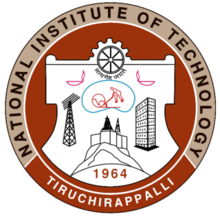
**National Institute of Technology, Tiruchirappalli**

****

**CA727 COMPUTER NETWORKS**

**Voice chat using Socket Programming (TCP) and Pyaudio module**

Submitted to: Submitted by:

Dr. Ruso T Akriti Upadhyay (**205120007**)

Paras Tiwari (**205120069**)

MCA - 3rd Sem

**AIM: Voice chat using Socket Programming (TCP) and Pyaudio module**

**Basic Idea:** Voice Chat that allows multiple people to communicate over the internet using their microphones

**Technologies Used:**

* Python 3
* PyAudio
* Socket Module (standard library)
* Threading Module (standard library)

**Requirements:** requires PyAudio and Port Audio modules

**PyAudio:**

* PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library.
* With PyAudio, we can easily use Python to play and record audio on a variety of platforms.

**Port Audio:**

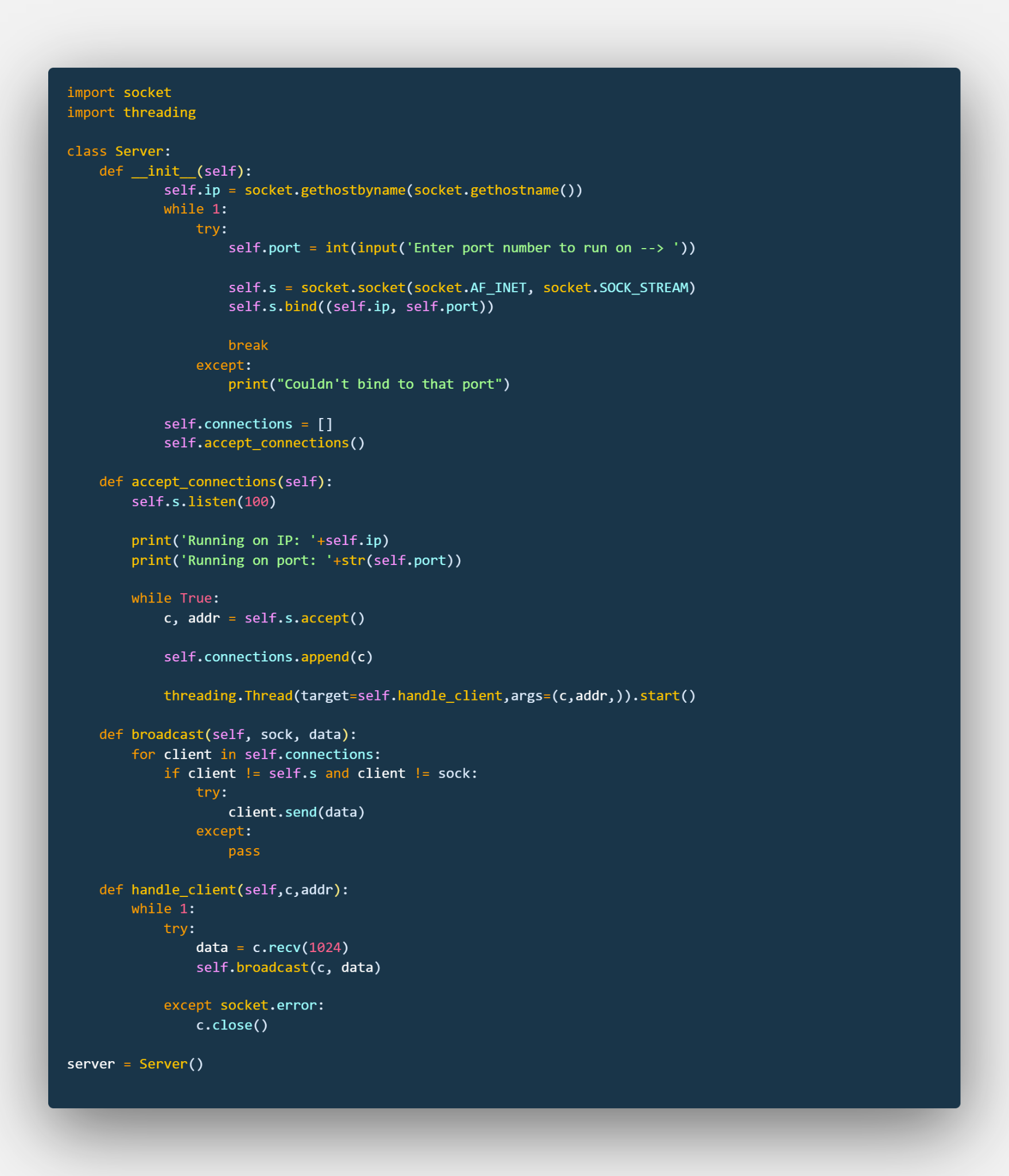
* PortAudio is a portable audio I/O library designed for cross-platform support of audio.
* It uses a callback mechanism to request audio processing.
* Audio can be generated in various formats, including 32 bit floating point, and will be converted to the native format internally.

