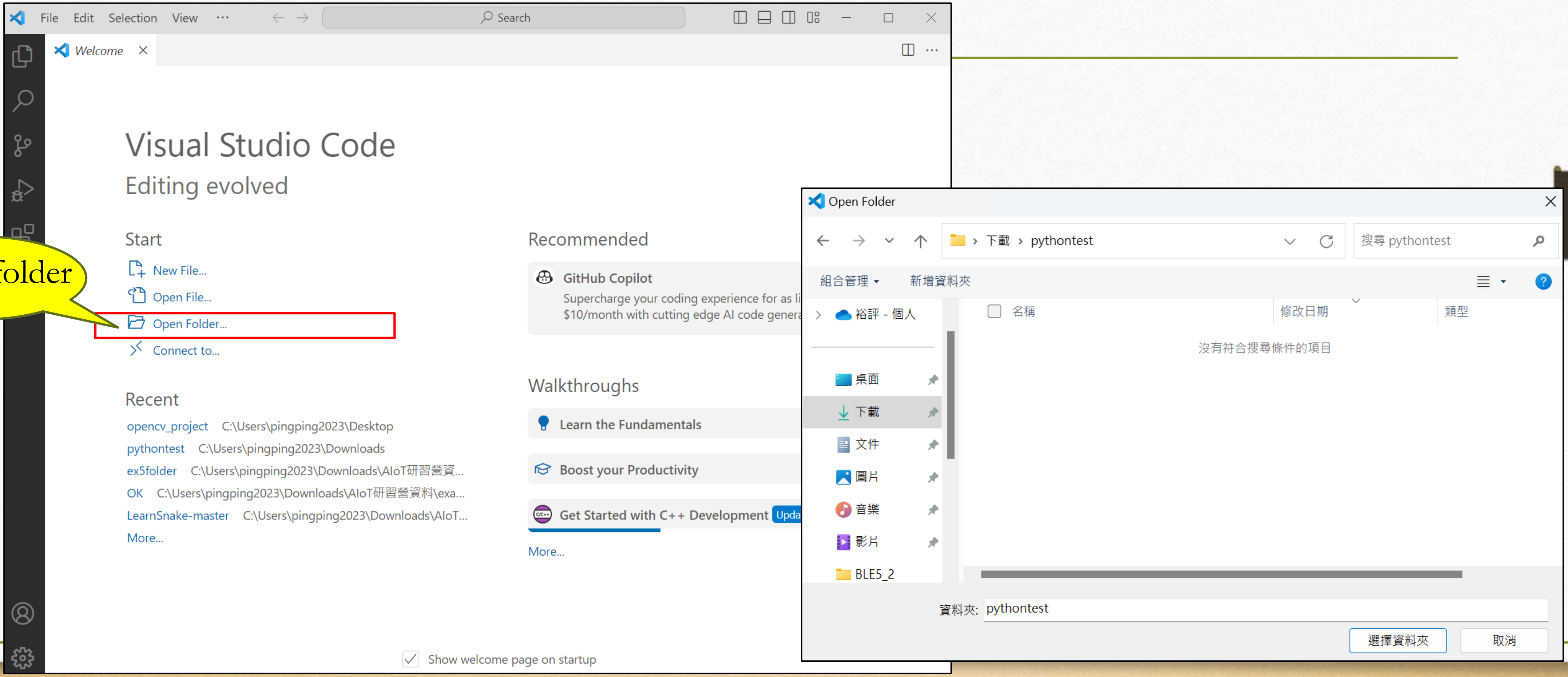
物聯網實務

(六)

廖裕評

Open Visual Studio Code



Open folder

Visual Studio Code

Editing evolved

Start

Recent

opencv_project C:\Users\pingping2023\Desktop

pythontest C:\Users\pingping2023\Downloads

ex5folder C:\Users\pingping2023\Downloads\AIoT研習營資...

OK C:\Users\pingping2023\Downloads\AIoT研習營資料\exa...

LearnSnake-master C:\Users\pingping2023\Downloads\AIoT...

More...

Recommended

GitHub Copilot

Supercharge your coding experience for as li

$10/month with cutting edge AI code genera

Walkthroughs

☑ Show welcome page on startup

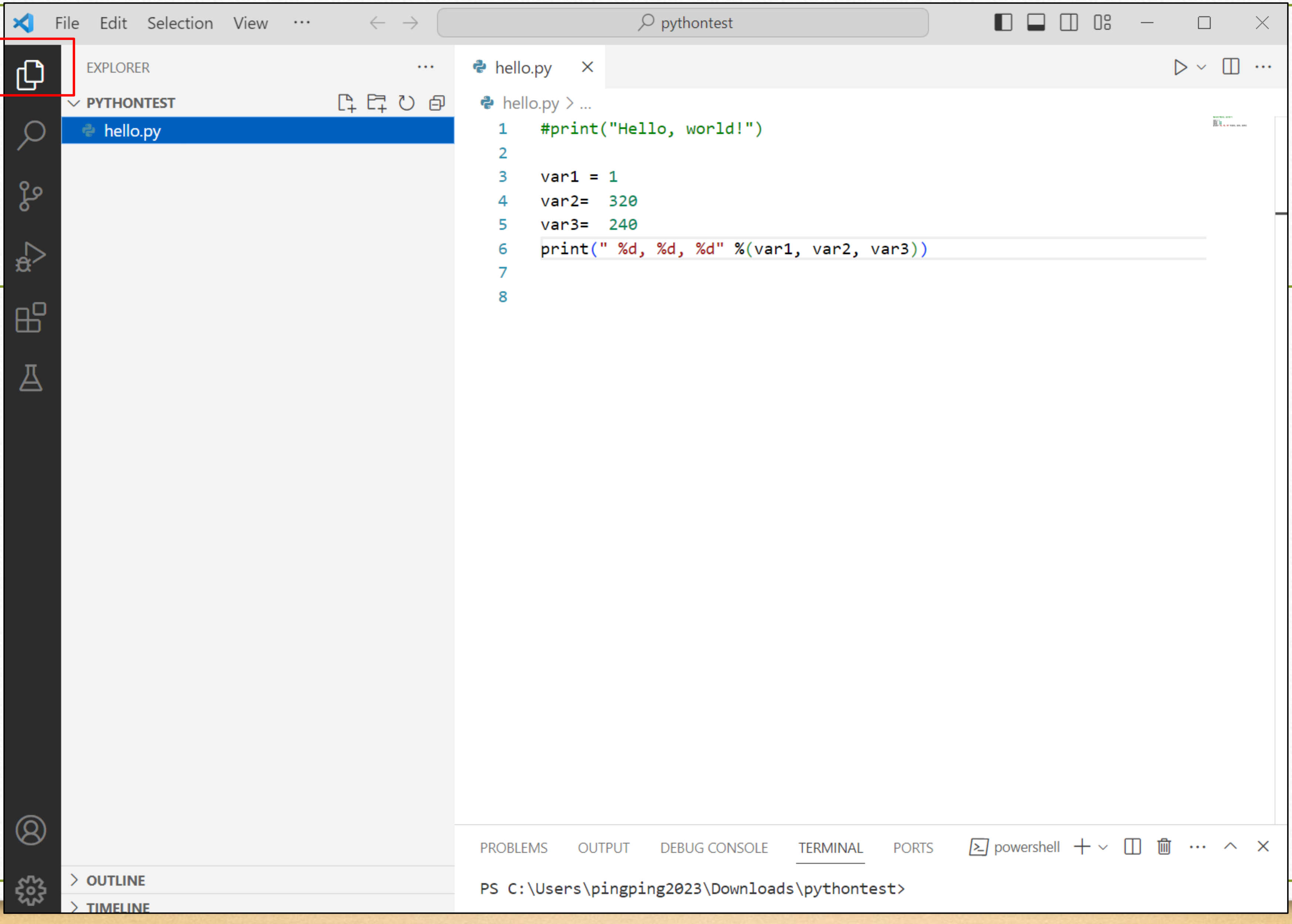
Open Folder

下載 › pythontest

搜尋 pythontest

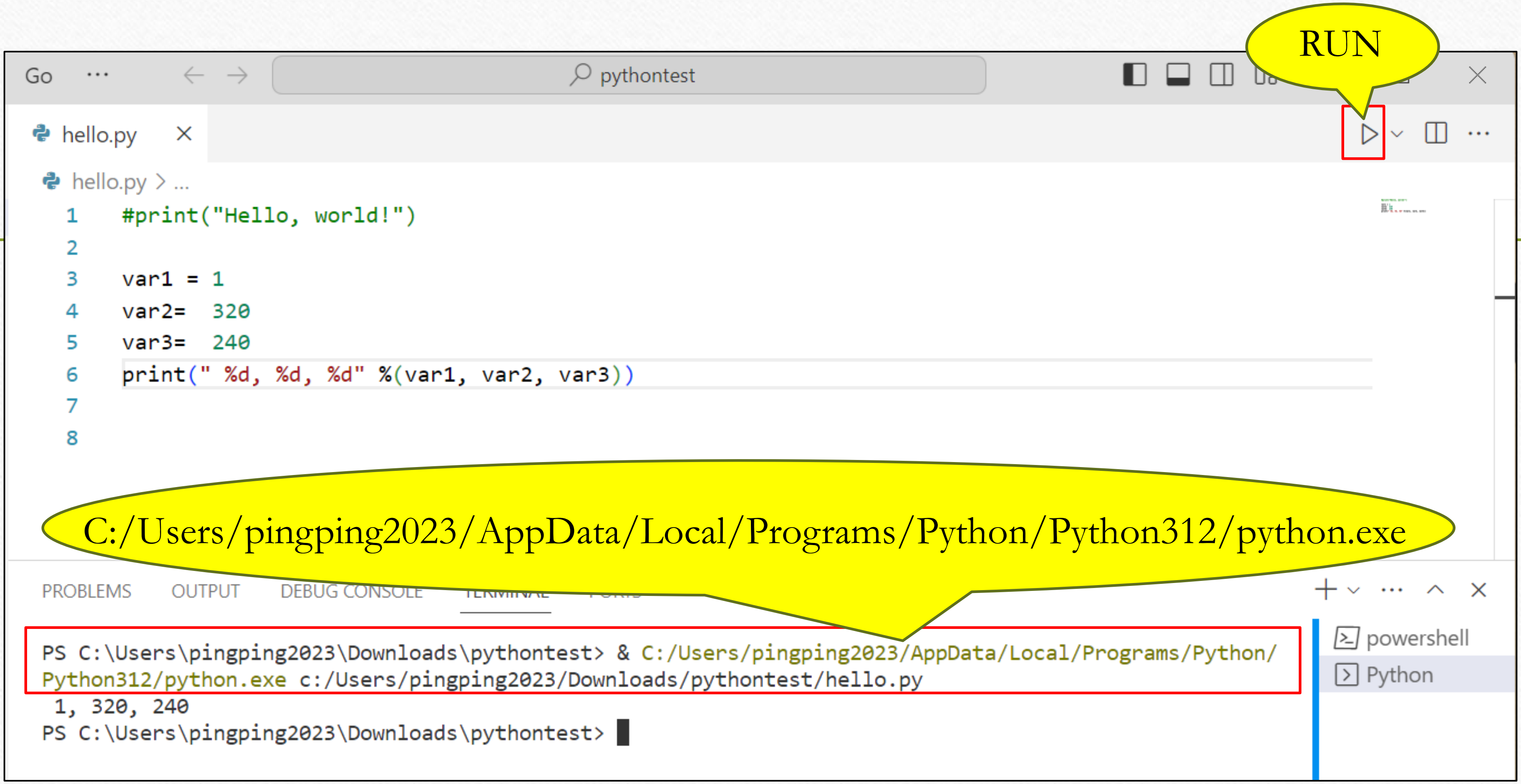
沒有符合搜尋條件的項目

資料夾: pythontest



```python
#print("Hello, world!")

var1 = 1
var2=  320
var3=  240
print(" %d, %d, %d" %(var1, var2, var3))
```

```
PS C:\Users\pingping2023\Downloads\pythontest>
```

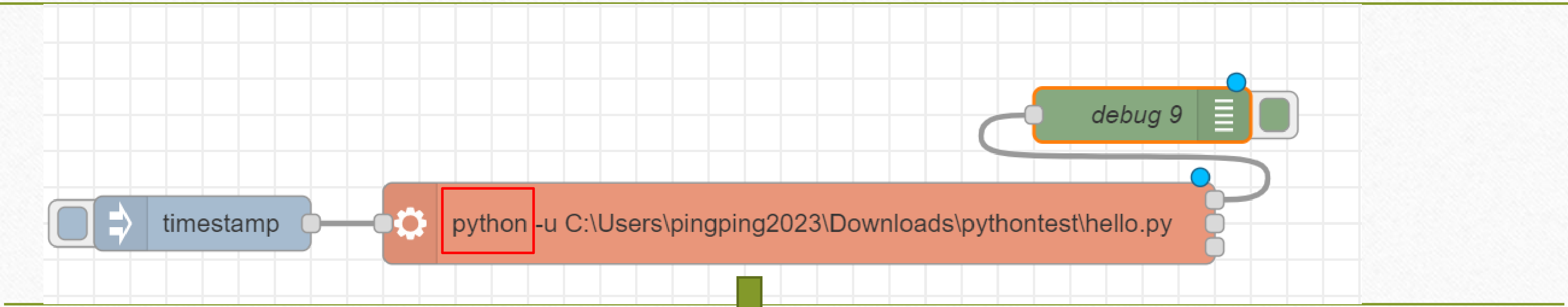


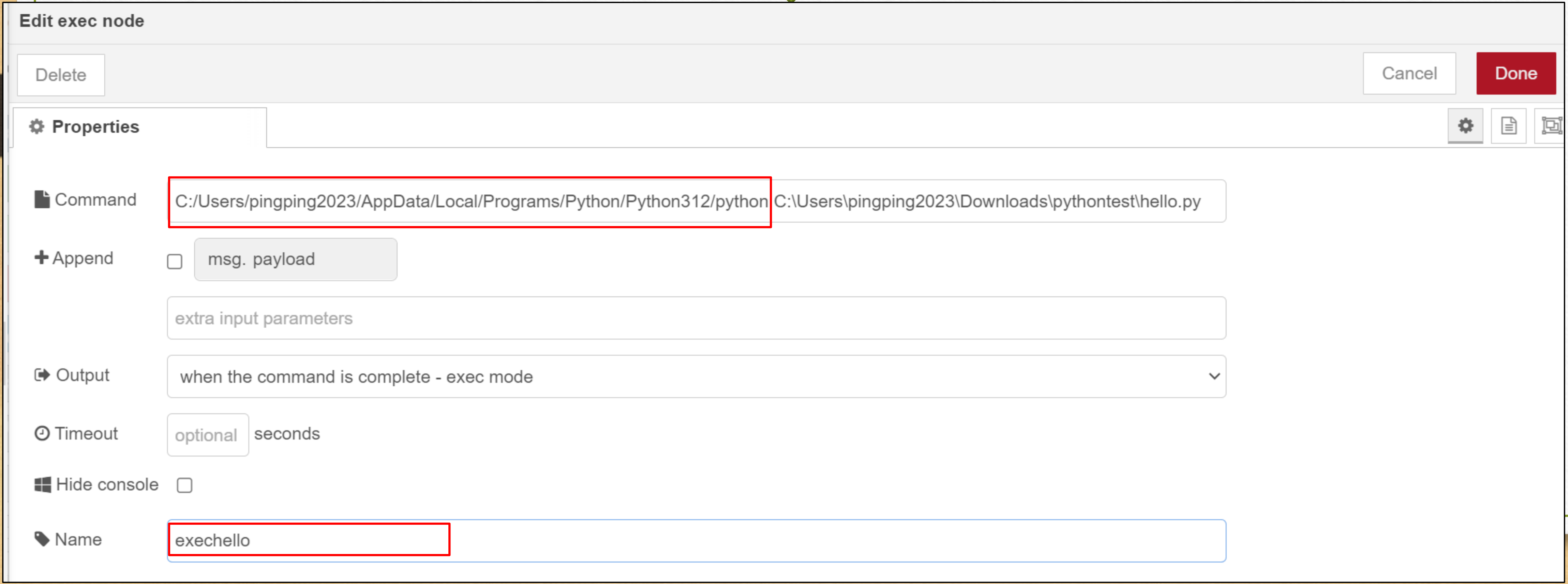
RUN

```python
#print("Hello, world!")

var1 = 1
var2=  320
var3=  240
print(" %d, %d, %d" %(var1, var2, var3))


```

C:/Users/pingping2023/AppData/Local/Programs/Python/Python312/python.exe

```
PS C:\Users\pingping2023\Downloads\pythontest> & C:/Users/pingping2023/AppData/Local/Programs/Python/
Python312/python.exe c:/Users/pingping2023/Downloads/pythontest/hello.py
 1, 320, 240
PS C:\Users\pingping2023\Downloads\pythontest> 
```


timestamp
python -u C:\Users\pingping2023\Downloads\pythontest\hello.py
debug 9


Edit exec node

Command: C:/Users/pingping2023/AppData/Local/Programs/Python/Python312/python C:\Users\pingping2023\Downloads\pythontest\hello.py

Append: □ msg. payload

extra input parameters

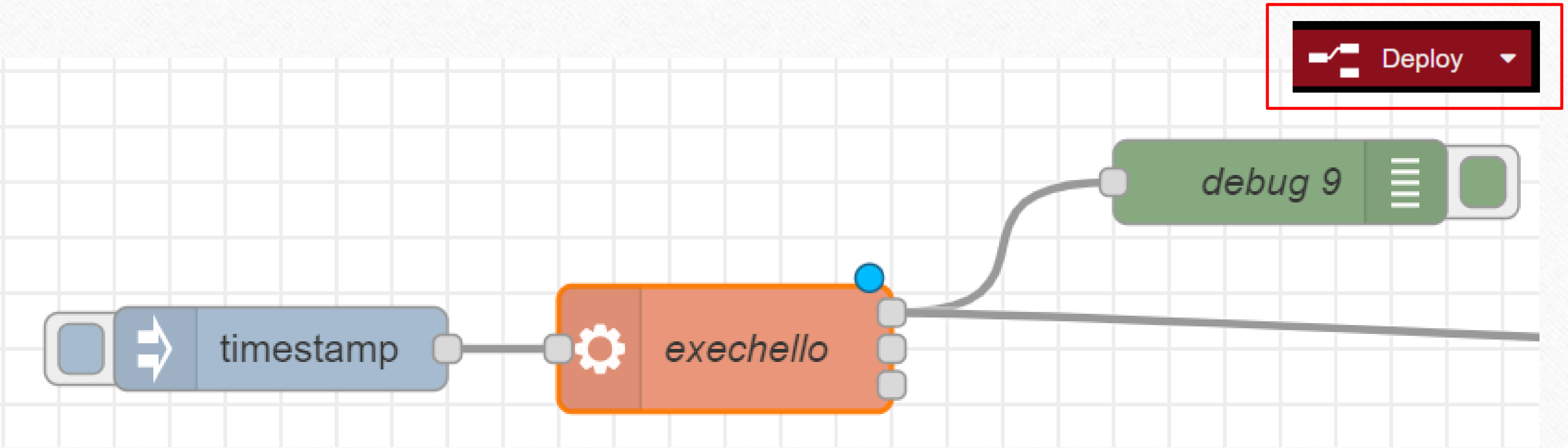
Output: when the command is complete - exec mode

