

NAME: NIRAJ THANKI

SID: 19376

CLASS: CS531

SERVER.py

```
C:\Users\Niraj_Home\Desktop\HW\Q1\Server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Client.py x Server.py
1 server.py
2
3 import sys
4 import socket
5 import select
6
7 HOST = ''
8 SOCKET_LIST = []
9 RECV_BUFFER = 4096
10 PORT = 9009
11
12 def chat_server():
13
14     server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
15     server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
16     server_socket.bind((HOST, PORT))
17     server_socket.listen(10)
18
19
20     SOCKET_LIST.append(server_socket)
21
22     print "Chat server started " + str(PORT)
23
24     while 1:
25
26
27         ready_to_read, ready_to_write, in_error = select.select(SOCKET_LIST, [], [], 0)
28
29         for sock in ready_to_read:
30
31             if sock == server_socket:
32                 sockfd, addr = server_socket.accept()
33                 SOCKET_LIST.append(sockfd)
34                 print "Client (%s, %s) connected" % addr
35
36                 broadcast(server_socket, sockfd, "[%s] entered our chatting room\n" % addr)
37
38             else:
39
40                 try:
41
42                     data = sock.recv(RECV_BUFFER)
43                     if data:
44                         broadcast(server_socket, sock, "\r" + '[' + str(sock.getpeername()) + ']' + data)
45                     else:
46                         if sock in SOCKET_LIST:
47                             SOCKET_LIST.remove(sock)
48
49                         broadcast(server_socket, sock, "Client (%s, %s) is Not Available\n" % addr)
50
51                 except:
52                     broadcast(server_socket, sock, "Client (%s, %s) is Not Available\n" % addr)
53                     continue
54
55     server_socket.close()
56
57
58
59
60
61
Line 1, Column 2 Spaces: 4 Python
7:12 PM 11/8/2017
```

```
62
63 def broadcast (server_socket, sock, message):
64     for socket in SOCKET_LIST:
65
66         if socket != server_socket and socket != sock :
67             try :
68                 socket.send(message)
69             except :
70
71                 socket.close()
72
73                 if socket in SOCKET_LIST:
74                     SOCKET_LIST.remove(socket)
75
76 if __name__ == "__main__":
77     sys.exit(chat_server())
78
```

Source Code:

#server.py

import sys

import socket

import select

HOST = "

SOCKET_LIST = []

RECV_BUFFER = 4096

PORT = 9009

```
def chat_server():

    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    server_socket.bind((HOST, PORT))
    server_socket.listen(10)

    SOCKET_LIST.append(server_socket)

    print "Chat server started " + str(PORT)

    while 1:

        ready_to_read,ready_to_write,in_error = select.select(SOCKET_LIST,[],[],0)

        for sock in ready_to_read:

            if sock == server_socket:
                sockfd, addr = server_socket.accept()
                SOCKET_LIST.append(sockfd)
                print "Client (%s, %s) connected" % addr

                broadcast(server_socket, sockfd, "[%s:%s] entered our chatting room\n" % addr)

            else:
```

```
try:
```

```
    data = sock.recv(RECV_BUFFER)
```

```
    if data:
```

```
        broadcast(server_socket, sock, "\r" + '[' + str(sock.getpeername()) + ']' + data)
```

```
    else:
```

```
        if sock in SOCKET_LIST:
```

```
            SOCKET_LIST.remove(sock)
```

```
        broadcast(server_socket, sock, "Client (%s, %s) is Not Available\n" % addr)
```

```
except:
```

```
    broadcast(server_socket, sock, "Client (%s, %s) is Not Available\n" % addr)
```

```
    continue
```

```
server_socket.close()
```

```
def broadcast (server_socket, sock, message):
```

```
    for socket in SOCKET_LIST:
```

```
        if socket != server_socket and socket != sock :
```

```
            try :
```

```
                socket.send(message)
```

```
            except :
```

```
socket.close()
```

