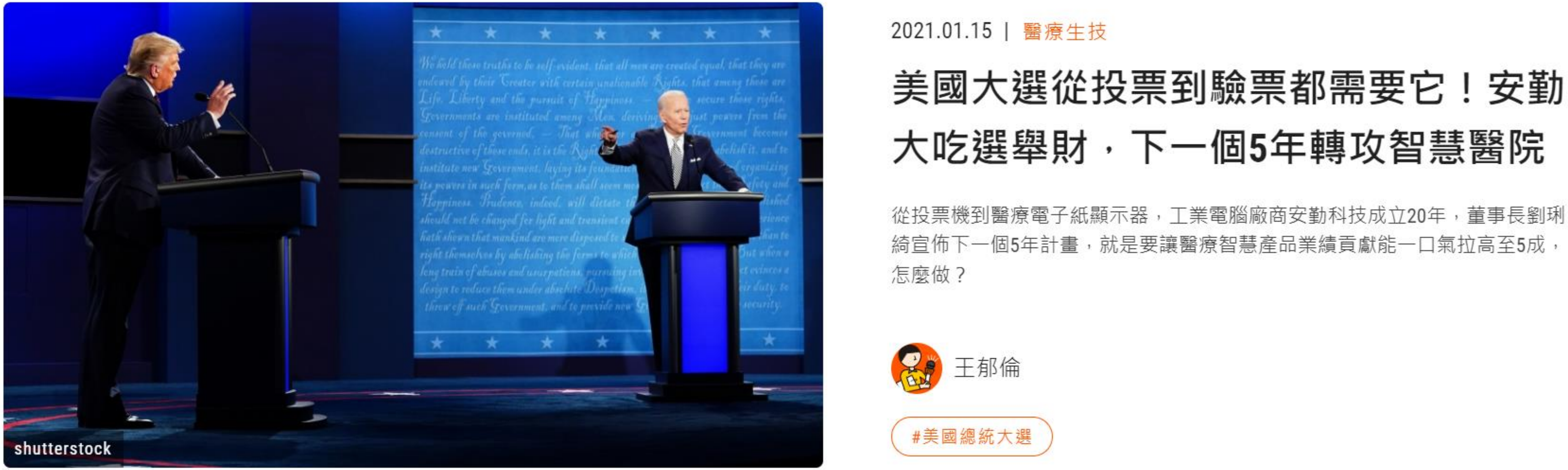
物聯網實務

10_19

投票機



2021.01.15 | 醫療生技

美國大選從投票到驗票都需要它！安勤大吃選舉財，下一個5年轉攻智慧醫院

從投票機到醫療電子紙顯示器，工業電腦廠商安勤科技成立20年，董事長劉琍綺宣佈下一個5年計畫，就是要讓醫療智慧產品業績貢獻能一口氣拉高至5成，怎麼做？

王郁倫

#美國總統大選

https://www.bnext.com.tw/article/60973/avalue-2021-medical-product-election-machine

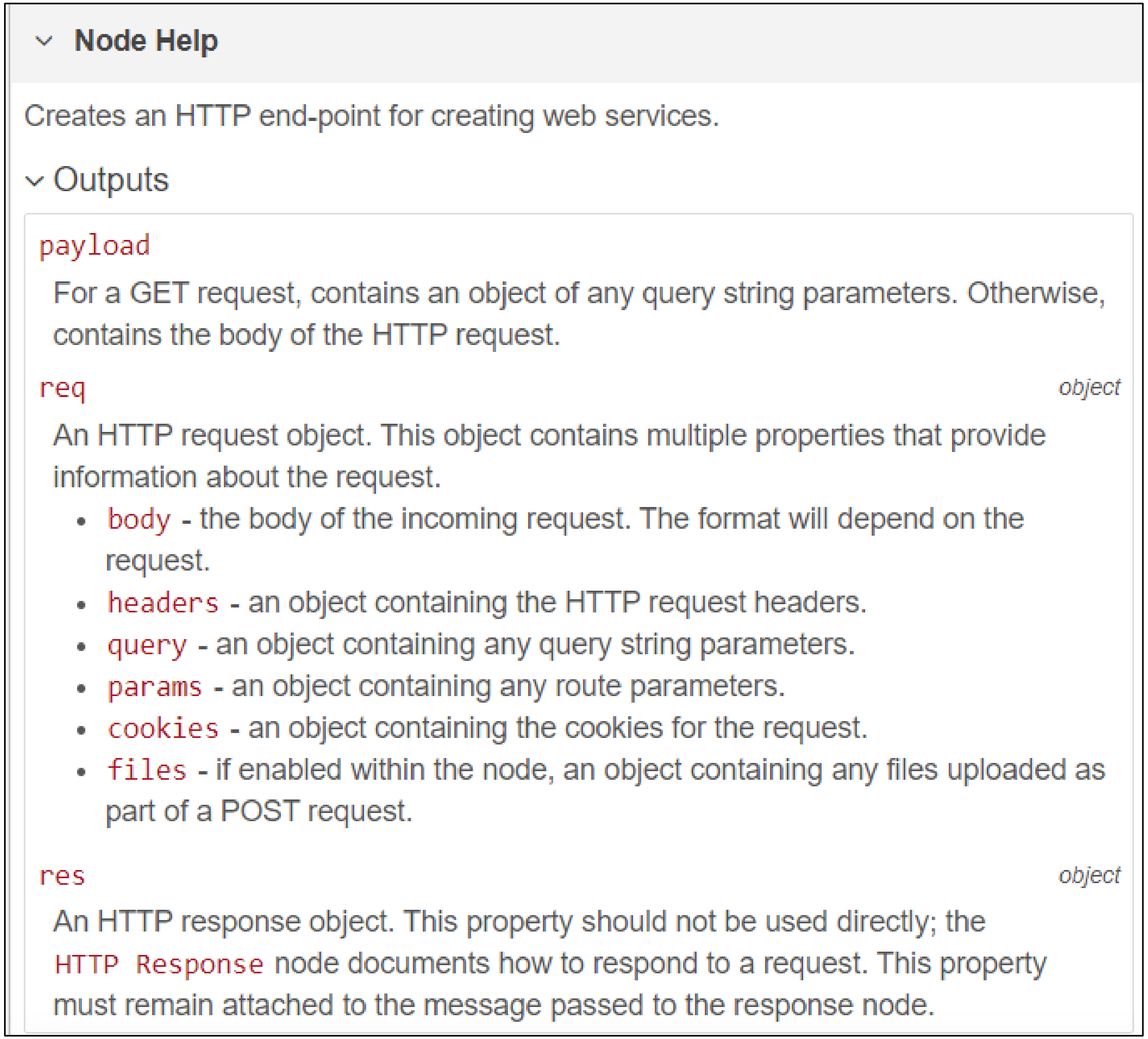
HTTP GET Method

- GET is used to request data from a specified resource.
- Note that the query string (name/value pairs) is sent in the URL of a GET request:

- **Some notes on GET requests:**
- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests are only used to request data (not modify)

https://www.w3schools.com/tags/ref_httpmethods.asp

http in node



Node Help

Creates an HTTP end-point for creating web services.

Outputs

`payload`

For a GET request, contains an object of any query string parameters. Otherwise, contains the body of the HTTP request.

`req` *object*

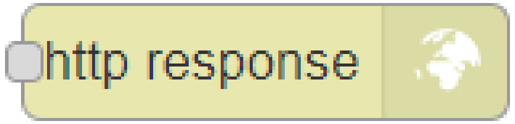
An HTTP request object. This object contains multiple properties that provide information about the request.

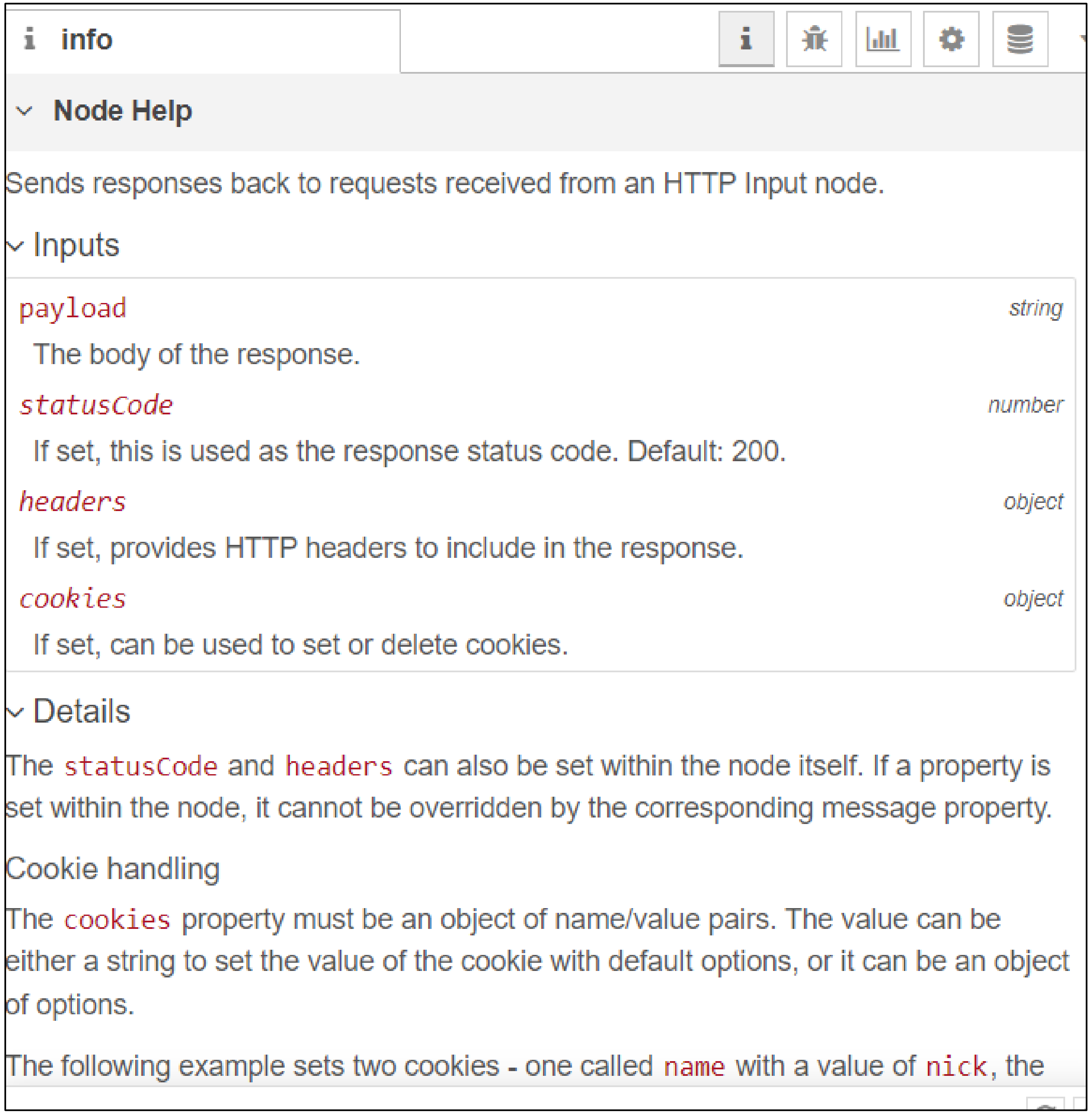
- `body` - the body of the incoming request. The format will depend on the request.
- `headers` - an object containing the HTTP request headers.
- `query` - an object containing any query string parameters.
- `params` - an object containing any route parameters.
- `cookies` - an object containing the cookies for the request.
- `files` - if enabled within the node, an object containing any files uploaded as part of a POST request.

`res` *object*

An HTTP response object. This property should not be used directly; the `HTTP Response` node documents how to respond to a request. This property must remain attached to the message passed to the response node.

http response





Node Help

Sends responses back to requests received from an HTTP Input node.

Inputs

payload *string*

The body of the response.

statusCode *number*

If set, this is used as the response status code. Default: 200.

headers *object*

If set, provides HTTP headers to include in the response.

cookies *object*

If set, can be used to set or delete cookies.

Details

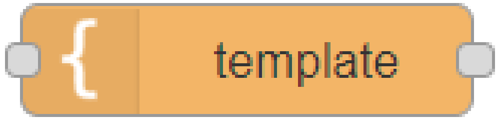
The statusCode and headers can also be set within the node itself. If a property is set within the node, it cannot be overridden by the corresponding message property.

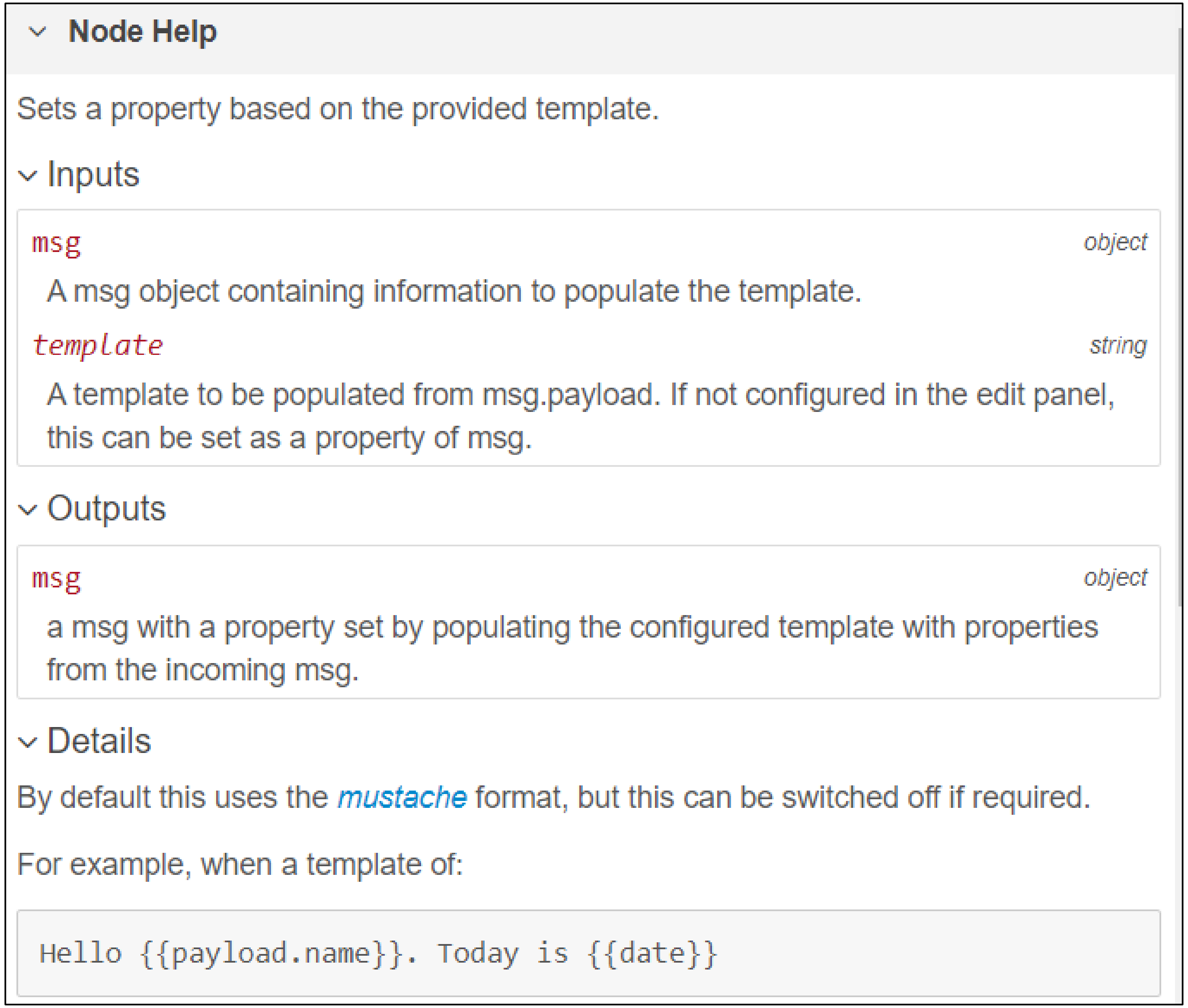
Cookie handling

The cookies property must be an object of name/value pairs. The value can be either a string to set the value of the cookie with default options, or it can be an object of options.

The following example sets two cookies - one called name with a value of nick, the

template node





Node Help

Sets a property based on the provided template.

Inputs

`msg` *object*

A msg object containing information to populate the template.

`template` *string*

A template to be populated from msg.payload. If not configured in the edit panel, this can be set as a property of msg.

Outputs

`msg` *object*

a msg with a property set by populating the configured template with properties from the incoming msg.

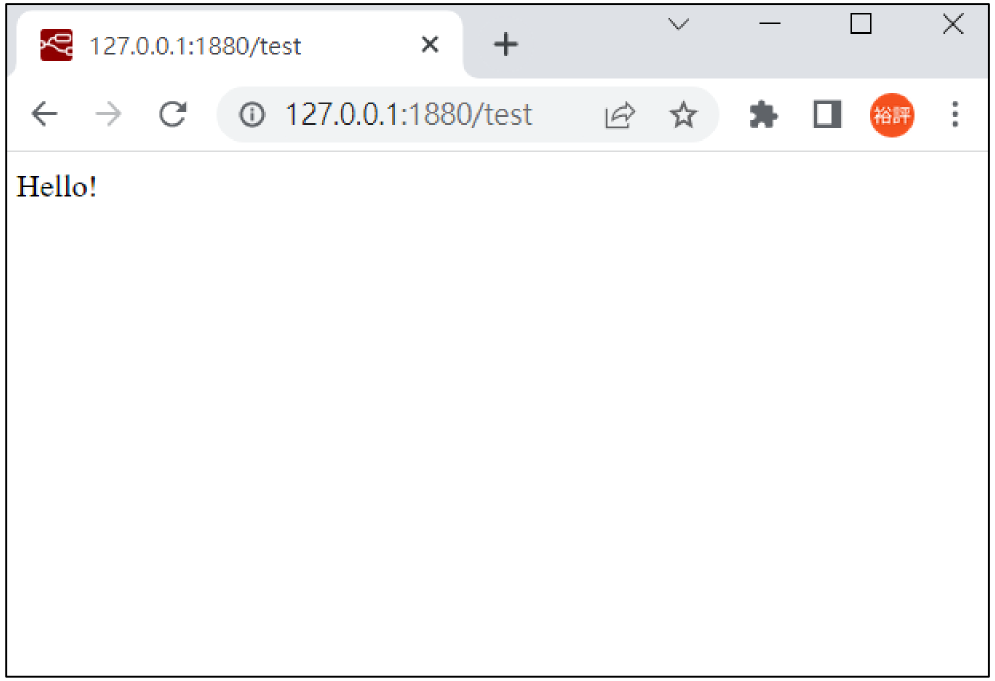
Details

By default this uses the *mustache* format, but this can be switched off if required.

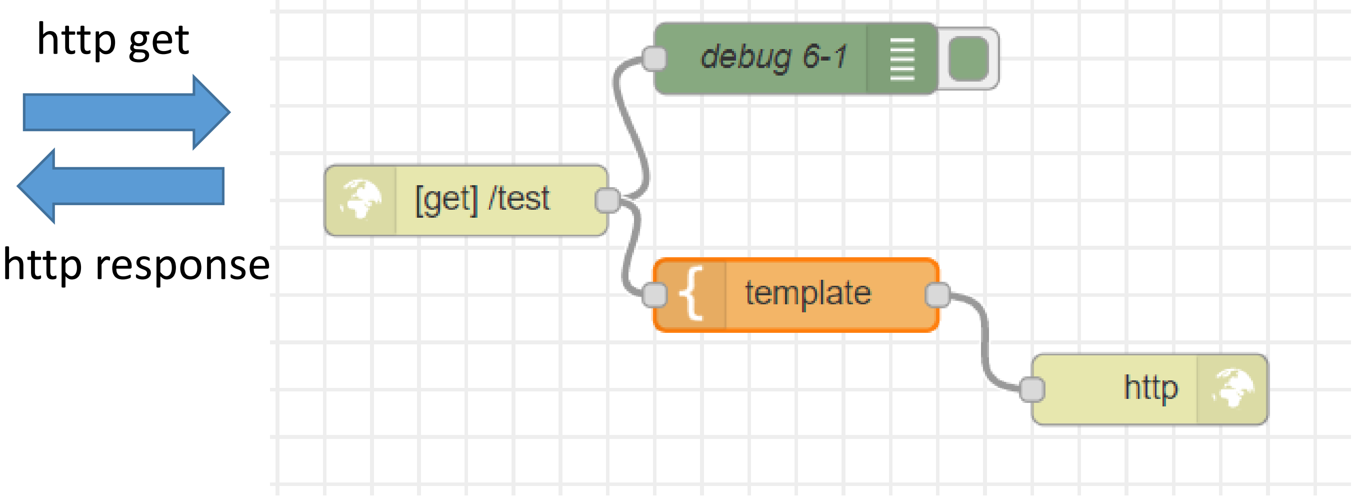
For example, when a template of:

```
Hello {{payload.name}}. Today is {{date}}
```

Exercise 6-1



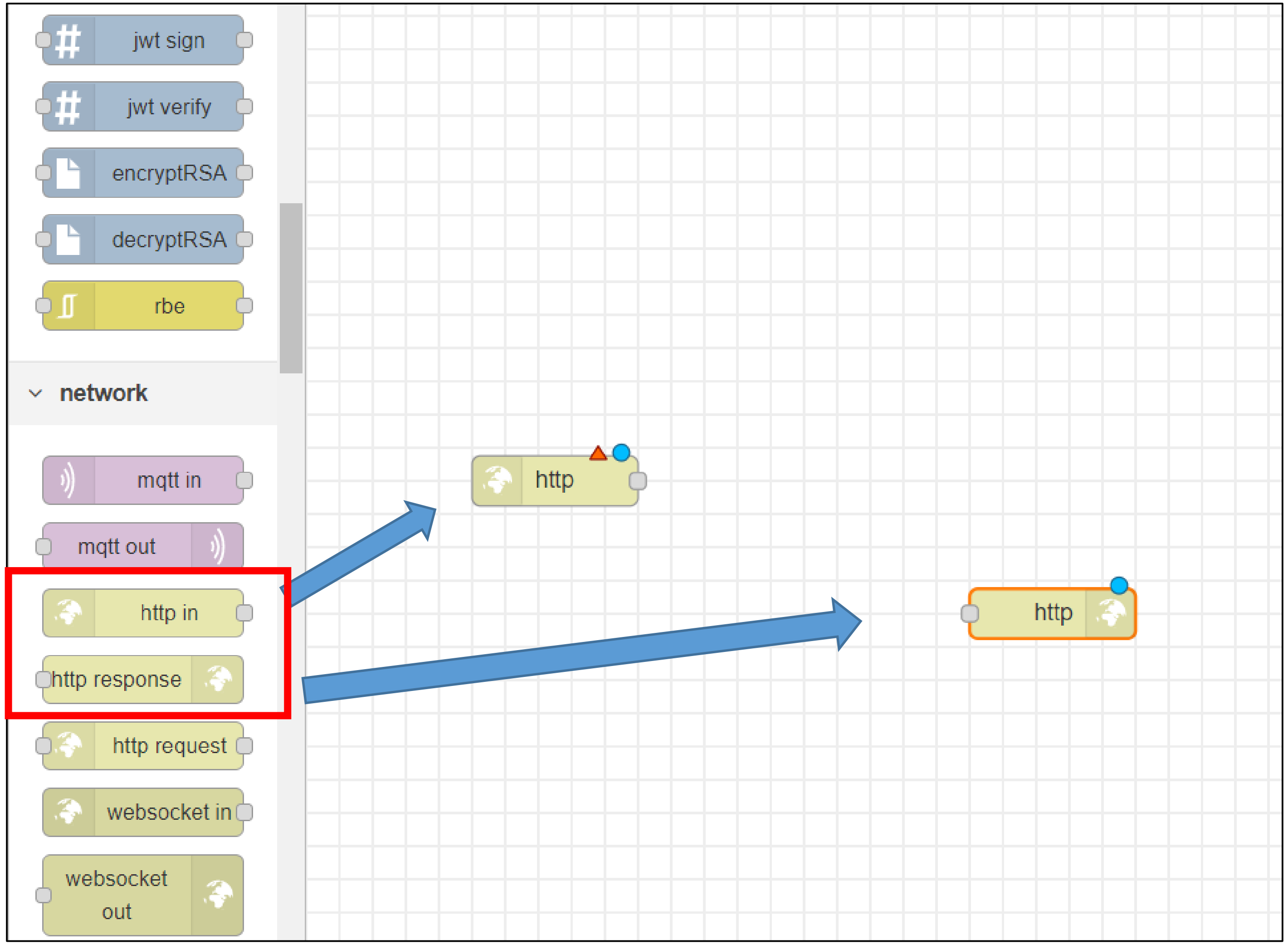
Hello!


