

# Echoserver

Echo server and client using python socket

## AIM:

To develop a simple webserver to serve html programming pages.

## DESIGN STEPS:

### Step 1:

Design of echo server and client using python socket

### Step 2:

Implementation using Python code

### Step 3:

Testing the server and client

## PROGRAM:

DEVELOPED BY: HARSHAVARDHAN

REGISTER NO: 212222240114

## SERVER SIDE

```
import socket
HOST = "127.0.0.1" # The server's hostname or IP address
PORT = 65432 # The port used by the server
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b"Hello, world")
    data = s.recv(1024)
    print(f"Received {data!r}")
```



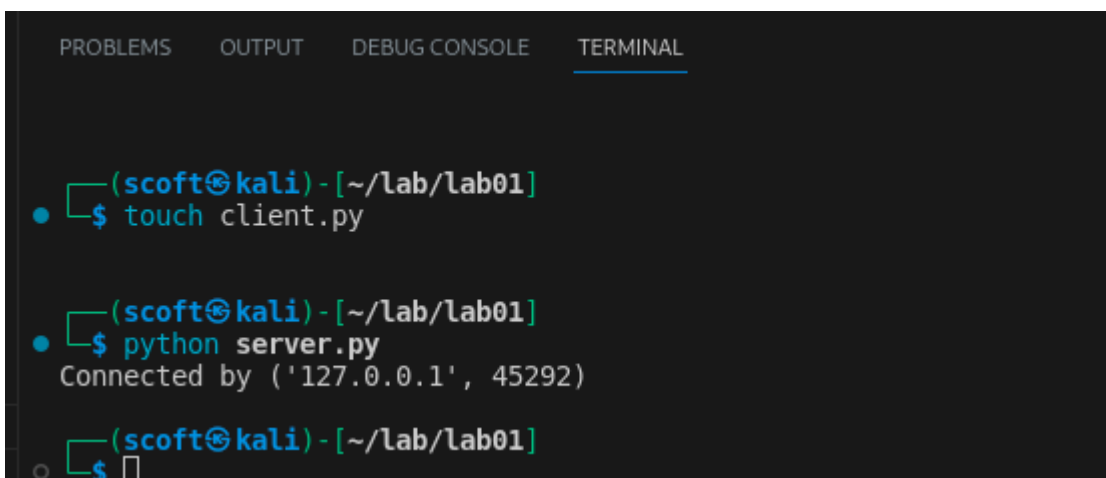
## CLIENT SIDE



```
import socket
HOST = "127.0.0.1" # Standard loopback interface address (localhost)
PORT = 65432 # Port to listen on (non-privileged ports are > 1023)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    try:
        s.bind((HOST, PORT))
    except Exception as e:
        print(f"Error binding to {HOST}:{PORT}: {e}")
        exit()
    s.listen()
    print(f"Listening on {HOST}:{PORT}...")
    try:
        conn, addr = s.accept()
    except Exception as e:
        print(f"Error accepting connection: {e}")
        exit()
    with conn:
        print(f"Connected by {addr}")
        while True:
            try:
                data = conn.recv(1024)
                if not data:
                    break
                conn.sendall(data)
            except Exception as e:
                print(f"Error receiving/sending data: {e}")
                exit()
```

## OUTPUT:

### SERVER SIDE



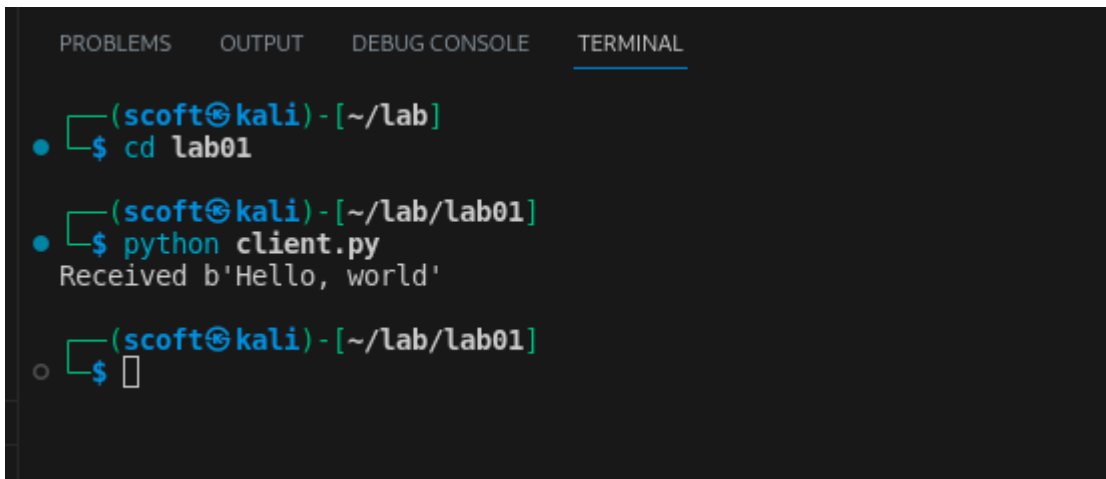
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(scoft@kali) - [~/lab/lab01]
$ touch client.py

(scoft@kali) - [~/lab/lab01]
$ python server.py
Connected by ('127.0.0.1', 45292)

(scoft@kali) - [~/lab/lab01]
$
```

### CLIENT SIDE



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there are four tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is selected and underlined. The terminal content shows three lines of command execution:

- Line 1: A prompt `(scoft@kali) - [~/lab]` followed by the command `$ cd lab01`.
- Line 2: A prompt `(scoft@kali) - [~/lab/lab01]` followed by the command `$ python client.py`. Below this command, the output `Received b'Hello, world'` is displayed.
- Line 3: A prompt `(scoft@kali) - [~/lab/lab01]` followed by a command prompt `$` and a cursor.

## RESULT:

The program is executed successfully