SOCKETS

PREVIOUS

This document was developed with an HP ProBook 450 G7 computer:

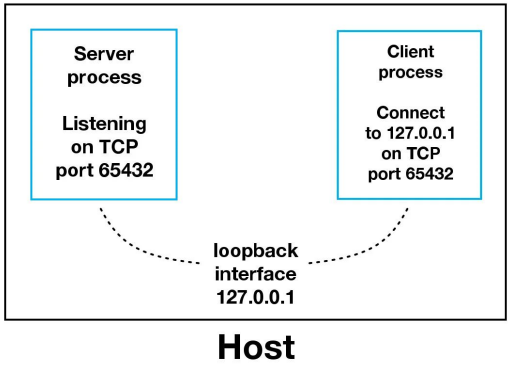
- CPU: Intel Core i5-10210U (4 cores and 8 threads)
- GPU: UHD Graphics
- RAM: 24 GB
- OS: Ubuntu 24.04 (Noble Numbat)

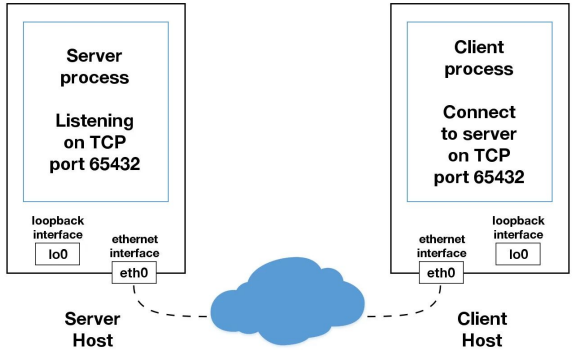
on fall 2024.

BASIC

A socket allows the transmission of data between two programs through a network, they can run on the same host or on different hosts.

To set up the connection, one computer must be running a program (server) that is waiting for a connection while other computer must be running other program (client) that must try to connect up to the first program (server). For the client to connect to the server, it must know the IP address of the computer that is running the server program and the port number that the server program is listening on.





https://docs.python.org/3/howto/sockets.html
https://docs.python.org/3/library/socket.html

SIMPLE HTTP CLIENT

client_http.py

```
import socket

# Create IPv4 (AF INET) socket as TCP (SOCK STREAM).
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Connection to google.com.
s.connect(("www.google.com", 80))

# HTTP is a client-server protocol,
# GET is an HTTP method for requesting data from the server.
s.send(b"GET /index.html HTTP/1.0\n\n")

# Receive data from the socket,
# bufsize should be a relatively small power of 2, for example 1024.
data = s.recv(1024)

# Show data.
print(data)

# Close socket.
s.close()
```

```
ricard@HP-ProBook:~$ python3 client_http.py
b'HTTP/1.0 200 OK\r\nDate: Fri, 29 Nov 2024 16:17:50
GMT\r\nExpires: -1\r\nCache-Control: private,
max-age=0\r\nContent-Type: text/html;
charset=ISO-8859-1\r\nContent-Security-Policy-Report-O
nly: object-src \'none\';base-uri \'self\';script-src
\'nonce-GkU1IQ25Dgs_85HBsm-0qw\' \'strict-dynamic\'
\'report-sample\' \'unsafe-eval\' \'unsafe-inline\'
https: http:;report-uri
https://csp.withgoogle.com/csp/gws/other-hp\r\nServer:
gws\r\nX-XSS-Protection: 0\r\nX-Frame-Options:
SAMEORIGIN\r\nAccept-Ranges: none\r\nVary:
Accept-Encoding\r\n\r\n<!doctype html><html
itemscope="" itemtype="http://schema.org/WebPage"
lang="es"><head><meta content="Google.es permite
acceder a la informaci\xf3n mundial en castellano,
catal\xe1n, gallego, euskara e ingl\xe9s."
name="description"><meta content="noodp, "
name="robots"><meta content="text/html; charset=UTF-8"
http-equiv="Content-Type"><meta
content="/images/branding/googleg/1x/googleg_standard_
color_128dp.png"
itemprop="image"><title>Google</title><script
nonce="GkU1IQ25Dgs_85HBsm-0qw">(function(){var
_g={kEI:\'LulJZ7LUG7nx0PE'
ricard@HP-ProBook:~$
```

