

物聯網實務

9_28

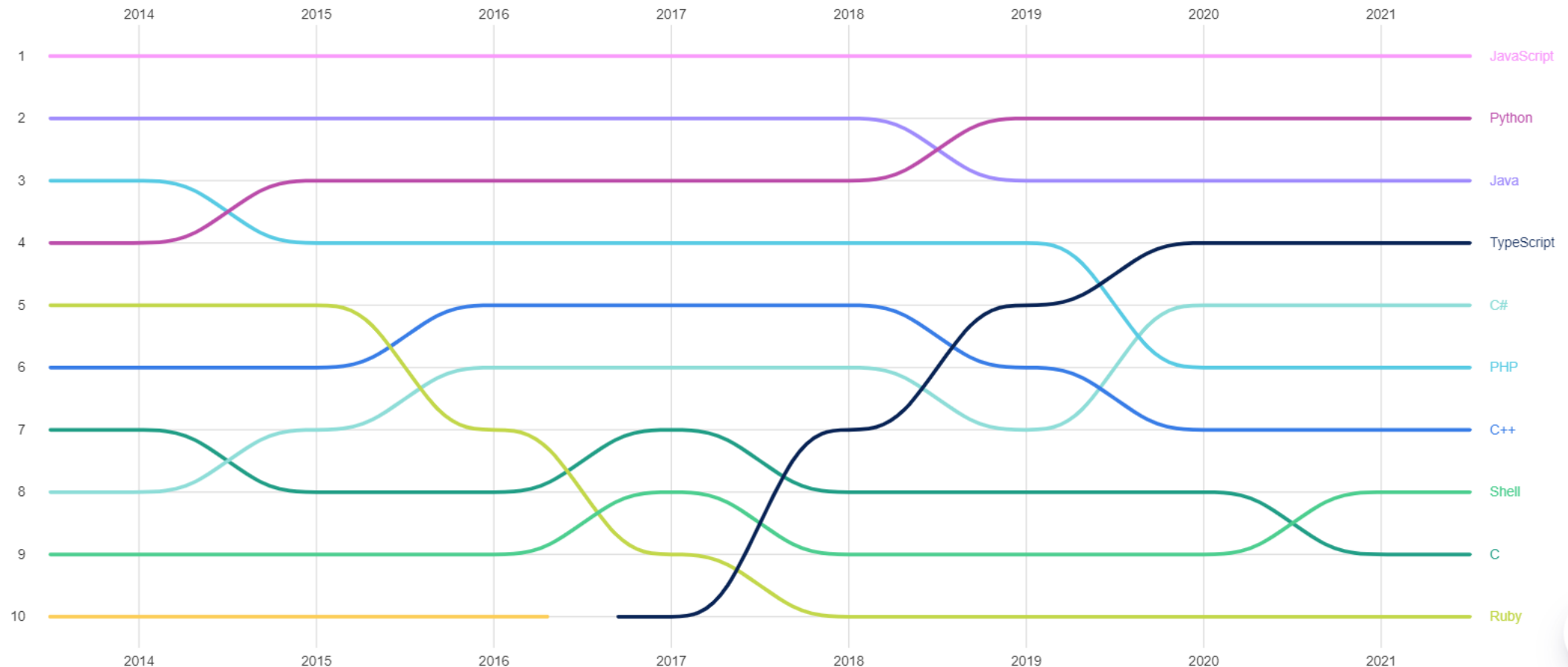
廖裕評

The Most Popular Programming Languages for 2022



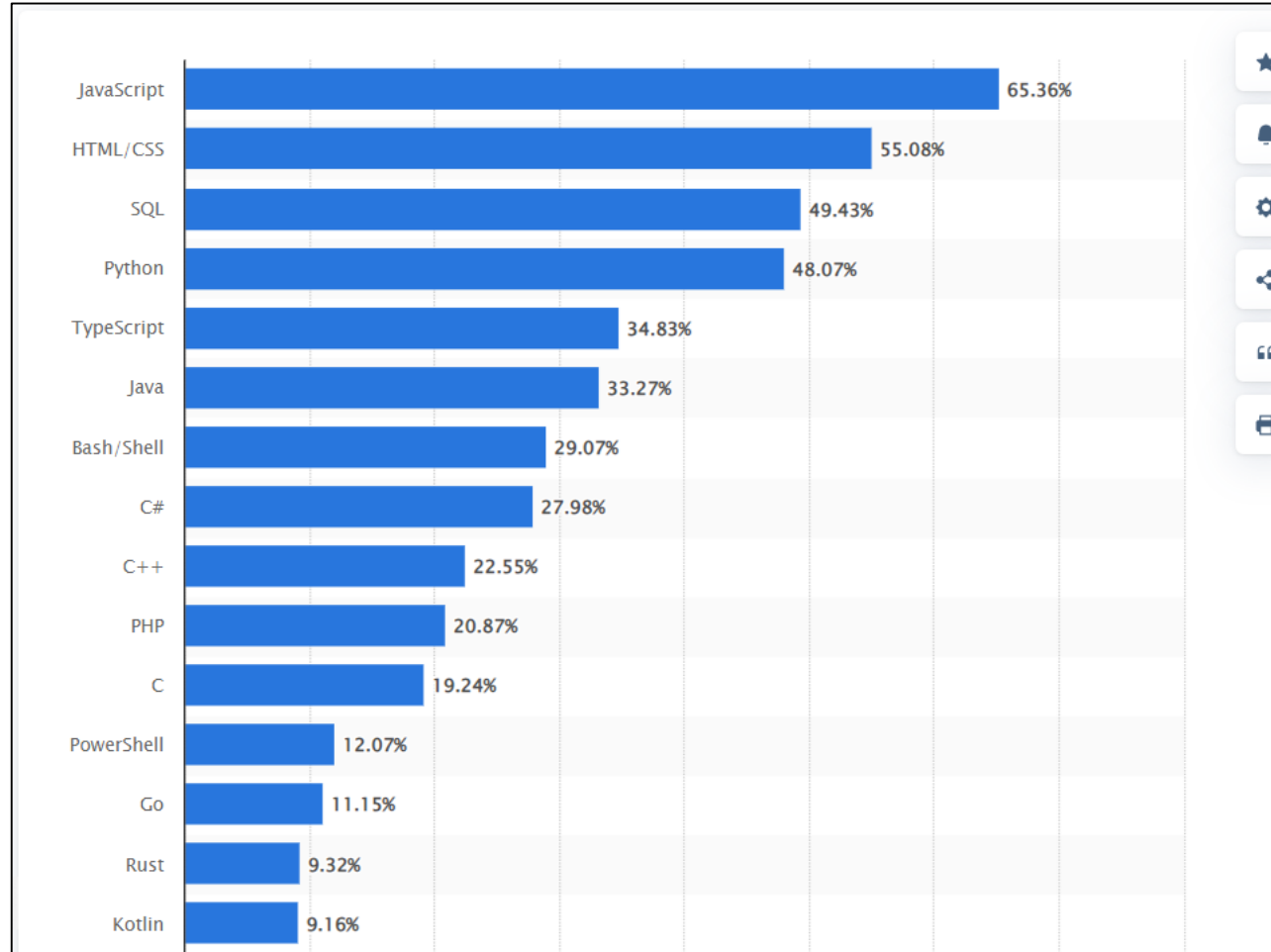
<https://www.magicbell.com/blog/most-popular-programming-languages>

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)



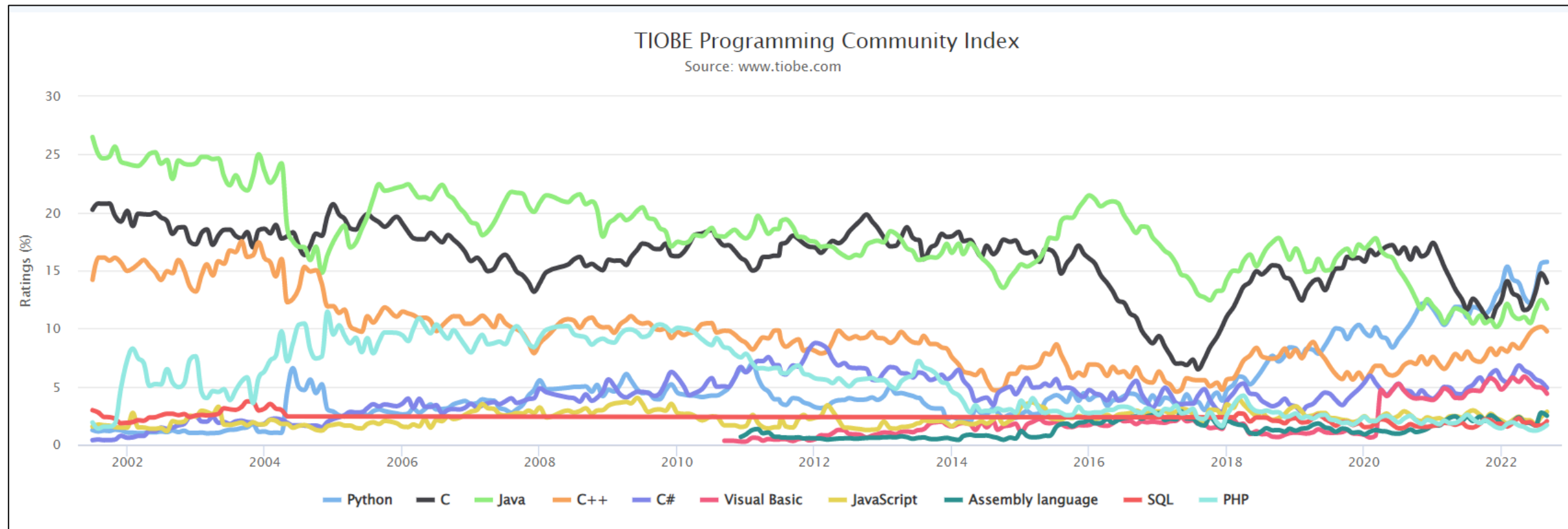
<https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>