Timeout: optional seconds

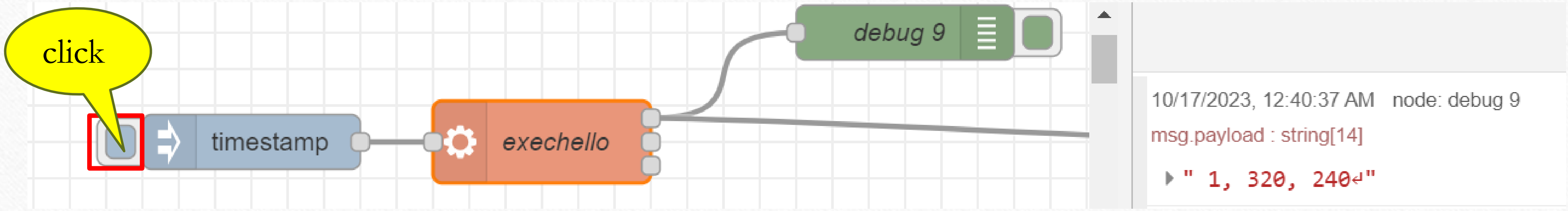
Hide console: □

Name: exechello

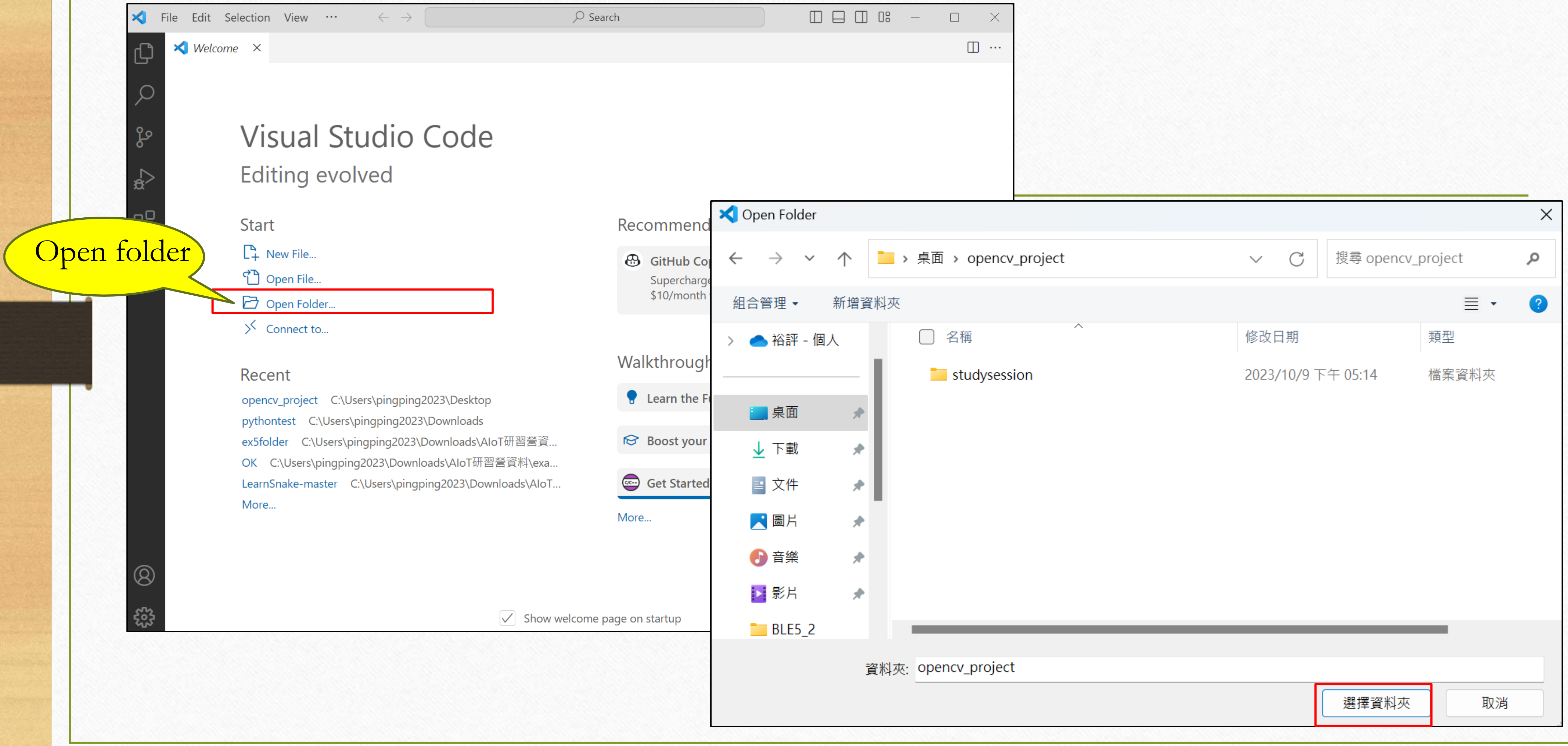
Deploy



Trigger



```
10/17/2023, 12:40:37 AM   node: debug 9
msg.payload : string[14]
▸ " 1, 320, 240↵"
```



Open folder

Visual Studio Code

Editing evolved

Start

New File...

Open File...

Open Folder...

Connect to...

Recent

opencv_project C:\Users\pingping2023\Desktop

pythontest C:\Users\pingping2023\Downloads

ex5folder C:\Users\pingping2023\Downloads\AIoT研習營資...

OK C:\Users\pingping2023\Downloads\AIoT研習營資料\exa...

LearnSnake-master C:\Users\pingping2023\Downloads\AIoT...

More...

Recommend

GitHub Co

Supercharge

$10/month

Walkthrough

Learn the F

Boost your

Get Started

More...

✓ Show welcome page on startup

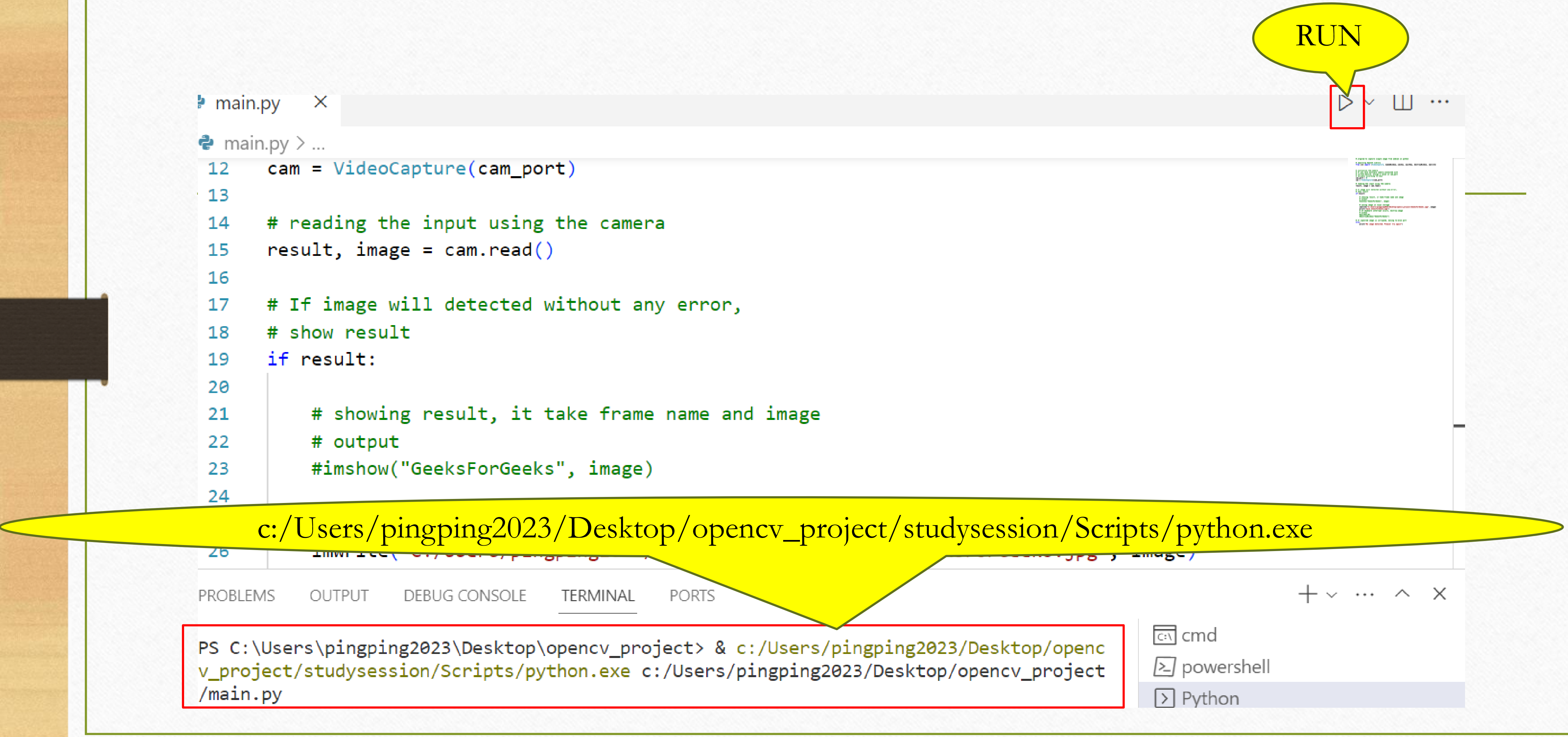
Open Folder

桌面 › opencv_project

搜尋 opencv_project

名稱	修改日期	類型
studysession	2023/10/9 下午 05:14	檔案資料夾

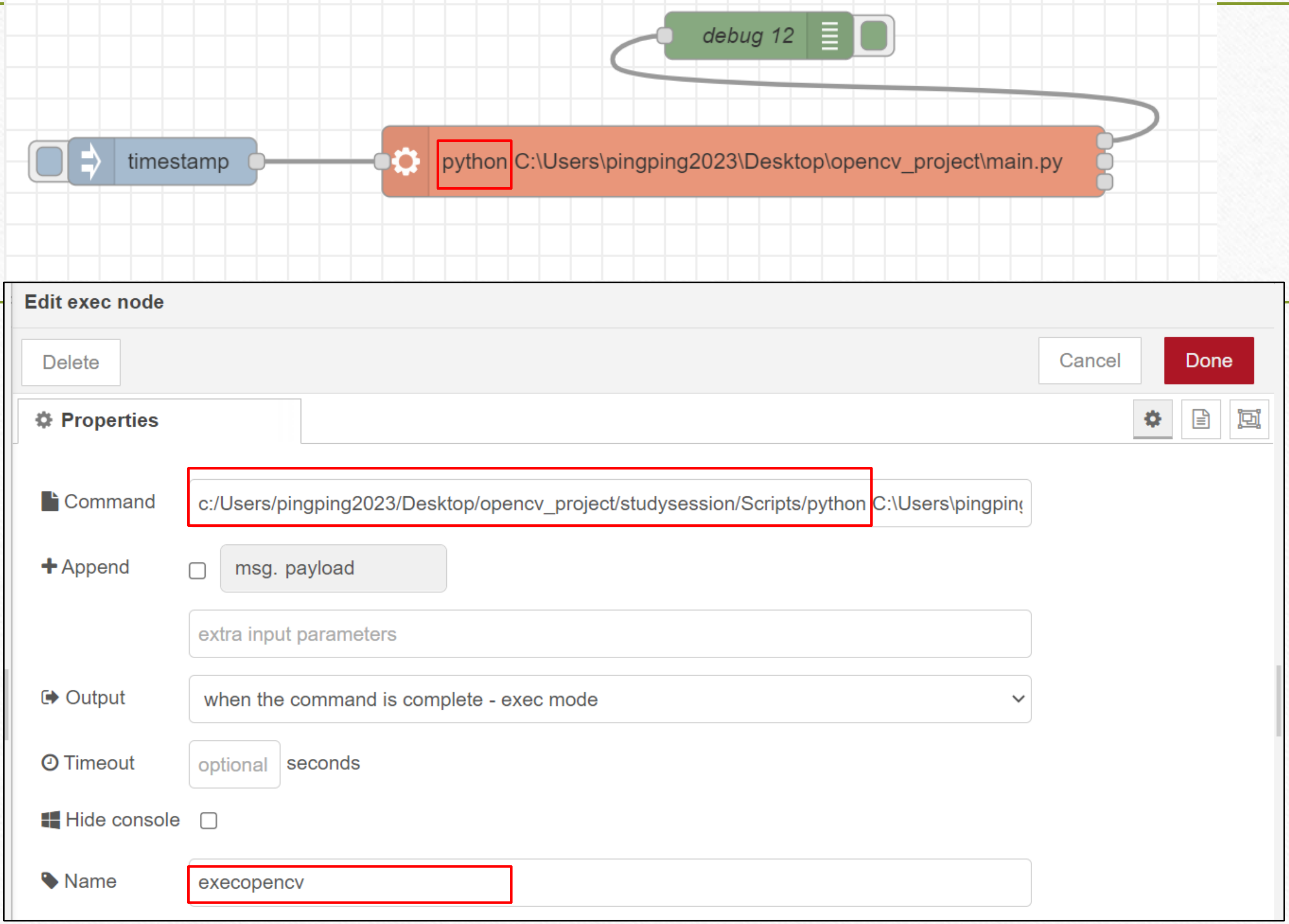
資料夾: opencv_project



```
cam = VideoCapture(cam_port)

# reading the input using the camera
result, image = cam.read()

# If image will detected without any error,
# show result
if result:

    # showing result, it take frame name and image
    # output
    #imshow("GeeksForGeeks", image)

    imwrite("c:/Users/pingping                    .jpg", image)
```

c:/Users/pingping2023/Desktop/opencv_project/studysession/Scripts/python.exe

```
PS C:\Users\pingping2023\Desktop\opencv_project> & c:/Users/pingping2023/Desktop/opencv_project/studysession/Scripts/python.exe c:/Users/pingping2023/Desktop/opencv_project/main.py
```



python C:\Users\pingping2023\Desktop\opencv_project\main.py

Command: c:/Users/pingping2023/Desktop/opencv_project/studysession/Scripts/python C:\Users\pingpinc

☐ Append: msg. payload

extra input parameters

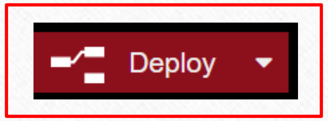
Output: when the command is complete - exec mode