SIMPLE HTTP SERVER

server_http.py

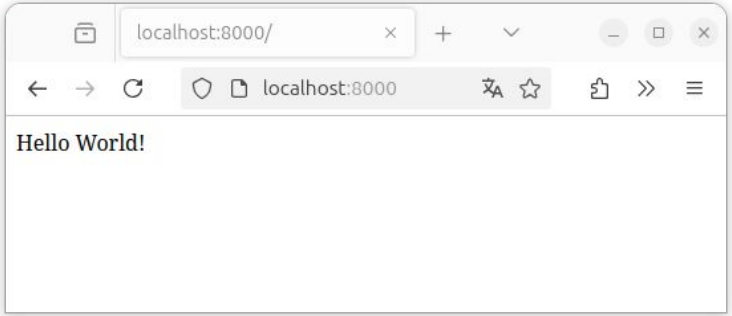
```
import datetime as dati
import socket

print(f"{dati.datetime.now()} server starts")

# Create IPv4 socket as TCP.
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Assigns localhost address and port 8000 to the socket.
server socket.bind(('127.0.0.1', 8000))
# Listen to at most one connection at a time.
server_socket.listen(1)

try:
    while True:
        print(f"{dati.datetime.now()} server is ready")
        # Set up a new connection from a client.
        client_socket, client_address = server_socket.accept()
        print(f"{dati.datetime.now()} server has established a connection")
        request = client socket.recv(1024)
        print(f"{dati.datetime.now()} request: {request}")
        response = b"HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\nHello World!"
        client_socket.send(response)
        print(f"{dati.datetime.now()} response: {response}")
        client socket.close()  # Because server_socket.listen(1).
except KeyboardInterrupt:
        print(f"\n{dati.datetime.now()} server interrupted!")  # Control + C.

print(f"{dati.datetime.now()} server ends")
```

```
ricard@HP-ProBook:~$ date
vie 29 nov 2024 18:29:48 CET
ricard@HP-ProBook:~$ python3 server_http.py
2024-11-29 18:29:53.818574 server starts
2024-11-29 18:29:53.818629 server is ready
2024-11-29 18:30:13.809242 server has established a connection
2024-11-29 18:30:13.809346 request: b'GET / HTTP/1.1\r\nHost:
localhost:8000\r\nUser-Agent: curl/8.5.0\r\nAccept: */*\r\n\r\n'
2024-11-29 18:30:13.809486 response: b'HTTP/1.1 200 OK\r\nContent-Type:
text/html\r\n\r\nHello World!'
2024-11-29 18:30:13.809660 server is ready
^C
2024-11-29 18:30:25.041362 server interrupted!
2024-11-29 18:30:25.041480 server ends
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
vie 29 nov 2024 18:30:04 CET
ricard@HP-ProBook:~$ curl localhost:8000; echo
Hello World!
ricard@HP-ProBook:~$ date
vie 29 nov 2024 18:30:17 CET
ricard@HP-ProBook:~$
```

https://en.wikipedia.org/wiki/CURL

```
ricard@HP-ProBook:~$ python3 server_http.py
2024-11-29 18:34:49.678158 server starts
2024-11-29 18:34:49.678237 server is ready
2024-11-29 18:35:04.567048 server has established a connection
2024-11-29 18:35:04.573625 request: b'GET / HTTP/1.1\r\nHost:
localhost:8000\r\nUser-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
rv:133.0) Gecko/20100101 Firefox/133.0\r\nAccept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept
-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3\r\nAccept-Encoding: gzip,
deflate, br, zstd\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests:
1\r\nSec-Fetch-Dest: document\r\nSec-Fetch-Mode:
navigate\r\nSec-Fetch-Site: none\r\nSec-Fetch-User: ?1\r\nPriority: u=0,
i\r\n\r\n'
2024-11-29 18:35:04.573759 response: b'HTTP/1.1 200 OK\r\nContent-Type:
text/html\r\n\r\nHello World!'
2024-11-29 18:35:04.573832 server is ready
2024-11-29 18:35:04.618622 server has established a connection
2024-11-29 18:35:04.618671 request: b'GET /favicon.ico HTTP/1.1\r\nHost:
localhost:8000\r\nUser-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
rv:133.0) Gecko/20100101 Firefox/133.0\r\nAccept:
image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5\r\n
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3\r\nAccept-Encoding:
gzip, deflate, br, zstd\r\nConnection: keep-alive\r\nReferer:
http://localhost:8000/\r\nSec-Fetch-Dest: image\r\nSec-Fetch-Mode:
no-cors\r\nSec-Fetch-Site: same-origin\r\nPriority: u=6\r\n\r\n'
2024-11-29 18:35:04.618717 response: b'HTTP/1.1 200 OK\r\nContent-Type:
text/html\r\n\r\nHello World!'
2024-11-29 18:35:04.618747 server is ready
```



https://en.wikipedia.org/wiki/Firefox
https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
https://en.wikipedia.org/wiki/Favicon