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%

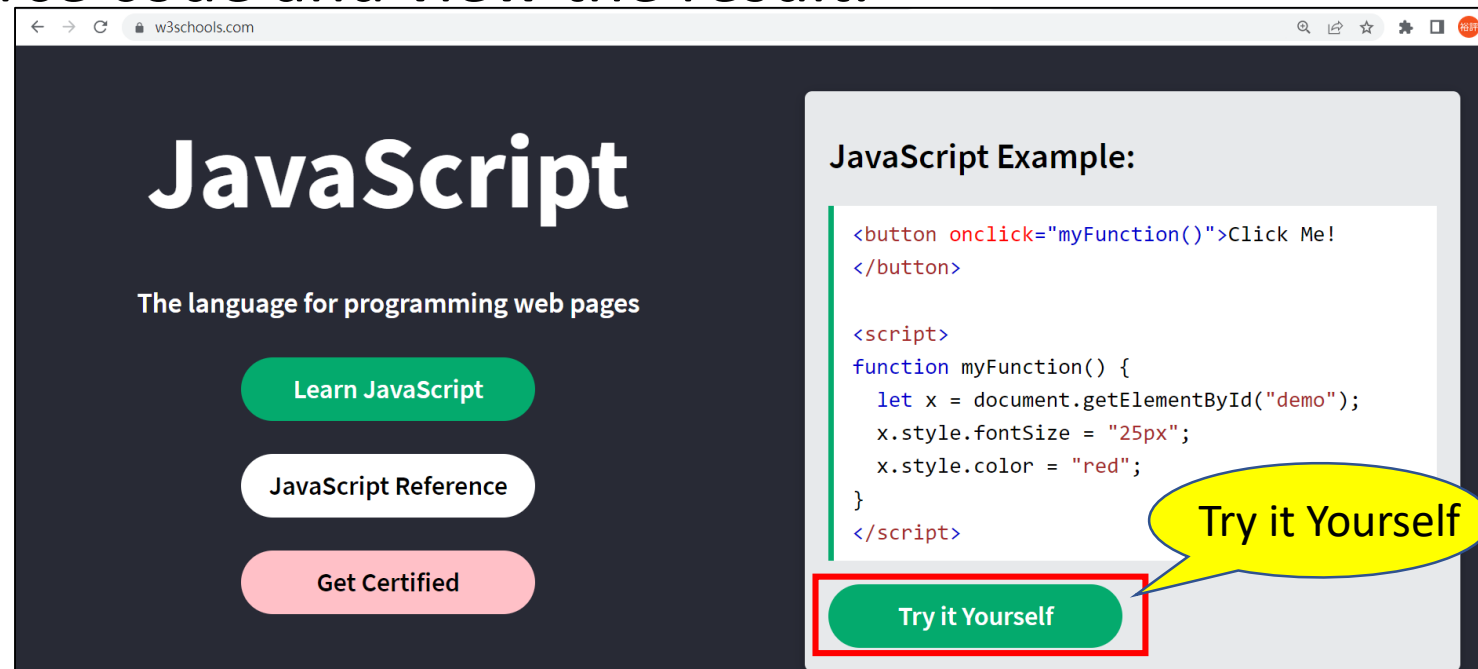
<https://www.tiobe.com/tiobe-index/>

TIOBE Index for September 2022

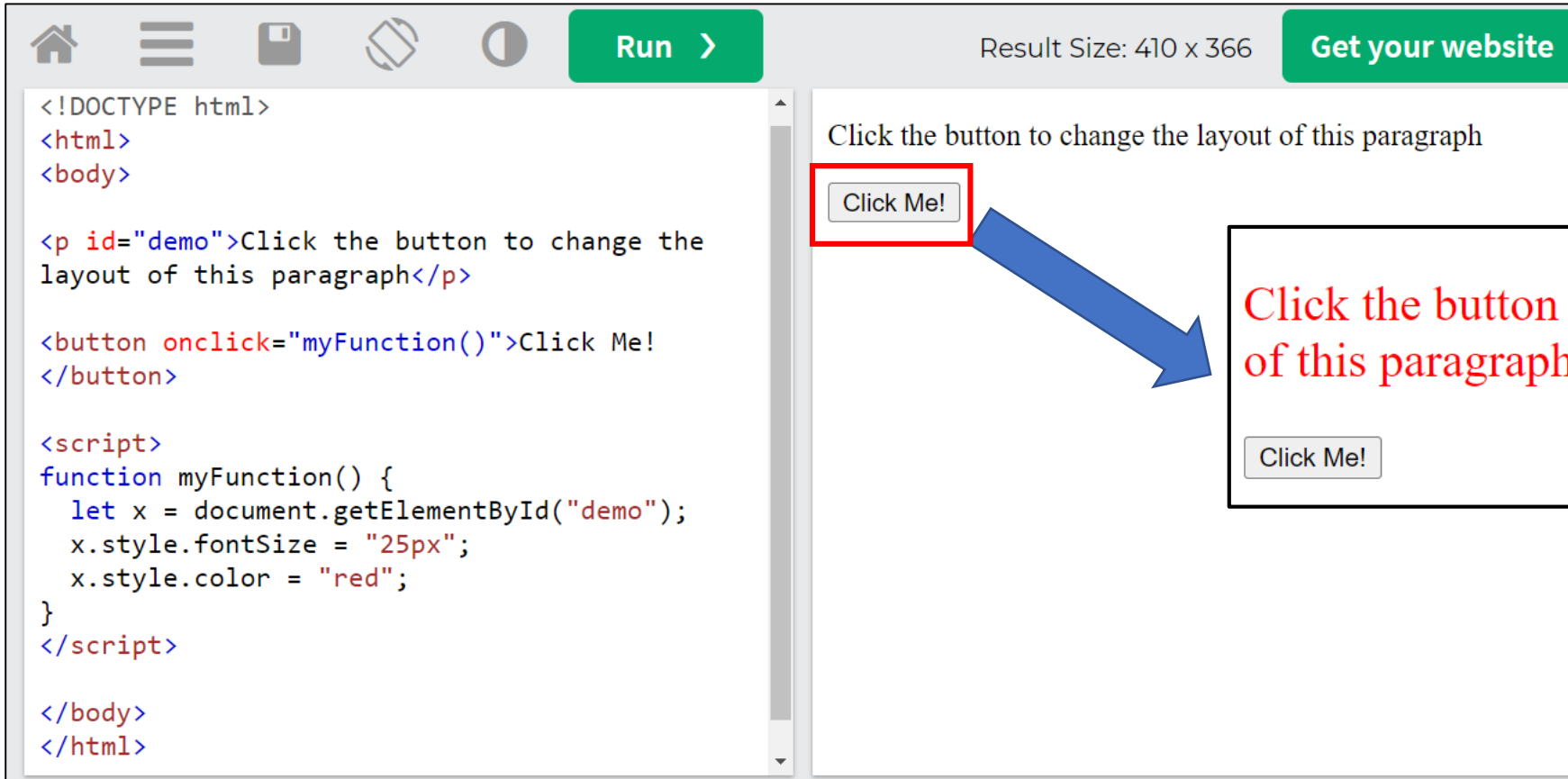


An Introduction to JavaScript

- 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



The screenshot shows a web development interface with a code editor on the left and a live preview on the right. The code editor contains the following HTML and JavaScript code:

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

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

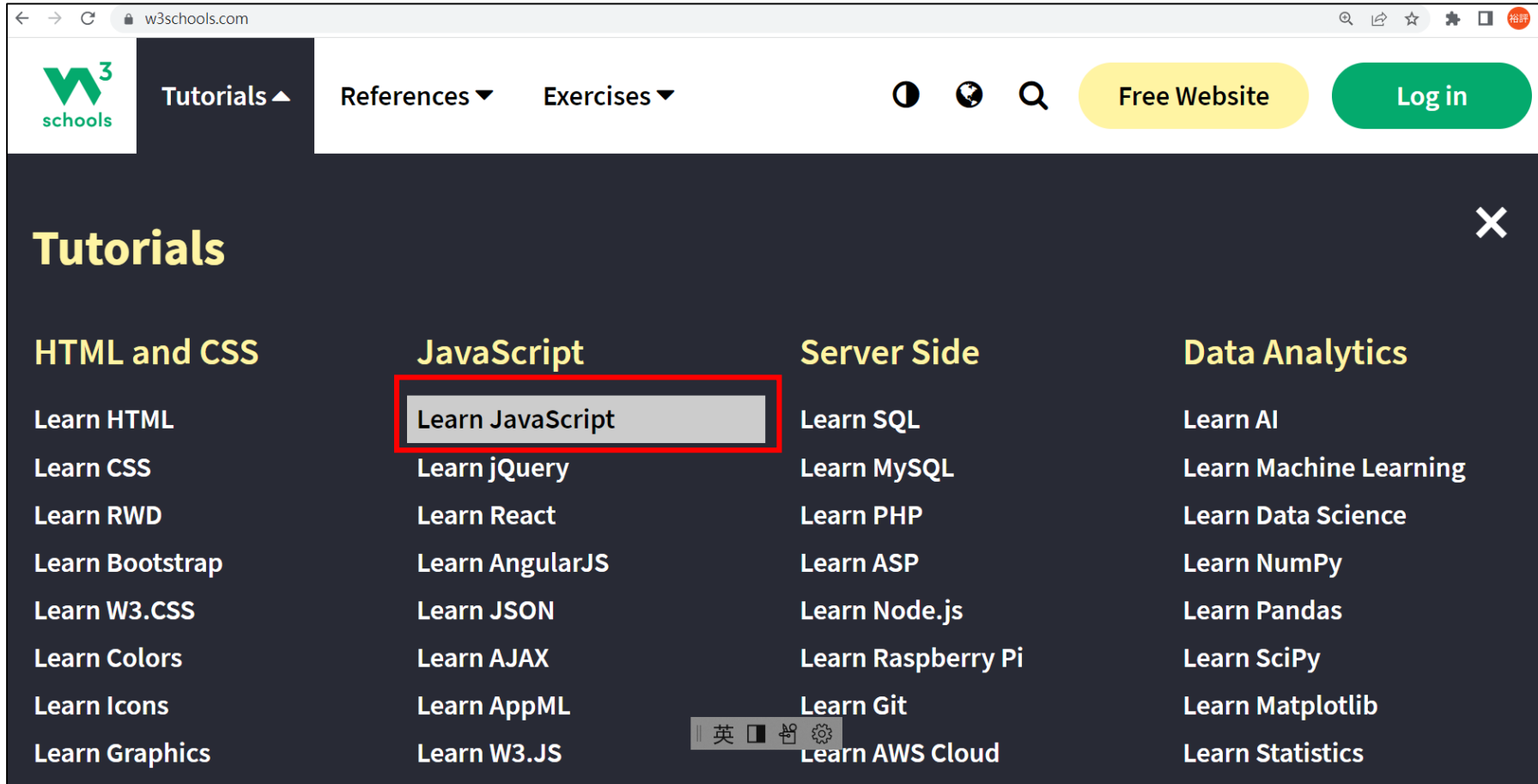
<button onclick="myFunction()">Click Me!
</button>

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

</body>
</html>
```

The live preview on the right shows a button labeled "Click Me!" and a paragraph with the text "Click the button to change the layout of this paragraph". The button is highlighted with a red border, and a blue arrow points from it to a larger, red-styled version of the paragraph in a separate box, illustrating the effect of the JavaScript function.

<https://www.w3schools.com/>



The screenshot shows the w3schools.com website with the 'Tutorials' dropdown menu open. The 'Learn JavaScript' option is highlighted with a red box. The website's header includes the w3schools logo, navigation links for Tutorials, References, and Exercises, and buttons for 'Free Website' and 'Log in'. The 'Tutorials' dropdown menu is organized into four columns: HTML and CSS, JavaScript, Server Side, and Data Analytics. Each column lists various topics to learn, with 'Learn JavaScript' being the first item in the JavaScript column.

Tutorials

- HTML and CSS**
 - Learn HTML
 - Learn CSS
 - Learn RWD
 - Learn Bootstrap
 - Learn W3.CSS
 - Learn Colors
 - Learn Icons
 - Learn Graphics
- JavaScript**
 - Learn JavaScript
 - Learn jQuery
 - Learn React
 - Learn AngularJS
 - Learn JSON
 - Learn AJAX
 - Learn AppML
 - Learn W3.JS
- Server Side**
 - Learn SQL
 - Learn MySQL
 - Learn PHP
 - Learn ASP
 - Learn Node.js
 - Learn Raspberry Pi
 - Learn Git
 - Learn AWS Cloud
- Data Analytics**
 - Learn AI
 - Learn Machine Learning
 - Learn Data Science
 - Learn NumPy
 - Learn Pandas
 - Learn SciPy
 - Learn Matplotlib
 - Learn Statistics

Try it Yourself

Examples in Each Chapter

With our "Try it Yourself" editor, you can edit the

Example

My First JavaScript

Click me to display Date and Time

Try it Yourself »

The screenshot shows the W3Schools 'Try it Yourself' editor interface. A blue arrow points from the 'Try it Yourself »' button in the left panel to the editor. The editor has a toolbar with icons for home, menu, save, print, and a green 'Run' button. A yellow callout bubble labeled '2.Run' points to the 'Run' button. The code editor contains the following HTML and JavaScript code:

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

<h2>My First JavaScript</h2>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.
</button>

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

</body>
</html>
```

A yellow callout bubble labeled '1.Edit' points to the code editor area. To the right of the code editor is a preview window titled 'My First JavaScript'. It shows the rendered output: a button labeled 'Click me to display Date and Time.' and a paragraph with the date and time. A yellow callout bubble labeled '3.Click' points to the button in the preview. A blue arrow points from the button in the preview to a second, identical preview window below it, which also shows the date and time: 'Sat Sep 24 2022 13:44:13 GMT+0800 (Taipei Standard Time)'.

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.

JSON

- JSON stands for **JavaScript Object Notation**
- 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準文件格式，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}
```


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
<i>string</i>	Required. A string written in JSON format
<i>reviver function</i>	Optional. A function used to transform the result. The function is called for each item. Any nested objects are transformed before the parent. <ul style="list-style-type: none">• 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;
 }
});

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"];
```

Try it Yourself

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

Home Menu Save Print Dark Mode Run > Result Size: 184 x 328 Get your website

```
<!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"}'
const obj = JSON.parse(txt);
document.getElementById("demo").innerHTML = obj.name + ", " + obj.age;
</script>







</body>
</html>
```

Creating an Object from a JSON String

John, 30

John 30

Exercise 3-1



Result Size: 454 x 448 [Get your own website](#)

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

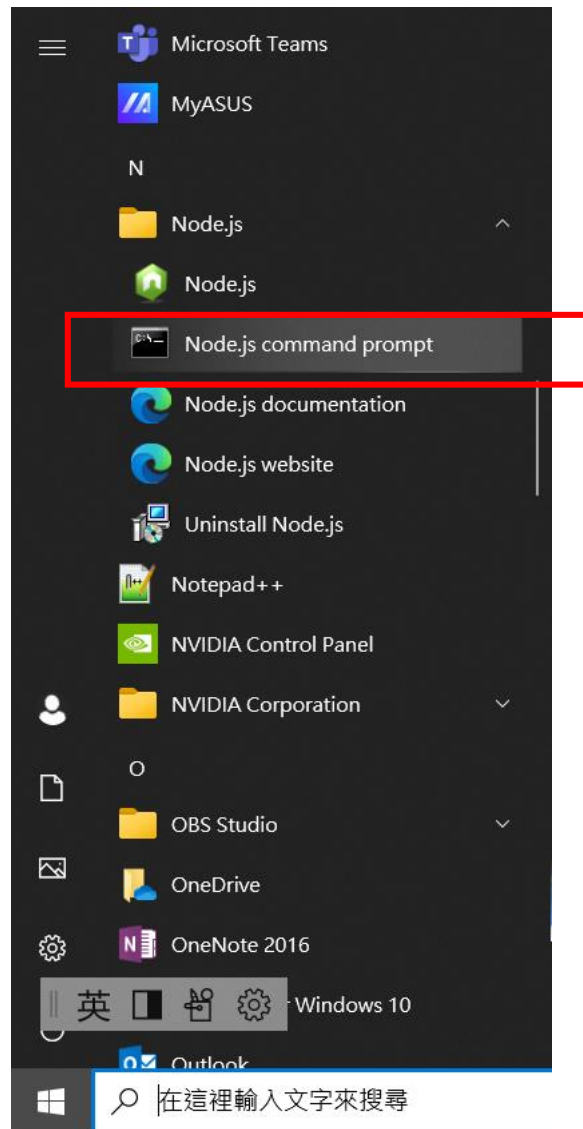
JavaScript Get Date Methods

- 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”



A screenshot of a terminal window titled 'node-red'. A yellow speech bubble with the text 'node-red' points to the terminal title bar. The terminal displays the following text:

```
C:\Users\88697>node-red
27 Sep 23:11:15 - [info] Your environment has been set up for using Node.js 10.16.3 (x64) and npm.
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-TK IK IDEP.json
27 Sep 23:11:21 - [info] Server now running at http://127.0.0.1:1880/
27 Sep 23:11:21 - [warn]
```

The command 'node-red' is entered at the prompt and is highlighted with a red rectangular box.