http get

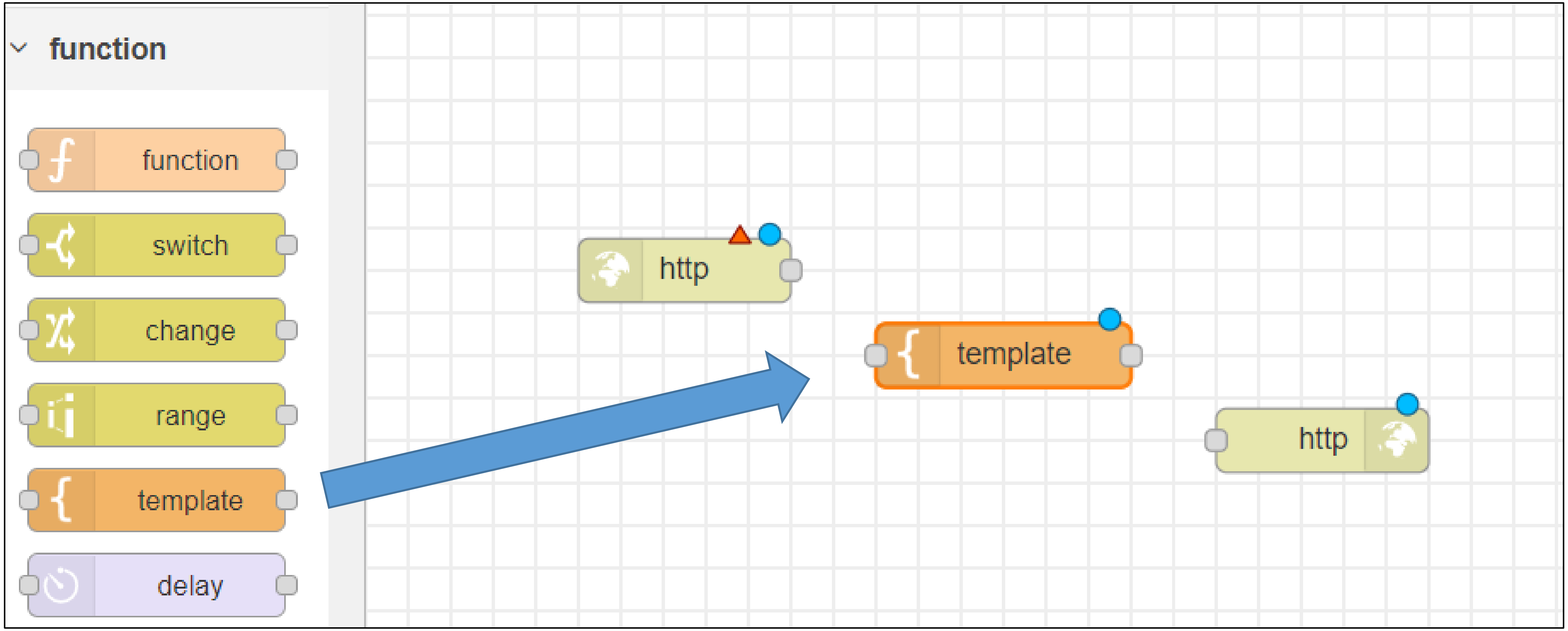
http response

Web server

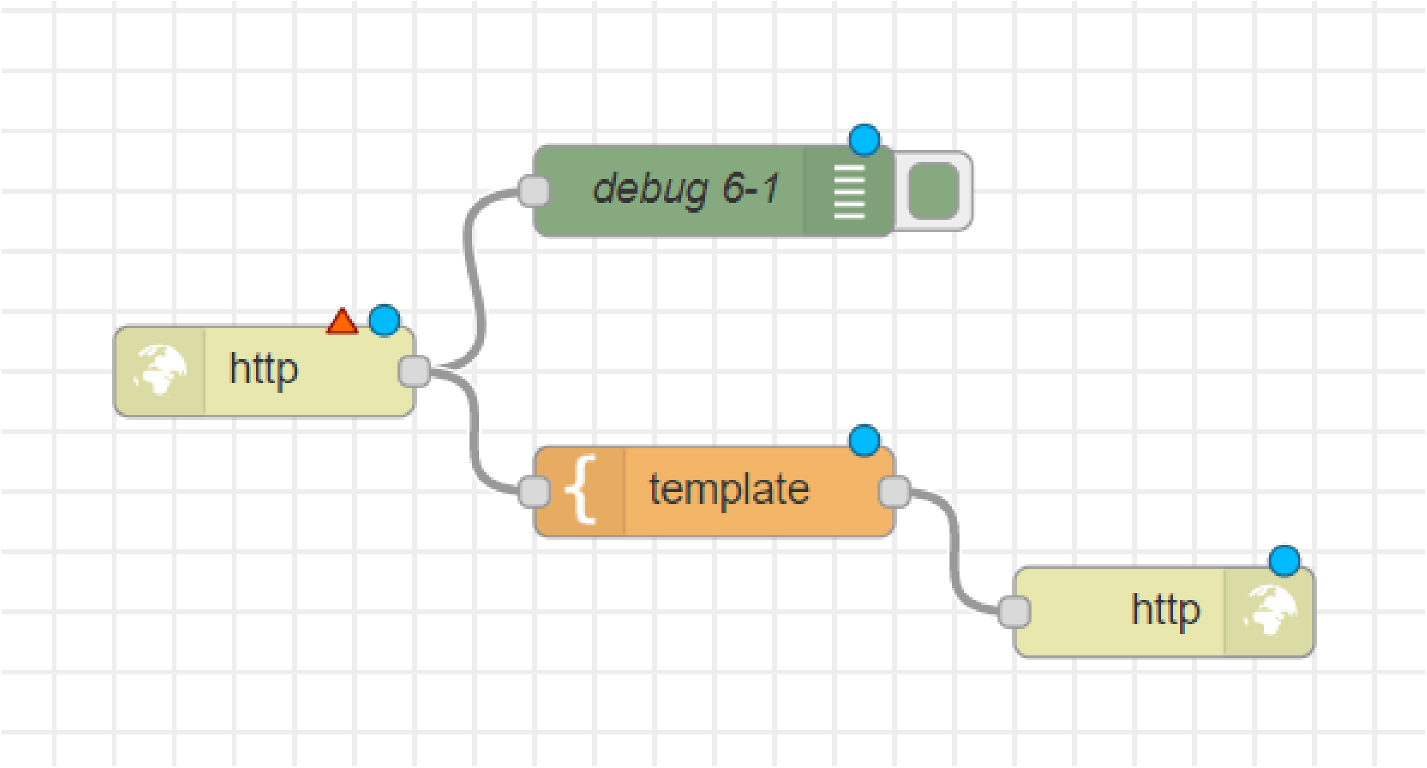
Add http in & http response

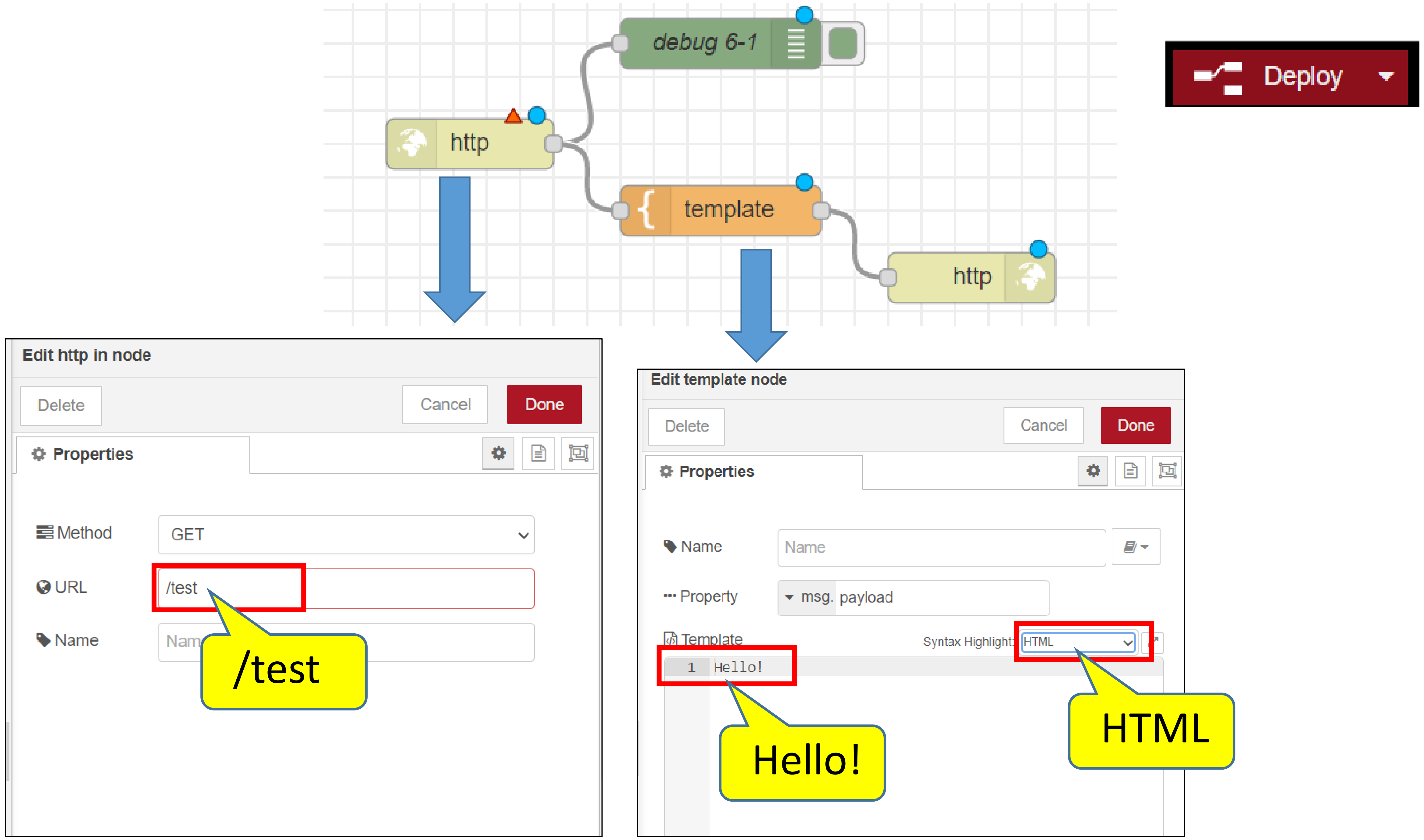


Add template



Add debug





Edit http in node

Delete · Cancel · Done

Properties

Method: GET

URL: /test

Name: Name

/test

Edit template node

Delete · Cancel · Done

Properties

Name: Name

Property: msg. payload

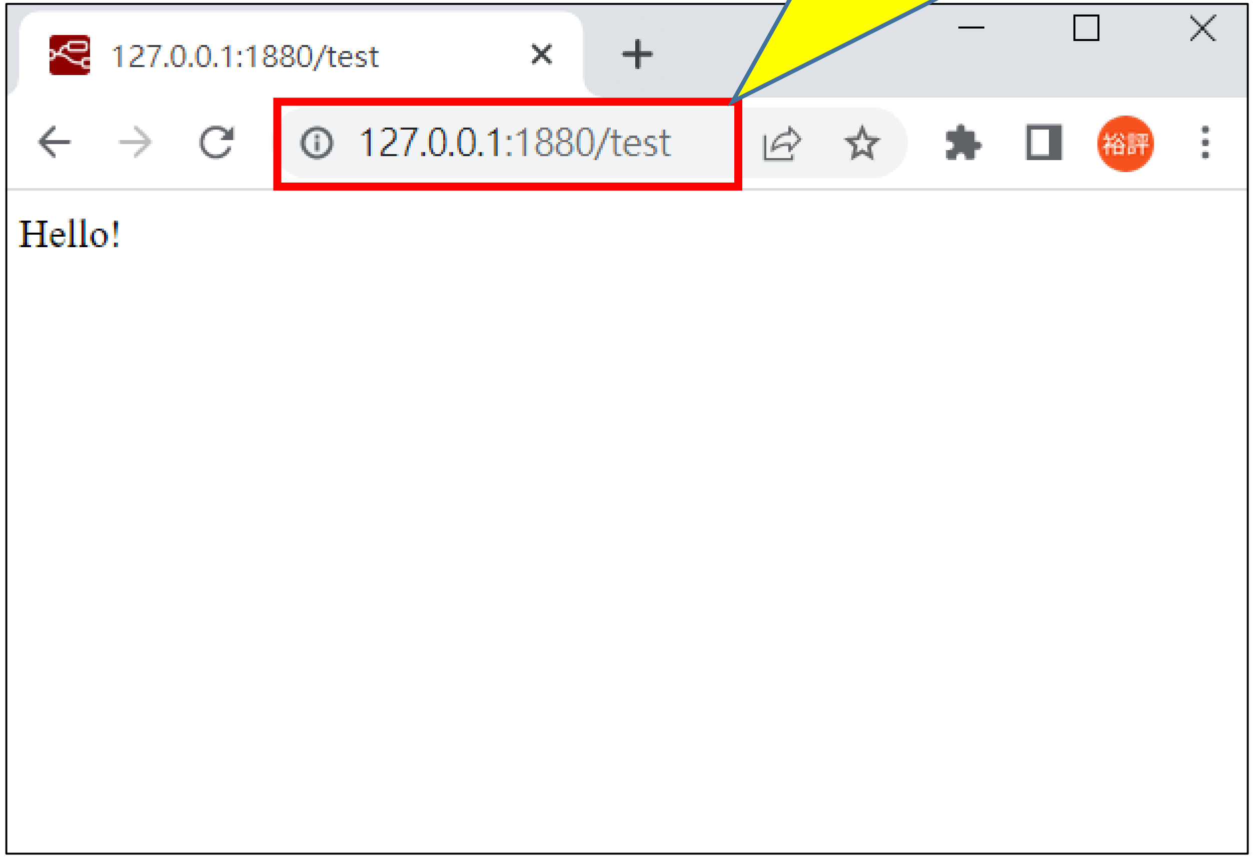
Template · Syntax Highlight: HTML

```
Hello!
```

Hello!

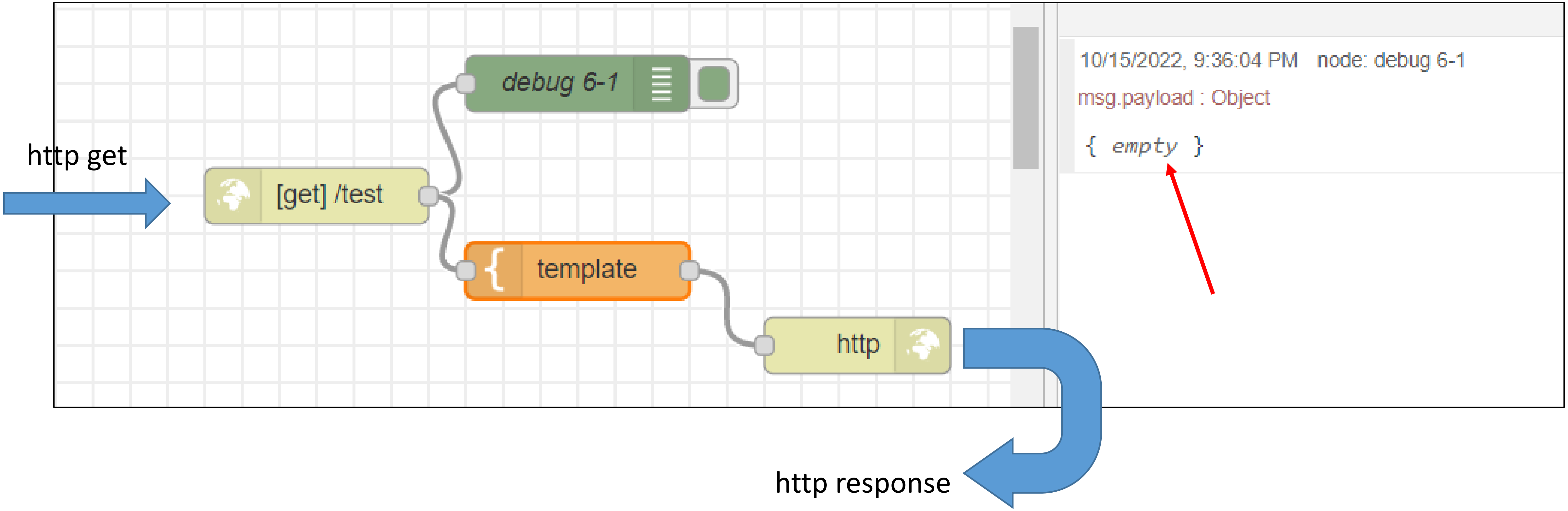
HTML

127.0.0.1:1880/test



Hello!

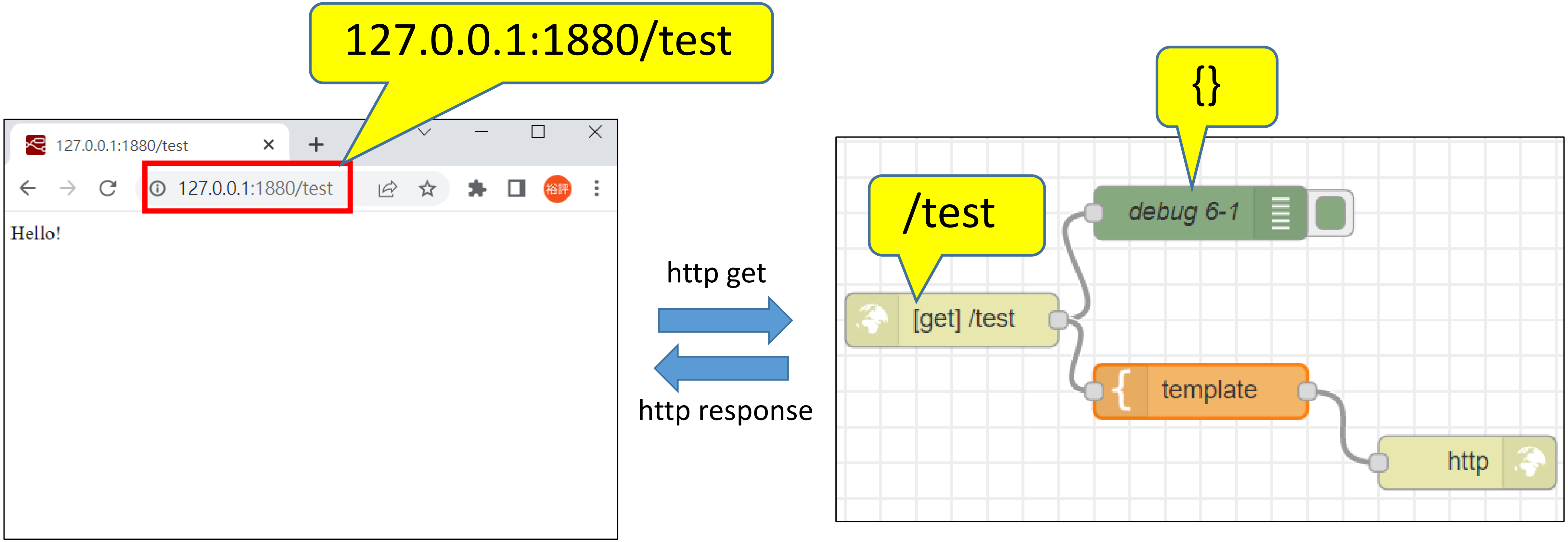
Node-RED



http get

```
10/15/2022, 9:36:04 PM   node: debug 6-1
msg.payload : Object
 { empty }
```

http response



127.0.0.1:1880/test

Hello!

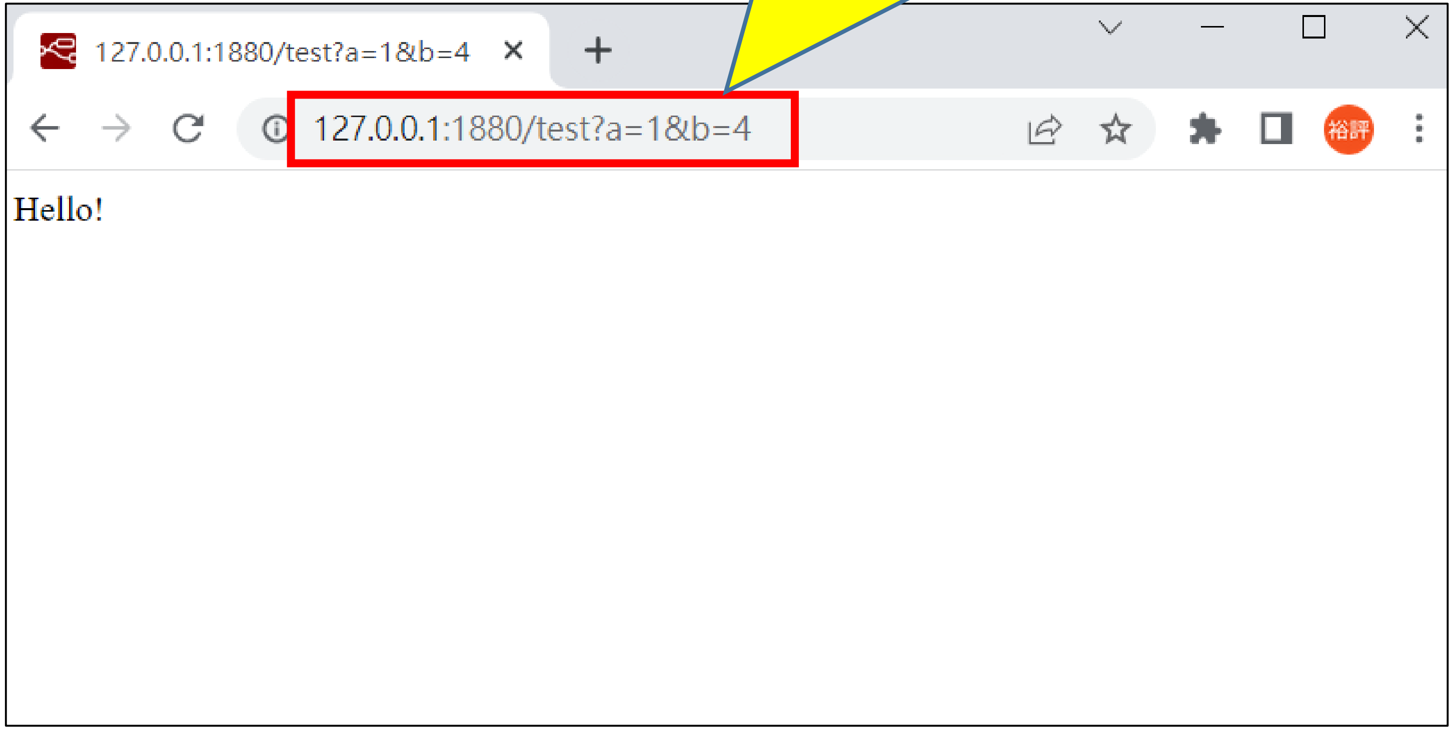
http get

http response

{}

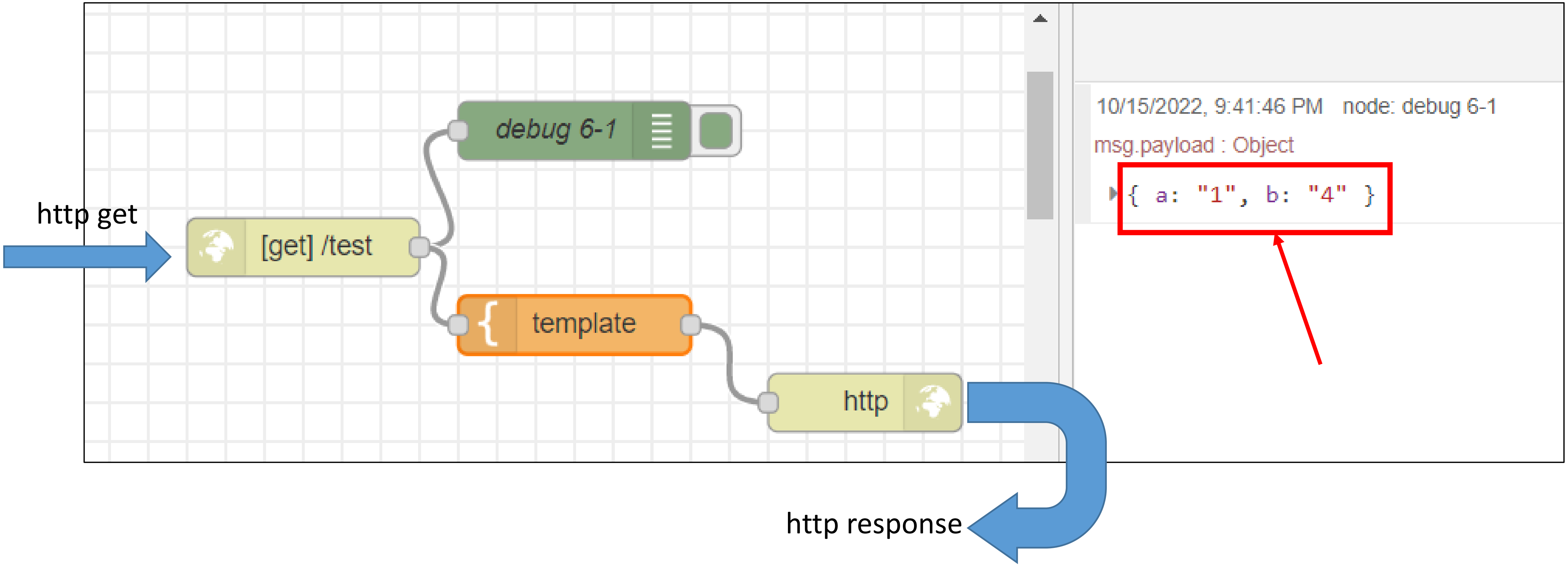
/test

127.0.0.1:1880/test?a=1&b=4



Hello!

Node-RED



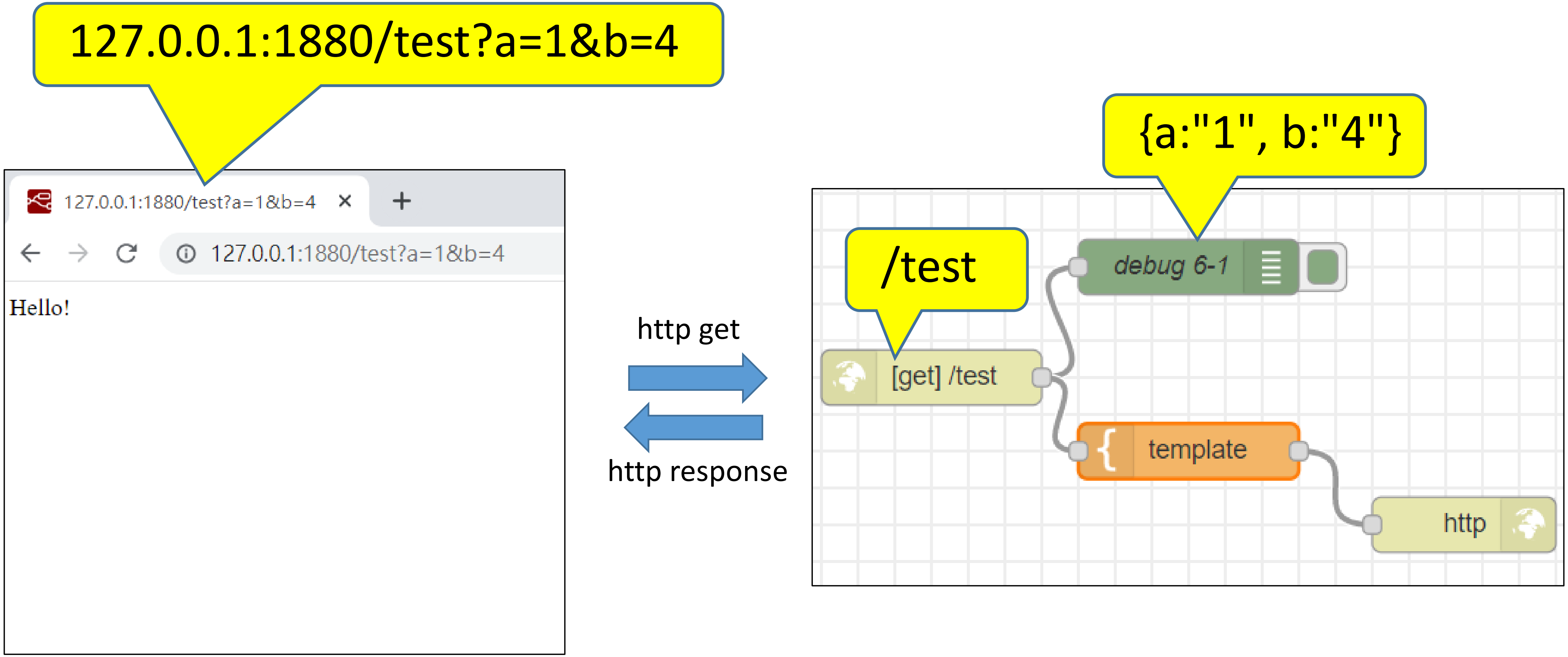
http get

10/15/2022, 9:41:46 PM node: debug 6-1

msg.payload : Object

```
{ a: "1", b: "4" }
```

http response



127.0.0.1:1880/test?a=1&b=4

127.0.0.1:1880/test?a=1&b=4

127.0.0.1:1880/test?a=1&b=4

Hello!