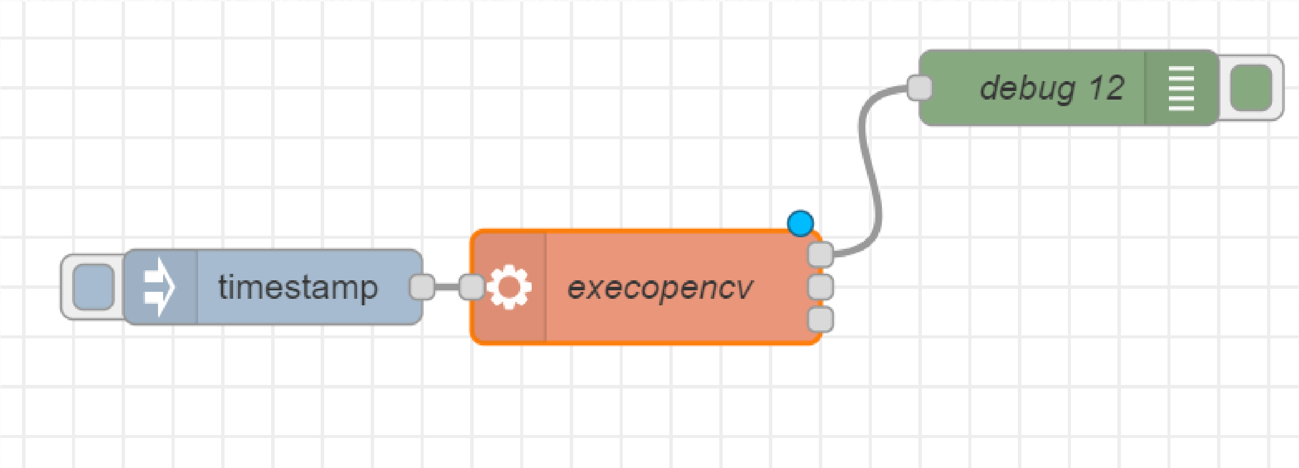
Timeout: optional seconds

Hide console ☐

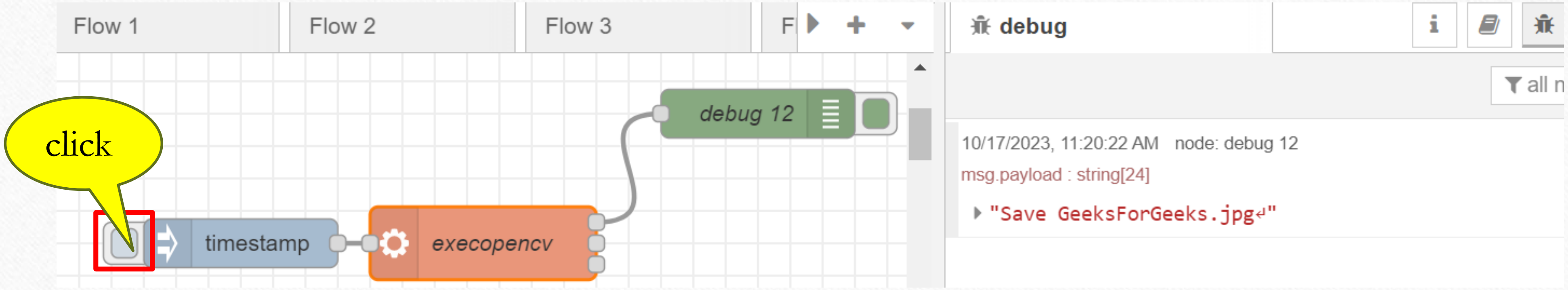
Name: execopencv

Deploy





Trigger

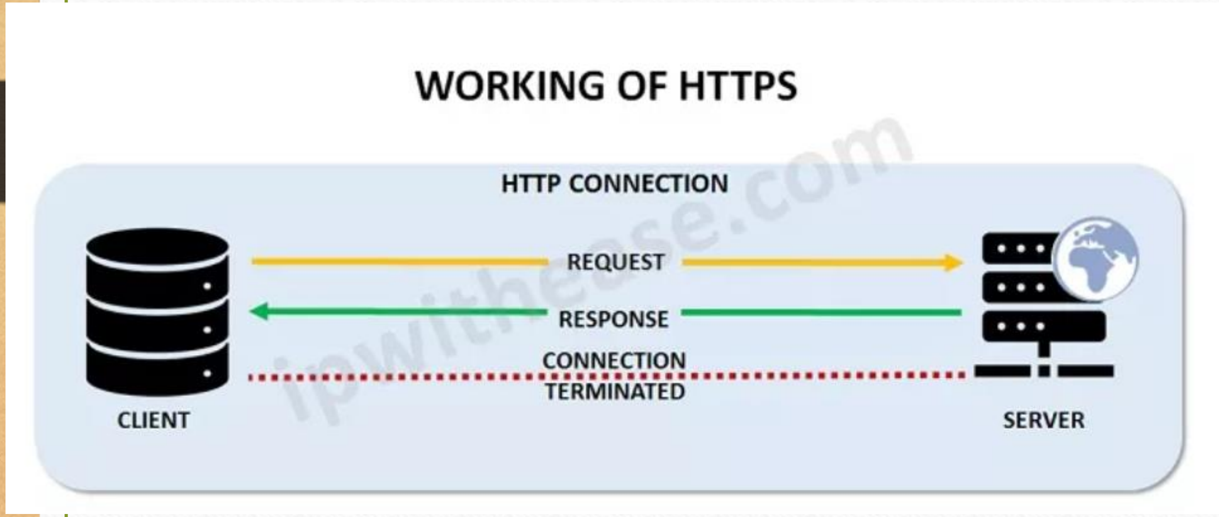


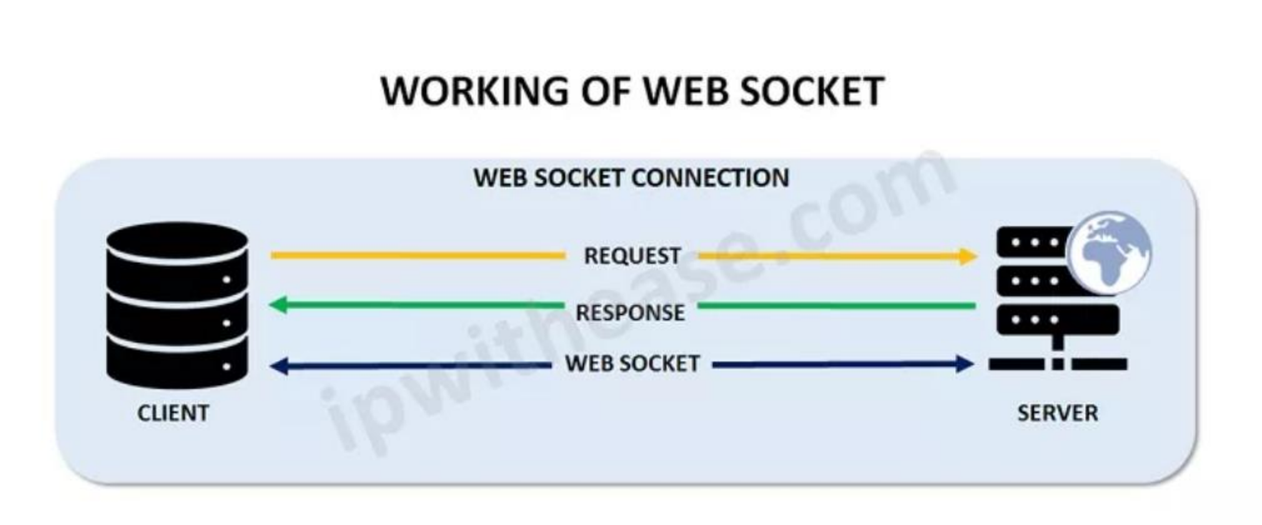
```
10/17/2023, 11:20:22 AM   node: debug 12
msg.payload : string[24]
▸ "Save GeeksForGeeks.jpg↵"
```

The WebSocket Protocol

- The WebSocket Protocol enables two-way communication between a client running untrusted code in a controlled environment to a remote host that has opted-in to communications from that code. The security model used for this is the origin-based security model commonly used by web browsers. The protocol consists of an opening handshake followed by basic message framing, layered over TCP. The goal of this technology is to provide a mechanism for browser-based applications that need two-way communication with servers that does not rely on opening multiple HTTP connections

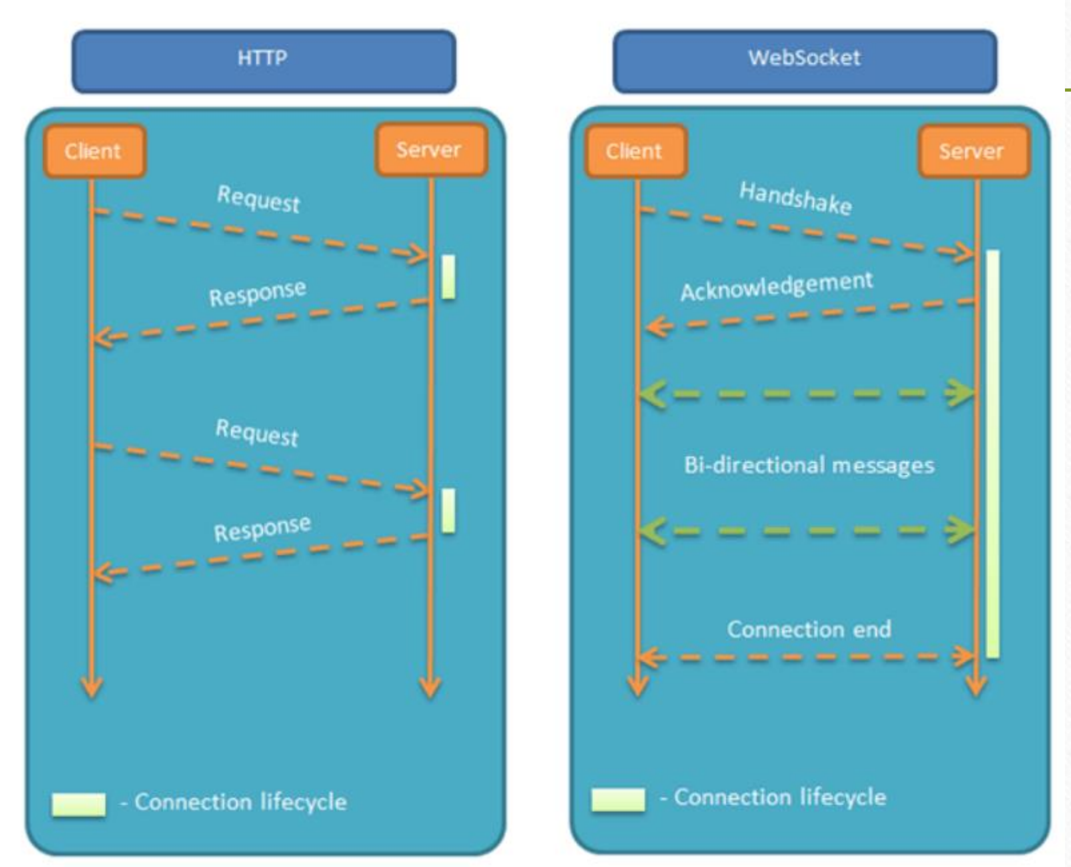
HTTP vs. WebSocket





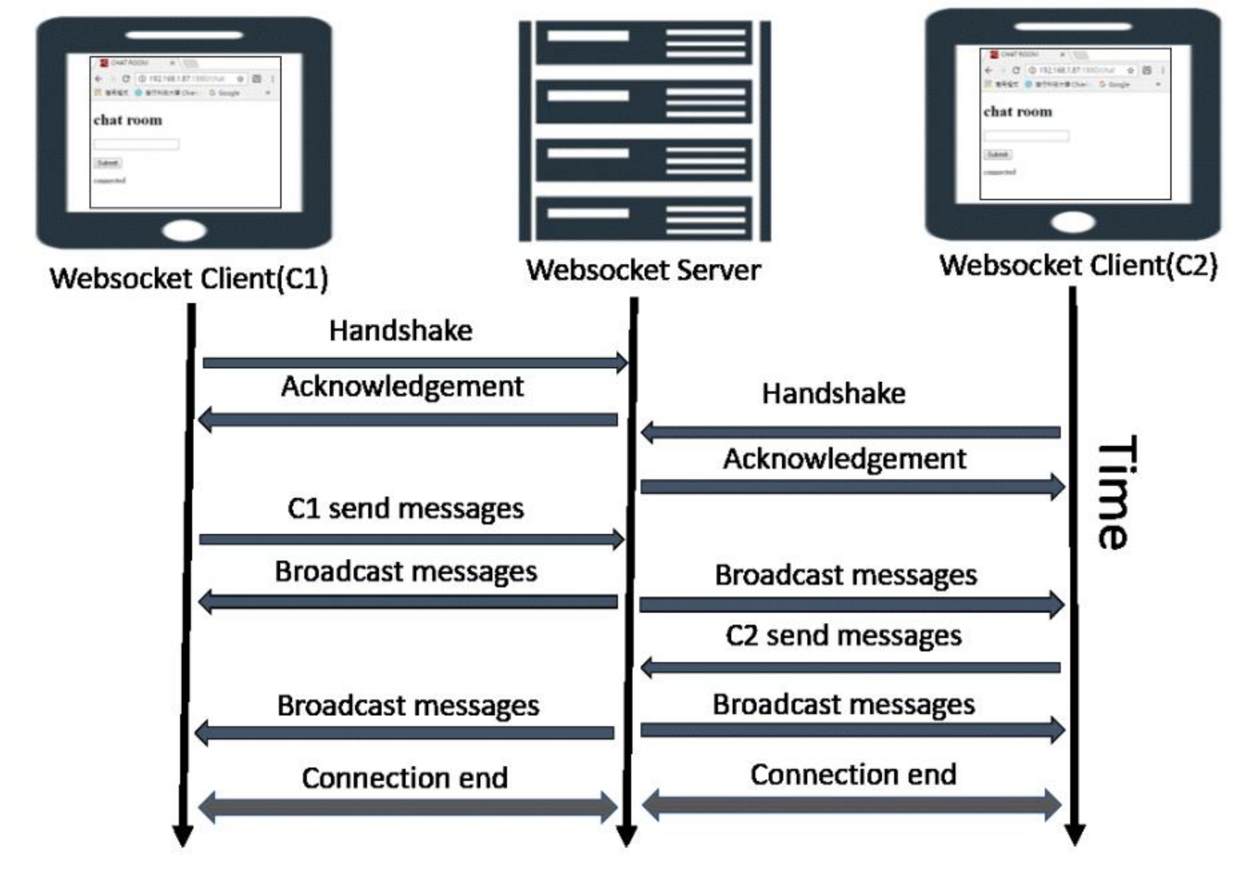
https://ipwithease.com/web-socket-vs-http/

HTTP vs. WebSocket

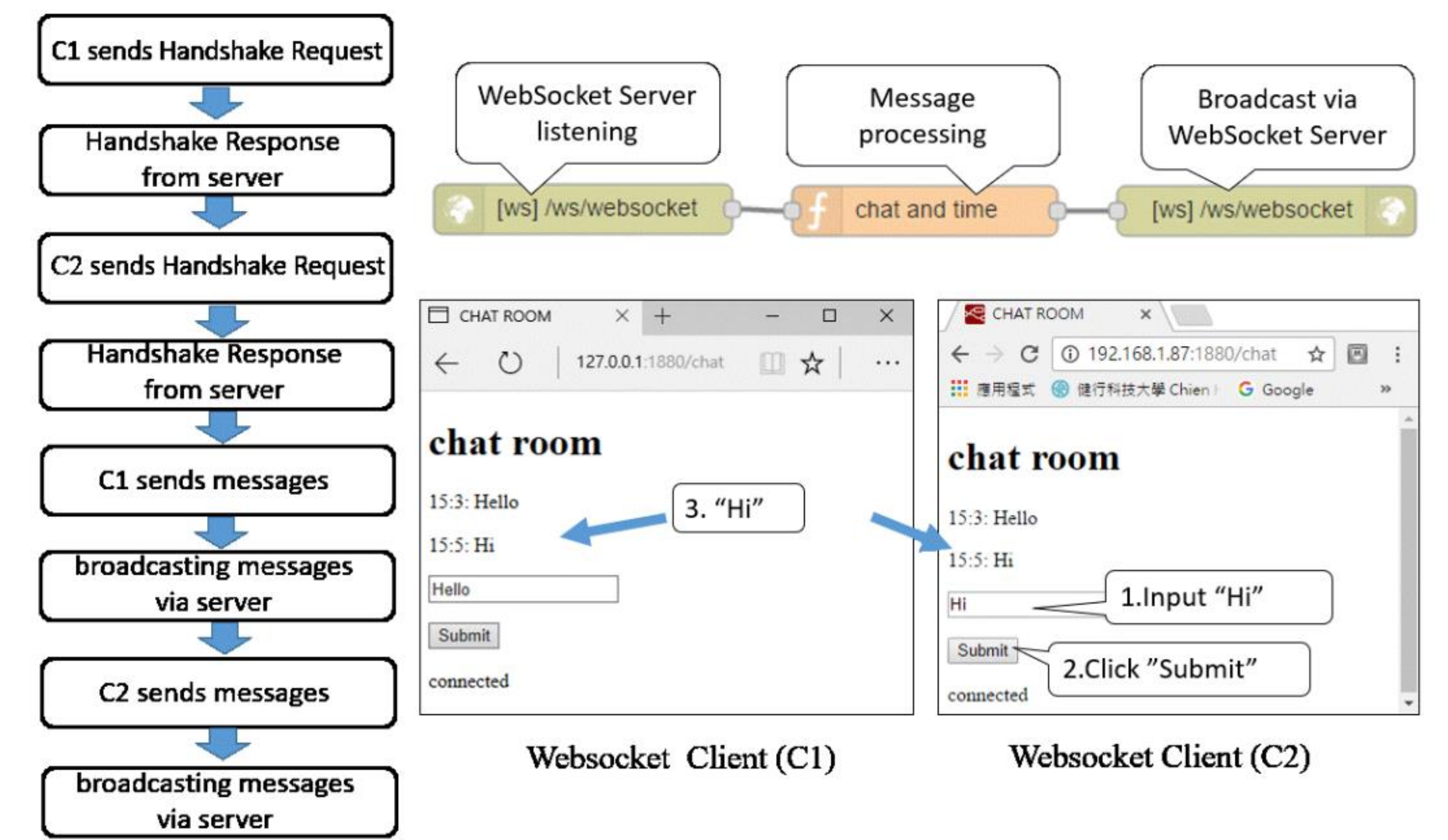


https://www.xoriant.com/blog/websocket-web-is-stateful-now

Exercise 6-1 Design a chat room



Design a chat room



C1 sends Handshake Request

Handshake Response from server

C2 sends Handshake Request

Handshake Response from server

C1 sends messages

broadcasting messages via server

C2 sends messages

broadcasting messages via server

WebSocket Server listening

Message processing

Broadcast via WebSocket Server

chat room

15:3: Hello

15:5: Hi

Hello

connected

3. "Hi"

Websocket Client (C1)

chat room

15:3: Hello

15:5: Hi

Hi

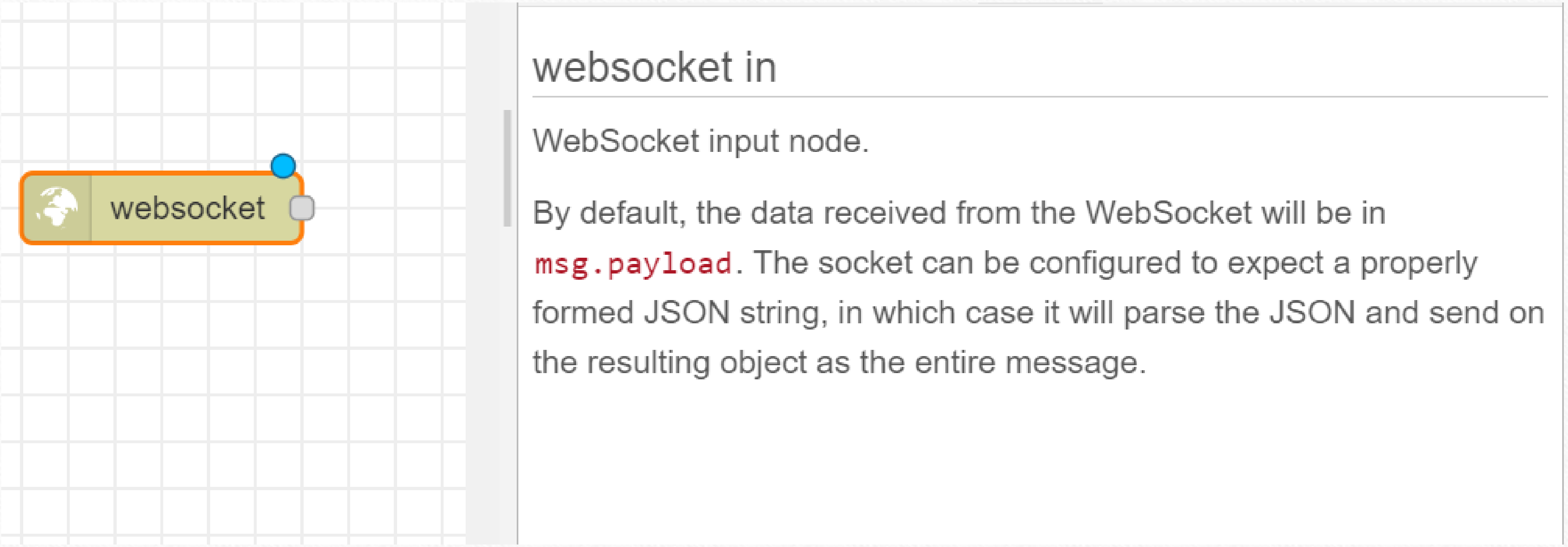
1.Input "Hi"

