物聯網實務

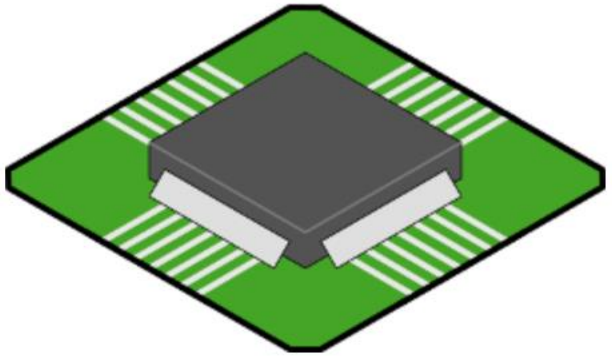
(三)

廖裕評

Node-RED

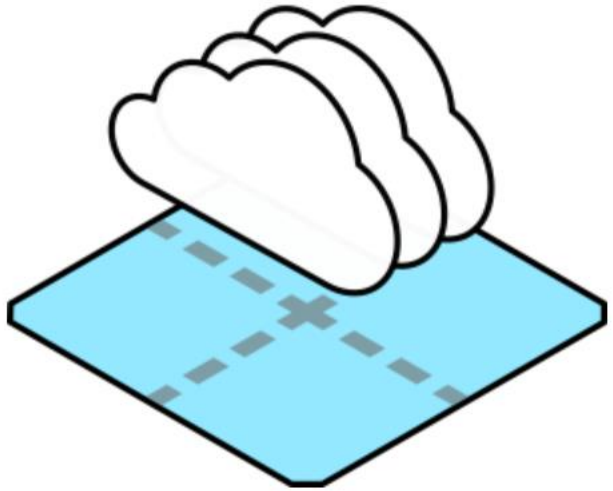
Run locally

- Getting started
- Docker



On a device

- Raspberry Pi
- BeagleBone Black
- Interacting with Arduino
- Android



In the cloud

- FlowFuse
- Amazon Web Services
- Microsoft Azure

https://nodered.org/

HTTP Request Methods

- https://www.w3schools.com/tags/ref_httpmethods.asp
- What is HTTP?
- The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.
- HTTP works as a request-response protocol between a client and server.
- Example: A client (browser) sends an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

Http request–response

A Customer

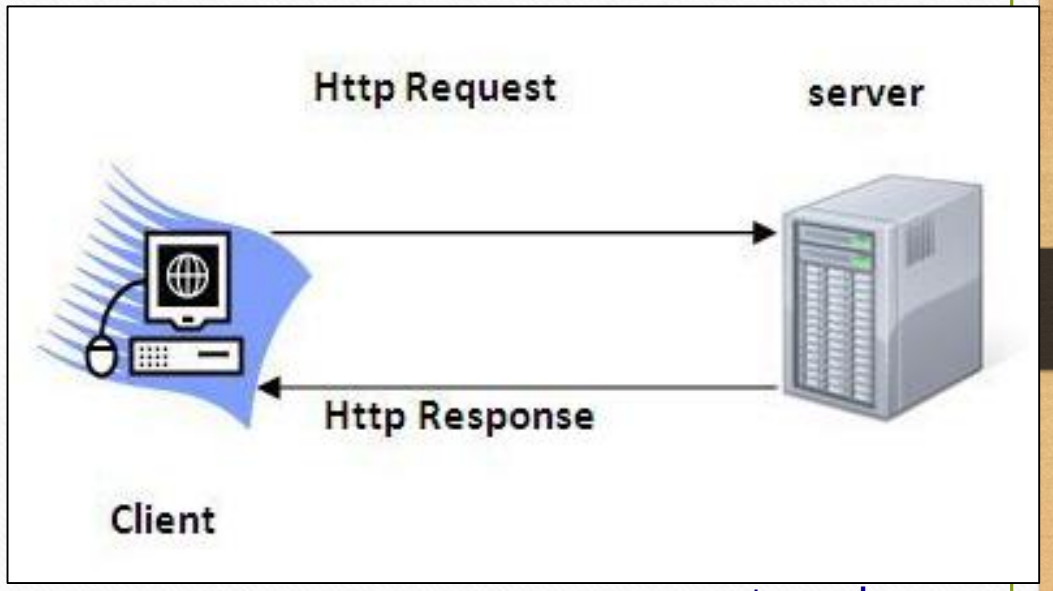
I am sorry about that

Response

request

I want to complain

A Support Center



Web browser

tw.yahoo.com

HTTP

(HyperText Transfer Protocol)

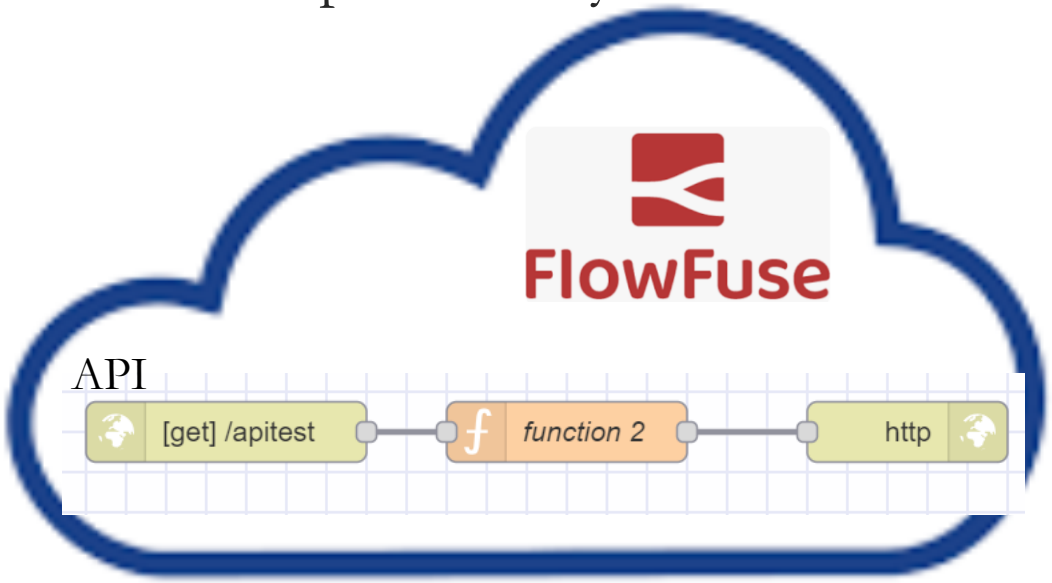
Exercise 3-1

- Communication between a browser and an API provided by FlowFuse

Local

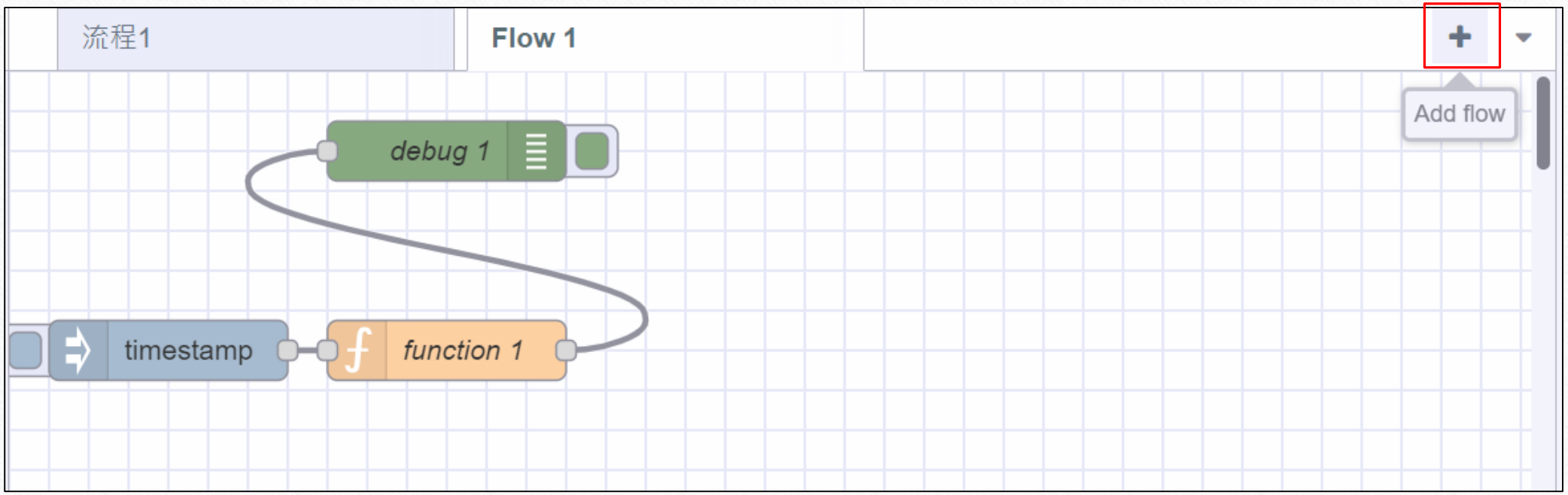
HTTP GET Request

Response



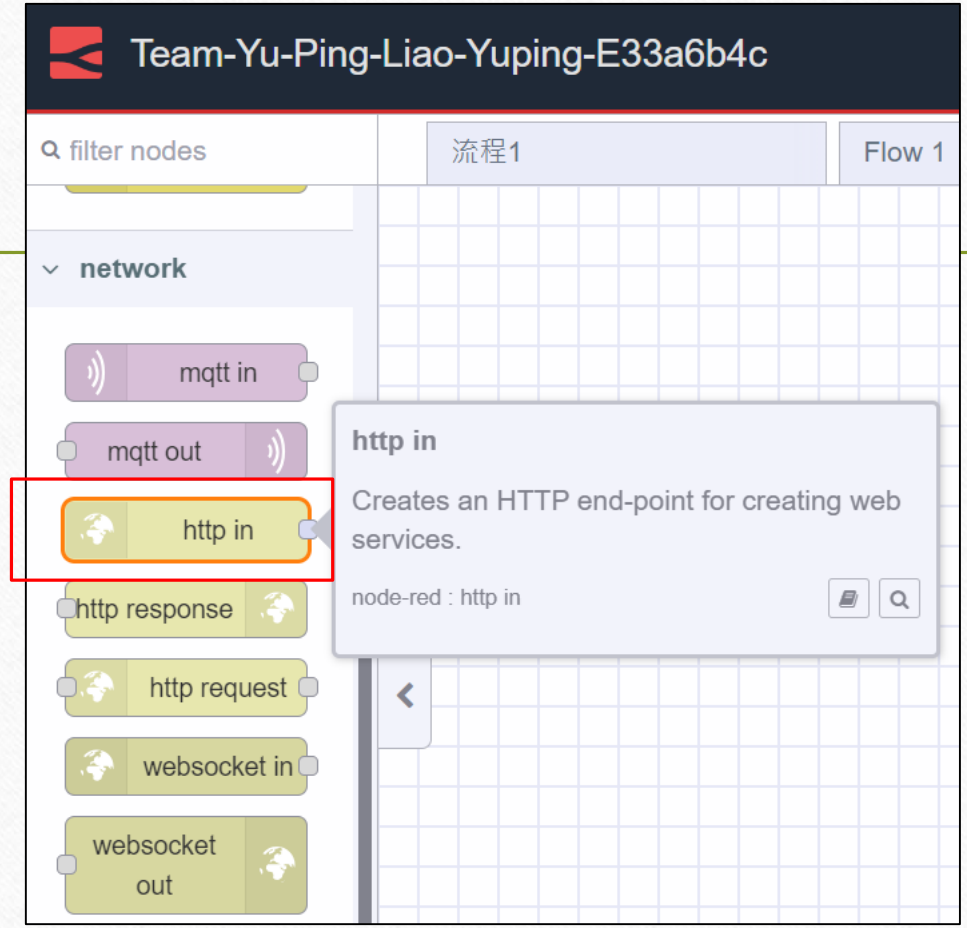
https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview

Add flow in FlowFuse



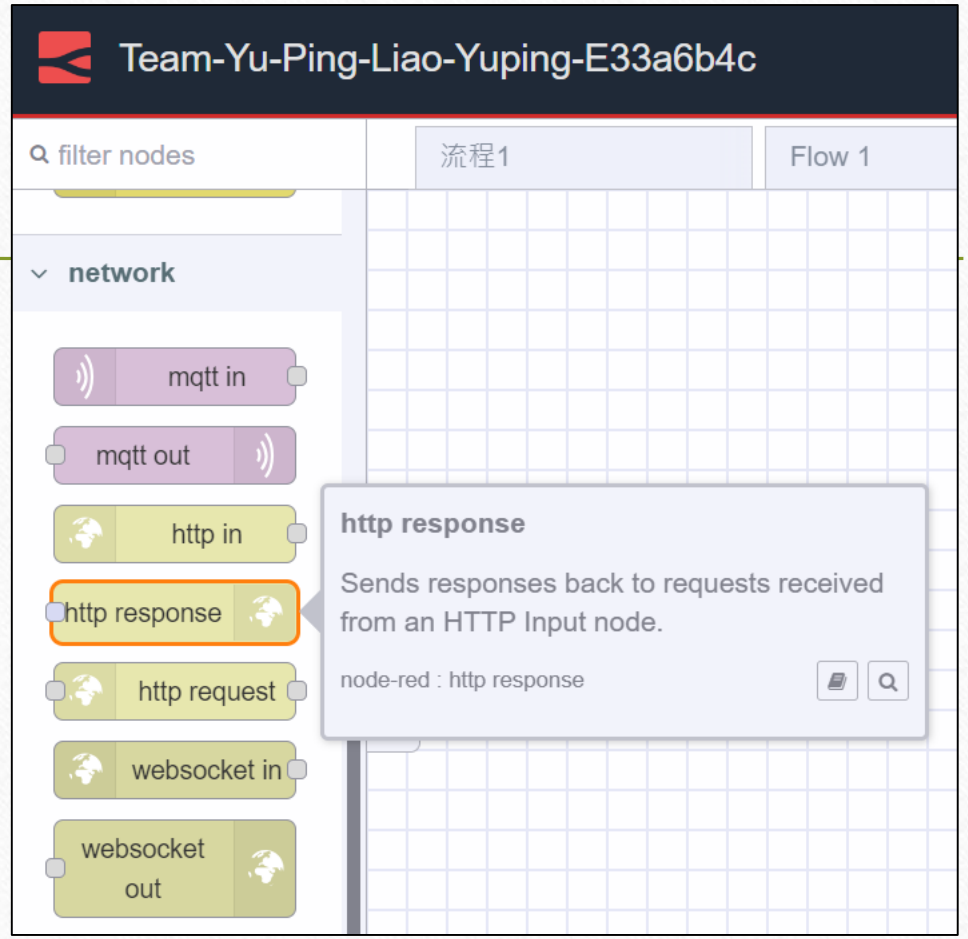
http in

- Creates an HTTP end-point for creating web services.

