**SIMPLE CHAT ROOM AVEC PYTHON**

1. Contexte et définition du problème

Les forums de discussion ont été utilisés pendant des années comme moyen de socialiser avec les autres. Les entreprises utilisent désormais des forums de discussion pour se connecter avec des clients, d'autres bureaux et des clients potentiels.

### Objectif :

### Je voulais offrir la facilité de communiquer entre des client dans un même serveur et ça va récupérais beaucoup de problèmes.

### Le chat room vise spécifiquement sur :

### Créer un espace d’échanges et de discussion

### Favoriser des échanges sur des questions sur certaine situation.

**Comment configurer un simple serveur de Chat Room et permettre à plusieurs clients de s'y connecter à l'aide d'un script côté client ?**

**Le code utilise le concept de sockets et de threading**.

### ****Socket programming :****

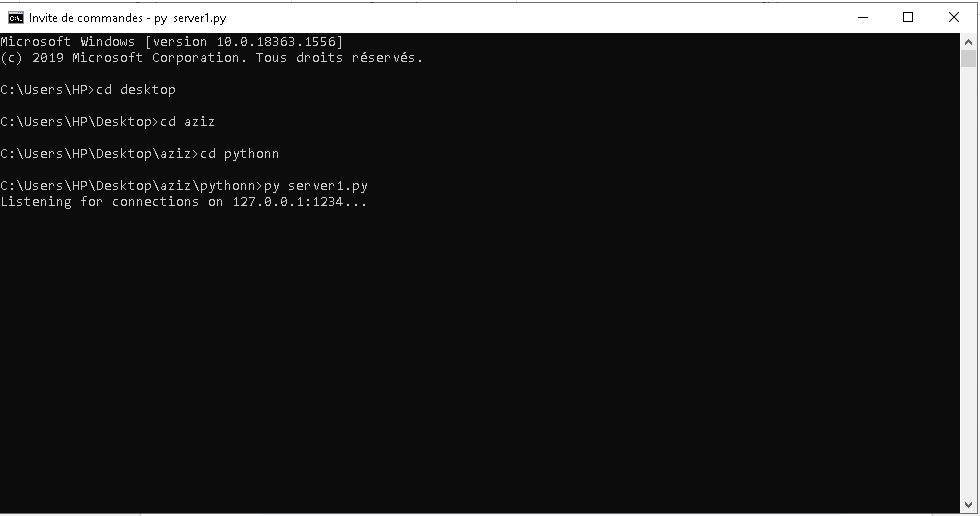
Les sockets peuvent être considérés comme des points de terminaison dans un canal de communication qui est bidirectionnel et établit une communication entre un serveur et un ou plusieurs clients.