2.Click "Submit"

connected

Websocket Client (C2)

"WebSocket in" Node

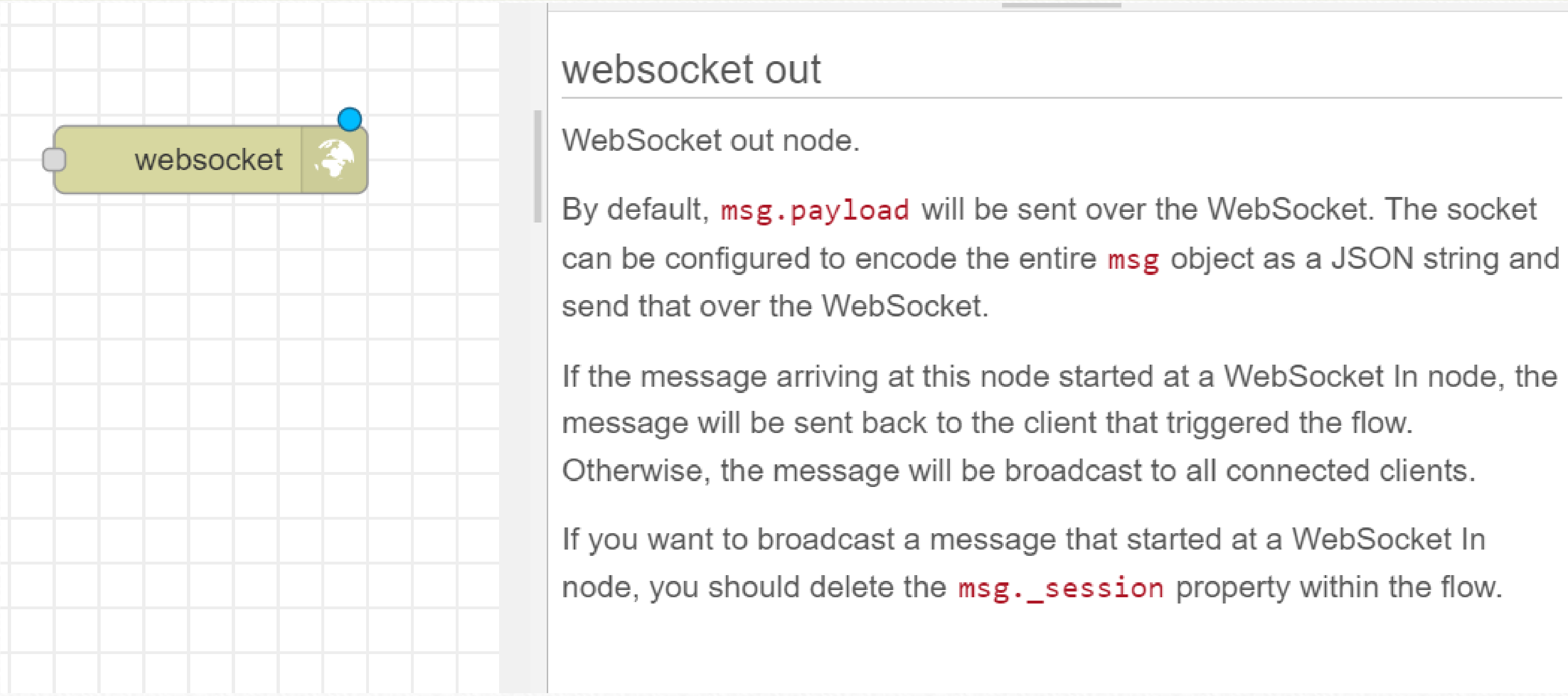


websocket in

WebSocket input node.

By default, the data received from the WebSocket will be in `msg.payload`. The socket can be configured to expect a properly formed JSON string, in which case it will parse the JSON and send on the resulting object as the entire message.

“WebSocket out” Node



websocket out

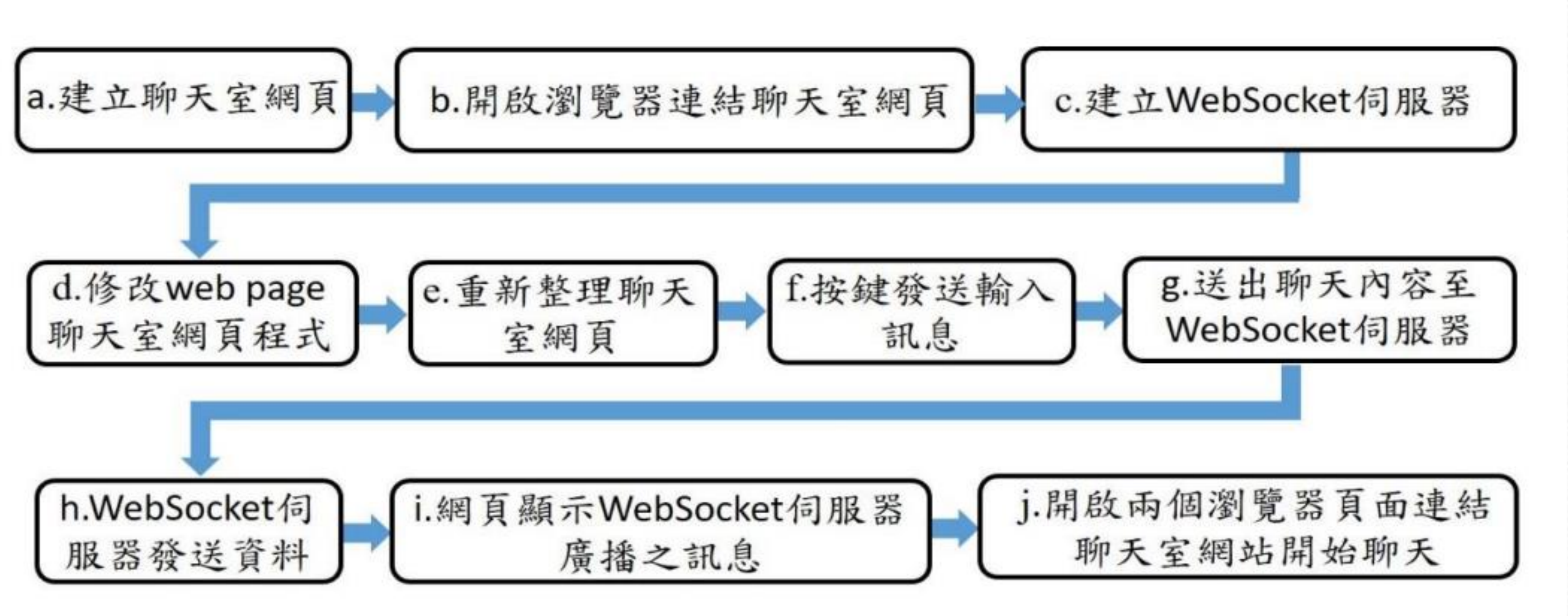
WebSocket out node.

By default, `msg.payload` will be sent over the WebSocket. The socket can be configured to encode the entire `msg` object as a JSON string and send that over the WebSocket.

If the message arriving at this node started at a WebSocket In node, the message will be sent back to the client that triggered the flow. Otherwise, the message will be broadcast to all connected clients.

If you want to broadcast a message that started at a WebSocket In node, you should delete the `msg._session` property within the flow.

Processes



a.建立聊天室網頁

b.開啟瀏覽器連結聊天室網頁

c.建立WebSocket伺服器

d.修改web page 聊天室網頁程式

e.重新整理聊天室網頁

f.按鍵發送輸入訊息

g.送出聊天內容至WebSocket伺服器

h.WebSocket伺服器發送資料

i.網頁顯示WebSocket伺服器廣播之訊息

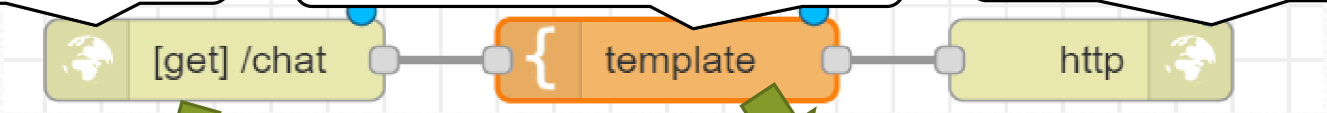
j.開啟兩個瀏覽器頁面連結聊天室網站開始聊天

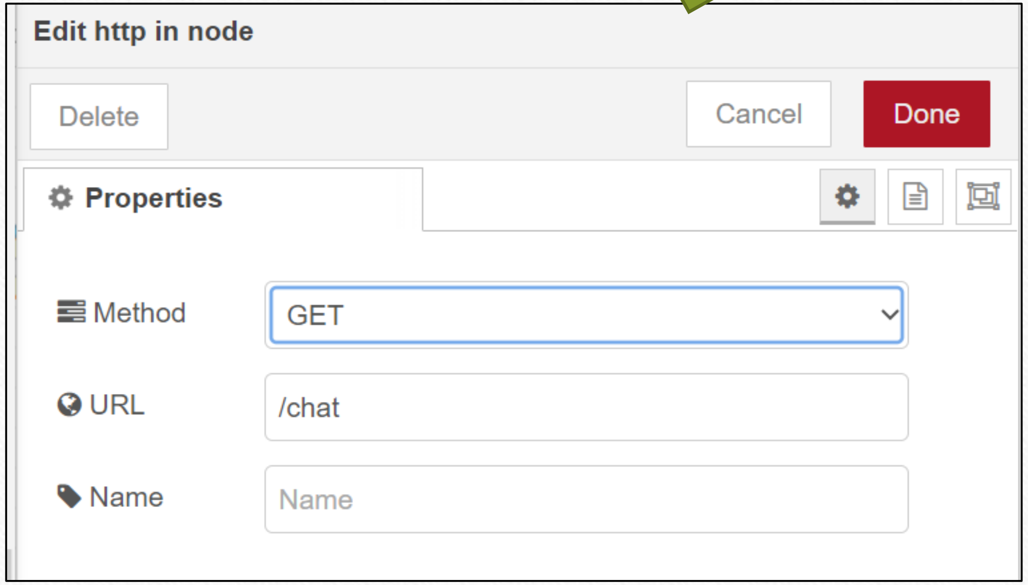
Step1: Create chat room HTML

network>http in

function>template

network> http response





Edit http in node

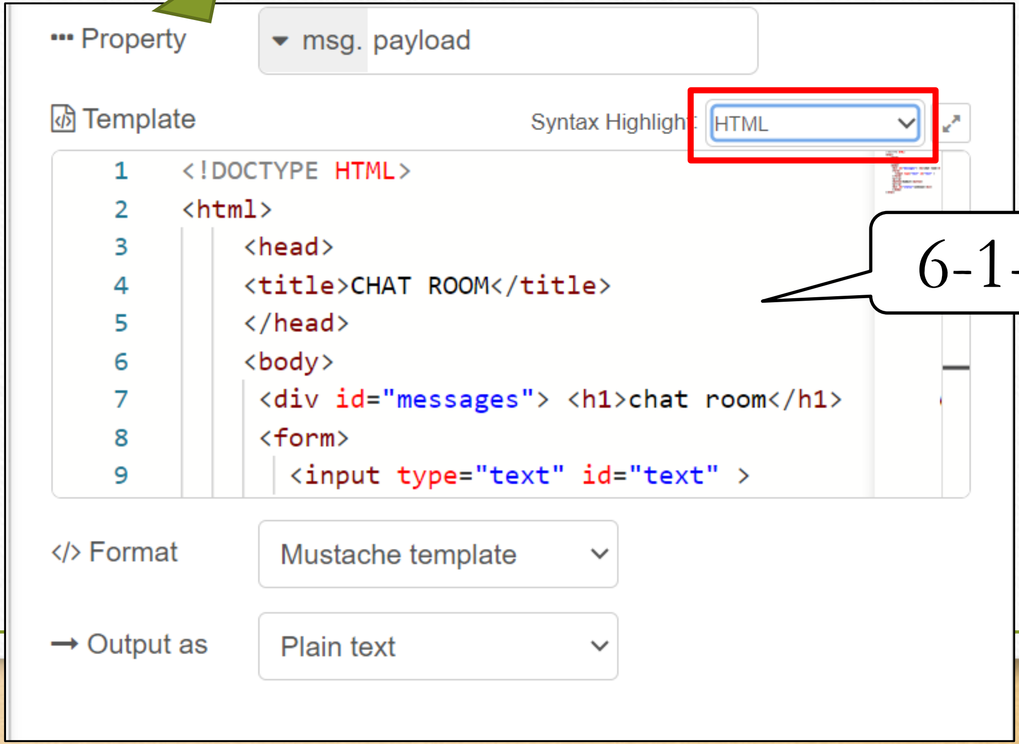
Delete | Cancel | Done

Properties

Method: GET

URL: /chat

Name: Name



Property: msg. payload

Template — Syntax Highlight: HTML

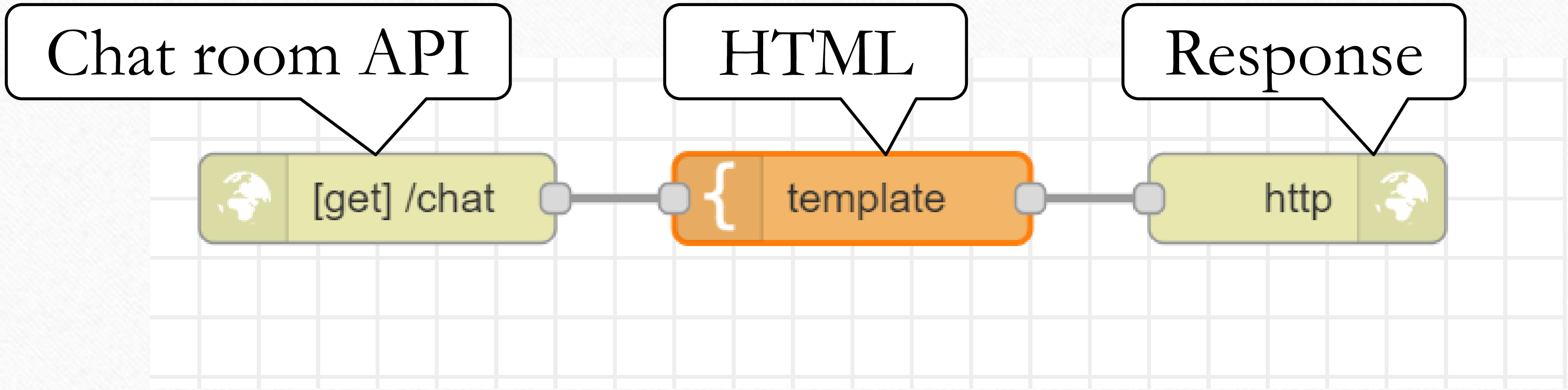
```
<!DOCTYPE HTML>
<html>
    <head>
    <title>CHAT ROOM</title>
    </head>
    <body>
     <div id="messages"> <h1>chat room</h1>
     <form>
       <input type="text" id="text" >
```

