

Solution: Write a UDP Chat App!

We'll cover the following ^

- Client
- Server

Client

The client program uses a `while` loop to keep the conversation with the server alive. Furthermore, it uses `connect()` to ensure that only one server is connected to, and only replies from that server are received.

```
1 import argparse, socket
2
3 MAX_SIZE_BYTES = 65535 # Maximum size of message
4
5 def client(port):
6     s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
7     host = '127.0.0.1'
8     while True:
9         s.connect((host, port))
10        message = input('Input your message: ')
11        data = message.encode('ascii')
12        s.send(data)
13        data = s.recv(MAX_SIZE_BYTES)
14        text = data.decode('ascii')
15        print('The server replied: %s' % text)
```



Server

```
1 import argparse, socket
2
3 MAX_SIZE_BYTES = 65535 # Maximum size of message
4
5 def server(port):
6     s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
7     hostname = '127.0.0.1'
8     s.bind((hostname, port))
```



```
9     print('Listening at {}'.format(s.getsockname()))
10    while True:
11        data, clientAddress = s.accept()
12        message = data.decode('utf-8')
13        print('The client at {} sent {}'.format(clientAddress, message))
14        msg_to_send = input('Input message to send: ')
15        data = msg_to_send.encode('utf-8')
16        s.sendto(data, clientAddress)
```

Great! Let's look at how server and client programs can be written to run on TCP in Python3 in the next lesson!