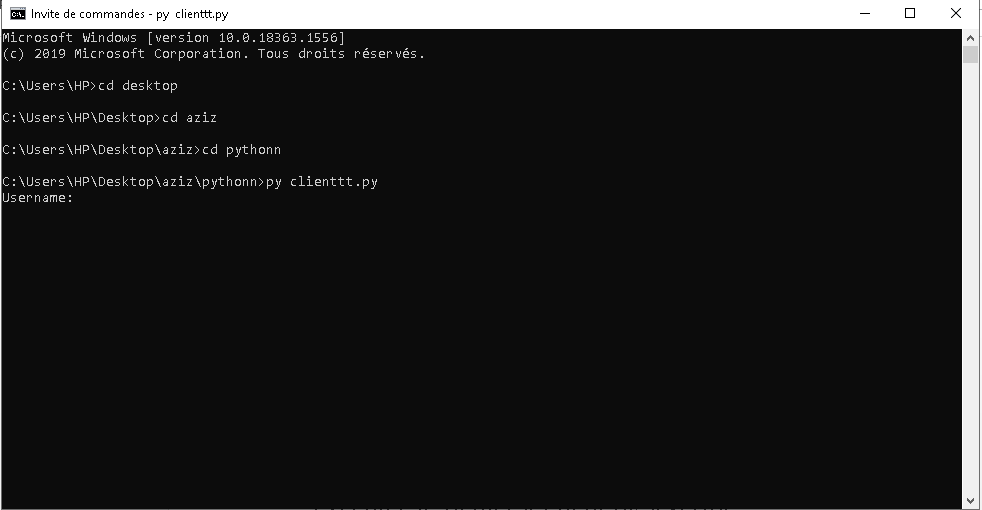
### ****Multi-Threading :****

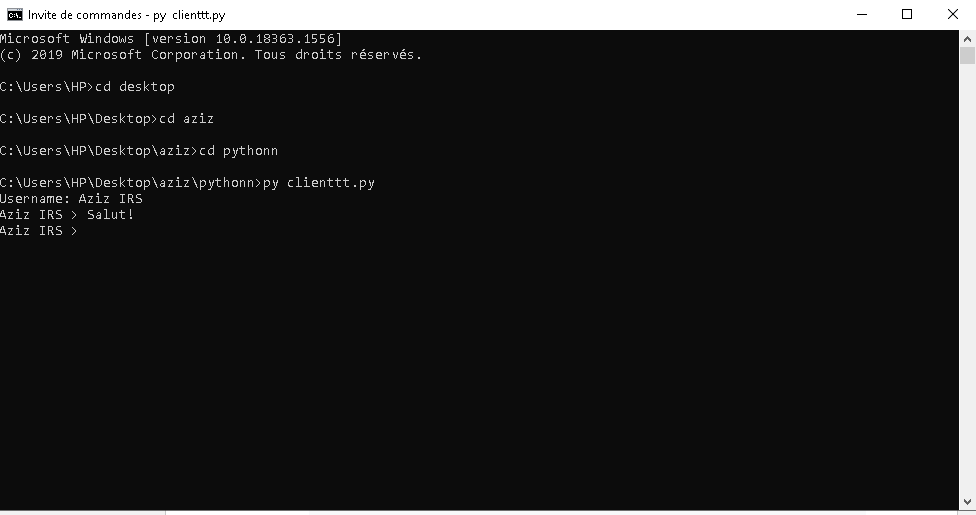
Un thread est un sous-processus qui exécute un ensemble de commandes individuellement de tout autre thread. Ainsi, chaque fois qu'un utilisateur se connecte au serveur, un thread séparé est créé pour cet utilisateur, et la communication du serveur au client a lieu le long de threads individuels basés sur des objets socket créés pour l'identité de chaque client.

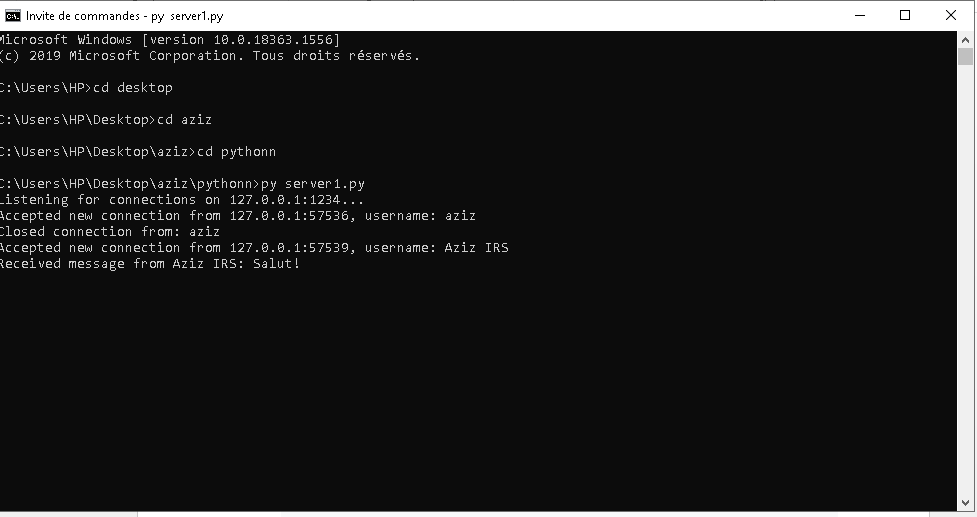
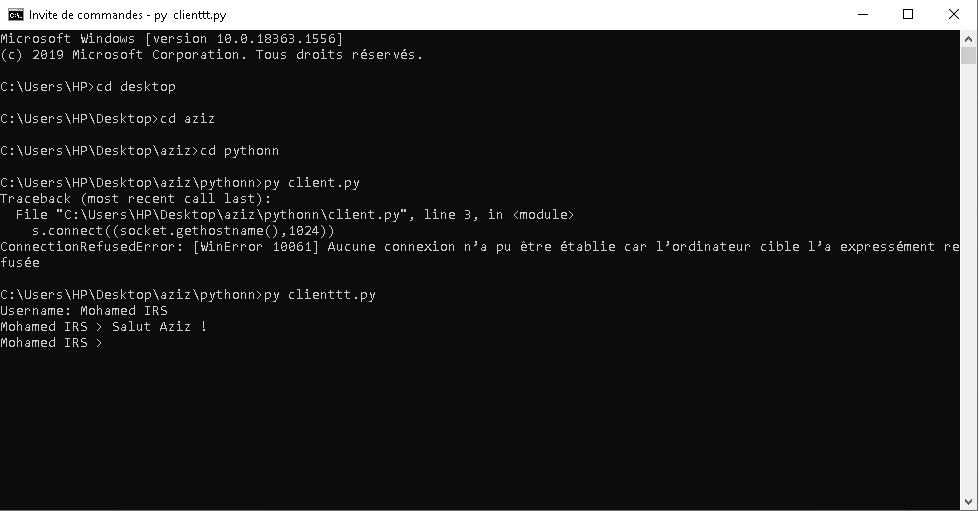
### Périmètre :