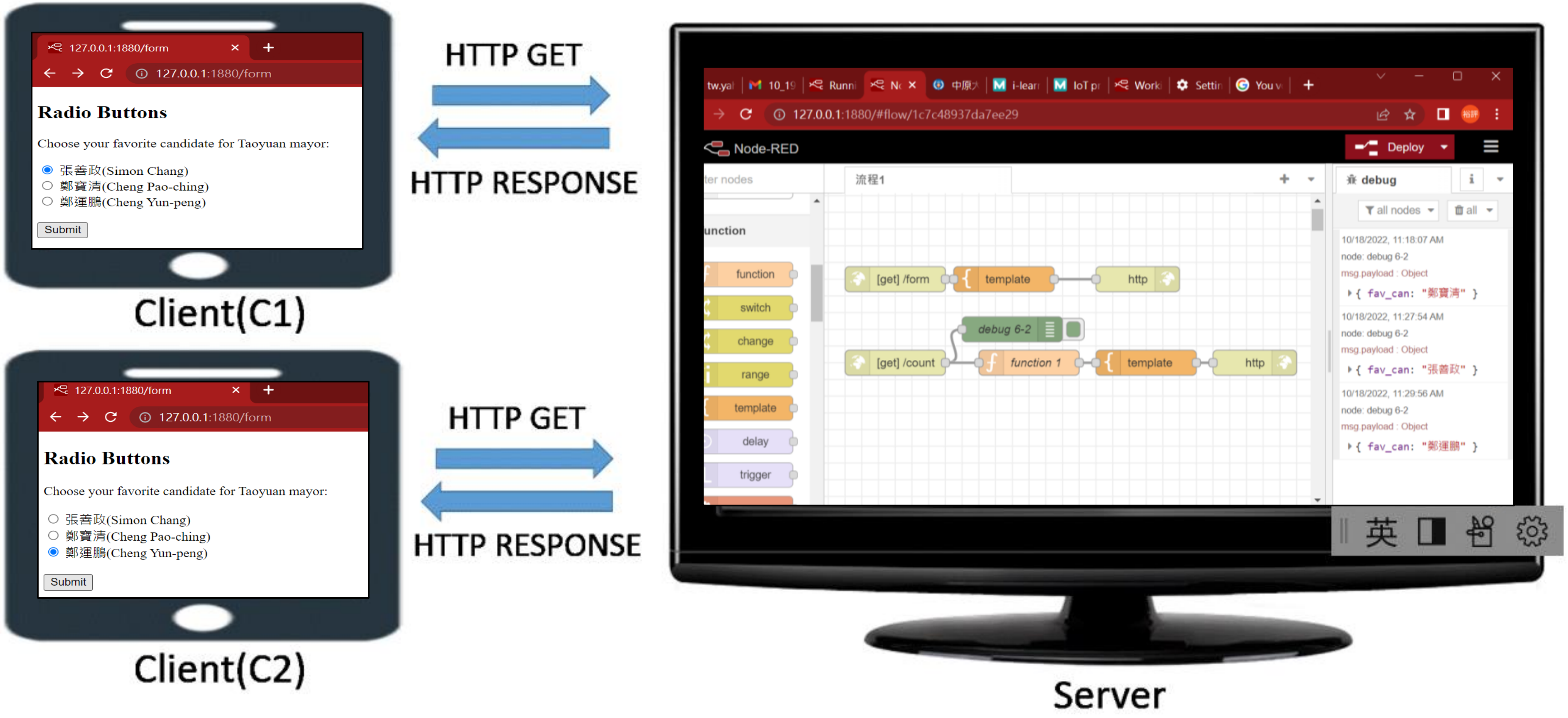
http get

http response

/test

{a:"1", b:"4"}

Exercise 6-2 Design of a voting system



127.0.0.1:1880/form

Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

● 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

Client(C1)

127.0.0.1:1880/form

Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
● 鄭運鵬(Cheng Yun-peng)

Submit

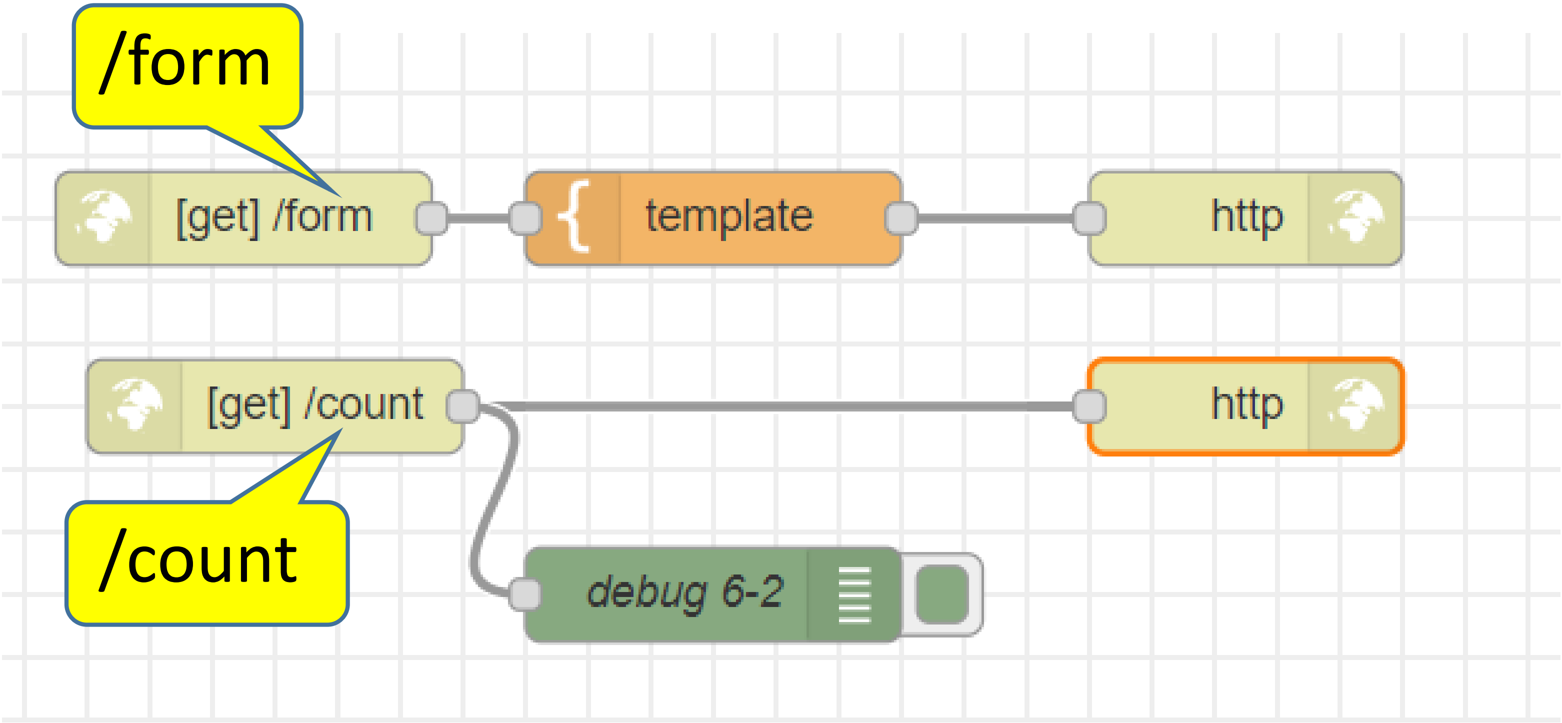
Client(C2)

127.0.0.1:1880/#flow/1c7c48937da7ee29

```
10/18/2022, 11:18:07 AM
node: debug 6-2
msg.payload : Object
▸{ fav_can: "鄭寶清" }
10/18/2022, 11:27:54 AM
node: debug 6-2
msg.payload : Object
▸{ fav_can: "張善政" }
10/18/2022, 11:29:56 AM
node: debug 6-2
msg.payload : Object
▸{ fav_can: "鄭運鵬" }
```

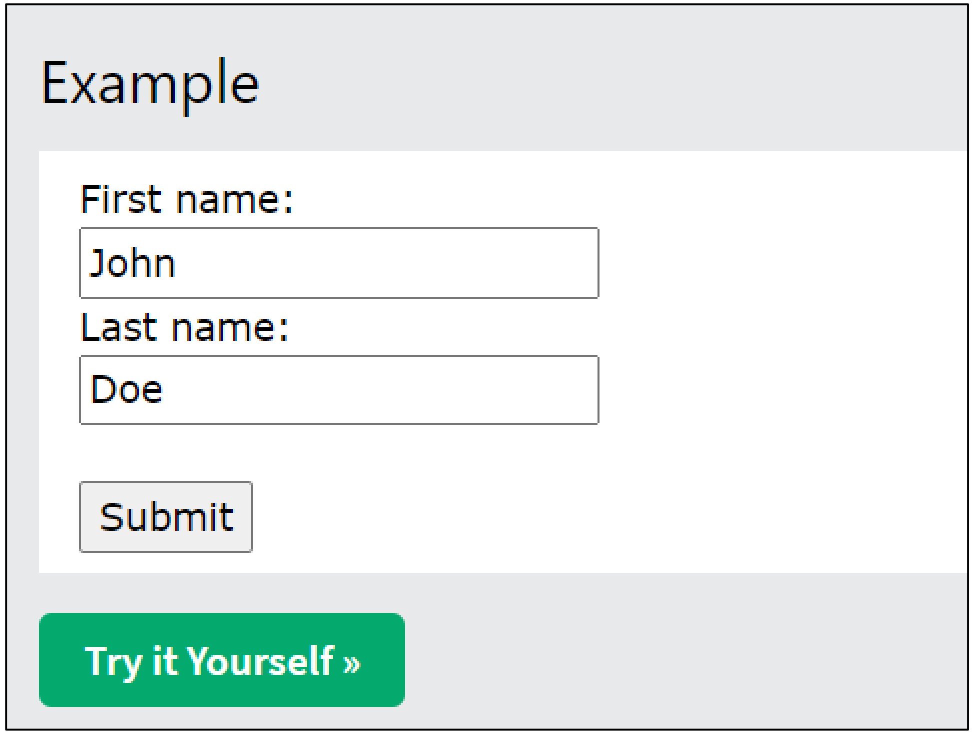
Server

Step 1



HTML Forms

- An HTML form is used to collect user input. The user input is most often sent to a server for processing.



Example

First name:

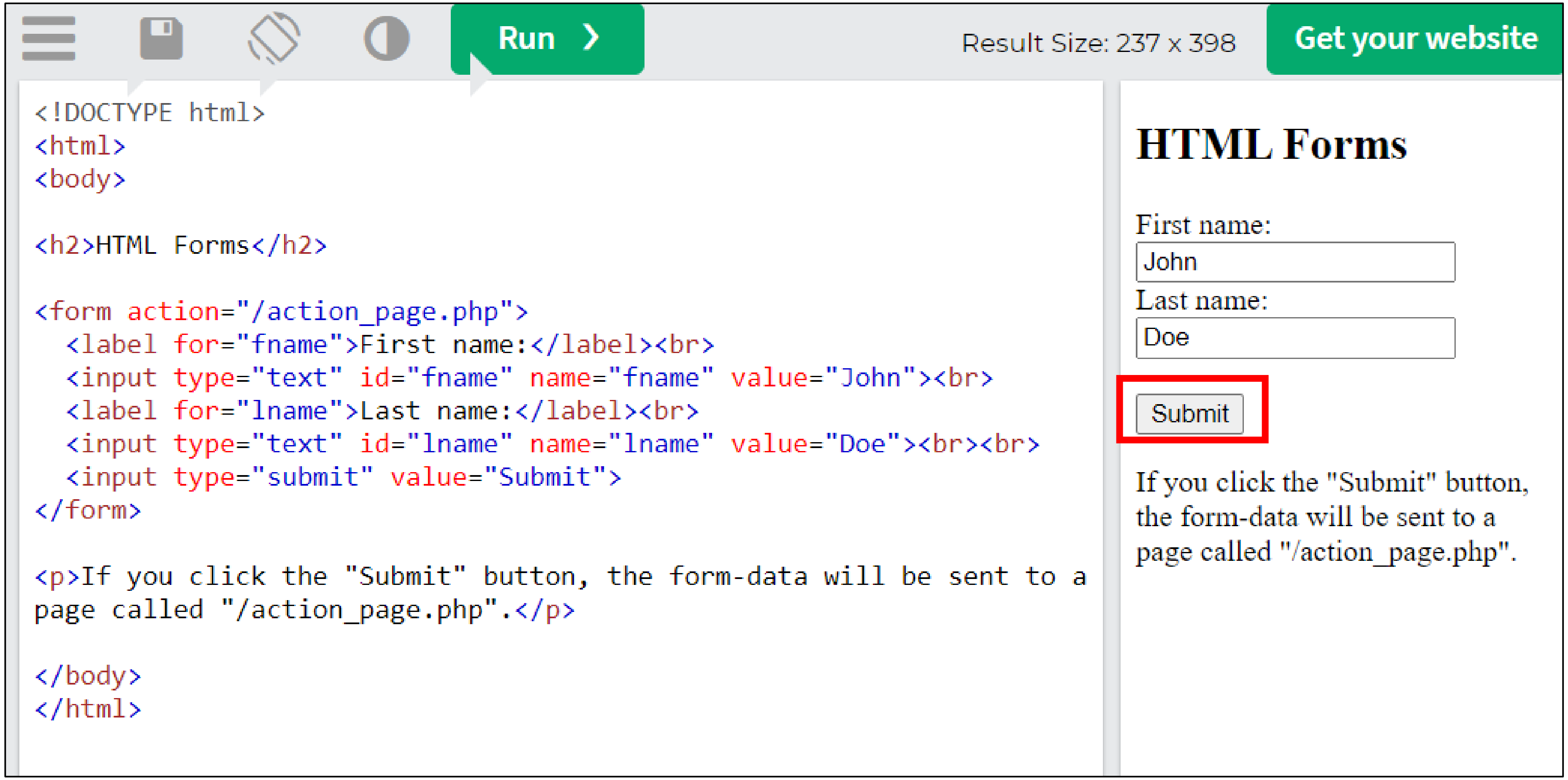
John

Last name:

Doe

https://www.w3schools.com/html/html_forms.asp

Try it yourself



```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>

<p>If you click the "Submit" button, the form-data will be sent to a
page called "/action_page.php".</p>

</body>
</html>
```

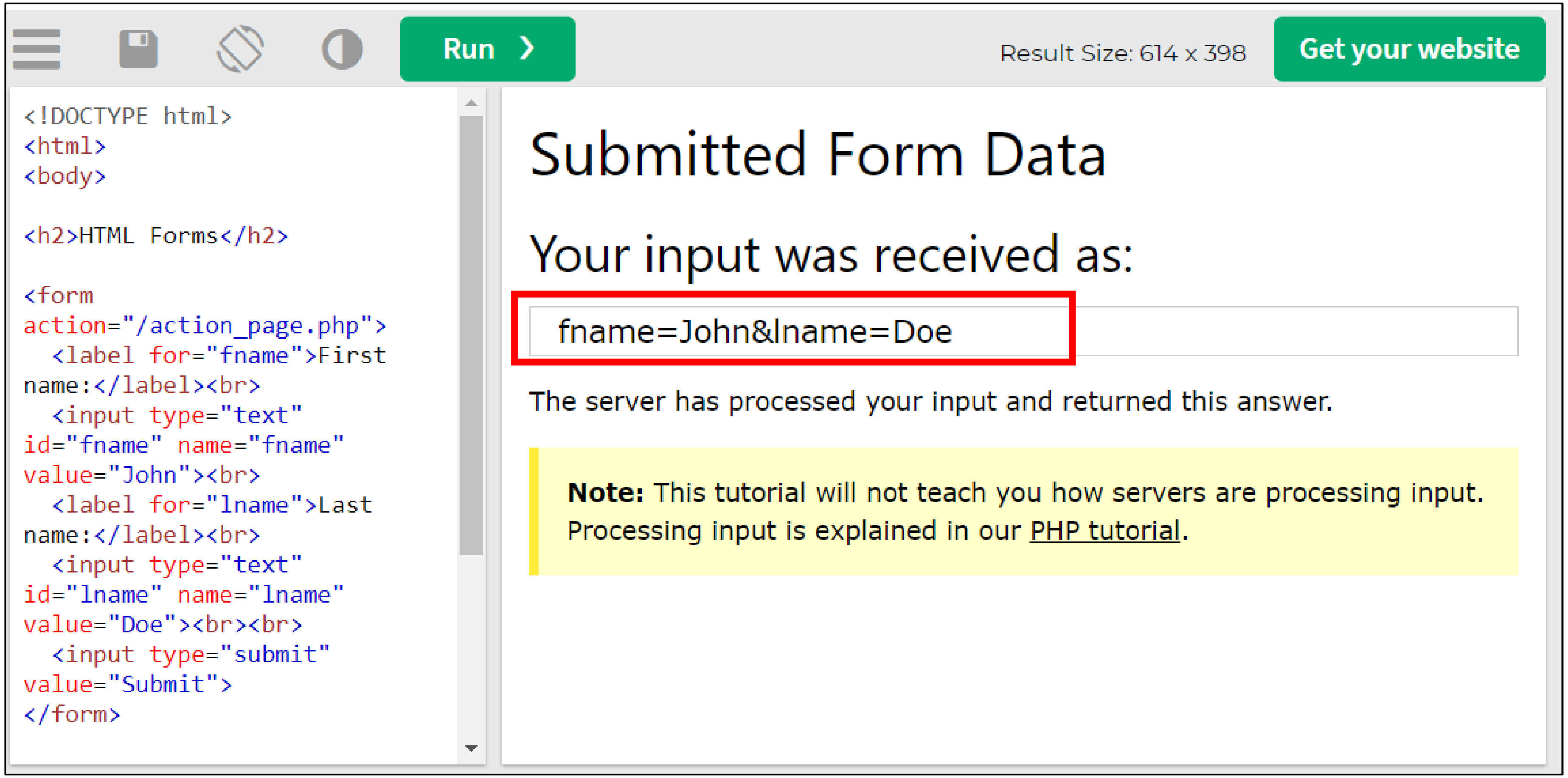
HTML Forms

First name:
John

Last name:
Doe

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_submit



```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Forms</h2>

<form
action="/action_page.php">
  <label for="fname">First
name:</label><br>
  <input type="text"
id="fname" name="fname"
value="John"><br>
  <label for="lname">Last
name:</label><br>
  <input type="text"
id="lname" name="lname"
value="Doe"><br><br>
  <input type="submit"
value="Submit">
</form>
```

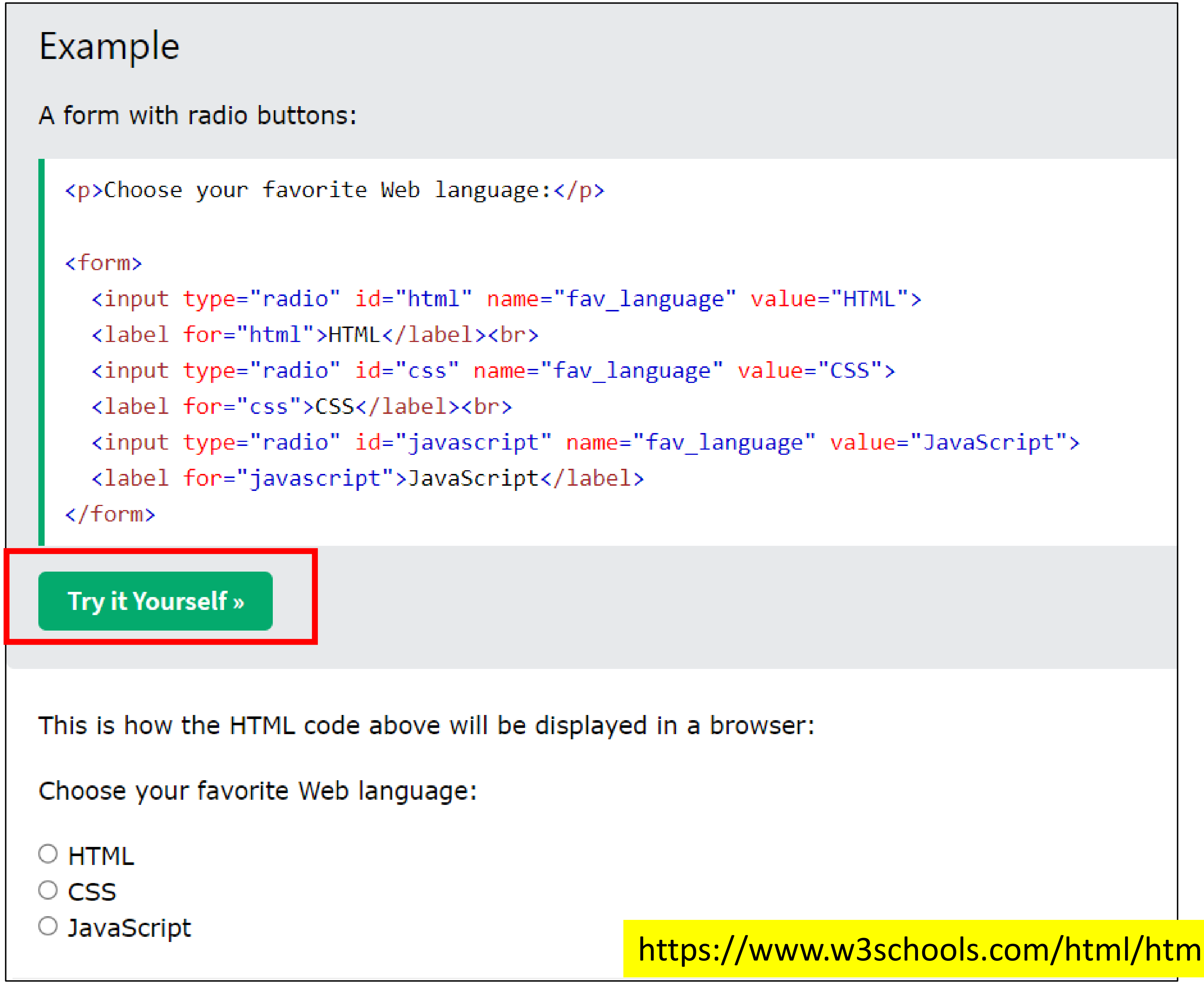
Submitted Form Data

Your input was received as:

```
fname=John&lname=Doe
```

The server has processed your input and returned this answer.

Note: This tutorial will not teach you how servers are processing input. Processing input is explained in our PHP tutorial.



Example

A form with radio buttons:

```
<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```

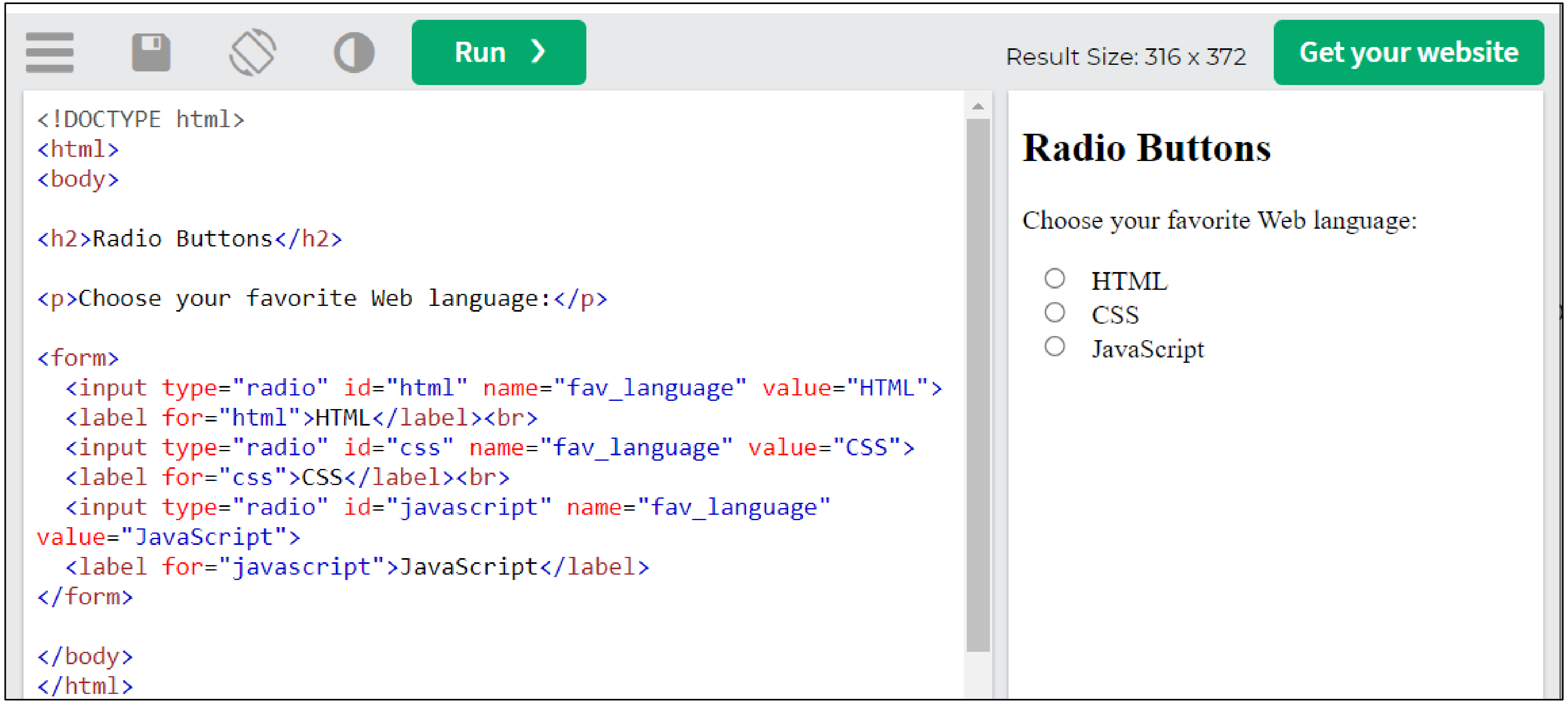
This is how the HTML code above will be displayed in a browser:

Choose your favorite Web language:

○ HTML
○ CSS
○ JavaScript

https://www.w3schools.com/html/html_forms.asp

Try it yourself



```
<!DOCTYPE html>
<html>
<body>

<h2>Radio Buttons</h2>

<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language"
value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>

</body>
</html>
```

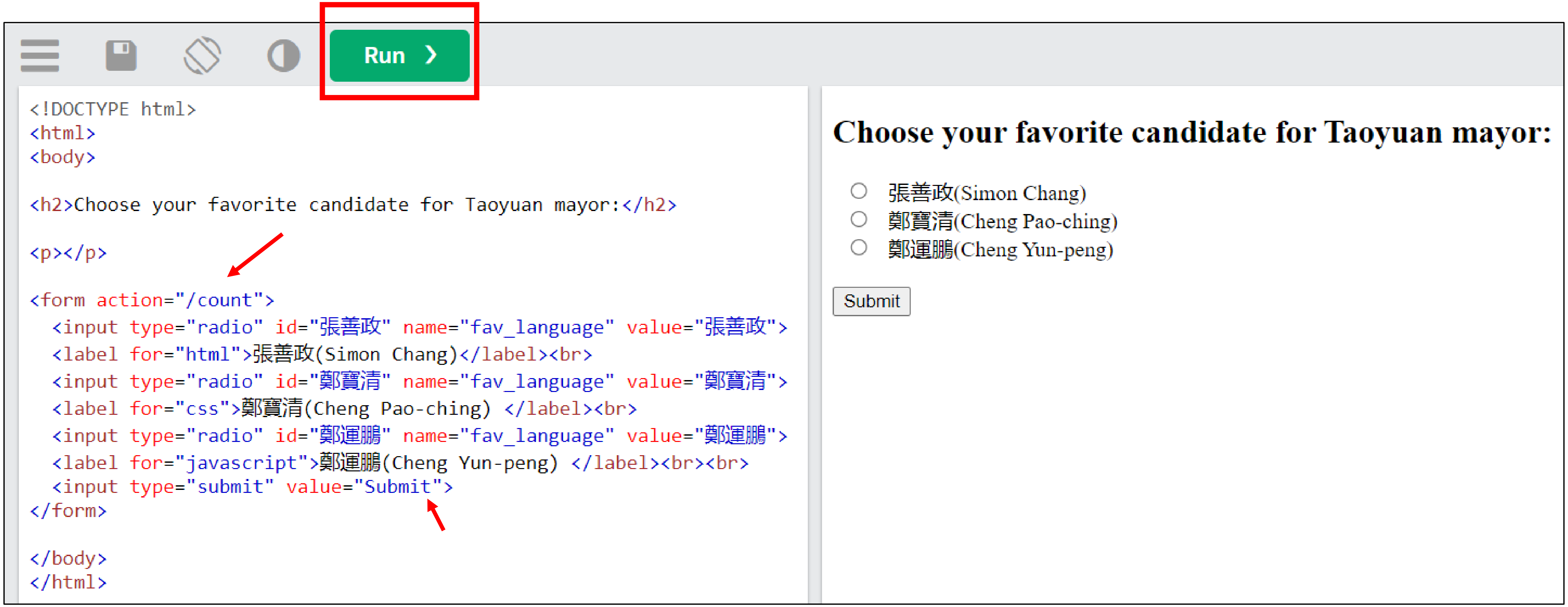
Radio Buttons

Choose your favorite Web language:

○ HTML
○ CSS
○ JavaScript

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_radio

Edit



```
<!DOCTYPE html>
<html>
<body>

<h2>Choose your favorite candidate for Taoyuan mayor:</h2>

<p></p>

<form action="/count">
  <input type="radio" id="張善政" name="fav_language" value="張善政">
  <label for="html">張善政(Simon Chang)</label><br>
  <input type="radio" id="鄭寶清" name="fav_language" value="鄭寶清">
  <label for="css">鄭寶清(Cheng Pao-ching) </label><br>
  <input type="radio" id="鄭運鵬" name="fav_language" value="鄭運鵬">
  <label for="javascript">鄭運鵬(Cheng Yun-peng) </label><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