Format: Mustache template

Output as: Plain text

6-1-1.txt

Chat Room API Flow



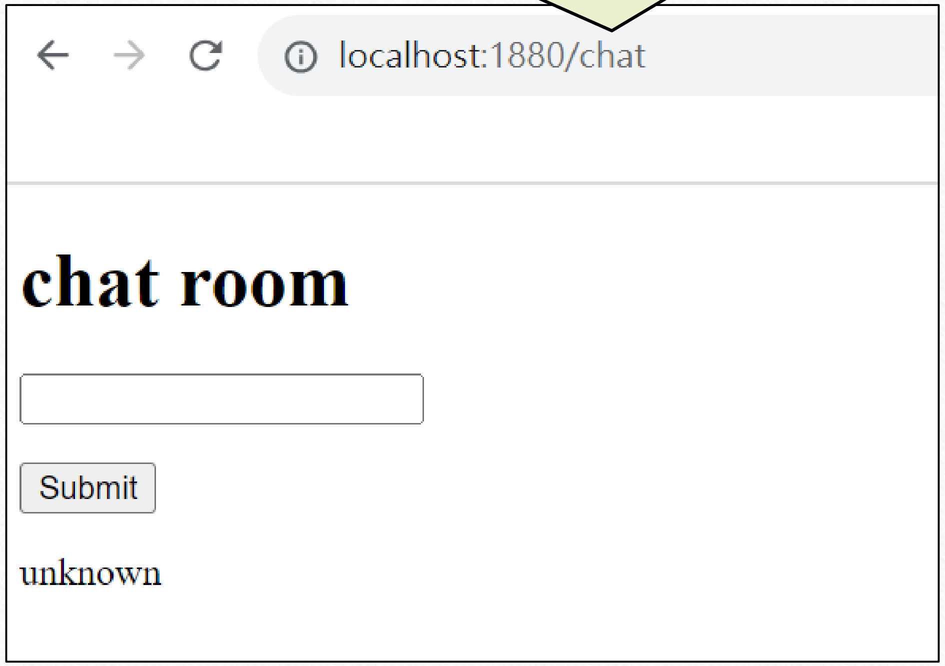
Chat room API

HTML

Response

Step2: open the chat room webpage

http://localhost:1880/chat

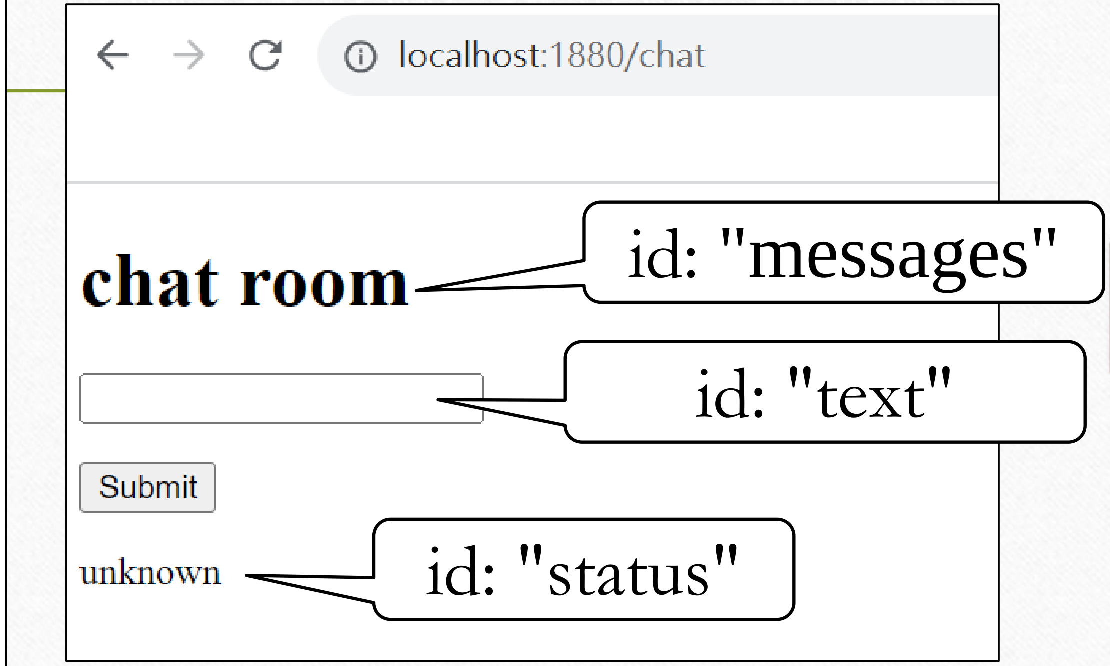


localhost:1880/chat

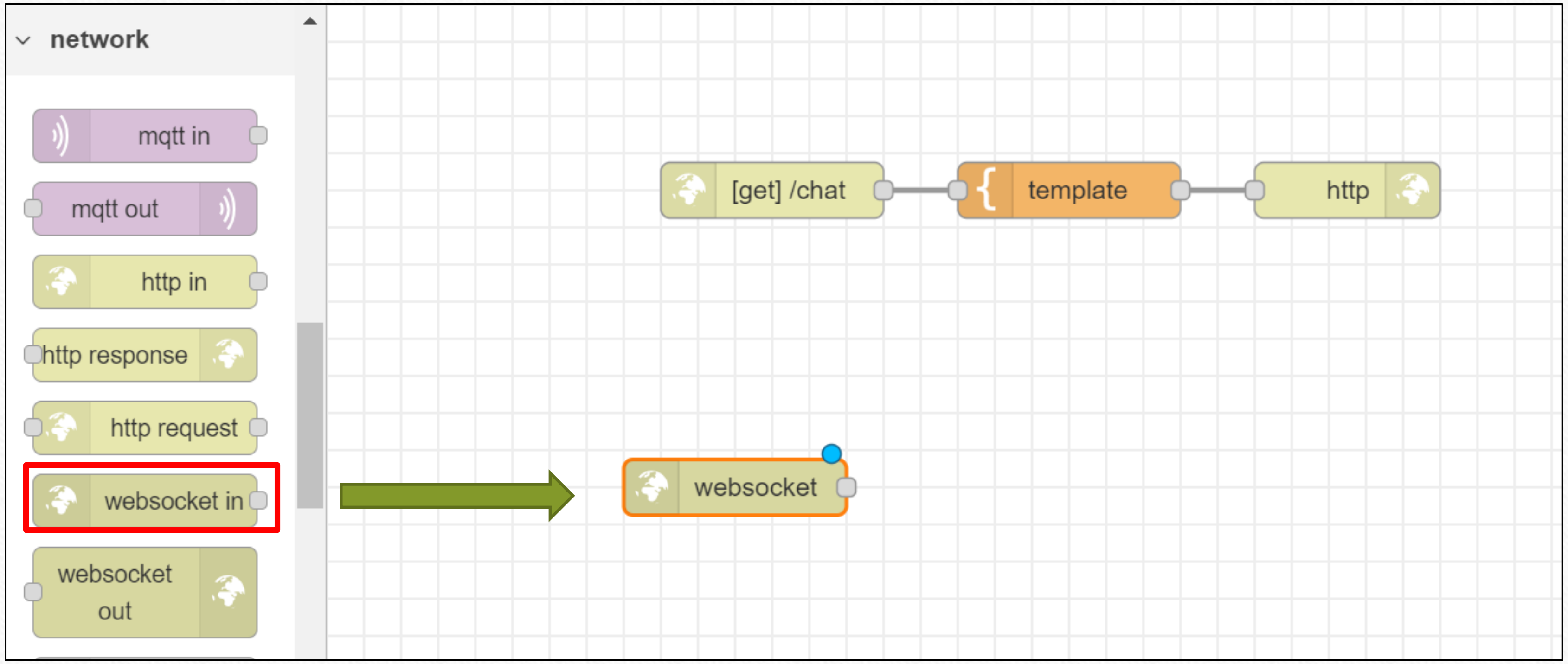
chat room

unknown

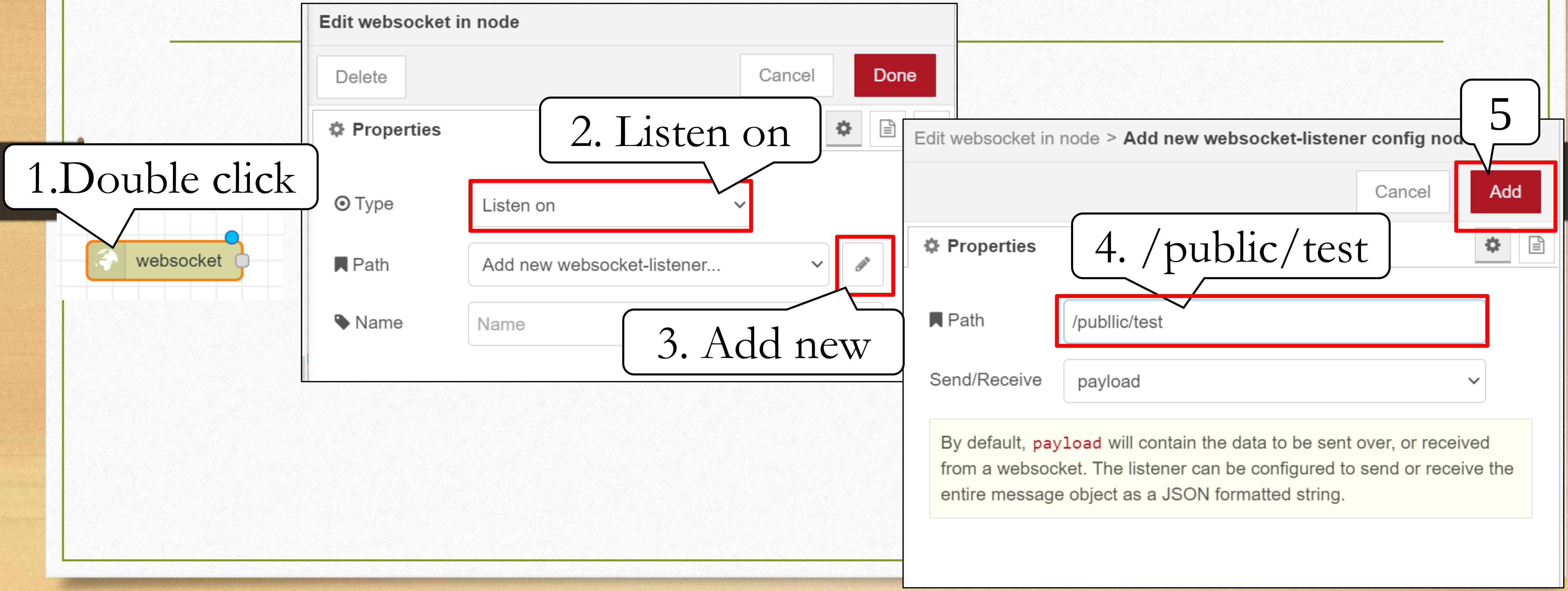
```
<!DOCTYPE HTML>
<html>
    <head>
    <title>CHAT ROOM</title>
    </head>
    <body>
     <div id="messages"> <h1>chat room</h1>  </div>
     <form>
       <input type="text" id="text" >
     </form>
     <p></p>
     <button>Submit</button>
     <p></p>
     <div id="status">unknown</div>
     </body>
</html>
```



Step3: Add “websocket in”



Edit “websocket in” node



1.Double click

websocket

Edit websocket in node

Delete | Cancel | Done

Properties

2. Listen on

Type: Listen on

Path: Add new websocket-listener...

3. Add new

Name: Name

Edit websocket in node > Add new websocket-listener config nod

5

Cancel | Add

Properties

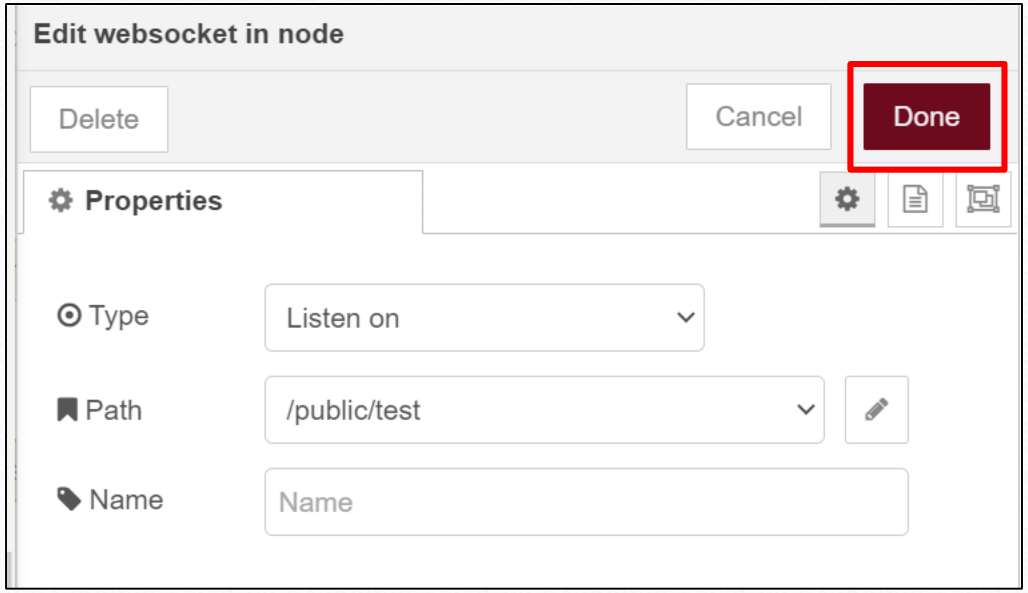
4. /public/test

Path: /publlic/test

Send/Receive: payload

By default, `payload` will contain the data to be sent over, or received from a websocket. The listener can be configured to send or receive the entire message object as a JSON formatted string.

Edit “websocket in” node

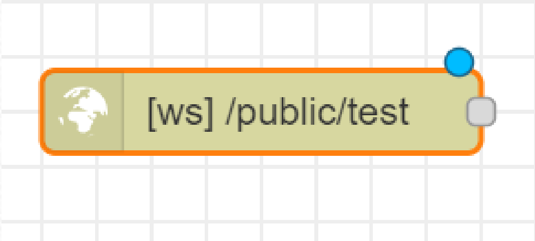


Edit websocket in node

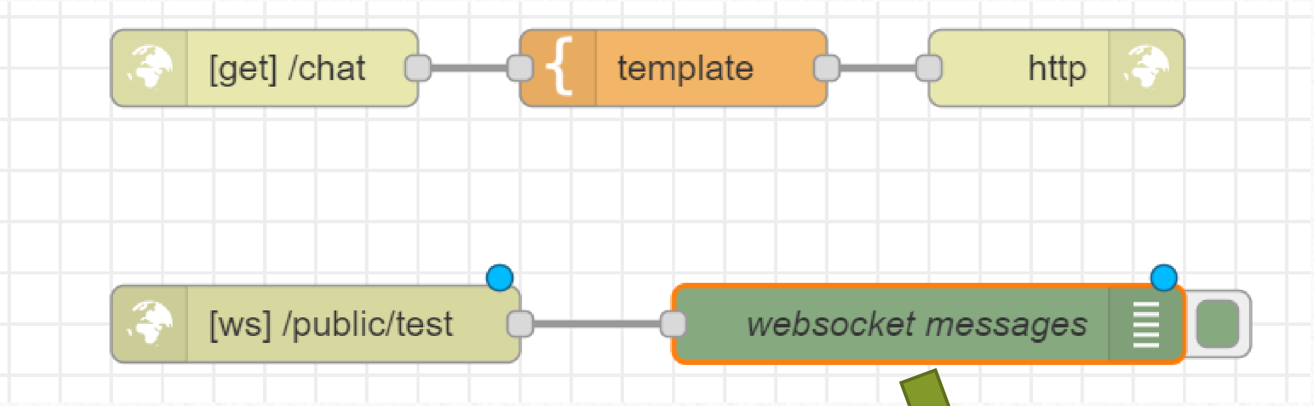
Type: Listen on

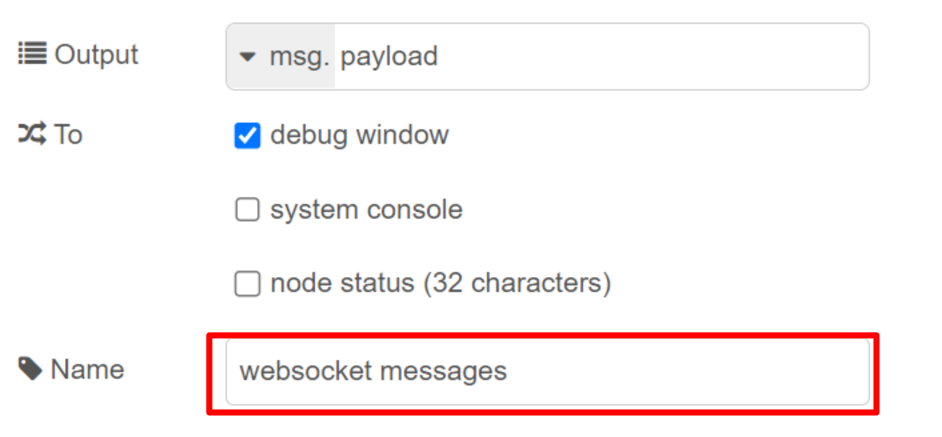
Path: /public/test

Name: Name



Step 4: Add a debug node



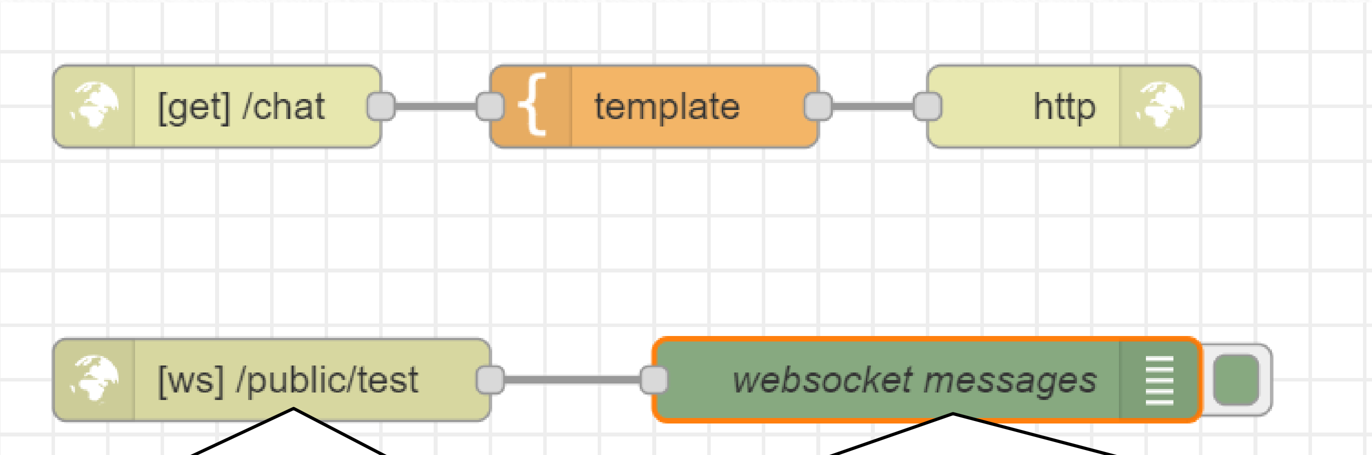


Output: msg. payload

To:
- ☑ debug window
- □ system console
- □ node status (32 characters)

