Chat Room Project

threading Module In Python

- **threading** module is used for creating, controlling and managing threads in python.
- **Thread class constructor**

```
Thread(group=None, target=None, name=None, args=(), kwargs={})
```

- **start() method**
- This method is used to start the thread's activity. When we call this method, internally the run() method is invoked which executes the target function or the callable object.

Thread class constructor

- group: Should be None. It is reserved for future extension.
- target: This is the callable object or task to be invoked by the run() method(the function names)
- name: This is used to specify the thread name. By default, a unique name is generated following the format Thread-N, where N is a small decimal number.
- args: This is the argument tuple for the target invocation. We can provide values in it which can be used in the traget method. It's default value is empty, i.e. ()
- kwargs: This is keyword argument dictionary for the target invocation. This defaults to {}.

Threading example

```
import time
import threading
def thread1(i):
    time.sleep(3)
    print('No. printed by Thread 1: %d' % i)
def thread2(i):
    print('No. printed by Thread 2: %d' % i)

if __name__ == '__main__':
    t1 = threading.Thread(target=thread1, args=(10,))
    t2 = threading.Thread(target=thread2, args=(12,))
    # start the threads
    t1.start()
    t2.start()
```

LISTS IN PYTHON

- Lists - collections of data
- numbers = [10, 5, 7, 2, 1]
- **Adding elements to a list:**
  - list.**append**(value)
    - A new element may be *glued* to the end of the existing list
    - The list's length then increases by one.
  - The insert() method is a bit smarter - it can add a new element at any place in the list, not only at the end.
    - list.**insert**(location, value)

Removing elements from a list

- clear(): Remove all items
- pop() : Remove an item by index and get its value
- remove(): Remove an item by value
- Del: Remove items by index or slice

Making use of lists

```
my_list = [10, 1, 8, 3, 5]
total = 0
for i in my_list:
    total += i
print(total)
```

Chat room project

- We need to create chat project , when any client send message this message will broad cast to all client
- The server should run thread for each client
- In client side receive and send will be in separate thread
- Nick name should be used for each client
-

Client

- Create socket object
- Connect to server
- Take the nick name from the user
- Receive function
- Write function
- Start thread for receive
- Start thread for write

Client: Receive function

- **def receive():**
  - **While true:**
    - **try:**
      - **#receive message from server**
      - **If message==‘NICK’:**
        - **Send the nick name to server**
      - **Else:**
        - **Print(message)**
    - **except:**
      - **Print receive error**
      - **Close connection**
      - **break**

Client Write function

- **def Write():**
  - **While true:**
    - **try:**
      - **#read message from user**
      - **Message=nick name+message**
      - **Send message to server**
    - **except:**
    - **print write error**

Client Start thread for receive and write method

```
rec_Thread=threading.Thread(target=recive)
rec_Thread.start()

wr_Thread=threading.Thread(target=Write)
wr_Thread.start()
```

Server

- Create Socket object
- bind server to socket
- listen()
- Create list for clients and list for nicknames
- broadcast(message) function
- Handle(client) function
- Receive function
- Call receive function

Server: broadcast(message)

- def broadCast(message):
- For each client in clients list
  - Client.send(message)

Server: handle(client)

- While True:
  - Try:
    - Msg=Client.recv(1024)
    - broadcast(msg. decode())
  - except:
    - Print handle error
    - Remove client from clients list
    - Remove nick name from nickNames list
    - broadCast((nickName+" Left chat room").encode())
    - break

Server: recive() function

- While True:
  - Cli,add =server.accept()
  - Send ‘NICK’ to client
  - Recive the nick name from client
  - Add cli to clients list and nickname to nicknames list
  - broadCast(nickName+' join the chat room')
  - #start thread for handle function
  - thread=threading.Thread(target=handle,args=(cli,))
  - thread.start()

Task2

- Implement the chat room project

Thank You