物聯網實務

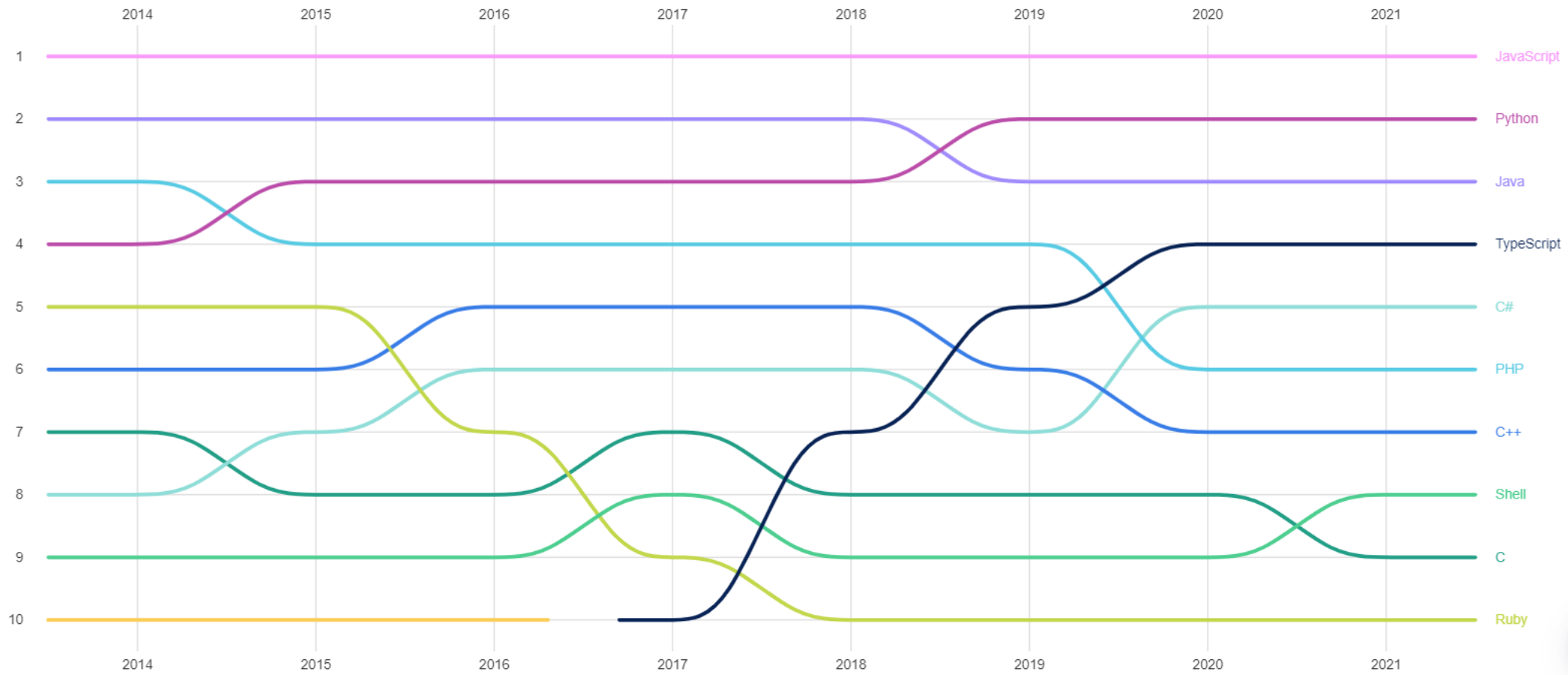
9_28

廖裕評

The Most Popular Programming Languages for 2022

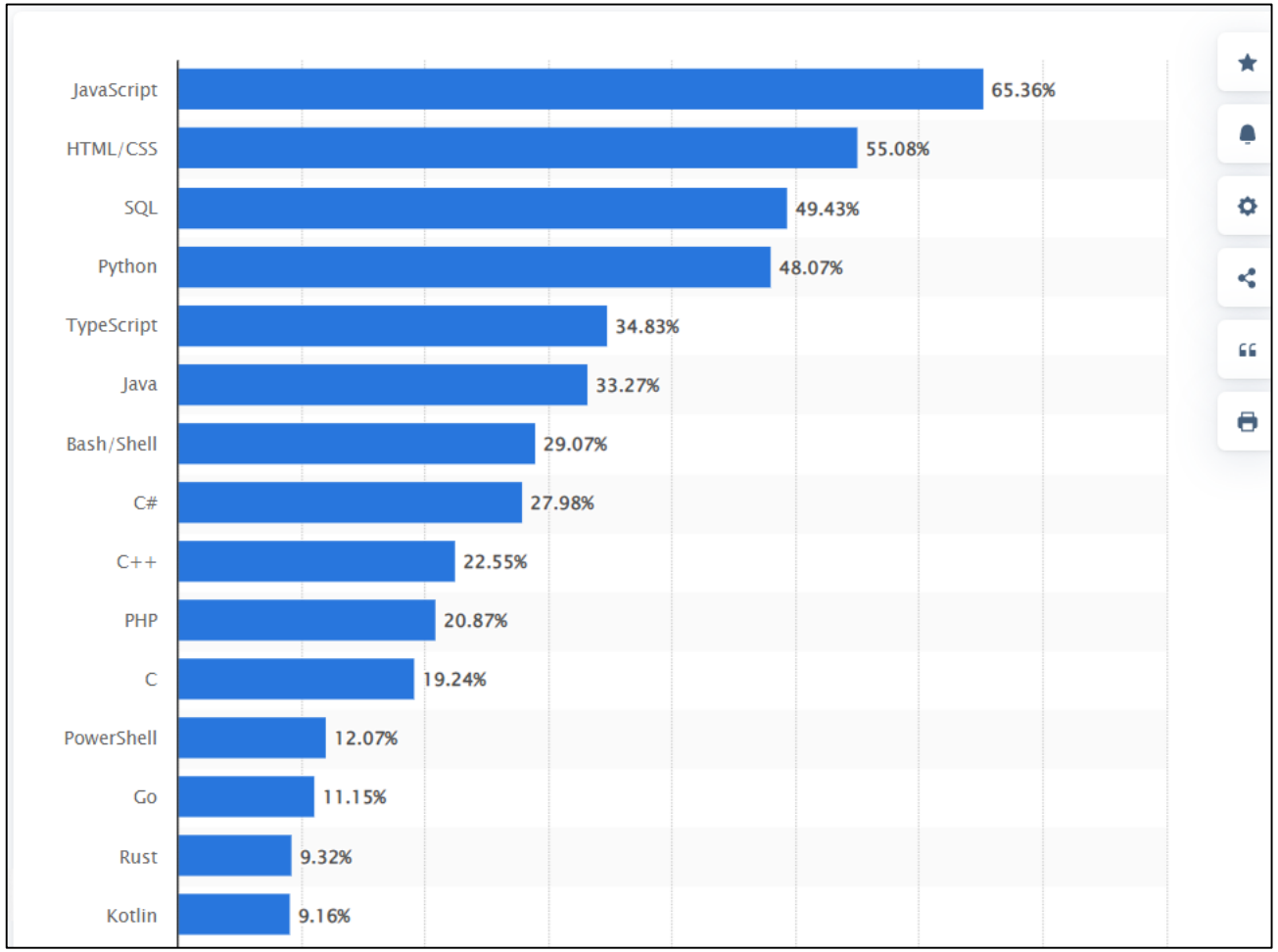
https://www.magicbell.com/blog/most-popular-programming-languages

Top languages over the years (**GitHub**)



https://octoverse.github.com/#top-languages-over-the-years

Most used programming languages among developers worldwide as of 2022 (Statista)



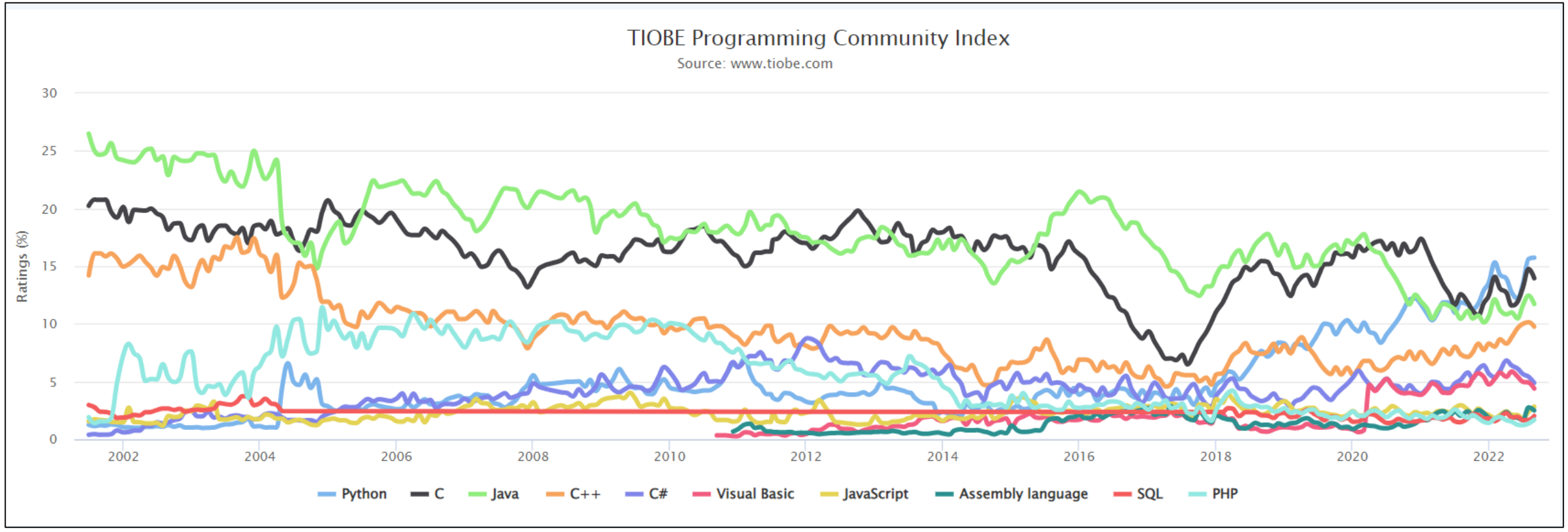
https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

TIOBE Index for September 2022

Sep 2022	Sep 2021	Change	Programming Language	Ratings	Change
1	2	▲	Python	15.74%	+4.07%
2	1	▼	C	13.96%	+2.13%
3	3		Java	11.72%	+0.60%
4	4		C++	9.76%	+2.63%
5	5		C#	4.88%	-0.89%
6	6		Visual Basic	4.39%	-0.22%
7	7		JavaScript	2.82%	+0.27%
8	8		Assembly language	2.49%	+0.07%
9	10	▲	SQL	2.01%	+0.21%

