

w12-p1: Making soup demo for Async JavaScript with log info

The screenshot shows a Mac desktop with two browser windows and a terminal window.

Left Browser Window: Displays a code editor with a file named `app.js`. The code contains a recursive function `boilingWater()` that logs various steps of a cooking process to the console. A red box highlights the first few lines of the code.

```
//Make Soup
//boil water 3 min
//chop carrots
//add carrots boil for 2 min
//chop onion
//add onion boil for 2 min
//BROWSE!!!! Fetch Data, Get Geolocation, setTimeout, setTimer,
boilingWater();
console.log('chop carrots');

function boilingWater() {
    console.log('boiling...');

    setTimeout(() => {
        console.log('boiling done.');
        console.log('add carrots');
        setTimeout(() => {
            console.log('carrots boiling done.');
            console.log('add onion');
            setTimeout(() => {
                console.log('onion boiling done.');
            }, 500);
        }, 2000);
    }, 3000);
}
```

Right Browser Window: Displays a web page titled "Asynchronous Javascript". It features several "click me" buttons. The page's content is as follows:

Asynchronous Javascript
hello world
hello people
hello Javascript

Terminal Window: Shows a GitHub commit message and a command-line history. The commit message is:

```
commit b68b980e520a1d5a97f07bb576c1e376f01996c5 (HEAD -> main, origin/main)
Author: Weiyi HuangFelix <a0908882001@gmail.com>
Date: Thu May 5 19:32:31 2022 +0800

    w11

commit 0176208aa5333077df9f199855660004d8d959c
Author: Weiyi HuangFelix <a0908882001@gmail.com>
Date: Thu May 5 19:38:44 2022 +0800
```

Mac Dock: Shows various application icons including Finder, Mail, Safari, and others.

w12-p2 DOM call-back function demo -- colors change from red(1s), green (2s), blue(1s)

The screenshot shows a Mac desktop with two browser windows and a terminal window.

Left Browser Window: The title bar says "app.js – 1102-javascript-210410626". The content area displays the following JavaScript code:

```
1 //Make Soup
2 //boil water 3 min
3 //chop carrots
4 //add carrots boil for 2 min
5 //chop onion
6 //add onion boil for 2 min
7 //BROWSER!!!! Fetch Data, Get Geolocation, setTimeout, setTimer,
8
9 boilingWater();
10 console.log(`chop carrots`);
11
12 function boilingWater() {
13     console.log('boiling...');
14
15     setTimeout(() => {
16         console.log('boiling done.');
17         console.log('add carrots');
18         setTimeout(() => {
19             console.log('carrots boiling done.');
20             console.log('add onion');
21             setTimeout(() => {
22                 console.log('onion boiling done.');
23             }, 500);
24         }, 2000);
25     }, 3000);
26 }
```

Right Browser Window: The title bar says "assignm – 1102-11 – 上傳面 x". The content area shows the output of the console logs from the script:

Log	File	Line
boiling...	app.js	13
chop carrots	app.js	10
Live reload enabled.		
boiling done.	app.js	16
add carrots	app.js	17
carrots boiling done.	app.js	19
add onion	app.js	20
onion boiling done.	app.js	22

Bottom Terminal: The title bar says "zsh". The command line shows:

```
huangweiyi@huangweiyideMacBook-Pro 1102-javascript-210410626 %
```

The Dock at the bottom of the screen contains icons for various applications including Finder, Mail, Safari, and others.

w12-p3 use promise addColor(attribute, color, time) to solve-call back hell in w12-p2

```

19 //         });
20 //     );
21 //
22 //     btn.addEventListener('click', () => {
23 //         changeColor(helloworldRed, 'red', 1)
24 //         .then(() => {
25 //             changeColor(helloworldGreen, 'green', 2);
26 //         })
27 //         .then(() => {
28 //             changeColor(helloworldBlue, 'blue', 1);
29 //         })
30 //         .catch(err => console.log(err));
31 //     });
32 //
33 btn.addEventListener('click', () => {
34     addColor(heading1, 1000, 'red')
35     .then(() => addColor(heading2, 2000, 'green'))
36     .then(() => addColor(heading3, 1000, 'blue'))
37     .catch((err) => console.log(err));
38 });
39
40 const addColor = (element, time, color) => {
41     return new Promise((resolve, reject) => {
42         if(element) {
43             setTimeout(() => {
44                 element.style.color = color;
45                 resolve();
46             }, time);
47         } else {
48             reject(new Error(`There is no such element ${element}`));
49         }
50     });
51 };