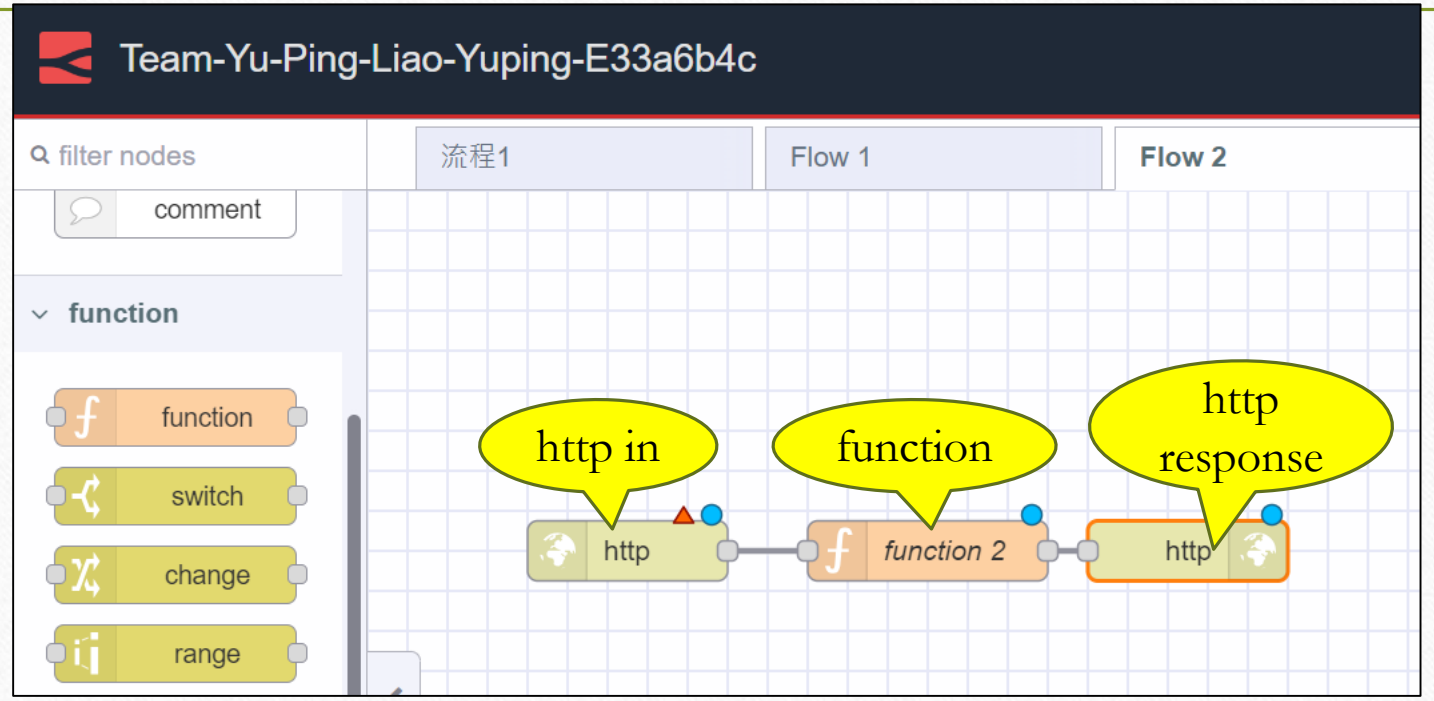


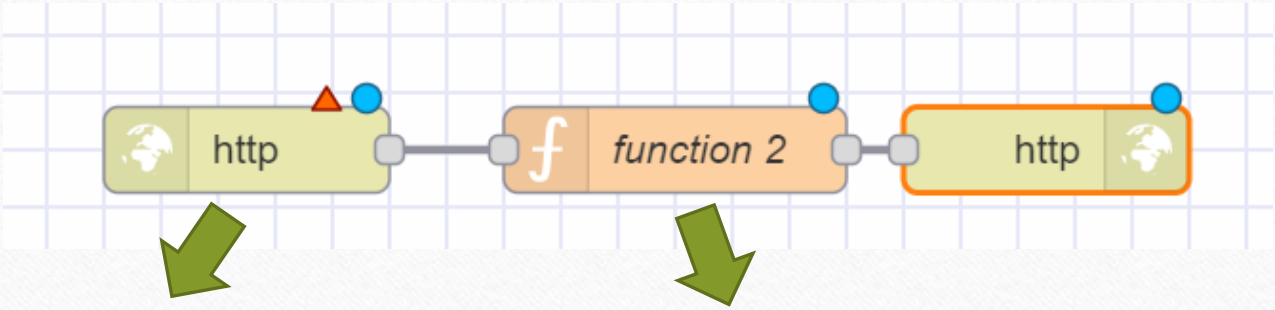
http response

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



Add ‘http in’, ‘function’, ‘http response’ nodes





Edit http in node

Delete | Cancel | Done

Properties

Method: GET

/apitest

URL: /apitest

Name: Name

Edit function node

Delete | Cancel | Done

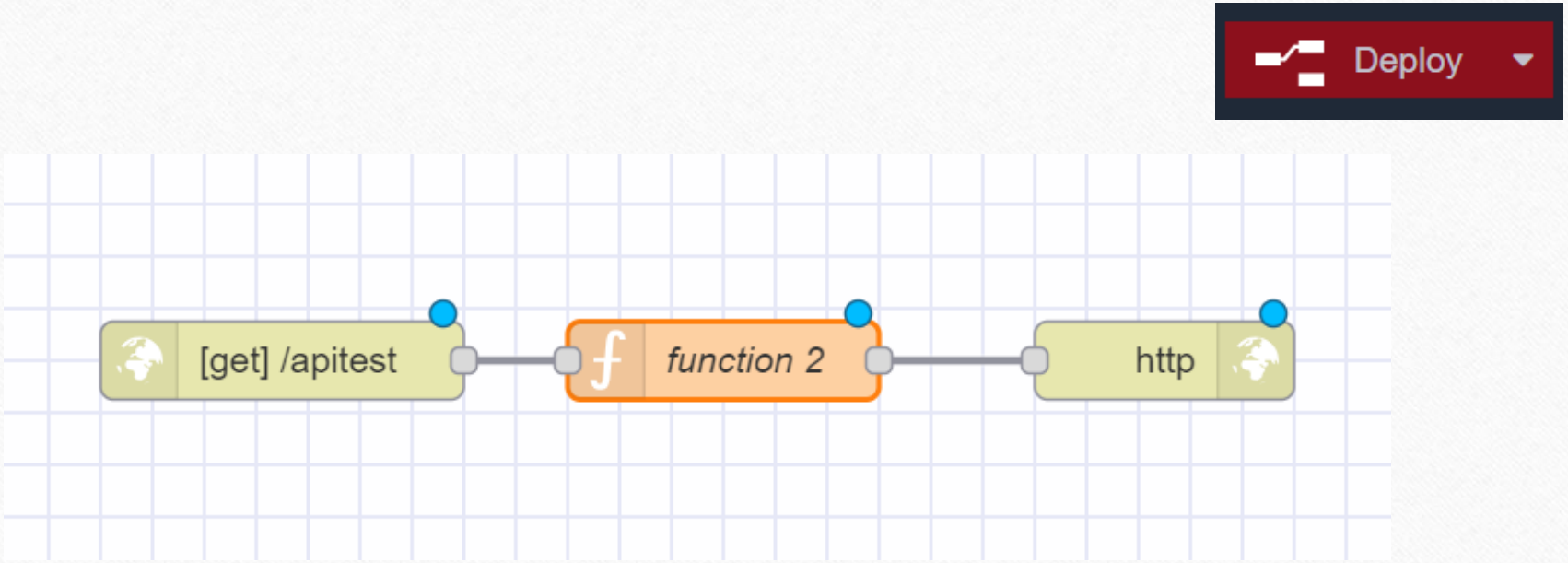
Properties

Name: function 2

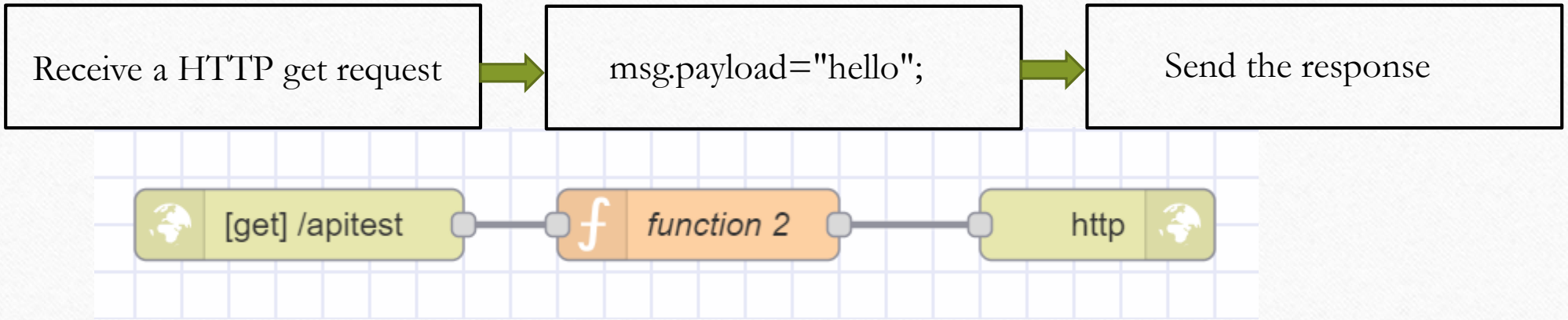
Setup | On Start | On Message | On Stop

```
1  msg.payload="hello";
2  return msg;
```

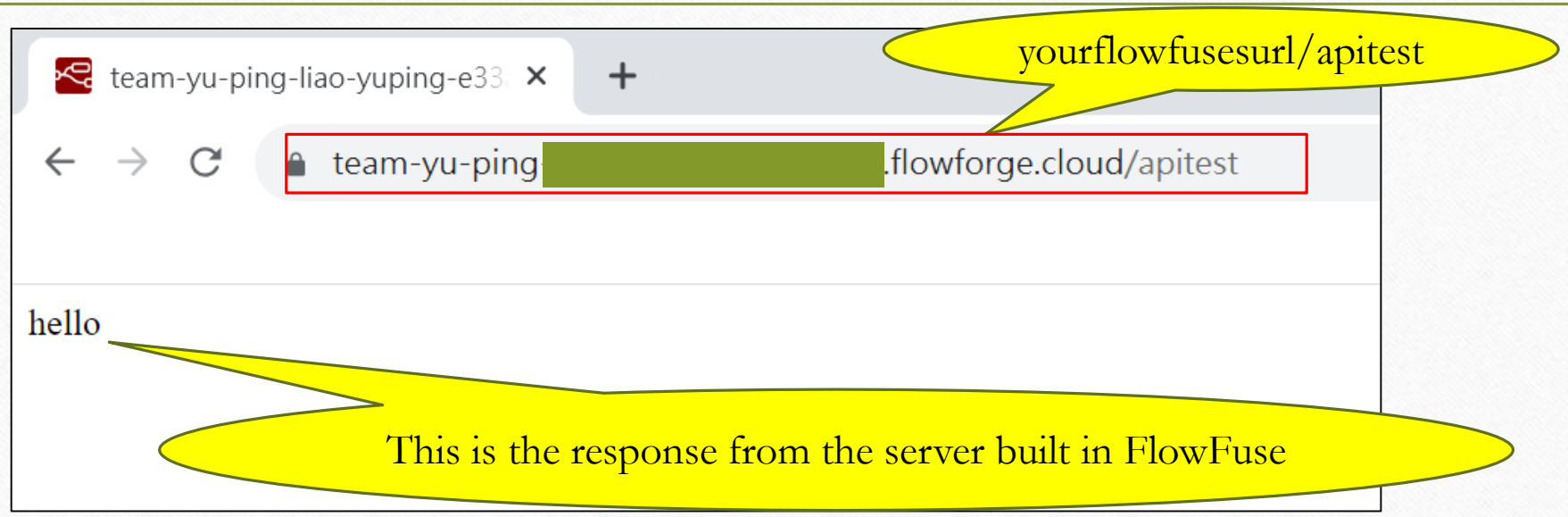
Deploy



Flow chart


Receive a HTTP get request
msg.payload="hello";
Send the response
[get] /apitest
function 2
http

Test with a browser



Exercise 3-2

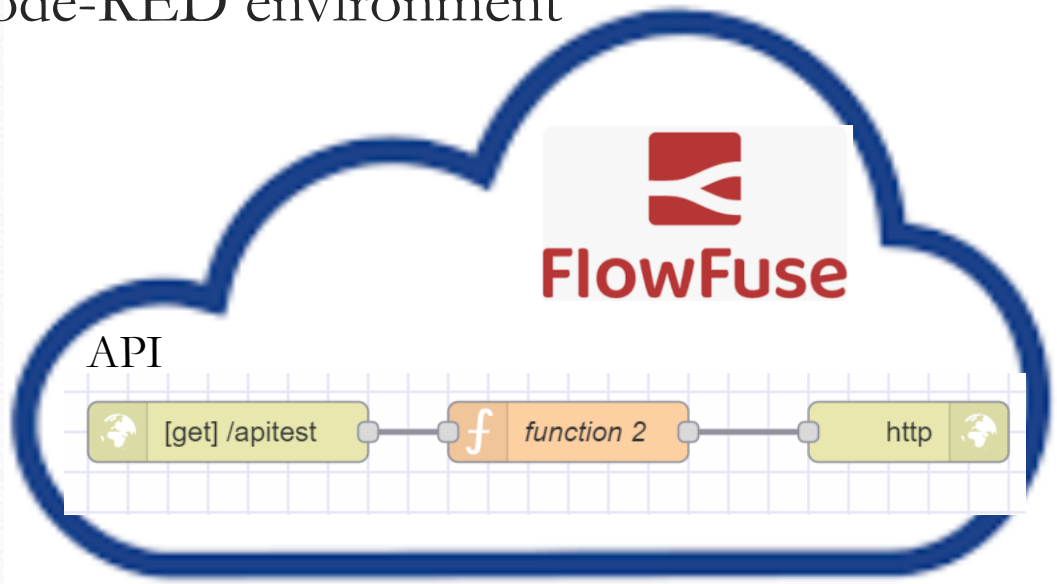
- Send the http request with local Node-RED environment