**How to operate/Work:**

1. Run server.py or server.exe specifying the port number.
2. To use this program across the internet (using Cloud services such as AWS or GCloud), create an instance and make sure to keep it publically accessible. Using its external IP (using PuTTy) or using the instances SSH terminal instantiate and run the server.py file.
3. Clients can connect across the internet by entering your public IP (external IP address of the server) and the port the machine is running on or in the same network by entering the IP displayed on the server.
4. If the client displays "Connected to Server", you can now communicate with others in the same server by speaking into a connected microphone.

**Code:**

* server.py

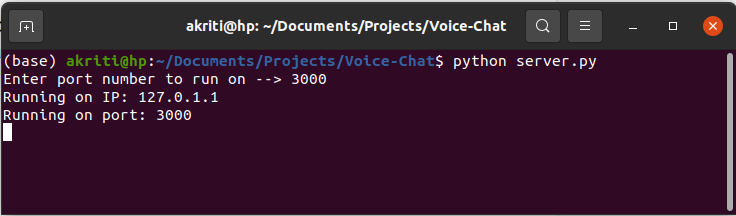


* client.py

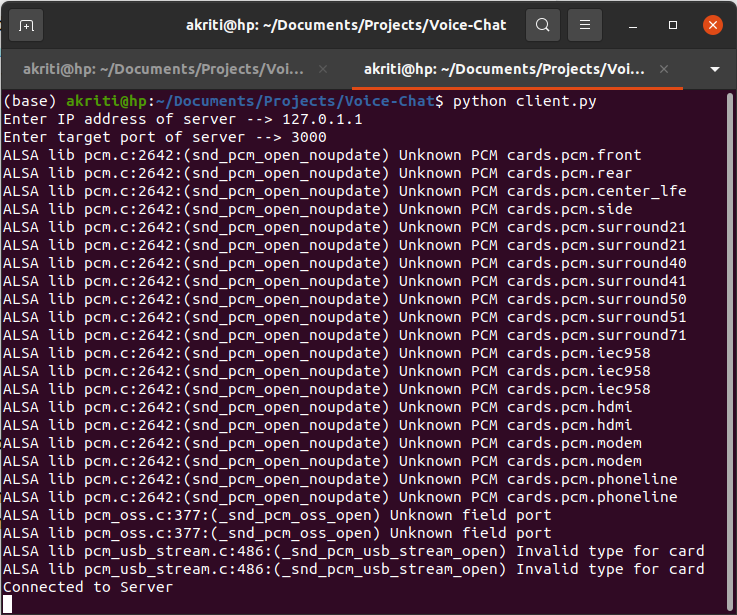


**Working Demo:**

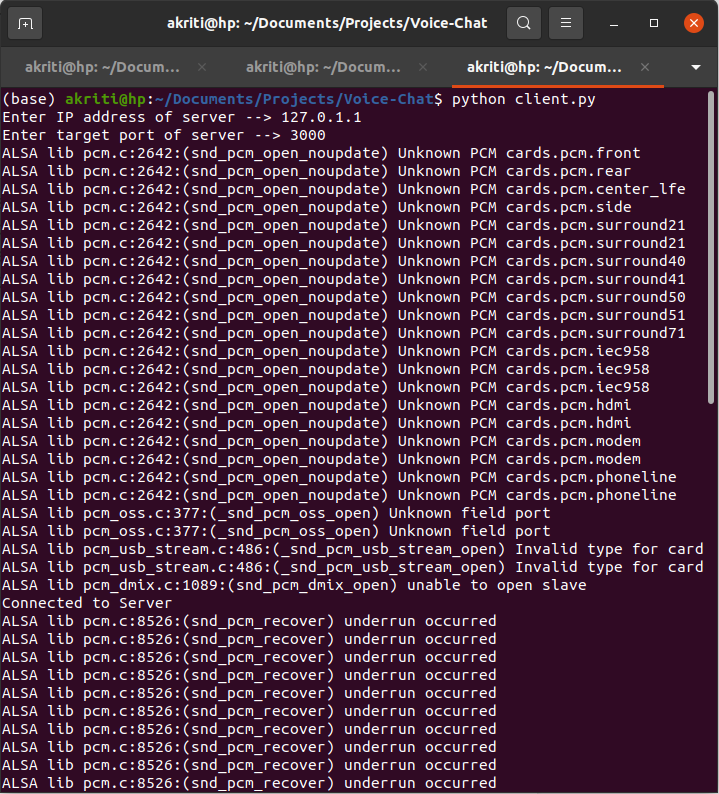
* Server running:

****

* Client 1:

****

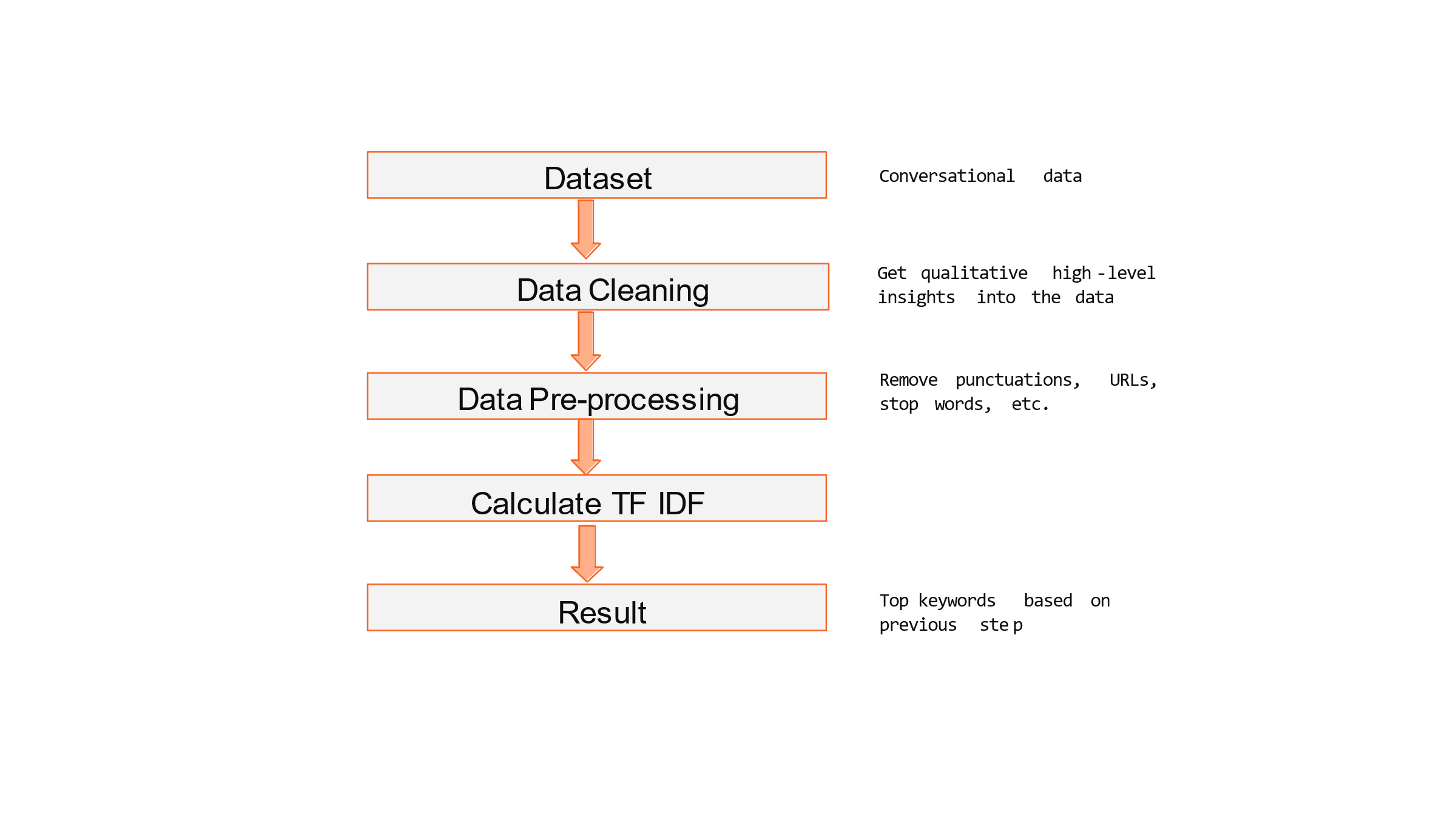
* Client 2:

****

**Note:** We need at least two clients (one speaker, one listener), as minimum two people are required to hold a conversation

**Extension to the project:**

**Workflow:**



**TF-IDF:**

TF-IDF stands for **“Term Frequency — Inverse Document Frequency”**. This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining.

**Raw Data:**

**Text

Description automatically generated**

**Preprocessed Data:**

**Text

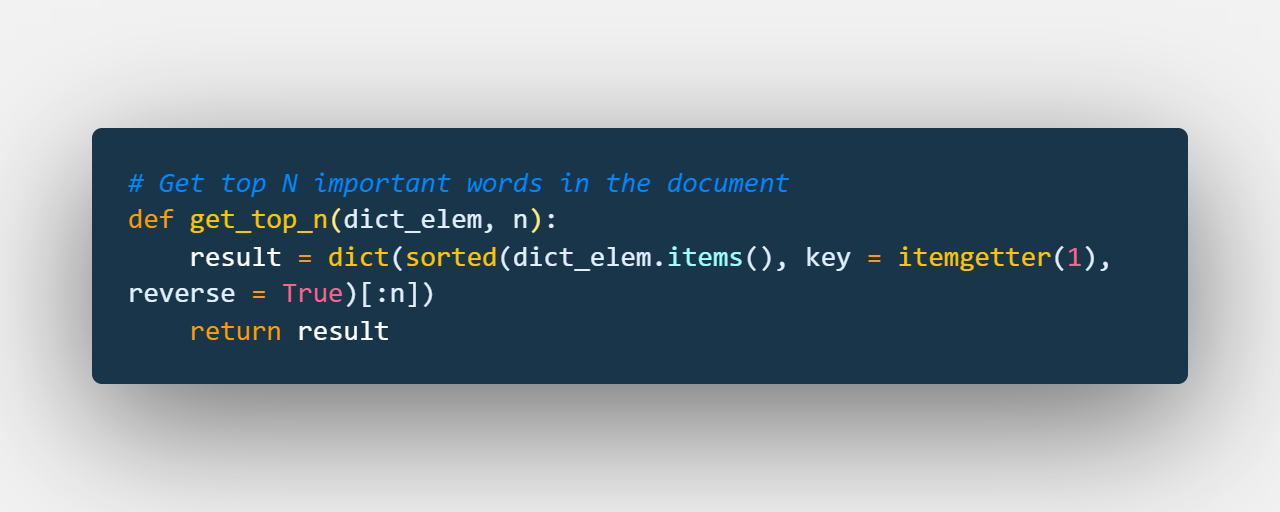
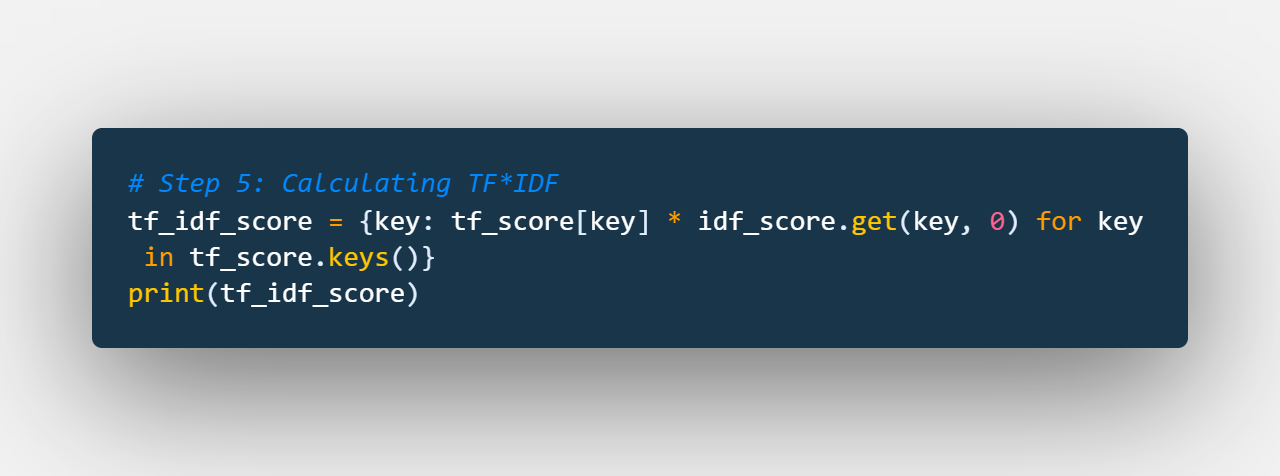
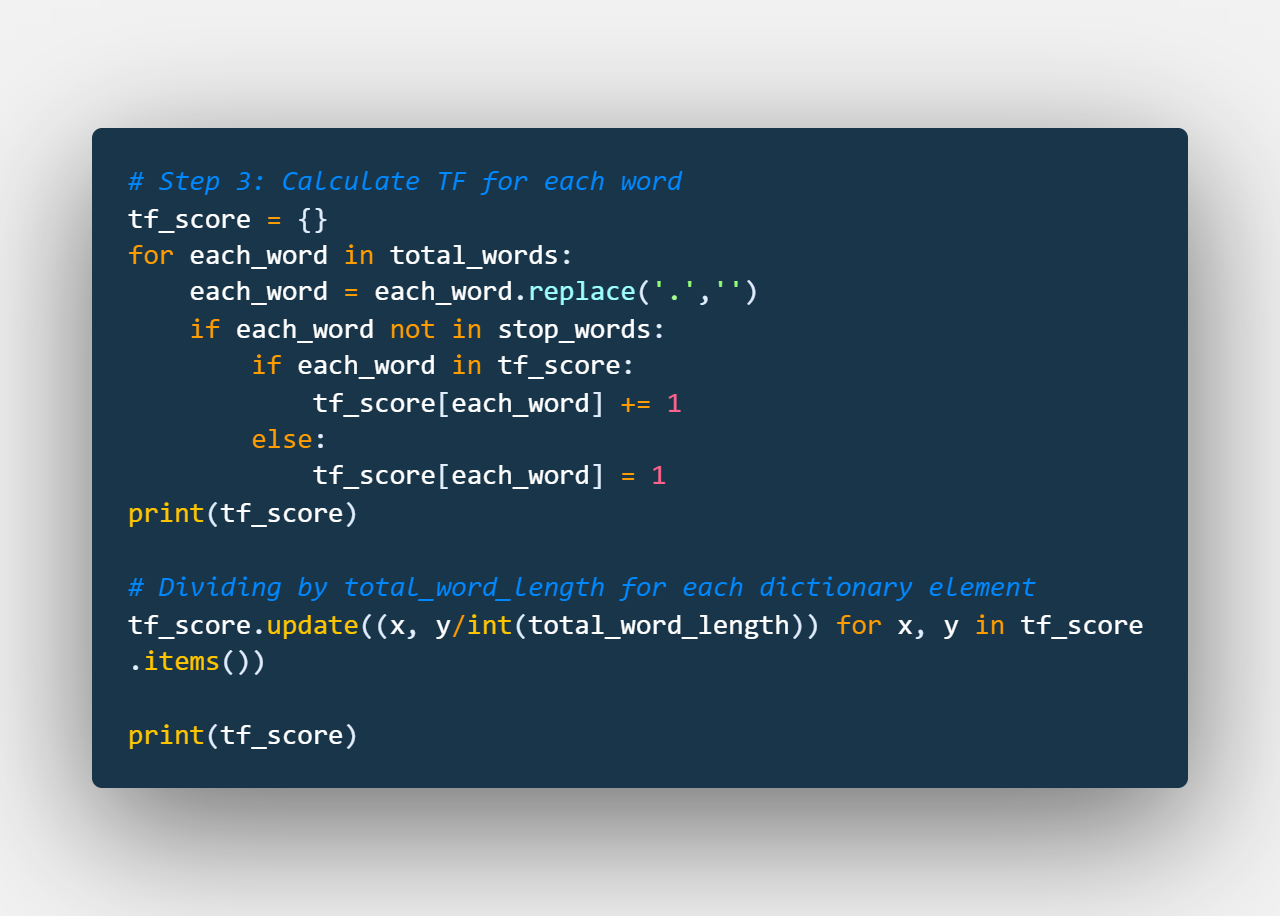
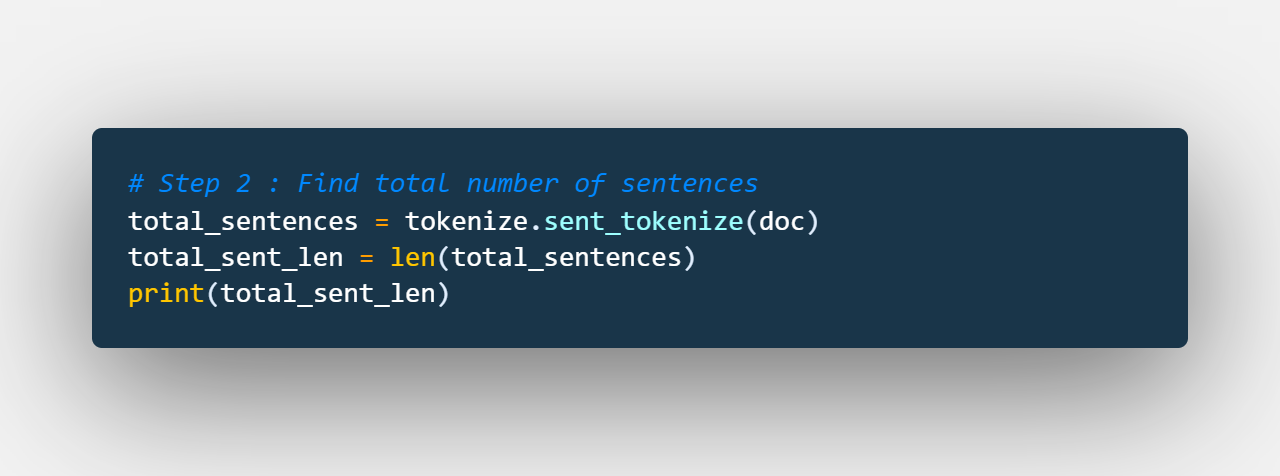
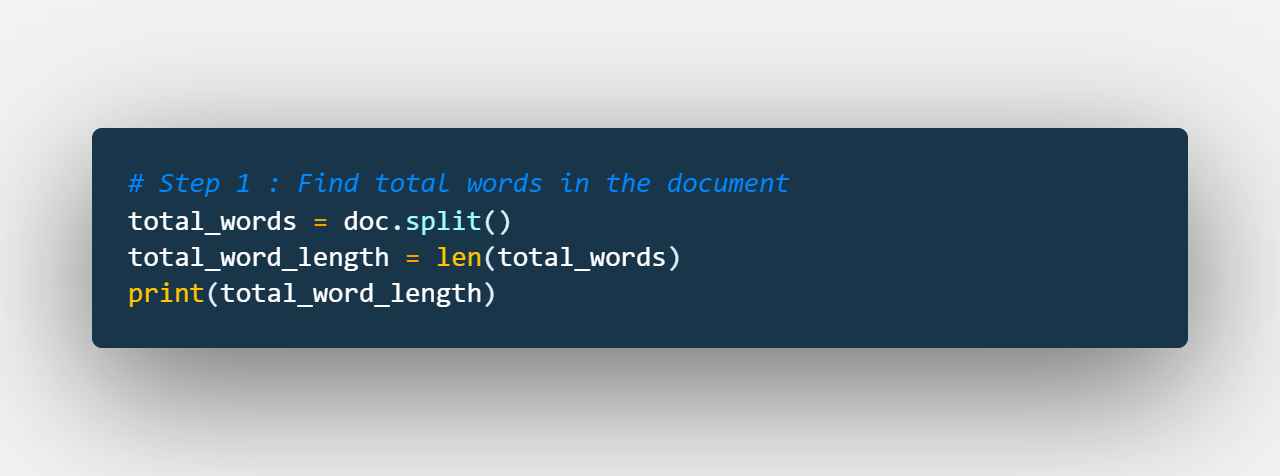