```

```

1 const heading1 = document.querySelector('.one');
2 const heading2 = document.querySelector('.two');
3 const heading3 = document.querySelector('.three');
4
5 const btn = document.querySelector('.btn');
6
7 // const changeColor = function(attribute, color, time) {
8 //     return new Promise(function(resolve, reject) {
9 //         if(attribute) {
10 //             setTimeout(() => {
11 //                 const helloworldBth = attribute;
12 //                 helloworldBth.style.color = `${color}`;
13 //
14 //                 resolve(helloworldBth);
15 //             }, time * 1000);
16 //         } else {
17 //             reject(new Error(`There is no such attribut ${attribute}`));
18 //         }
19 //     });
20 // };
21
22 // btn.addEventListener('click', () => {
23 //     changeColor(helloworldRed, 'red', 1)
24 //     .then(() => {
25 //         changeColor(helloworldGreen, 'green', 2);
26 //     })
27 //     .then(() => {
28 //         changeColor(helloworldBlue, 'blue', 1);
29 //     })
30 //     .catch(err => console.log(err));
31 // });
32
33 btn.addEventListener('click', () => {
34     addColor(heading1, 1000, 'red')

```

w12-p4 use async / await for w12-p3

The screenshot shows a Mac desktop environment. On the left is a code editor window for VS Code, displaying a file named 'app.js'. The code contains an asynchronous function 'displayColor' that changes the color of three heading elements ('.one', '.two', '.three') over time. On the right is a web browser window showing a simple HTML page with three colored headings. The DevTools panel is open, highlighting the CSS rules for each heading's color: red for '.one', green for '.two', and blue for '.three'.

This screenshot is similar to the one above, but the browser page now displays an error message: 'Error: There is no such element null'. This indicates a problem with the DOM selection in the code, likely due to a typo or incorrect selector.

w12-p5 fetch people.json with 4 data shown on console change name to people.json will result in error

```

app.js — 1102-javascript-210410626
JS app.js U X w1_26.md U index.html U people.json ...
demo > w12 > Async_JavaScript > 7_fetch_api > api > JS app.js > ...
1 const url = './api/people.json';
2
3 fetch(url).then((Response) => console.log(Response.json())).then()

```

```

Promise {<pending>} app.js:3
  [[Prototype]]: Promise
  [[PromiseState]]: "fulfilled"
  [[PromiseResult]]: Array(4)
    0: {name: 'john', id: 1}
    1: {name: 'peter', id: 2}
    2: {name: 'anna', id: 3}
    3: {name: 'susan', id: 4}
    length: 4
  [[Prototype]]: Array(0)

```

```

app.js — 1102-javascript-210410626
JS app.js U X w1_26.md U index.html U people.json ...
demo > w12 > Async_JavaScript > 7_fetch_api > api > JS app.js > then() callback
1 .../api/people2.json';
2
3 en((Response) => console.log(Response.json())).then().catch()

```

```

GET http://127.0.0.1:5500/demo/w12/Async_JavaScript/7_fetch_api/api/people2.json 404 (Not Found)
Promise {<pending>} app.js:3
Uncaught (in promise) SyntaxError: Unexpected token < in JSON at position 0

```

LAST log on "2022-05-13"

```
huangweiyi@huangweiyideMacBook-Pro 1102-javascript-210410626 % git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-05-13"
huangweiyi@huangweiyideMacBook-Pro 1102-javascript-210410626 % git log --pretty=format:"%h%x09%an%x09%ad%x09%s"
0513b02 WeiYiHuangfelix Fri May 13 16:41:25 2022 +0800 w12
b68b980 WeiYiHuangfelix Thu May 5 19:32:31 2022 +0800 w11
0176208 WeiYiHuangfelix Thu May 5 19:30:44 2022 +0800 w11
23caa34 WeiYiHuangfelix Wed Apr 27 22:54:42 2022 +0800 w09 midprep p3_26 answer
0471f2b WeiYiHuangfelix Thu Apr 21 10:32:28 2022 +0800 w09 midprep p3_26 answer
2d807e2 WeiYiHuangfelix Thu Apr 21 09:36:18 2022 +0800 w09 midprep p2_26 answer
cc05feb WeiYiHuangfelix Thu Apr 21 08:28:10 2022 +0800 midprep p1_26 answer
80ad4ea WeiYiHuangfelix Thu Apr 21 08:17:59 2022 +0800 w09 midprep starter
e9879d4 WeiYiHuangfelix Thu Apr 14 15:23:29 2022 +0800 W08-P5: Conut wealth total for all users money
5c75084 WeiYiHuangfelix Thu Apr 14 12:34:24 2022 +0800 w08-P2: add w08 md link in RWADME.md
9ed4fc6 WeiYiHuangfelix Thu Apr 14 12:19:06 2022 +0800 Initial commit
4-07165 WeiYiHuangfelix Thu Apr 14 12:09:52 2022 +0800 w08-P1: add P1 link
```

```
huangweiyi@huangweiyideMacBook-Pro 1102-javascript-210410626 % git log --
pretty=format:"%h%x09%an%x09%ad%x09%s"
0513b02 WeiYiHuangfelix Fri May 13 16:41:25 2022 +0800 w12
```