Local Node-RED

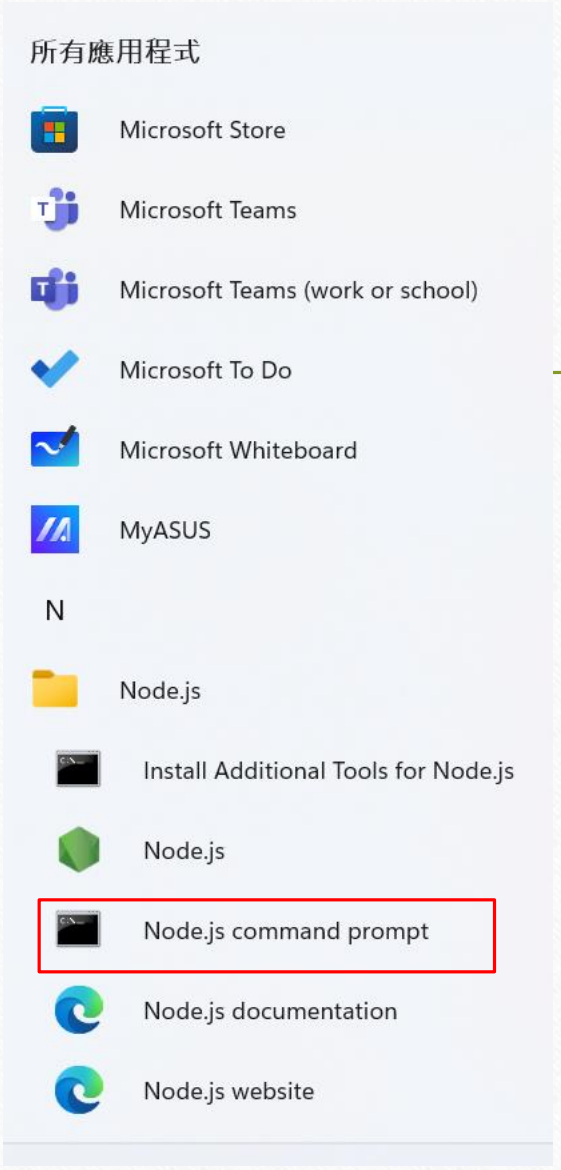
HTTP GET

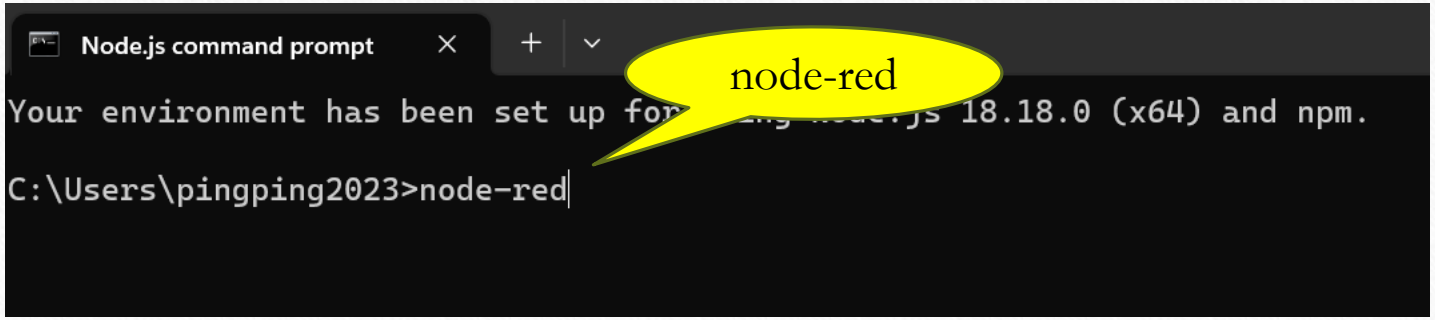
Request

Response

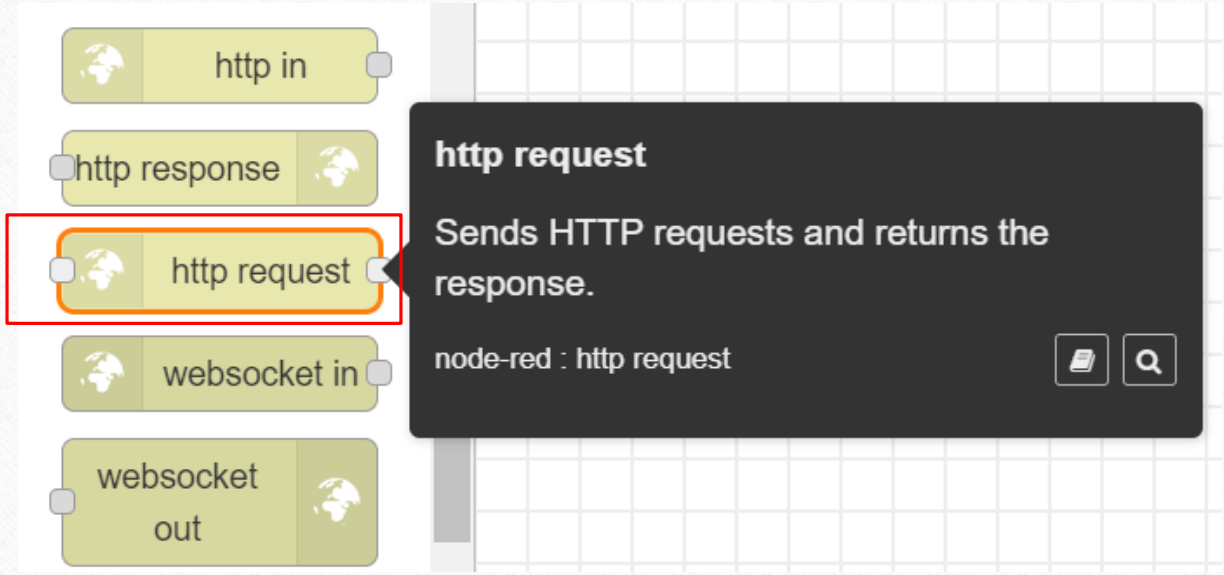


Run locally

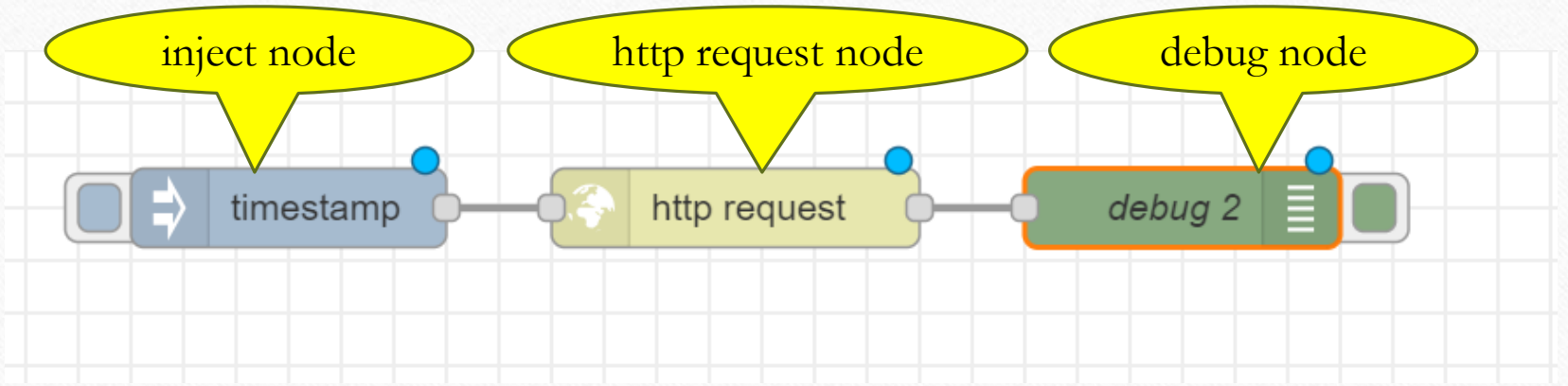

所有應用程式
Microsoft Store
Microsoft Teams
Microsoft Teams (work or school)
Microsoft To Do
Microsoft Whiteboard
MyASUS
N
Node.js
Install Additional Tools for Node.js
Node.js
Node.js command prompt
Node.js documentation
Node.js website

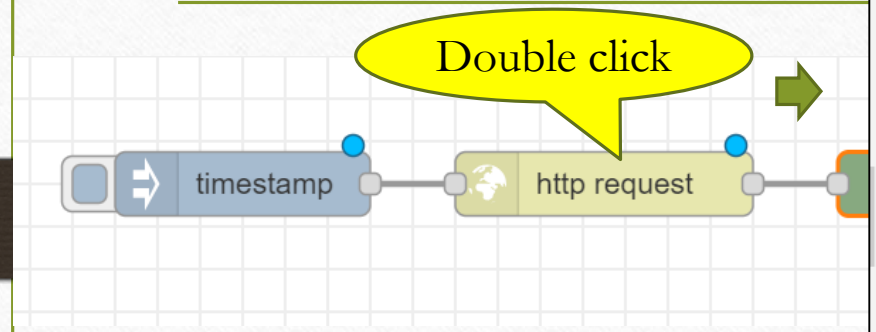

Node.js command prompt
node-red
Your environment has been set up fo
18.18.0 (x64) and npm.
C:\Users\pingping2023>node-red

http request



Add nodes


inject node
http request node
debug node
timestamp
http request
debug 2



Edit http request node

Delete | Cancel | Done

Properties

yourflowfusesurl/apitest

Method: GET

URL: https://team-yu-ping-l[redacted].cloud/apitest

Payload: Ignore

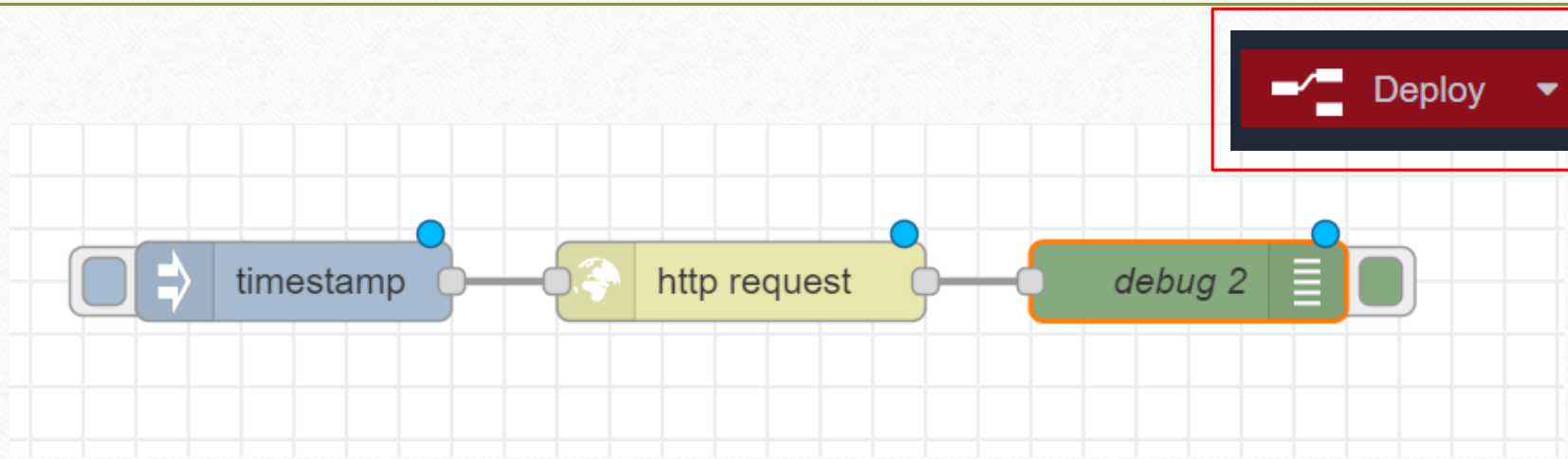
- [ ] Enable secure (SSL/TLS) connection
- [ ] Use authentication
- [ ] Enable connection keep-alive
- [ ] Use proxy
- [ ] Only send non-2xx responses to Catch node
- [ ] Disable strict HTTP parsing

Return: a UTF-8 string

Headers

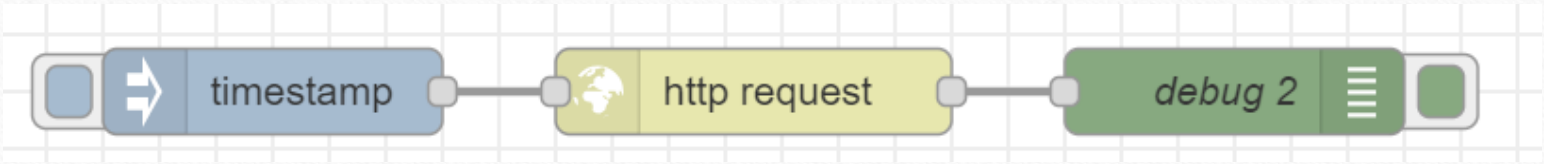
Enabled

Deploy

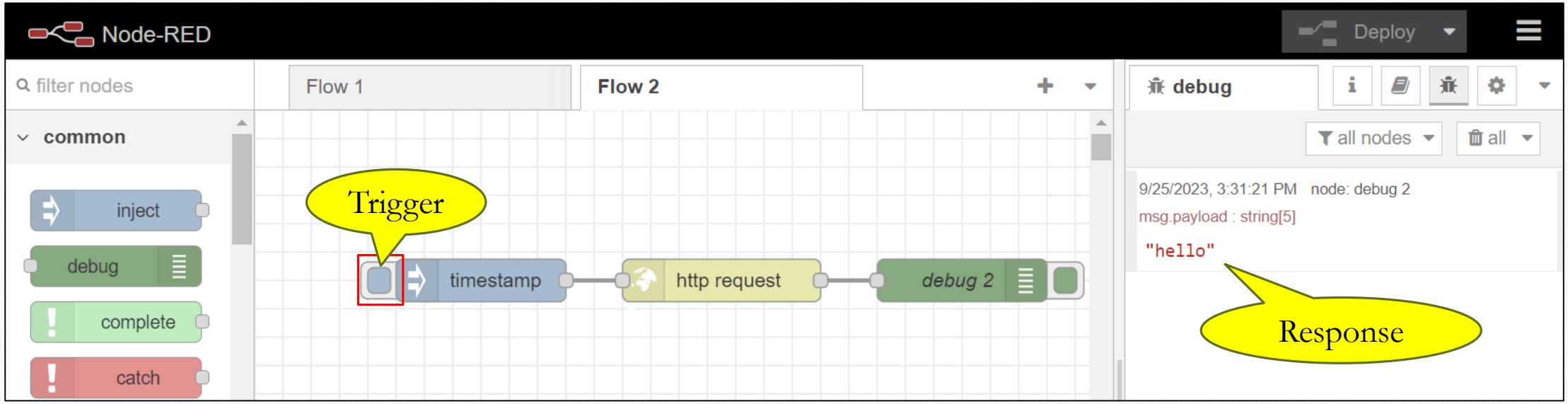


Flow chart


Trigger
Send HTTP GET request
Show the response from the server

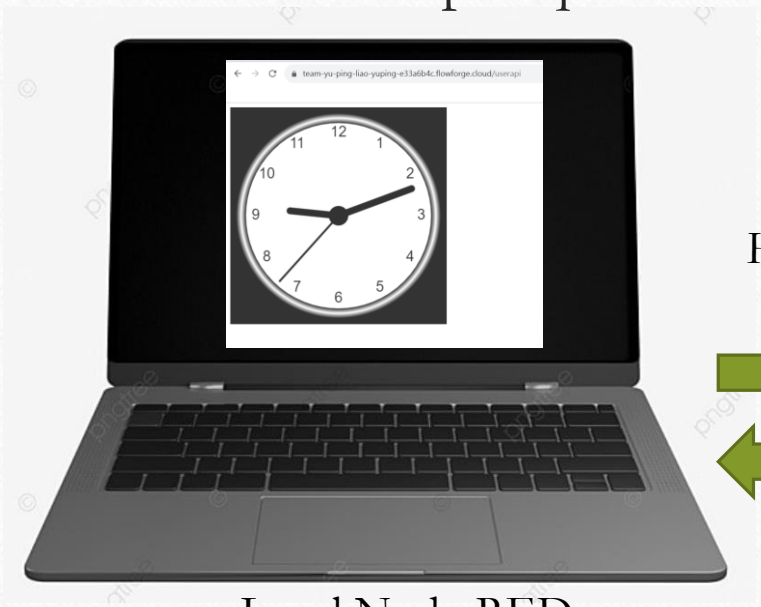

timestamp
http request
debug 2

Trigger


Node-RED
Deploy
filter nodes
Flow 1
Flow 2
common
inject
debug
complete
catch
Trigger
timestamp
http request
debug 2
debug
all nodes
all
9/25/2023, 3:31:21 PM node: debug 2
msg.payload : string[5]
"hello"
Response

