale audio, lo faccia ri-ascoltare e ne plotti la *waveform*
- realizzare un programma python per trasferire stringhe di testo via *socket*
- realizzare un programma python per trasferire un array via *socket*
- realizzare un programma python che registri un segnale audio (definire un *protocollo!*), lo trasferisca via *socket* e il ricevente lo faccia ri-ascoltare e ne plotti la *waveform*

Socket TCP

- Fonti utili:

<https://docs.python.org/3/library/socket.html>

<https://stackoverflow.com/questions/34653875/python-how-to-send-data-over-tcp>

<https://python-reference.readthedocs.io/en/latest/docs/str/encode.html>

<https://python-reference.readthedocs.io/en/latest/docs/str/decode.html>

[https://stackoverflow.com/questions/24423162/how-to-send-an-array-over-a-socket-in-](https://stackoverflow.com/questions/24423162/how-to-send-an-array-over-a-socket-in-python)

[python](#)

pc-or-terminal1> python3 server_simple_message.py

Host: , Port: 12345

Connected by ('127.0.0.1', 63255)

Client Says: Hello, world

pc-or-terminal2> python3 client_simple_message.py

Server replied: Received!

● ● ● client_simple_message.py

```
import socket

#host = socket.gethostname()
host = "localhost"
#host = "192.168.1.131"

port = 12345          # The same port as used by the server

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((host, port))

s.sendall('Hello, world'.encode()) # we send this string

data = s.recv(1024) # we received the answer
print('Server replied: ', data.decode())

s.close()
```

● ● ● server_simple_message.py

```
import socket

host = ''             # Symbolic name meaning all available interfaces
port = 12345          # Arbitrary non-privileged port

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((host, port))

print("Host: ", host, ", Port: ", port)

s.listen(1)
conn, addr = s.accept()

print('Connected by ', addr)

while True:
    try:
        data = conn.recv(1024)

        if not data: break

        print("Client Says: ", data.decode()) # we received this

        conn.sendall('Received!'.encode()) # we reply

    except socket.error:
        print("Error Occured.")
        break

conn.close()
```