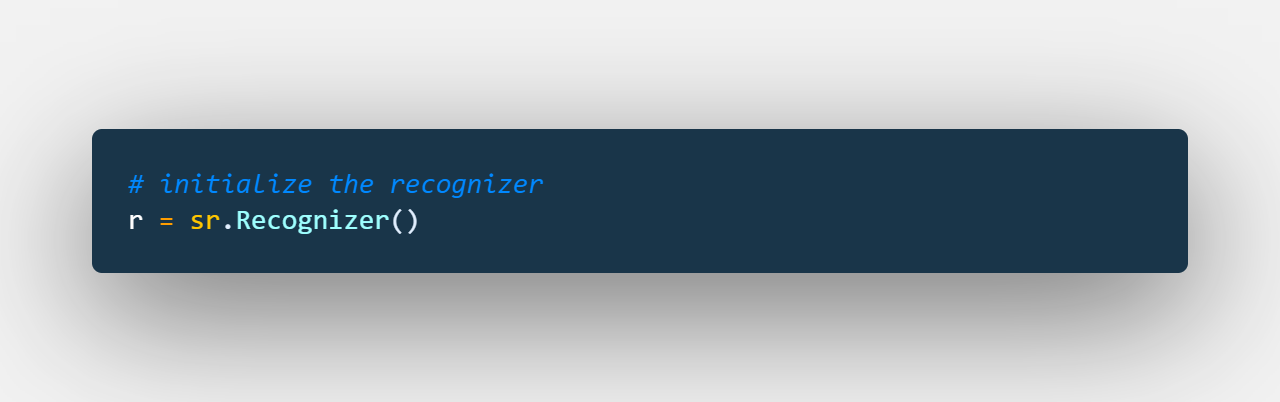
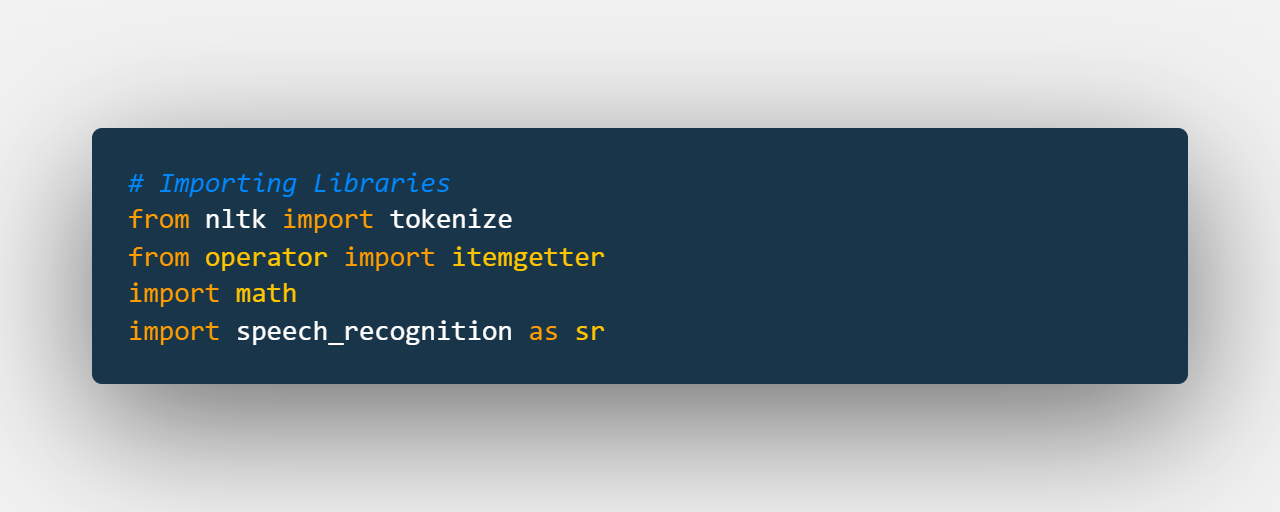
Description automatically generated**

**Result:**

**Text

Description automatically generated**

**Code:**



**X-X-X-X**