A CLIENT SENDS NUMBERS TO A SERVER

server.py

```
import datetime as dati
import os
import socket
import time

def show_text(t):
    print(f"{dati.datetime.now()} {t}")

process_id = os.getpid()
show_text(f"server (PID {process_id}) starts")

# Standard loopback interface address (localhost).
host = "127.0.0.1"
# Port to listen on (range 49152-65535 contains dynamic or private ports).
port = 65432

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

s.bind((host, port))
s.listen(1)  # Enable a server to accept one connection.
show_text(f"server listens for one connection")

conn, addr = s.accept()
show_text(f"server accepts connection with client {addr}")

while True:
    message = conn.recv(2).decode()  # Message is two characters.
    show_text(f"server has received {message} from client")
    if message == "00":
       break

time.sleep(1)

conn.close()
show_text(f"server closes connection with client {addr[0]}:{addr[1]}")

s.close()
show_text(f"server closes socket")

show_text(f"server (PID {process_id}) ends")
```

client.py

```
import datetime as dati
import os
import random
import socket
import time

def show_text(t):
    print(f"{dati.datetime.now()} {t}")

process_id = os.getpid()
show_text(f"client (PID {process_id}) starts")

host, port = "127.0.0.1", 65432

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

s.connect((host, port))
show_text(f"client connected ...")

while True:
    # Message is two characters.
    message = str(random.randint(0, 9)).zfill(2)

    s.send(message.encode())
    show_text(f"client has sent {message} to server")
    if message == "00":
        break
    time.sleep(1)

s.close()  # Server will have already closed the connection.
show_text(f"client (PID {process_id}) ends")
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 09:43:15 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 09:43:16.042506 server (PID 8664) starts
2024-12-02 09:43:16.042723 server listens for one connection
2024-12-02 09:43:22.847727 server accepts connection with client ('127.0.0.1', 48392)
2024-12-02 09:43:22.847849 server has received 05 from client
2024-12-02 09:43:23.848210 server has received 07 from client
2024-12-02 09:43:24.848962 server has received 02 from client
2024-12-02 09:43:25.849430 server has received 01 from client
2024-12-02 09:43:26.849920 server has received 03 from client
2024-12-02 09:43:27.850746 server has received 08 from client
2024-12-02 09:43:28.851451 server has received 02 from client
2024-12-02 09:43:29.852158 server has received 00 from client
2024-12-02 09:43:30.852994 server closes connection with client 127.0.0.1:48392
2024-12-02 09:43:30.853144 server closes socket
2024-12-02 09:43:30.853200 server (PID 8664) ends
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 09:43:19 CET
ricard@HP-ProBook:~$ python3 client.py
2024-12-02 09:43:22.847141 client (PID 8667) starts
2024-12-02 09:43:22.847525 client connected ...
2024-12-02 09:43:22.847718 client has sent 05 to server
2024-12-02 09:43:23.848117 client has sent 07 to server
2024-12-02 09:43:24.848878 client has sent 02 to server
2024-12-02 09:43:25.849339 client has sent 01 to server
2024-12-02 09:43:26.849833 client has sent 03 to server
2024-12-02 09:43:27.850661 client has sent 08 to server
2024-12-02 09:43:28.851332 client has sent 02 to server
2024-12-02 09:43:29.852072 client has sent 00 to server
2024-12-02 09:43:29.852300 client (PID 8667) ends
ricard@HP-ProBook:~$
```

TWO CLIENTS AND A SERVER

server.py

```
import datetime as dati
import os
import socket
import time

def show text(t):
    print(f"{dati.datetime.now()} {t}")

process id = os.getpid()
show_text(f"server (PID {process_id}) starts")

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

s.bind(("127.0.0.1", 65432))
s.listen(2)  # Enable a server to accept two connections.
show_text(f"server listens for connections")

conn1, addr1 = s.accept()
show text(f"server accepts connection with client {addr1}")
conn2, addr2 = s.accept()
show_text(f"server accepts connection with client {addr2}")

# Server notifies clients that they can send messages.
conn1.send("OK".encode())
show text(f"server sends OK to client {addr1}")
conn2.send("OK".encode())
show_text(f"server sends OK to client {addr2}")

while True:
    message1 = conn1.recv(3).decode()  # Three character message sent by client.
    if message1 != "":
        show_text(f"server has received {message1} from client {addr1}")

    message2 = conn2.recv(3).decode() # Three character message sent by client.
    if message2 != "":
        show_text(f"server has received {message2} from client {addr2}")

    if message1 == "" and message2 == "":
        break

time.sleep(1)

conn1.close()
conn2.close()
show_text(f"server closes connections")

s.close()
show_text(f"server closes socket")

show_text(f"server (PID {process_id}) ends")
```