127.0.0.1:1880

Manage palette

The screenshot displays the Node-RED web interface in a browser. The address bar shows the URL `127.0.0.1:1880/#flow/d95717a5.3a4738`. The interface includes a left sidebar with a node palette containing 'common' and 'function' categories. The main workspace is a grid labeled 'Flow 1'. On the right, a 'Manage' menu is open, showing options like 'View', 'Import', 'Export', 'Search flows', 'Configuration nodes', 'Flows', 'Subflows', 'Settings', 'Keyboard shortcuts', and 'Node-RED website v1.0.4'. The 'Manage palette' option is highlighted. A red box highlights the menu icon in the top right corner, and another red box highlights the 'Manage palette' option in the menu. A yellow speech bubble points to the address bar with the text '127.0.0.1:1880', and another yellow speech bubble points to the 'Manage palette' option with the text 'Manage palette'.

Node-RED

filter nodes

Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link out
- comment

function

- function
- switch
- change
- range
- template

Manage palette

View

- Import
- Export
- Search flows
- Configuration nodes
- Flows
- Subflows
- Manage palette
- Settings
- Keyboard shortcuts
- Node-RED website v1.0.4

Switch flow tabs with `ctrl-↑j` and `ctrl-↑k`

Install

The screenshot shows the 'User Settings' dialog box in Node-RED, specifically the 'Install' tab. The interface includes a 'View' sidebar with 'Nodes' and 'Install' tabs, a search bar, and a list of available nodes. The 'node-red-dashboard' node is highlighted with a red box. A yellow callout bubble points to the node name, and another points to its 'install' button.

User Settings Close

View: **Nodes** | **Install**

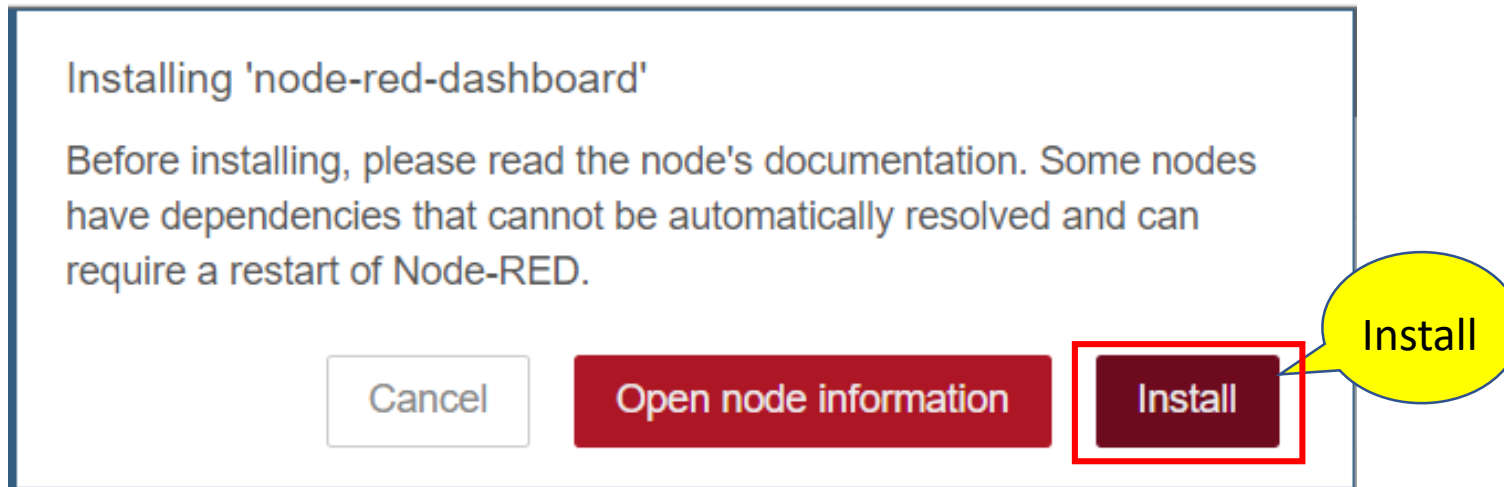
Keyboard: sort: a-z recent

Search: dashboard 71 / 4041

Palette

- node-red-dashboard**
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

Install



Close



User Settings

View

Nodes

Install

Keyboard

sort:  a-z recent 

Palette

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

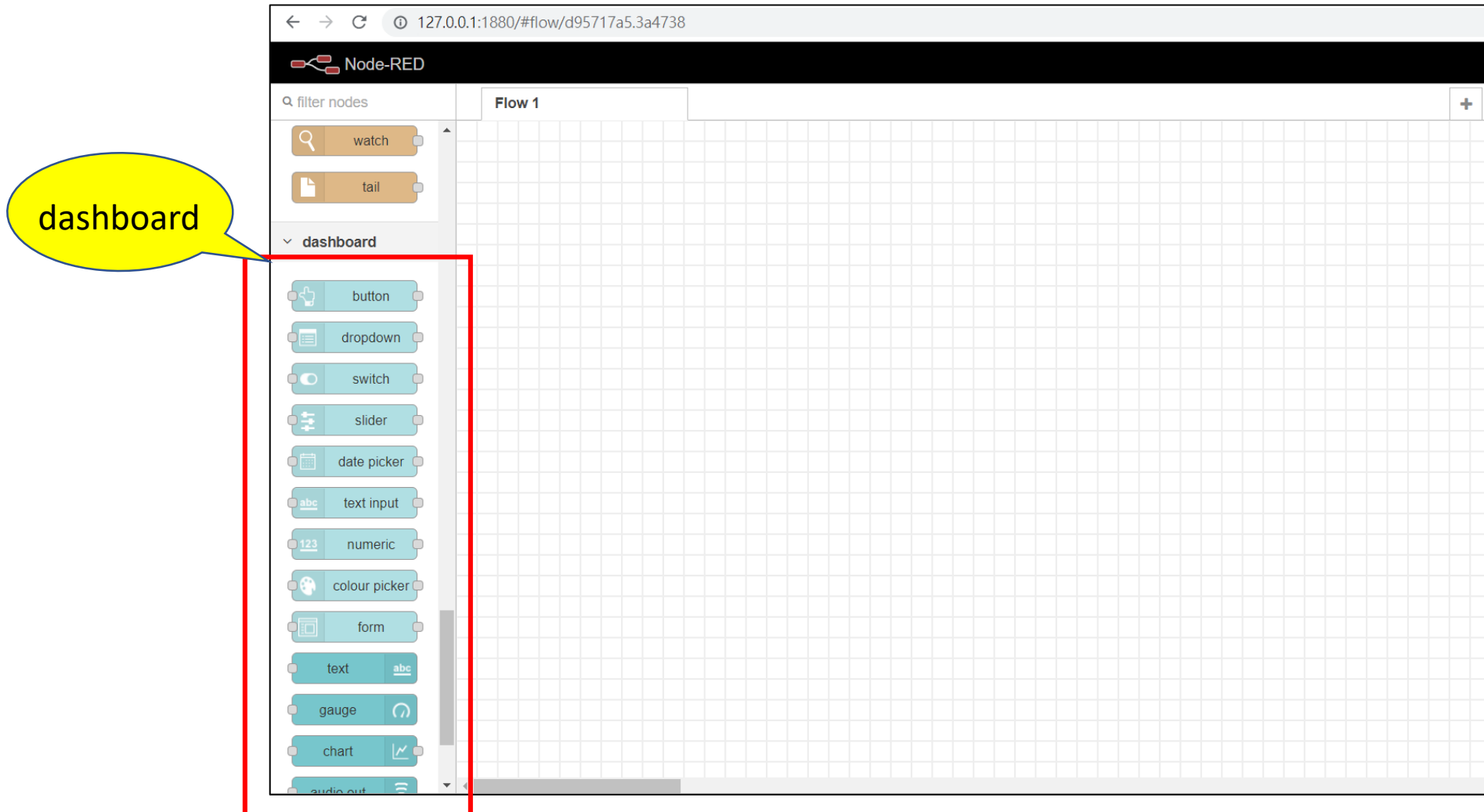
node-red-contrib-dashboard-average-bars

Calculate and display the average values of msg.payload in a bar chart.

