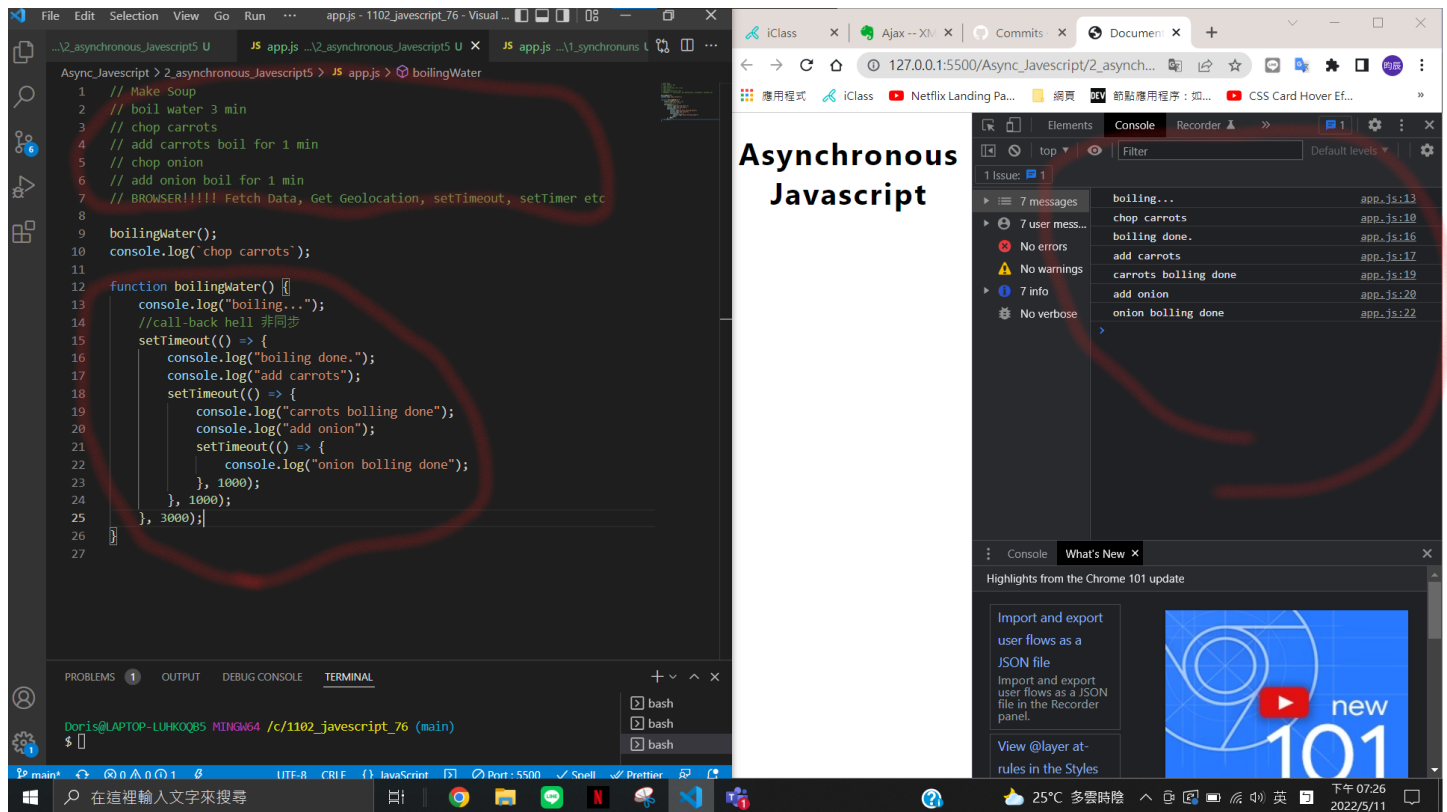


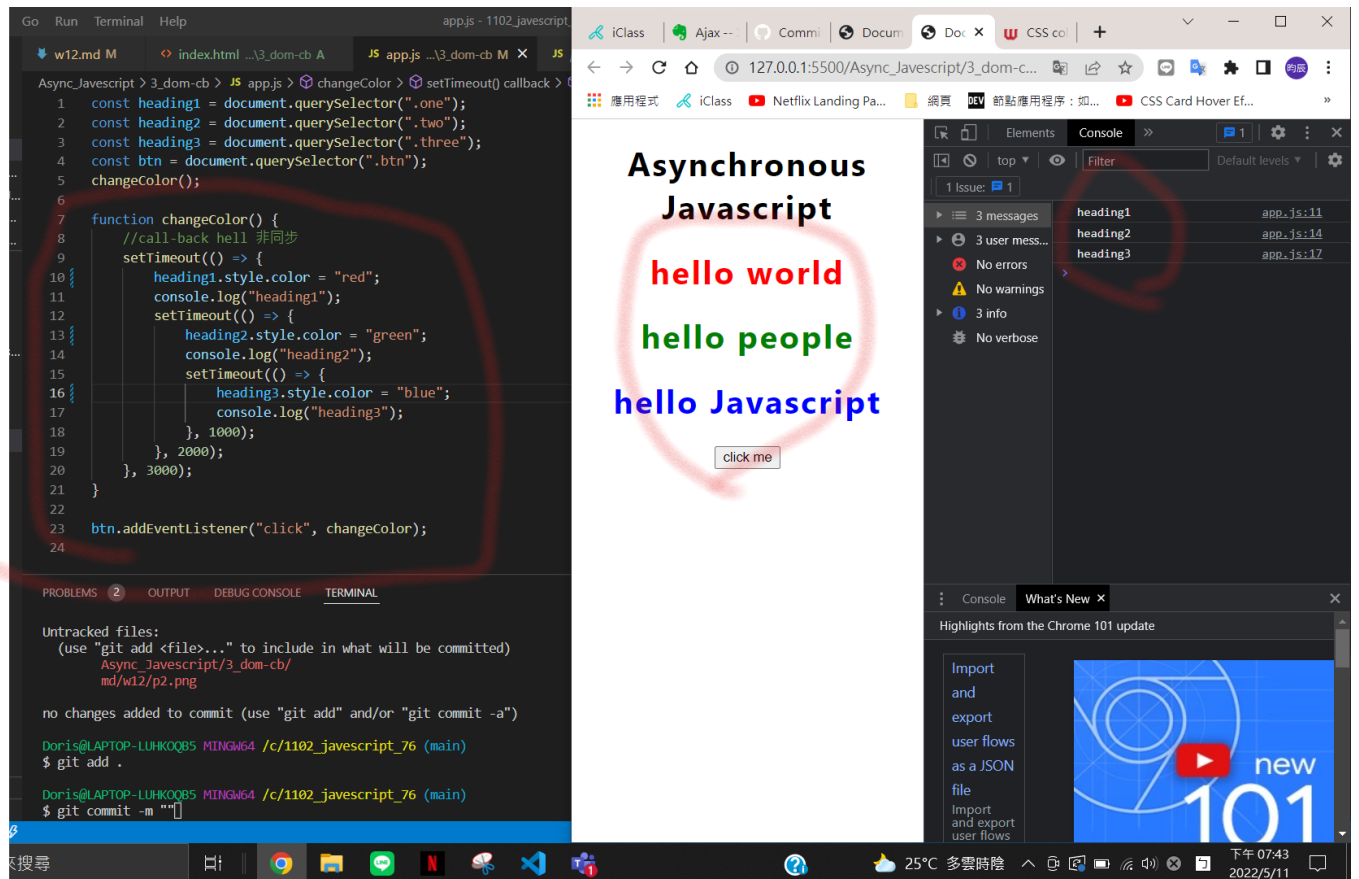