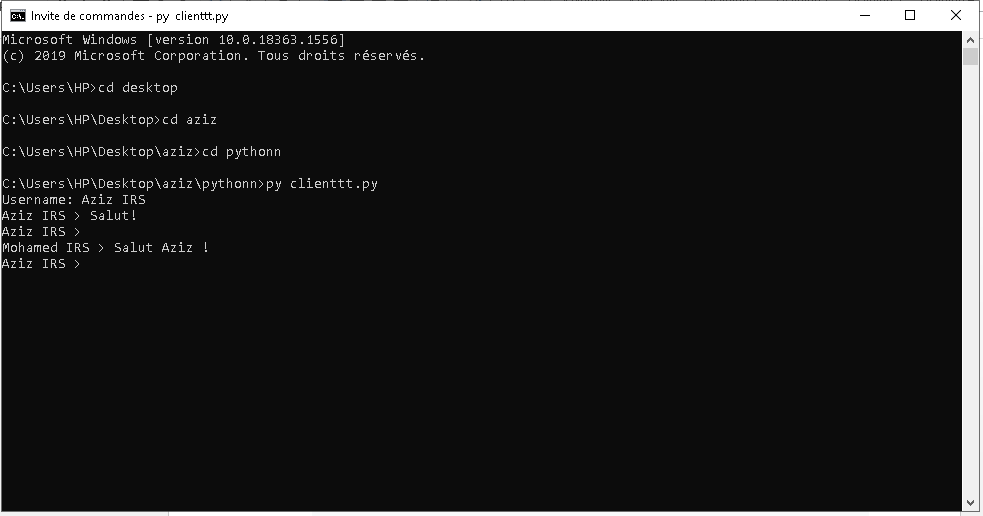
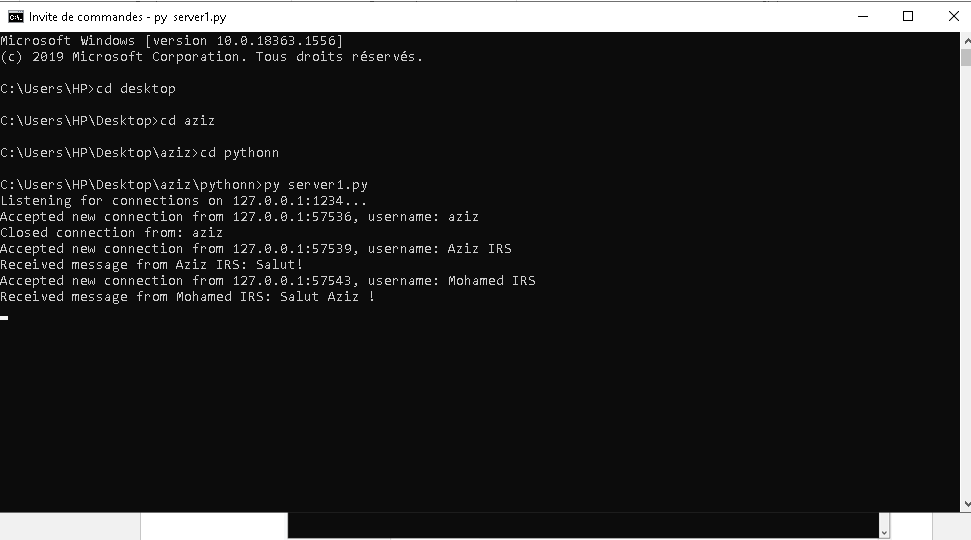
Nous aurons besoin de deux scripts pour établir cette salle de discussion. Un pour maintenir le service en cours d'exécution, et un autre que chaque client doit exécuter afin de se connecter au serveur.

1. Description fonctionnelle des besoins :  
     
    **Server Side Script :**  
      
   Le script côté serveur tentera d'établir un socket et de le lier à une adresse IP et un port spécifiés par l'utilisateur Le script restera alors ouvert et recevra les demandes de connexion et ajoutera les objets socket respectifs à une liste pour garder une trace des connexions actives. Chaque fois qu'un utilisateur se connecte, un fil distinct sera créé pour cet utilisateur. Dans chaque fil, le serveur attend un message et envoie ce message aux autres utilisateurs actuellement sur le chat.  
     
   **le script côté serveur doit être exécuté à tout moment pour que la salle de discussion continue de fonctionner.  
     
    Client-side script :** Le script côté client tentera simplement d'accéder au socket du serveur créé à l'adresse IP et au port spécifié. Une fois connecté, il vérifiera en permanence si l'entrée provient du serveur ou du client, et redirigera en conséquence la sortie. Si l'entrée provient du serveur, il affiche le message sur le terminal. Si l'entrée provient de l'utilisateur, il envoie le message que l'utilisateur entre au serveur pour qu'il soit diffusé aux autres utilisateurs.
2. OUT PUT de CMD :  
   ouvre le CMD :  
     
      
   cédez vers l’emplacement de vos fichiers et execute « server.py » script :   
   

maintenant , ouvre une autre fenêtre cmd pour exécuter le fichier « client.py » script :   
  
ensuite , entrer un userName pour envoyer des messages vers le serveur :



Le serveur a recevoir le nom de l’username connecté et son message envoyé :   
  
  
maintenant, on entre un nouveau client par ouvrir une autre fenêtre cmd et exécuter le fichier « client.py » une autre fois , et envoyé un nouveau message :  
  
  
le serveur va recevoir le message envoyé et les autres client va aussi le recevoir :

« Le script de serveur et client seront envoyés par un fichier compressé »

### 