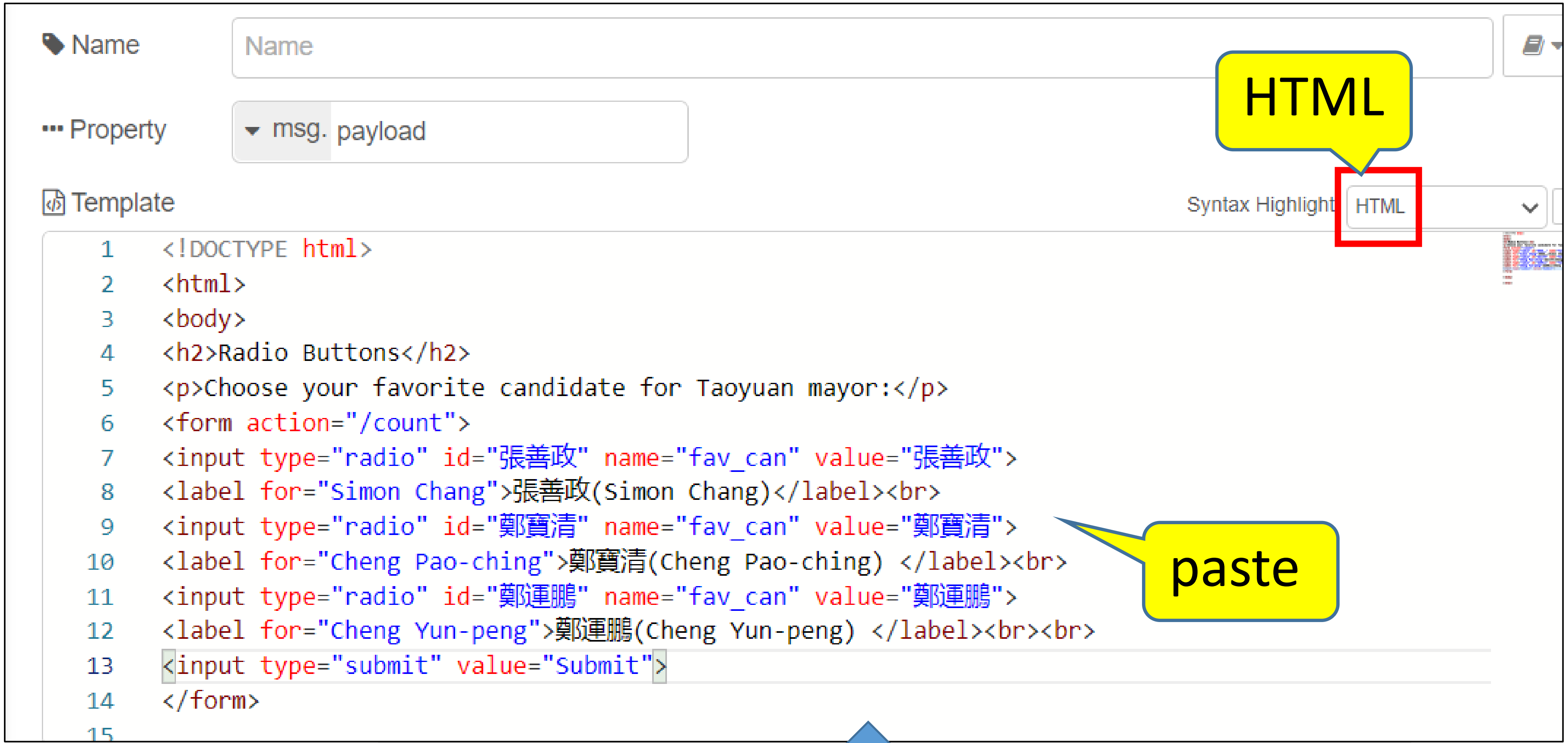
Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

```
<!DOCTYPE html>
<html>
<body>
<h2>Radio Buttons</h2>
<p>Choose your favorite candidate for Taoyuan mayor:</p>
<form action="/count">
<input type="radio" id="張善政" name="fav_can" value="張善政">
<label for="Simon Chang">張善政(Simon Chang)</label><br>
<input type="radio" id="鄭寶清" name="fav_can" value="鄭寶清">
<label for="Cheng Pao-ching">鄭寶清(Cheng Pao-ching) </label><br>
<input type="radio" id="鄭運鵬" name="fav_can" value="鄭運鵬">
<label for="Cheng Yun-peng">鄭運鵬(Cheng Yun-peng) </label><br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

copy



Name: Name

Property: msg. payload

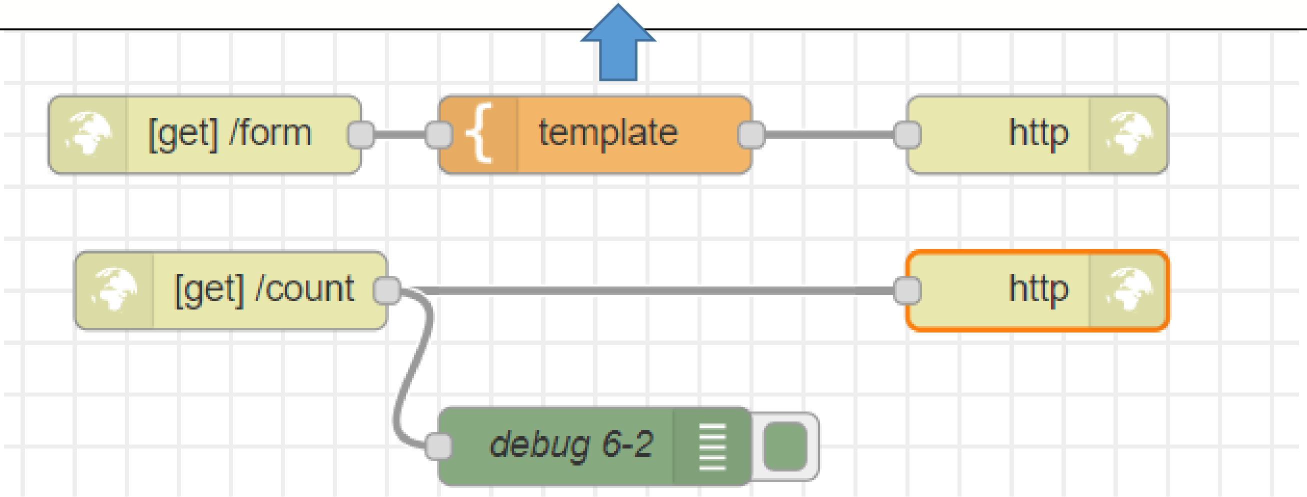
Template

Syntax Highlight: HTML

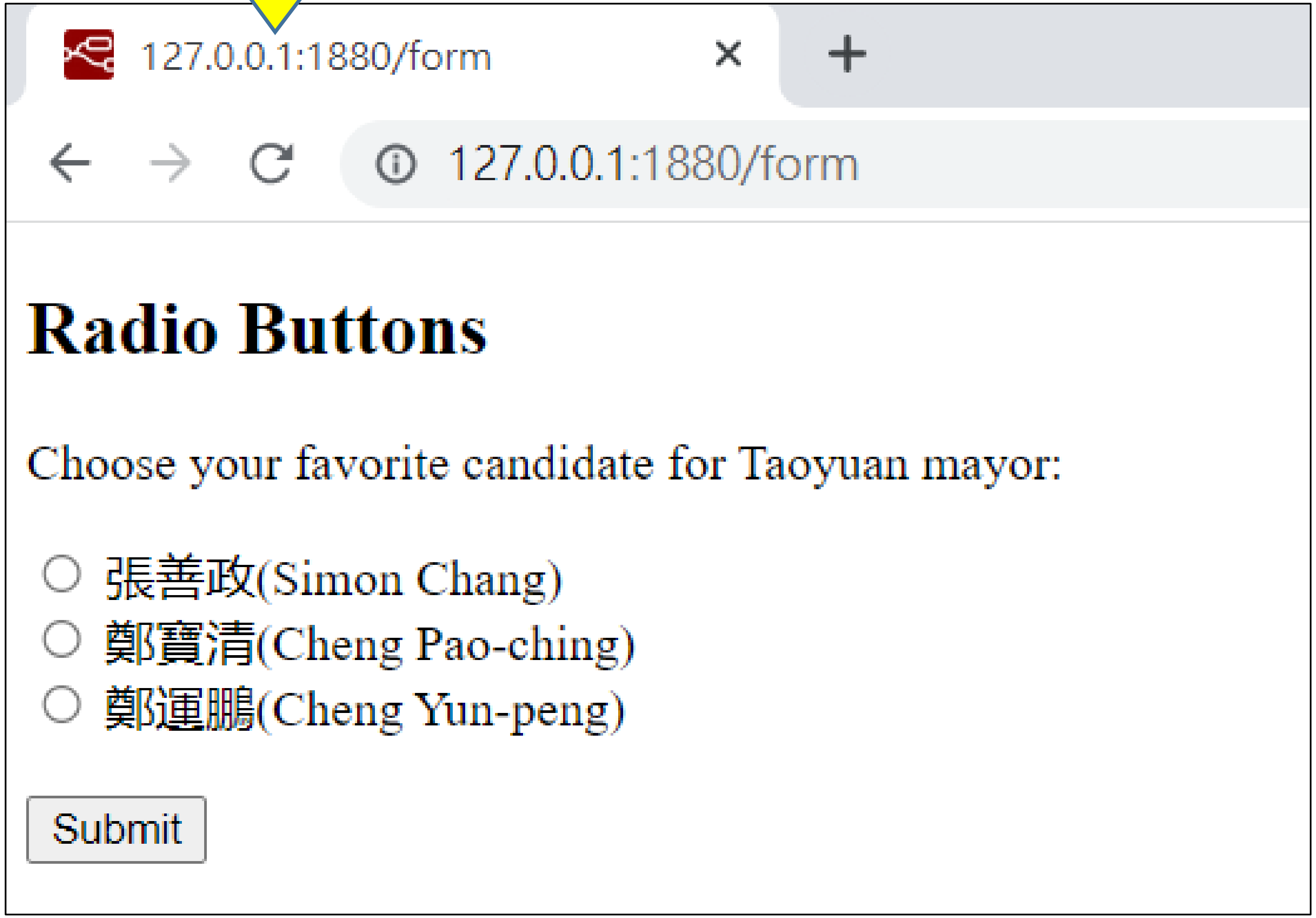
```
<!DOCTYPE html>
<html>
<body>
<h2>Radio Buttons</h2>
<p>Choose your favorite candidate for Taoyuan mayor:</p>
<form action="/count">
<input type="radio" id="張善政" name="fav_can" value="張善政">
<label for="Simon Chang">張善政(Simon Chang)</label><br>
<input type="radio" id="鄭寶清" name="fav_can" value="鄭寶清">
<label for="Cheng Pao-ching">鄭寶清(Cheng Pao-ching) </label><br>
<input type="radio" id="鄭運鵬" name="fav_can" value="鄭運鵬">
<label for="Cheng Yun-peng">鄭運鵬(Cheng Yun-peng) </label><br><br>
<input type="submit" value="Submit">
</form>
```

HTML

paste



127.0.0.1:1880/form



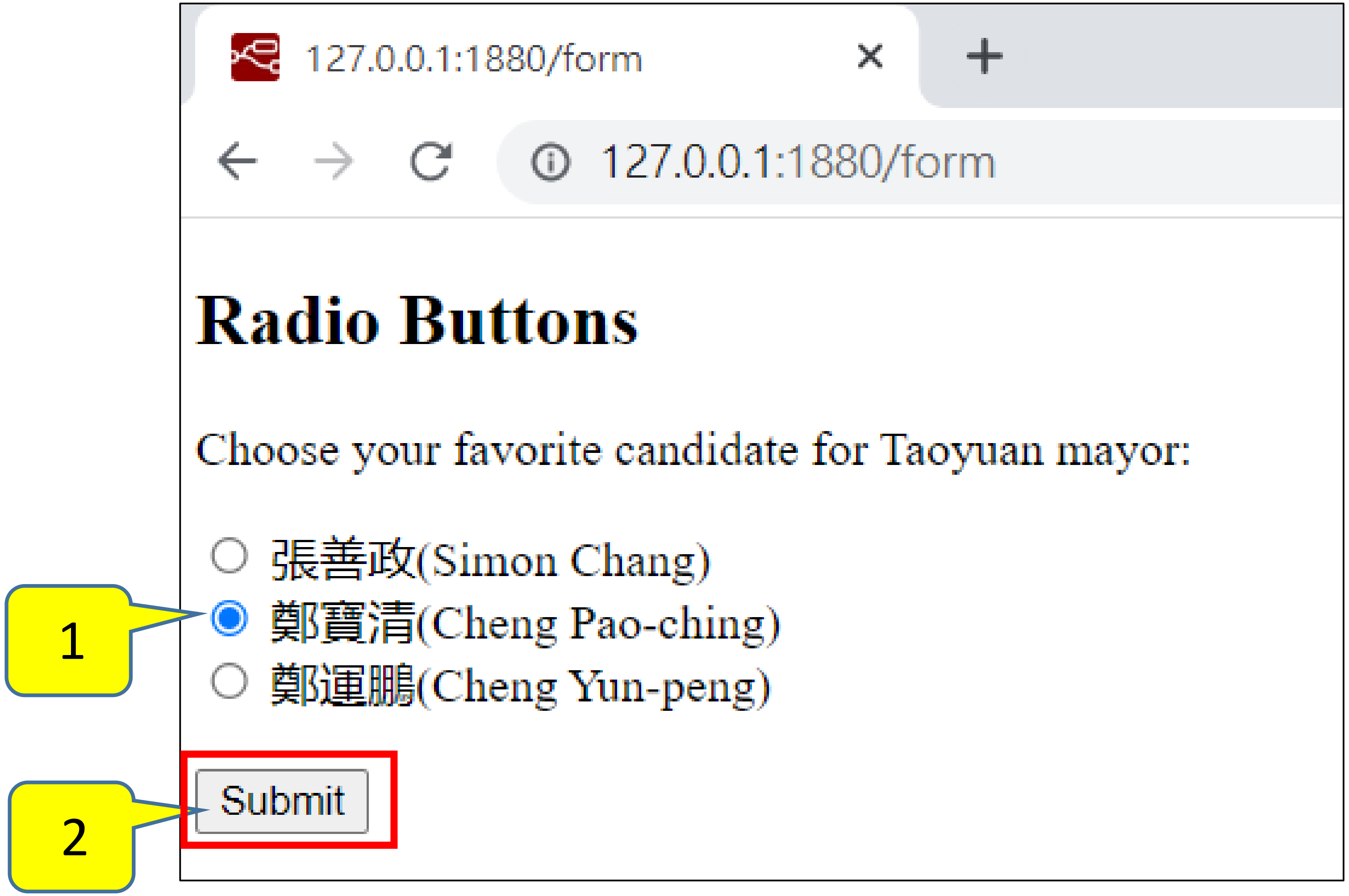
Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

Select -> Submit

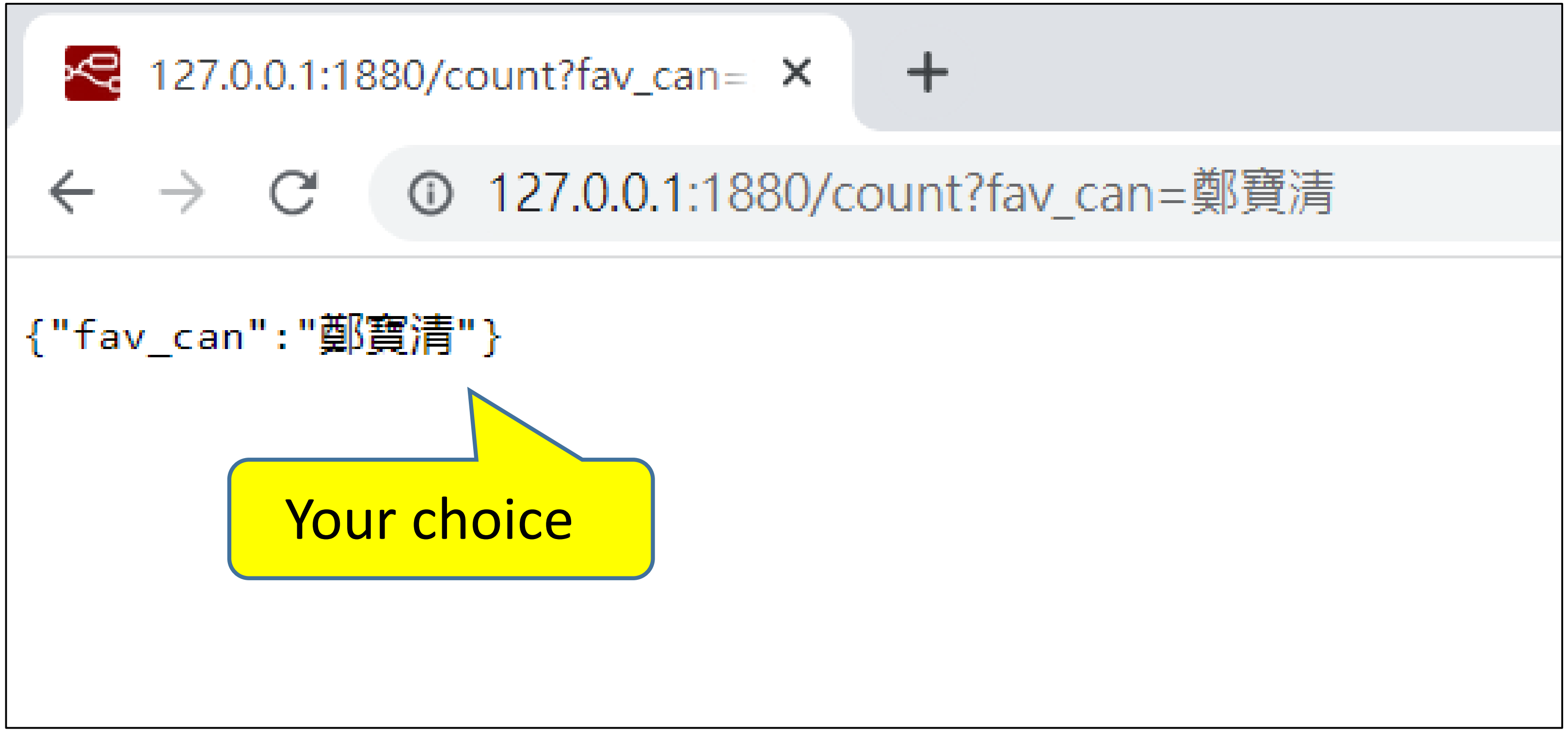


Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
● 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

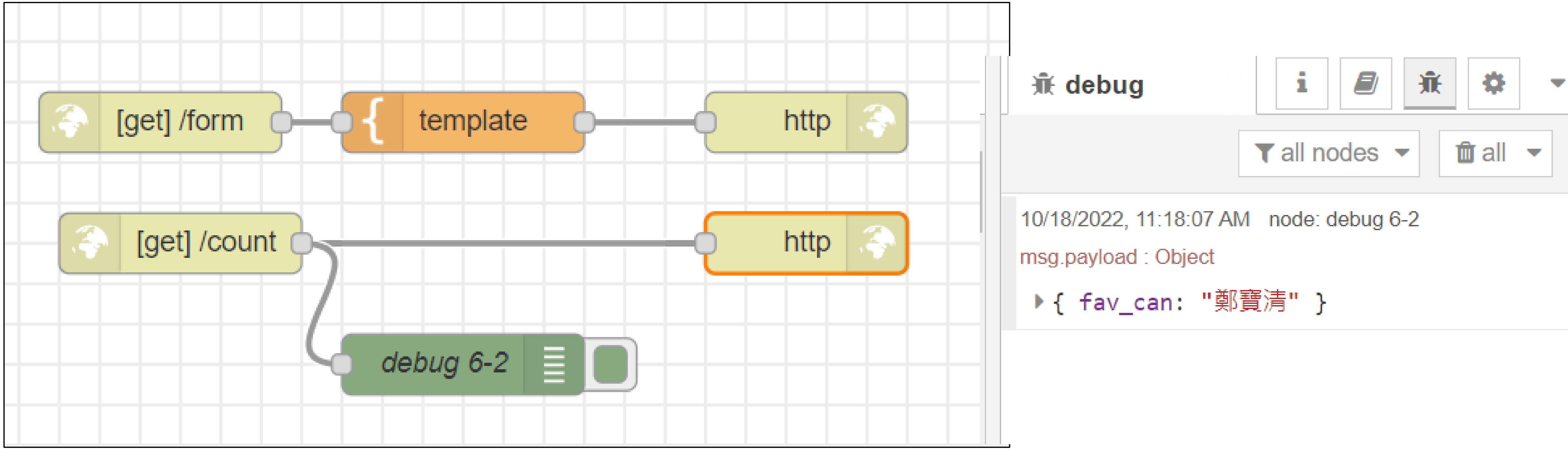


127.0.0.1:1880/count?fav_can=
127.0.0.1:1880/count?fav_can=鄭寶清

```
{"fav_can":"鄭寶清"}
```

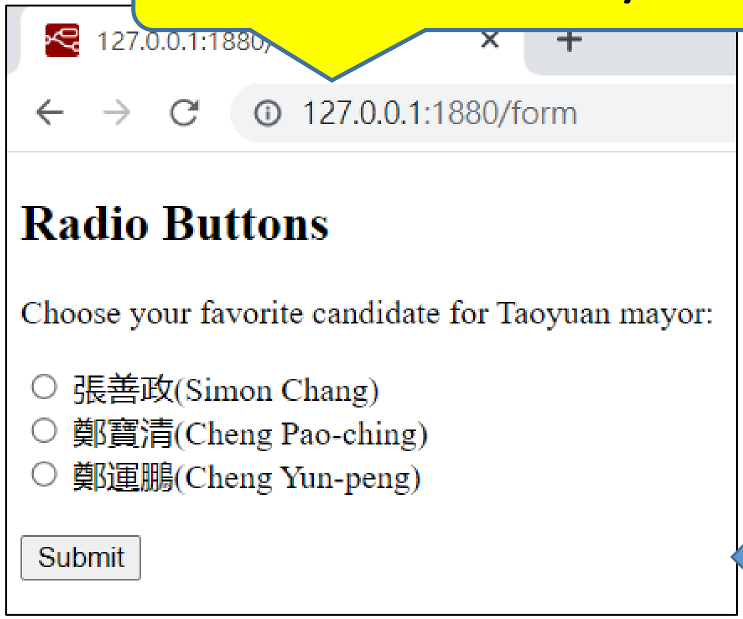
Your choice

debug



```
10/18/2022, 11:18:07 AM   node: debug 6-2
msg.payload : Object
▸{ fav_can: "鄭寶清" }
```

127.0.0.1:1880/form



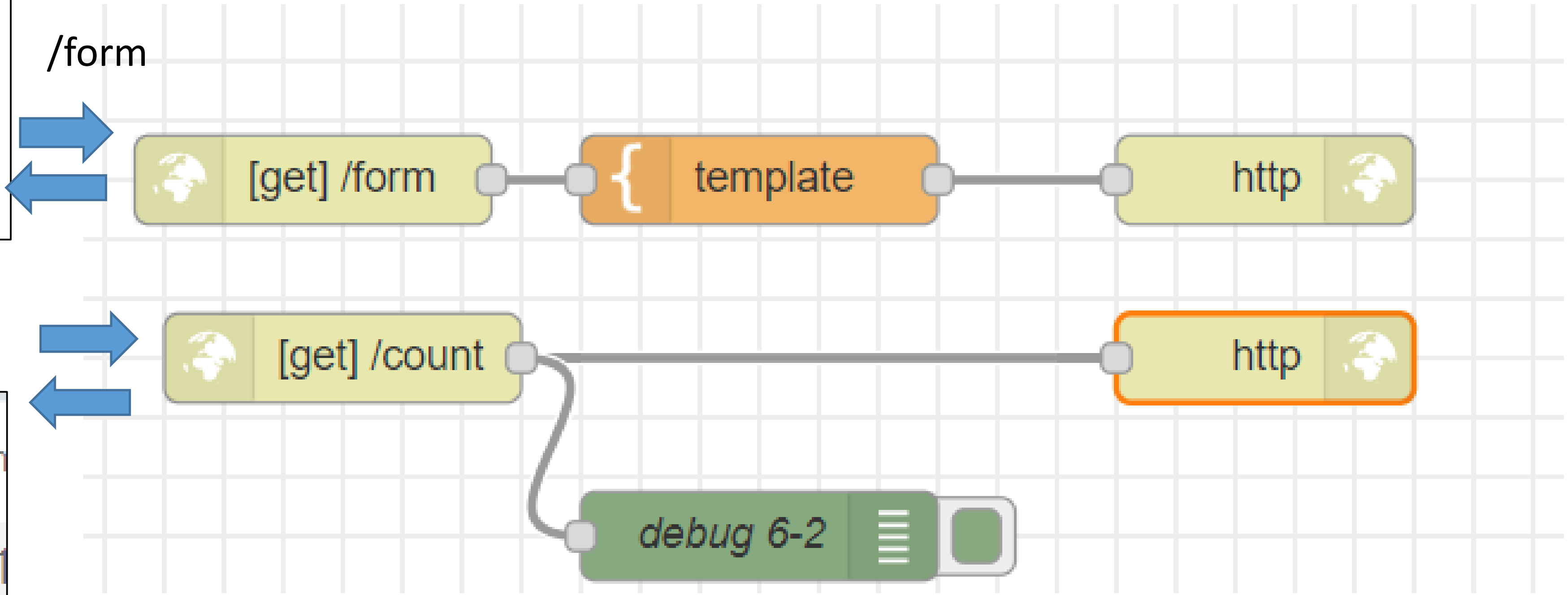
127.0.0.1:1880/form

Radio Buttons

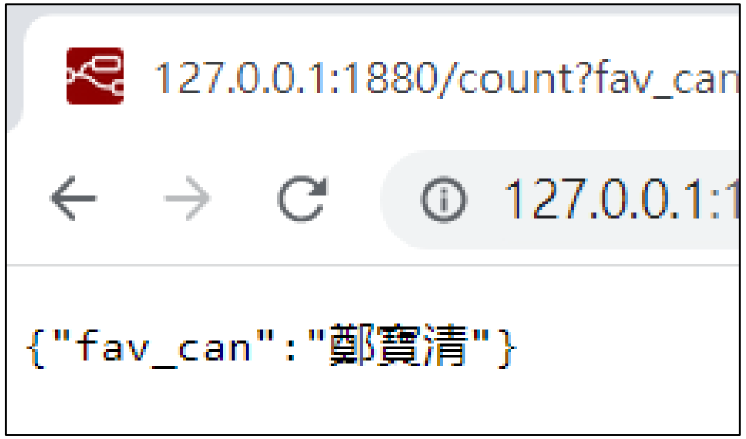
Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

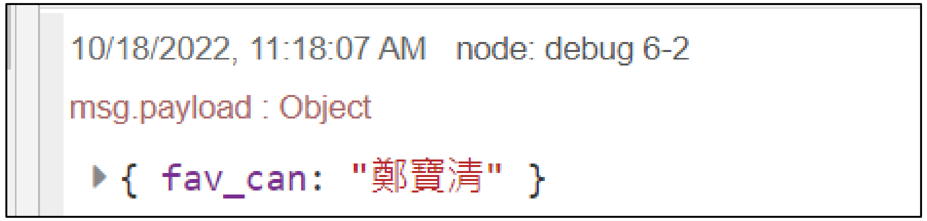


/count?fav_can=鄭寶清



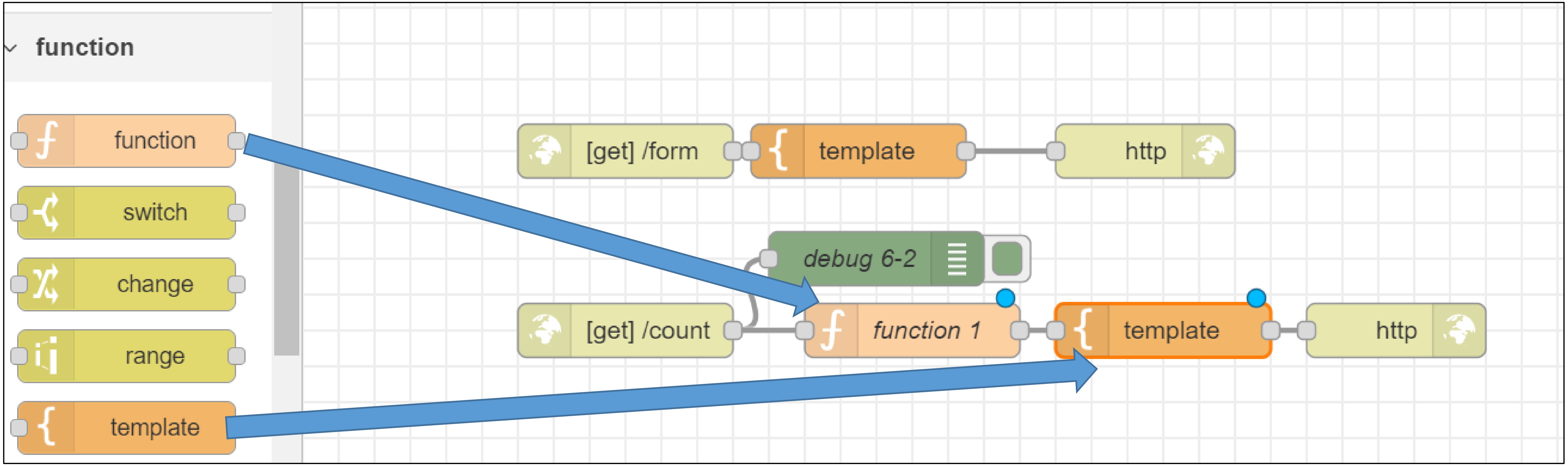
127.0.0.1:1880/count?fav_can

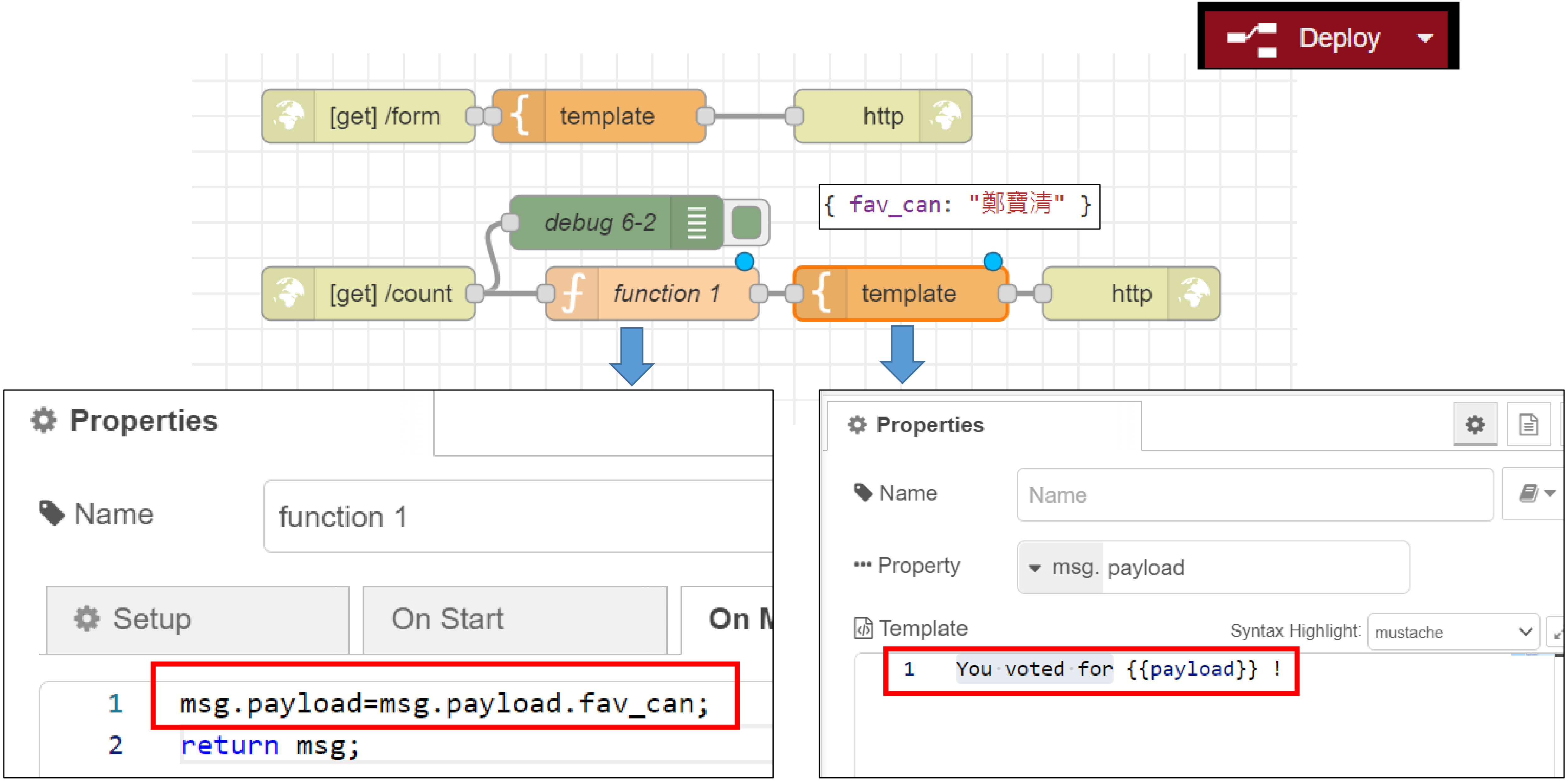
```
{"fav_can":"鄭寶清"}
```



10/18/2022, 11:18:07 AM node: debug 6-2
msg.payload : Object
▸{ fav_can: "鄭寶清" }

add a function node & a template node