client.py

```
import datetime as dati
import os
import random
import socket
import sys
import time

def show_text(t):
    print(f"{dati.datetime.now()} {t}")

# Must be a single character.
arg = sys.argv[1]

process id = os.getpid()
show_text(f"client (PID {process_id}) starts")

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

s.connect(("127.0.0.1", 65432))
show_text(f"client connected ...")

# Client waits for server to tell it that it can start sending.
ms = s.recv(2).decode()
show_text(f"client has received {ms}")

n = random.randint(5, 10)
for i in range(n):
    mc = f"{arg}{str(n - i).zfill(2)}"  # Three character message.
    s.send(mc.encode())
    show text(f"client has sent {mc} to server")
    time.sleep(1)

s.close()
show_text(f"client (PID {process_id}) ends")
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 09:55:51 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 09:55:54.133782 server (PID 9066) starts
2024-12-02 09:55:54.133984 server listens for connections
2024-12-02 09:56:34.769118 server accepts connection with client ('127.0.0.1', 43296)
2024-12-02 09:56:43.517076 server accepts connection with client ('127.0.0.1', 59134)
2024-12-02 09:56:43.517313 server sends OK to client ('127.0.0.1', 43296)
2024-12-02 09:56:43.517495 server sends OK to client ('127.0.0.1', 59134)
2024-12-02 09:56:43.517763 server has received A05 from client ('127.0.0.1', 43296)
2024-12-02 09:56:43.517851 server has received B07 from client ('127.0.0.1', 59134)
2024-12-02 09:56:44.518162 server has received A04 from client ('127.0.0.1', 43296)
2024-12-02 09:56:44.518286 server has received B06 from client ('127.0.0.1', 59134)
2024-12-02 09:56:45.518877 server has received A03 from client ('127.0.0.1', 43296)
2024-12-02 09:56:45.518989 server has received B05 from client ('127.0.0.1', 59134)
2024-12-02 09:56:46.519740 server has received A02 from client ('127.0.0.1', 43296)
2024-12-02 09:56:46.519880 server has received B04 from client ('127.0.0.1', 59134)
2024-12-02 09:56:47.520560 server has received A01 from client ('127.0.0.1', 43296)
2024-12-02 09:56:47.520685 server has received B03 from client ('127.0.0.1', 59134)
2024-12-02 09:56:48.521300 server has received B02 from client ('127.0.0.1', 59134)
2024-12-02 09:56:49.521696 server has received B01 from client ('127.0.0.1', 59134)
2024-12-02 09:56:51.522916 server closes connections
2024-12-02 09:56:51.523054 server closes socket
2024-12-02 09:56:51.523094 server (PID 9066) ends
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 09:56:30 CET
ricard@HP-ProBook:~$ python3 client.py A
2024-12-02 09:56:34.768541 client (PID 9082) starts
2024-12-02 09:56:34.768910 client connected ...
2024-12-02 09:56:43.517427 client has received OK
2024-12-02 09:56:43.517686 client has sent A05 to server
2024-12-02 09:56:44.518046 client has sent A04 to server
2024-12-02 09:56:45.518769 client has sent A03 to server
2024-12-02 09:56:46.519610 client has sent A02 to server
2024-12-02 09:56:47.520465 client has sent A01 to server
2024-12-02 09:56:48.521174 client (PID 9082) ends
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 09:56:36 CET
ricard@HP-ProBook:~$ python3 client.py B
2024-12-02 09:56:43.516472 client (PID 9086) starts
2024-12-02 09:56:43.516895 client connected ...
2024-12-02 09:56:43.517541 client has received OK
2024-12-02 09:56:43.517797 client has sent B07 to server
2024-12-02 09:56:44.518116 client has sent B06 to server
2024-12-02 09:56:45.518769 client has sent B05 to server
2024-12-02 09:56:46.519610 client has sent B04 to server
2024-12-02 09:56:47.520450 client has sent B03 to server
2024-12-02 09:56:48.521161 client has sent B02 to server
2024-12-02 09:56:49.521610 client has sent B01 to server
2024-12-02 09:56:50.522468 client (PID 9086) ends
ricard@HP-ProBook:~$
```

