

W12-P1: call back hell DOM demo

The screenshot shows a code editor on the left with the following JavaScript code:

```
1 const heading1 = document.querySelector('.one');
2 const heading2 = document.querySelector('.two');
3 const heading3 = document.querySelector('.three');
4 const heading4 = document.querySelector('.four');
5 const btn = document.querySelector('.btn');
6
7 btn.addEventListener('click', function () {
8   setTimeout(() => {
9     heading1.style.color = 'red';
10    setTimeout(() => {
11      heading2.style.color = 'green';
12      setTimeout(() => {
13        heading3.style.color = 'blue';
14        setTimeout(() => {
15          heading4.style.color = 'brown';
16        }, 500);
17      }, 1000);
18    }, 2000);
19  }, 1000);
20 });
```

The browser on the right displays the output of the code, which is a white box with the text "Asynchronous Javascript" at the top. Below it, the text "hello world" (red), "hello people" (green), "hello Javascript" (blue), and "hello Async JS" (brown) is displayed in a sequence. A red box highlights the text "hello Async JS". A "click me" button is at the bottom.

ec87582 changiojen1 Thu May 4 19:20:02 2023 +0800 P1: call back hell DOM demo

W12-P2: use promise to solve the cb hell problem

The screenshot shows a code editor on the left with the following JavaScript code:

```
1 const heading1 = document.querySelector('.one');
2 const heading2 = document.querySelector('.two');
3 const heading3 = document.querySelector('.three');
4 const heading4 = document.querySelector('.four');
5 const btn = document.querySelector('.btn');
6
7 btn.addEventListener('click', () => {
8   addColor(1000, heading1, 'red')
9     .then(() => addColor(2000, heading2, 'green'))
10    .then(() => addColor(1000, heading3, 'blue'))
11    .then(() => addColor(500, heading4, 'brown'))
12    .catch((err) => console.log(err));
13 });
14
15 const addColor = (time, element, color) => {
16   return new Promise((resolve, reject) => {
17     if (element) {
18       setTimeout(() => {
19         element.style.color = color;
20         resolve();
21       }, time);
22     } else {
23       reject(new Error('There is no such element ${element}'))
24     }
25   });
26 }
```

The browser on the right displays the output of the code, which is a white box with the text "Asynchronous Javascript" at the top. Below it, the text "hello world" (red), "hello people" (green), "hello Javascript" (blue), and "hello Async JS" (brown) is displayed in a sequence. A red box highlights the text "hello Async JS". A "click me" button is at the bottom.

451abce changiojen1 Thu May 4 20:20:44 2023 +0800 W12-P2: use promise to solve the cb hell problem

W12-P3: use async/await to solve the cb hell problem

```
1 const heading1 = document.querySelector('.one');
2 const heading2 = document.querySelector('.two');
3 const heading3 = document.querySelector('.three');
4 const heading4 = document.querySelector('.four');
5 const btn = document.querySelector('.btn');
6
7 btn.addEventListener('click', async () => {
8   const result = await displayColor();
9   console.log('result', result);
10 });
11
12 const displayColor = async () => {
13   try {
14     await addColor(1000, heading1, 'red');
15     await addColor(2000, heading2, 'green');
16     await addColor(1000, heading3, 'blue');
17     await addColor(500, heading4, 'brown');
18     console.log('success');
19   } catch (error) {
20     console.log(error);
21   }
22 }
23
24 const addColor = (time, element, color) => {
```

Asynchronous Javascript

hello world

hello people

hello Javascript

hello Async JS

click me

5d02c80 changiojen1 Thu May 4 20:44:25 2023 +0800 W12-P3: use async/await to solve the cb hell problem

