

```
import socket, platform, os

SRV_ADDR = ""
SRV_PORT = 1234
```

Importiamo il modulo socket, e impostiamo l'indirizzo IP del server e la porta che il server ascolterà

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((SRV_ADDR, SRV_PORT))
```

Creazione del socket di tipo IPV4 e TCP, e associazione all'IP e alla porta specifica

```
s.listen(1)
connection, address = s.accept()
```

Mette il socket in modalità ascolto . Il parametro 1 indica che il server può accettare una sola connessione in entrata alla volta. Accettazione di una connessione e fino all'ottenimento della connessione e dell'indirizzo IP del client

```
print ("client connected: ", address)
```

Stampa un messaggio per indicare che un client è stato connesso e visualizza l'indirizzo dello stesso

```
while 1:
    try:
        data = connection.recv(1024)
    except:continue
```

Avvia un ciclo infinito che gestisce le richieste del cliente, tenta di ricevere dati e prosegue in caso di eccezione.

```
if(data.decode('utf-8') == '1'):
    tosend = platform.platform() + " " + platform.machine()
    connection.sendall(tosend.encode())
elif(data.decode('utf-8') == '2'):
    data = connection.recv(1024)
    try:
        filelist = os.listdir(data.decode('utf-8'))
        tosend = ""
        for x in filelist:
            tosend += "," + x
    except:
        tosend = "Wrong path"
    connection.sendall(tosend.encode())
elif(data.decode('utf-8') == '0'):
    connection.close()
    connection, address = s.accept()
```

Le richieste del cliente vengono elaborate in base ai dati ricevuti: se è '1', vengono inviate informazioni sulla piattaforma; se è '2', vengono elencati i file nella directory specificata; se è '0', viene chiusa la connessione corrente e ne avviene nuovamente un magari con un altro client