SOME CLIENTS AND A SERVER

server.py

```
import datetime as dati
import socket
import threading

def show text(t):
    print(f"{dati.datetime.now()} {t}")

def reverse_string(s):
    return s[::-1]

def handle client(co, ad):
    show_text(f"new connection on {ad}")

    connected = True
    while connected:
        msg = co.recv(1024).decode("utf-8")
        if msg == "DISCONNECT":
            show_text(f"disconnected {ad}")
            connected = False
        else:
            msg = reverse string(msg)
            show text(f"send to {ad} {msg}")
            co.send(msg.encode("utf-8"))

    co.close()

def main():
    show_text("starts")
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(('localhost', 0))
    show text(f"listens for connections on {server.getsockname()}")
    server.listen()

    while True:
        try:
            conn, addr = server.accept()
            th = threading.Thread(target=handle_client, args=(conn, addr))
            th.start()
            show_text(f"active connections {threading.active_count() - 1}")
        except:
            show text("\nshutting down the server")
            break

    server.close()

if   name _ == "__main__":
    main()
```

client.py

```
import datetime as dati
import socket
import sys

def show text(t):
    print(f"{dati.datetime.now()} {t}")

def main():
    ip, port = sys.argv[1], int(sys.argv[2])
    client name = sys.argv[3]
    show_text(f"{client_name} starts")
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((ip, port))
    show text(f"{client name} {client.getsockname()[0]}:{client.getsockname()[1]}")
    show_text(f"{client_name} connected to server at {ip}:{port}")

    connected = True
    while connected:
        msg = input("> ")
        client.send(msg.encode("utf-8"))
        if msg == "DISCONNECT":
            connected = False
        else:
            msg = client.recv(1024).decode("utf-8")
            show_text(f"server has sent: '{msg}'")

if __name__ == "__main__":
        main()
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:08 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 10:18:09.665510 starts
2024-12-02 10:18:09.666104 listens for connections on ('127.0.0.1', 43075)
```

```
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:08 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 10:18:09.665510 starts
2024-12-02 10:18:09.666104 listens for connections on ('127.0.0.1', 43075)
2024-12-02 10:18:23.773678 new connection on ('127.0.0.1', 40892)
2024-12-02 10:18:23.773839 active connections 1
2024-12-02 10:18:26.098167 send to ('127.0.0.1', 40892) olleH
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:13 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Alice
2024-12-02 10:18:23.772256 Alice starts
2024-12-02 10:18:23.772682 Alice 127.0.0.1:40892
2024-12-02 10:18:23.772733 Alice connected to server at 127.0.0.1:43075
> Hello
2024-12-02 10:18:26.098447 server has sent: 'olleH'
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:08 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 10:18:09.665510 starts
2024-12-02 10:18:09.666104 listens for connections on ('127.0.0.1', 43075)
2024-12-02 10:18:23.773678 new connection on ('127.0.0.1', 40892)
2024-12-02 10:18:23.773839 active connections 1
2024-12-02 10:18:26.098167 send to ('127.0.0.1', 40892) olleH
2024-12-02 10:21:24.532946 new connection on ('127.0.0.1', 40274)
2024-12-02 10:21:24.533079 active connections 2
2024-12-02 10:21:32.382411 new connection on ('127.0.0.1', 52734)
2024-12-02 10:21:32.382589 active connections 3
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:13 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Alice
2024-12-02 10:18:23.772256 Alice starts
2024-12-02 10:18:23.772682 Alice 127.0.0.1:40892
2024-12-02 10:18:23.772733 Alice connected to server at 127.0.0.1:43075
> Hello
2024-12-02 10:18:26.098447 server has sent: 'olleH'
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:18 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Bob
2024-12-02 10:21:24.531723 Bob starts
2024-12-02 10:21:24.532111 Bob 127.0.0.1:40274
2024-12-02 10:21:24.532162 Bob connected to server at 127.0.0.1:43075
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:26 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Eve
2024-12-02 10:21:32.381254 Eve starts
2024-12-02 10:21:32.381662 Eve 127.0.0.1:52734
2024-12-02 10:21:32.381714 Eve connected to server at 127.0.0.1:43075
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:08 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 10:18:09.665510 starts
2024-12-02 10:18:09.666104 listens for connections on ('127.0.0.1', 43075)
2024-12-02 10:18:23.773678 new connection on ('127.0.0.1', 40892)
2024-12-02 10:18:23.773839 active connections 1
2024-12-02 10:18:26.098167 send to ('127.0.0.1', 40892) olleH
2024-12-02 10:21:24.532946 new connection on ('127.0.0.1', 40274)
2024-12-02 10:21:24.533079 active connections 2
2024-12-02 10:21:32.382411 new connection on ('127.0.0.1', 52734)
2024-12-02 10:21:32.382589 active connections 3
2024-12-02 10:23:32.131202 send to ('127.0.0.1', 40892) ecilA m'I
2024-12-02 10:23:45.683050 send to ('127.0.0.1', 40274) boB m'I ,olleH
2024-12-02 10:23:52.082395 send to ('127.0.0.1', 52734) evE m'I ,olleH
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:13 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Alice
2024-12-02 10:18:23.772256 Alice starts
2024-12-02 10:18:23.772682 Alice 127.0.0.1:40892
2024-12-02 10:18:23.772733 Alice connected to server at 127.0.0.1:43075
> Hello
2024-12-02 10:18:26.098447 server has sent: 'olleH'
> I'm Alice
2024-12-02 10:23:32.131585 server has sent: 'ecilA m'I'
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:18 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Bob
2024-12-02 10:21:24.531723 Bob starts
2024-12-02 10:21:24.532111 Bob 127.0.0.1:40274
2024-12-02 10:21:24.532162 Bob connected to server at 127.0.0.1:43075
> Hello, I'm Bob
2024-12-02 10:23:45.683303 server has sent: 'boB m'I ,olleH'
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:26 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Eve
2024-12-02 10:21:32.381254 Eve starts
2024-12-02 10:21:32.381662 Eve 127.0.0.1:52734
2024-12-02 10:21:32.381714 Eve connected to server at 127.0.0.1:43075
> Hello, I'm Eve
2024-12-02 10:23:52.082682 server has sent: 'evE m'I ,olleH'