Exercise 3-3

- Send the http request with the local Node-RED environment

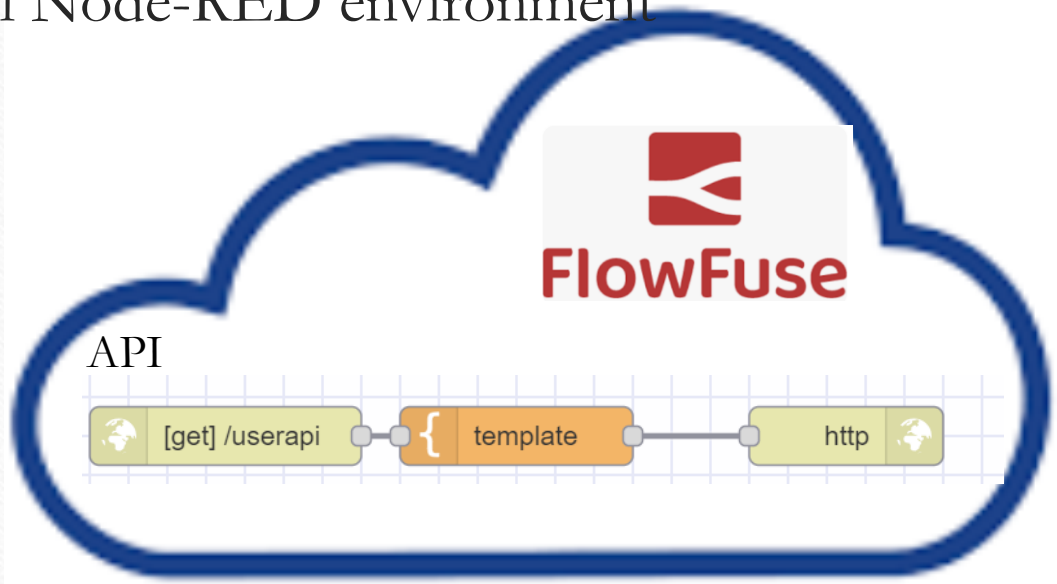


Local Node-RED

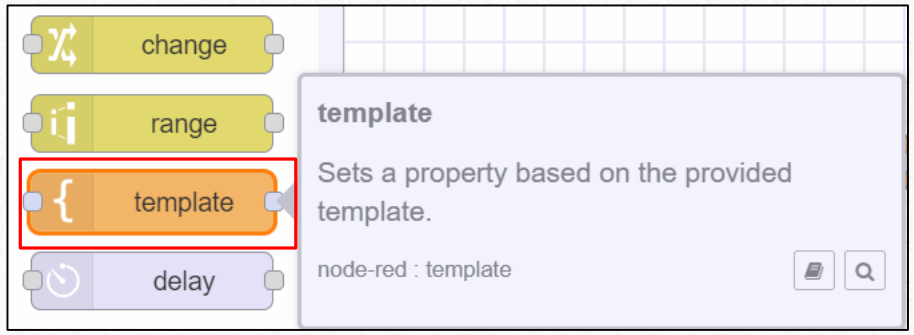
HTTP GET

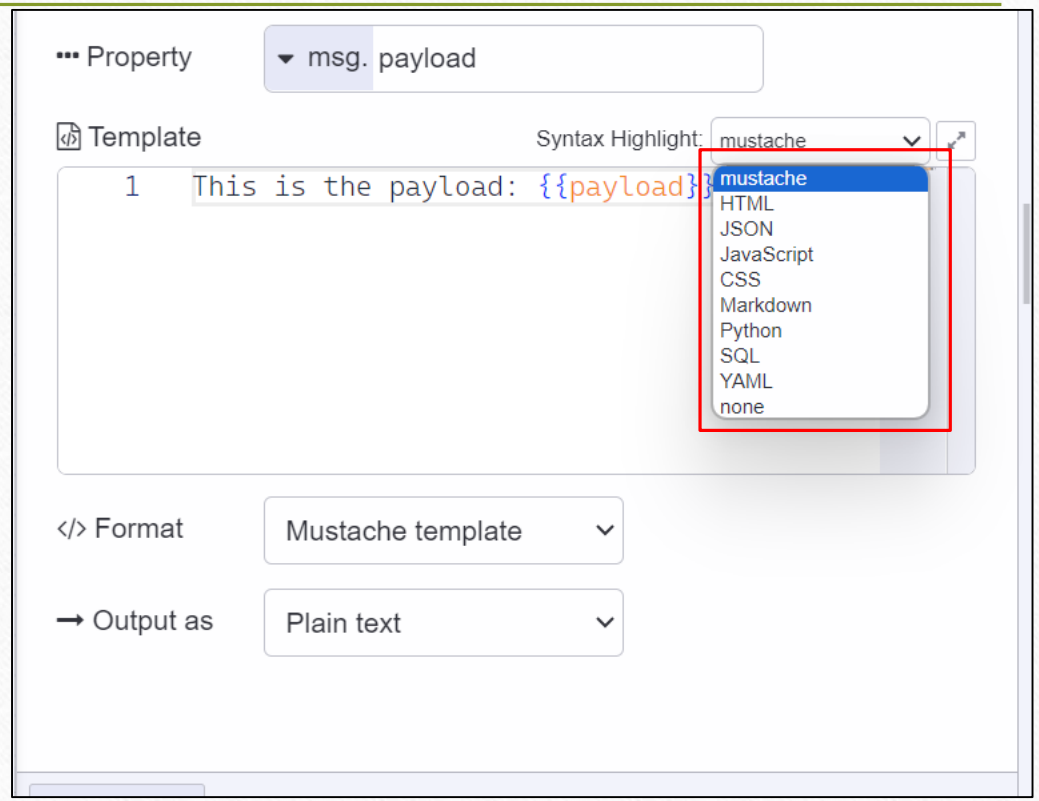
Request

Response

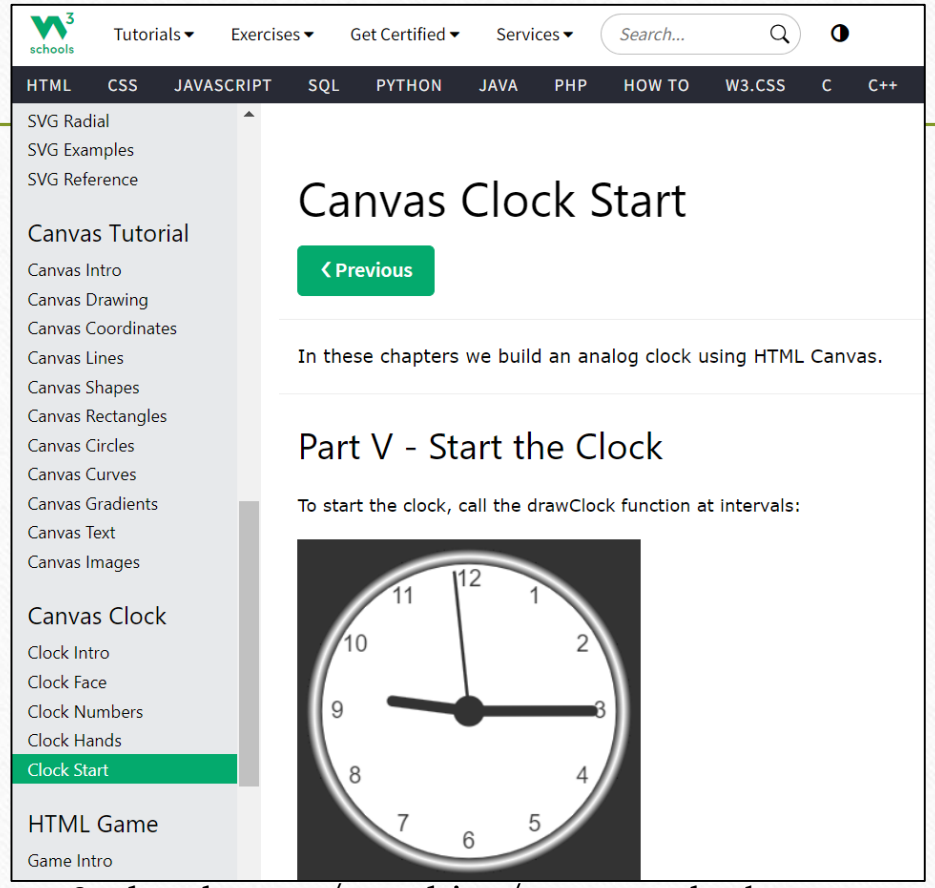


Template node





Clock



https://www.w3schools.com/graphics/canvas_clock_start.asp

To start the clock, call the drawClock function at intervals:

JavaScript:

```
const canvas = document.getElementById("canvas");
const ctx = canvas.getContext("2d");
let radius = canvas.height / 2;
ctx.translate(radius, radius);
radius = radius * 0.90
//drawClock();
setInterval(drawClock, 1000);
```

Try it Yourself »

Copy

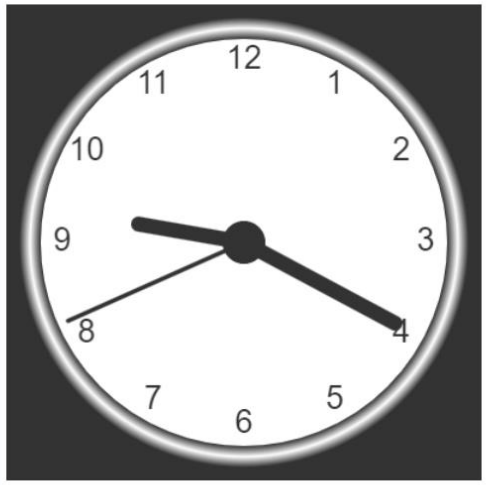
Run ›

Result Size: 785 x 649

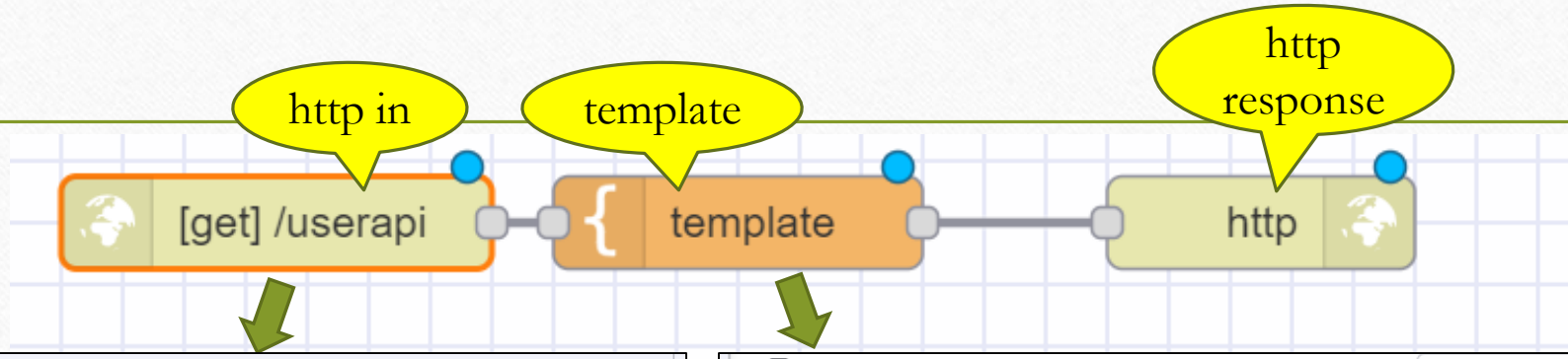
Get your own website

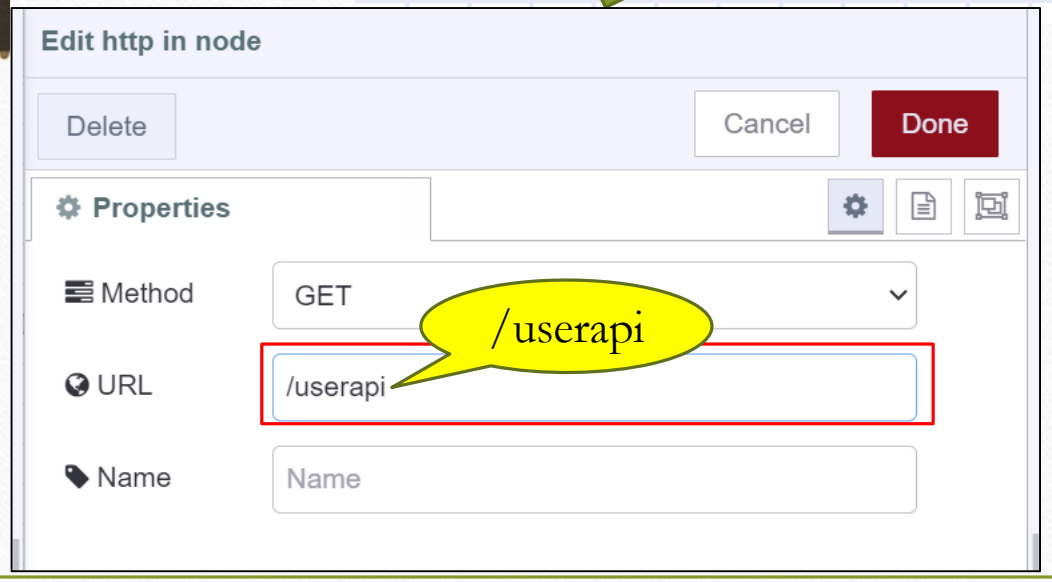
```
<!DOCTYPE html>
<html>
<body>

<canvas id="canvas" width="400" height="400"
style="background-color:#333">
</canvas>

<script>
const canvas = document.getElementById("canvas");
const ctx = canvas.getContext("2d");
let radius = canvas.height / 2;
ctx.translate(radius, radius);
radius = radius * 0.90
setInterval(drawClock, 1000);

function drawClock() {
  drawFace(ctx, radius);
  drawNumbers(ctx, radius);
  drawTime(ctx, radius);
}

function drawFace(ctx, radius) {
  const grad = ctx.createRadialGradient(0,0,radius*0.95, 0,0,radius*1.05);
  grad.addColorStop(0, '#333');
  grad.addColorStop(0.5, 'white');
  grad.addColorStop(1, '#333');
  ctx.beginPath();
  ctx.arc(0, 0, radius, 0, 2*Math.PI);
  ctx.fillStyle = 'white';
  ctx.fill();
  ctx.strokeStyle = grad;
  ctx.lineWidth = radius*0.1;
  ctx.stroke();
  ctx.beginPath();
```

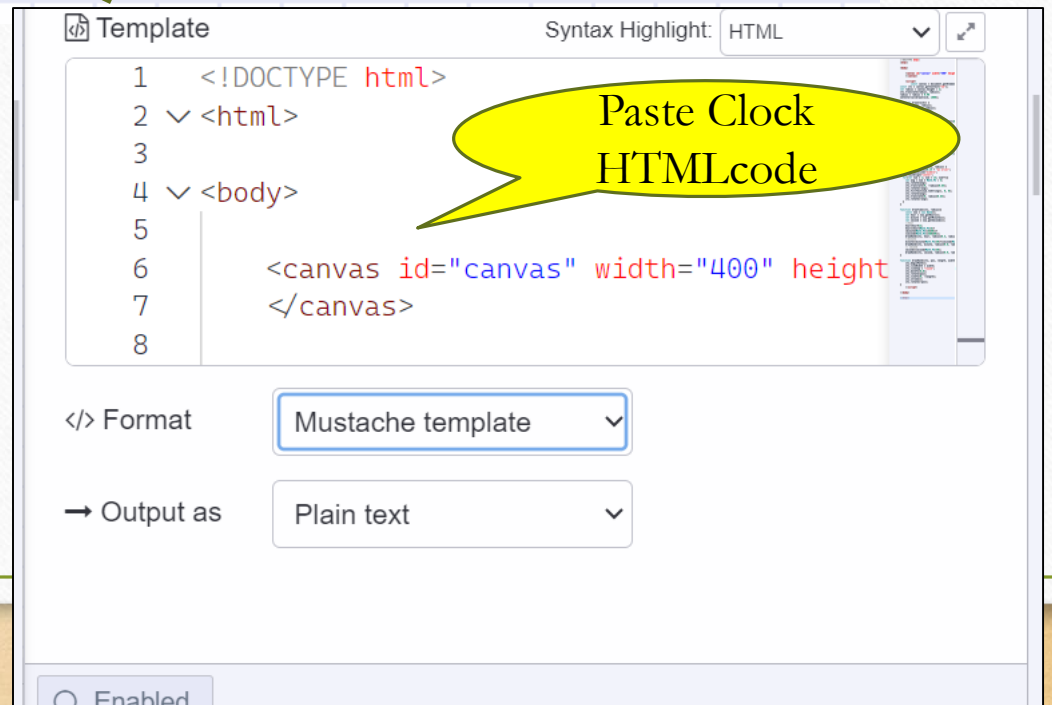
Copy



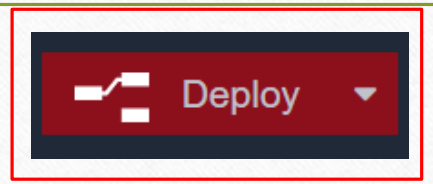
Add nodes

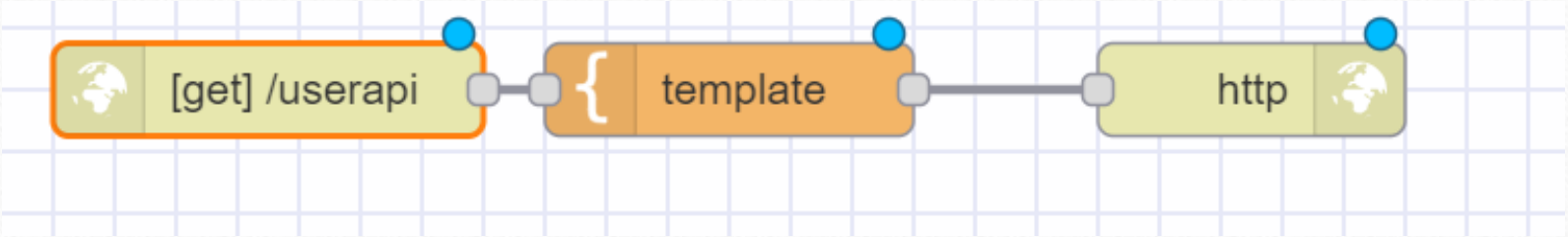

http in
template
http response
[get] /userapi
template
http


Edit http in node
Delete
Cancel
Done
Properties
Method
GET
/userapi
URL
/userapi
Name
Name

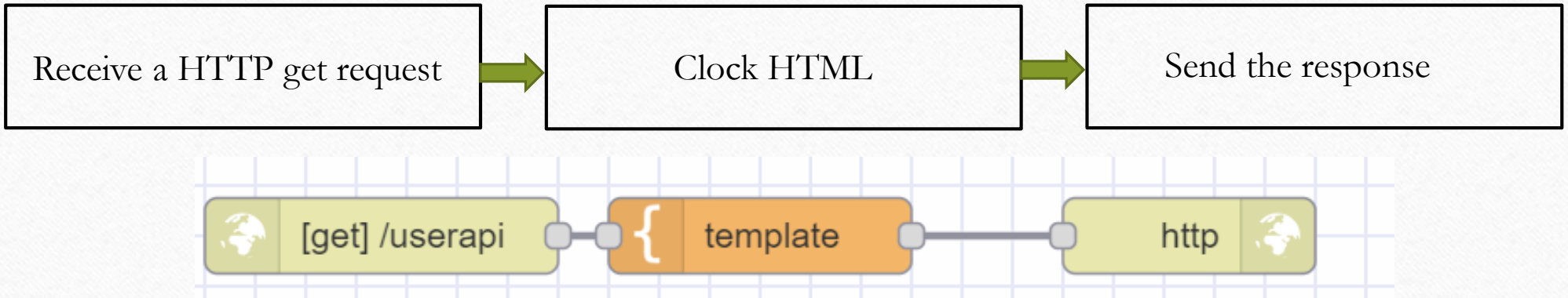

Template
Syntax Highlight:
HTML
1 <!DOCTYPE html>
2 <html>
3
4 <body>
5
6 <canvas id="canvas" width="400" height
7 </canvas>
8
Paste Clock HTMLcode
Format
Mustache template
Output as
Plain text
Enabled

Deploy

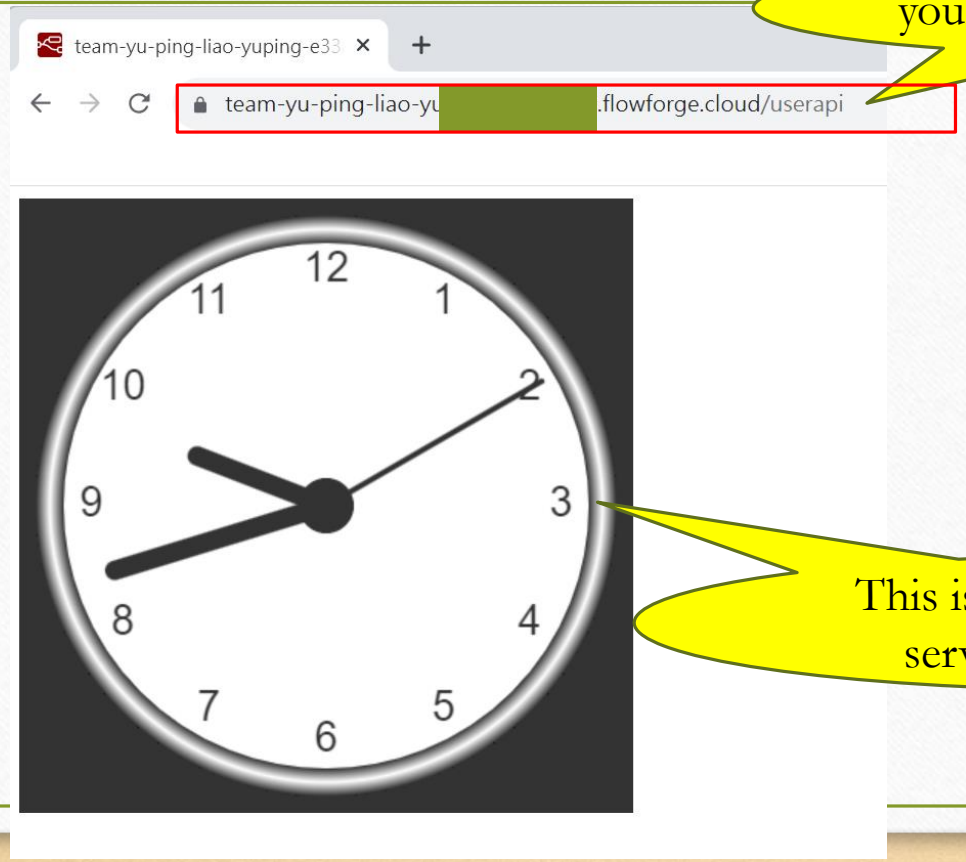




Flow chart


Receive a HTTP get request
Clock HTML
Send the response
[get] /userapi
template
http