0.0.6 4 years, 3 months ago

install

dashboard



Exercise 3-4

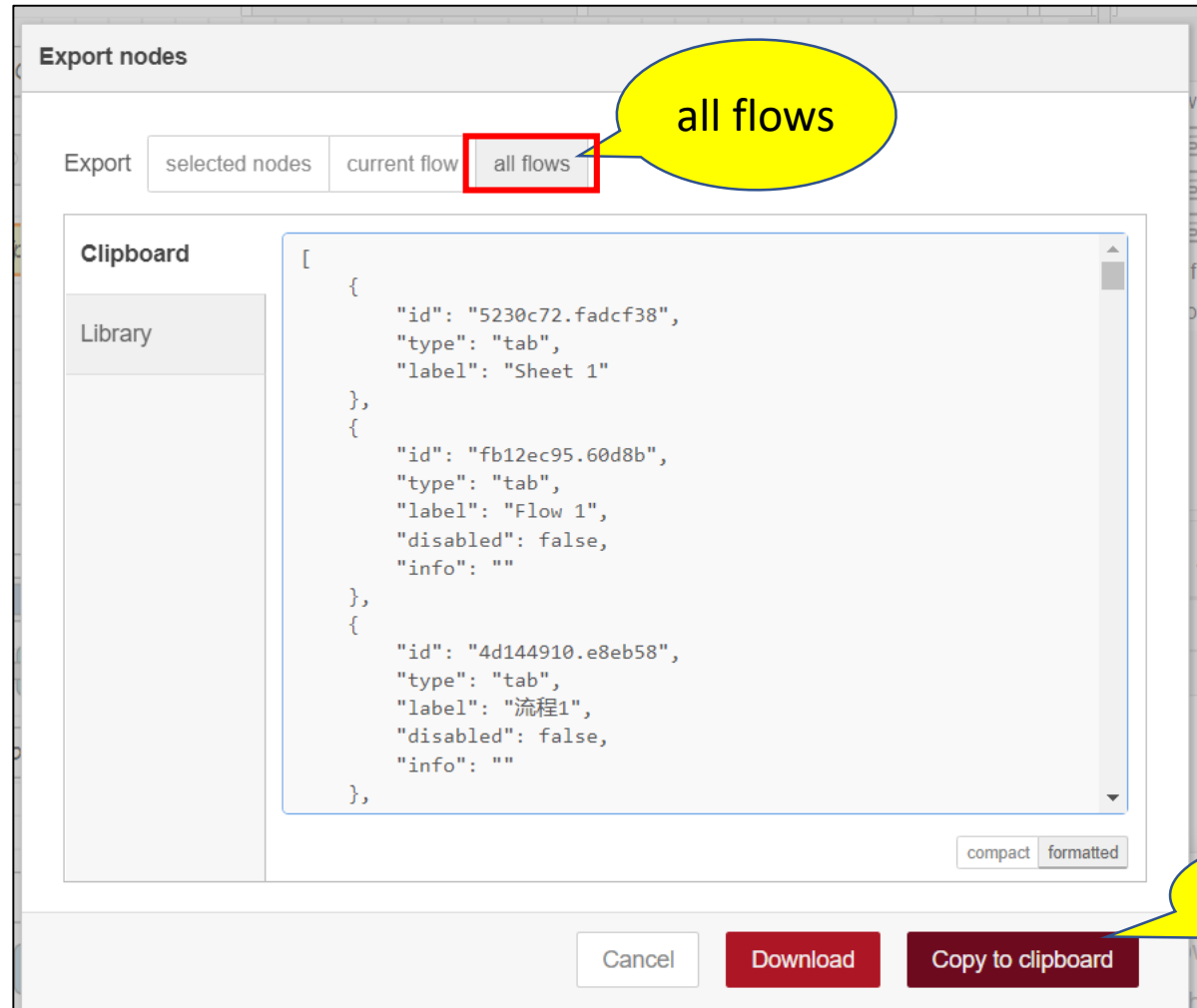
- Copy flows in fred to local node-red.

Export

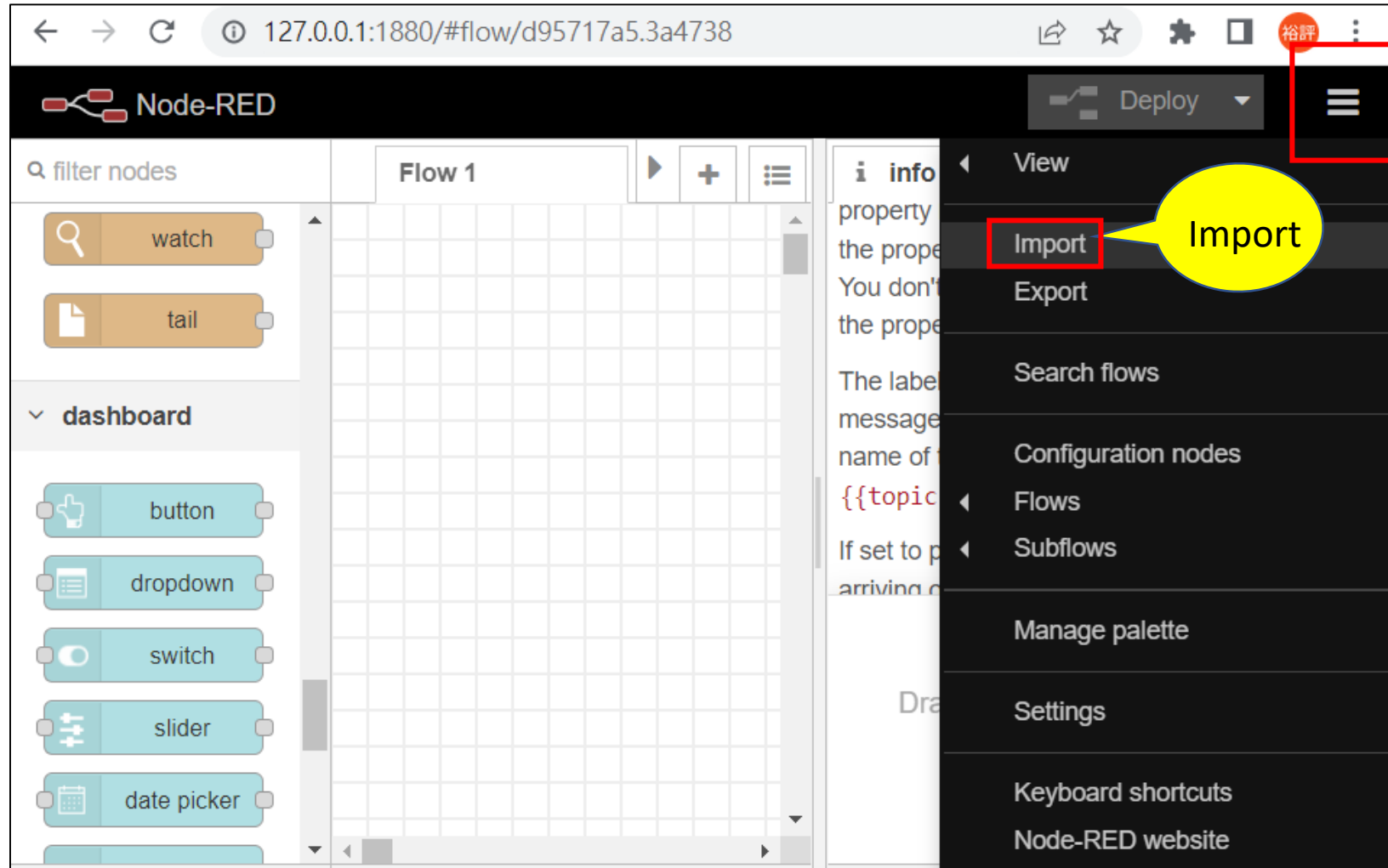
<https://fred.sensetecnic.com/>

The screenshot displays the FRED - Cloud Hosted Node-RED interface. The top bar includes the FRED logo, the text "FRED - Cloud Hosted Node-RED", a "Deploy" button, and a hamburger menu icon (three horizontal lines) which is highlighted with a red box. Below the top bar, there are tabs for "Sheet 1", "Flow 1", and "流程1". The main workspace shows a flow diagram with nodes like "GET /public/finance", "Get market data (Google)", "save original headers", "Parse indexes", and "Generate random number". On the left, there is a "filter nodes" search bar and a list of common and function nodes. On the right, a sidebar menu is open, showing options like "View", "Import", "Export", "Search flows", "Configuration nodes", "Flows", "Subflows", "Groups", "Settings", "Keyboard shortcuts", "Node-RED website", and "v1.2.7". The "Export" option is highlighted with a red box and a yellow callout bubble with the word "Export" inside. The "Node" section of the sidebar shows a "http in" node.

Copy to clipboard



127.0.0.1:1880



Paste



Deploy

The screenshot shows the Node-RED web interface. The top bar has a "Node-RED" logo and a "Deploy" button highlighted with a red rectangle. Below the top bar, there are tabs for "Flow 1", "Sheet 1", "Flow 1", and "流程1". The main workspace displays three flows:

- Flow 1:** A sequence of nodes starting with a "CLICK ME" button, followed by a "Get market data (Google)" node, a "Format data" node, a "Parse indexes" node, and a "Return indexes" node. It also includes a "save original headers" node and a "Try this by visiting" message box.
- Flow 2:** A sequence of nodes starting with a "Fire every 5 seconds" node, followed by a "Generate random number" node, a "chart" node, a "Gauge" node, and a "Data output" node. It also includes a "Use Slider to input values" node and a "You can see this dashboard at" message box.
- Flow 3:** A sequence of nodes starting with a "Tweets about sports teams" node, followed by a "msg.payload" node, and a "Twitter feed" node. It also includes a "Flow 3. Capture & display tweets (needs to be configured)" message box.