Name: websocket messages

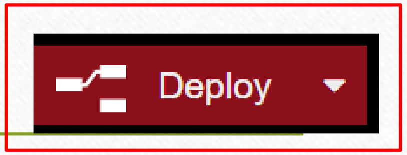
WebSocket Flow

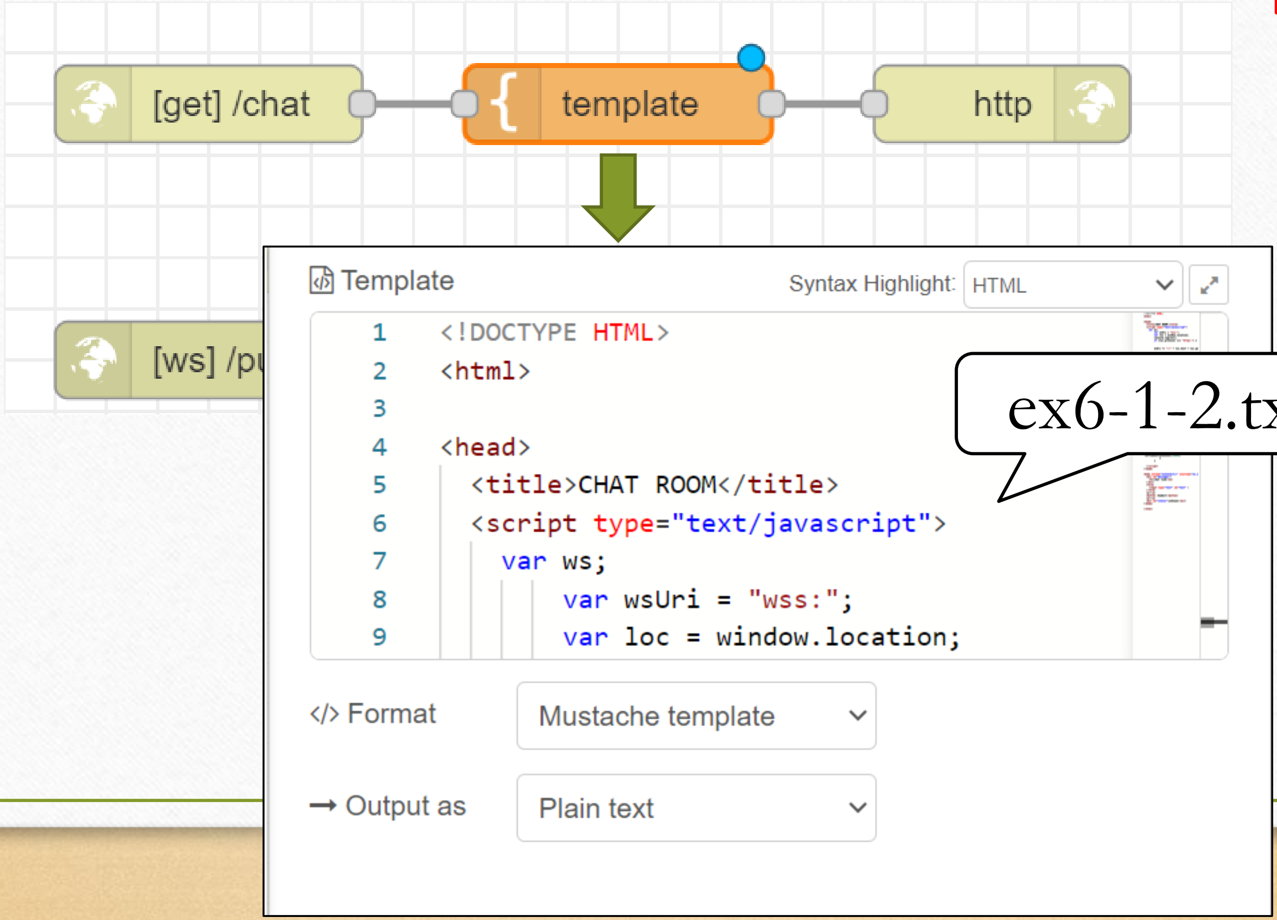


WebSocket Server Listening

Show the messages on the debug window

Step 5: Modify HTML





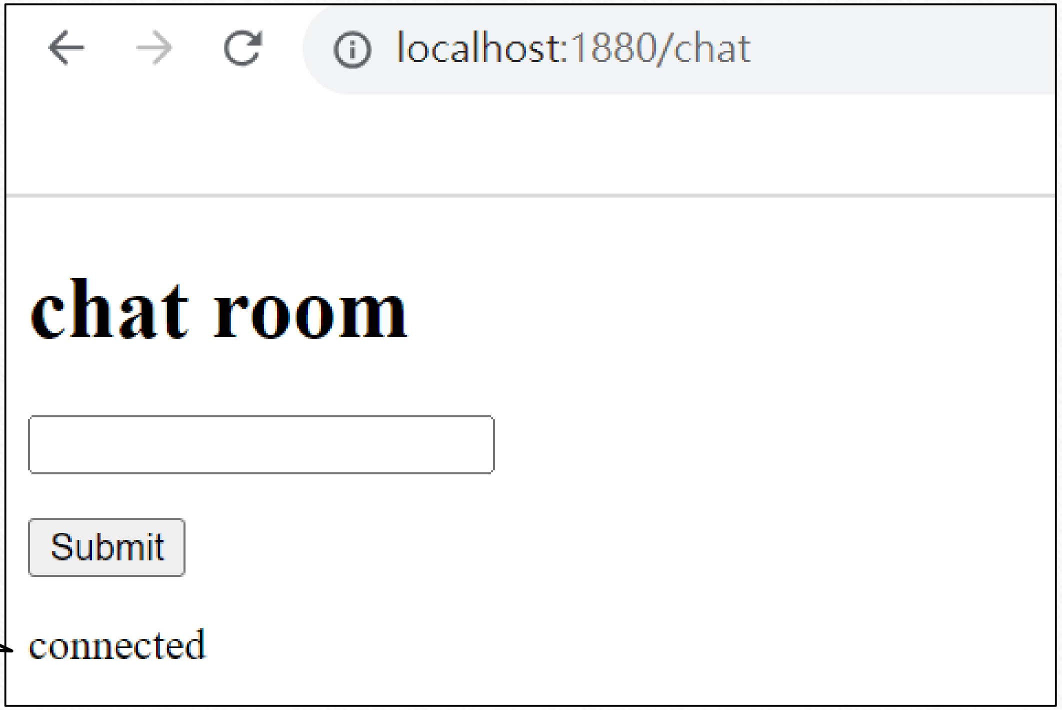
```
<!DOCTYPE HTML>
<html>

<head>
  <title>CHAT ROOM</title>
  <script type="text/javascript">
    var ws;
        var wsUri = "wss:";
        var loc = window.location;
```

Format: Mustache template

Output as: Plain text

ex6-1-2.txt

Step 6: Refresh the chat room web page



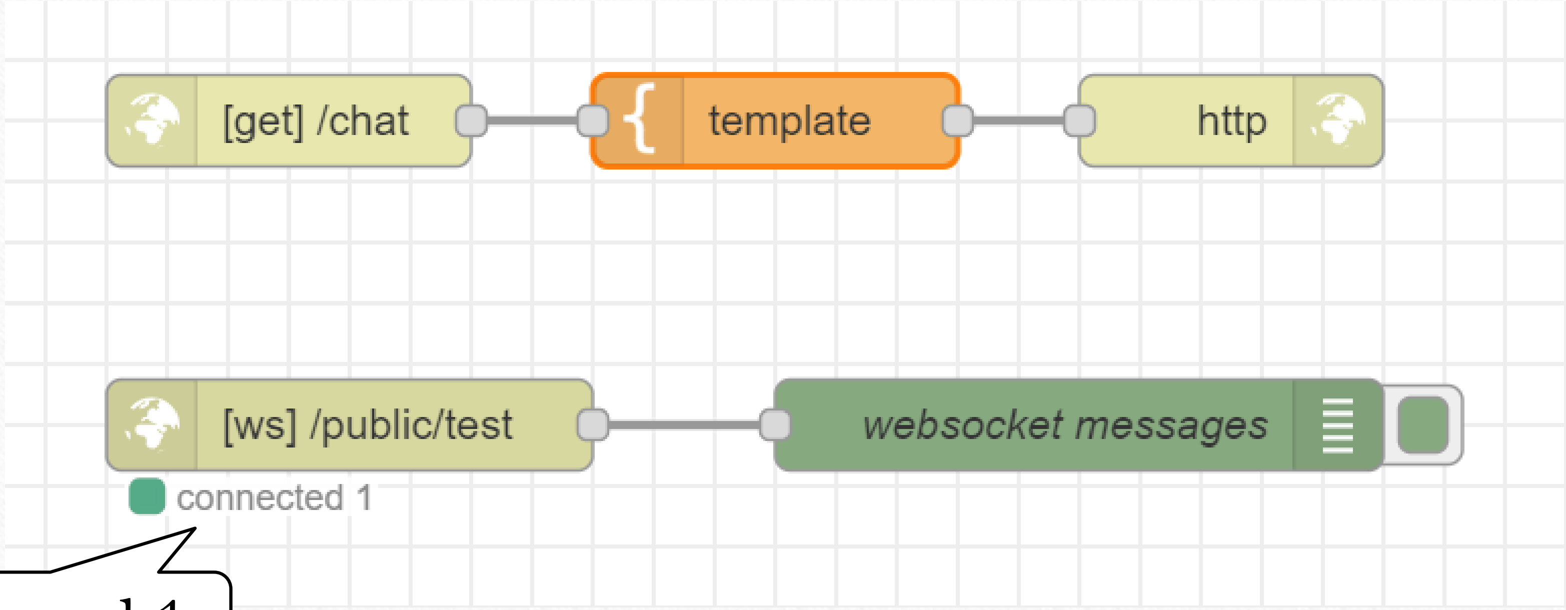
localhost:1880/chat

chat room

connected

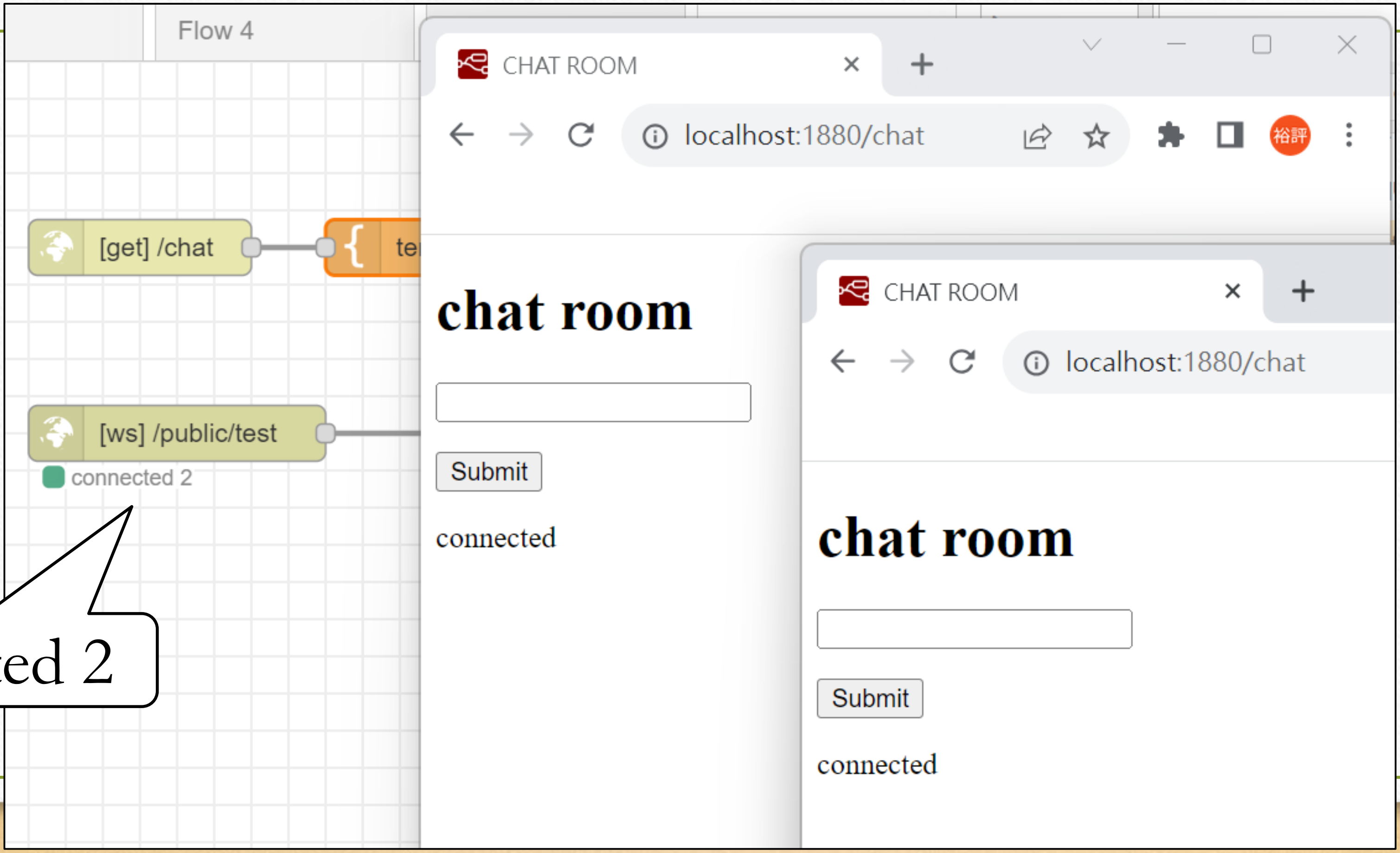
connected

Node-RED



connected 1

Step 7: Open the chat room web page in new window



chat room

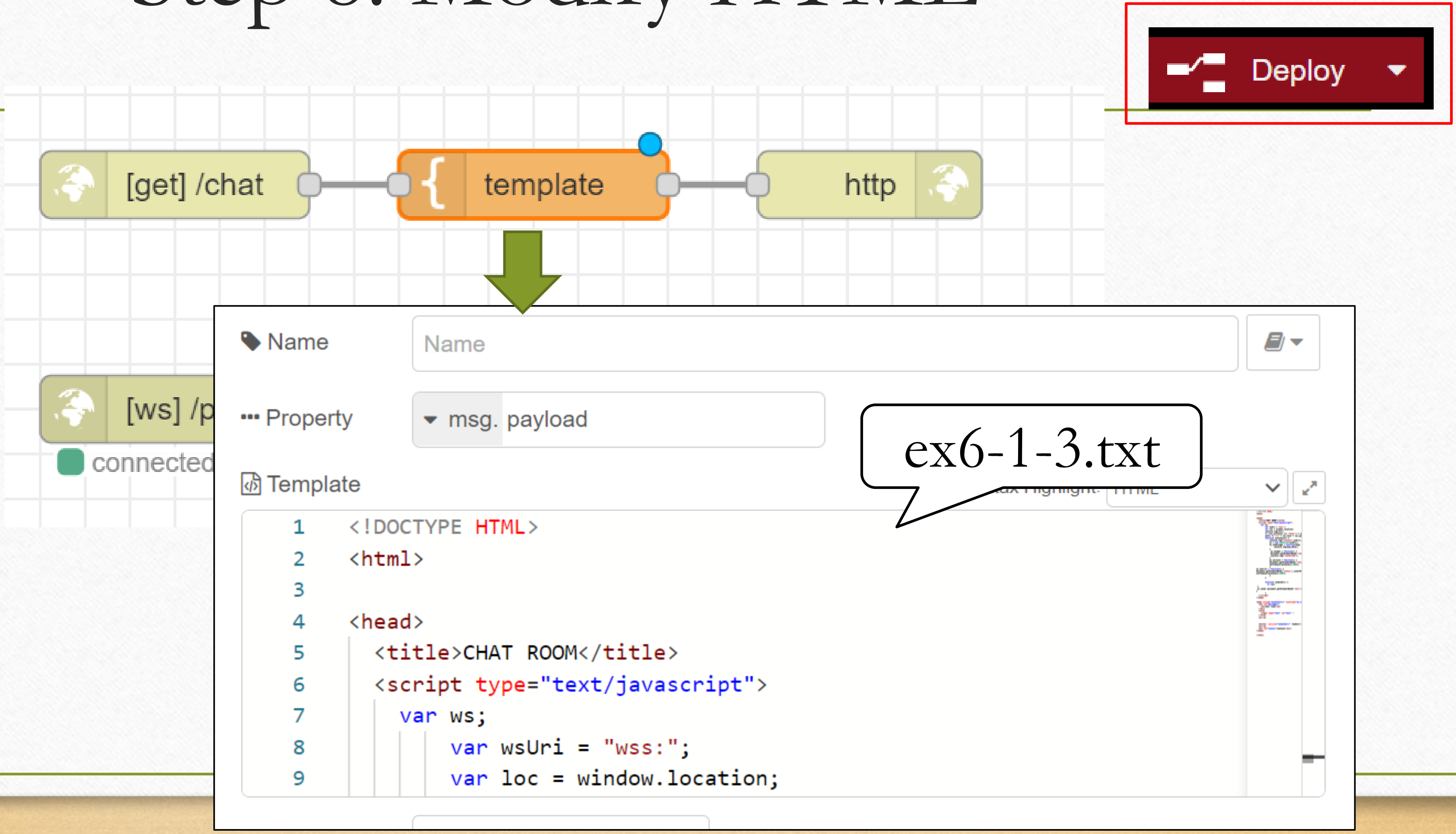
connected

chat room

connected

connected 2

Step 8: Modify HTML



Name: Name

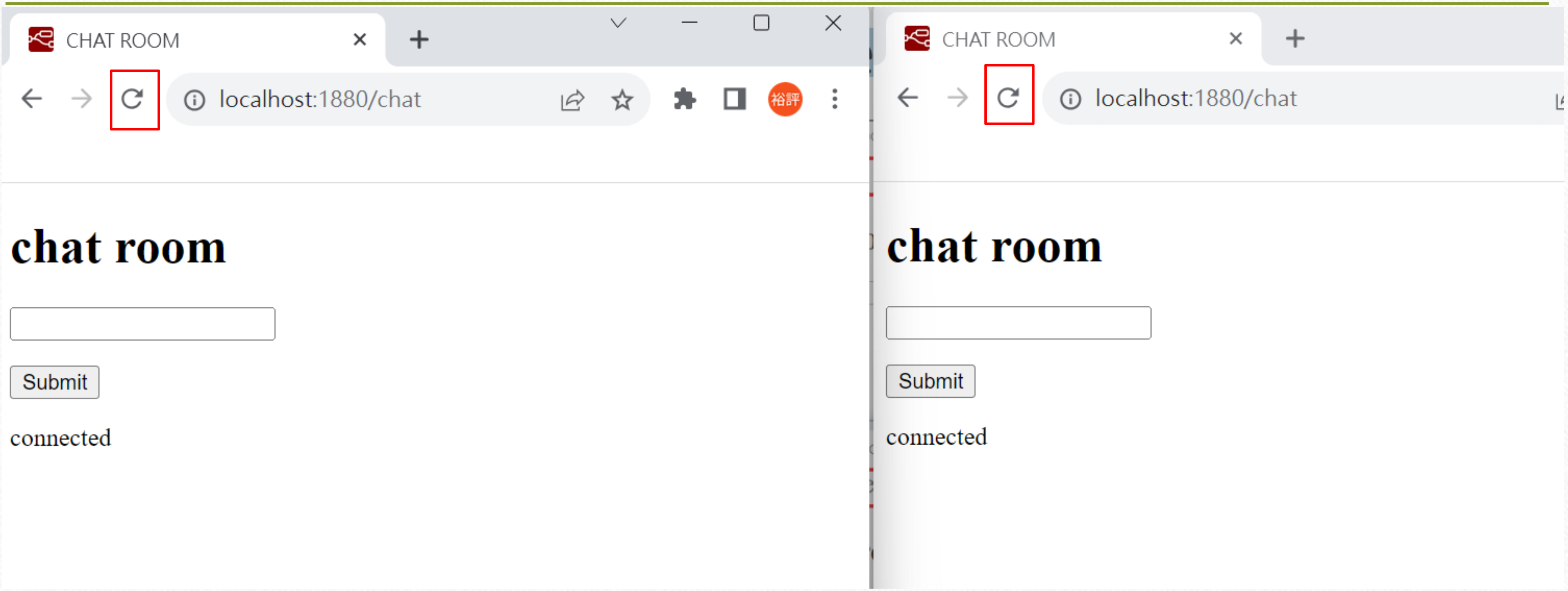
Property: msg. payload

Template

```
<!DOCTYPE HTML>
<html>

<head>
  <title>CHAT ROOM</title>
  <script type="text/javascript">
    var ws;
        var wsUri = "wss:";
        var loc = window.location;
```