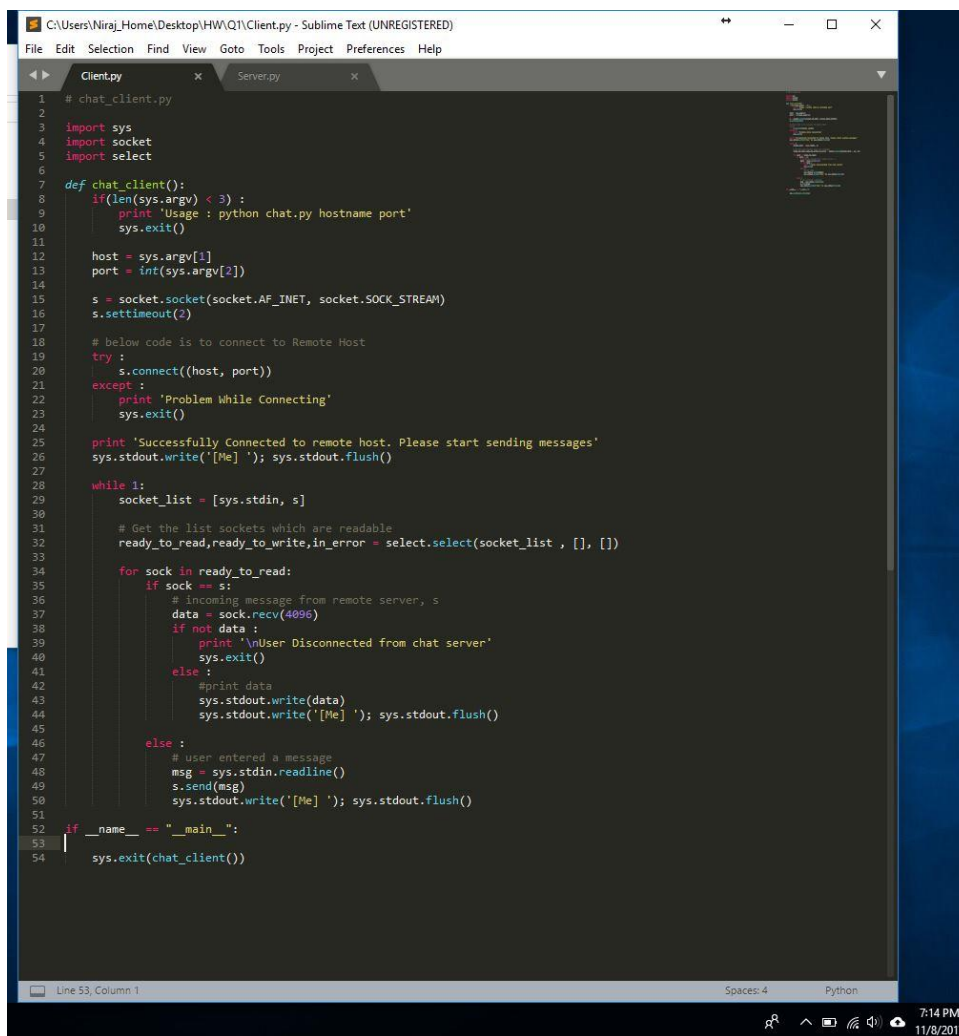
```
if socket in SOCKET_LIST:
```

```
    SOCKET_LIST.remove(socket)
```

```
if __name__ == "__main__":
```

```
    sys.exit(chat_server())
```