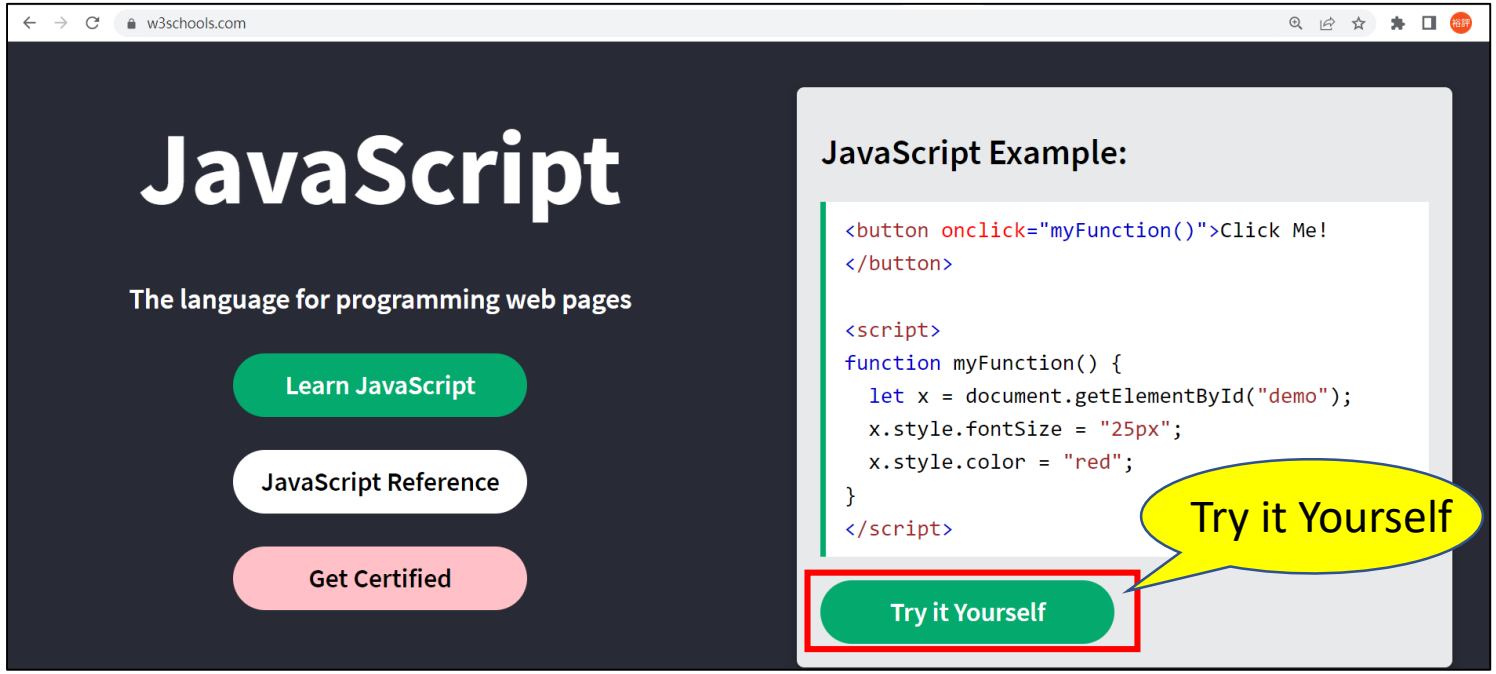
https://www.tiobe.com/tiobe-index/

TIOBE Index for September 2022

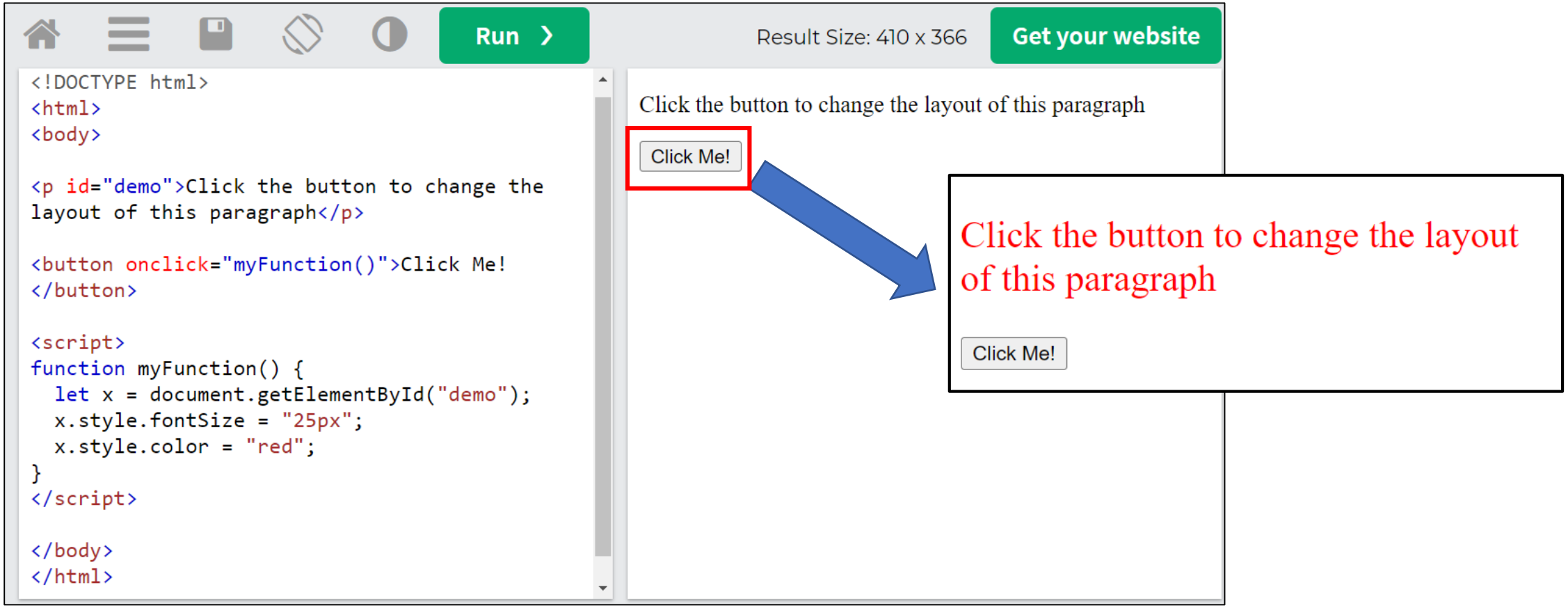


An Introduction to JavaScript

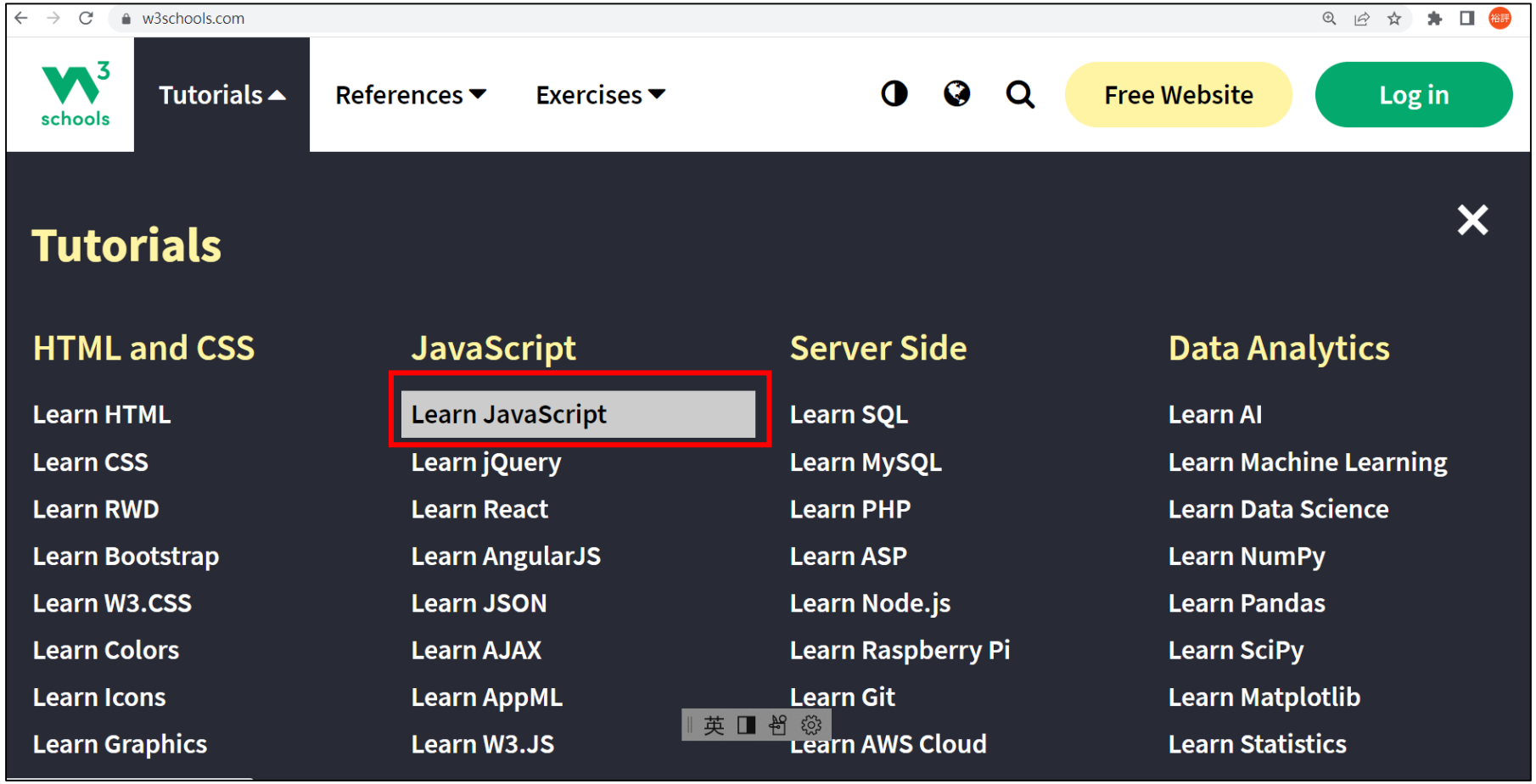
- JavaScript is the programming language of HTML and the Web.
- JavaScript is easy to learn.
- With "Try it Yourself" editor on https://www.w3schools.com/, you can edit the source code and view the result.



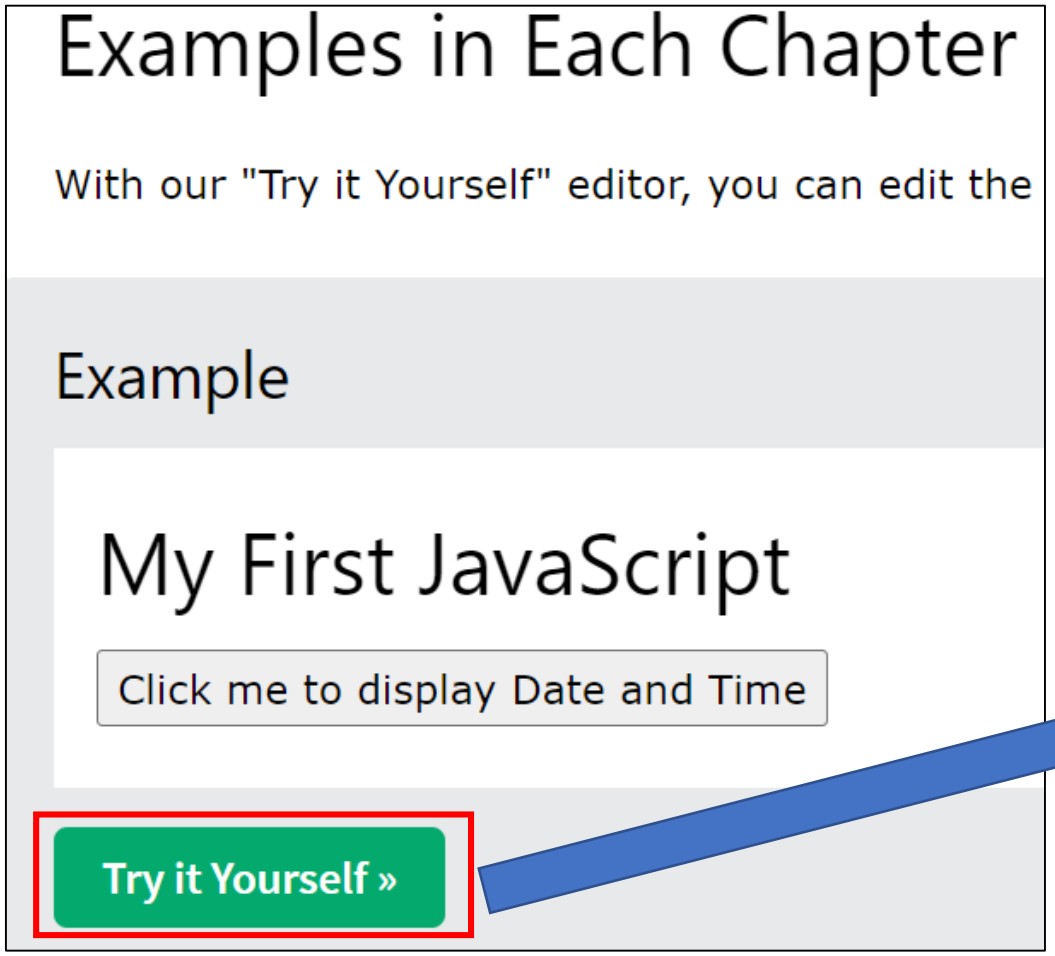
Click the button

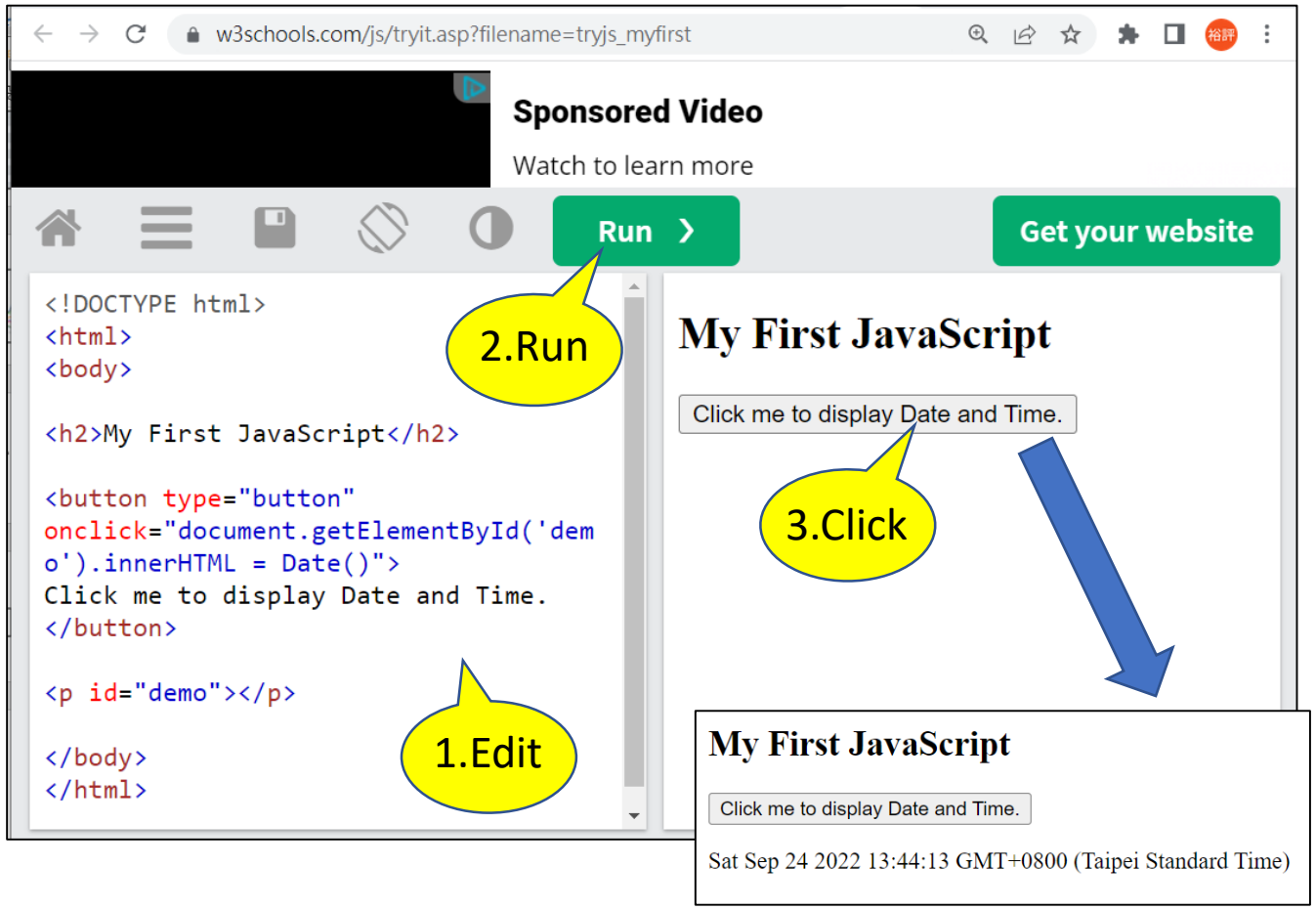


https://www.w3schools.com/



Try it Yourself





Why Study JavaScript?

Why Study JavaScript?