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:08 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 10:18:09.665510 starts
2024-12-02 10:18:09.666104 listens for connections on ('127.0.0.1', 43075)
2024-12-02 10:18:23.773678 new connection on ('127.0.0.1', 40892)
2024-12-02 10:18:23.773839 active connections 1
2024-12-02 10:18:26.098167 send to ('127.0.0.1', 40892) olleH
2024-12-02 10:21:24.532946 new connection on ('127.0.0.1', 40274)
2024-12-02 10:21:24.533079 active connections 2
2024-12-02 10:21:32.382411 new connection on ('127.0.0.1', 52734)
2024-12-02 10:21:32.382589 active connections 3
2024-12-02 10:23:32.131202 send to ('127.0.0.1', 40892) ecilA m'I
2024-12-02 10:23:45.683050 send to ('127.0.0.1', 40274) boB m'I ,olleH
2024-12-02 10:23:52.082395 send to ('127.0.0.1', 52734) evE m'I ,olleH
2024-12-02 10:25:38.618687 disconnected ('127.0.0.1', 40274)
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:13 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Alice
2024-12-02 10:18:23.772256 Alice starts
2024-12-02 10:18:23.772682 Alice 127.0.0.1:40892
2024-12-02 10:18:23.772733 Alice connected to server at 127.0.0.1:43075
> Hello
2024-12-02 10:18:26.098447 server has sent: 'olleH'
> I'm Alice
2024-12-02 10:23:32.131585 server has sent: 'ecilA m'I'
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:18 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Bob
2024-12-02 10:21:24.531723 Bob starts
2024-12-02 10:21:24.532111 Bob 127.0.0.1:40274
2024-12-02 10:21:24.532162 Bob connected to server at 127.0.0.1:43075
> Hello, I'm Bob
2024-12-02 10:23:45.683303 server has sent: 'boB m'I ,olleH'
> DISCONNECT
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:26 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Eve
2024-12-02 10:21:32.381254 Eve starts
2024-12-02 10:21:32.381662 Eve 127.0.0.1:52734
2024-12-02 10:21:32.381714 Eve connected to server at 127.0.0.1:43075
> Hello, I'm Eve
2024-12-02 10:23:52.082682 server has sent: 'evE m'I ,olleH'
>
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:08 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 10:18:09.665510 starts
2024-12-02 10:18:09.666104 listens for connections on ('127.0.0.1', 43075)
2024-12-02 10:18:23.773678 new connection on ('127.0.0.1', 40892)
2024-12-02 10:18:23.773839 active connections 1
2024-12-02 10:18:26.098167 send to ('127.0.0.1', 40892) olleH
2024-12-02 10:21:24.532946 new connection on ('127.0.0.1', 40274)
2024-12-02 10:21:24.533079 active connections 2
2024-12-02 10:21:32.382411 new connection on ('127.0.0.1', 52734)
2024-12-02 10:21:32.382589 active connections 3
2024-12-02 10:23:32.131202 send to ('127.0.0.1', 40892) ecilA m'I
2024-12-02 10:23:45.683050 send to ('127.0.0.1', 40274) boB m'I ,olleH
2024-12-02 10:23:52.082395 send to ('127.0.0.1', 52734) evE m'I ,olleH
2024-12-02 10:25:38.618687 disconnected ('127.0.0.1', 40274)
2024-12-02 10:26:46.154255 disconnected ('127.0.0.1', 40892)
2024-12-02 10:26:53.299046 disconnected ('127.0.0.1', 52734)
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:13 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Alice
2024-12-02 10:18:23.772256 Alice starts
2024-12-02 10:18:23.772682 Alice 127.0.0.1:40892
2024-12-02 10:18:23.772733 Alice connected to server at 127.0.0.1:43075
> Hello
2024-12-02 10:18:26.098447 server has sent: 'olleH'
> I'm Alice
2024-12-02 10:23:32.131585 server has sent: 'ecilA m'I'
> DISCONNECT
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:18 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Bob
2024-12-02 10:21:24.531723 Bob starts
2024-12-02 10:21:24.532111 Bob 127.0.0.1:40274
2024-12-02 10:21:24.532162 Bob connected to server at 127.0.0.1:43075
> Hello, I'm Bob
2024-12-02 10:23:45.683303 server has sent: 'boB m'I ,olleH'
> DISCONNECT
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:26 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Eve
2024-12-02 10:21:32.381254 Eve starts
2024-12-02 10:21:32.381662 Eve 127.0.0.1:52734
2024-12-02 10:21:32.381714 Eve connected to server at 127.0.0.1:43075
> Hello, I'm Eve
2024-12-02 10:23:52.082682 server has sent: 'evE m'I ,olleH'
> DISCONNECT
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:08 CET
ricard@HP-ProBook:~$ python3 server.py
2024-12-02 10:18:09.665510 starts
2024-12-02 10:18:09.666104 listens for connections on ('127.0.0.1', 43075)
2024-12-02 10:18:23.773678 new connection on ('127.0.0.1', 40892)
2024-12-02 10:18:23.773839 active connections 1
2024-12-02 10:18:26.098167 send to ('127.0.0.1', 40892) olleH
2024-12-02 10:21:24.532946 new connection on ('127.0.0.1', 40274)
2024-12-02 10:21:24.533079 active connections 2
2024-12-02 10:21:32.382411 new connection on ('127.0.0.1', 52734)
2024-12-02 10:21:32.382589 active connections 3
2024-12-02 10:23:32.131202 send to ('127.0.0.1', 40892) ecilA m'I
2024-12-02 10:23:45.683050 send to ('127.0.0.1', 40274) boB m'I ,olleH
2024-12-02 10:23:52.082395 send to ('127.0.0.1', 52734) evE m'I ,olleH
2024-12-02 10:25:38.618687 disconnected ('127.0.0.1', 40274)
2024-12-02 10:26:46.154255 disconnected ('127.0.0.1', 40892)
2024-12-02 10:26:53.299046 disconnected ('127.0.0.1', 52734)
^C2024-12-02 10:27:54.658781
shutting down the server
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:18:13 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Alice
2024-12-02 10:18:23.772256 Alice starts
2024-12-02 10:18:23.772682 Alice 127.0.0.1:40892
2024-12-02 10:18:23.772733 Alice connected to server at 127.0.0.1:43075
> Hello
2024-12-02 10:18:26.098447 server has sent: 'olleH'
> I'm Alice
2024-12-02 10:23:32.131585 server has sent: 'ecilA m'I'
> DISCONNECT
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:18 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Bob
2024-12-02 10:21:24.531723 Bob starts
2024-12-02 10:21:24.532111 Bob 127.0.0.1:40274
2024-12-02 10:21:24.532162 Bob connected to server at 127.0.0.1:43075
> Hello, I'm Bob
2024-12-02 10:23:45.683303 server has sent: 'boB m'I ,olleH'
> DISCONNECT
ricard@HP-ProBook:~$
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 10:21:26 CET
ricard@HP-ProBook:~$ python3 client.py 127.0.0.1 43075 Eve
2024-12-02 10:21:32.381254 Eve starts
2024-12-02 10:21:32.381662 Eve 127.0.0.1:52734
2024-12-02 10:21:32.381714 Eve connected to server at 127.0.0.1:43075
> Hello, I'm Eve
2024-12-02 10:23:52.082682 server has sent: 'evE m'I ,olleH'
> DISCONNECT
ricard@HP-ProBook:~$
```

