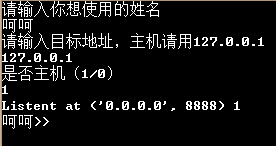
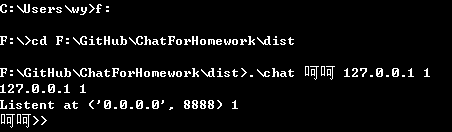
TCP聊天程序

# 简介：

本程序是一个用python编写的命令行TCP聊天室程序。主要是用于练习socket通信。因为技术原因，本程序使用明码通讯，请勿用于传递需要保密的信息。

# 使用方法：

1. 直接启动。  
   本程序可以直接启动，启动后会进入命令行界面，按照提示依次输入所需参数即可。  
     
   （注：姓名是指客户端的用户名，只要不重名用什么都可以；目标地址是指主机的IP地址；是否主机填0（否）或1（是））  
   然后就可以使用了。
2. 使用命令行参数启动。  
   本程序还可以使用命令行传递参数的方式启动。  
   请将命令行工作路径切换至程序目录后用以下命令启动：  
   ./chat 姓名 目标地址 [是否主机]  
   例如：  
     
   效果和直接启动时一样。
3. 启动后就可以输入想要传递的内容，所发表的内容会显示在所有用本程序连接到主机的客户端界面。

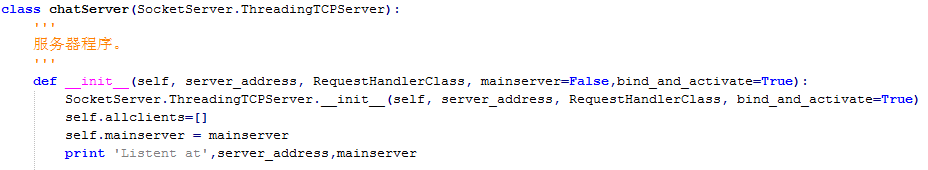
# 源码说明

## 综述

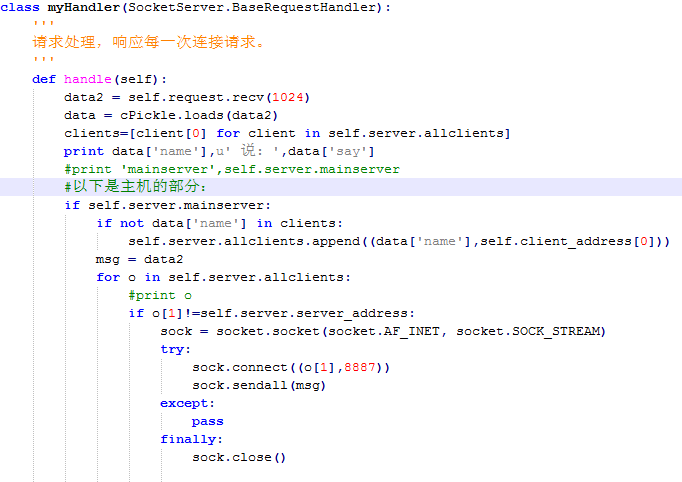
本程序使用python 2.73编写，采用了主从一体的模式。每个程序启动都会启动一个server模块（用于异步监听端口并打印收到的信息）和一个client模块（用于连接主机并对外发送信息）。所引用的模块有sys,socket,SocketServer,cPickle，均为python标准库。

## Server模块

Server模块主要分为两个部分：

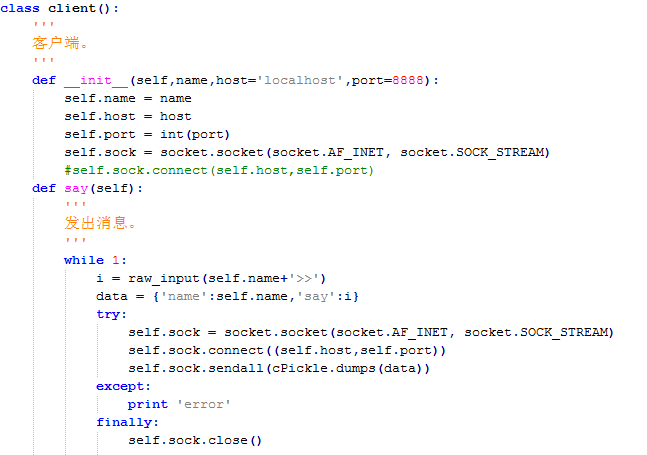
1、chatServer:  
   
 此类是从SocketServer.ThreadingTCPServer这个异步响应服务器类派生的，主要的功能是循环监听端口并将请求传递给Handler。我在其中加入了allclients和mainserver两个属性，分别用于存储所连接的客户端信息和主机标识。

2、myHandler:



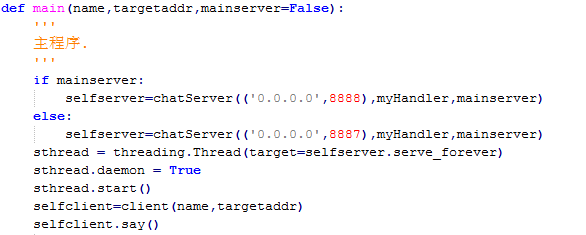
chatServer收到的请求会新建一个线程并被传递到这里处理。其功能是接收收到的信息并将其打印在屏幕上。主机的话还有一个向所有客户端转发的功能。

## Client模块：



如图所示：client模块只有一个功能——连接主机并发出信息。

## 程序运行逻辑：



如图，程序启动后在新线程先建立一个server实例，用于监听8888（主机）或8887（客户端）端口，随后建立一个client实例向外发信。

# 不足之处

1. 没有GUI。（真的不会做）
2. 只能容许五个人同时说话。（多少人登录都没关系，只要不同时说话就行）
3. 因为响应是异步的，而且没做资源锁，所以多人同时讲话时会出现所讲的话混杂在一起的情况。
4. 对话向所有人转发，不能单独聊天。（下一版会改善）