The left sidebar shows a "filter nodes" search bar and a list of nodes categorized by "storage", "social", and "dashboard". The right sidebar shows an "info" panel with "Information" and "Description" sections. The "Information" section displays the flow ID "c93f8680.21d248", the name "Sheet 1", and the status "Enabled". The "Description" section is empty. At the bottom right, there is a search bar with the text "Search for nodes using ctrl-f".

Confirm deploy

The workspace contains some nodes that are not properly configured:

- [Sheet 1] Tweets about sports teams (twitter in)

Are you sure you want to deploy?

Cancel

Confirm deploy

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: abc

template

You can see this dashboard at: https://{your-username}.fred.sensetecnic.com/api/ui/

Flow 3. Capture & display tweets (needs to be configured)

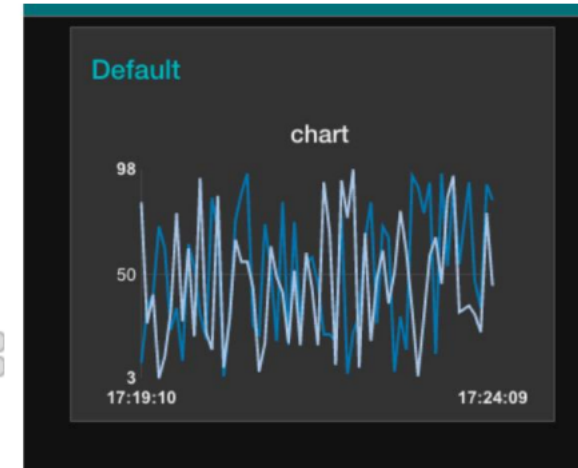
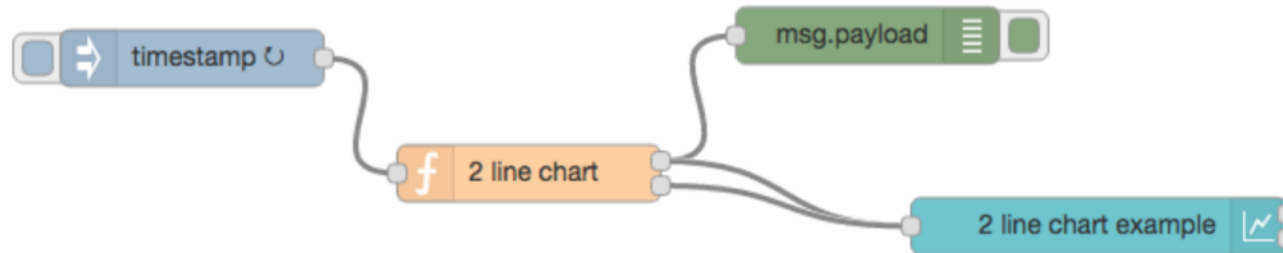
Tweets about sports teams

msg.payload

Twitter feed: abc

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

- The 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.
-



Web Client



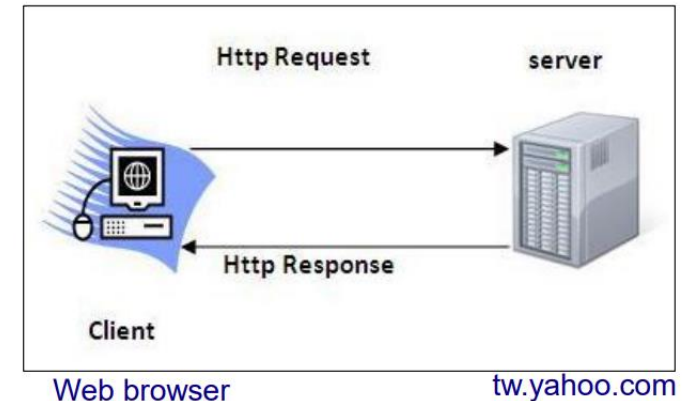
Cloud



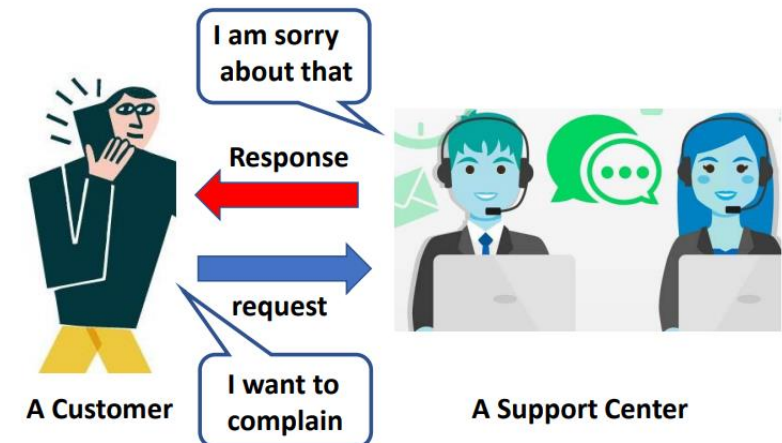
Web Server

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

Firestore Realtime Database REST: Saving Data

- **Saving Data**

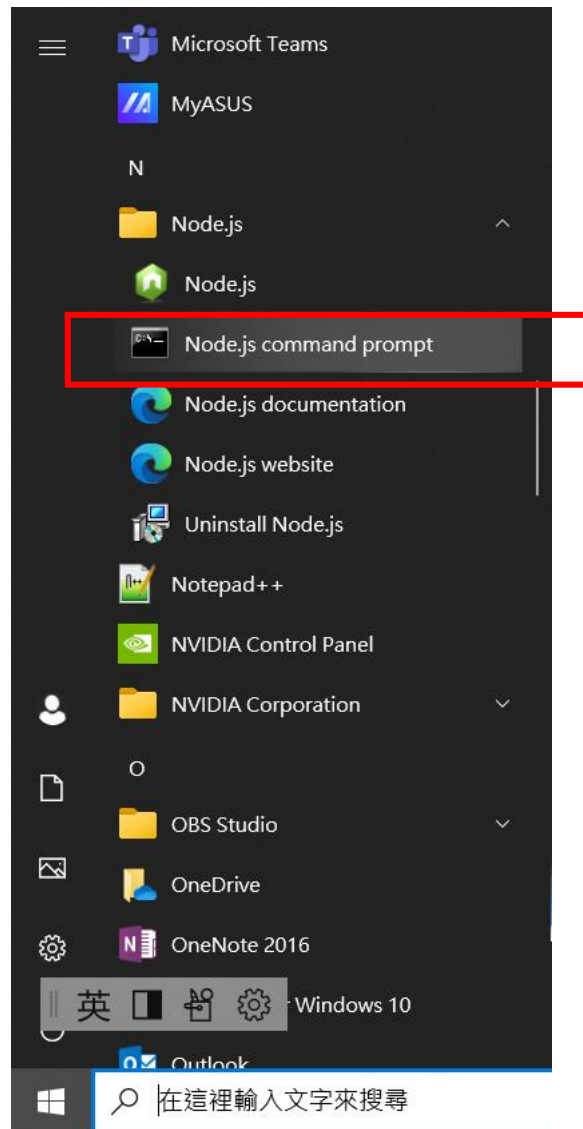
Ways to Save Data	
PUT	Write or replace data to a defined path , like <code>fireblog/users/user1/<data></code>
PATCH	Update some of the keys for a defined path without replacing all of the data.
POST	Add to a list of data in our Firestore database. Every time we send a POST request, the Firestore client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