Test with a browser



Homework 3-1

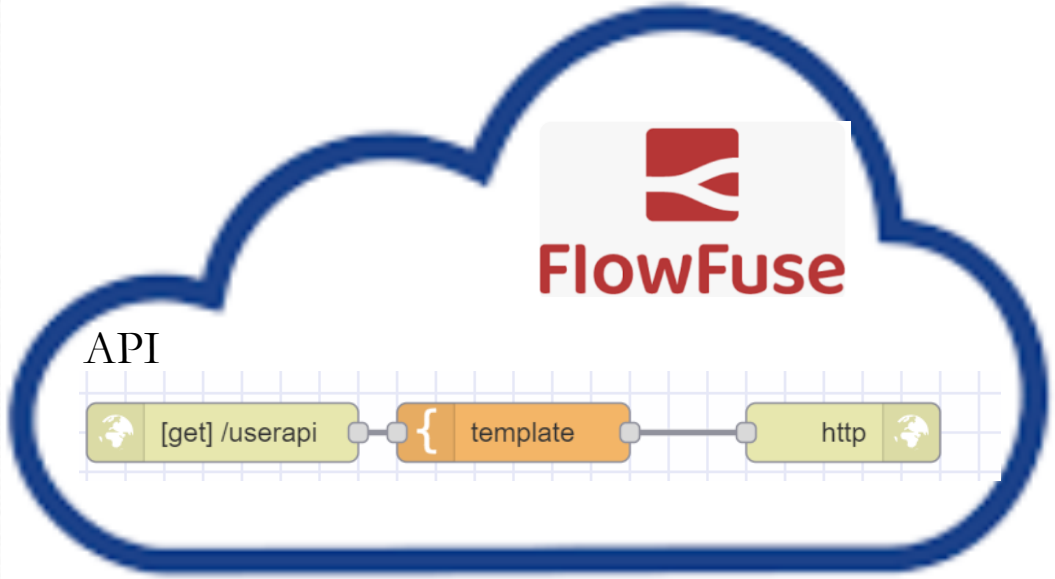
- Test with your local Node-RED

Local Node-RED

HTTP GET

Request

Response



Homework 3-2

- Read the paper: https://ieeexplore.ieee.org/document/9678449
- **Design of Humidity Control with Automatic Drip Irrigation System Based on Fuzzy Logic Using Node-RED and MQTT on Cactus Plants**

Top 10 programming languages in 2023

Position	PYPL ranking September 2023	Stack Overflow's Developer Survey 2023
#1	Python	JavaScript
#2	Java	HTML/CSS
#3	JavaScript	Python
#4	C#	SQL
#5	C/C++	TypeScript
#6	PHP	Bash/Shell
#7	R	Java
#8	TypeScript	C#

https://www.stackscale.com/blog/most-popular-programming-languages/

Top 20 Best Programming Languages To Learn in 2023

1. Javascript

JavaScript is a high-level programming language that is one of the core technologies of the World Wide Web. It is used as a client-side programming language by 97.8 percent of all websites. JavaScript was originally used only to develop web browsers, but they are now used for server-side website deployments and non-web browser applications as well.

Javascript was created in 1995 and was initially known as LiveScript. However, Java was a very popular language at that time, so it was advertised as a “younger brother” of Java. As it evolved over time, JavaScript became a fully independent language. Nowadays, JavaScript is often confused with Java, and although there are some similarities between them, the two languages are distinct.

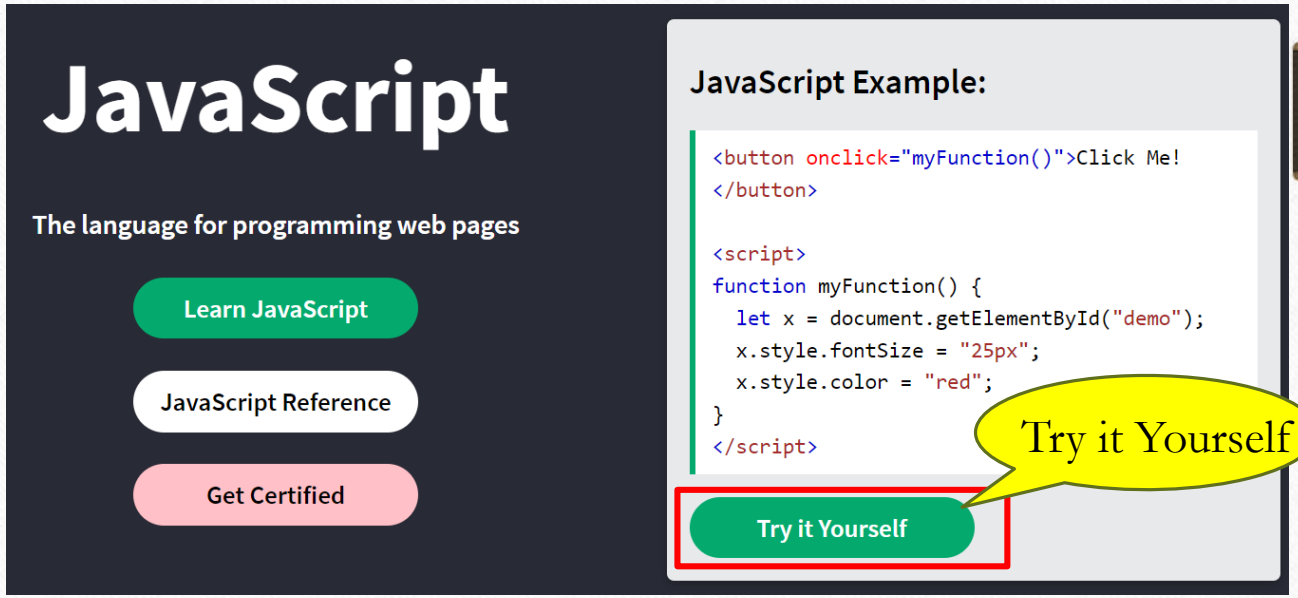
Also Read: Top 10 Reasons to Learn JavaScript

Javascript is the most popular programming language in the world and is in high demand among various organizations. The average Java developer earns around $112,152 each year.

https://www.simplilearn.com/best-programming-languages-start-learning-today-article

An Introduction to JavaScript

- JavaScript is the programming language of HTML and the Web.
- JavaScript is easy to learn.
- With "Try it Yourself" editor on https://www.w3schools.com/, you can edit the source code and view the result.



https://www.w3schools.com/

Click the button

Run ›

Result Size: 410 x 366

Get your website

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

<p id="demo">Click the button to change the
layout of this paragraph</p>

<button onclick="myFunction()">Click Me!
</button>

<script>
function myFunction() {
  let x = document.getElementById("demo");
  x.style.fontSize = "25px";
  x.style.color = "red";
}
</script>

</body>
</html>
```

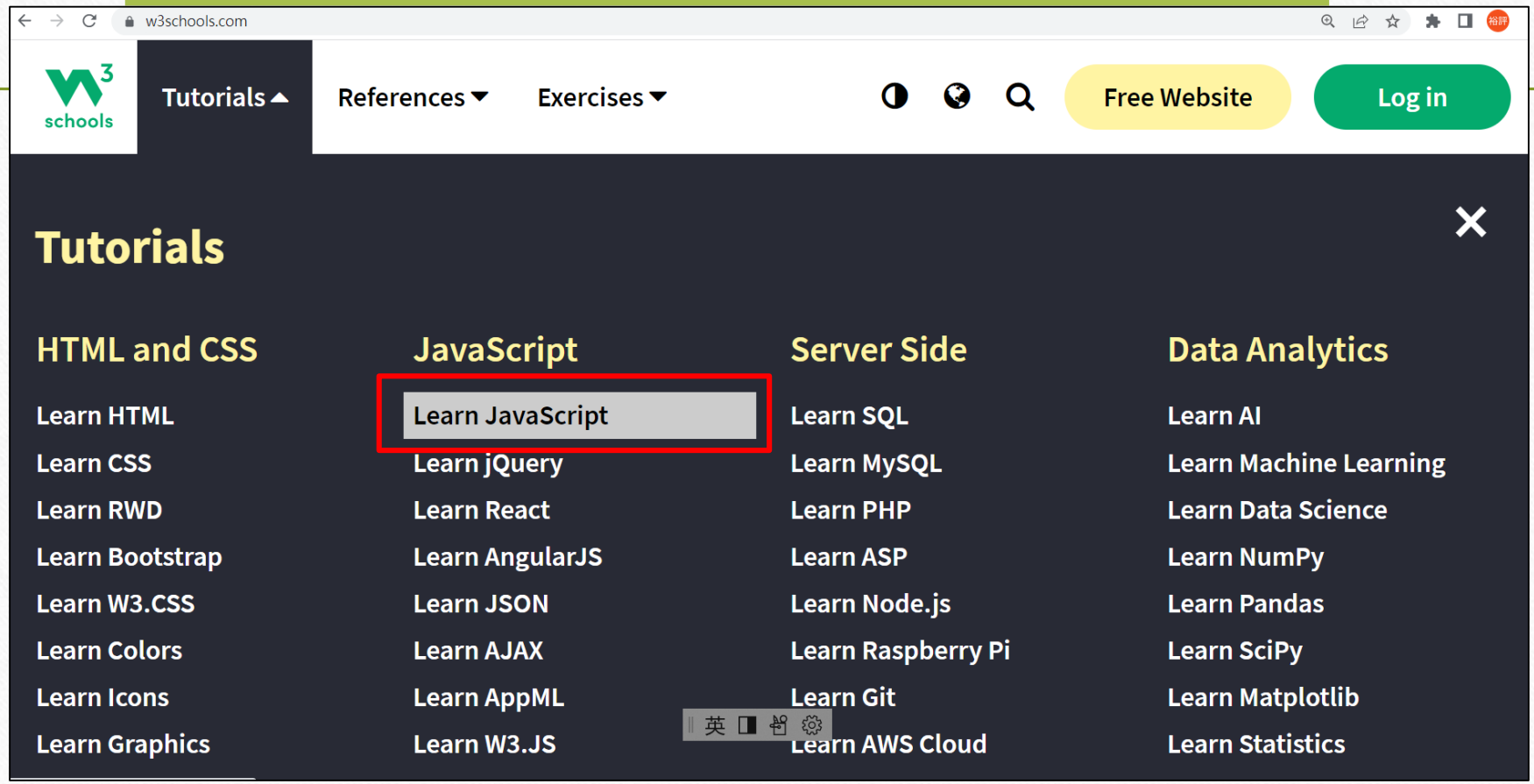
Click the button to change the layout of this paragraph

Click Me!

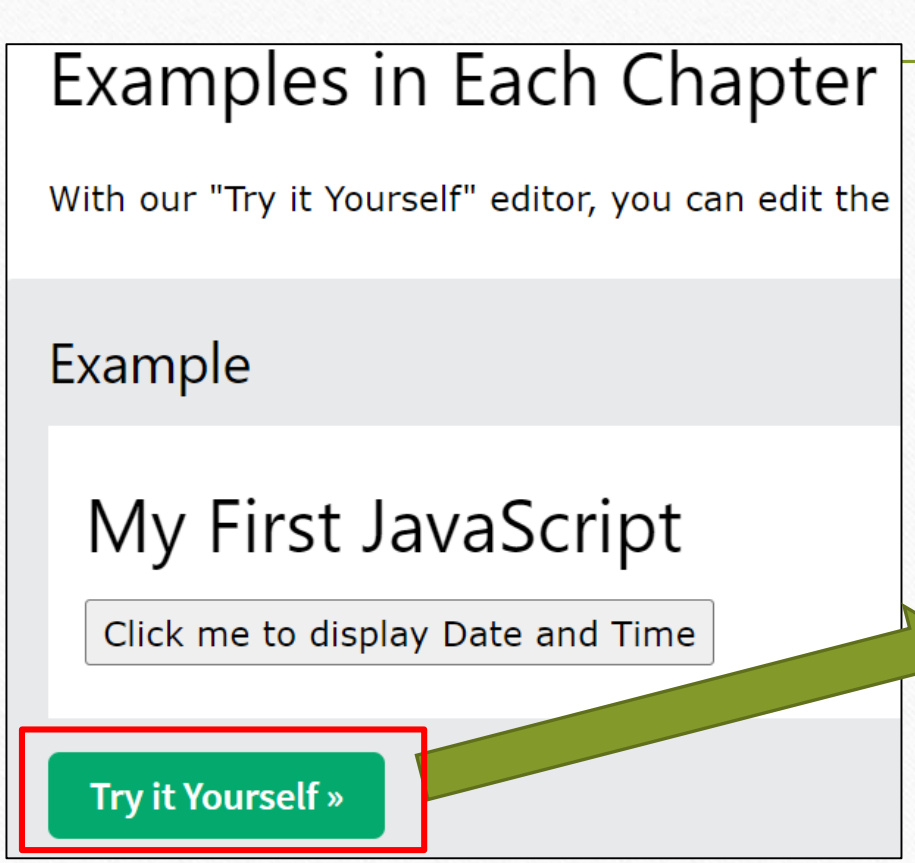
Click the button to change the layout of this paragraph

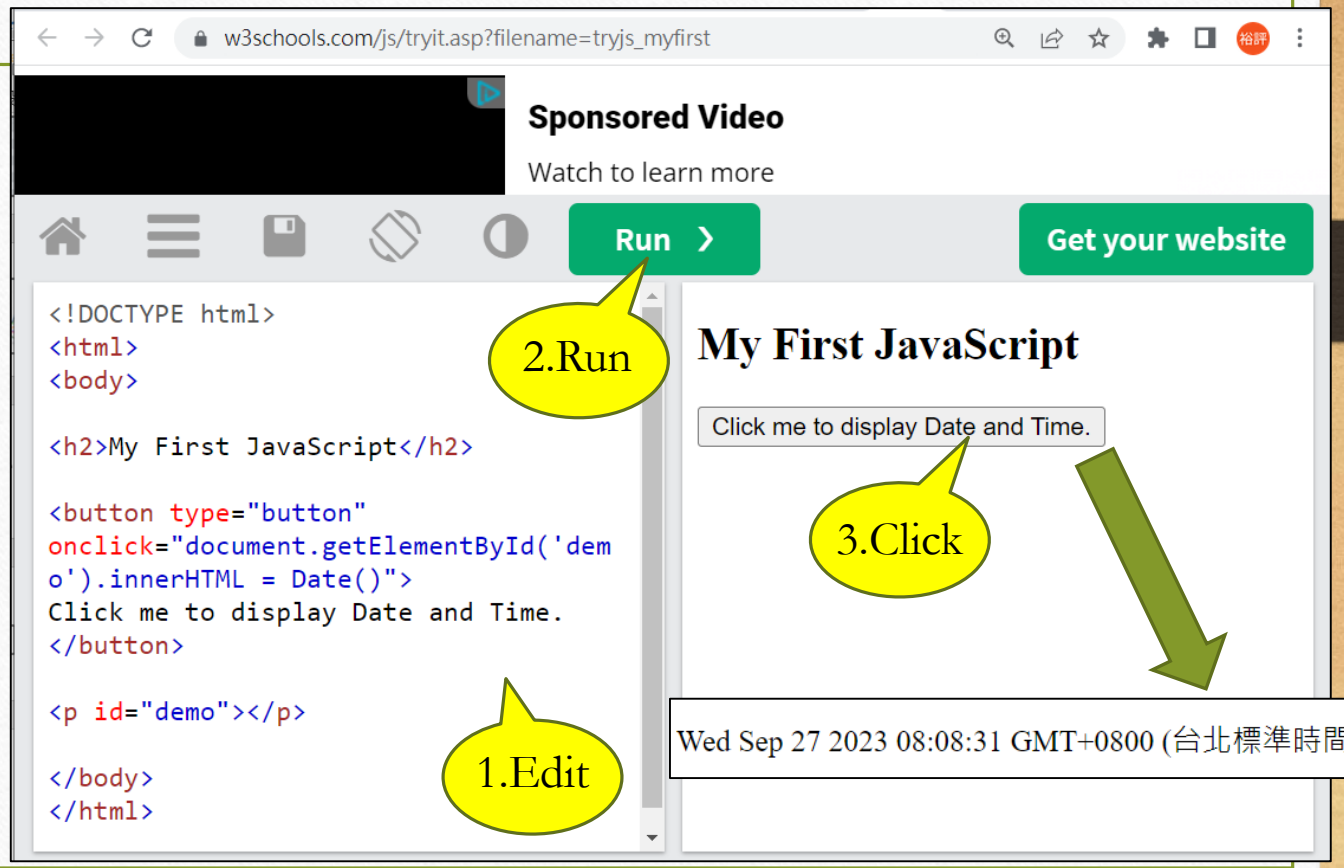
Click Me!

https://www.w3schools.com/



Try it Yourself





Why Study JavaScript?

Why Study JavaScript?

JavaScript is one of the **3 languages** all web developers **must** learn:

1. **HTML** to define the content of web pages
2. **CSS** to specify the layout of web pages
3. **JavaScript** to program the behavior of web pages

This tutorial covers every version of JavaScript:

- The Original JavaScript ES1 ES2 ES3 (1997-1999)
- The First Main Revision ES5 (2009)
- The Second Revision ES6 (2015)
- The Yearly Additions (2016, 2017, 2018)

Node.js

- Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.
- Node.js was written initially by Ryan Dahl in 2009
- https://en.wikipedia.org/wiki/Node.js

Node-RED

- **Node-RED** is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things.
- Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. Elements of applications can be saved or shared for re-use. The runtime is built on Node.js. The flows created in Node-RED are stored using JSON.
- In 2016, IBM contributed Node-RED as an open source OpenJS Foundation project.

https://en.wikipedia.org/wiki/Node-RED

JSON

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is a **text format** for storing and transporting data
- JSON is "self-describing" and easy to understand

JSON is an open standard file format for sharing data that uses human-readable text to store and transmit data. JSON files are stored with the .json extension. JSON requires less formatting and is a good alternative for XML. JSON is derived from JavaScript but is a language-independent data format. The generation and parsing of JSON is supported by many modern programming languages. *application/json* is the media type used for JSON.

JSON 是一種用於共享數據的開放標准文件格式，它使用人類可讀的文本來存儲和傳輸數據。 JSON 文件以 .json 擴展名存儲。 JSON 需要較少的格式，是 XML 的一個很好的替代品。 JSON 源自 JavaScript，但它是一種獨立於語言的數據格式。許多現代編程語言都支持 JSON 的生成和解析。 application/json 是用於 JSON 的媒體類型。

https://www.w3schools.com/js/js_json_intro.asp

JSON

- JSON object literals are surrounded by curly braces {}.
- JSON object literals contains key/value pairs.
- Keys and values are separated by a colon.
- Keys must be strings, and values must be a valid JSON data type:
- string
- number
- object
- array
- boolean
- null

Example 1:

```
{"name":"John", "age":30, "car":null}
```

Example 2:

```
{
"employee":{"name":"John", "age":30,
"city":"New York"}
}
```

Example 3:

```
{
"employees":["John", "Anna", "Peter"]
}
```

Example 4:

```
{"sale":true}
```

https://www.w3schools.com/js/js_json_datatypes.asp

JSON vs XML

Both JSON and XML can be used to receive data from a web server.
The following JSON and XML examples both define an employees object, with an array of 3 employees:

JSON Example

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

XML Example

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

https://www.w3schools.com/js/js_json_xml.asp

JSON.parse()

- Use the JavaScript function JSON.parse() to convert text into a JavaScript object.