Github-repo url

https://github.com/chen945/1102_javascript_76

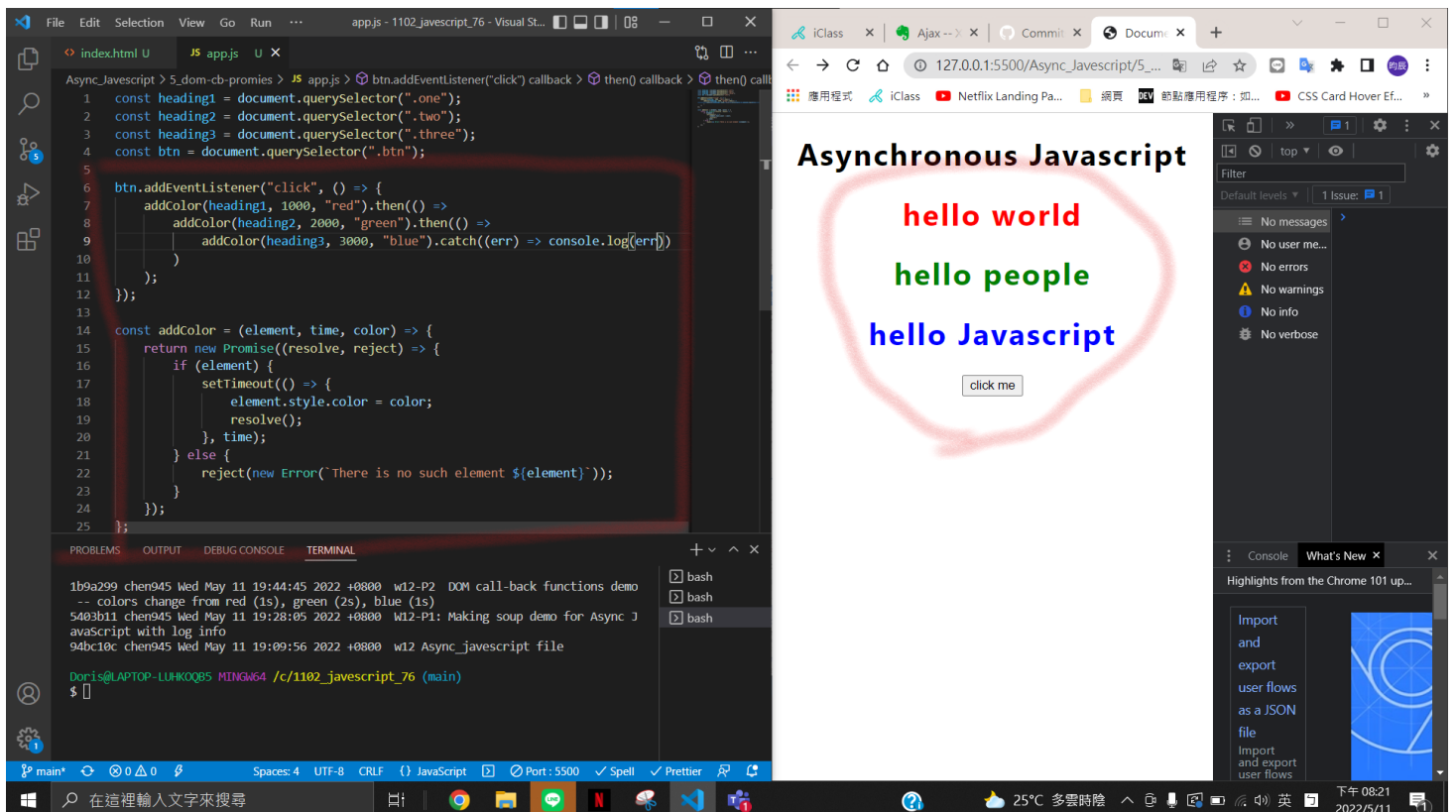
W12-P1: Making soup demo for Async JavaScript with log info