CLIENT.py



```
C:\Users\Niraj_Home\Desktop\HW\Q1\Client.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Client.py x Server.py x

1 # chat_client.py
2
3 import sys
4 import socket
5 import select
6
7 def chat_client():
8     if (len(sys.argv) < 3):
9         print 'Usage : python chat.py hostname port'
10        sys.exit()
11
12    host = sys.argv[1]
13    port = int(sys.argv[2])
14
15    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
16    s.settimeout(2)
17
18    # below code is to connect to Remote Host
19    try:
20        s.connect((host, port))
21    except:
22        print 'Problem While Connecting'
23        sys.exit()
24
25    print 'Successfully Connected to remote host. Please start sending messages'
26    sys.stdout.write('[Me] '); sys.stdout.flush()
27
28    while 1:
29        socket_list = [sys.stdin, s]
30
31        # Get the list sockets which are readable
32        ready_to_read, ready_to_write, in_error = select.select(socket_list, [], [])
33
34        for sock in ready_to_read:
35            if sock == s:
36                # incoming message from remote server, s
37                data = sock.recv(4096)
38                if not data:
39                    print '\nUser Disconnected from chat server'
40                    sys.exit()
41                else:
42                    # print data
43                    sys.stdout.write(data)
44                    sys.stdout.write('[Me] '); sys.stdout.flush()
45
46            else:
47                # user entered a message
48                msg = sys.stdin.readline()
49                s.send(msg)
50                sys.stdout.write('[Me] '); sys.stdout.flush()
51
52    if __name__ == "__main__":
53
54        sys.exit(chat_client())

Line 53, Column 1 Spaces: 4 Python
7:14 PM 11/8/2017
```

Source Code:

```
# chat_client.py
```

```
import sys
```

```
import socket
```

```
import select
```

```
def chat_client():
```

```
    if(len(sys.argv) < 3) :
```

```
        print 'Usage : python chat.py hostname port'
```

```
        sys.exit()
```

```
    host = sys.argv[1]
```

```
    port = int(sys.argv[2])
```

```
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
    s.settimeout(2)
```

```
    # below code is to connect to Remote Host
```

```
    try :
```

```
        s.connect((host, port))
```

```
    except :
```

```
        print 'Problem While Connecting'
```

```
        sys.exit()
```

```
    print 'Successfully Connected to remote host. Please start sending messages'
```

```
    sys.stdout.write('[Me] '); sys.stdout.flush()
```

```
    while 1:
```

```

socket_list = [sys.stdin, s]

# Get the list sockets which are readable
ready_to_read,ready_to_write,in_error = select.select(socket_list , [], [])

for sock in ready_to_read:
    if sock == s:
        # incoming message from remote server, s
        data = sock.recv(4096)
        if not data :
            print '\nUser Disconnected from chat server'
            sys.exit()
        else :
            #print data
            sys.stdout.write(data)
            sys.stdout.write('[Me] '); sys.stdout.flush()

    else :
        # user entered a message
        msg = sys.stdin.readline()
        s.send(msg)
        sys.stdout.write('[Me] '); sys.stdout.flush()

if __name__ == "__main__":

    sys.exit(chat_client())

```

Output:

```
# Error - Page Not Found
# Student Index
# Exercises for Network Pro...
npu85.nyu.edu/~henry/npy/classes/python/net_prog/slide/es...
17 And coming to your name server for popular resou...
2) Add logging capability to your name server, keepin...
3) Your name server should periodically "ping" the ...
You may implement real services for the servers that...
6-14. Error Checking and Graceful Shutdown. All o...
input, nor do we check other improper input to raw_in...
client/server pair of one of our examples, and add encl...
6-15. Asynchronicity and SocketServer. Take the exam...
output that your server is getting requests from both...
6-16. ...
File Edit View Search Terminal Help
niraj_ubuntu@niraj-ubuntu:~/cs531/Q1$ python Client.py localhost 9009
Connected to remote host. You can start sending messages
b) Int[Me] How is everything?
['127.0.0.1', 57934]] I Miss you!
Please [['127.0.0.1', 57934]] I miss you too.Why do you contact me?
[Me] I met a very nice girl.I am wondering weather my allowance can be increased
???
How [['127.0.0.1', 57934]] ???
[Miss me?]

????
refered
o S
ycha
o P

modified on: 10/27/2017 09:06:45
```