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

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'
```

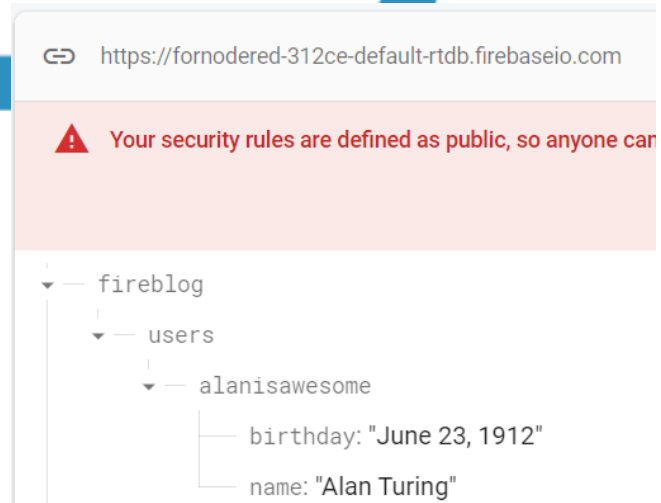
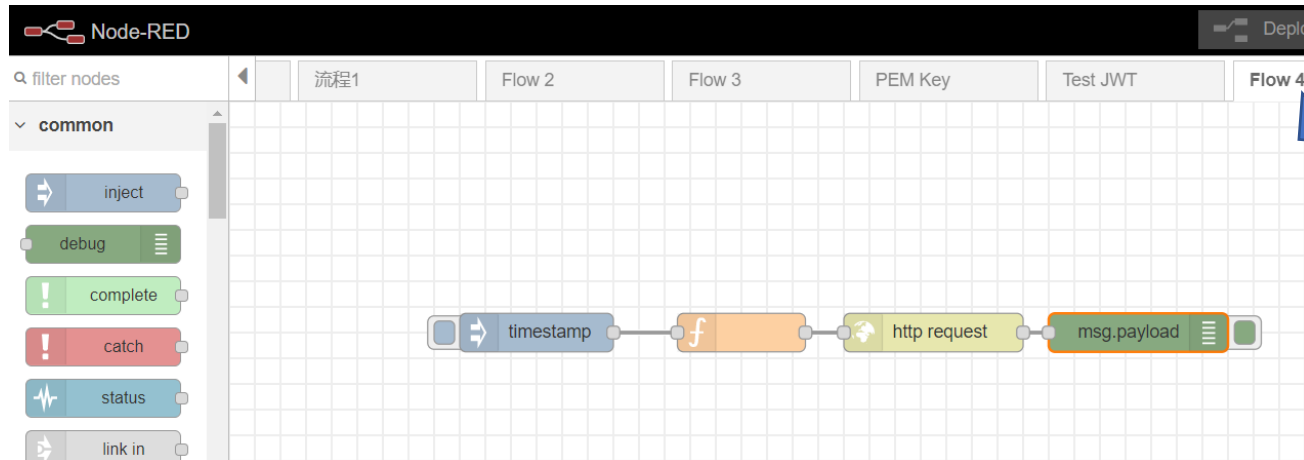
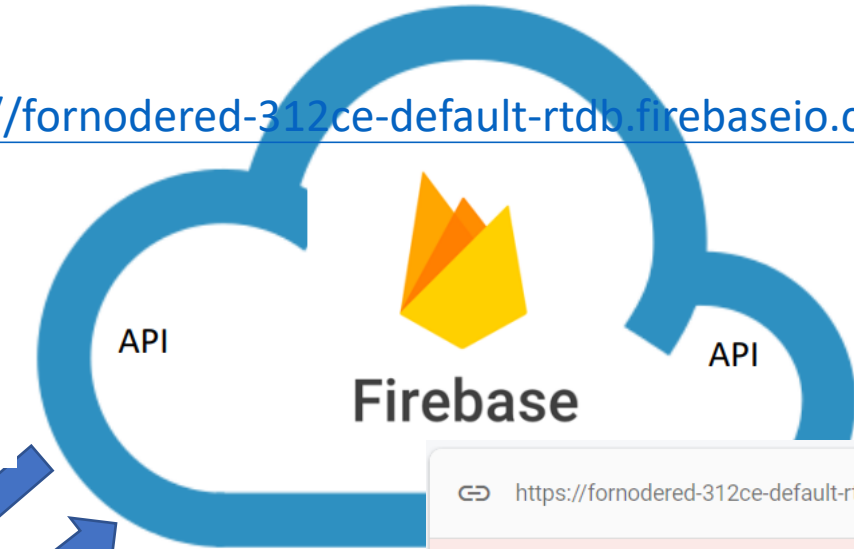


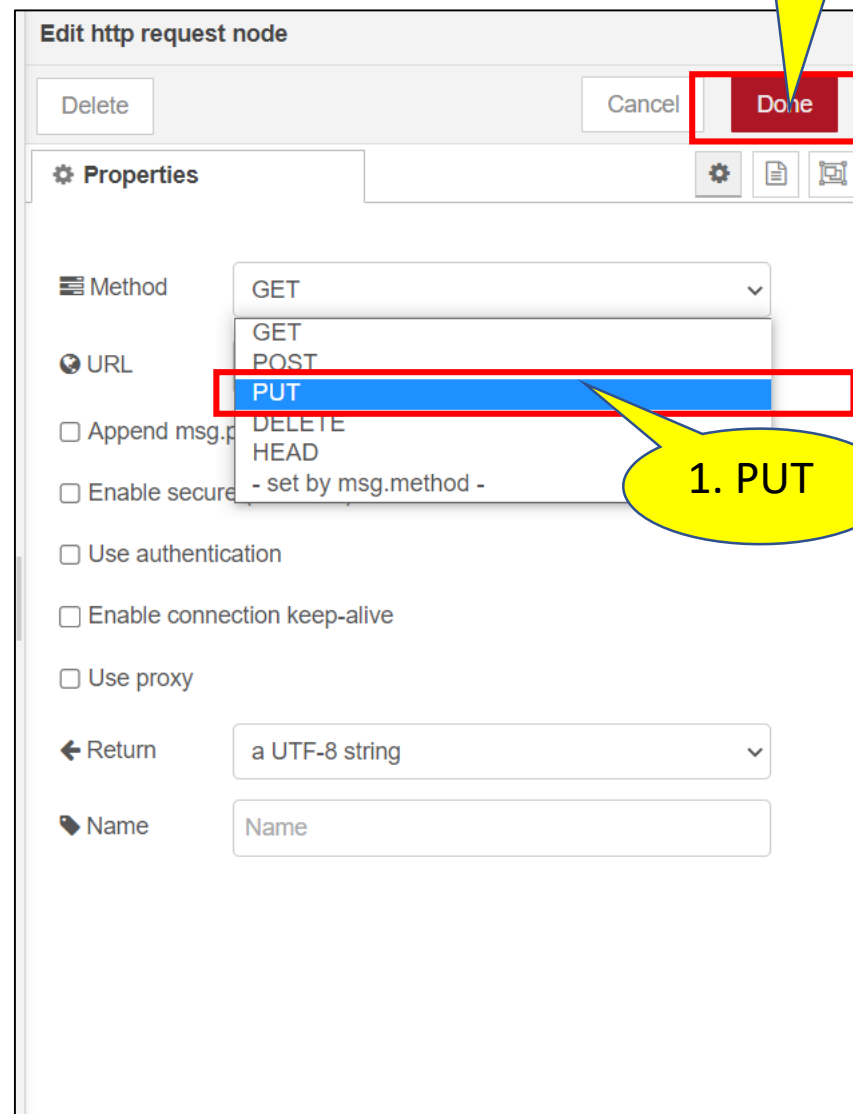
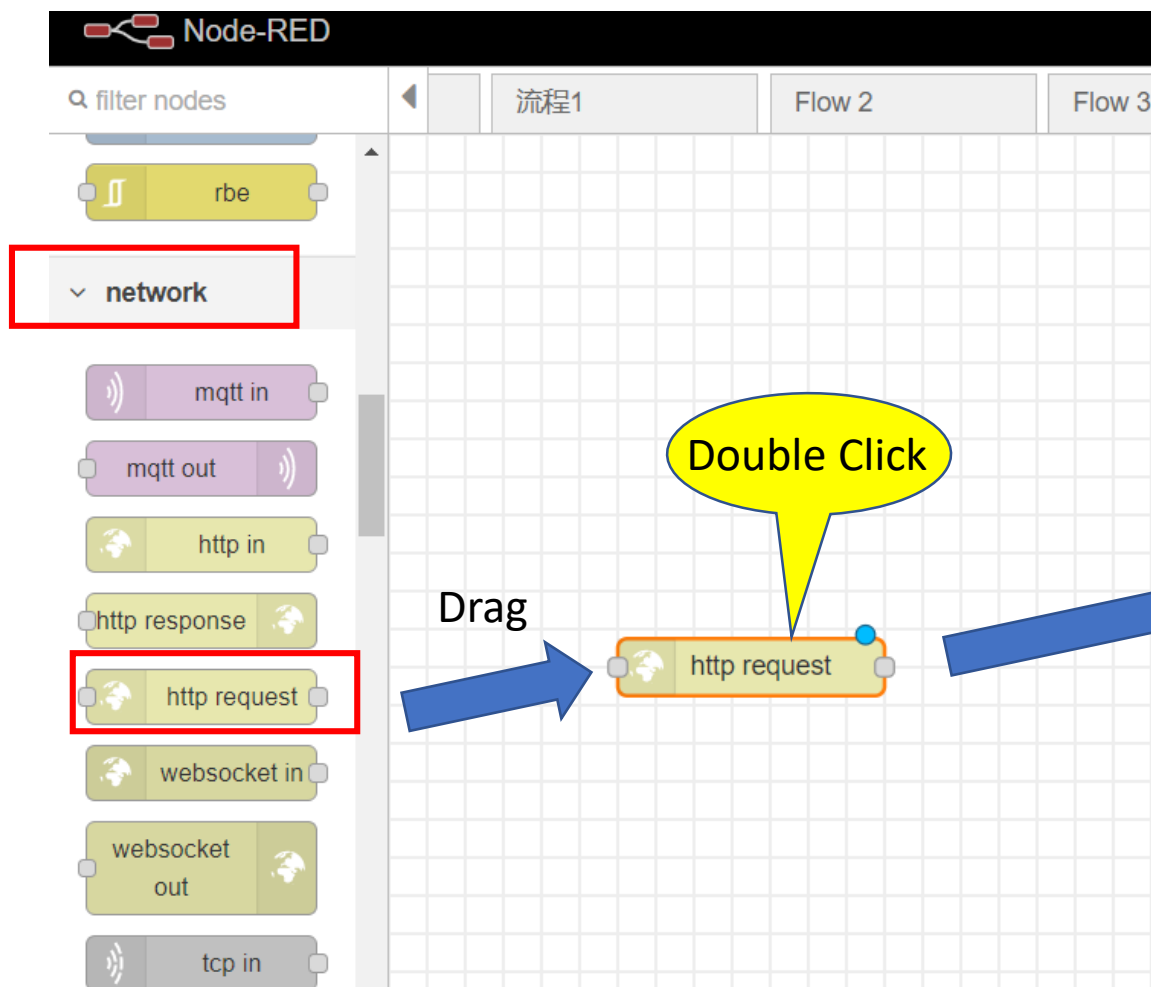
A screenshot of a terminal window titled 'node-red'. A yellow speech bubble with the text 'node-red' points to the terminal title bar. The terminal displays the following text:

```
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-TK IK IDEP.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>





http request node

The screenshot displays the Node-RED web interface. On the left, the 'network' category is expanded in the node palette, showing various nodes including 'http request'. In the center workspace, an 'http request' node is placed on a grid. On the right, the configuration panel for the 'http request' node is open. It features a description, an 'Inputs' section, and a list of properties. Two yellow callout bubbles are present: one labeled 'url' pointing to the 'url' property in the Inputs section, and another labeled 'payload' pointing to the 'payload' property in the main list. Both the 'url' and 'payload' property names are highlighted with red rectangular boxes. The interface also shows a 'Deploy' button in the top right and a search bar for nodes on the left.

Node-RED

filter nodes

流程1

network

mqtt in

mqtt out

http in

http response

http request

websocket in

websocket out

tcp in

tcp out

http request

url

Send HTTP requests and returns the response.

Inputs

url string

If not configured in the node, this optional property sets the url of the request.

method string

If not configured in the node, this optional property sets the HTTP method of the request. Must be one of GET, PUT, POST, PATCH or DELETE.

headers object

Sets the HTTP headers of the request.

cookies object

If set, can be used with the request.

payload

Sent as the body of the request.

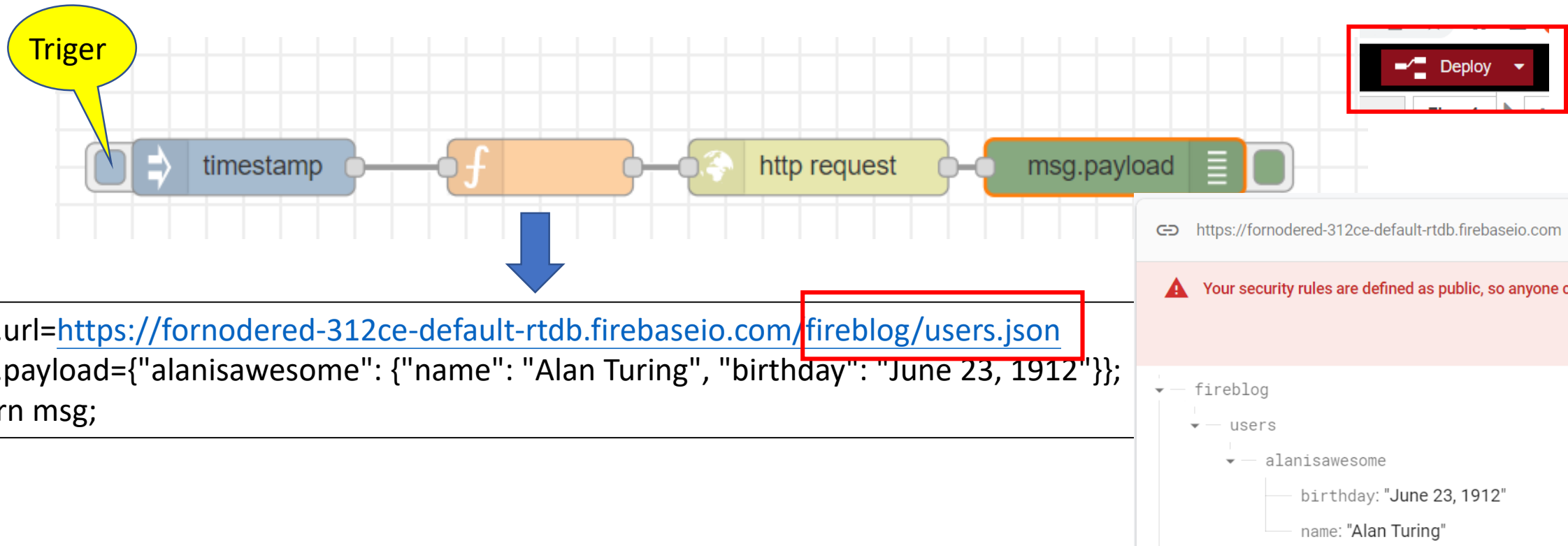
rejectUnauthorized

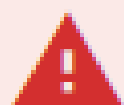
If set to false, allows requests to be made to https sites that use self signed certificates.

followRedirects

If set to false prevent following Redirect (HTTP 301). true by default

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





Your security rules are defined as public, so anyone can

▼ — fireblog

▼ — users

▼ — alanisawesome

— birthday: "June 23, 1912"

— name: "Alan Turing"

Change -> Deploy -> Trigger

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



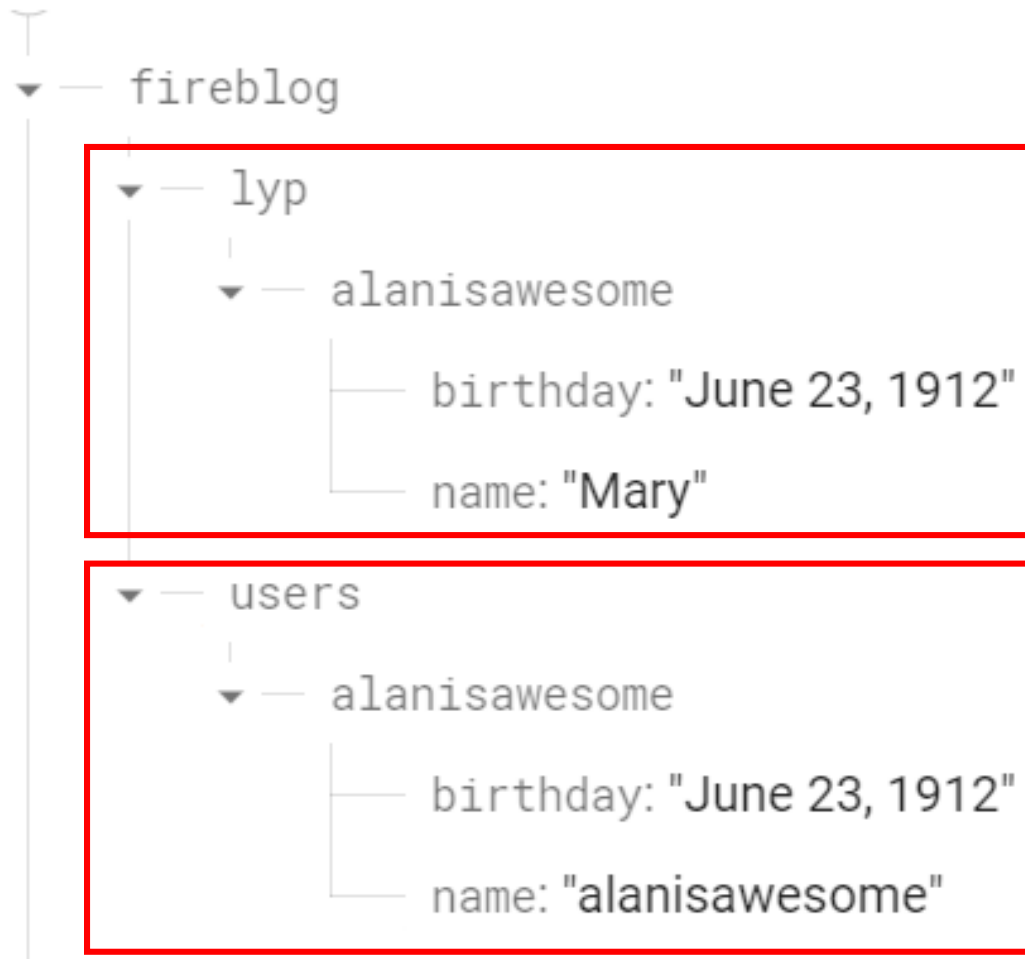
Student ID



Name



Your security rules are defined as public, so anyone can steal, modify, or delete data in your database



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