```
const txt = '{"name":"John", "age":30, "city":"New York"}'
```

JSON.parse(txt)

```
{"name":"John", "age":30, "city":"New York"}
```

JSON.parse()

```
var text = '{ "name":"John", "age":"39", "city":"New York"}';
var obj = JSON.parse(text, function (key, value) {
  if (key == "city") {
    return value.toUpperCase();
  } else {
    return value;
  }
});
```

Syntax

```
JSON.parse(string, function)
```

Parameter Values

Parameter	Description
string	Required. A string written in JSON format
reviver function	Optional. A function used to transform the result. The function is called for each item. Any nested objects are transformed before the parent. • If the function returns a valid value, the item value is replaced with the transformed value • If the function returns undefined, the item is deleted

JSON.stringify()

- Use the JavaScript function JSON.stringify() to convert an object into a string.

```
const obj = {name: "John", age: 30, city: "New York"};
```

```
JSON.stringify(obj)
```

```
'{"name":"John", "age":30, "city":"New York"}'
```

JSON.stringify()

- /*replace the value of "city" to upper case:*/

```
var obj = { "name":"John", "age":"39", "city":"New York"};
var text = JSON.stringify(obj, function (key, value) {
  if (key == "city") {
    return value.toUpperCase();
  } else {
    return value;
  }
});
```

https://www.w3schools.com/jsref/jsref_obj_json.asp

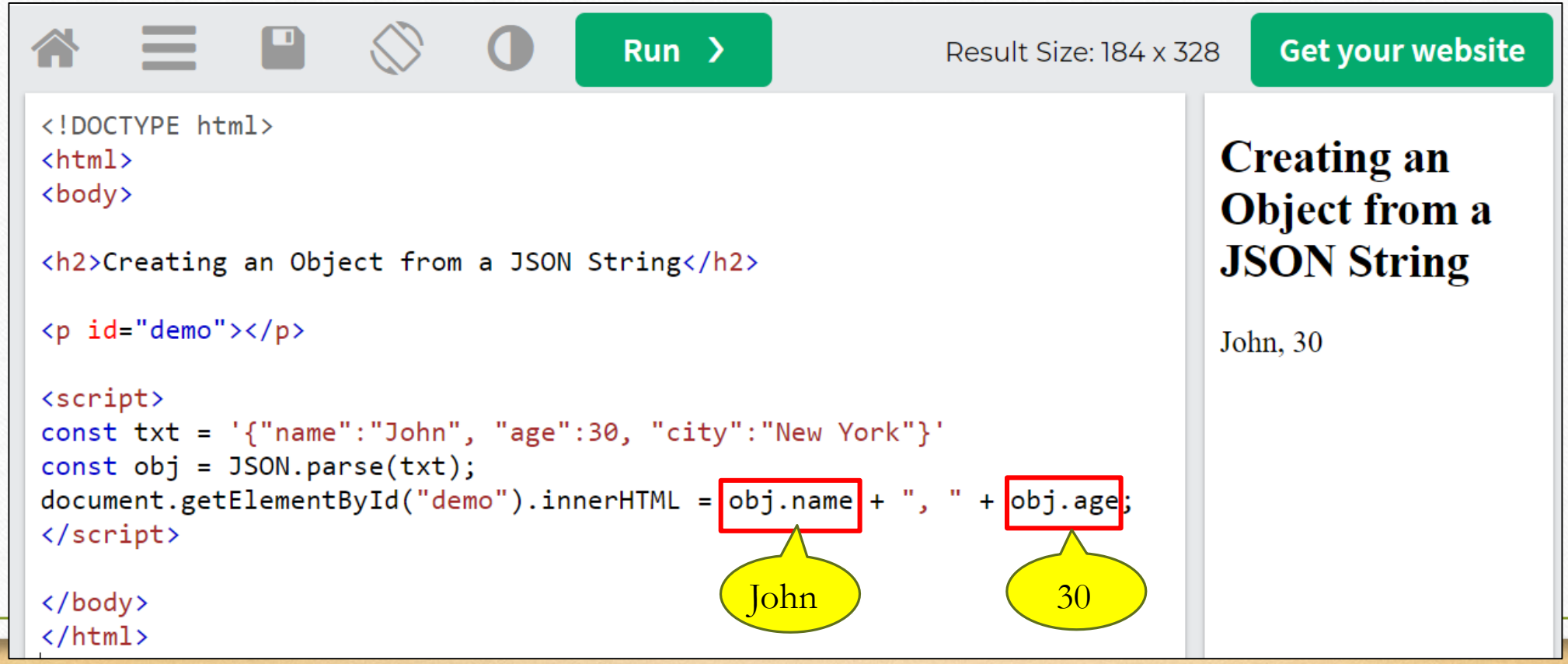
Accessing Object Values

- You can access object values by using dot (.) notation:

```
const myJSON = '{"name":"John", "age":30, "car":null}';
const myObj = JSON.parse(myJSON);
x = myObj.name;
y = myObj["name"];
```

https://www.w3schools.com/js/js_json_objects.asp

Try it Yourself



Exercise 3-2

- const txt = '{"name":"John", "age":30, "city":"New York","employees":["Mary","Tom","Jack"],"book":{"Electronics":"Neamen","Electromagnetics":"D.K. cheng"}}'
- const obj = JSON.parse(txt);

- Write the results:

(a) JSON.stringify(obj.employees)

(b) JSON.stringify(obj.book)

(c) JSON.stringify(obj.book.Electronics)

JavaScript Math Object

JavaScript Math Methods

Method	Description
abs(x)	Returns the absolute value of x
acos(x)	Returns the arccosine of x, in radians
acosh(x)	Returns the hyperbolic arccosine of x
asin(x)	Returns the arcsine of x, in radians
asinh(x)	Returns the hyperbolic arcsine of x
atan(x)	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
atan2(y, x)	Returns the arctangent of the quotient of its arguments
atanh(x)	Returns the hyperbolic arctangent of x
cbrt(x)	Returns the cubic root of x
ceil(x)	Returns x, rounded upwards to the nearest integer
cos(x)	Returns the cosine of x (x is in radians)
cosh(x)	Returns the hyperbolic cosine of x
exp(x)	Returns the value of E^x
floor(x)	Returns x, rounded downwards to the nearest integer

https://www.w3schools.com/js/js_math.asp

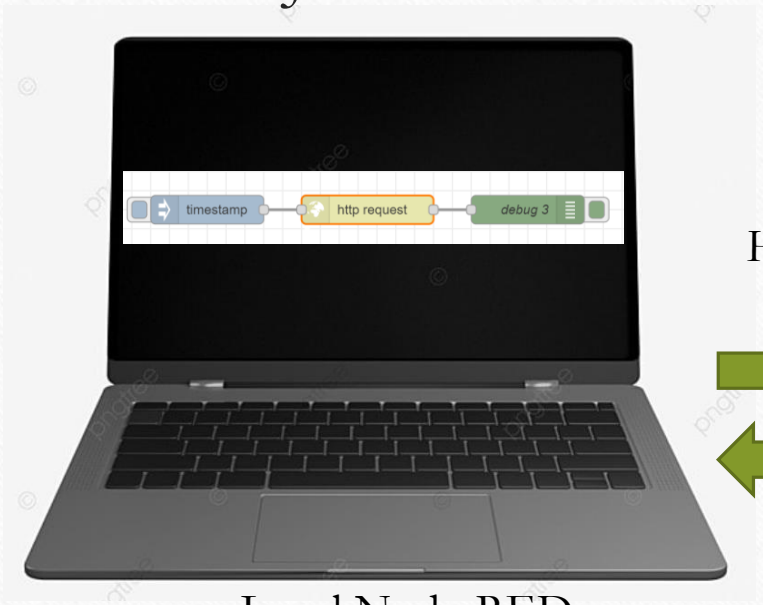
JavaScript Get Date Methods

Method	Description
getFullYear()	Get the **year** as a four digit number (yyyy)
getMonth()	Get the **month** as a number (0-11)
getDate()	Get the **day** as a number (1-31)
getHours()	Get the **hour** (0-23)
getMinutes()	Get the **minute** (0-59)
getSeconds()	Get the **second** (0-59)
getMilliseconds()	Get the **millisecond** (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

https://www.w3schools.com/js/js_date_methods.asp

Exercise 3-4

- GET JSON format data in the server.

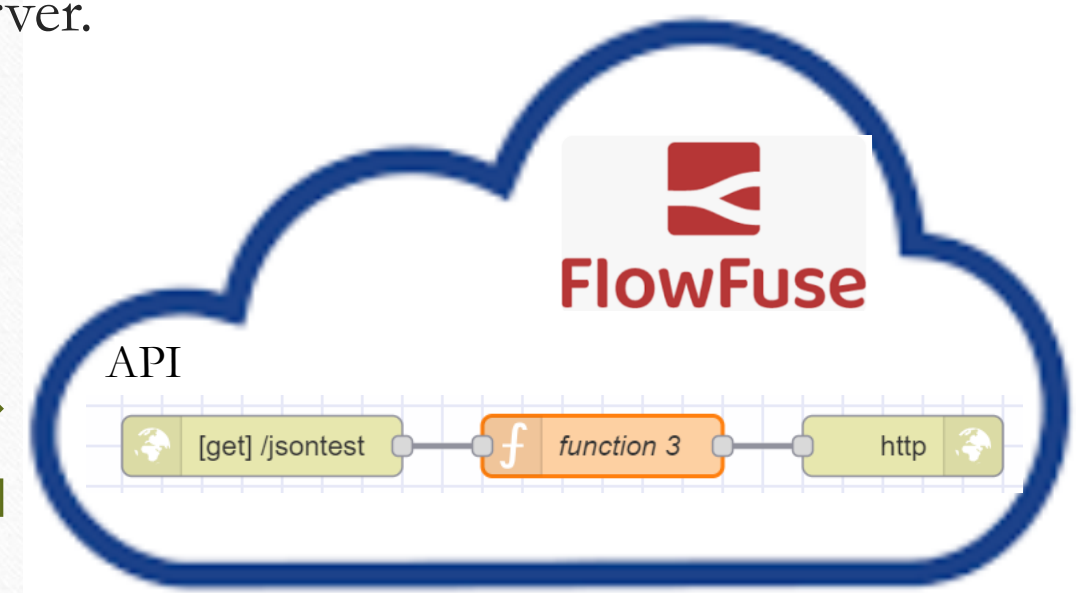


Local Node-RED

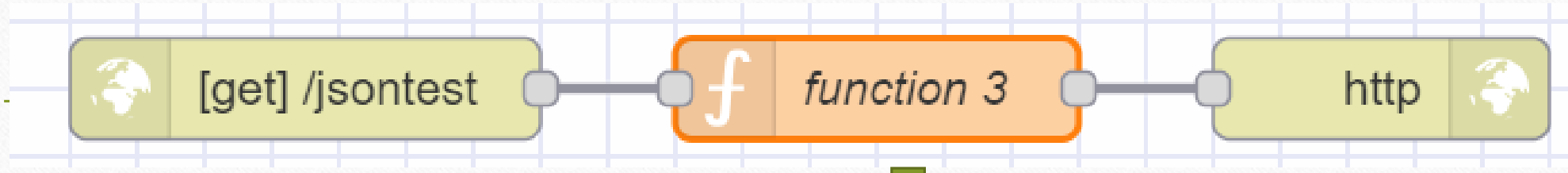
HTTP GET Request

Response

API



FlowFuse



Edit http in node

Delete | Cancel | Done

Properties

Method: GET

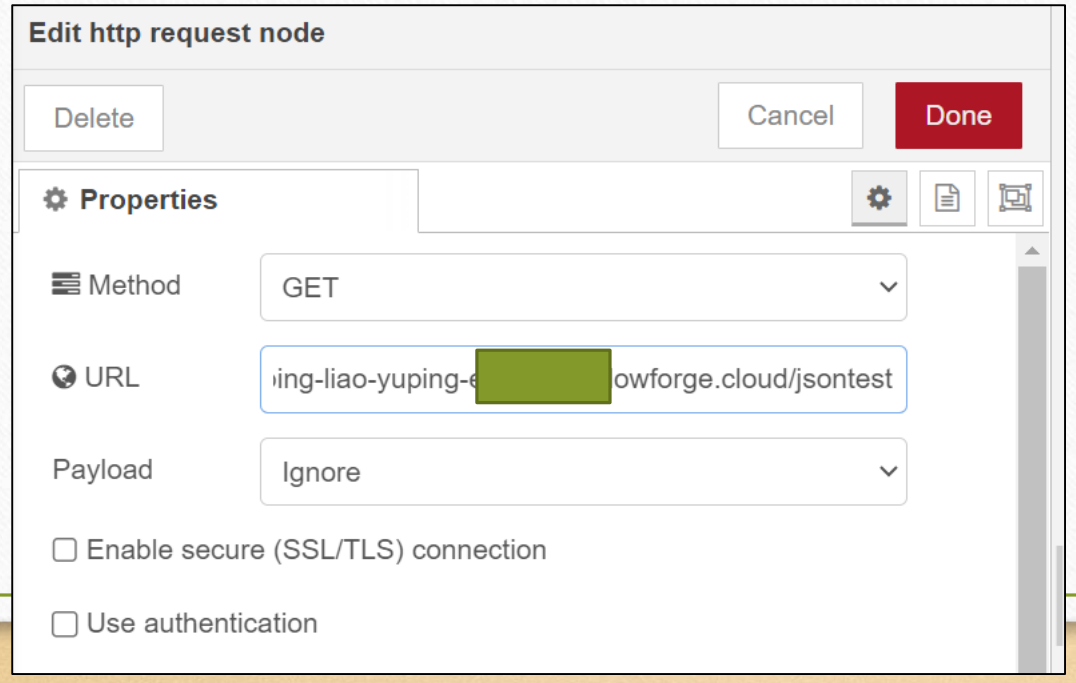
URL: /jsontest

Name: Name

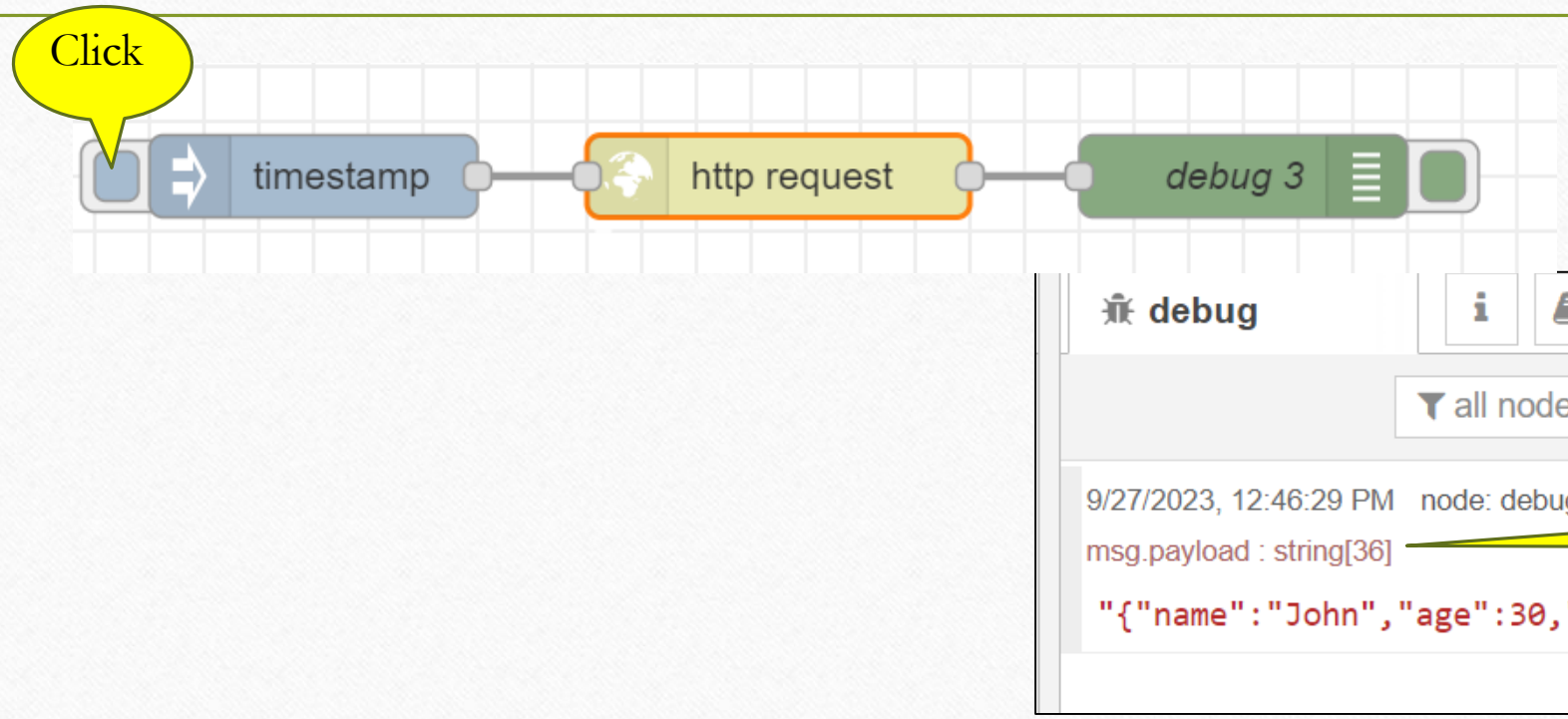
Setup | On Start | On Message | On Stop