```
{ fav_can: "鄭寶清" }
```

Properties

Name: function 1

```
msg.payload=msg.payload.fav_can;
return msg;
```

Properties

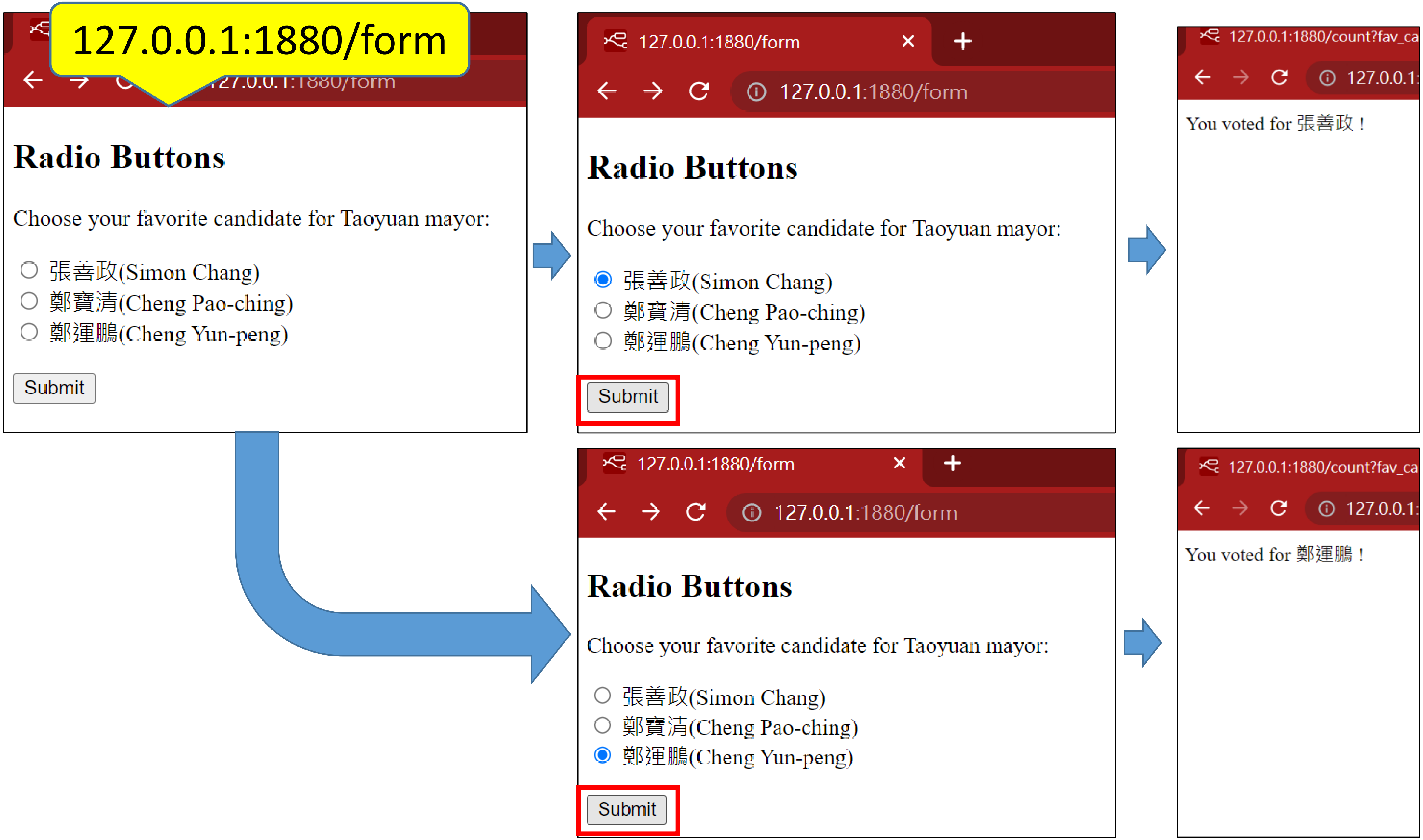
Name: Name

Property: msg. payload

Template

Syntax Highlight: mustache

```
You voted for {{payload}} !
```



127.0.0.1:1880/form

127.0.0.1:1880/form

Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

127.0.0.1:1880/form

Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

● 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

127.0.0.1:1880/count?fav_ca

You voted for 張善政！

127.0.0.1:1880/form

Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
● 鄭運鵬(Cheng Yun-peng)

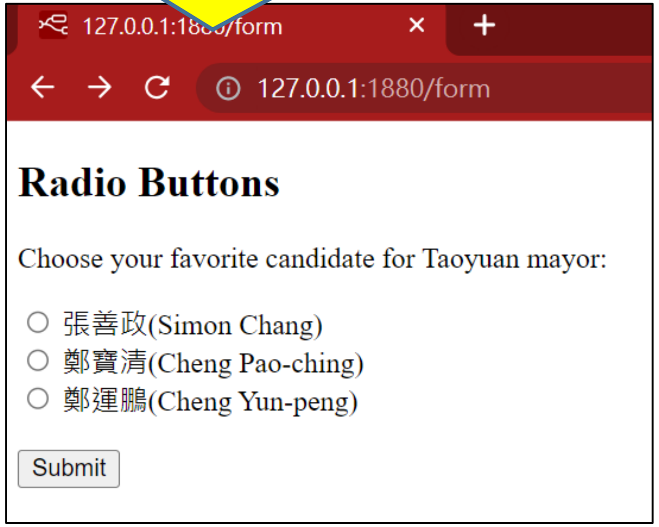
Submit

127.0.0.1:1880/count?fav_ca

You voted for 鄭運鵬！

Exercise 6-3 Evaluate the voting results

127.0.0.1:1880/form

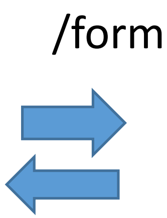


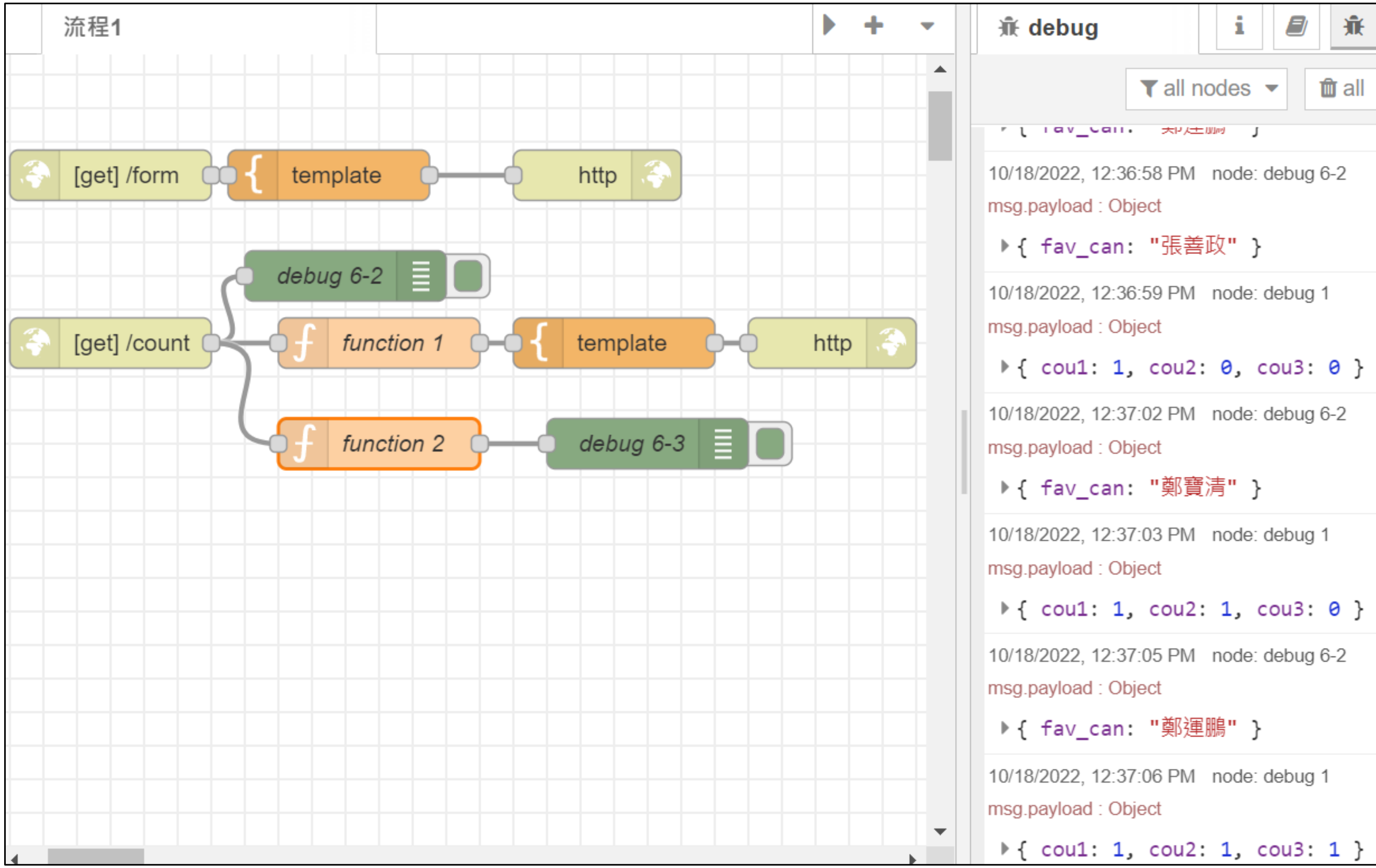
Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

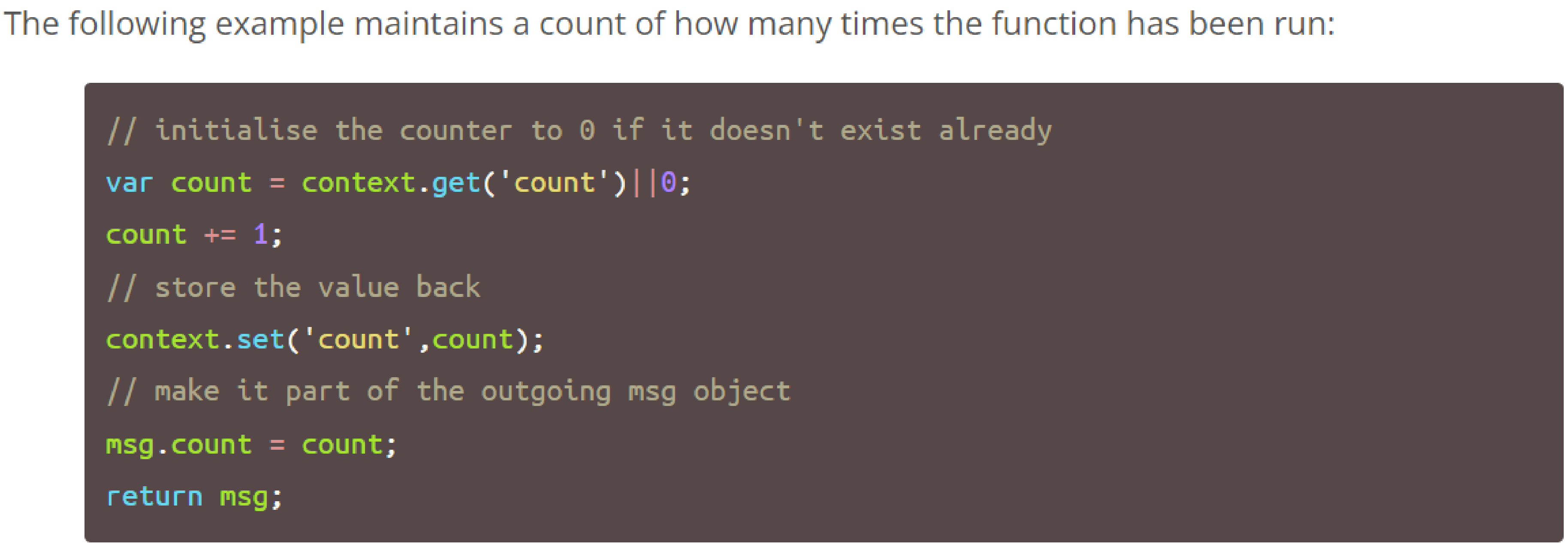




```
10/18/2022, 12:36:58 PM   node: debug 6-2
msg.payload : Object
▸{ fav_can: "張善政" }
10/18/2022, 12:36:59 PM   node: debug 1
msg.payload : Object
▸{ cou1: 1, cou2: 0, cou3: 0 }
10/18/2022, 12:37:02 PM   node: debug 6-2
msg.payload : Object
▸{ fav_can: "鄭寶清" }
10/18/2022, 12:37:03 PM   node: debug 1
msg.payload : Object
▸{ cou1: 1, cou2: 1, cou3: 0 }
10/18/2022, 12:37:05 PM   node: debug 6-2
msg.payload : Object
▸{ fav_can: "鄭運鵬" }
10/18/2022, 12:37:06 PM   node: debug 1
msg.payload : Object
▸{ cou1: 1, cou2: 1, cou3: 1 }
```

What is context?

- Node-RED provides a way to store information that can be shared between different nodes without using the messages that pass through a flow. This is called ‘context’.

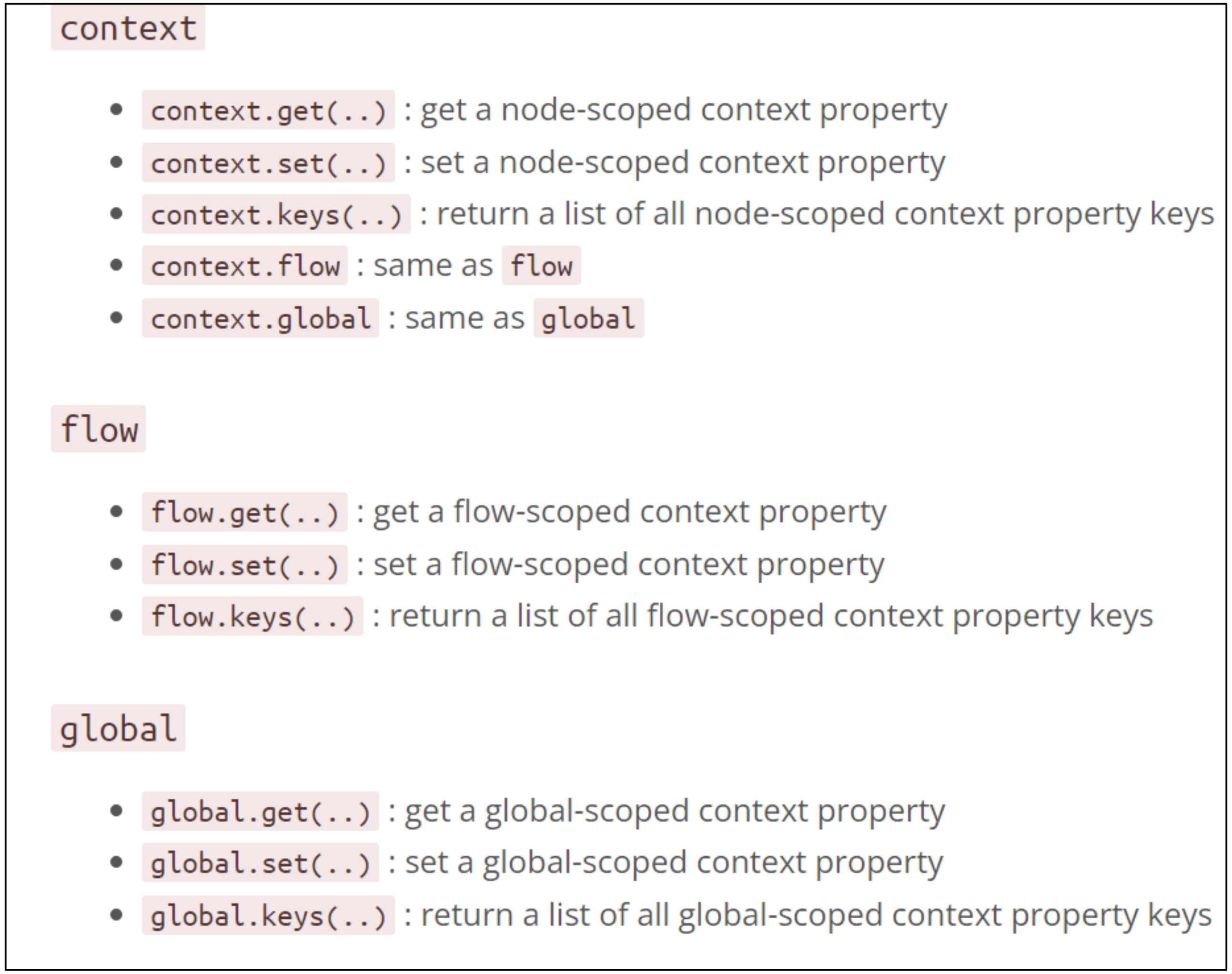


The following example maintains a count of how many times the function has been run:

```
// initialise the counter to 0 if it doesn't exist already
var count = context.get('count')||0;
count += 1;
// store the value back
context.set('count',count);
// make it part of the outgoing msg object
msg.count = count;
return msg;
```

https://nodered.org/docs/user-guide/writing-functions

API Reference



`context`

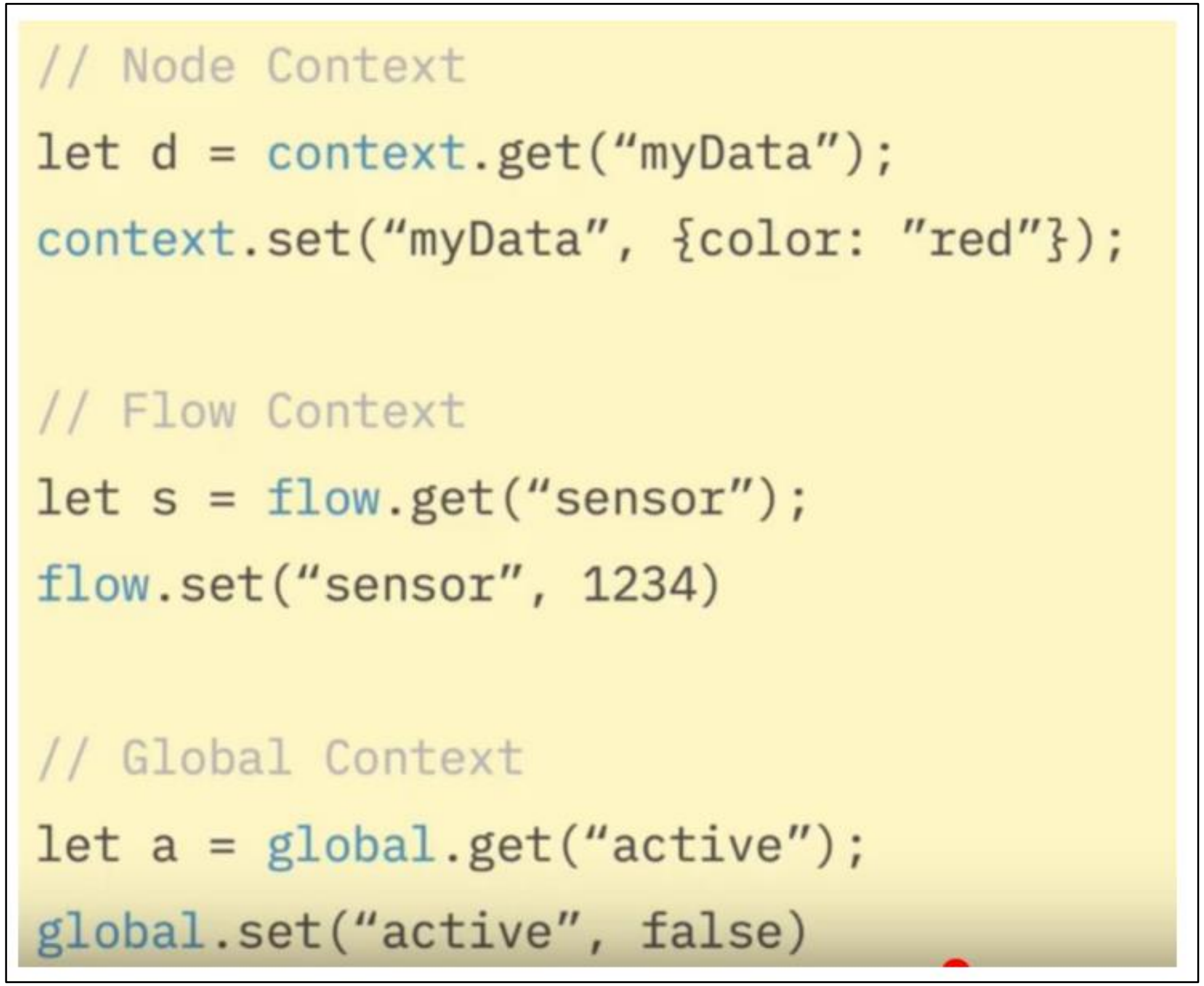
- `context.get(..)` : get a node-scoped context property
- `context.set(..)` : set a node-scoped context property
- `context.keys(..)` : return a list of all node-scoped context property keys
- `context.flow` : same as `flow`
- `context.global` : same as `global`

`flow`

- `flow.get(..)` : get a flow-scoped context property
- `flow.set(..)` : set a flow-scoped context property
- `flow.keys(..)` : return a list of all flow-scoped context property keys

`global`

- `global.get(..)` : get a global-scoped context property
- `global.set(..)` : set a global-scoped context property
- `global.keys(..)` : return a list of all global-scoped context property keys

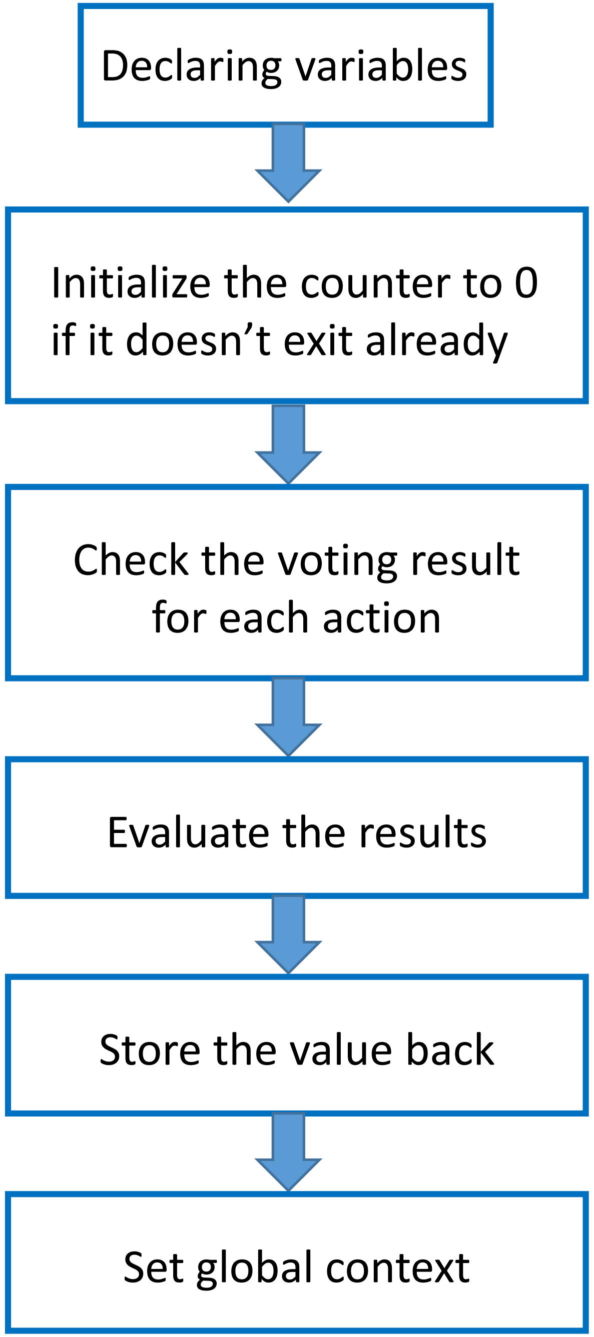