W12-P4: xhr, get sample.txt

success reading

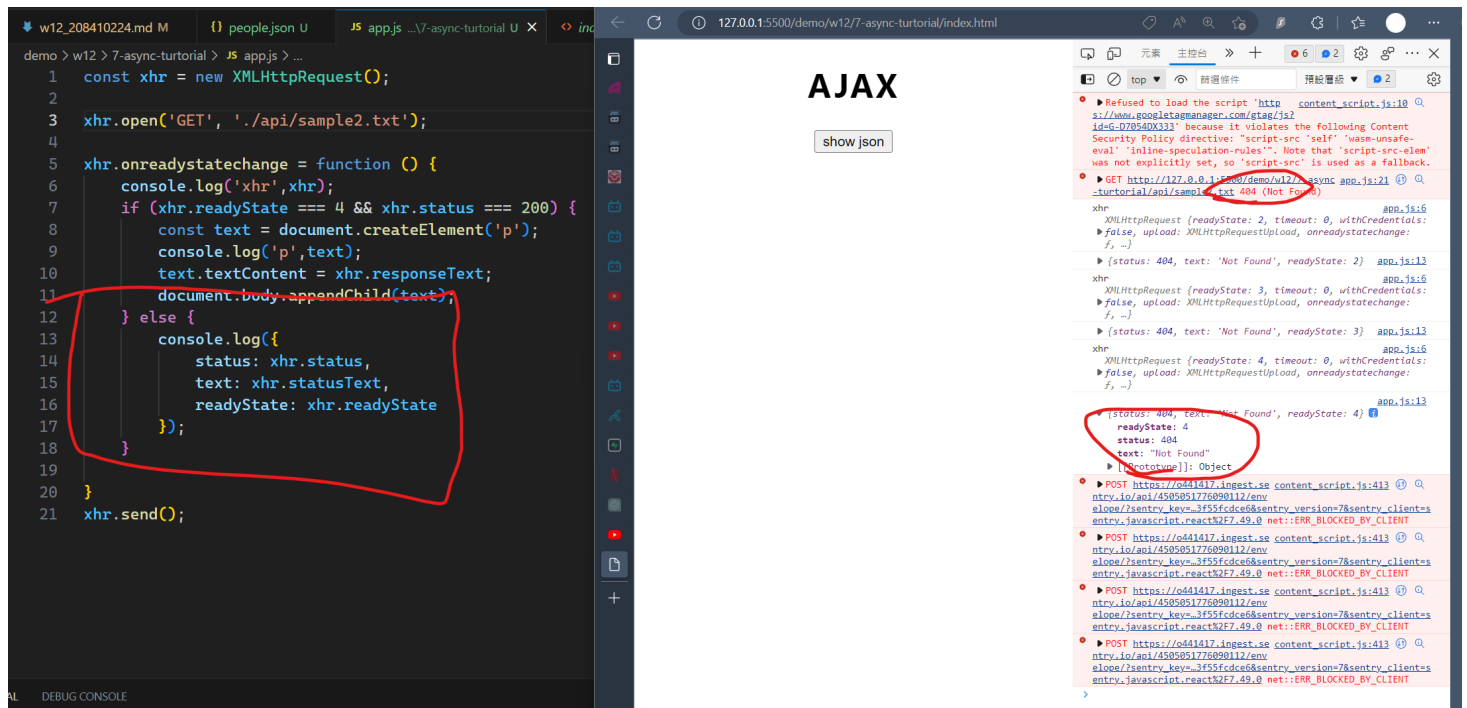
```
1 const xhr = new XMLHttpRequest();
2 xhr.open('GET', './api/sample.txt');
3
4 xhr.onreadystatechange = function () {
5   console.log('xhr', xhr);
6   if (xhr.readyState === 4 && xhr.status === 200) {
7     const text = document.createElement('p');
8     console.log('p', text);
9     text.textContent = xhr.responseText;
10    document.body.appendChild(text);
11  } else {
12    console.log({
13      status: xhr.status,
14      text: xhr.statusText,
15      state: xhr.readyState
16    });
17  }
18 }
19
20
21 xhr.send();
```

AJAX

show json

Lorem ipsum dolor sit amet consectetur adipisicing elit. Fuga facere accusantium laborum maxime adipisci ipsam eveniet itaque molestias excepturi sed, obcaecati, repudiandae porro impedit temporibus pariat quam modi asperiores. Deleniti aliquid adipisci, optio doloribus fugiat molestias a consequuntur nemo, quibusdam tempora placeat. Impedit voluptates velit repudiandae consequatur eos modi ex.