ex6-1-3.txt

Step 9: Refresh chat room web pages



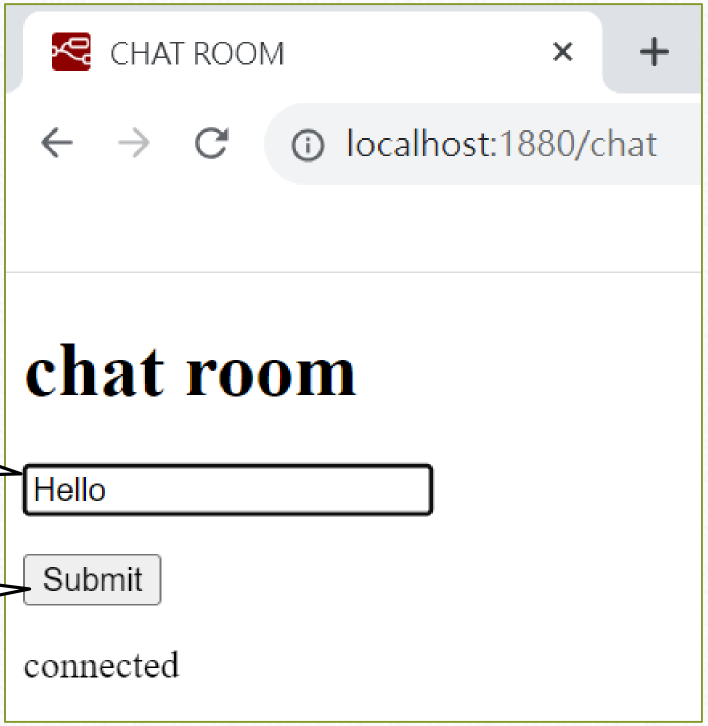
chat room

connected

chat room

connected

Step 10: Submit a message



chat room

Hello

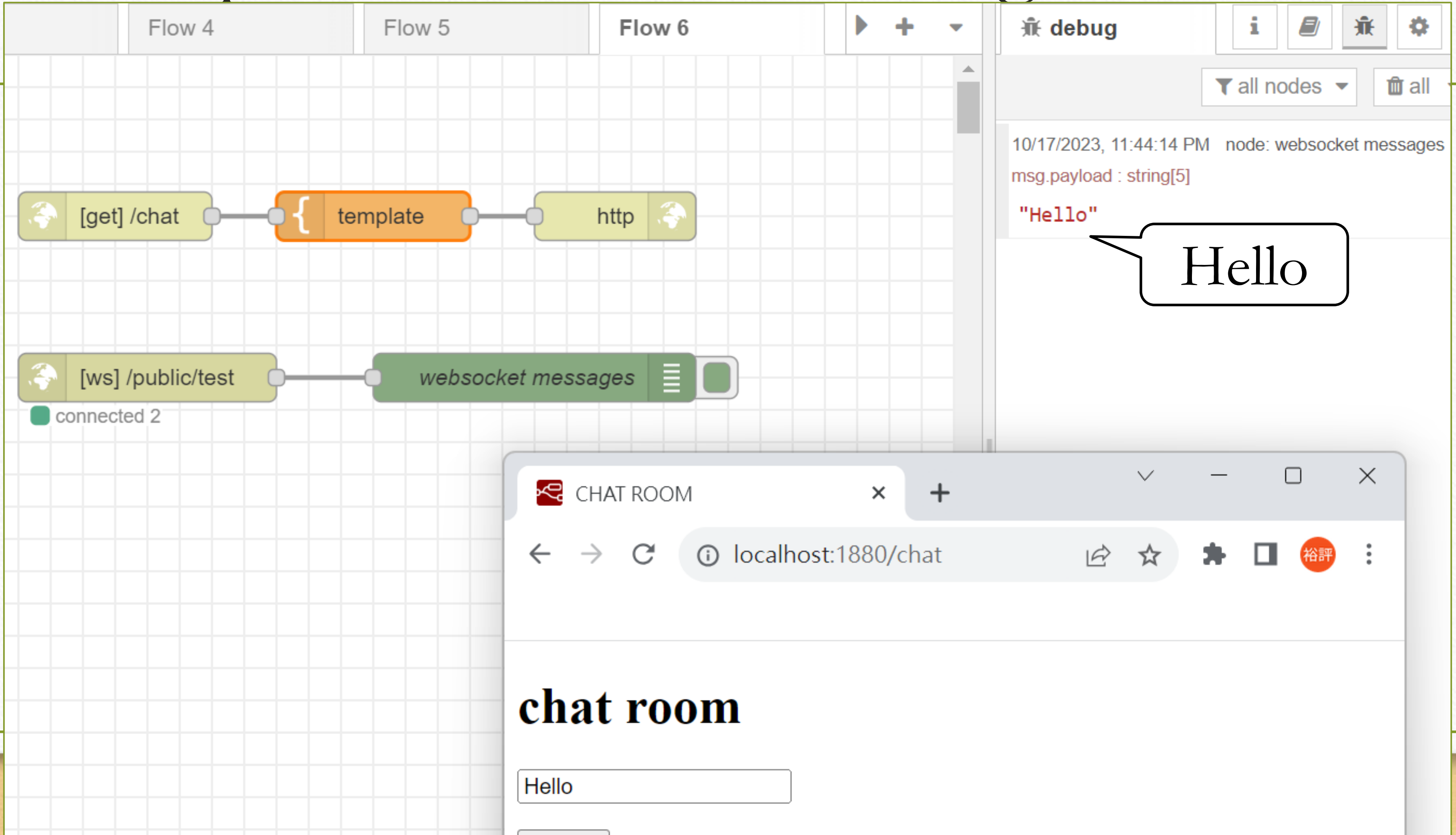
Submit

connected

1.Hello

2.Submit

Step 11: Check the debug window



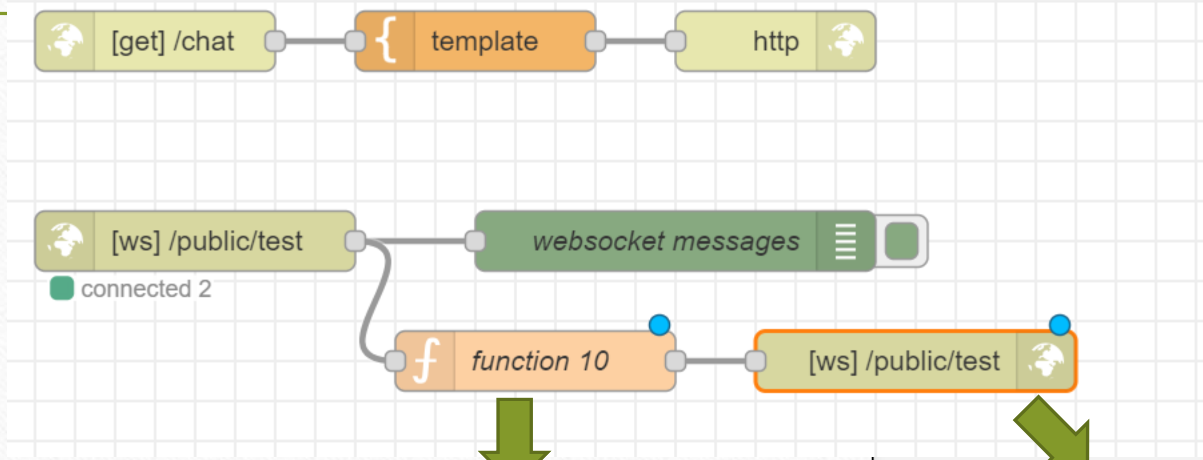
10/17/2023, 11:44:14 PM node: websocket messages
msg.payload : string[5]
"Hello"

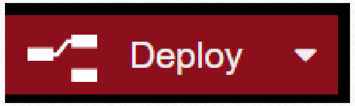
Hello

chat room

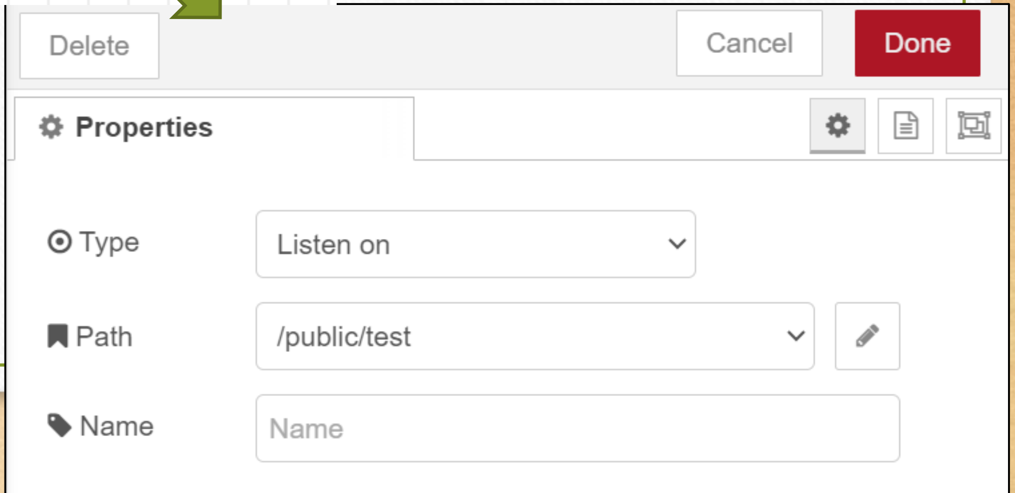
Hello

Step 12: Add a “function” node and a “websocket out” node





```
var d = new Date();
var time= d.getHours()+":"+d.getMinutes()+":";
var chat=msg.payload;
msg={};
msg.payload=time+" "+chat;

return msg;
```

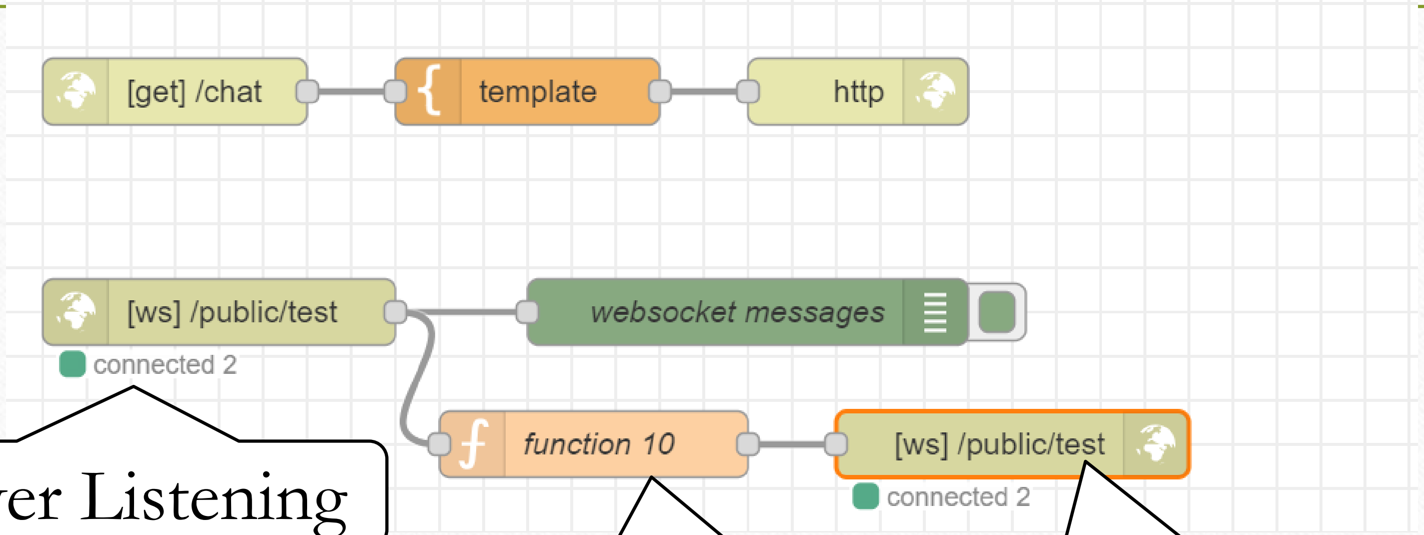


Type: Listen on

Path: /public/test

Name: Name

Flow

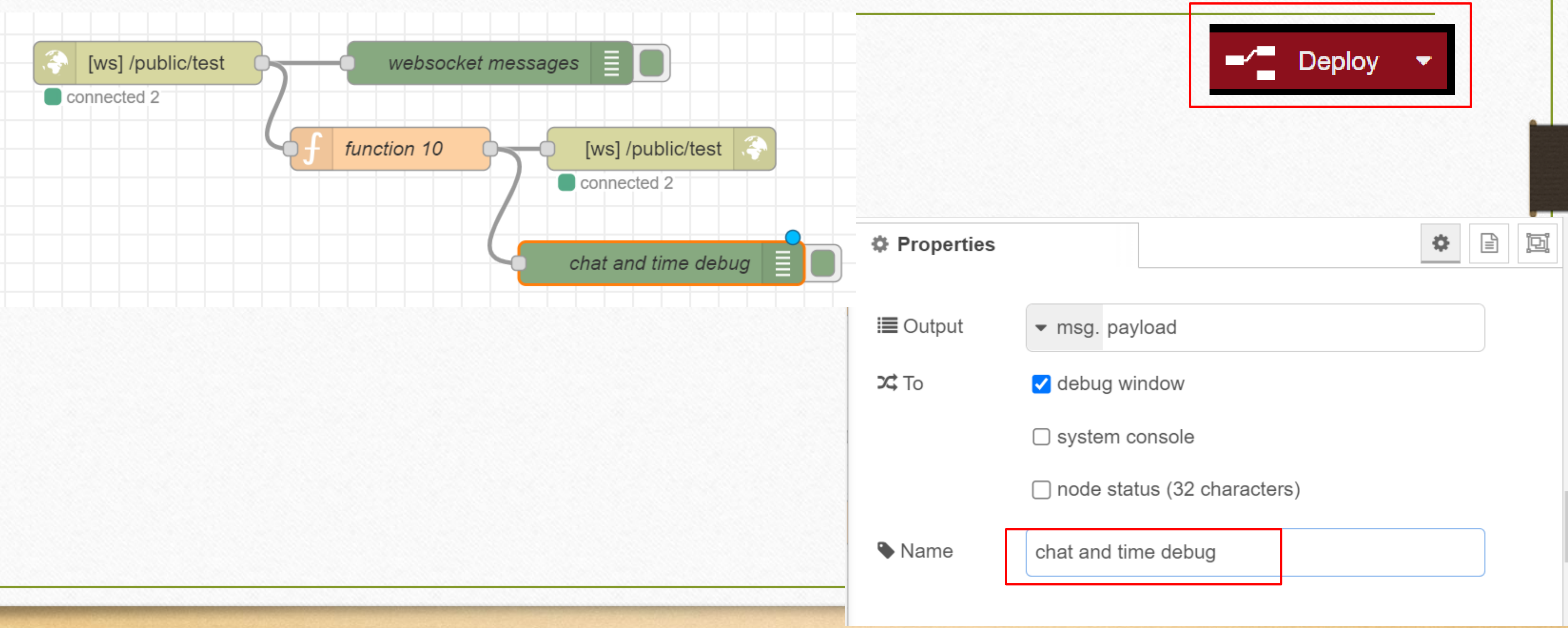


WebSocket Server Listening

Message processing

Broadcast via
WebSocket Server

Step 13: Add a debug node

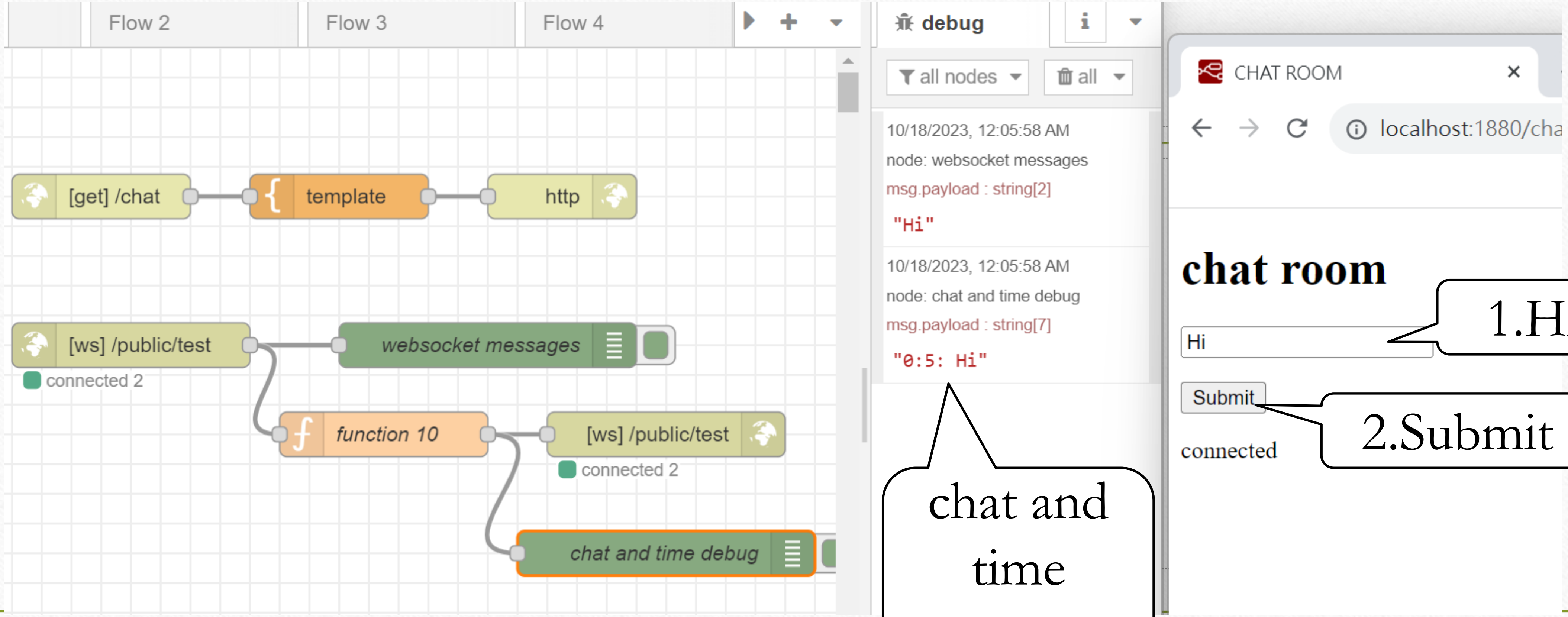


Output: msg. payload

To:
☑ debug window
□ system console
□ node status (32 characters)