```
// Node Context
let d = context.get(“myData”);
context.set(“myData”, {color: ”red”});


// Flow Context
let s = flow.get(“sensor”);
flow.set(“sensor”, 1234)


// Global Context
let a = global.get(“active”);
global.set(“active”, false)
```

Flow chart



Declaring variables

Initialize the counter to 0 if it doesn’t exit already

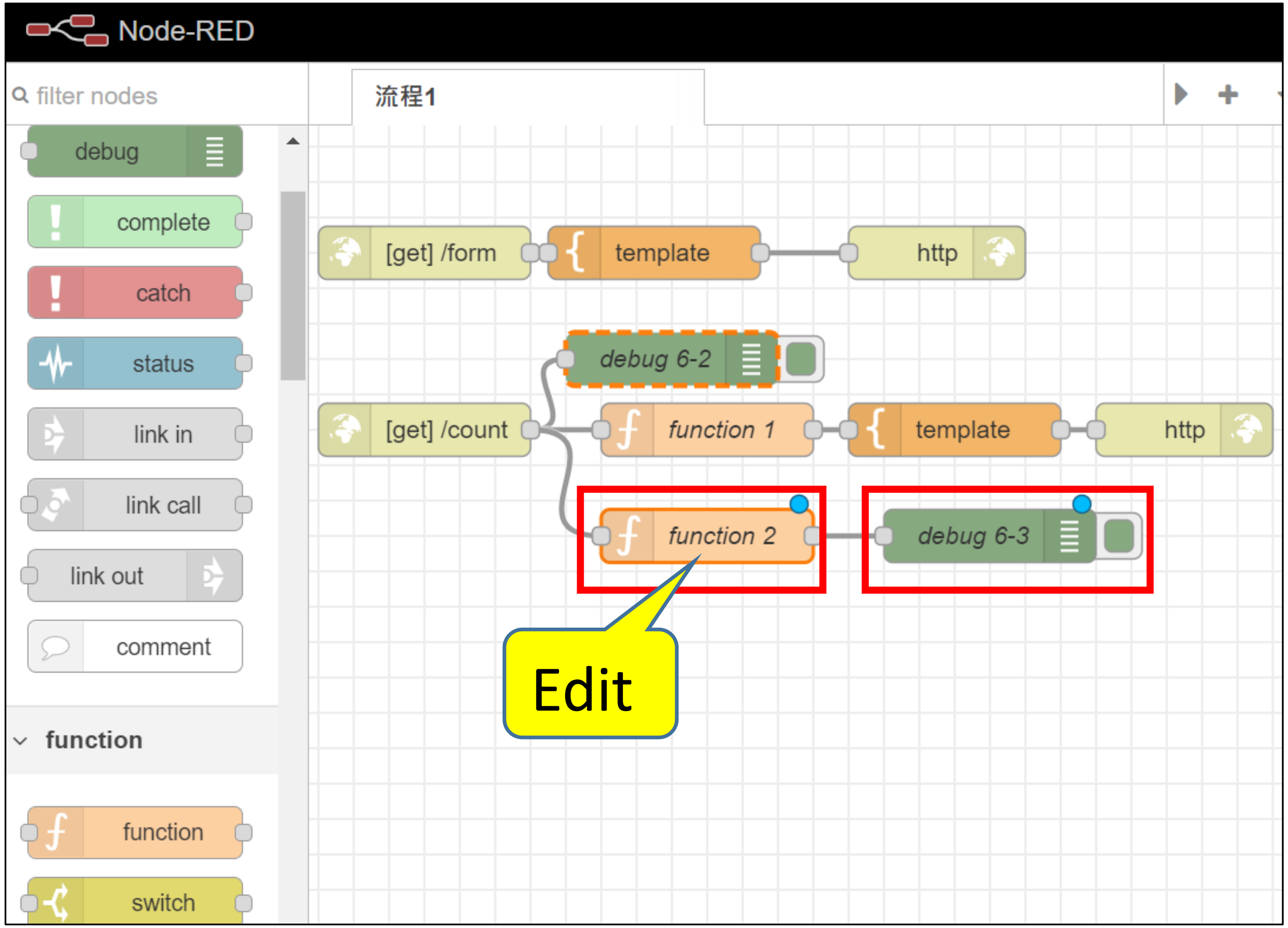
Check the voting result for each action

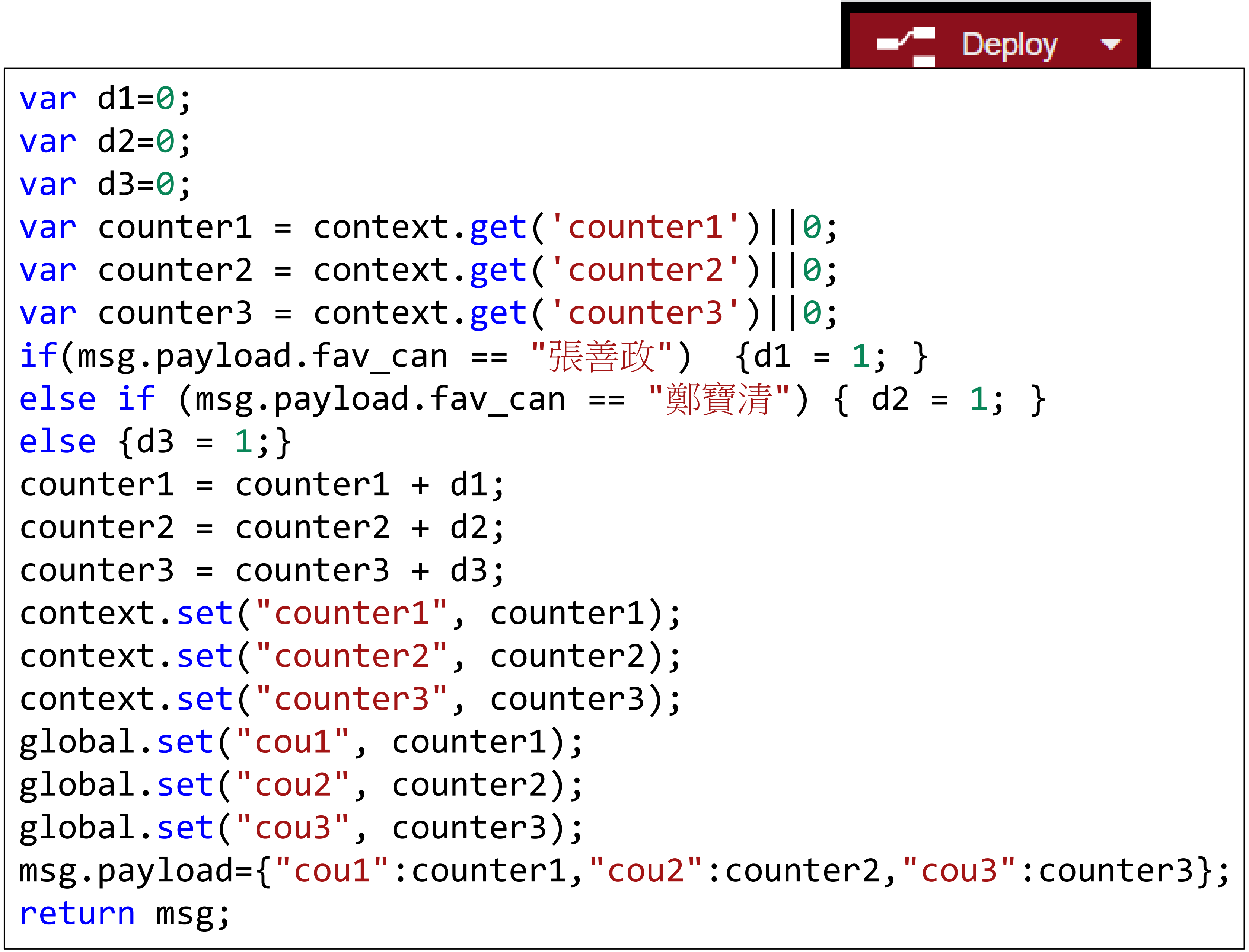
Evaluate the results

Store the value back

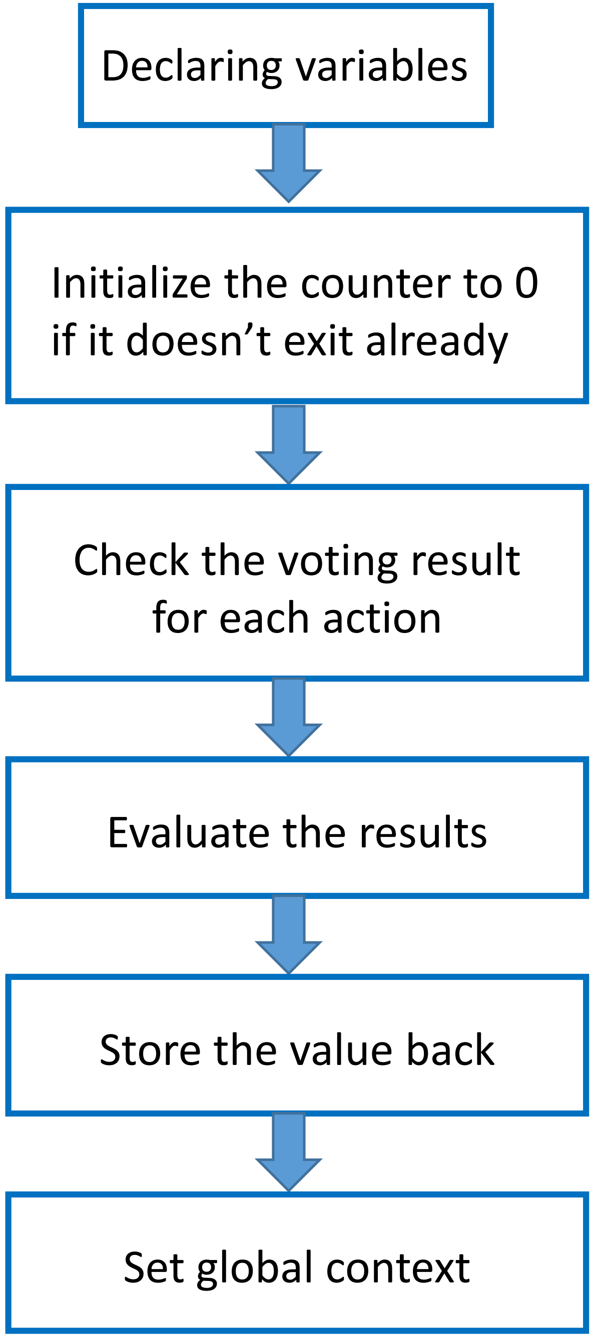
Set global context

add a function node & a debug node





```
var d1=0;
var d2=0;
var d3=0;
var counter1 = context.get('counter1')||0;
var counter2 = context.get('counter2')||0;
var counter3 = context.get('counter3')||0;
if(msg.payload.fav_can == "張善政")  {d1 = 1; }
else if (msg.payload.fav_can == "鄭寶清") { d2 = 1; }
else {d3 = 1;}
counter1 = counter1 + d1;
counter2 = counter2 + d2;
counter3 = counter3 + d3;
context.set("counter1", counter1);
context.set("counter2", counter2);
context.set("counter3", counter3);
global.set("cou1", counter1);
global.set("cou2", counter2);
global.set("cou3", counter3);
msg.payload={"cou1":counter1,"cou2":counter2,"cou3":counter3};
return msg;
```



Declaring variables

Initialize the counter to 0 if it doesn’t exit already

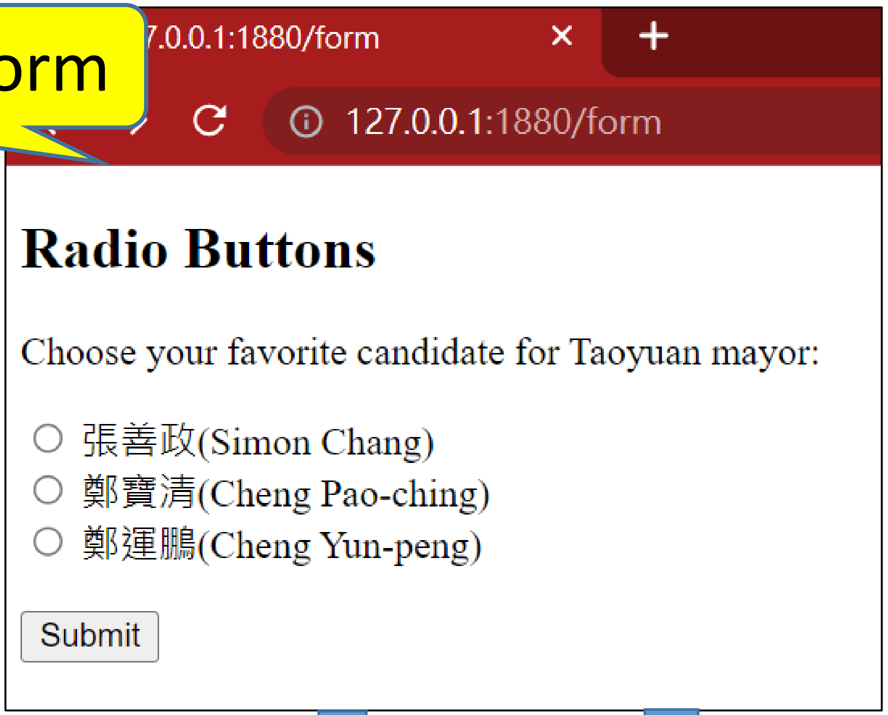
Check the voting result for each action

Evaluate the results

Store the value back

Set global context

127.0.0.1:1880/form

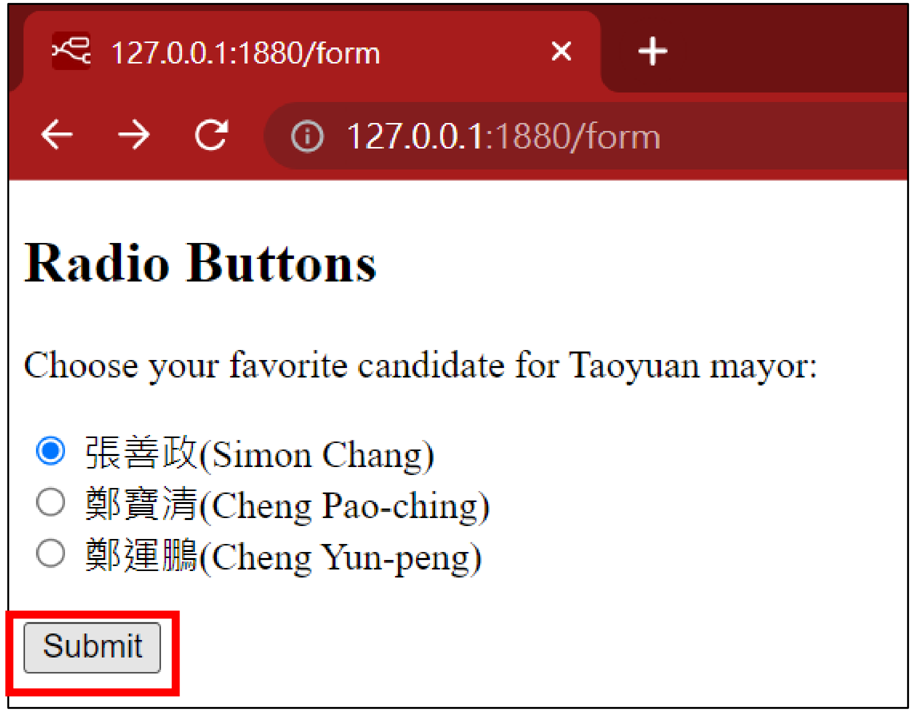


Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

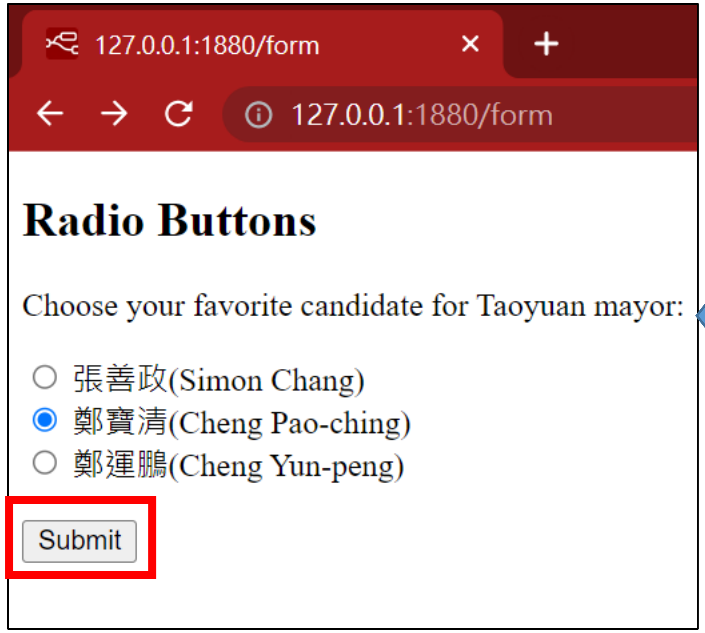


Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

● 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

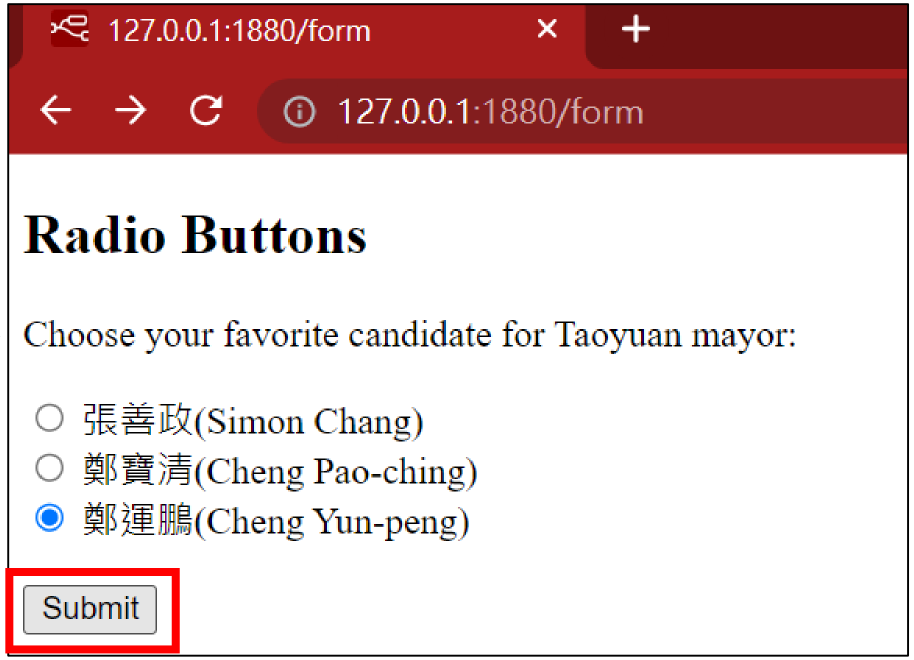


Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
● 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

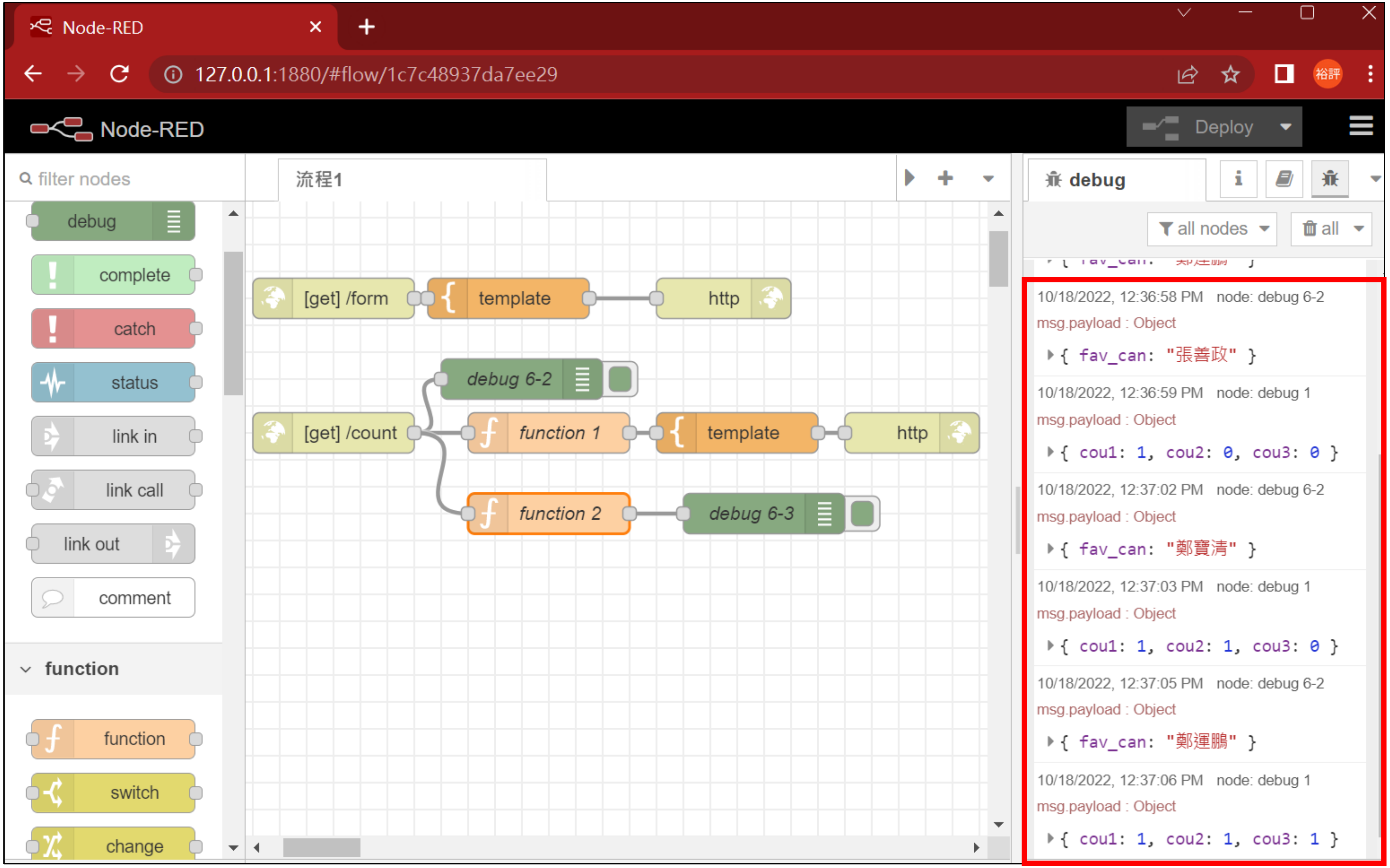


Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

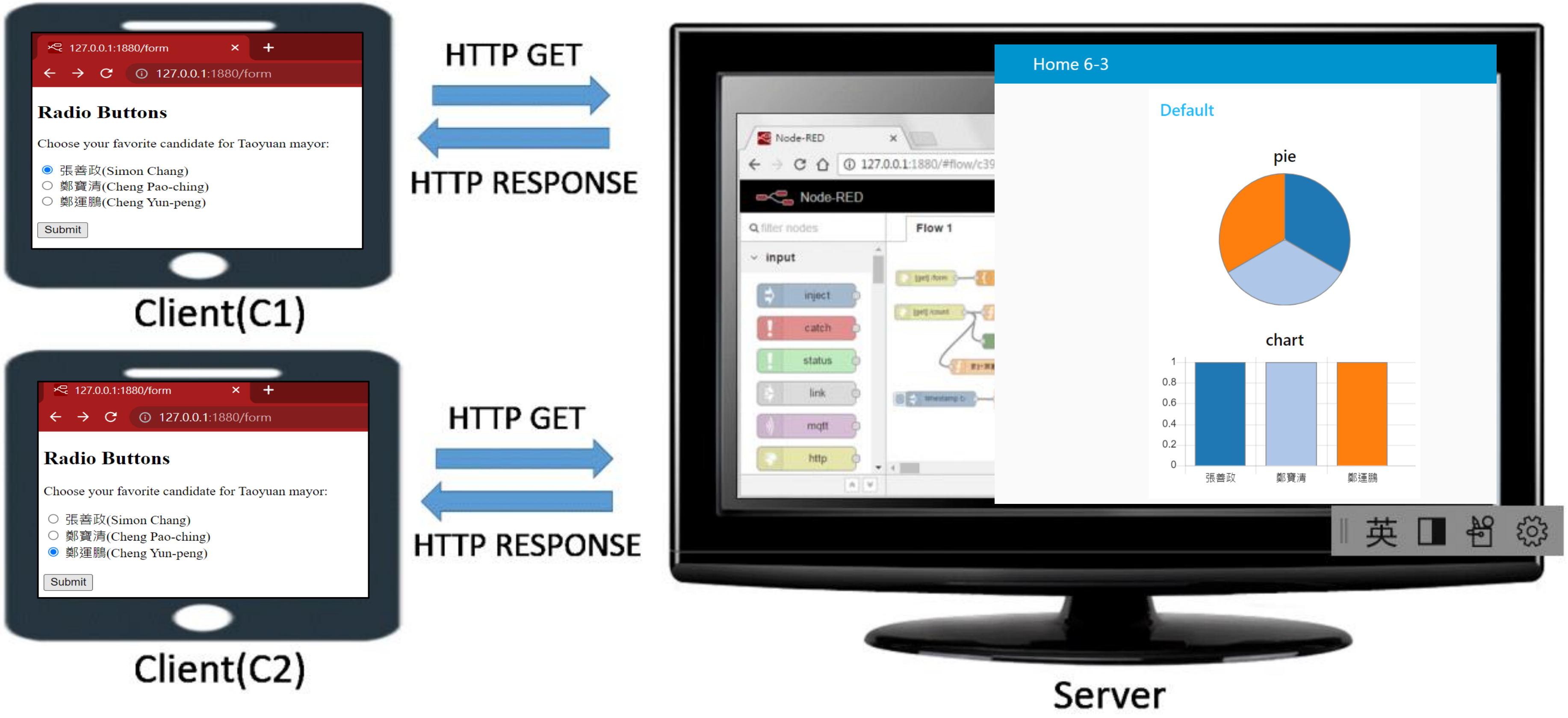
○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
● 鄭運鵬(Cheng Yun-peng)

Submit