TWO CLIENTS CHATTING THROUGH A SERVER

server.py

```
import datetime as dati
import socket
import threading

def show text(t):
    print(f"{dati.datetime.now()} {t}")

def broadcast(co, ci, m):
    for c in ci:
        if c != co:
            c.send(m)
            show_text(f"send {c.getpeername()} message {m.decode('utf-8')}")

def handle(co, cl):
    while True:
        try:
            message = co.recv(1024)
            broadcast(co, cl, message)
        except:
            index = clients.index(co)
            clients.remove(co)
            co.close()
            nickname = nicknames[index]
            broadcast(co, cl, f"{nickname} left".encode('utf-8'))
            nicknames.remove(nickname)
            break

def main():
    clients = []
    nicknames = []

    server = socket.socket(socket.AF INET, socket.SOCK_STREAM)
    server.bind(('192.168.1.45', 0))
    server.listen()

    show_text(f"starts {server.getsockname()[0]}:{server.getsockname()[1]}")

    while True:
        conn, addr = server.accept()
        show_text(f"connected with {addr}")

        conn.send('NICK'.encode('utf-8'))
        nickname = conn.recv(1024).decode('utf-8')
        nicknames.append(nickname)
        clients.append(conn)

        show text(f"new client: '{nickname}'")
        broadcast(conn, clients, f"{nickname} joined".encode('utf-8'))
        conn.send('connected to server!'.encode('utf-8'))

        thread = threading.Thread(target=handle, args=(conn, clients))
        thread.start()

    server.close()

if   name   == "__main__":
    main()
```