Name: chat and time debug

Step 14: Send a chat message



```
10/18/2023, 12:05:58 AM
node: websocket messages
msg.payload : string[2]
 "Hi"

10/18/2023, 12:05:58 AM
node: chat and time debug
msg.payload : string[7]
 "0:5: Hi"
```

chat room

Hi

Submit

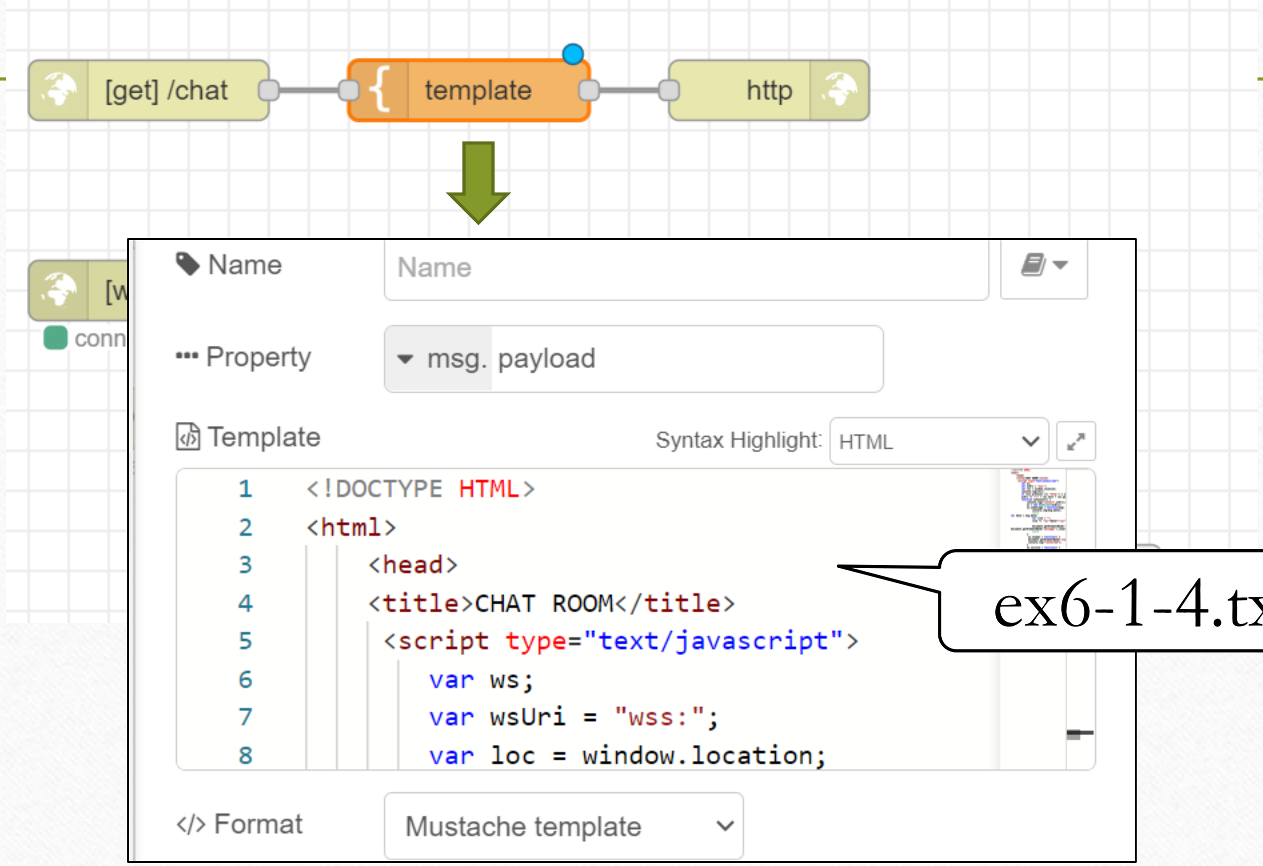
connected

1.Hi

2.Submit

chat and time message

Step 15: Modify HTML



Name: Name

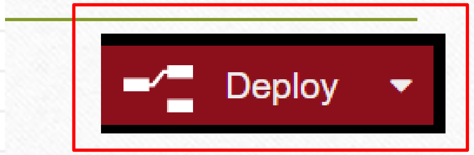
Property: msg. payload

Template — Syntax Highlight: HTML

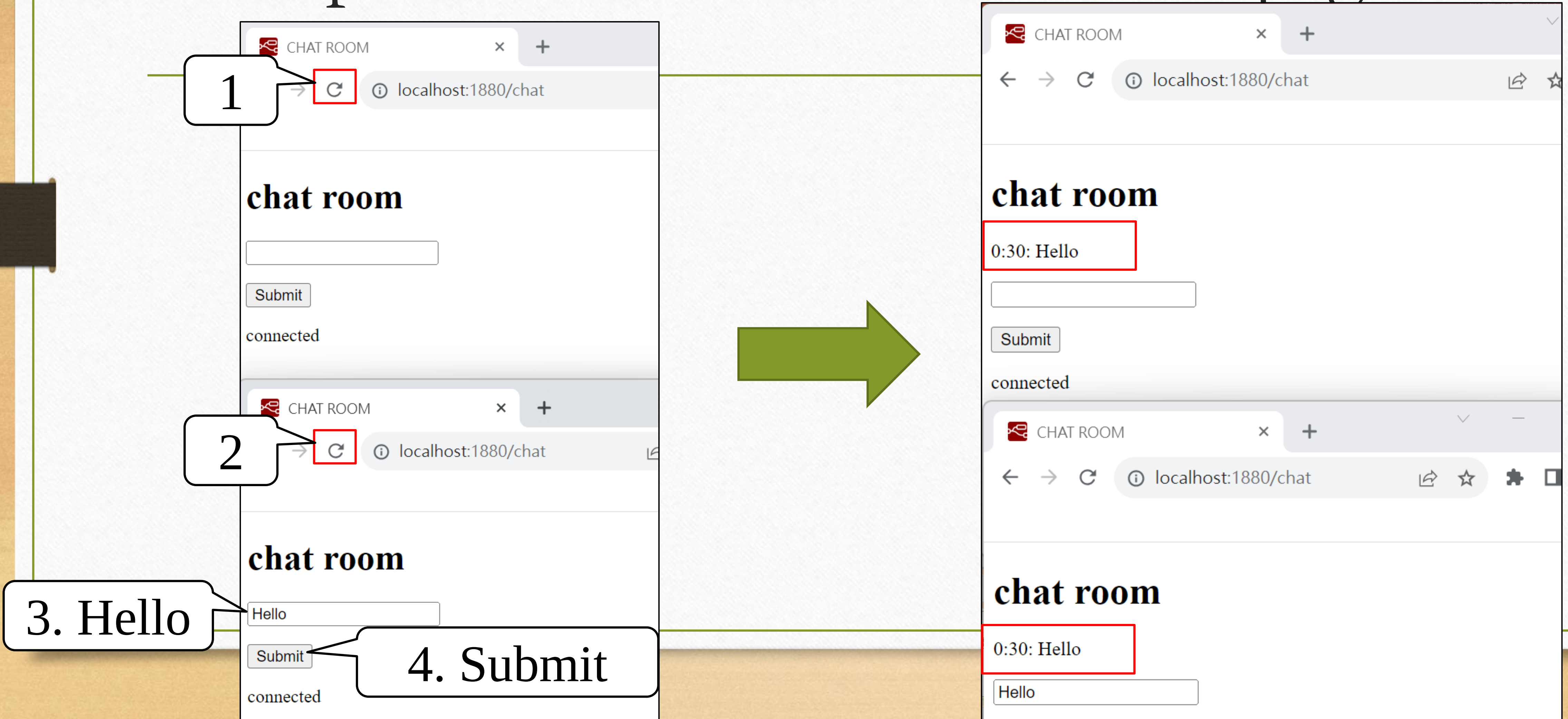
```
<!DOCTYPE HTML>
<html>
    <head>
    <title>CHAT ROOM</title>
     <script type="text/javascript">
        var ws;
        var wsUri = "wss:";
        var loc = window.location;
```

Format: Mustache template

ex6-1-4.txt



Step 16: Refresh chat room web pages



chat room

Hello

connected

chat room

0:30: Hello

connected

chat room

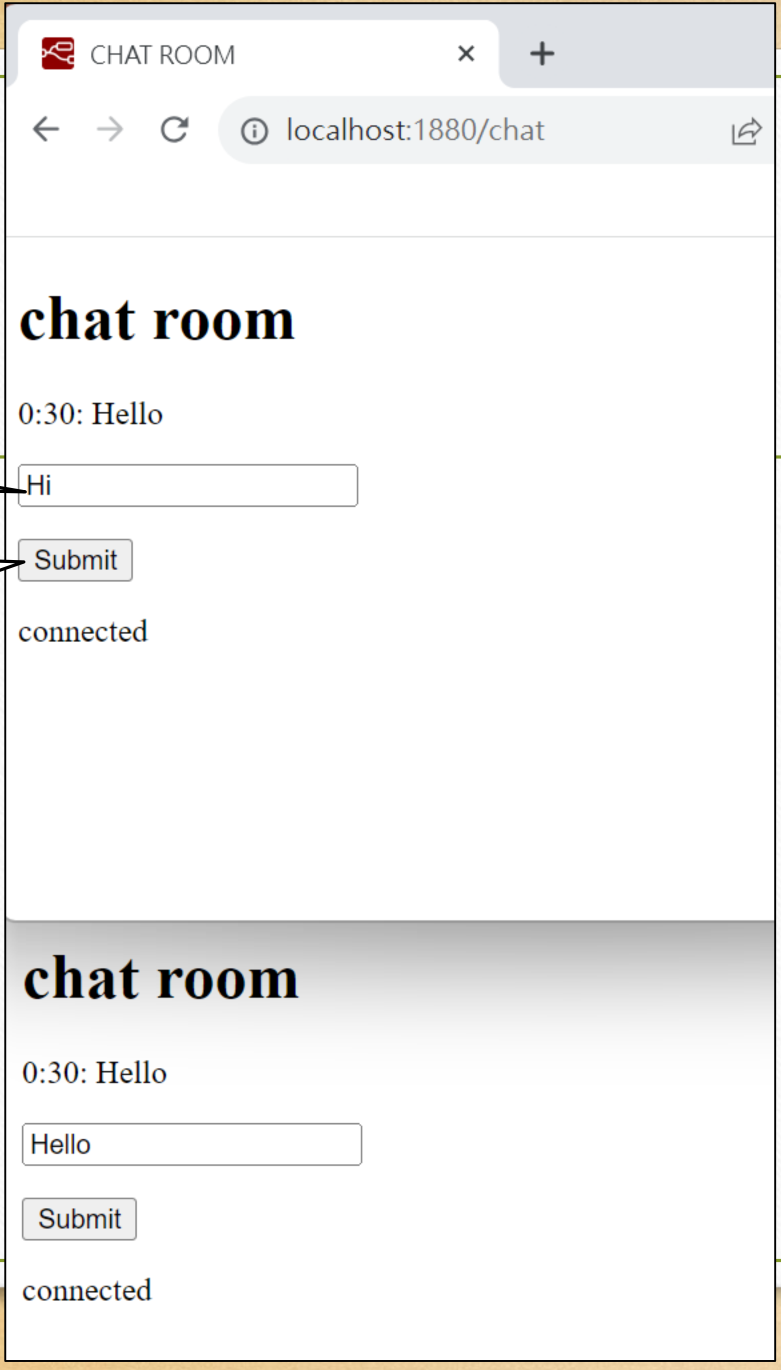
0:30: Hello

Hello

Step 17

1. Hi

2. Submit



chat room

0:30: Hello

Hi

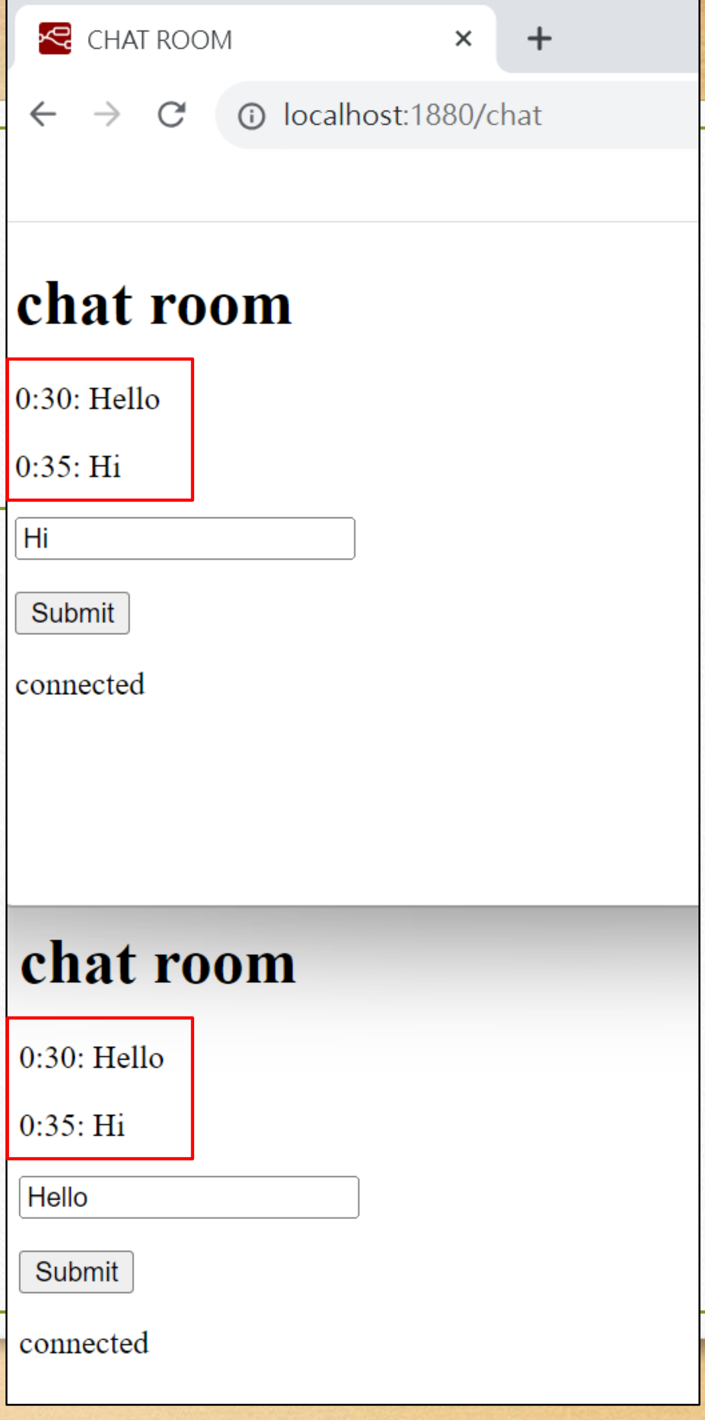
connected

chat room

0:30: Hello

Hello

connected



chat room

0:30: Hello

0:35: Hi

Hi

connected

chat room

0:30: Hello

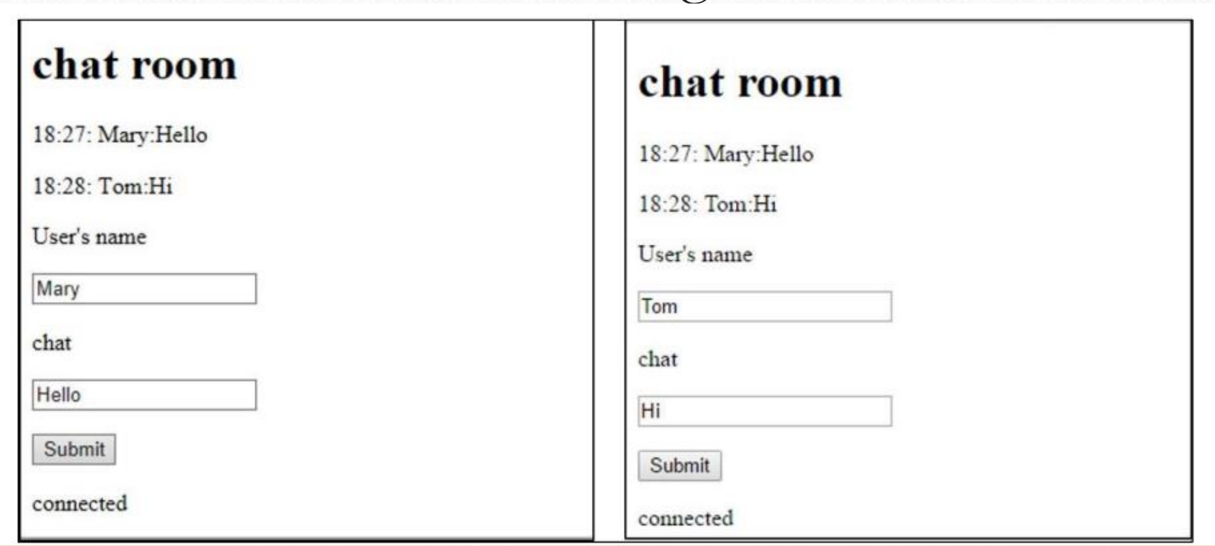
0:35: Hi

Hello

connected

Homework 6-1

- 請加上一個可以輸入用戶端代號的文字表單，讓聊天室網頁呈現進行聊天者的代號、聊天內容與聊天時間，如下圖所示。 Please add another input for user’s name. When you receive or send a message, show the time, the user’s name and the chat message



chat room

18:27: Mary:Hello

18:28: Tom:Hi

User's name

Mary

chat

Hello

Submit

connected

chat room

18:27: Mary:Hello

18:28: Tom:Hi

User's name

Tom

chat

Hi

Submit

connected