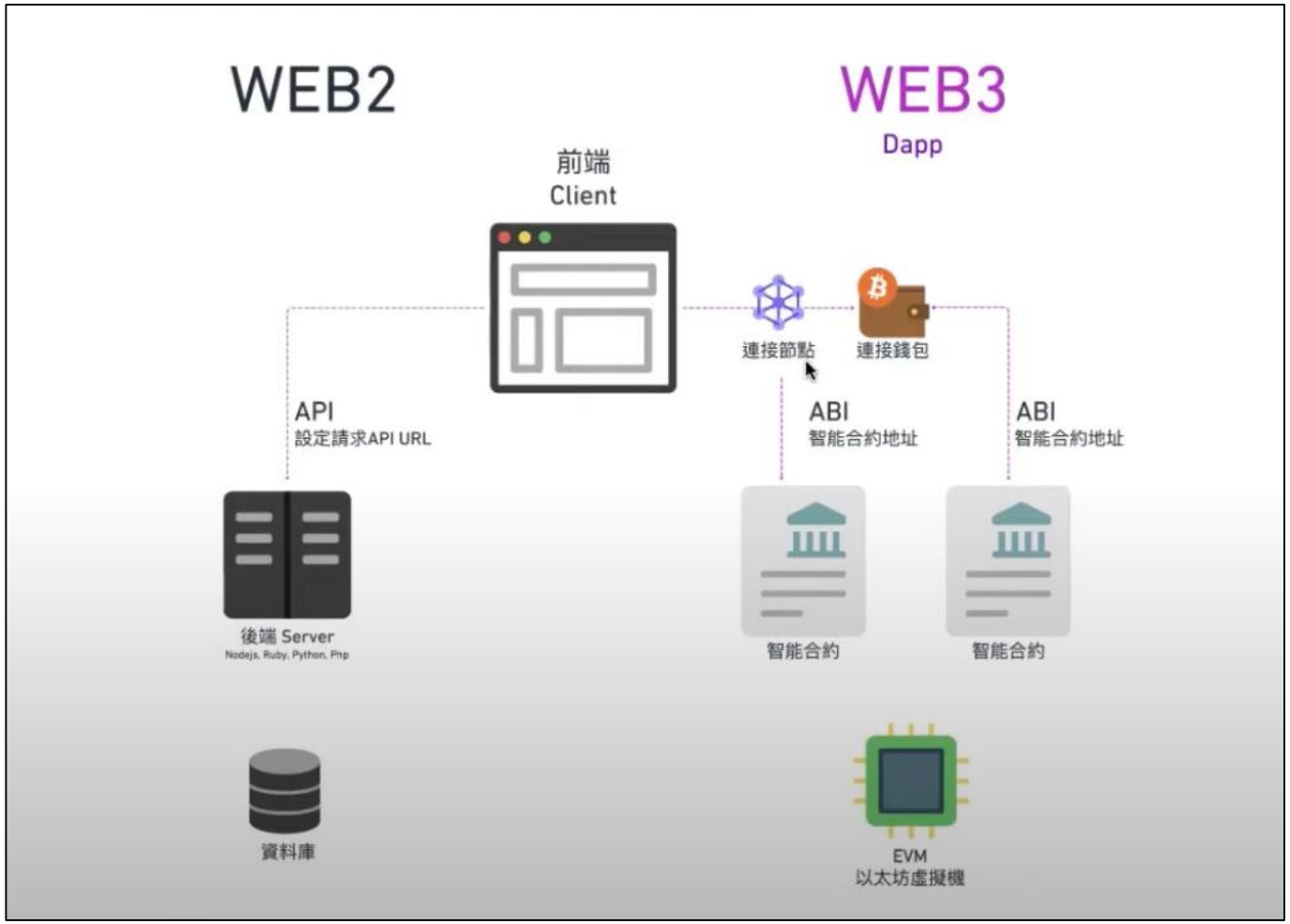
JavaScript is one of the **3 languages** all web developers **must** learn:

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)

Web 2 vs Web 3



https://www.youtube.com/watch?v=diwodzURGFo

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。

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:

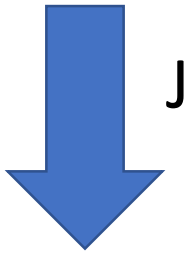
```
{"sale":true}
```

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

JSON.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

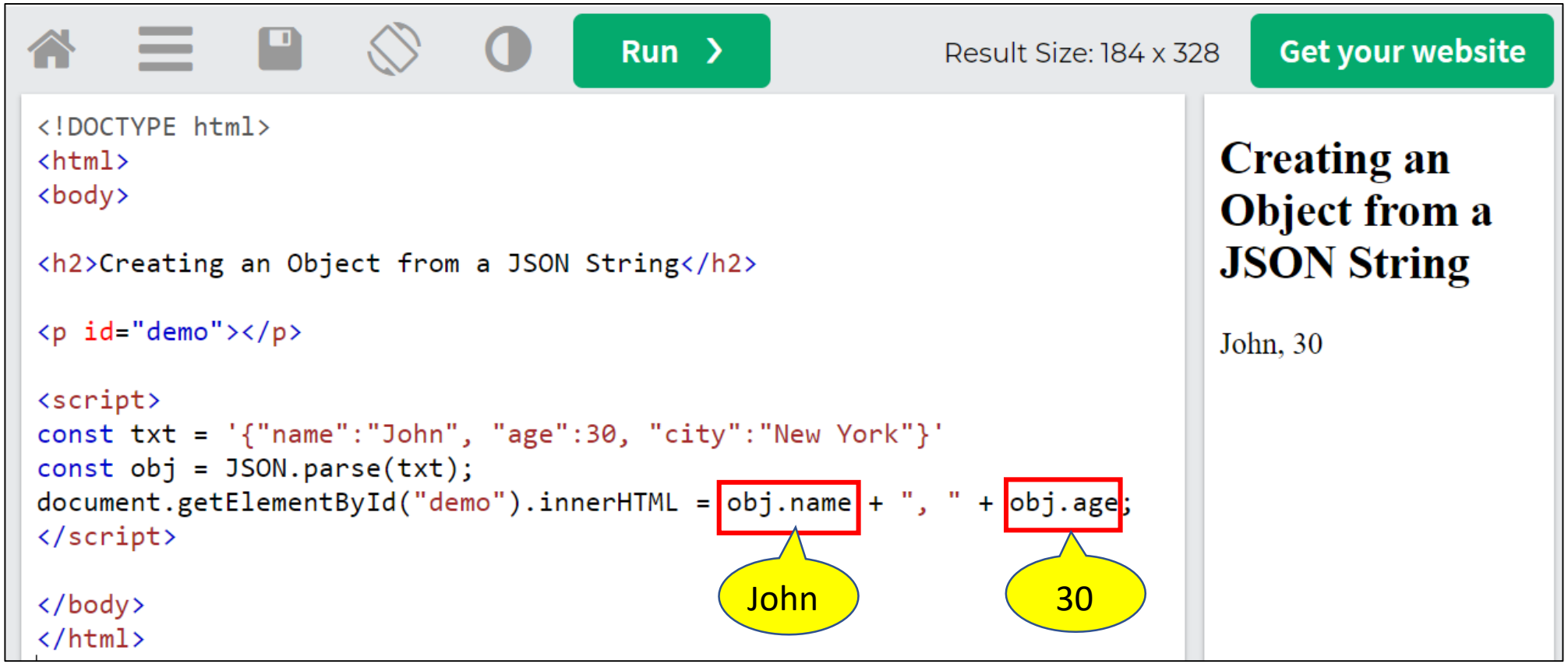
- You can access object values by using dot (.) notation:

```
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

- https://www.w3schools.com/js/tryit.asp?filename=tryjson_parse



Exercise 3-1

Run ›

Result Size: 454 x 448

Get your own websit

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

<h2>Creating an Object from a JSON String</h2>

<p id="demo"></p>

<script>
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);
document.getElementById("demo").innerHTML = obj.name + ", " +
obj.age+ obj.employees+ obj.book ;
</script>

</body>
</html>
```

Creating an Object from a JSON String

??????

(A) John, 30,Mary,Tom,Jack{"Electronics":"Neamen","Electromagnetics":"D.K. cheng"}
(B) John, 30 Mary,Tom,Jack{"Electronics":"Neamen","Electromagnetics":"D.K. cheng"}
(C) John, 30Mary,Tom,Jack{"Electronics":"Neamen","Electromagnetics":"D.K. cheng"}
(D) John, 30Mary,Tom,Jack[object Object]

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);

- What is the result?

(a) JSON.stringify(obj.employees)

(b) JSON.stringify(obj.book)

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

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

JavaScript Math Object

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

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

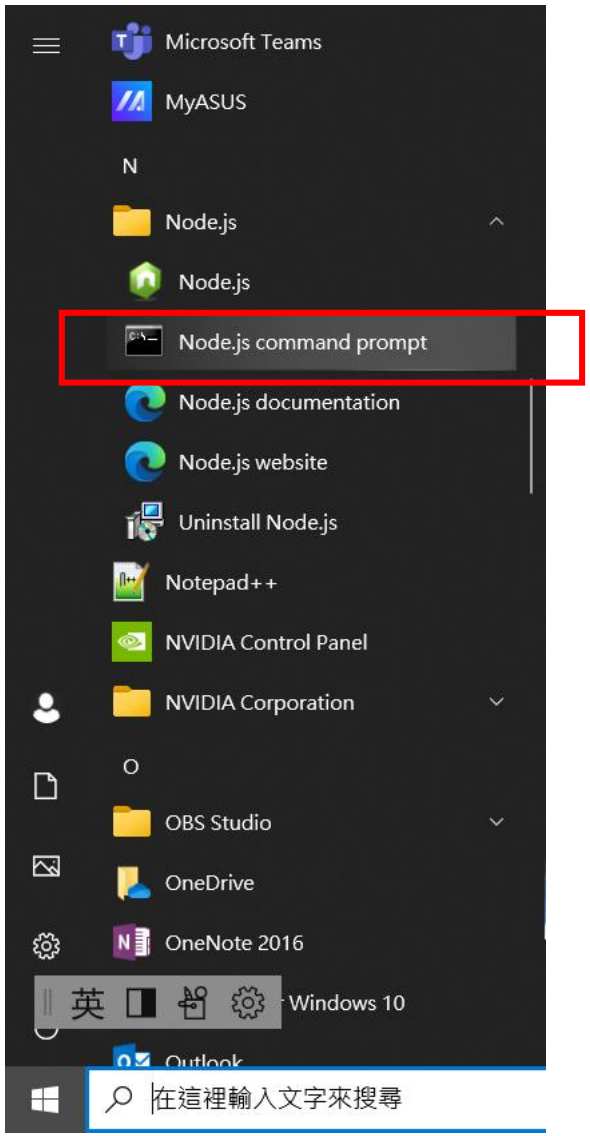
JavaScript Get Date Methods

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

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.

Exercise 3-3

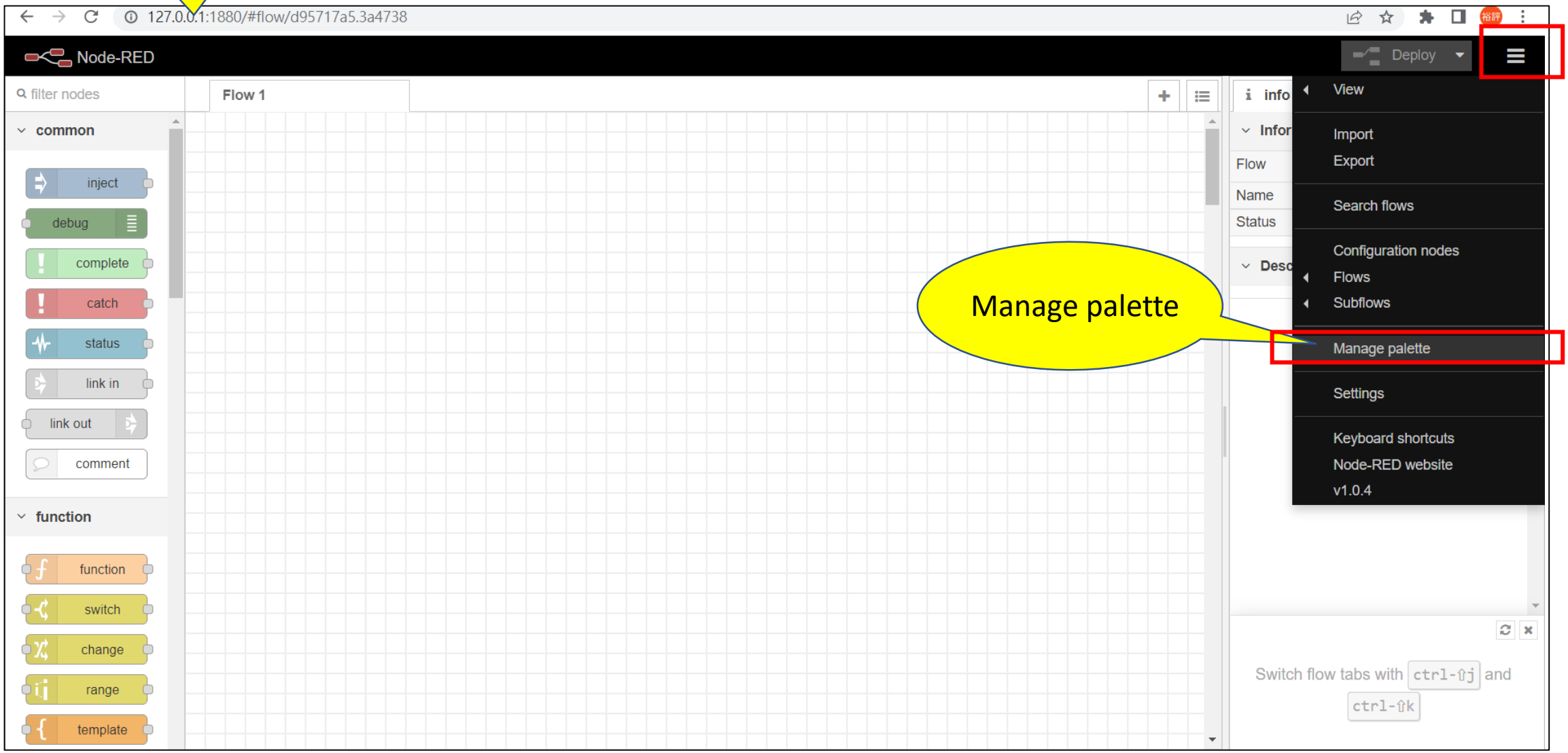
- Install “node-red-dashboard”