fail reading

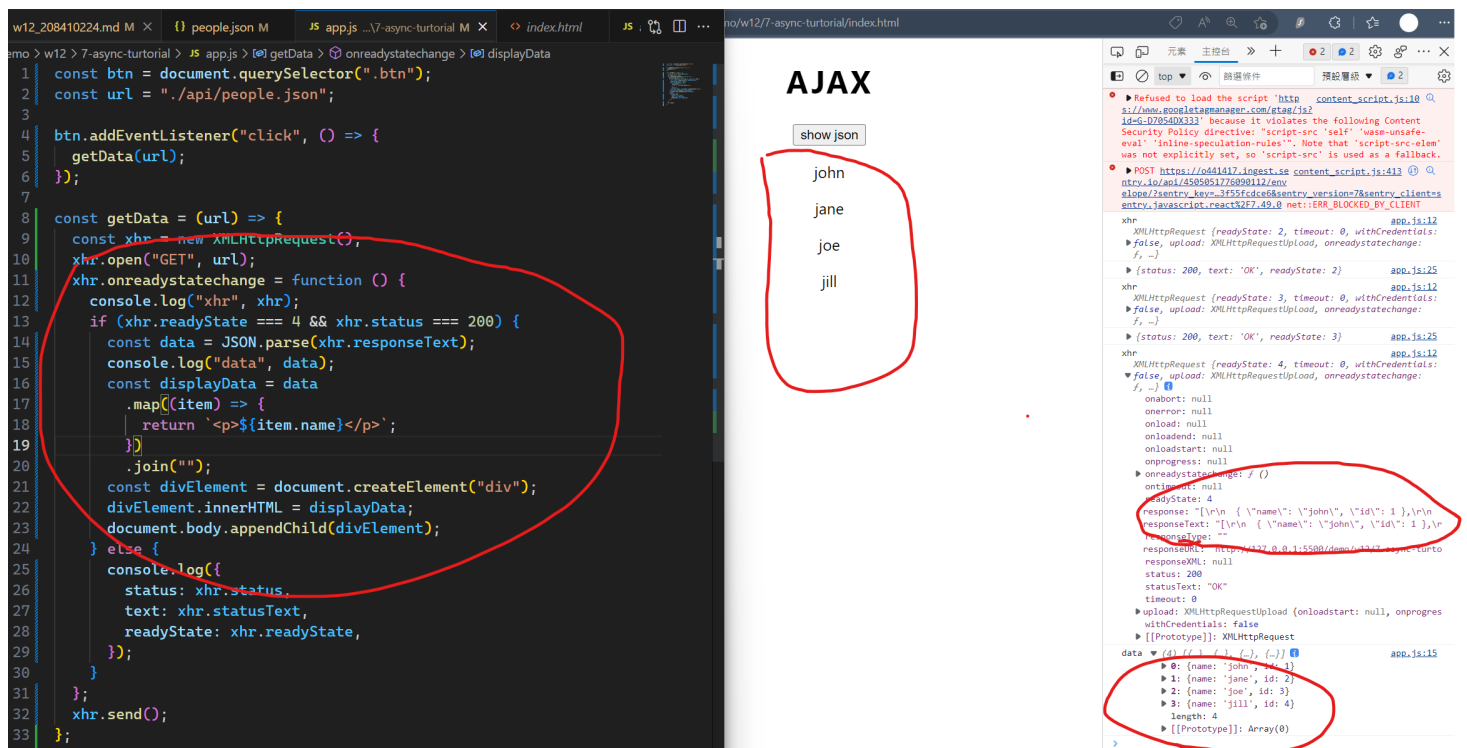


f22880d changiojen1

Thu May 4 21:24:20 2023 +0800

W12-P4: xhr, get sample.txt

W12-P5: xhr, get people.json, and show names in browser



1adfcfd changiojen1

Thu May 4 21:55:52 2023 +0800

W12-P5: xhr, get people.json, and show names in browser

W12-logs

1adfcde changiojen1	Thu May 4 21:55:52 2023 +0800	W12-P5: xhr, get people.json, and show names in browser
f22880d changiojen1	Thu May 4 21:24:20 2023 +0800	W12-P4: xhr, get sample.txt
5d02c80 changiojen1	Thu May 4 20:44:25 2023 +0800	W12-P3: use async/await to solve the cb hell problem
451abce changiojen1	Thu May 4 20:20:44 2023 +0800	W12-P2: use promise to solve the cb hell problem
ec87582 changiojen1	Thu May 4 19:20:02 2023 +0800	P1: call back hell DOM demo