```
10/18/2022, 12:36:58 PM   node: debug 6-2
msg.payload : Object
▸{ fav_can: "張善政" }
10/18/2022, 12:36:59 PM   node: debug 1
msg.payload : Object
▸{ cou1: 1, cou2: 0, cou3: 0 }
10/18/2022, 12:37:02 PM   node: debug 6-2
msg.payload : Object
▸{ fav_can: "鄭寶清" }
10/18/2022, 12:37:03 PM   node: debug 1
msg.payload : Object
▸{ cou1: 1, cou2: 1, cou3: 0 }
10/18/2022, 12:37:05 PM   node: debug 6-2
msg.payload : Object
▸{ fav_can: "鄭運鵬" }
10/18/2022, 12:37:06 PM   node: debug 1
msg.payload : Object
▸{ cou1: 1, cou2: 1, cou3: 1 }
```

Exercise 6-4 Visualize the voting results



Radio Buttons

Choose your favorite candidate for Taoyuan mayor:

● 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
○ 鄭運鵬(Cheng Yun-peng)

Submit

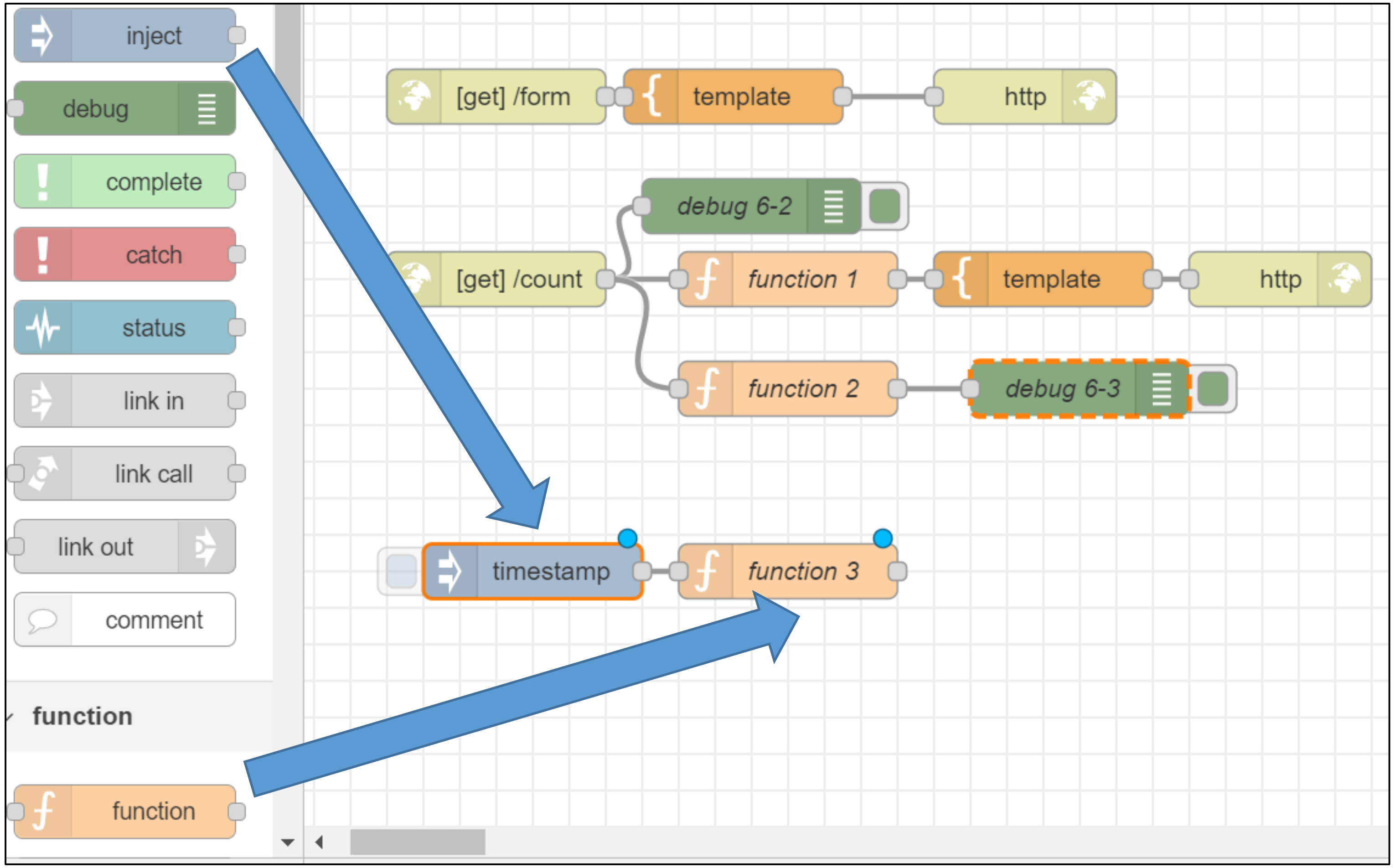
Radio Buttons

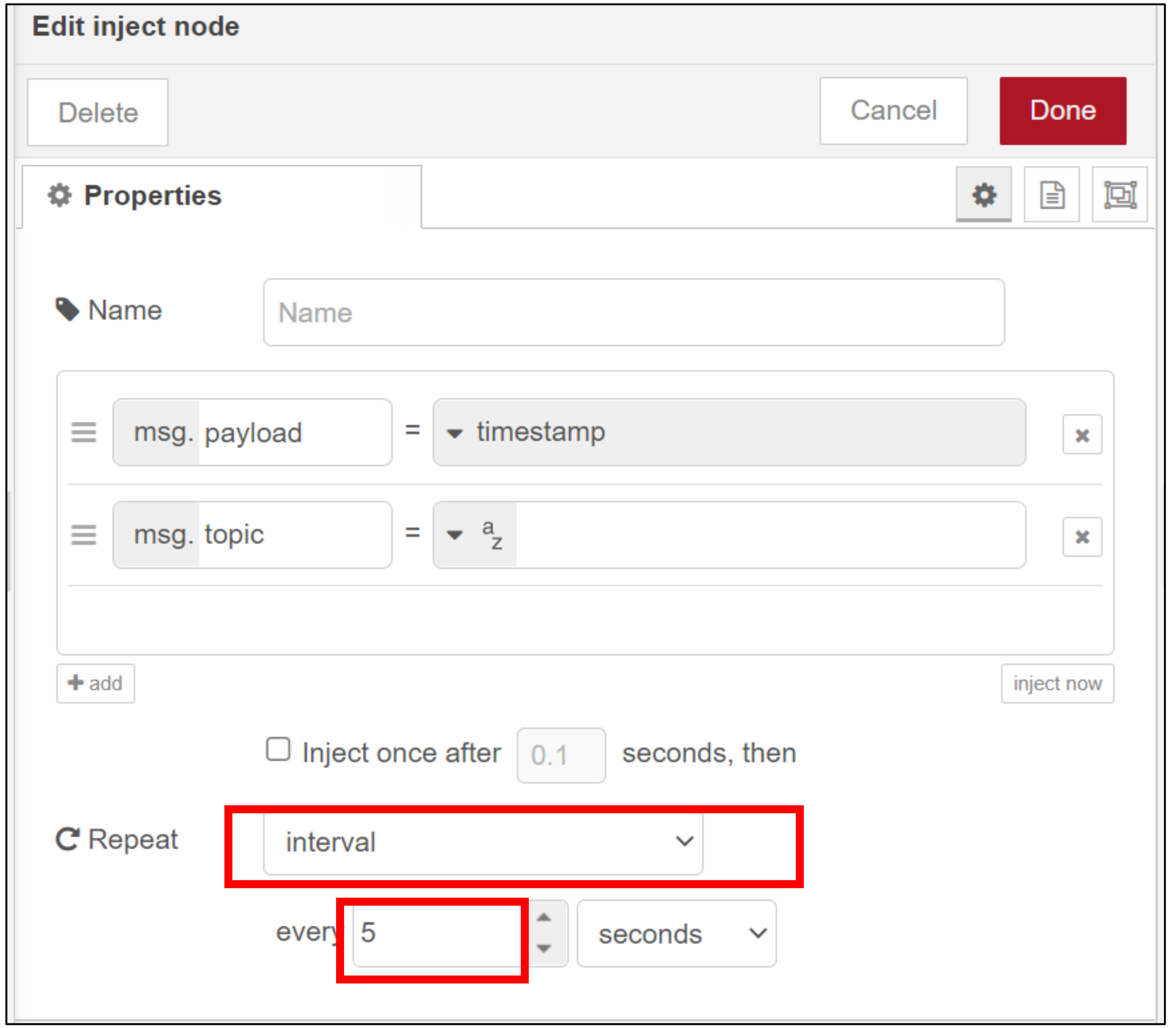
Choose your favorite candidate for Taoyuan mayor:

○ 張善政(Simon Chang)
○ 鄭寶清(Cheng Pao-ching)
● 鄭運鵬(Cheng Yun-peng)

Submit

add a inject node & a function node





Edit inject node

Name: Name

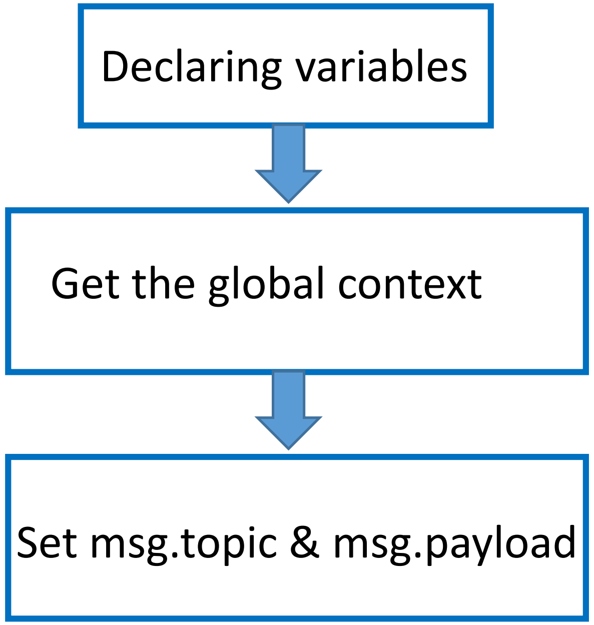
msg. payload = timestamp

msg. topic = ^a_z

□ Inject once after 0.1 seconds, then

Repeat: interval

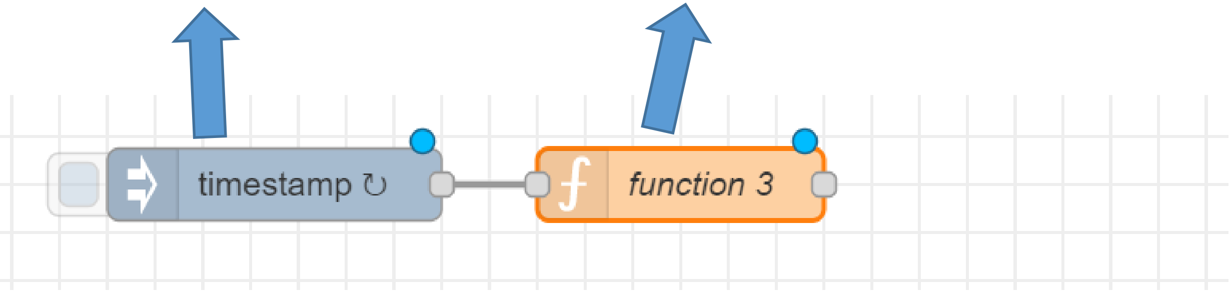
every 5 seconds



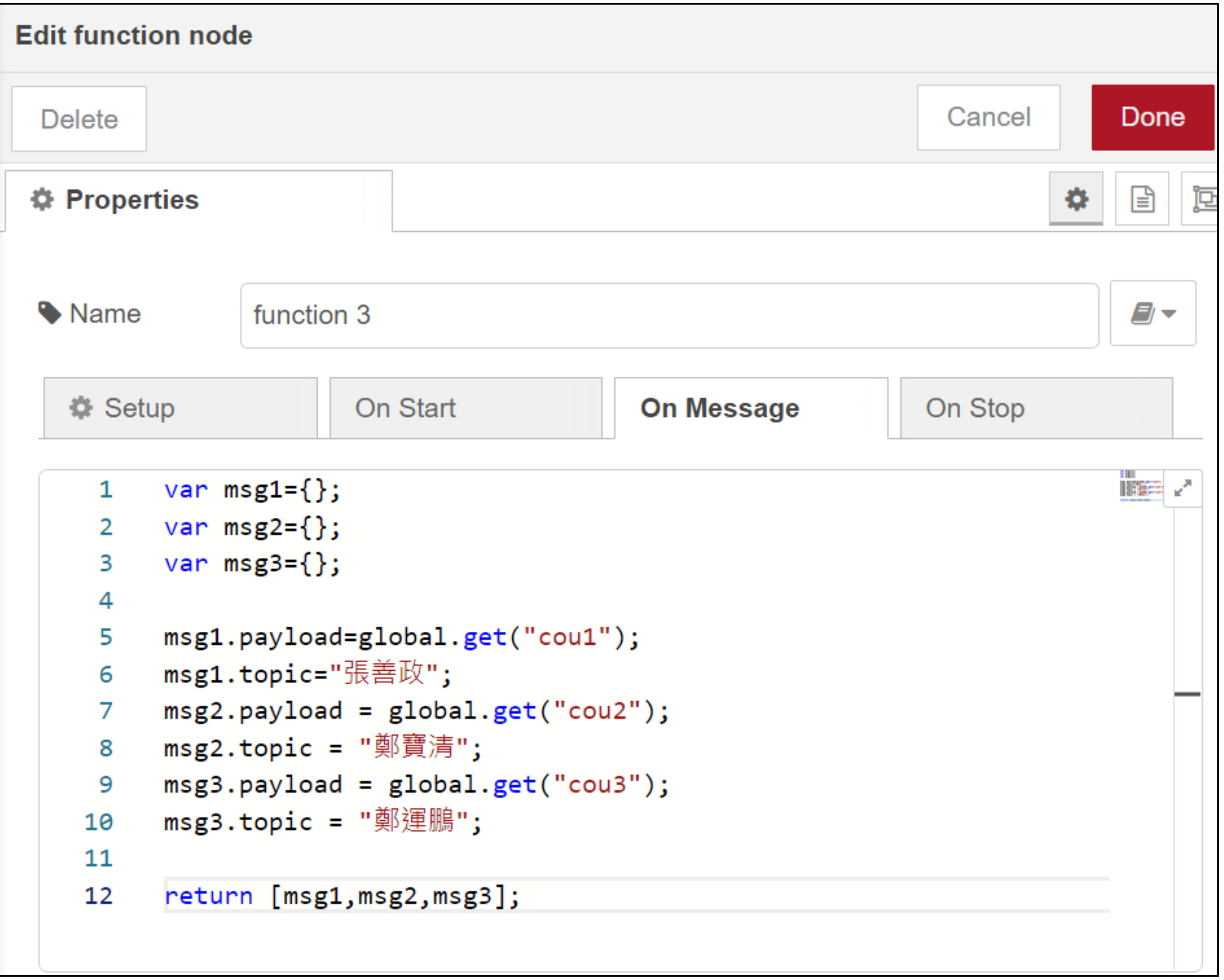
Declaring variables

Get the global context

Set msg.topic & msg.payload



Edit function node



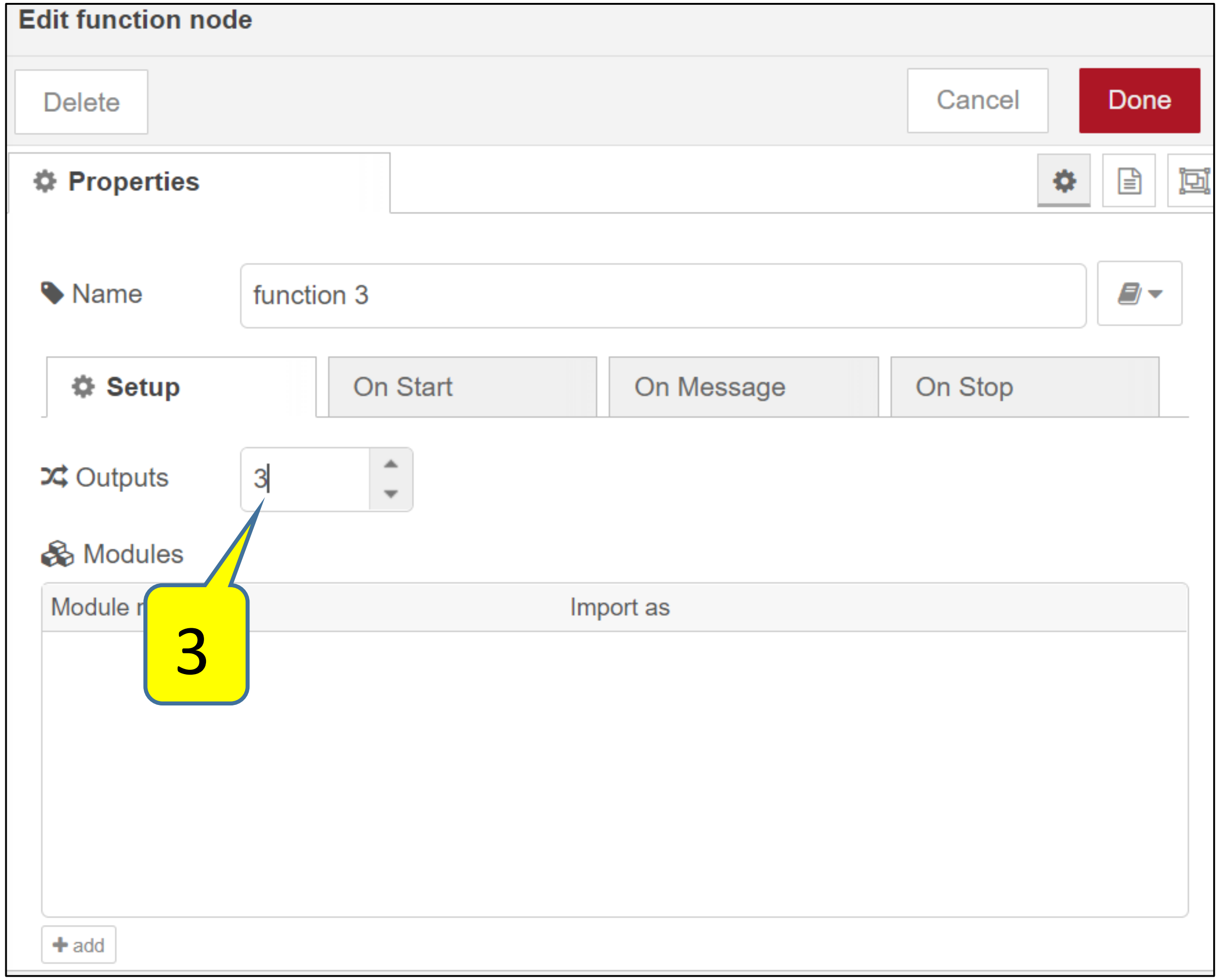
Edit function node

Name: function 3

```
var msg1={};
var msg2={};
var msg3={};

msg1.payload=global.get("cou1");
msg1.topic="張善政";
msg2.payload = global.get("cou2");
msg2.topic = "鄭寶清";
msg3.payload = global.get("cou3");
msg3.topic = "鄭運鵬";