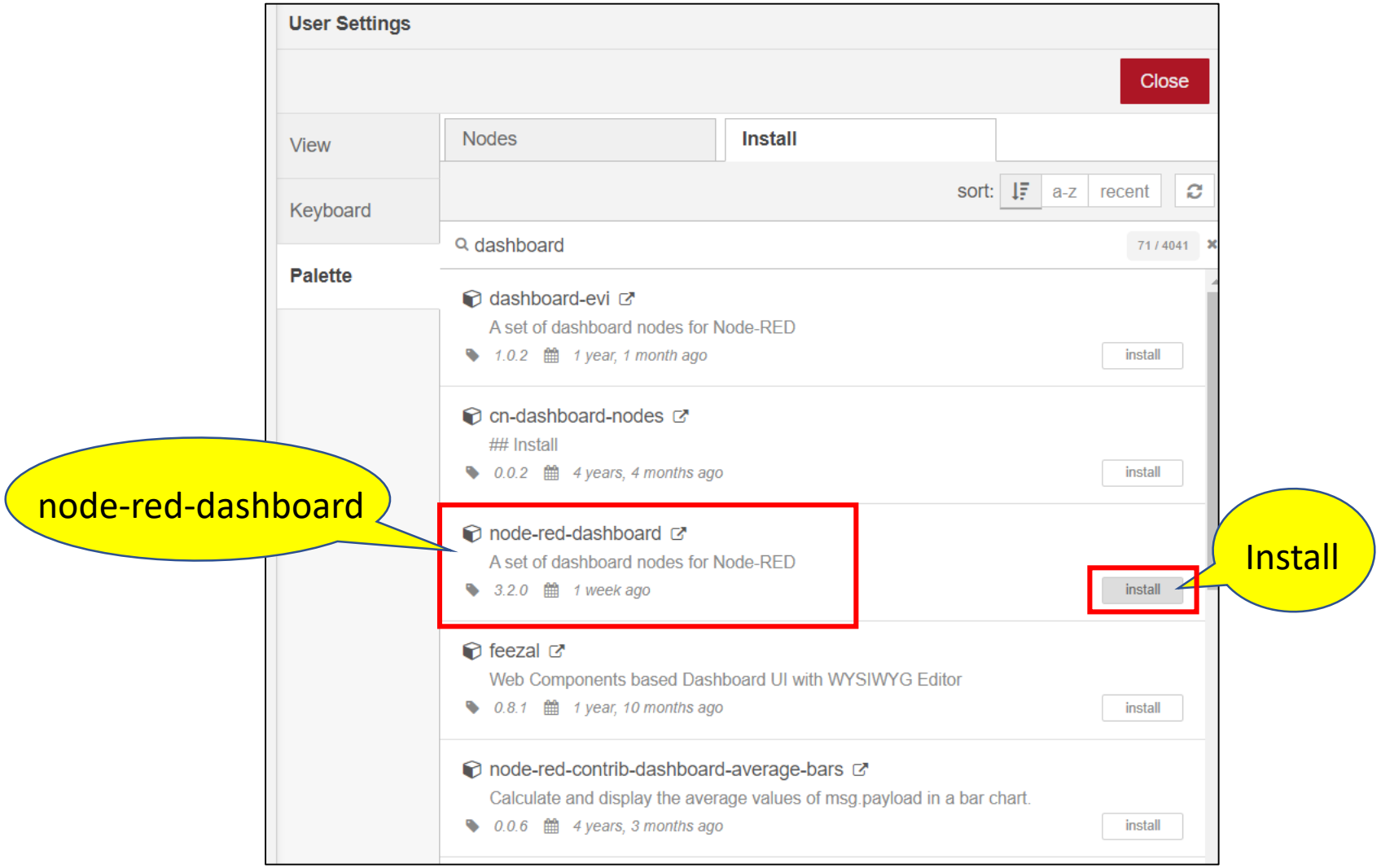
node-red

```
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]
```

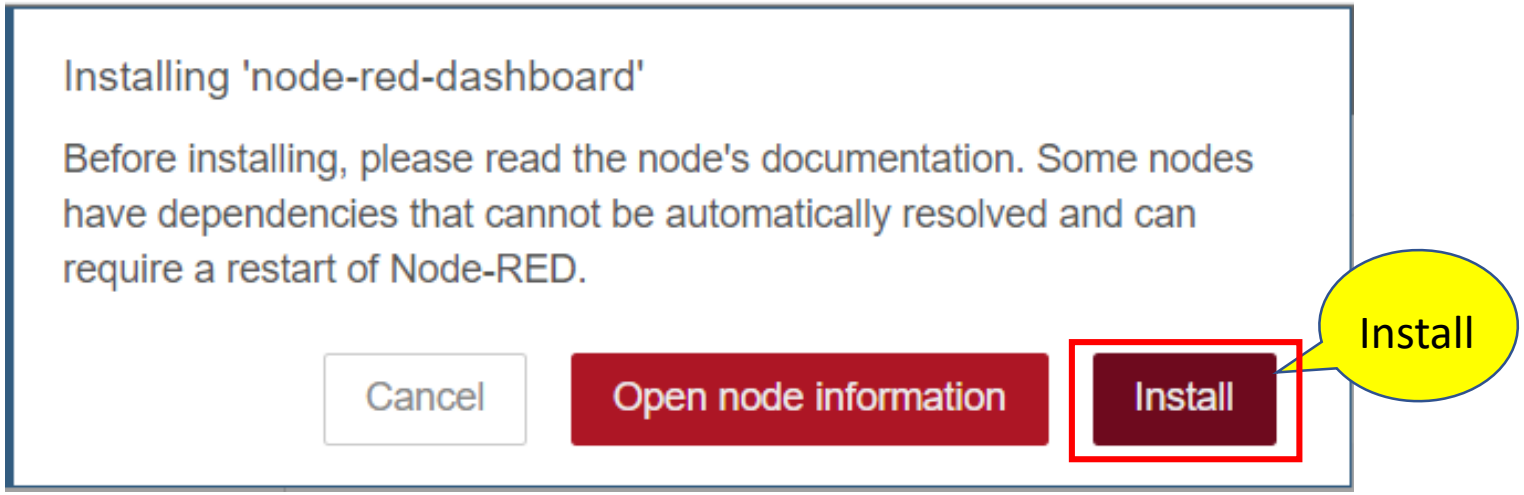
Manage palette



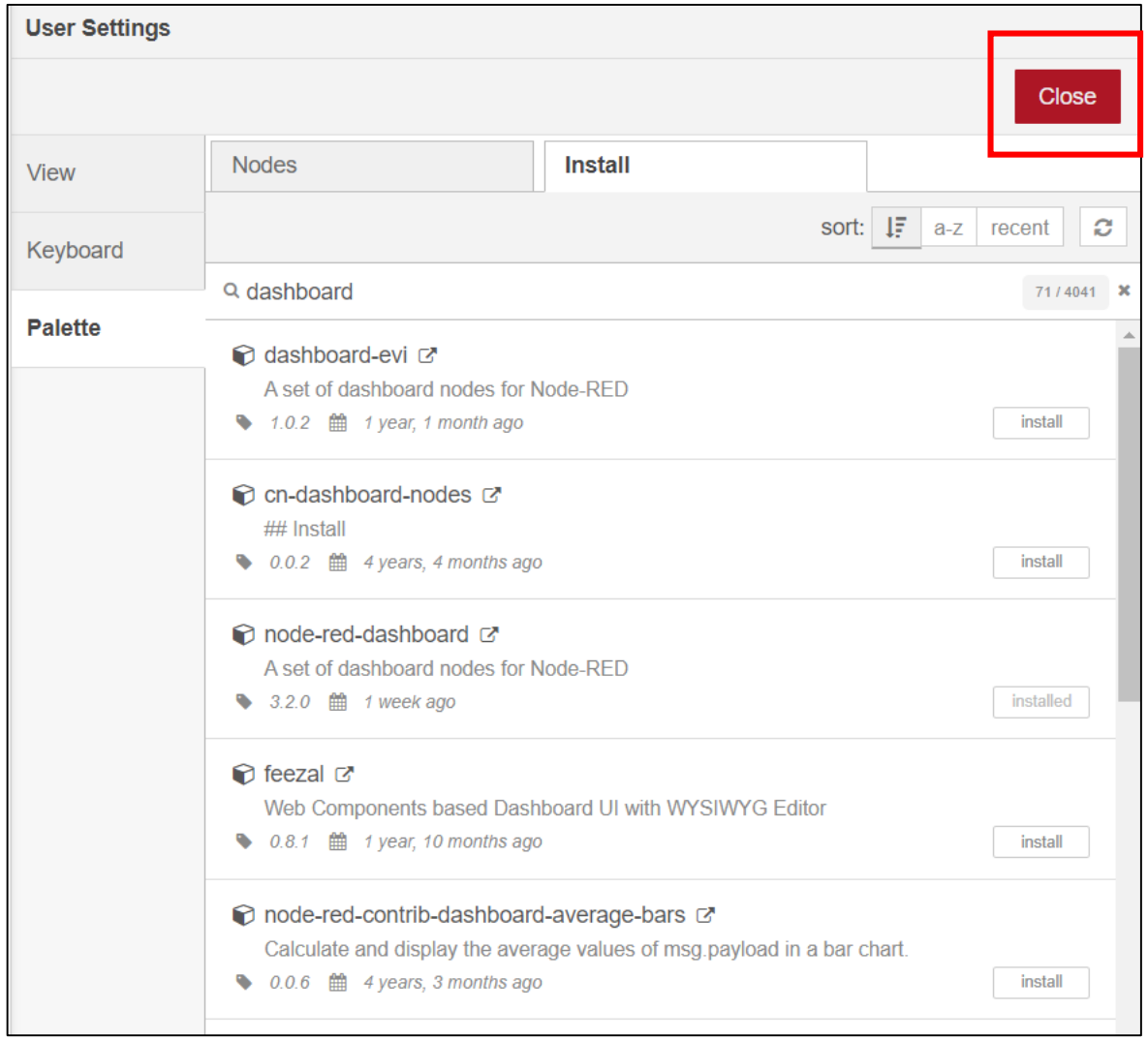
Install


User Settings
Close
View
Keyboard
Palette
Nodes
Install
sort:
a-z
recent
dashboard
71 / 4041
dashboard-evi
A set of dashboard nodes for Node-RED
1.0.2
1 year, 1 month ago
install
cn-dashboard-nodes
Install
0.0.2
4 years, 4 months ago
install
node-red-dashboard
A set of dashboard nodes for Node-RED
3.2.0
1 week ago
install
feezal
Web Components based Dashboard UI with WYSIWYG Editor
0.8.1
1 year, 10 months ago
install
node-red-contrib-dashboard-average-bars
Calculate and display the average values of msg.payload in a bar chart.
0.0.6
4 years, 3 months ago
install
node-red-dashboard
Install

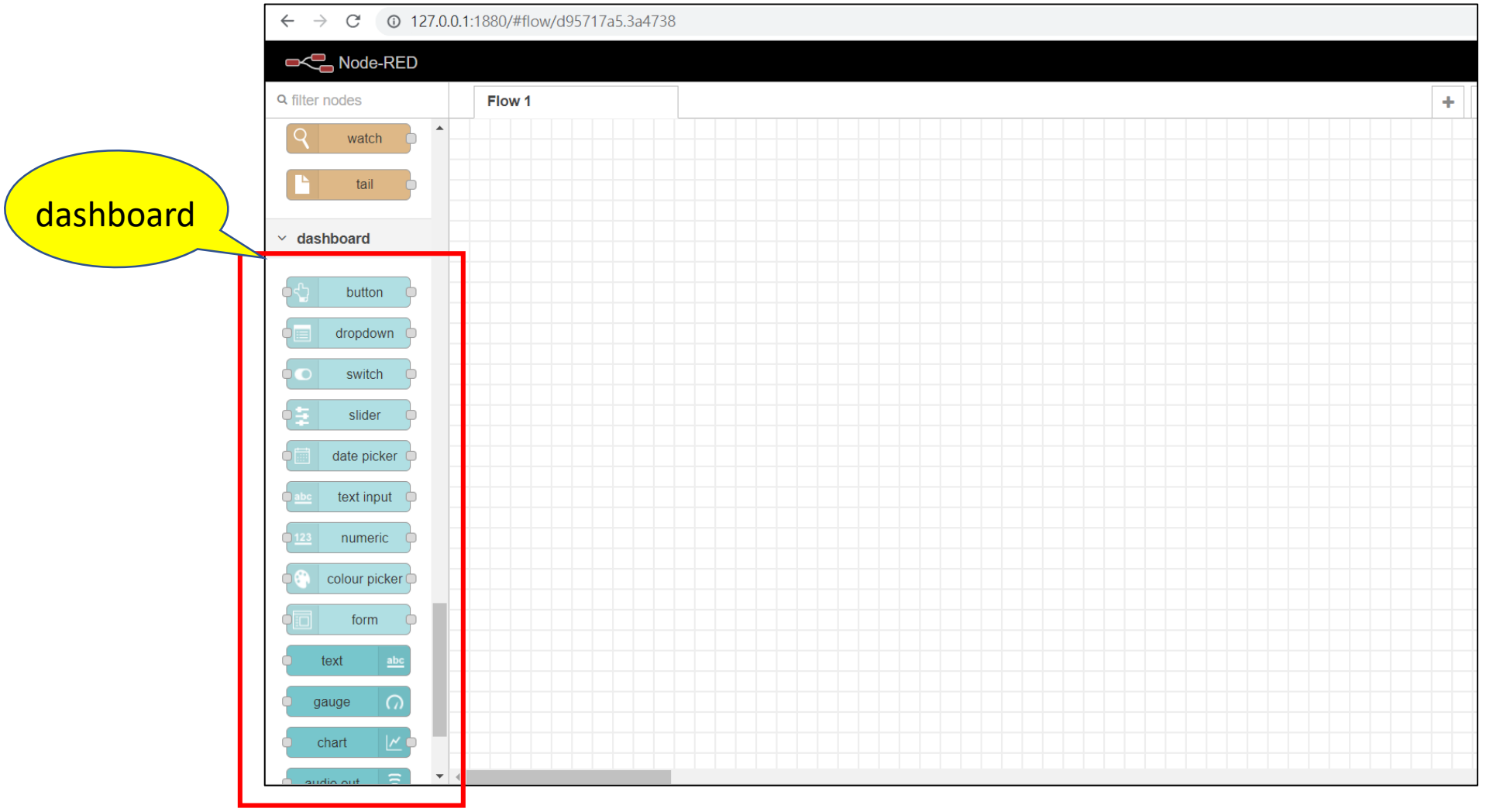
Install


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
Keyboard
Palette
Nodes
Install
sort:
a-z
recent
dashboard
71 / 4041
dashboard-evi
A set of dashboard nodes for Node-RED
1.0.2
1 year, 1 month ago
install
cn-dashboard-nodes
Install
0.0.2
4 years, 4 months ago
install
node-red-dashboard
A set of dashboard nodes for Node-RED
3.2.0
1 week ago
installed
feezal
Web Components based Dashboard UI with WYSIWYG Editor
0.8.1
1 year, 10 months ago
install
node-red-contrib-dashboard-average-bars
Calculate and display the average values of msg.payload in a bar chart.
0.0.6
4 years, 3 months ago
install