```
1  msg.payload = {"name":"John", "age":30, "car":"BMW"};
2  return msg;
```

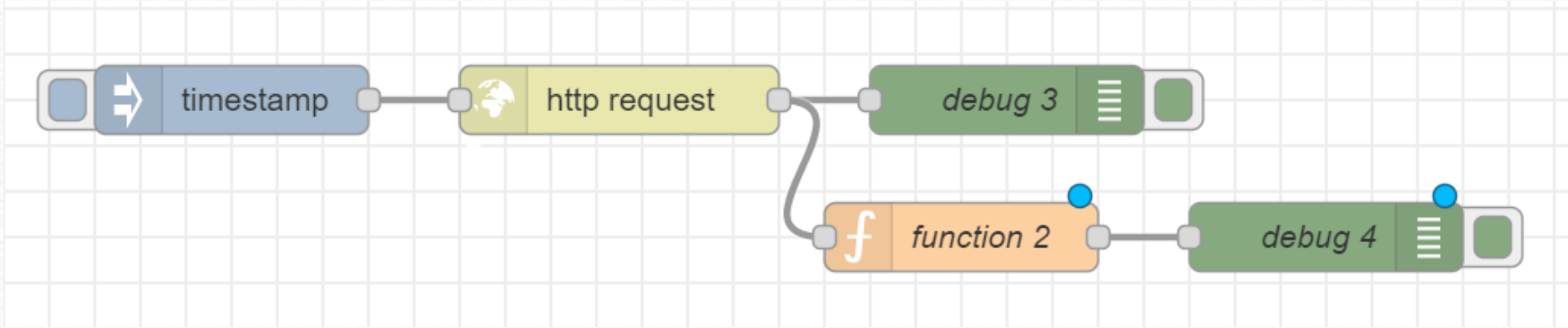
http://localhost:1880/



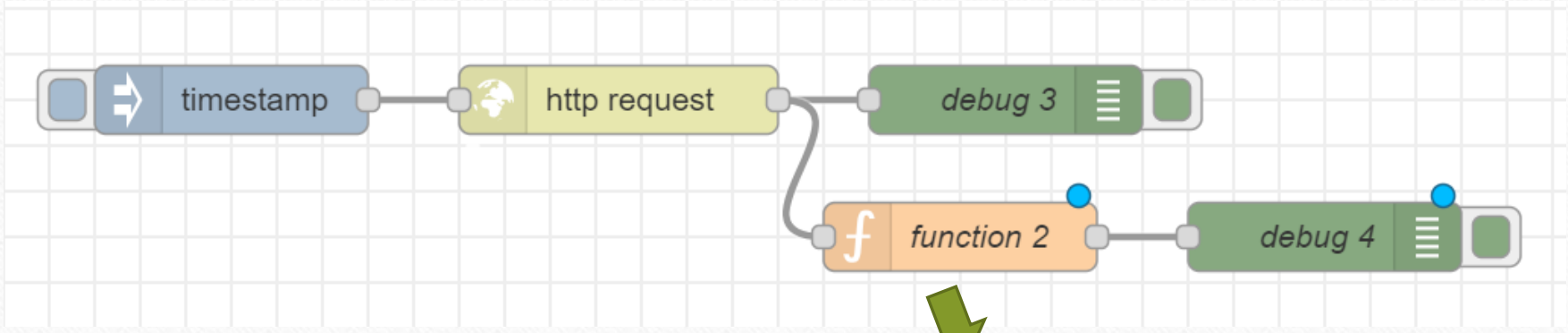
Trigger


Click
timestamp
http request
debug 3
debug
all nodes
all
9/27/2023, 12:46:29 PM node: debug 3
msg.payload : string[36]
String
"{"name":"John","age":30,"car":"BMW"}"

Add “function” & “debug” nodes



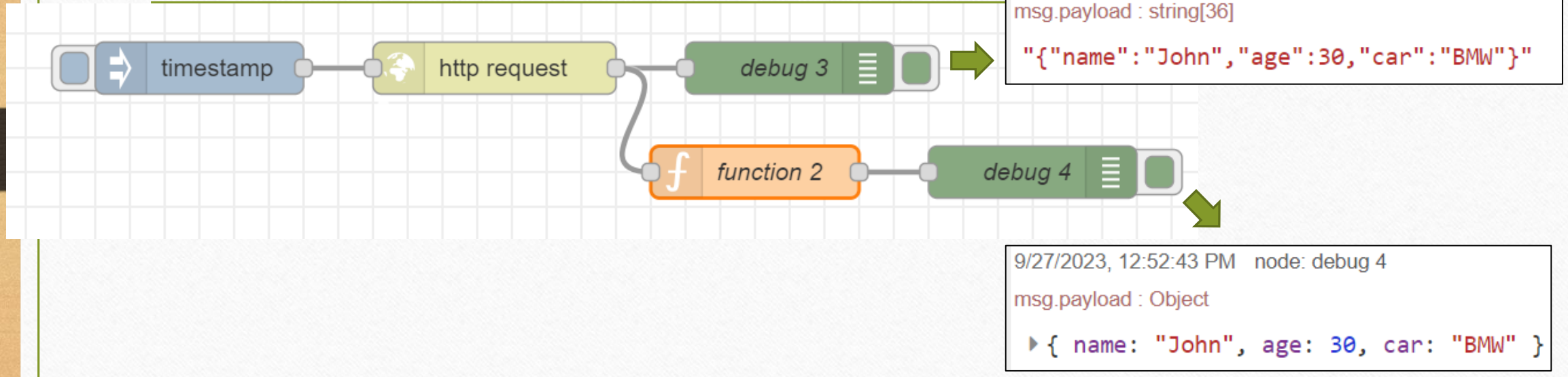
Edit function 2



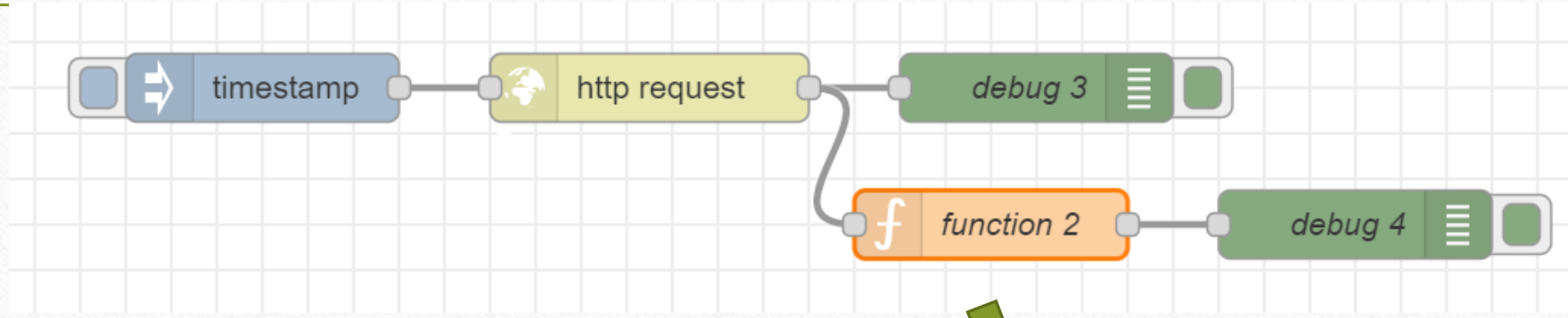
Setup | On Start | **On Message** | On Stop

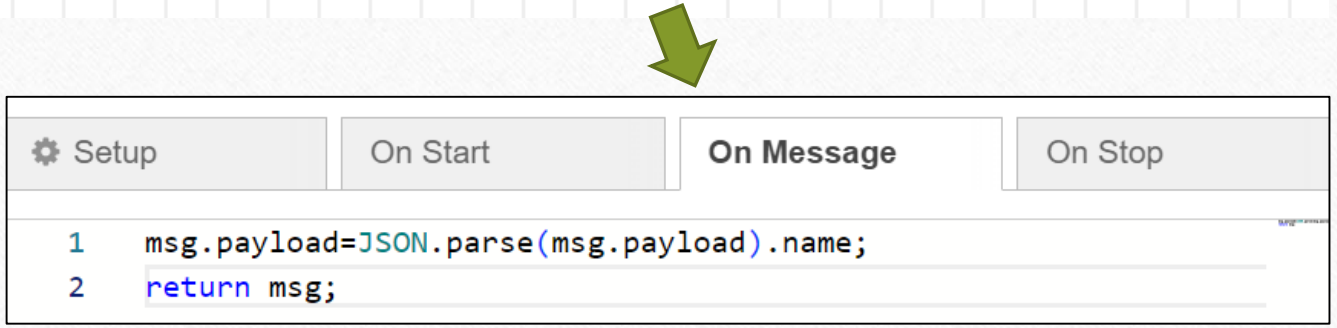
```
1  msg.payload=JSON.parse(msg.payload);
2  return msg;
```

Deploy & Trigger



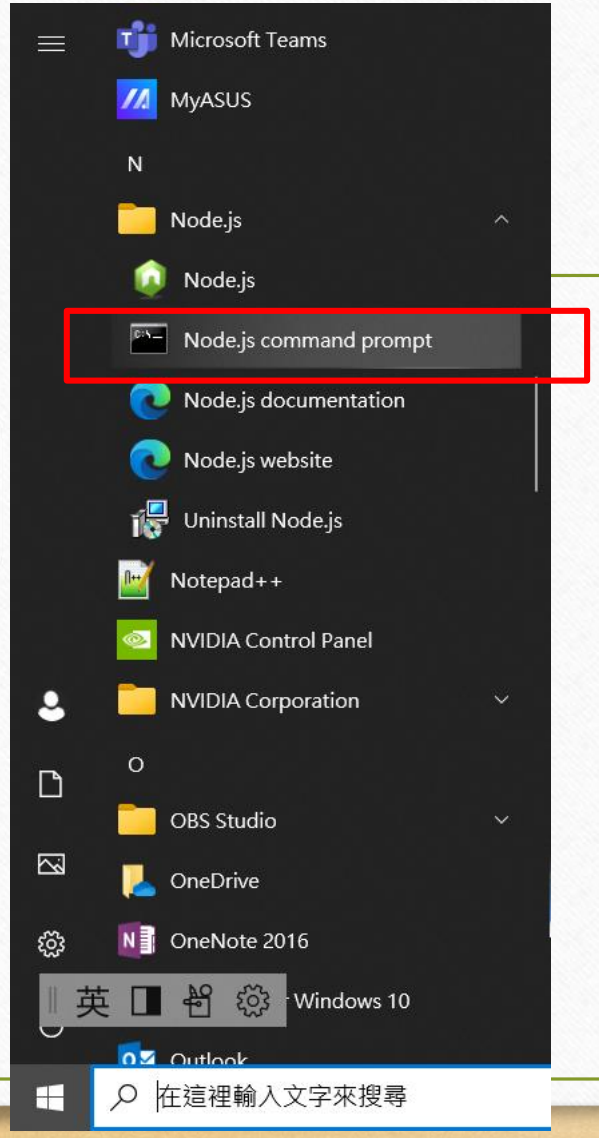
Edit function 2





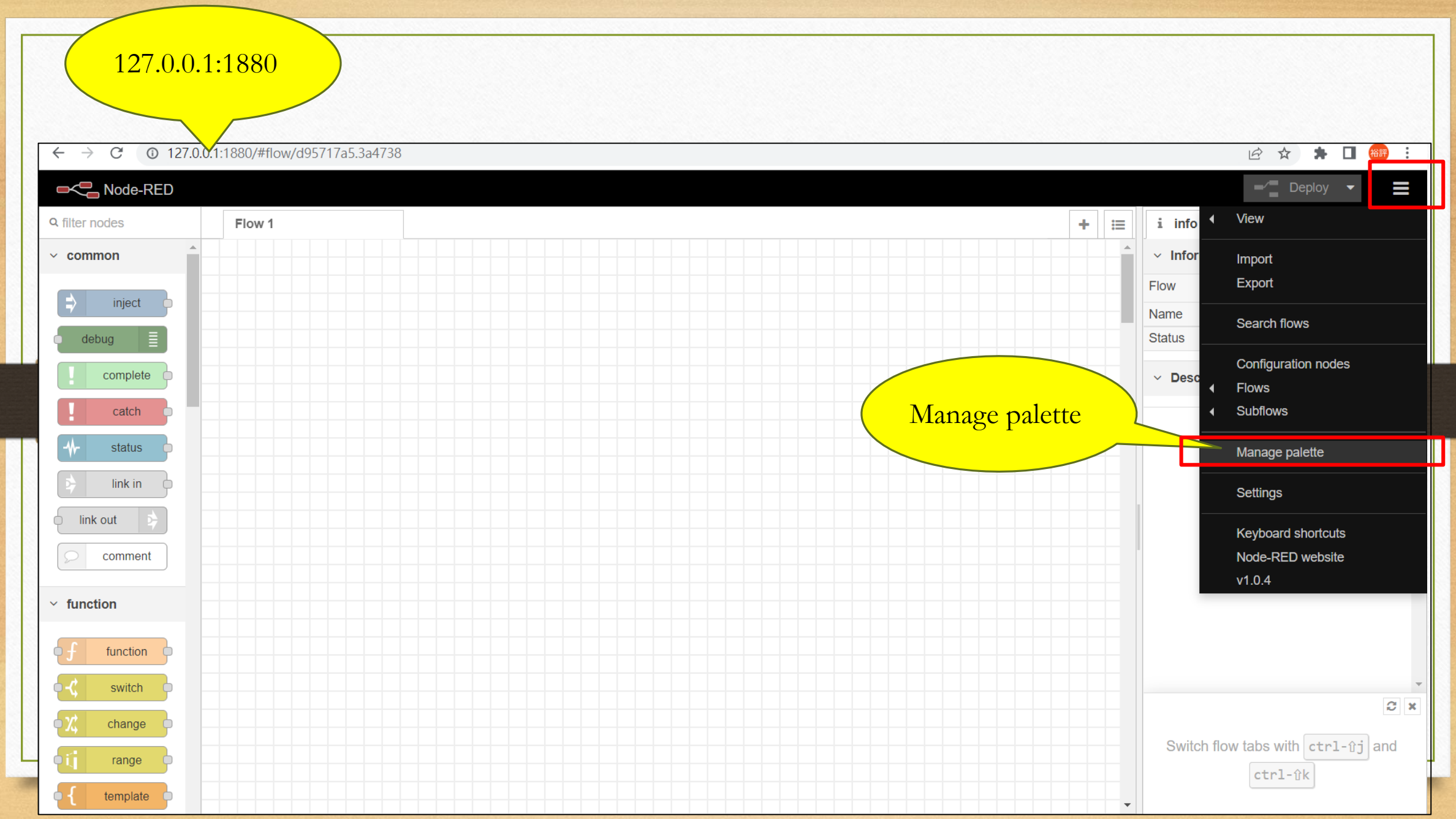
Exercise 3-5

- Install “node-red-dashboard”

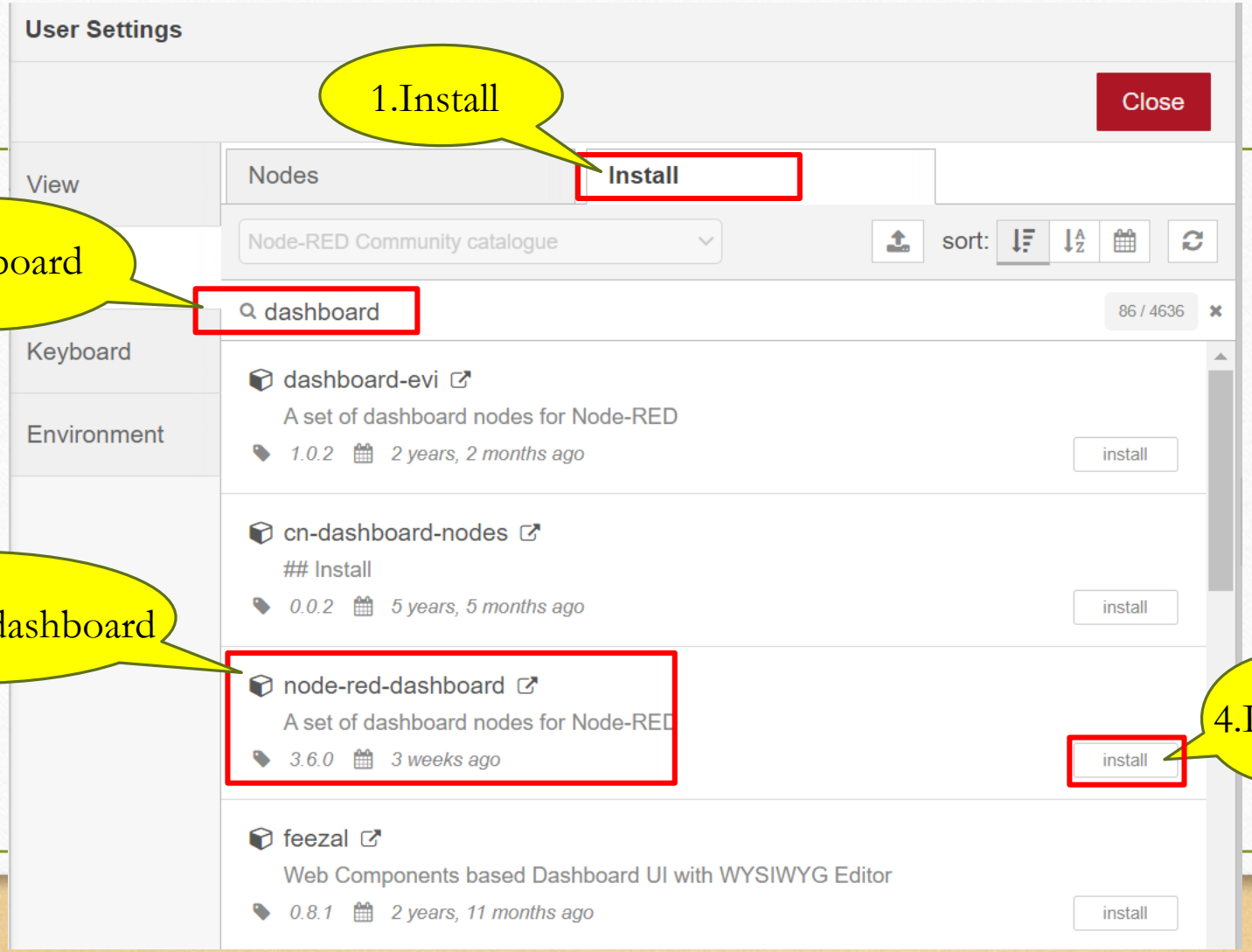


node-red

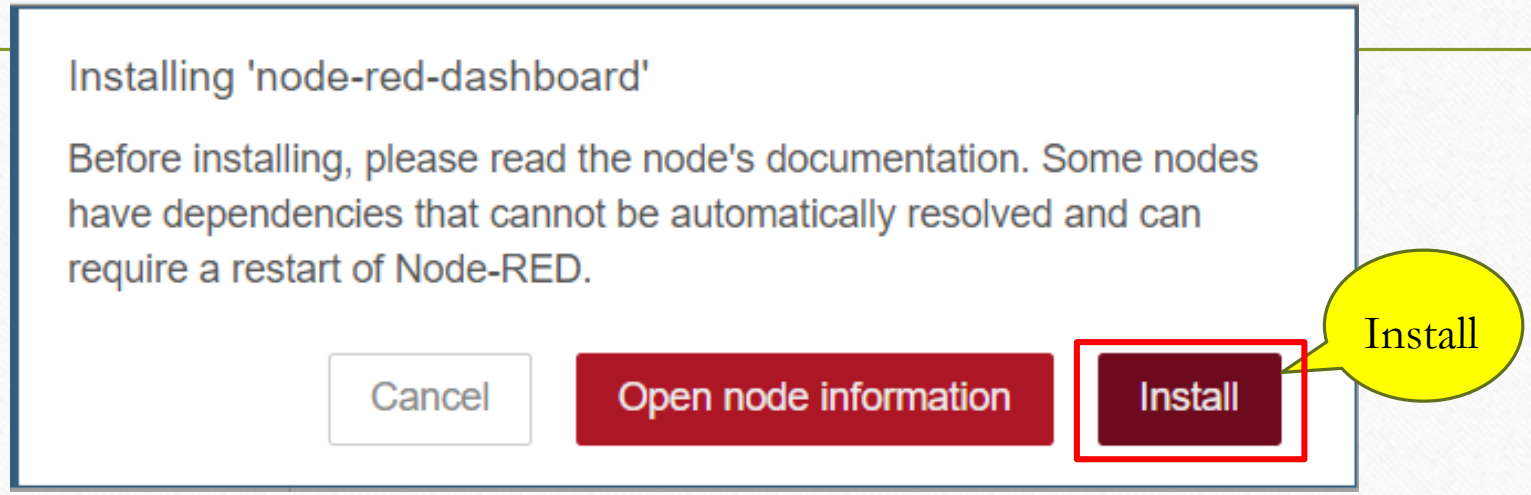
```
node-red
Your environment has been set up for using Node.js 10.16.3 (x64) and npm.

