

#threading_server.py: threading 모듈을 이용한 TCP 채팅 서버 프로그램

```
import socket
import threading

#데이터를 수신하여 모든 클라이언트에게 다시 전송하는 함수
#연결이 종료된 소켓은 연결 소켓 목록에서 제거
def handler(c, a): #c: 소켓, a: 주소
    global connections # 연결 소켓 목록

    while True:
        try:
            data = c.recv(1024)
            if not data: #데이터가 없으면 연결이 종료된 것으로 소켓 제거
                connections.remove(c)
                c.close()
                continue
        except:
            continue

        for connection in connections:
            try:
                connection.send(data) #모든 클라이언트에게 데이터 전송
            except:
                continue

    # 메인 함수
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.bind("", 2500)
    sock.listen(1)

    #서버와 연결된 클라이언트 목록
    connections = []
    print("Waiting for clients")

    #연결 요청을 수락하고 데이터 송수신을 서브스레드로 실행
    while True:
        c, a = sock.accept()

        #스레드 객체를 생성하고 실행
        cThread = threading.Thread(target=handler, args=(c,a))
        cThread.daemon = True #서브스레드가 종료되면 메인 스레드도 종료
        cThread.start()

        # 새 클라이언트를 목록에 추가
        connections.append(c)
        print("연결된 클라이언트: ",connections)
```