dashboard

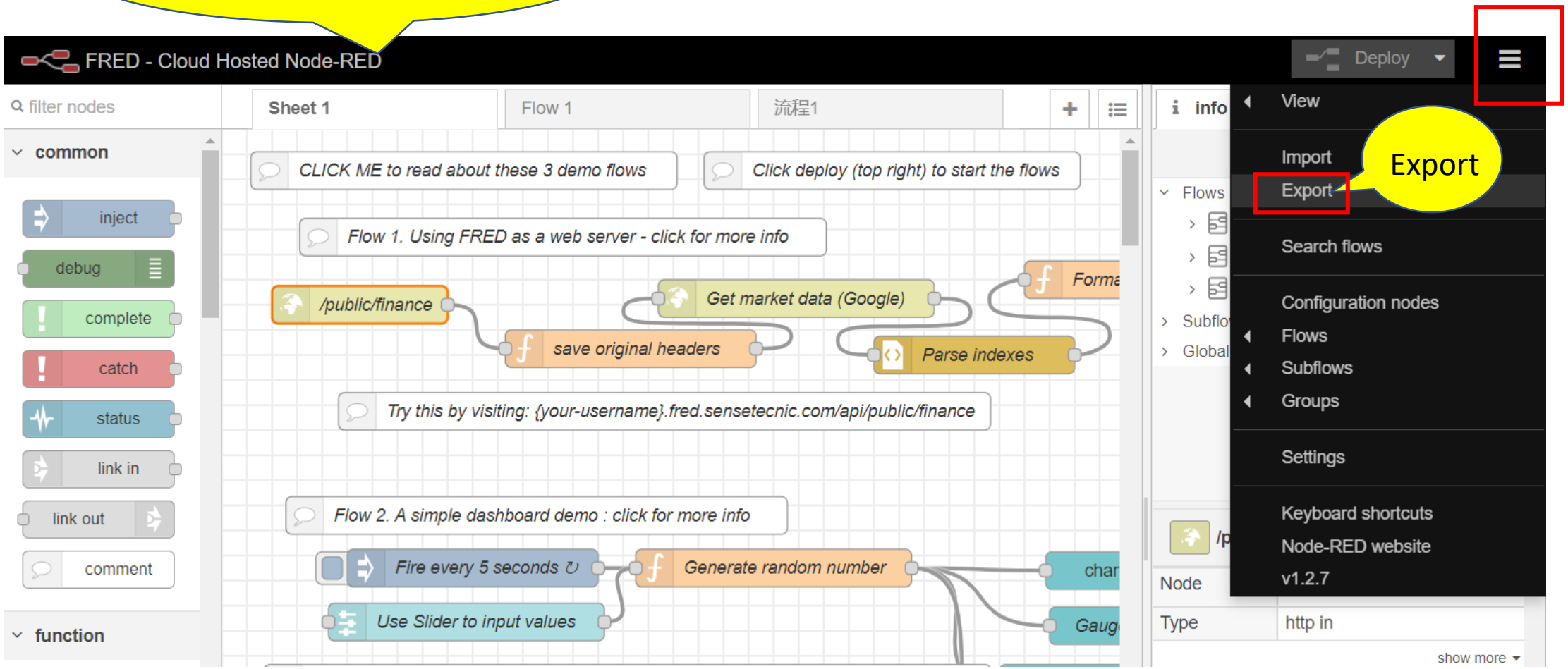

dashboard
127.0.0.1:1880/#flow/d95717a5.3a4738
Node-RED
filter nodes
Flow 1
watch
tail
dashboard
button
dropdown
switch
slider
date picker
text input
numeric
colour picker
form
text
gauge
chart

Exercise 3-4

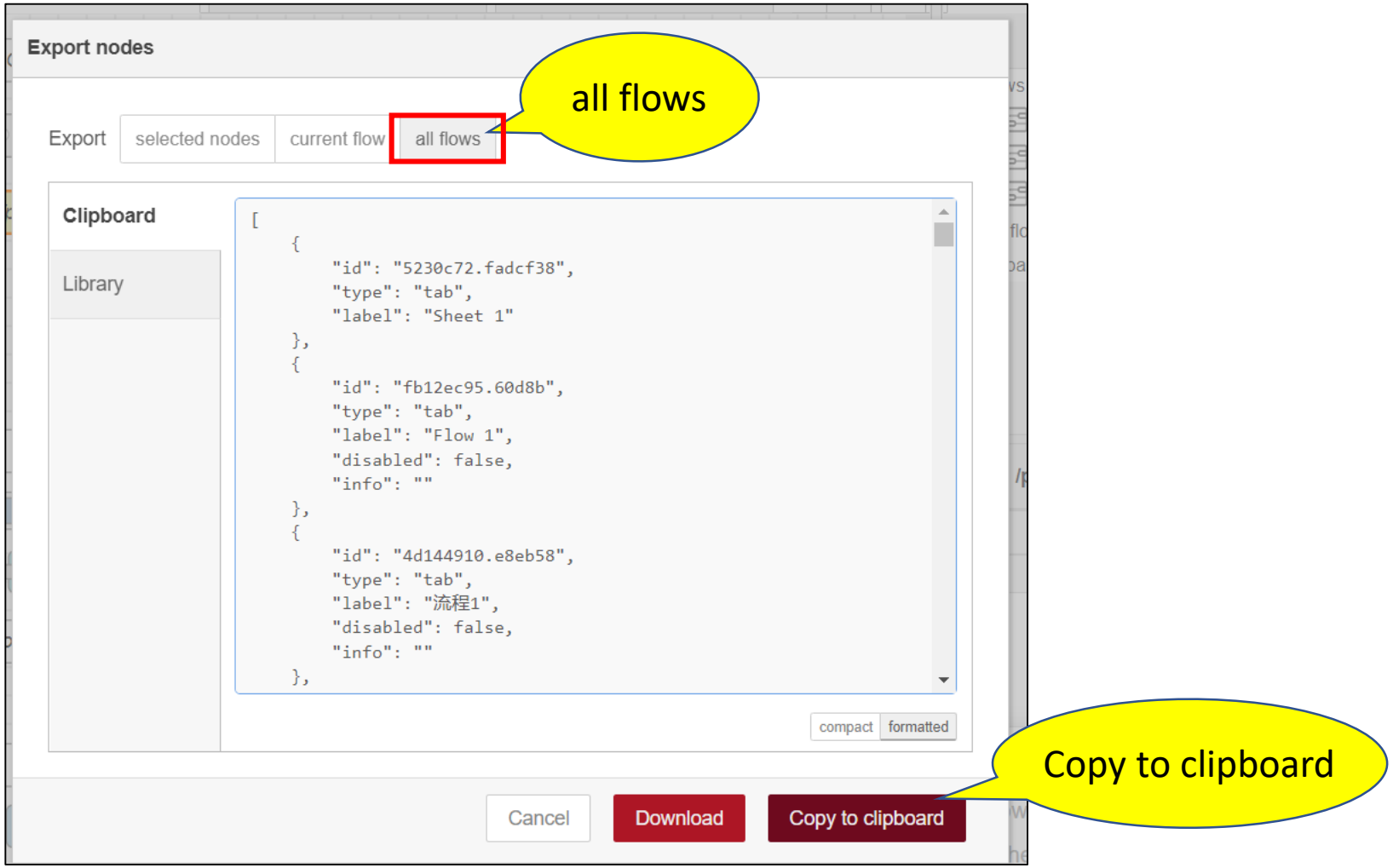
- Copy flows in fred to local node-red.

Export

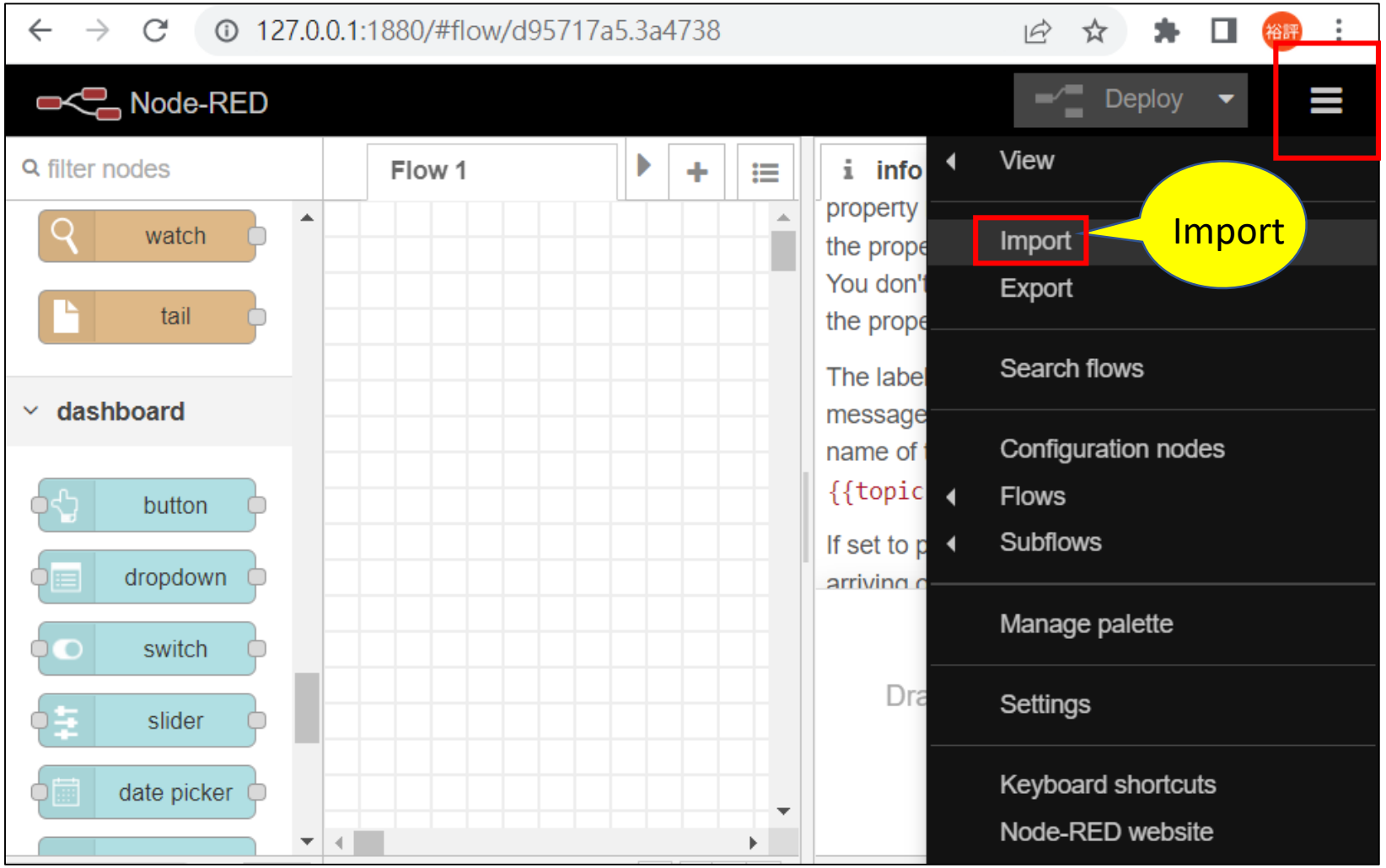
https://fred.sensetecnic.com/



Copy to clipboard



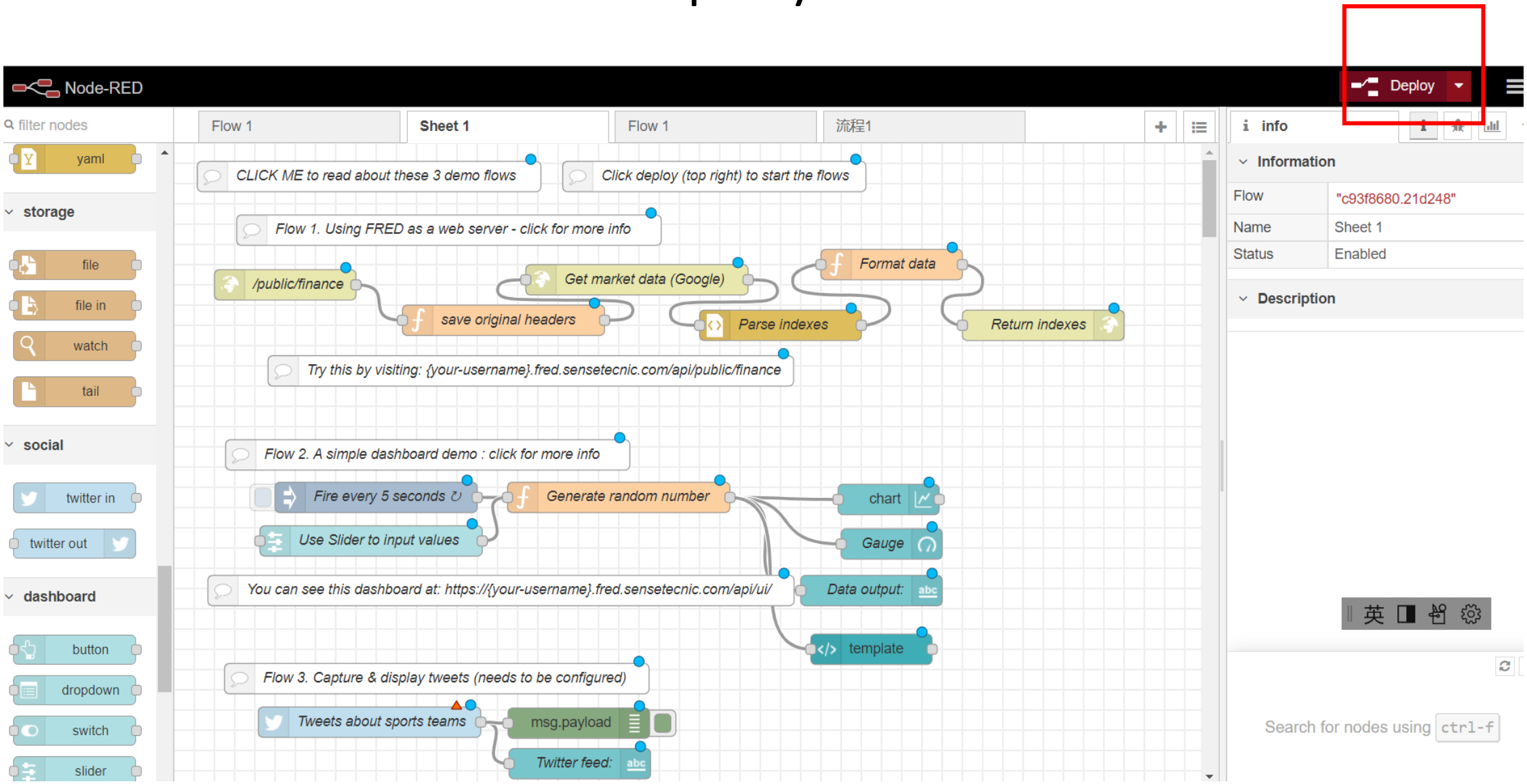
127.0.0.1:1880


127.0.0.1:1880/#flow/d95717a5.3a4738
Node-RED
Deploy
filter nodes
Flow 1
info
watch
tail
dashboard
button
dropdown
switch
slider
date picker
View
Import
Export
Search flows
Configuration nodes
Flows
Subflows
Manage palette
Settings
Keyboard shortcuts
Node-RED website
Import