client.py

```
import datetime as dati
import socket
import sys
import threading

def show text(t):
    print(f"{dati.datetime.now()} {t}")

def receive message(cl, nina):
    while True:
        try:
            message = cl.recv(1024).decode('utf-8')
            if message == 'NICK':
                cl.send(nina.encode('utf-8'))
            else:
                print(message)
        except:
            show text("an error occured!")
            cl.close()
            break

def send message(cl, nina):
    while True:
        message = '{}: {}'.format(nina, input(''))
        cl.send(message.encode('utf-8'))

def main():
    show text("starts")
    nickname = input("choose your nickname: ")

    client = socket.socket(socket.AF INET, socket.SOCK_STREAM)
    client.connect((sys.argv[1], int(sys.argv[2])))

    show_text(f"{nickname} {client.getsockname()[0]}:{client.getsockname()[1]}")

    receive thread = threading.Thread(target=receive_message, args=(client, nickname))
    receive_thread.start()

    write thread = threading.Thread(target=send_message, args=(client, nickname))
    write_thread.start()

if   name   == "__main__":
    main()
```

If testing on a single computer, in the server.py file, replace '192.168.1.45' with 'localhost'.

For didactic reasons, so that the source code is understandable, the programs do not have mechanisms for ending.

```
ricard@ThinkStation-P360-Tower:~$ date
lun 02 dic 2024 12:04:40 CET
ricard@ThinkStation-P360-Tower:~$ ip --brief a | grep eno1 | awk '{print $3}'
192.168.1.45/24
ricard@ThinkStation-P360-Tower:~$ cat /etc/os-release | grep PRETTY
PRETTY_NAME="Ubuntu 22.04.5 LTS"
ricard@ThinkStation-P360-Tower:~$ python3 server.py
2024-12-02 12:04:54.711314 starts 192.168.1.45:41051
2024-12-02 12:05:24.101542 connected with ('192.168.1.38', 54580)
2024-12-02 12:05:24.102605 new client: 'Alice'
2024-12-02 12:05:48.542053 connected with ('192.168.1.132', 48812)
2024-12-02 12:05:48.546346 new client: 'Bob'
2024-12-02 12:05:48.546412 send ('192.168.1.38', 54580) message Bob joined
2024-12-02 12:05:58.221483 send ('192.168.1.132', 48812) message Alice: Hello Bob!
2024-12-02 12:06:05.818250 send ('192.168.1.38', 54580) message Bob: Hello Alice!
2024-12-02 12:06:14.685737 send ('192.168.1.132', 48812) message Alice: How are you?
2024-12-02 12:06:23.125522 send ('192.168.1.38', 54580) message Bob: Fine.
```

```
ricard@HP-ProBook:~$ date
lun 02 dic 2024 12:05:04 CET
ricard@HP-ProBook:~$ ip --brief a | grep enp1s0 | awk '{print $3}'
192.168.1.38/24
ricard@HP-ProBook:~$ cat /etc/os-release | grep PRETTY
PRETTY_NAME="Ubuntu 24.04.1 LTS"
ricard@HP-ProBook:~$ python3 client.py 192.168.1.45 41051
2024-12-02 12:05:20.639131 starts
choose your nickname: Alice
2024-12-02 12:05:24.099458 Alice 192.168.1.38:54580
connected to server!
Bob joined
Hello Bob!
Bob: Hello Alice!
How are you?
Bob: Fine.
```

```
ricard@Lenovo-V510-15IKB:~$ date
lun 02 dic 2024 12:05:28 CET
ricard@Lenovo-V510-15IKB:~$ ip --brief a | grep wlp3s0 | awk '{print $3}'
192.168.1.132/24
ricard@Lenovo-V510-15IKB:~$ cat /etc/os-release | grep PRETTY
PRETTY_NAME="Ubuntu 24.04.1 LTS"
ricard@Lenovo-V510-15IKB:~$ python3 client.py 192.168.1.45 41051
2024-12-02 12:05:46.138383 starts
choose your nickname: Bob
2024-12-02 12:05:48.539064 Bob 192.168.1.132:48812
connected to server!
Alice: Hello Bob!
Hello Alice!
Alice: How are you?
Fine.
```