

chat.html

Create a file called `chat.html` and open it in a browser.

Run the program after each section by refreshing your browser with the `F5` key.

See extra debug/errors with `F12` .

base

```
<!DOCTYPE html>
<html>
<head>
  <title>Chat Client</title>
  <meta charset="UTF-8">
</head>
<body>
  <h1>Chat Client</h1>
<script>

</script></body></html>
```

connect

Replace `localhost` with the servers websocket url address. (Starts with `wss://` or `ws://`).

```
<script>

+const address = "ws://localhost:9800/test1.ws"
+let socket = new WebSocket(address)
+
</script></body></html>
```

Look at the server to see if your IP address connected.

send_one

```
let socket = new WebSocket(address)

+function open() {
+  socket.send("Hello I am JAVASCRIPT"+"\\n")
+}
+socket.addEventListener("open", open)
+
</script></body></html>
```

Look at the server to see if the server received your message.

send

```
socket.addEventListener("open", open)
+while (true) { socket.send(prompt()+"\\n") }
```

Type a message and press enter. See if the server got your message.

recv

```

while (true) { socket.send(prompt()+"\n") }
+
+function receive(msg) {
+  console.log("got: " + msg.data)
+}
+socket.addEventListener("message", receive)

```

See messages by showing devtools `F12` and viewing `console` .

gui

```

<h1>Chat Client</h1>
+ <input id="text_field" style="width:300px;" />

```

```

let socket      = new WebSocket(address)

-function open(){
  socket.send("Hello I am Bob"+"\\n")
-}
-socket.addEventListener("open", open)
-while (true) { socket.send(prompt()+"\\n") }
+let text_field = document.getElementById("text_field")
+text_field.addEventListener("keydown", textEventKeyDown, true)
+function textEventKeyDown(event) {
+  if (event.keyCode==13) {
+    socket.send(text_field.value+"\\n")
+    text_field.value = ""
+  }
+}

function receive(msg) {

```

gui_recv

```

<h1>Chat Client</h1>
+ <textarea id="text_area" style="width:300px;" rows="15" readonly></textarea><br/>
  <input id="text_field" style="width:300px;" />

```

```

+let text_area = document.getElementById("text_area")
function receive(msg) {
  console.log("got: " + msg.data)
+  text_area.value = text_area.value + msg.data
}
socket.addEventListener("message", receive)

```

gui_scroll

```

function receive(msg) {
  text_area.value = text_area.value + msg.data
+  text_area.scrollTop = text_area.scrollHeight
}

```

gui_username

```

if (event.keyCode==13) {
  socket.send(text_field.value+"\\n")
+  socket.send("Yourname: "+text_field.value+"\\n")
  text_field.value = ""
}

```

chat.py

Create a file `chat.py` .

Run the program after each addition in a terminal with `python chat.py` . If using IDLE run with `F5` .

Stop the program with `ctrl+c`

base

```
import socket, threading
```

connect

Replace `localhost` with the servers IP address example: 192.168.0.1

```
import socket, threading

+address = ("localhost", 9801)
+sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
+sock.connect(address)
```

Look at the server to see if your IP address connected.

send_one

```
sock.connect(address)

+sock.sendall('Hello I am PYTHON\n'.encode('utf-8'))
```

Look at the server to see if the server received your message.

send

```
sock.sendall('Hello I am PYTHON\n'.encode('utf-8'))
+while True:
+    sock.sendall(f'{input()}\n'.encode('utf-8'))
```

Type a message and press enter. See if the server got your message.

recv

```
sock.connect(address)

+def connection(sock):
+    while True:
+        data_recv = sock.recv(4098)
+        if not data_recv:
+            break
+        print(data_recv)
+        sock.close()
+connection(sock)
+
sock.sendall('Hello I am PYTHON\n'.encode('utf-8'))
```

Windows Only Issue: `ctrl+c` does not stop the program until another message is received. Press `ctrl+c` and wait patiently.

send_recv

```
~connection(sock)

+thread = threading.Thread(target=connection, args=(sock,))
+thread.daemon=True
+thread.start()
```

Windows Only Issue: Windows terminal blocks on `input()` - received messages are displayed garbled and only after sending a message.

gui

```
import socket, threading
+import tkinter
```

```
sock.connect(address)
+
+root = tkinter.Tk()
+root.title("Chat Client")
+input_box = tkinter.Entry(root)
+input_box.pack(fill=tkinter.BOTH)
+def handle_user_input(e):
+    sock.sendall(f'{input_box.get()}\n'.encode('utf-8'))
+    input_box.delete(0, tkinter.END)
+input_box.bind("<KeyRelease-Return", handle_user_input)
+input_box.focus_set()

def connection(sock):
```

```
~sock.sendall('Hello I am PYTHON\n'.encode('utf-8'))
~while True:
~    sock.sendall(f'{input()}\n'.encode('utf-8'))
+
+root.mainloop()
```

gui_recv

```
import tkinter
+import tkinter.scrolledtext
```

```
~print(data_recv)
+    output_box.insert(tkinter.END, data_recv)
```

```
root.title("Chat Client")
+output_box = tkinter.scrolledtext.ScrolledText(root, width=40, height=15)
+output_box.pack(fill=tkinter.BOTH, expand=1)
input_box = tkinter.Entry(root)
```

gui_scroll

```
output_box.insert(tkinter.END, data_recv)
+    output_box.yview(tkinter.END)
```

gui_username

```
~sock.sendall(f'{input_box.get()}\n'.encode('utf-8'))
+    sock.sendall((f'Username: {input_box.get()}\n').encode('utf-8'))
```