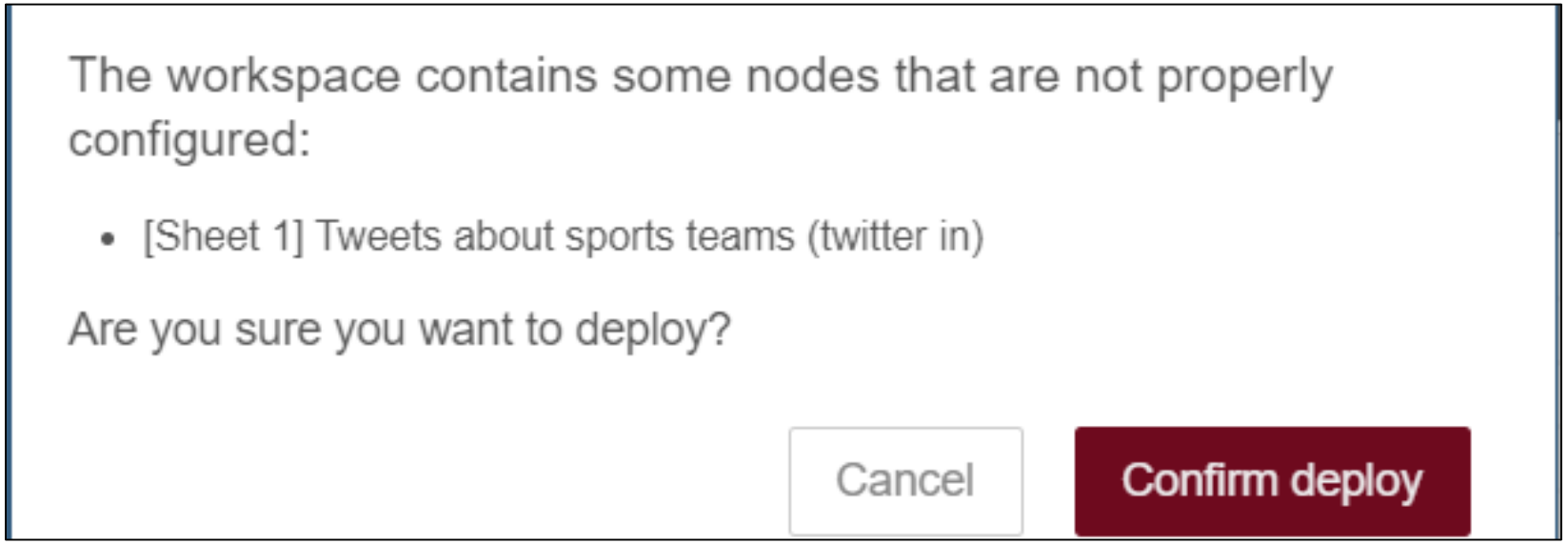
Paste

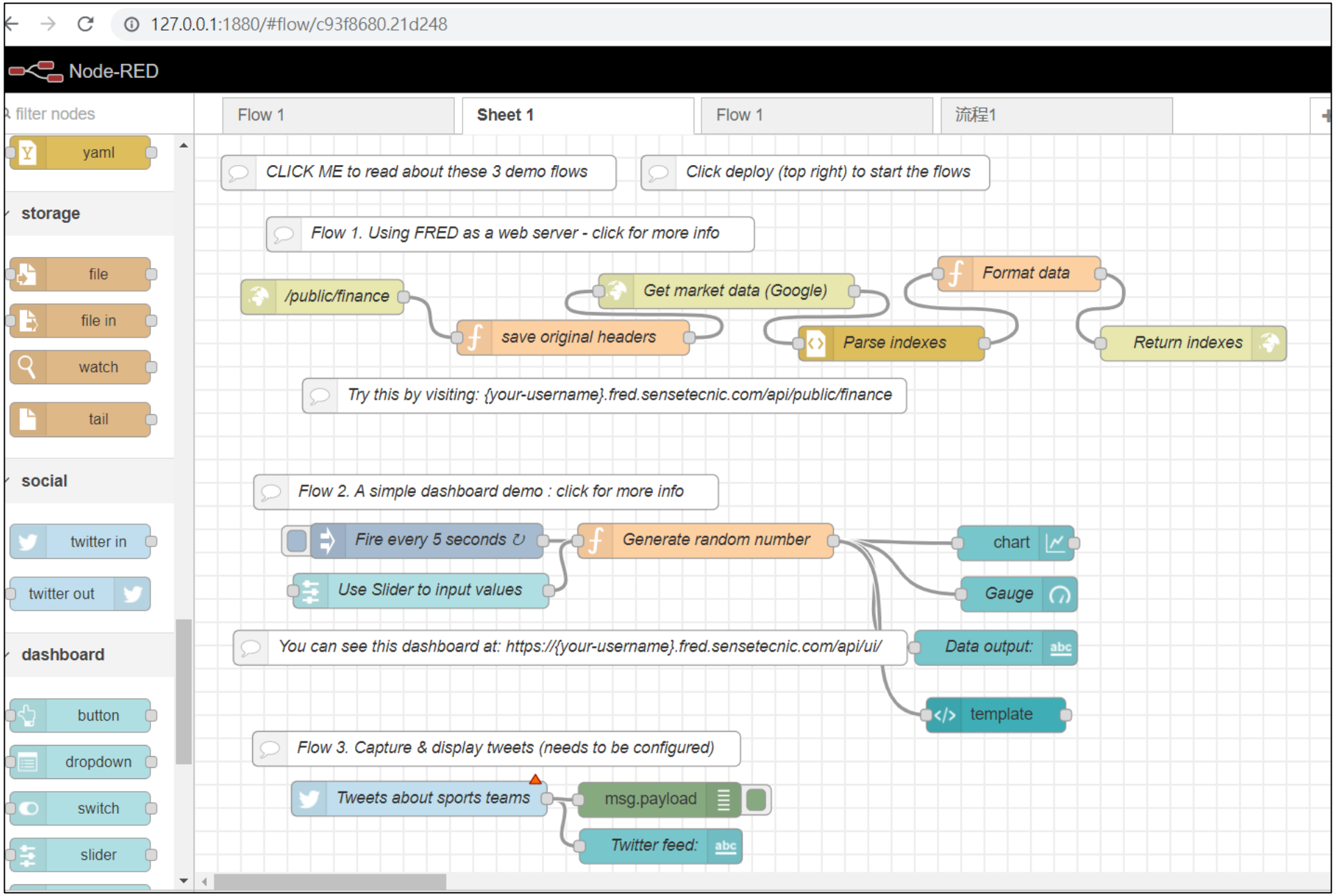

Import nodes
Clipboard
Paste flow json or
select a file to import
Library
Examples
"z": "fb12ec95.60d8b",
"name": "",
"func": "msg.payload={id:\"b12200000\",
name:\"yupingliao\",age:53};\nreturn msg;",
"outputs": 1,
"noerr": 0,
"initialize": "",
"finalize": "",
"x": 230,
"y": 240,
"wires": [
[
"c0ca81c9.3f358"
]
]
}
]
1.paste
Import to
current flow
new flow
Cancel
Import
2.Import

Deploy


Node-RED
Deploy
filter nodes
Flow 1
Sheet 1
Flow 1
流程1
info
yaml
storage
file
file in
watch
tail
social
twitter in
twitter out
dashboard
button
dropdown
switch
slider
CLICK ME to read about these 3 demo flows
Click deploy (top right) to start the flows
Flow 1. Using FRED as a web server - click for more info
/public/finance
Get market data (Google)
Format data
save original headers
Parse indexes
Return indexes
Try this by visiting: {your-username}.fred.sensetecnic.com/api/public/finance
Flow 2. A simple dashboard demo : click for more info
Fire every 5 seconds
Generate random number
chart
Use Slider to input values
Gauge
You can see this dashboard at: https://{your-username}.fred.sensetecnic.com/api/ui/
Data output:
template
Flow 3. Capture & display tweets (needs to be configured)
Tweets about sports teams
msg.payload
Twitter feed:
Information
Flow
"c93f8680.21d248"
Name
Sheet 1
Status
Enabled
Description
Search for nodes using ctrl-f

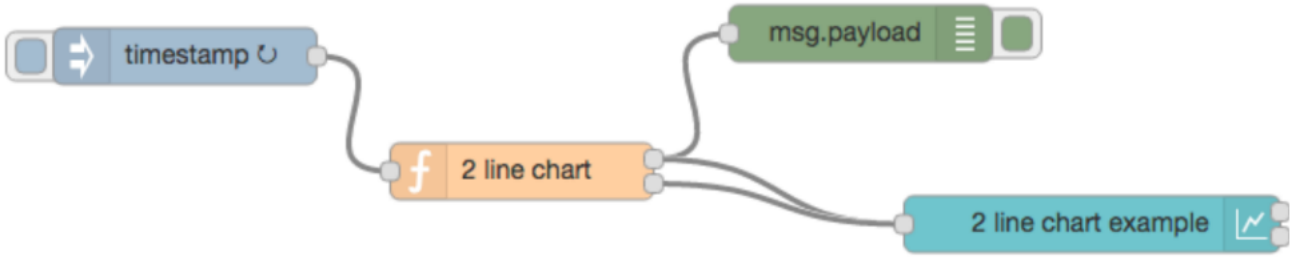
Confirm deploy

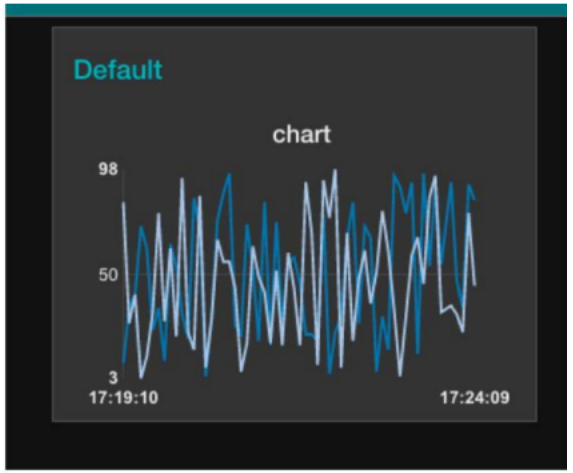



127.0.0.1:1880/#flow/c93f8680.21d248
Node-RED
filter nodes
yaml
storage
file
file in
watch
tail
social
twitter in
twitter out
dashboard
button
dropdown
switch
slider
Flow 1
Sheet 1
Flow 1
流程1
CLICK ME to read about these 3 demo flows
Click deploy (top right) to start the flows
Flow 1. Using FRED as a web server - click for more info
/public/finance
save original headers
Get market data (Google)
Parse indexes
Format data
Return indexes
Try this by visiting: {your-username}.fred.sensetecnic.com/api/public/finance
Flow 2. A simple dashboard demo : click for more info
Fire every 5 seconds
Use Slider to input values
Generate random number
chart
Gauge
Data output:
template
You can see this dashboard at: https://{your-username}.fred.sensetecnic.com/api/ui/
Flow 3. Capture & display tweets (needs to be configured)
Tweets about sports teams
msg.payload
Twitter feed:

Exercise 3-5

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



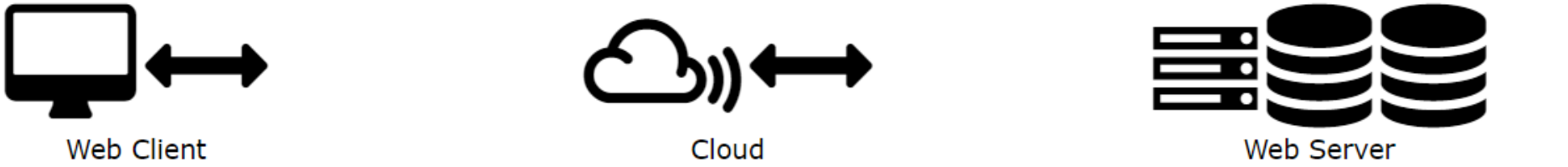


Hyper text Transfer Protocol

- **HTTP** stands for **H**yper **T**ext **T**ransfer **P**rotocol
- **WWW** is about communication between web **clients** and **servers**
- Communication between client computers and web servers is done by sending **HTTP Requests** and receiving **HTTP Responses**

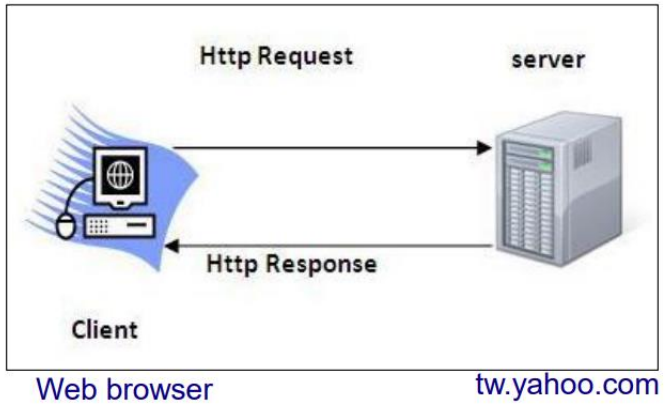
World Wide Web Communication

- he World Wide Web is about communication between web **clients** and web **servers**.
- **Clients** are often browsers (Chrome, Edge, Safari), but they can be any type of program or device.
- **Servers** are most often computers in the cloud.
-

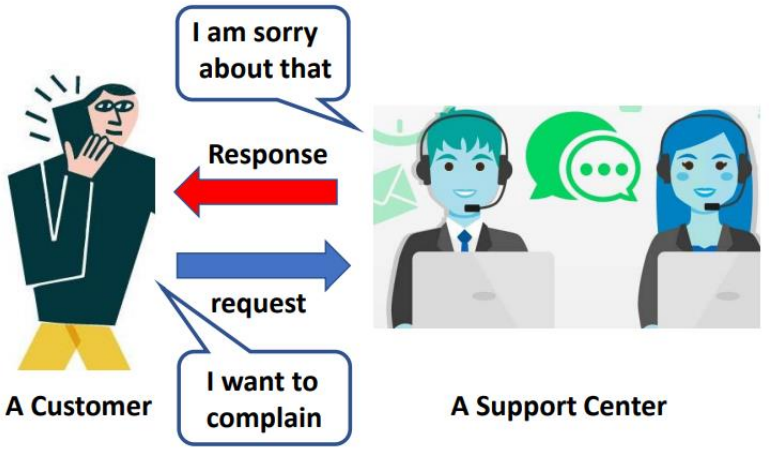


HTTP Request / Response

- Communication between clients and servers is done by **requests** and **responses**:
- A client (a browser) sends an **HTTP request** to the web
- A web server receives the request
- The server runs an application to process the request
- The server returns an **HTTP response** (output) to the browser
- The client (the browser) receives the response



HTTP
(HyperText Transfer Protocol)



What is Firebase ?

