

Example app: UDP client

Python UDPClient

include Python's socket
library

```
from socket import *  
serverName = 'hostname'  
serverPort = 12000
```

create UDP socket for
server

```
clientSocket = socket(AF_INET,  
                      SOCK_DGRAM)
```

get user keyboard
input

```
message = raw_input('Input lowercase sentence:')  
clientSocket.sendto(message.encode(),
```

Attach server name, port to
message; send into socket

```
(serverName, serverPort))  
modifiedMessage, serverAddress =
```

read reply characters from
socket into string

```
clientSocket.recvfrom(2048)  
print modifiedMessage.decode()  
clientSocket.close()
```

print out received string
and close socket

Example app: UDP server

Python UDPServer

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print ("The server is ready to receive")
while True:
    message, clientAddress = serverSocket.recvfrom(2048)
    modifiedMessage = message.decode().upper()
    serverSocket.sendto(modifiedMessage.encode(),
                        clientAddress)
```

create UDP socket →

bind socket to local port
number 12000 →

loop forever →

Read from UDP socket into
message, getting client's
address (client IP and port) →

send upper case string
back to this client →

Example app: TCP client

Python TCPClient

```
from socket import *
serverName = 'servername'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = raw_input('Input lowercase sentence:')
clientSocket.send(sentence.encode())
modifiedSentence = clientSocket.recv(1024)
print ('From Server:', modifiedSentence.decode())
clientSocket.close()
```

create TCP client socket

Connect to remote server

No need to attach server
name, port

Example app: TCP server

Python TCPServer

create TCP welcoming
socket

server begins listening for
incoming TCP requests

loop forever

server waits on accept()
for incoming requests, new
socket created on return

read bytes from socket (but
not address as in UDP)

close connection to this
client (but *not* welcoming
socket)

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', serverPort))
serverSocket.listen(1)
print 'The server is ready to receive'
while True:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence.
                           encode())
    connectionSocket.close()
```