# Python实现socket

# 服务端实现

## 1、接入流程

**（1）导入库**

from socket import \*

**（2）创建socket**

SOCK\_DGRAM:UDP协议

SOCK\_STREAM:TCP协议

AF\_INET:IPV4协议

Tcp\_server = socket(AF\_INET, SOCK\_DGRAM)

**（3）绑定IP**

# 0.0.0.0 允许任意ip访问

# 127.0.0.1 允许本地连接

# 绑定IP，端口；这里是一个元组

Tcp\_server.bind(("", 6789))

**（4）等待连接**

# 创建一个客户端，这里返回的是2个值

# sock: 接口

# addr: 地址

sock, addr = Tcp\_server.accept()

**（5）获取数据**

这里说明一下，数据接收需要服务器与客户端相配合，需要共同**商议结束标识**

# 数据有可能大于1024，循环获取

#

data = ‘’ # 这不是

while True:

tmp\_data = sock.recv(1024) # 从sock中获取1024个字符串

If tmp\_data:

data += tmp\_data.decode(“utf-8”)

if tmp\_data.decode(“utf-8”).endswith(“#”):

break

else:

break;

**（6）发送数据**

Sock.send(b”body”)

Sock.send(“body”,encode(“utf-8”))

**（7）关闭连接**

Sock.close()

---------UDP不用下面2句

UdpSocker.listen() #监听

----------

recvInfor = udpSocket.recvfrom(1024)#接受数据，1024字节

recvInfor[0].decode("gb2312")#解码

Recv/send:TCP协议

Recvfrom/sendto : UDP协议

Send(b””)#必须加个B

# 客户端实现

1. 接入流程

（1）引入库

From socket import \*

（2）创建socket

client = socket(AF\_INET,SOCK\_DGRAM)

（3）连接到服务

client.connect((“127.0.0.1”, 8000))

（4）发送数据

client.sendto(sendData.encode("gb2312"), (IP, Port))#发送，设置编码

Udpsocker.recvfrom(1024)

类如在收数据的同时还要发数据，多线程/进程