- Firebase is Google's mobile platform that helps you quickly develop high-quality apps and grow your business.

How data is structured: it's a JSON tree

```
{
  "users": {
    "alanisawesome": {
      "date_of_birth": "June 23, 1912",
      "full_name": "Alan Turing",
      "nickname": "Alan The Machine"
    },
    "gracehop": {
      "date_of_birth": "December 9, 1906",
      "full_name": "Grace Hopper",
      "nickname": "Amazing Grace"
    }
  }
}
```

Key-Value Stores
「鍵-值」資料儲存法

https://firebase.google.com/docs/database/rest/structure-data

Firebase Realtime Database REST: Saving Data

- **Saving Data**

Ways to Save Data	
PUT	Write or replace data to a **defined path**, like `fireblog/users/user1/<data>`
PATCH	Update some of the keys for a defined path without replacing all of the data.
POST	**Add to a list of data** in our Firebase database. Every time we send a POST request, the Firebase client generates a unique key, like `fireblog/users/<unique-id>/<data>`
DELETE	Remove data from the specified Firebase database reference.

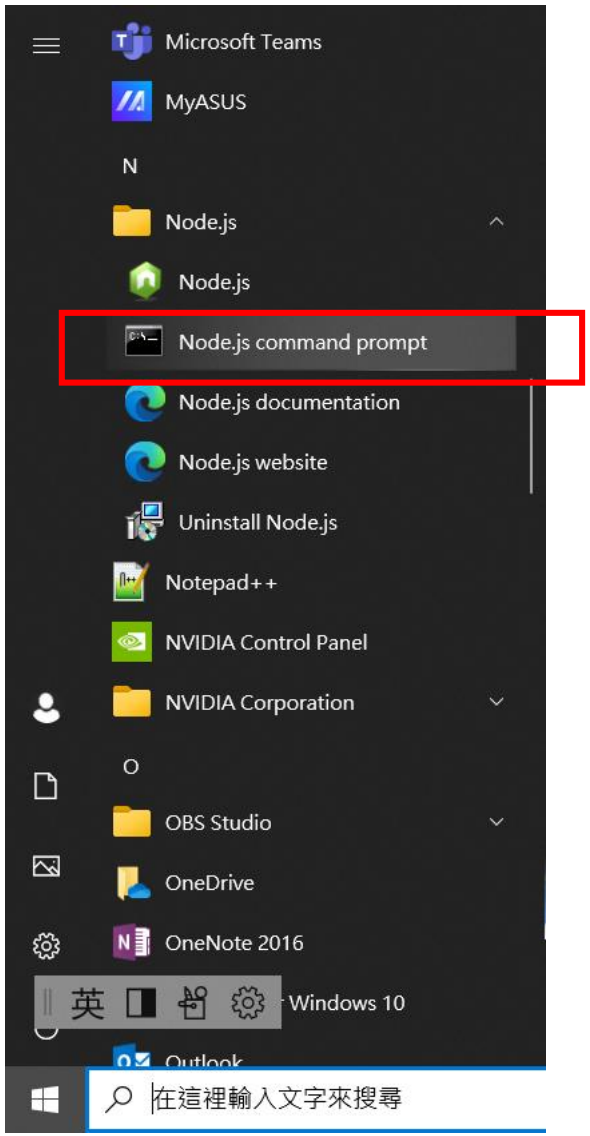
https://firebase.google.com/docs/database/rest/save-data

curl -X PUT -d '"Alan Turing"' \ 'https://docs-examples.firebaseio.com/fireblog/users/alanisawesome/name.json'

Writing Data with PUT

- Write or replace data to a defined path

```
curl -X PUT -d '{
  "alanisawesome": {
    "name": "Alan Turing",
    "birthday": "June 23, 1912"
  }
}' 'https://docs-examples.firebaseio.com/fireblog/users.json'
```

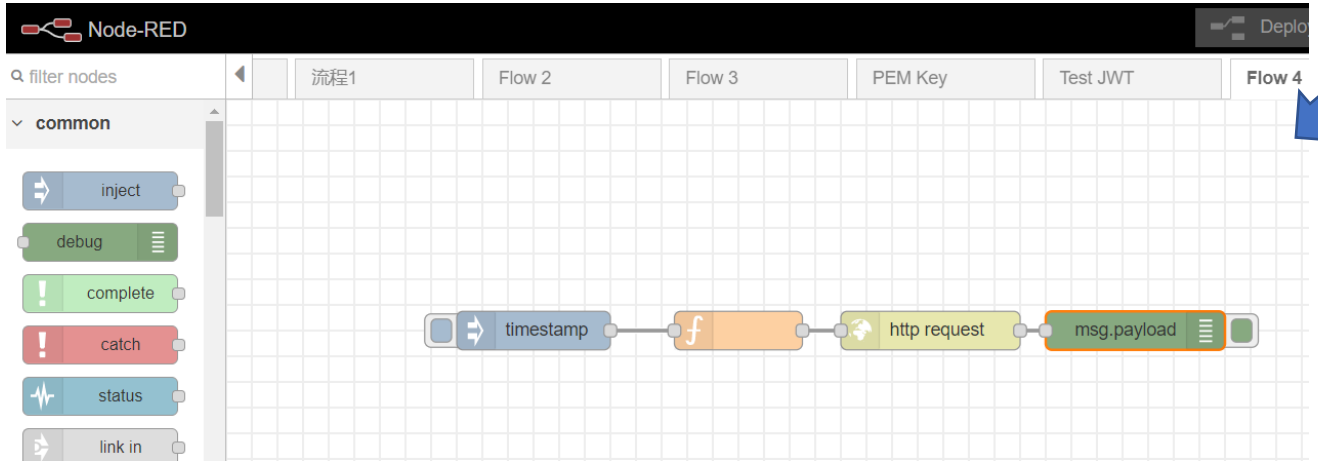


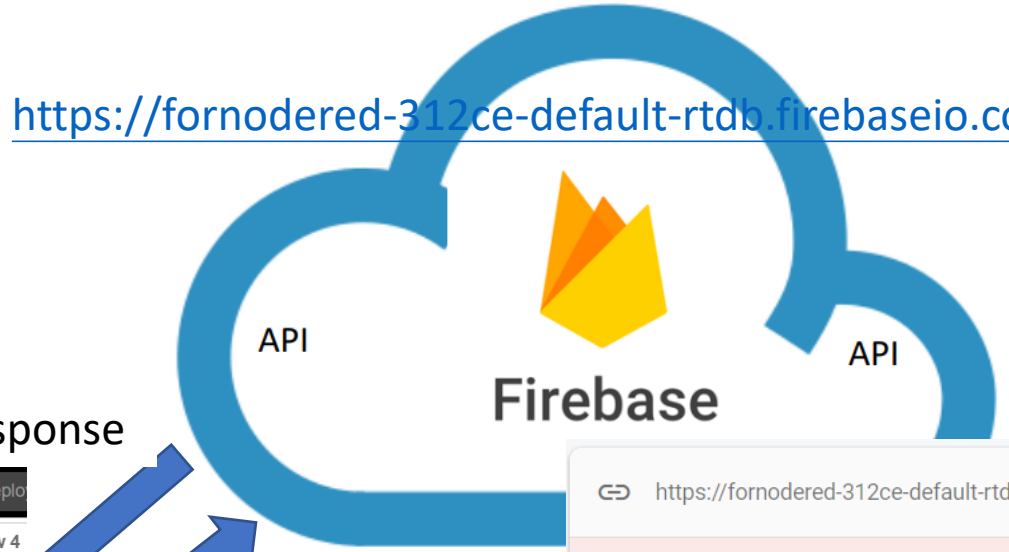
node-red

```
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]
```

Exercise 3-6

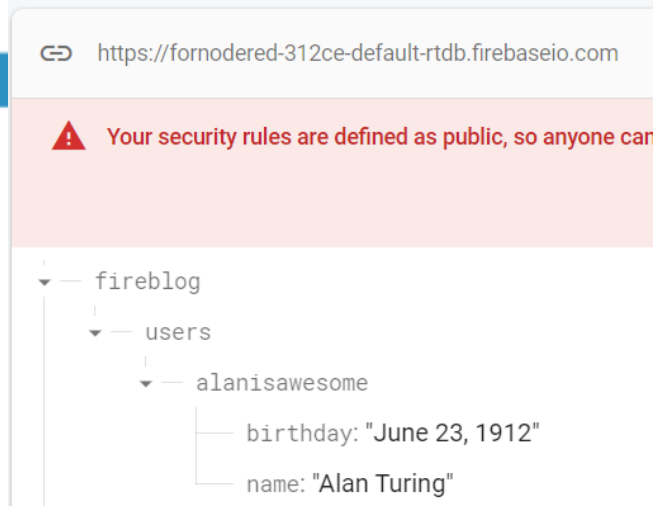
https://fornodered-312ce-default-rtdb.firebaseio.com