C:\Users\88697>node-red
27 Sep 23:11:15 - [info]

Welcome to Node-RED
===================

27 Sep 23:11:15 - [info] Node-RED version: v1.0.4
27 Sep 23:11:15 - [info] Node.js  version: v10.16.3
27 Sep 23:11:15 - [info] Windows_NT 10.0.19044 x64 LE
27 Sep 23:11:19 - [info] Loading palette nodes
27 Sep 23:11:20 - [error] Dashboard version 3.2.0 requires Nodejs 12 or more recent
27 Sep 23:11:21 - [info] Settings file  : \Users\88697\.node-red\settings.js
27 Sep 23:11:21 - [info] Context store  : 'default' [module=memory]
27 Sep 23:11:21 - [info] User directory : \Users\88697\.node-red
27 Sep 23:11:21 - [warn] Projects disabled : editorTheme.projects.enabled=false
27 Sep 23:11:21 - [info] Flows file     : \Users\88697\.node-red\flows_LAPTOP-TKIKIDEP.json
27 Sep 23:11:21 - [info] Server now running at http://127.0.0.1:1880/
27 Sep 23:11:21 - [warn]
```


127.0.0.1:1880
127.0.0.1:1880/#flow/d95717a5.3a4738
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
info
Flow
Name
Status
View
Import
Export
Search flows
Configuration nodes
Flows
Subflows
Manage palette
Settings
Keyboard shortcuts
Node-RED website
v1.0.4
Manage palette
Switch flow tabs with ctrl-⇧j and ctrl-⇧k

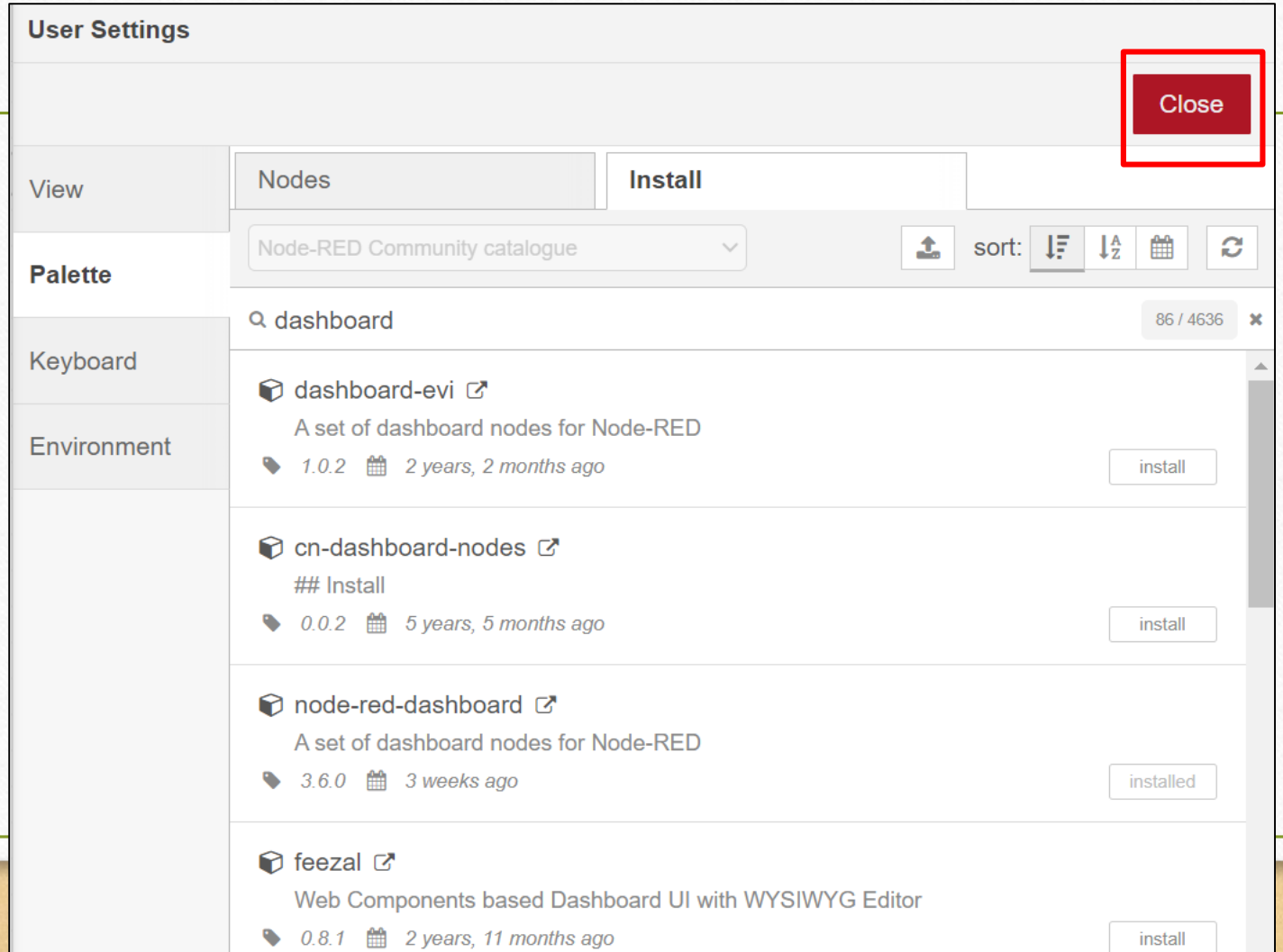
Install


User Settings
1.Install
Close
View
Nodes
Install
2.dashboard
Node-RED Community catalogue
sort:
dashboard
86 / 4636
Keyboard
Environment
dashboard-evi
A set of dashboard nodes for Node-RED
1.0.2
2 years, 2 months ago
install
cn-dashboard-nodes
Install
0.0.2
5 years, 5 months ago
install
3.node-red-dashboard
node-red-dashboard
A set of dashboard nodes for Node-RED
3.6.0
3 weeks ago
install
4.Install
feezal
Web Components based Dashboard UI with WYSIWYG Editor
0.8.1
2 years, 11 months ago
install

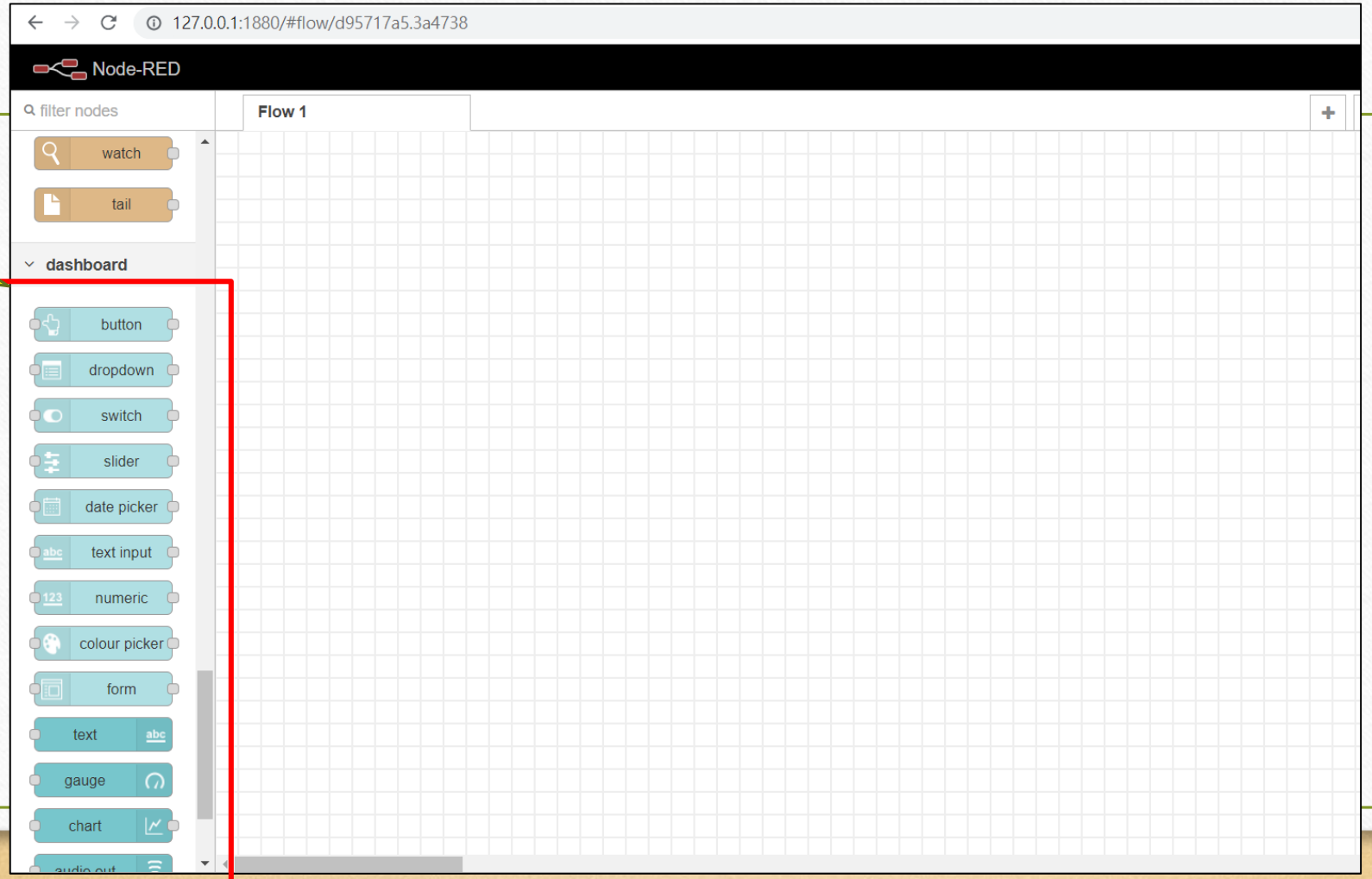
Install


Installing 'node-red-dashboard'
Before installing, please read the node's documentation. Some nodes have dependencies that cannot be automatically resolved and can require a restart of Node-RED.
Cancel
Open node information
Install
Install

Close


User Settings
Close
View
Palette
Keyboard
Environment
Nodes
Install
Node-RED Community catalogue
sort:
dashboard
86 / 4636
dashboard-evi
A set of dashboard nodes for Node-RED
1.0.2
2 years, 2 months ago
install
cn-dashboard-nodes
Install
0.0.2
5 years, 5 months ago
install
node-red-dashboard
A set of dashboard nodes for Node-RED
3.6.0
3 weeks ago
installed
feezal
Web Components based Dashboard UI with WYSIWYG Editor
0.8.1
2 years, 11 months ago
install

dashboard


127.0.0.1:1880/#flow/d95717a5.3a4738
Node-RED
filter nodes
Flow 1
watch
tail
dashboard
button
dropdown
switch
slider
date picker
text input
numeric
colour picker
form
text
gauge
chart
dashboard

Homework 3-3

- http://noderedguide.com/tutorial-node-red-dashboards-multiple-lines-on-a-chart/