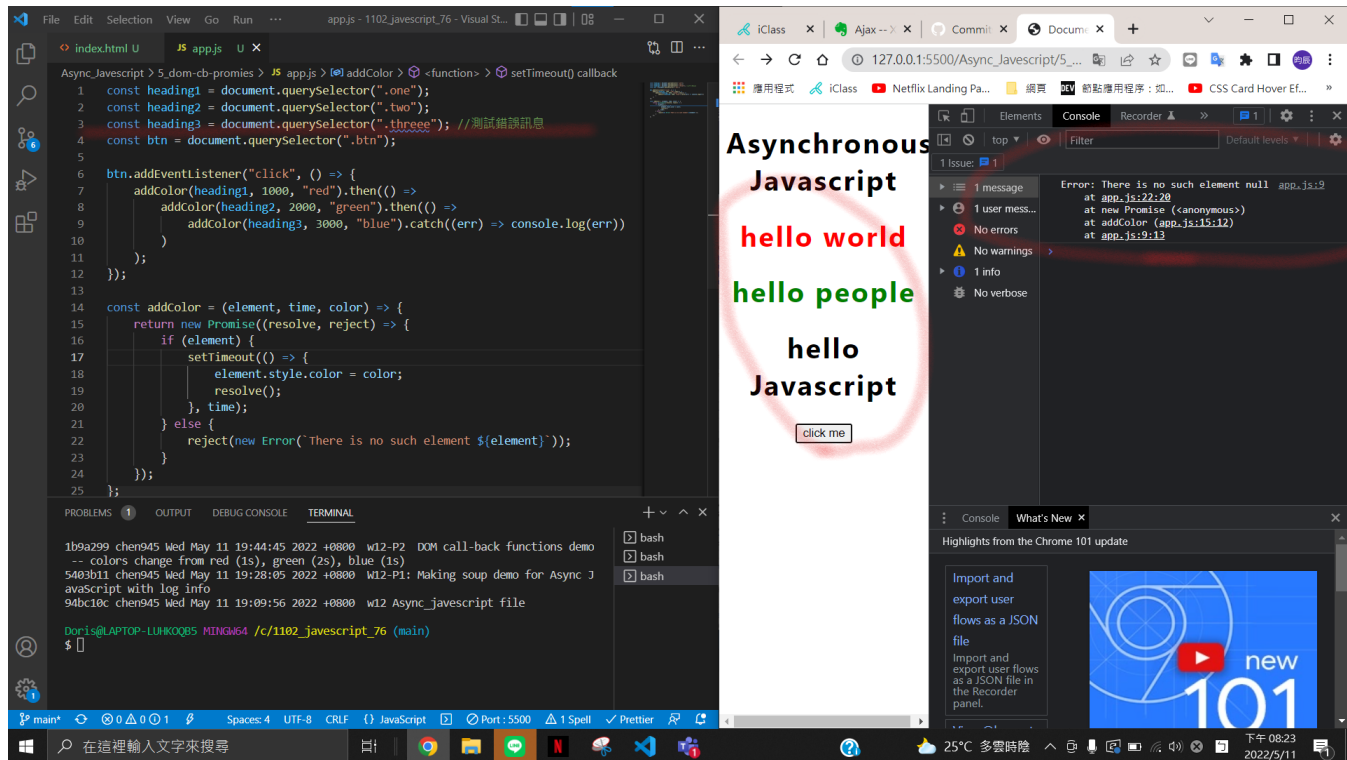
w12-P2 DOM call-back functions demo -- colors change from red (1s), green (2s), blue (1s)