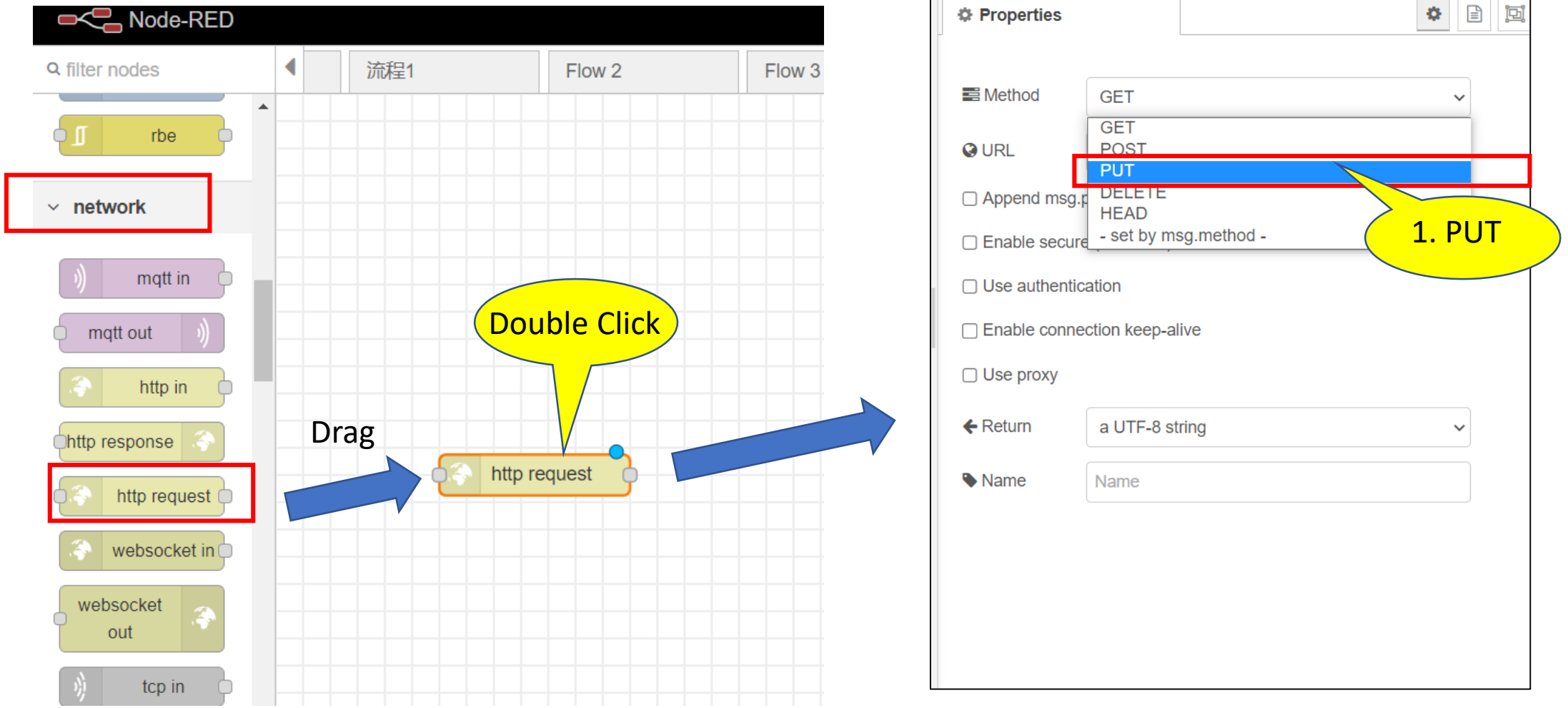




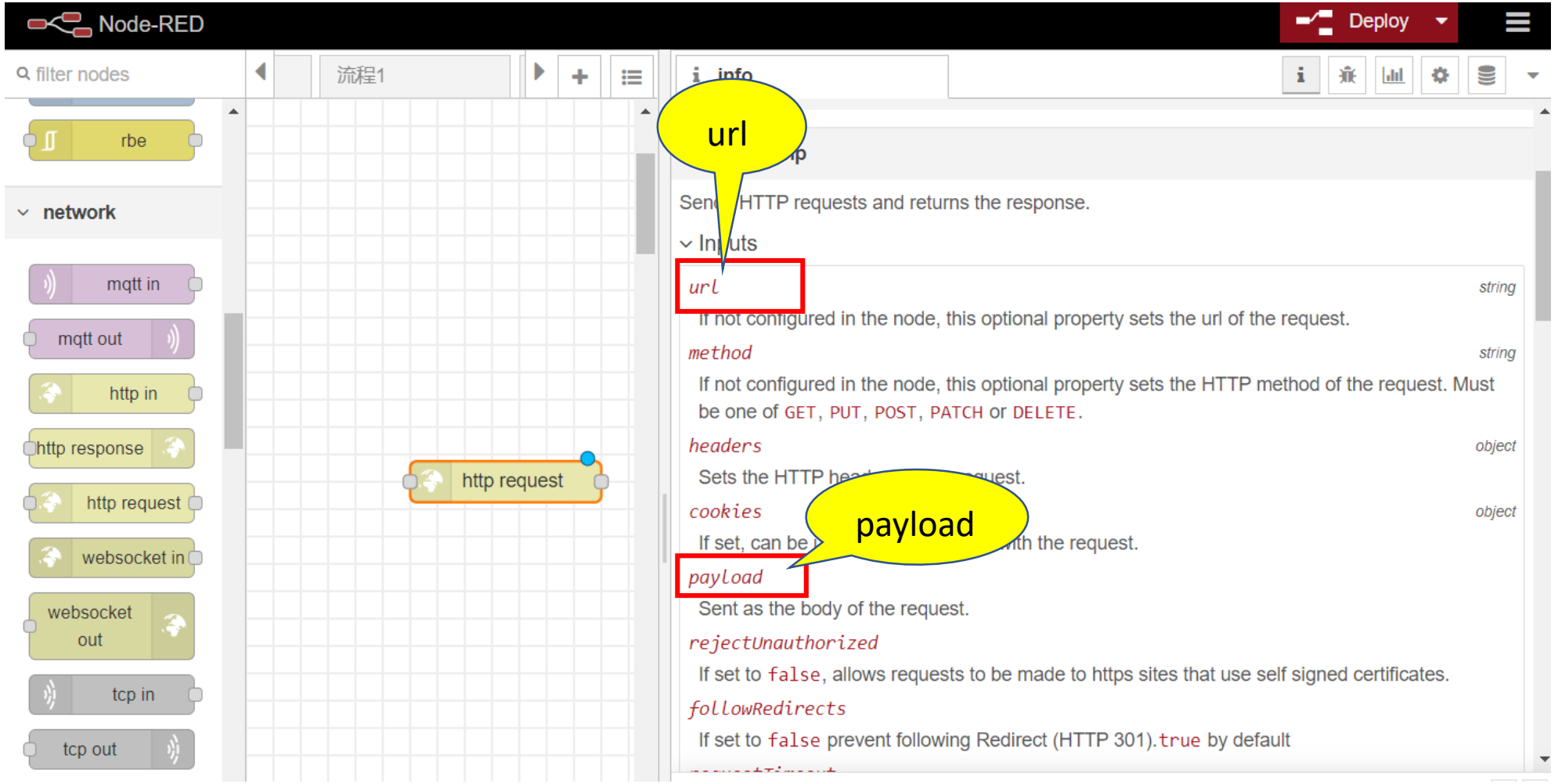
Response

HTTP PUT

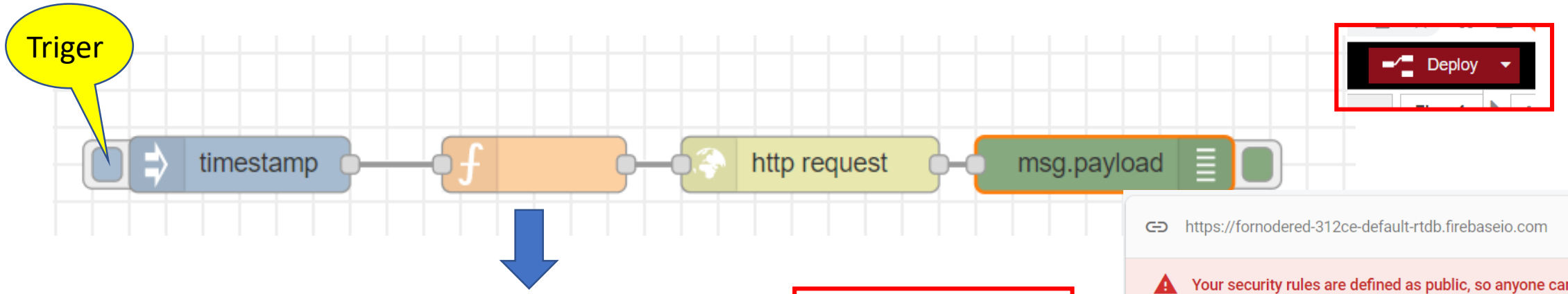



Node-RED
filter nodes
流程1
Flow 2
Flow 3
rbe
network
mqtt in
mqtt out
http in
http response
http request
websocket in
websocket out
tcp in
Drag
Double Click
http request
2.Done
Edit http request node
Delete
Cancel
Done
Properties
Method
GET
URL
GET
POST
PUT
DELETE
HEAD
- set by msg.method -
1. PUT
Append msg.p
Enable secure
Use authentication
Enable connection keep-alive
Use proxy
Return
a UTF-8 string
Name
Name

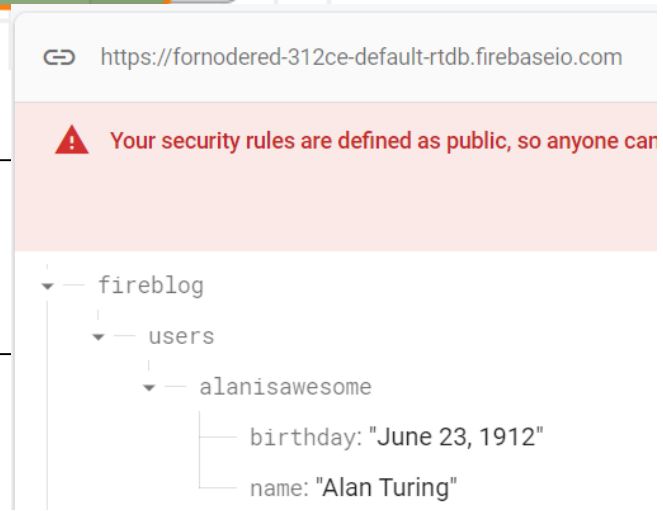
http request node

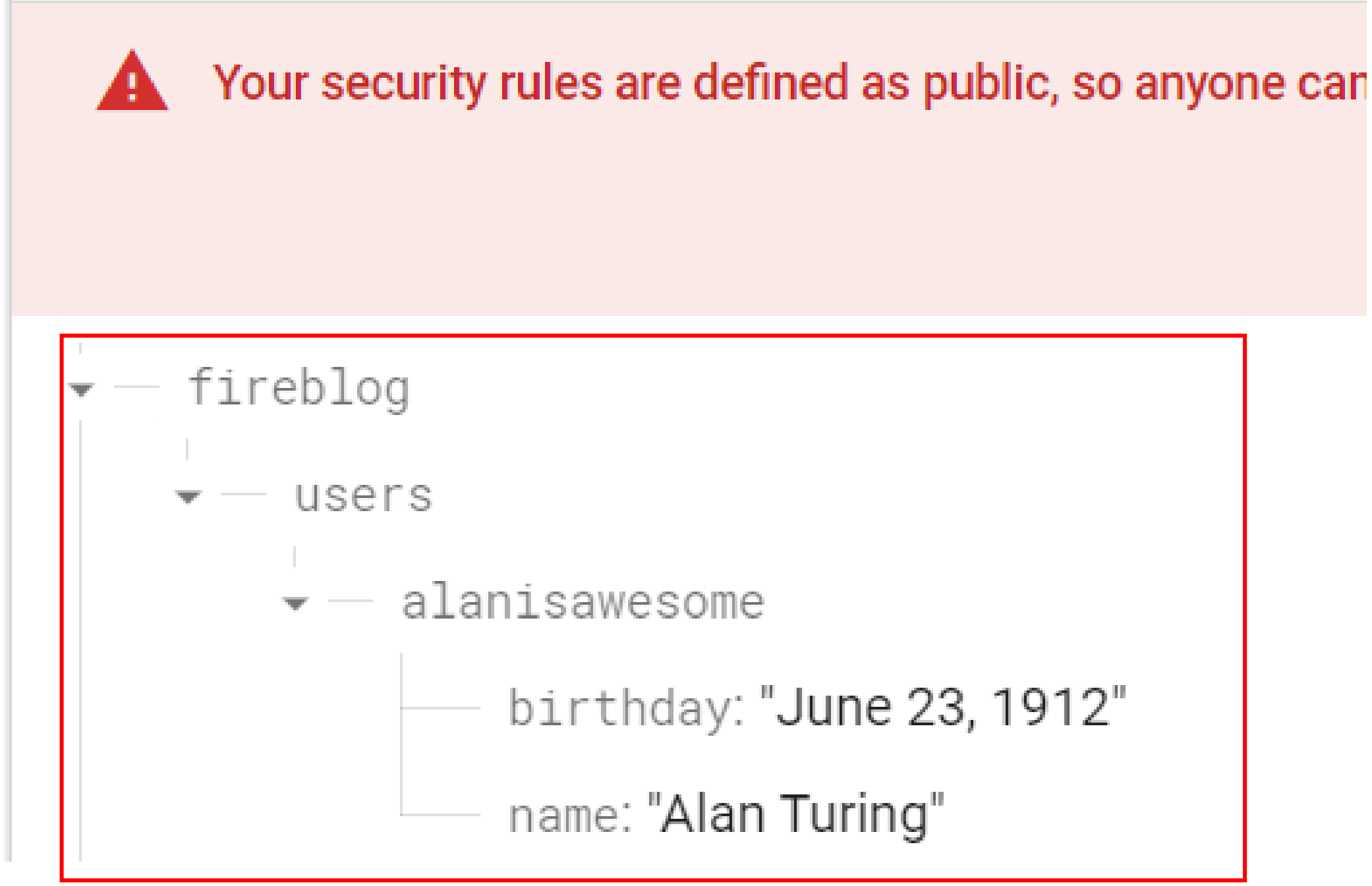


```
curl -X PUT -d '{
  "alanisawesome": {
    "name": "Alan Turing",
    "birthday": "June 23, 1912"
  }
}' 'https://docs-examples.firebaseio.com/fireblog/users.json'
```




```
msg.url=https://fornodered-312ce-default-rtdb.firebaseio.com/fireblog/users.json
msg.payload={"alanisawesome": {"name": "Alan Turing", "birthday": "June 23, 1912"}};
return msg;
```



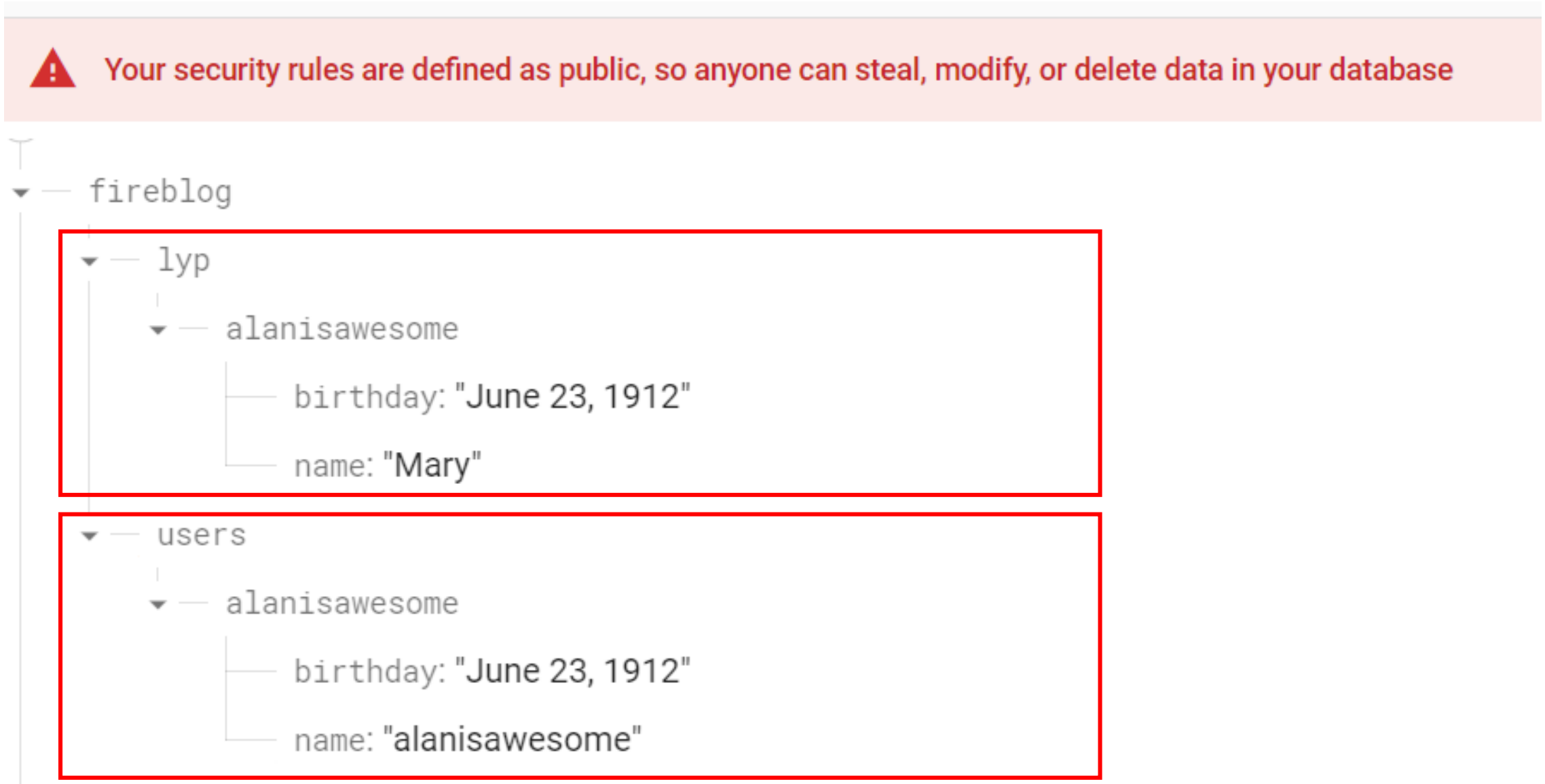

Your security rules are defined as public, so anyone can
fireblog
users
alanisawesome
birthday: "June 23, 1912"
name: "Alan Turing"

Change -> Deploy -> Triger

Student ID

```
msg.url=https://fornodered-312ce-default-rtdb.firebaseio.com/fireblog/b1221111.json
msg.payload={"alanisawesome": {"name":  “Tom", "birthday": "June 23, 1912"}};
return msg;
```

Name


Your security rules are defined as public, so anyone can steal, modify, or delete data in your database
fireblog
lyp
alanisawesome
birthday: "June 23, 1912"
name: "Mary"
users
alanisawesome
birthday: "June 23, 1912"
name: "alanisawesome"

Exercise 3-6

- Try PATCH and POST method to write data to your database.

PATCH : Update some of the keys for a defined path without replacing all of the data.

POST: Add to a list of data in our Firebase database. Every time we send a POST request, the Firebase client generates a unique key