return [msg1,msg2,msg3];
```



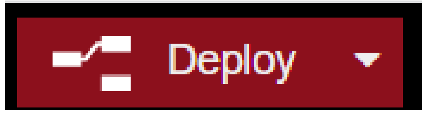
Edit function node

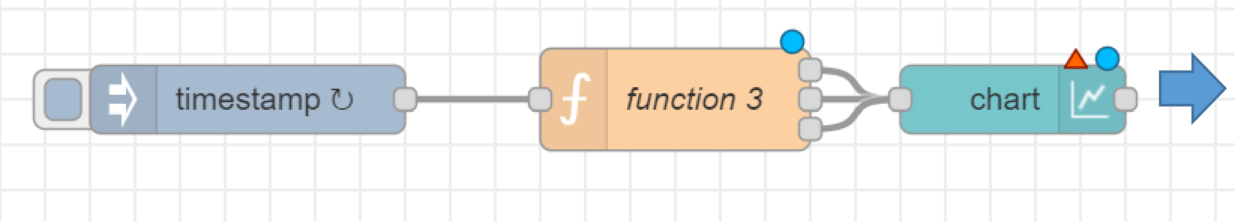
Name: function 3

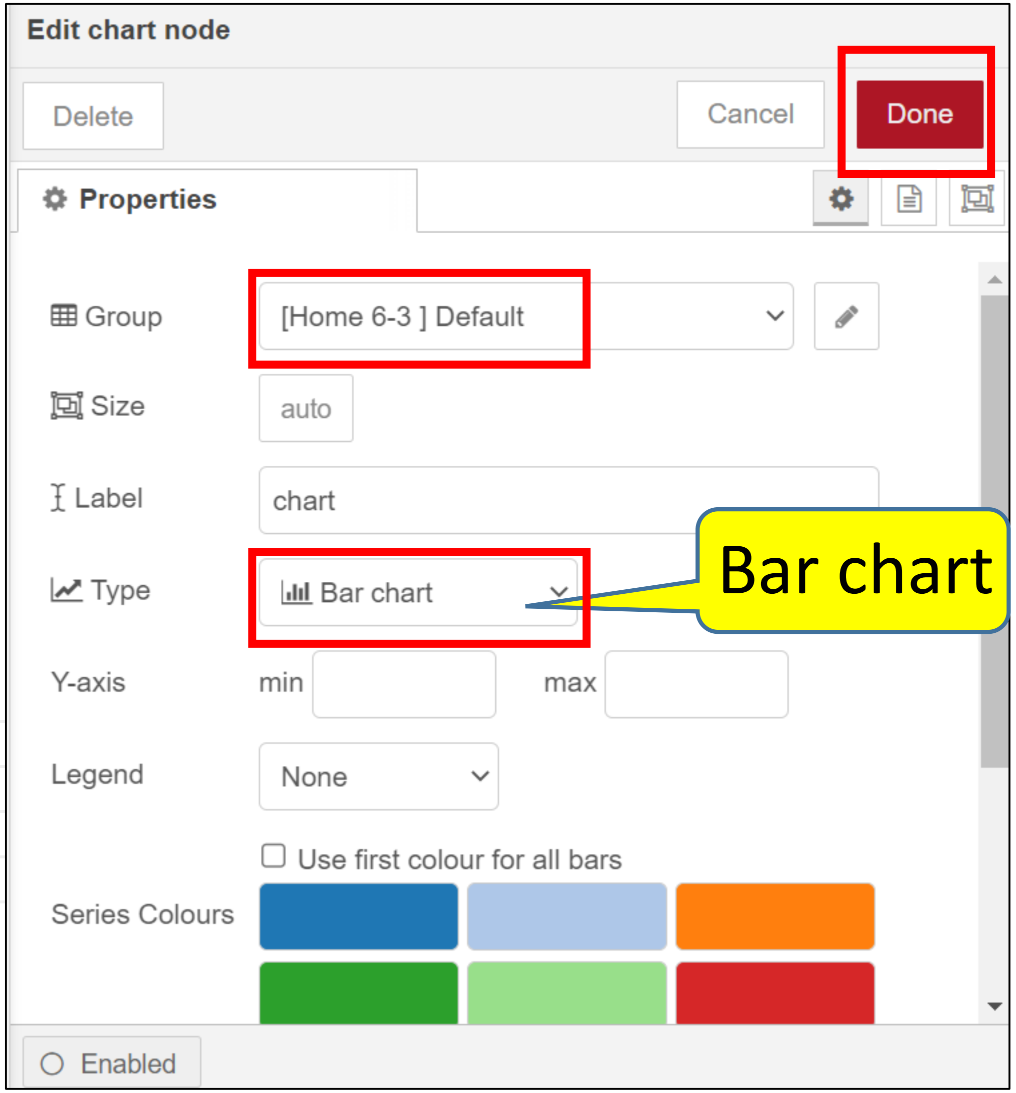
Outputs: 3

3

Add chart







Group [Home 6-3] Default

Size auto

Label chart

Type Bar chart

Y-axis min max

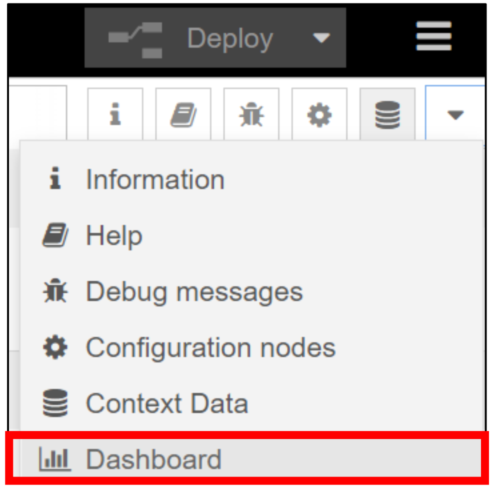
Legend None

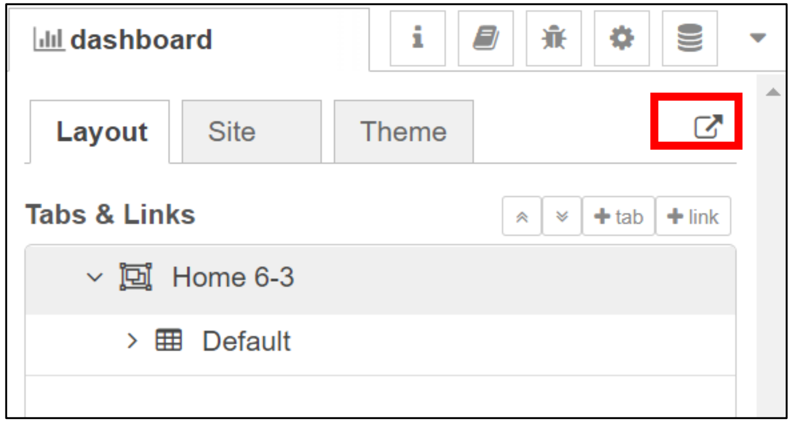
☐ Use first colour for all bars

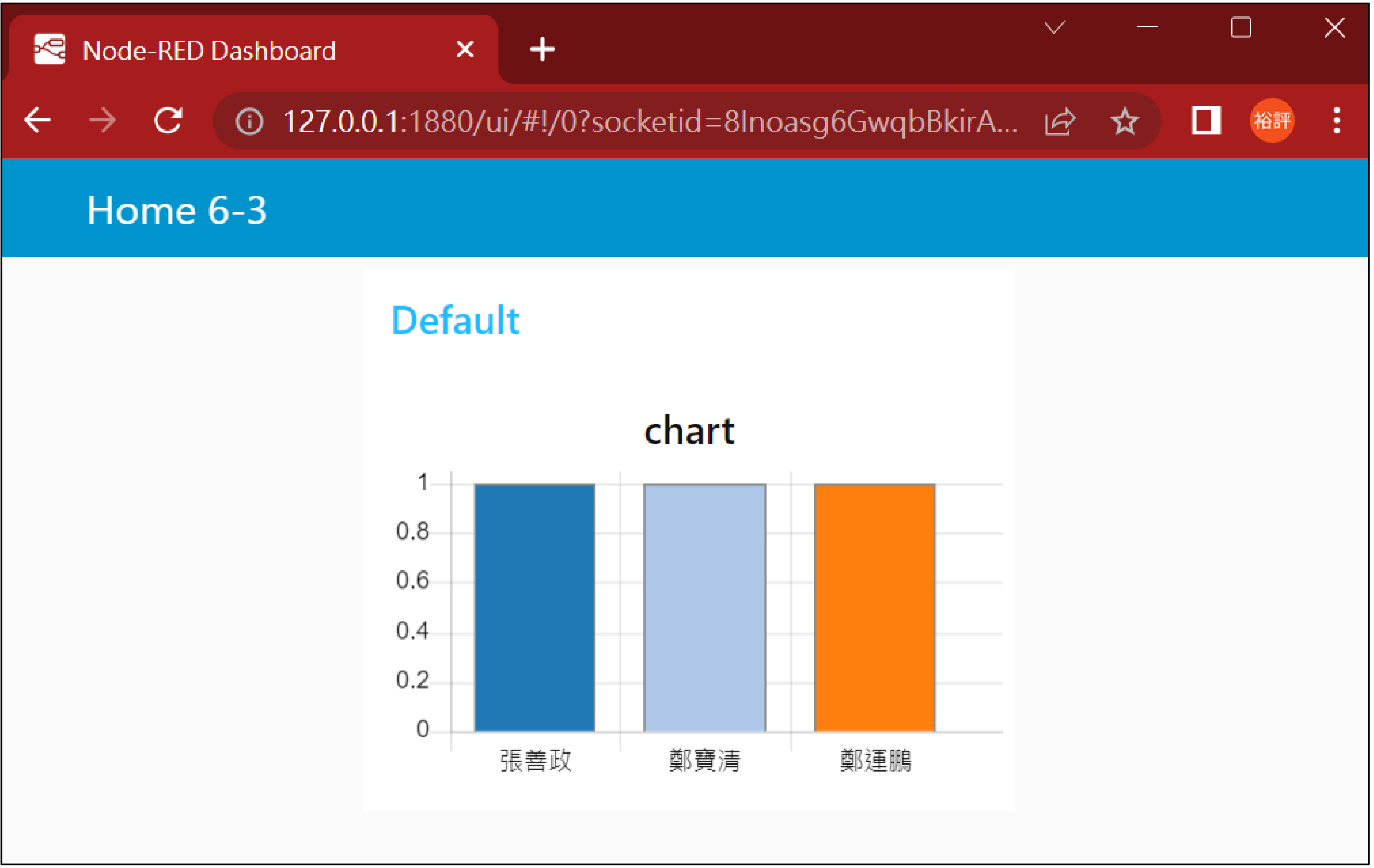
Series Colours

○ Enabled

Bar chart

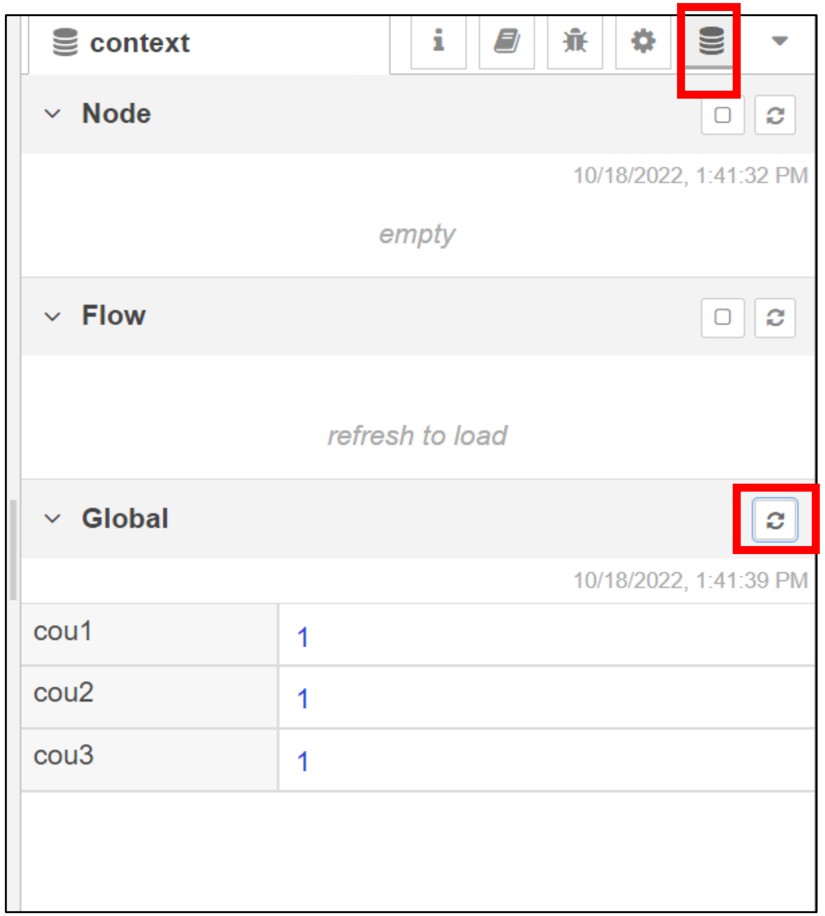






127.0.0.1:1880/ui/#!/0?socketid=8Inoasg6GwqbBkirA...

Context data



context

Node

10/18/2022, 1:41:32 PM

empty

Flow

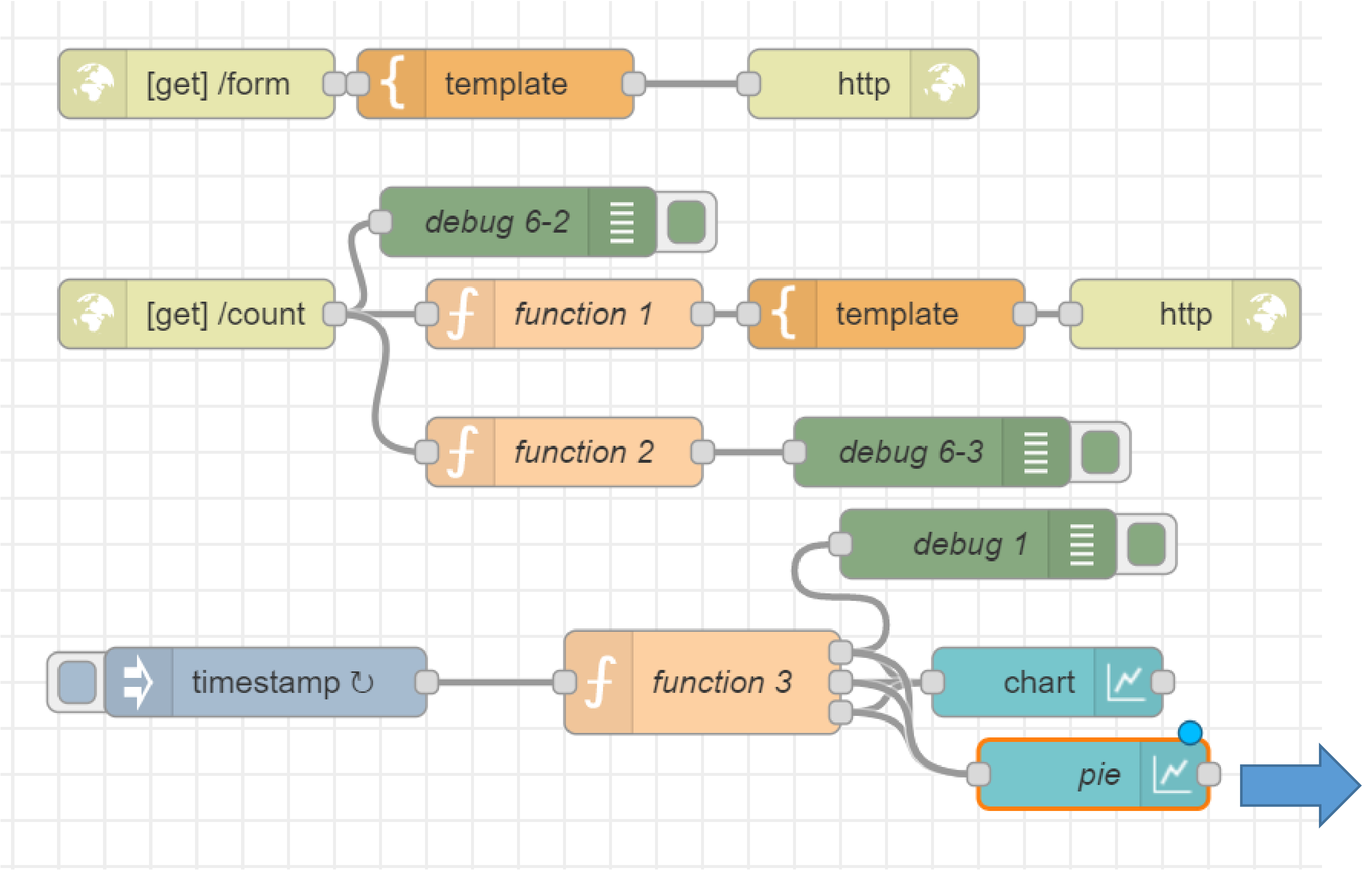
refresh to load

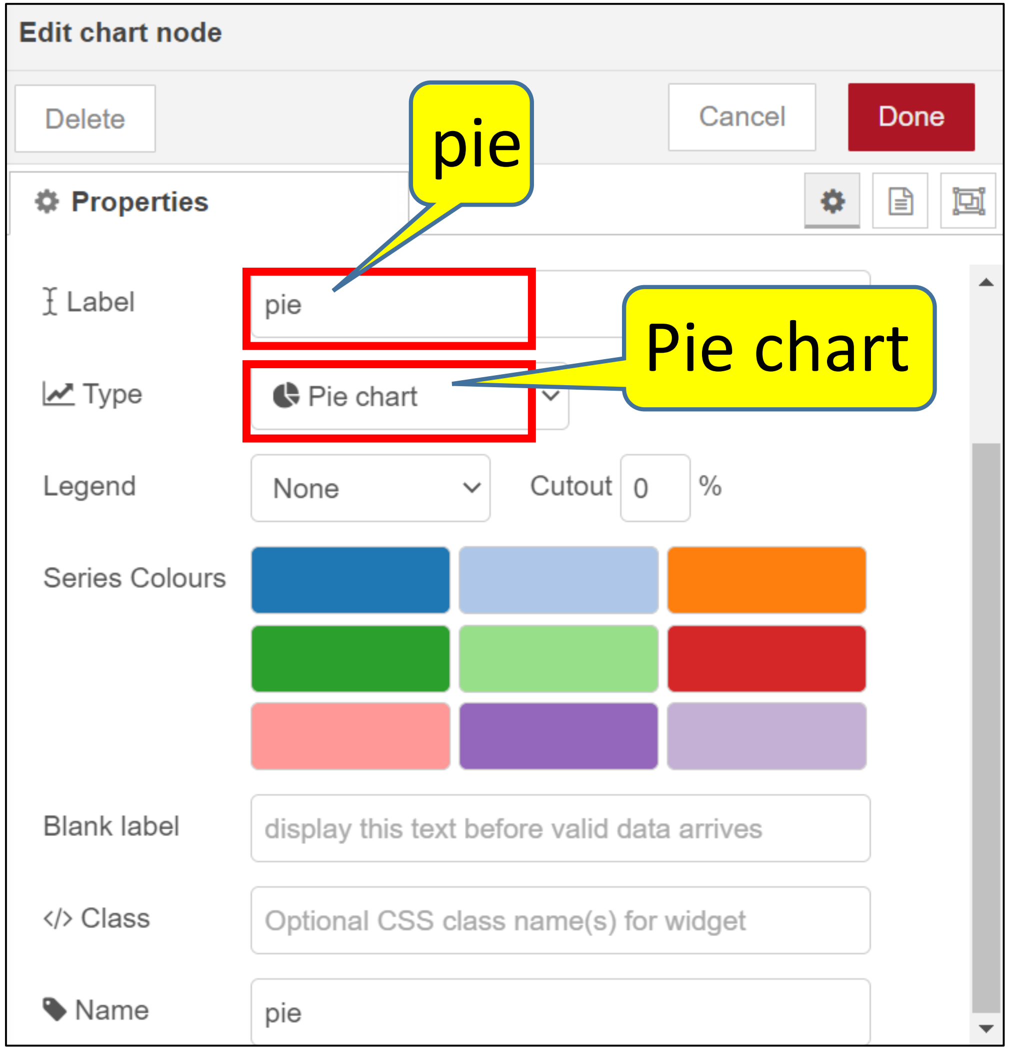
Global

10/18/2022, 1:41:39 PM

cou1	1
cou2	1
cou3	1

add another chart node





Label: pie

Type: Pie chart

Legend: None   Cutout 0 %

Series Colours: 🟦 🟦 🟧 🟩 🟩 🟥 🟥 🟪 🟪

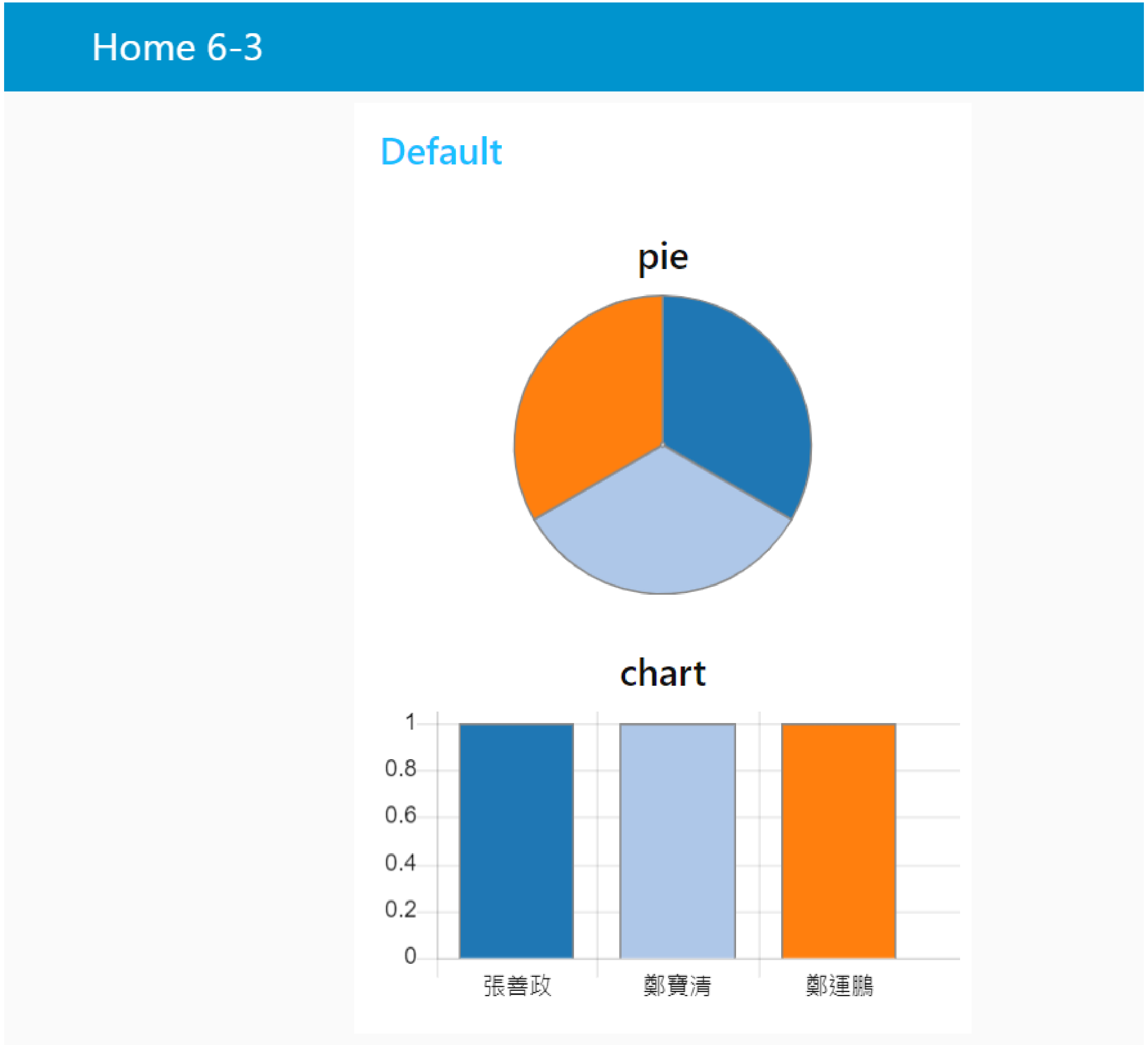
Blank label: display this text before valid data arrives

</> Class: Optional CSS class name(s) for widget

Name: pie

pie

Pie chart



Home 6-3

Default

pie

chart

Exercise 6-5

- Design a voting system for four candidates.

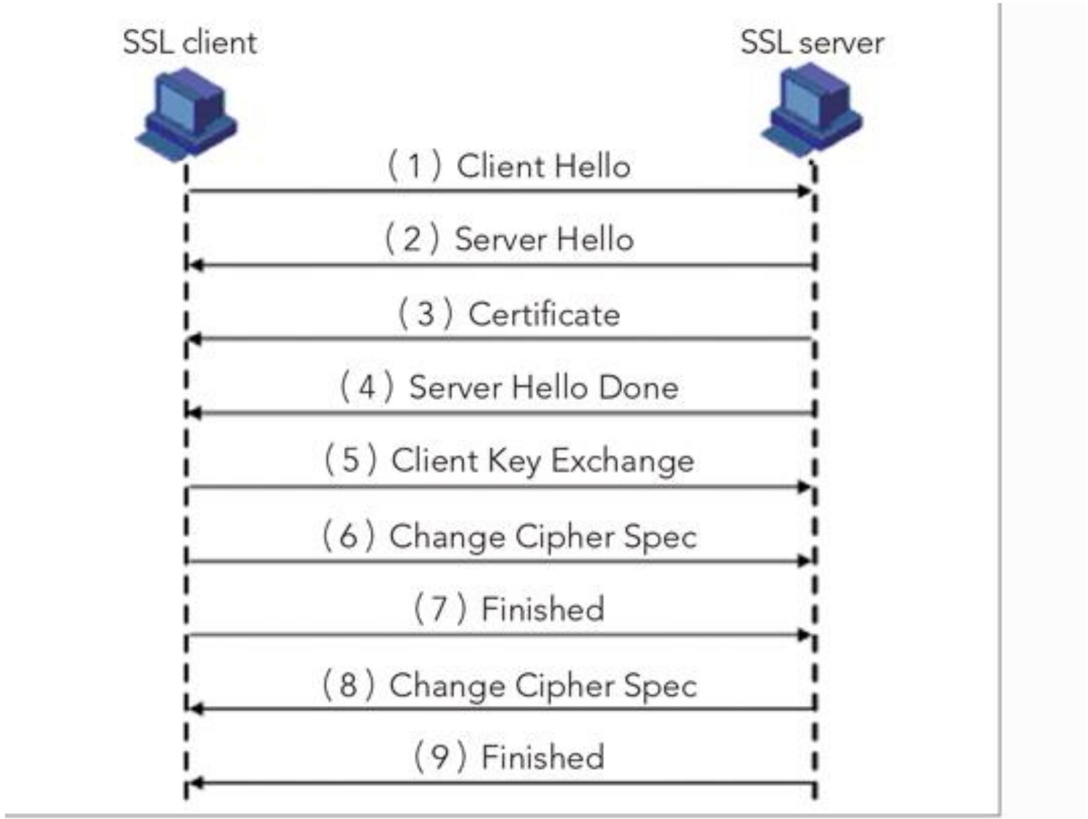
HTTPS

- HTTPS 為 TLS/SSL + HTTP，藉由 TLS/SSL 的加密機制，可保護資料不輕易外洩。TLS/SSL: 傳輸層安全性協定（Transport Layer Security，縮寫：TLS）及其前身安全通訊協定（Secure Sockets Layer，縮寫：SSL）。SSL 最初由網景公司(Netscape)發布，後來 IETF 將其標準化，並發布第一版 TLS。TLS 可說是 SSL 的後繼版本。

https://hackmd.io/@yyp/SyvtMbQlH?print-pdf#/

SSL Handshake

- *SSL是Secure Socket Layer（安全套接層協議）的縮寫，可以在Internet上提供秘密性傳輸。*



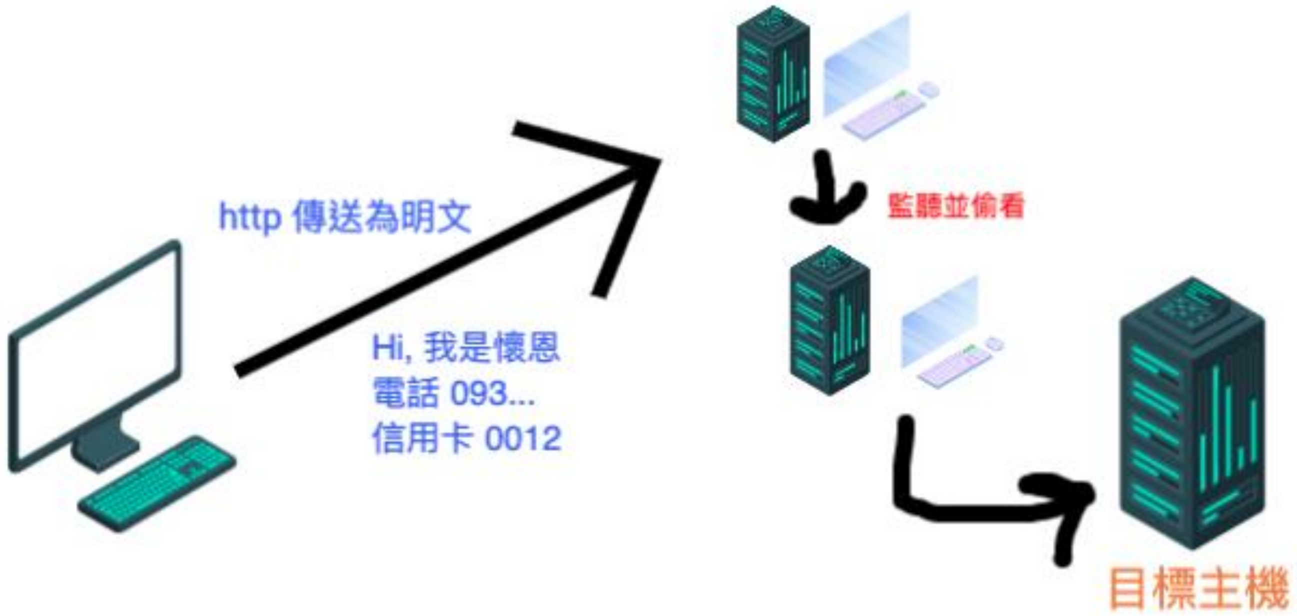
https://www.netadmin.com.tw/netadmin/zh-tw/technology/6F6D669EB83E4DC9BEA42F1C94636D46

Who is CA(Certificate Authority)?

- 誰是CA(Certificate Authority, 數位憑證認證機構)?
  AWS Certificate Manager https://aws.amazon.com/tw/certificate-manager/
  TWCA台灣網路認證 https://www.twca.com.tw/
  GRCA政府憑證總管理中心 https://grca.nat.gov.tw/
  而憑證的定價根據用途，一年期從免費到數萬元不等

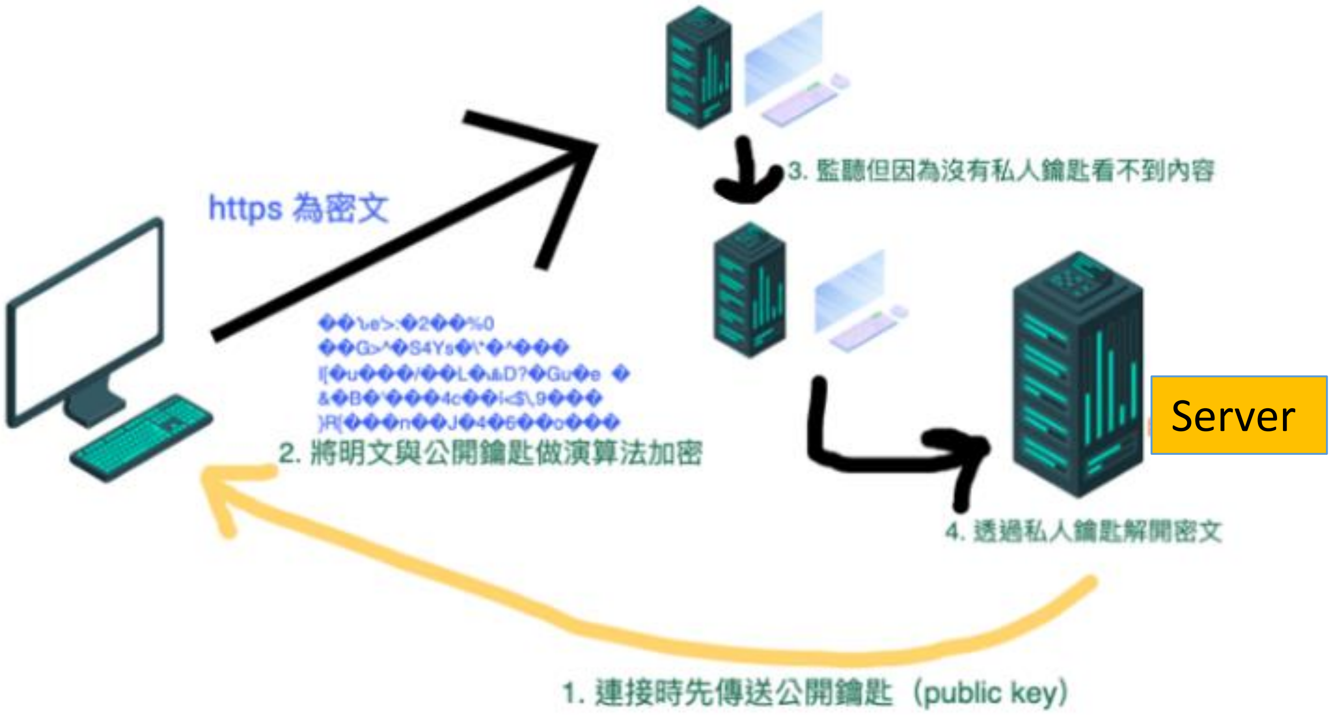
https://hackmd.io/@yyp/SyvtMbQlH?print-pdf#/

HTTP



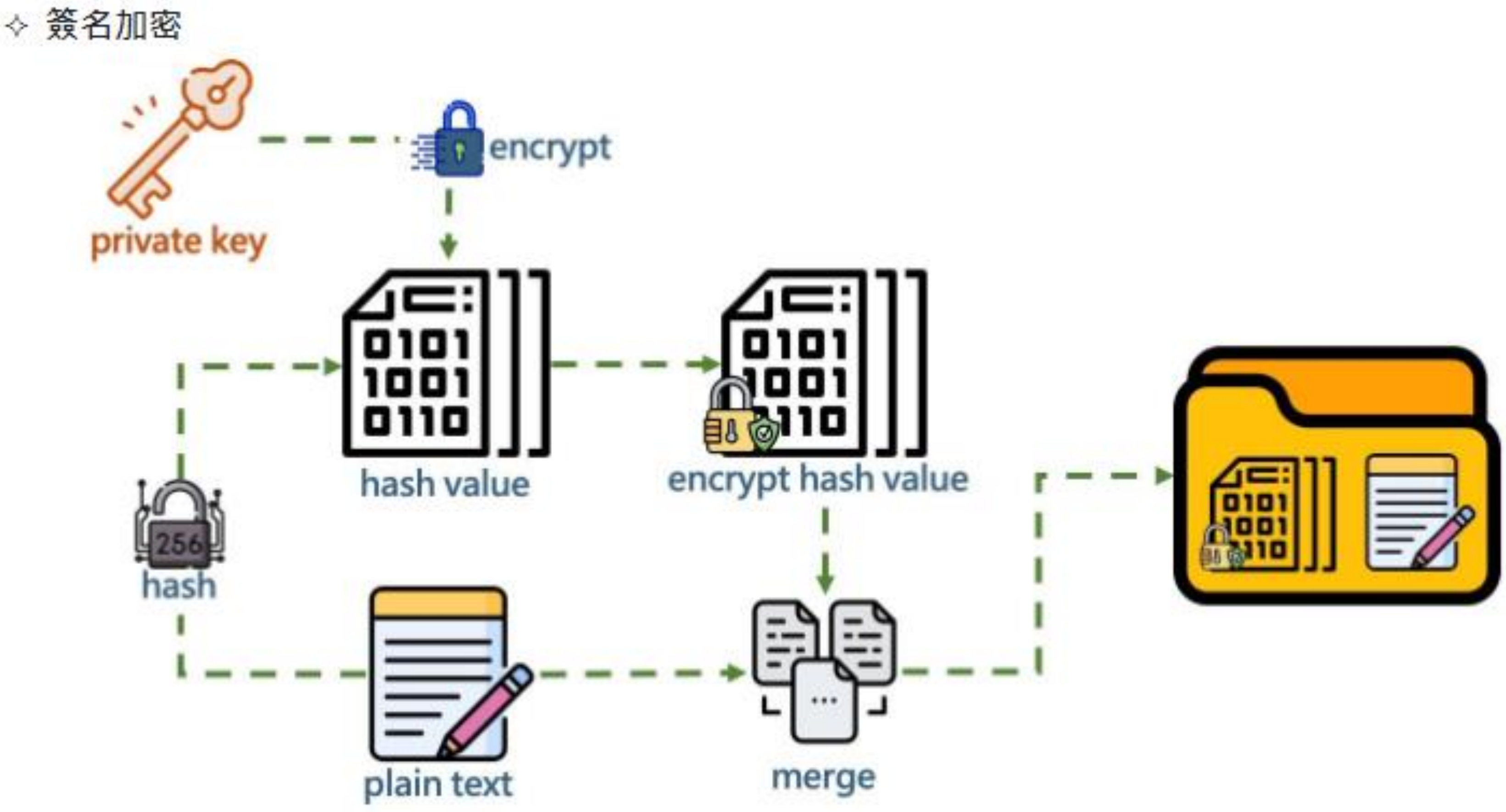
https://whien.medium.com/%E9%80%8F%E9%81%8E-nodejs-crypto-%E5%B9%AB%E5%8A%A9%E7%94%B7%E5%AD%A9%E8%88%87%E5%A5%B3%E5%AD%A9%E5%AE%8C%E6%88%90%E9%9D%9E%E5%B0%8D%E7%A8%B1%E5%8A%A0%E5%AF%86%E7%9A%84%E7%A7%81%E8%A8%8A%E5%82%B3%E8%BC%B8-207ccb54ec20

HTTPS


https 為密文
3. 監聽但因為沒有私人鑰匙看不到內容
Server
2. 將明文與公開鑰匙做演算法加密
4. 透過私人鑰匙解開密文
1. 連接時先傳送公開鑰匙（public key）

https://whien.medium.com/%E9%80%8F%E9%81%8E-nodejs-crypto-%E5%B9%AB%E5%8A%A9%E7%94%B7%E5%AD%A9%E8%88%87%E5%A5%B3%E5%AD%A9%E5%AE%8C%E6%88%90%E9%9D%9E%E5%B0%8D%E7%A8%B1%E5%8A%A0%E5%AF%86%E7%9A%84%E7%A7%81%E8%A8%8A%E5%82%B3%E8%BC%B8-207ccb54ec20

數位簽章



https://www.tpisoftware.com/tpu/articleDetails/2593