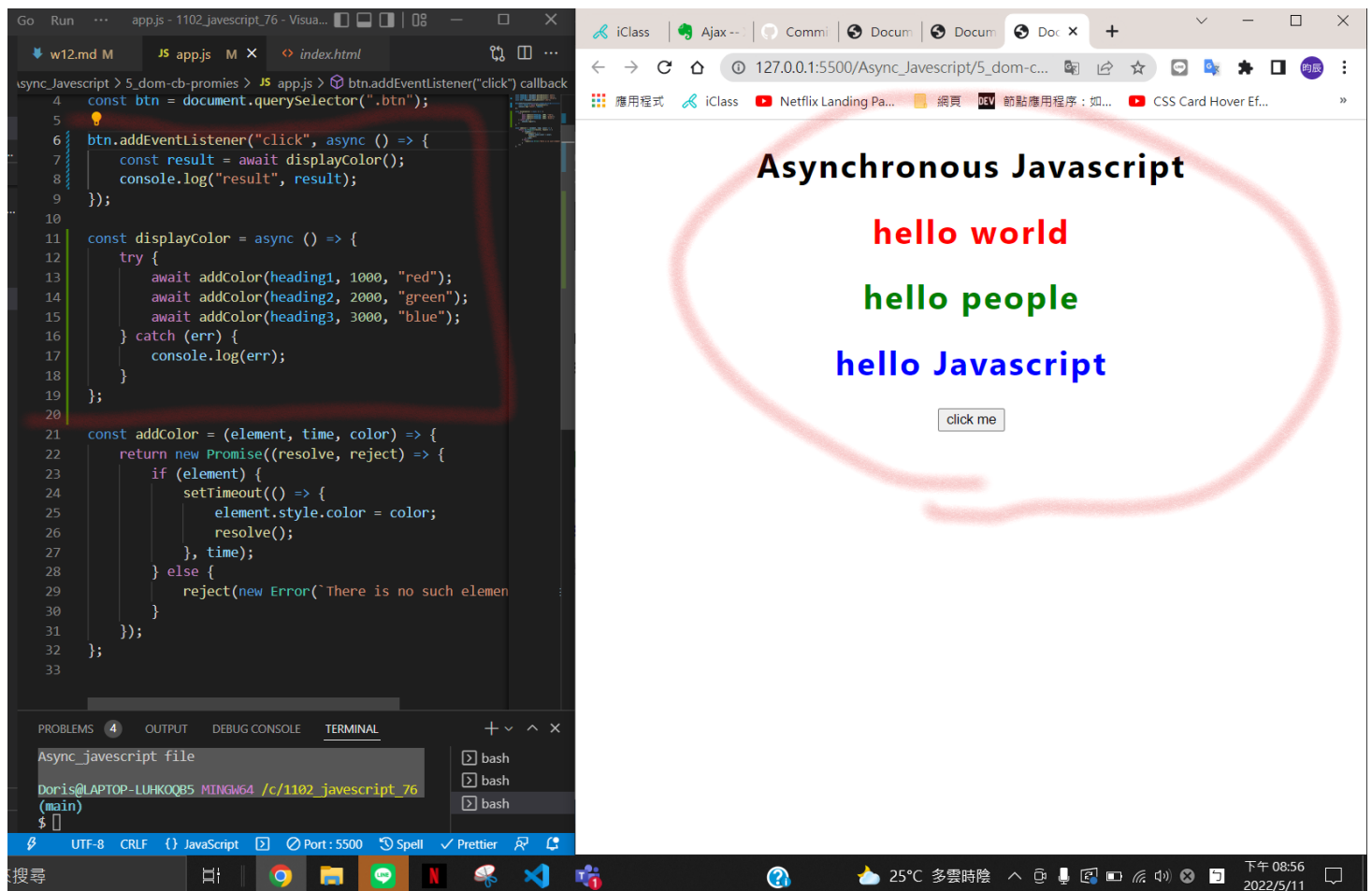
w12-P3 use promise addColor(element,time,color) to solve-call back hell in w12-P2



---測試錯誤訊息



w12-P4 use async /await for w12-P3



---測試錯誤訊息

The image shows a web browser window displaying a page titled "Asynchronous Javascript" with the text "hello world", "hello people", and "hello Javascript". A button labeled "click me" is visible. The browser's developer console is open, showing an error message: "Error: There is no such element null app.js:17". The error stack trace includes the following frames:

- at app.js:29:20
- at new Promise ()
- at addColor (app.js:22:12)
- at displayColor (app.js:15:15)
- at async HTMLButtonElement.<anonymous> (app.js:7:20)

The console also shows the result "result undefined" at app.js:8. The left side of the image shows the source code of the application, which includes the following JavaScript code:

```
1 const heading1 = document.querySelector(".one");
2 const heading2 = document.querySelector(".two");
3 const heading3 = document.querySelector(".three");
4 const btn = document.querySelector(".btn");
5
6 btn.addEventListener("click", async () => {
7   const result = await displayColor();
8   console.log("result", result);
9 });
10
11 const displayColor = async () => {
12   try {
13     await addColor(heading1, 1000, "red");
14     await addColor(heading2, 2000, "green");
15     await addColor(heading3, 3000, "blue");
16   } catch (err) {
17     console.log(err);
18   }
19 };
20
21 const addColor = (element, time, color) => {
22   return new Promise((resolve, reject) => {
23     if (element) {
24       setTimeout(() => {
25         element.style.color = color;
26         resolve();
27       }, time);
28     } else {
29       reject(new Error("There is no such element"));
30     }
31   });
32 }
```

w12-P5 fetch people.json with 4 data shown on console, change name to people2.json will result in error

