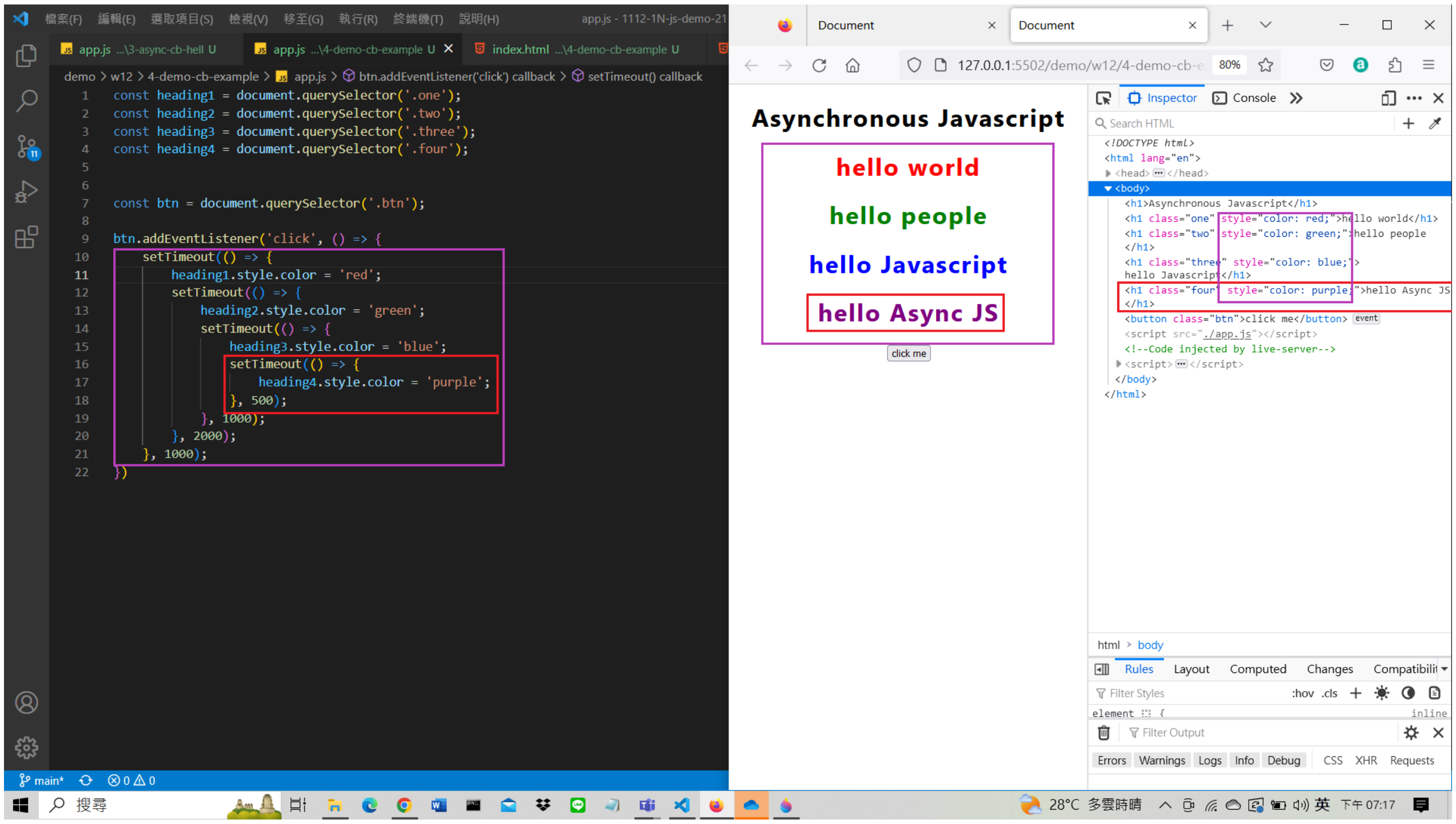
## Github URL

## W12-P1: call back hell DOM demo



```
const heading1 = document.querySelector('.one');
const heading2 = document.querySelector('.two');
const heading3 = document.querySelector('.three');
const heading4 = document.querySelector('.four');


const btn = document.querySelector('.btn');

btn.addEventListener('click', () => {
    setTimeout(() => {
        heading1.style.color = 'red';
        setTimeout(() => {
            heading2.style.color = 'green';
            setTimeout(() => {
                heading3.style.color = 'blue';
                setTimeout(() => {
                    heading4.style.color = 'purple';
                }, 500);
            }, 1000);
        }, 2000);
    }, 1000);
})
```