Asynchronous Javascript

```
const url = "/api/people.json";

fetch(url)
  .then((response) => response.json())
  .then((data) => console.log(data))
  .catch((err) => console.log(err));
```

Console output:

```
(4) [Object, Object, Object, Object]
  0: {name: 'John', id: 1}
  1: {name: 'peter', id: 2}
  2: {name: 'ann', id: 3}
  3: {name: 'susan', id: 4}
  length: 4
  [[Prototype]]: Array(0)
```

---測試錯誤訊息

Asynchronous Javascript

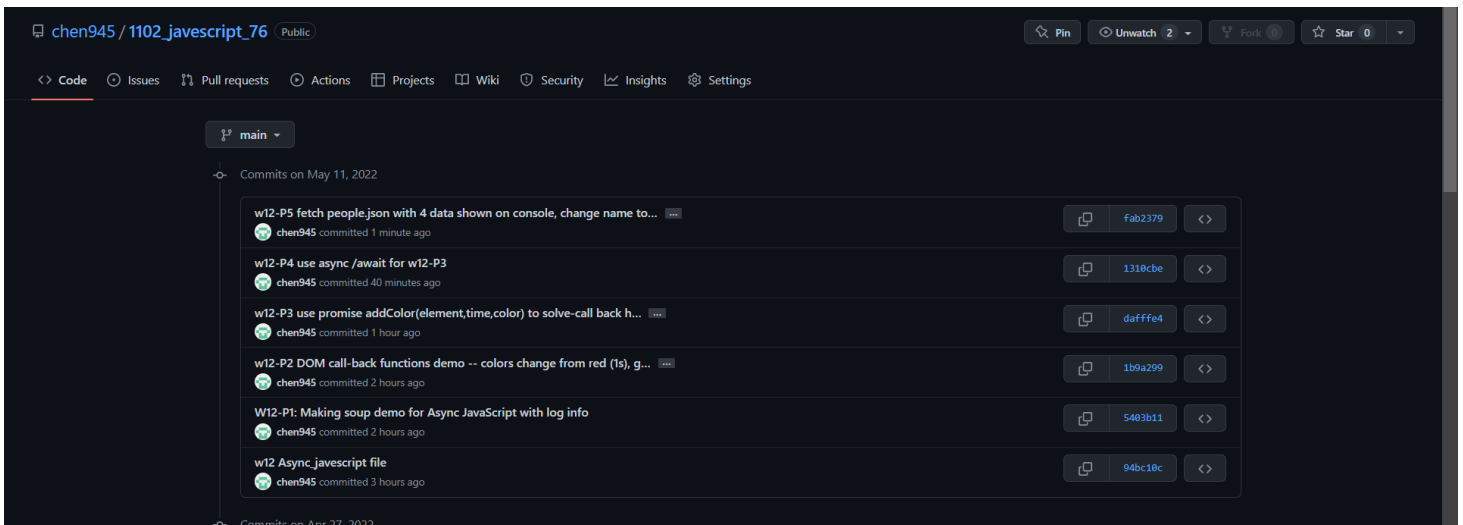
```
const url = "/api/people2.json";

fetch(url)
  .then((response) => response.json())
  .then((data) => console.log(data))
  .catch((err) => console.log("err", err));
```

Console output:

```
err SyntaxError: Unexpected token <
in JSON at position 0
```

log 紀錄



```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-05-10"
fab2379 chen945 Wed May 11 21:42:17 2022 +0800 w12-P5 fetch people.json with 4 data shown on console, change nan
1310cbe chen945 Wed May 11 21:02:56 2022 +0800 w12-P4 use async /await for w12-P3
dafffe4 chen945 Wed May 11 20:28:43 2022 +0800 w12-P3 use promise addColor(element,time,color) to solve-call ba
1b9a299 chen945 Wed May 11 19:44:45 2022 +0800 w12-P2 DOM call-back functions demo -- colors change from red (1
5403b11 chen945 Wed May 11 19:28:05 2022 +0800 W12-P1: Making soup demo for Async JavaScript with log info
94bc10c chen945 Wed May 11 19:09:56 2022 +0800 w12 Async_javascript file
```