Asynchronous Javascript

hello world

hello people

hello Javascript

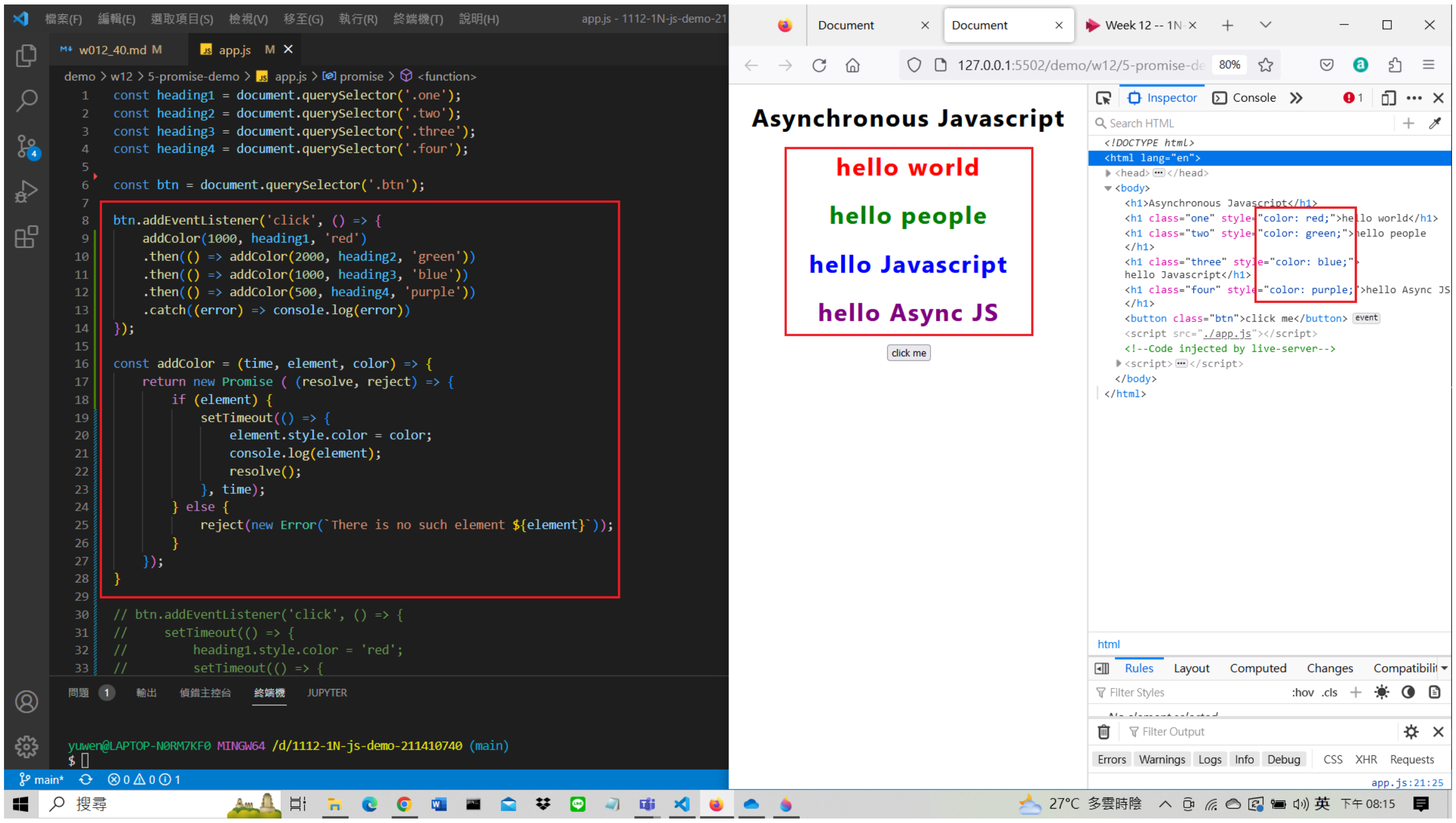
hello Async JS

```
<!DOCTYPE html>
<html lang="en">
<head>…</head>
<body>
  <h1>Asynchronous Javascript</h1>
  <h1 class="one" style="color: red;">hello world</h1>
  <h1 class="two" style="color: green;">hello people
  </h1>
  <h1 class="three" style="color: blue;">
  hello Javascript</h1>
  <h1 class="four" style="color: purple;">hello Async JS
  </h1>
  <button class="btn">click me</button>
  <script src="./app.js"></script>
  <!--Code injected by live-server-->
  <script>…</script>
</body>
</html>
```

demo > w12 > 4-demo-cb-example > app.js > btn.addEventListener('click') callback > setTimeout() callback
127.0.0.1:5502/demo/w12/4-demo-cb-e
click me
Inspector
Console

```
24ea441 yuwen1213        Thu May 4 19:27:48 2023 +0800    W12-P1: call back hell DOM demo
```

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



```
const heading1 = document.querySelector('.one');
const heading2 = document.querySelector('.two');
const heading3 = document.querySelector('.three');
const heading4 = document.querySelector('.four');

const btn = document.querySelector('.btn');

btn.addEventListener('click', () => {
    addColor(1000, heading1, 'red')
    .then(() => addColor(2000, heading2, 'green'))
    .then(() => addColor(1000, heading3, 'blue'))
    .then(() => addColor(500, heading4, 'purple'))
    .catch((error) => console.log(error))
});

const addColor = (time, element, color) => {
    return new Promise ( (resolve, reject) => {
        if (element) {
            setTimeout(() => {
                element.style.color = color;
                console.log(element);
                resolve();
            }, time);
        } else {
            reject(new Error(`There is no such element ${element}`));
        }
    });
}

// btn.addEventListener('click', () => {
//     setTimeout(() => {
//         heading1.style.color = 'red';
//         setTimeout(() => {
```

```
yuwen@LAPTOP-N0RM7KF0 MINGW64 /d/1112-1N-js-demo-211410740 (main)
$ 
```

Asynchronous Javascript

hello world

hello people

hello Javascript

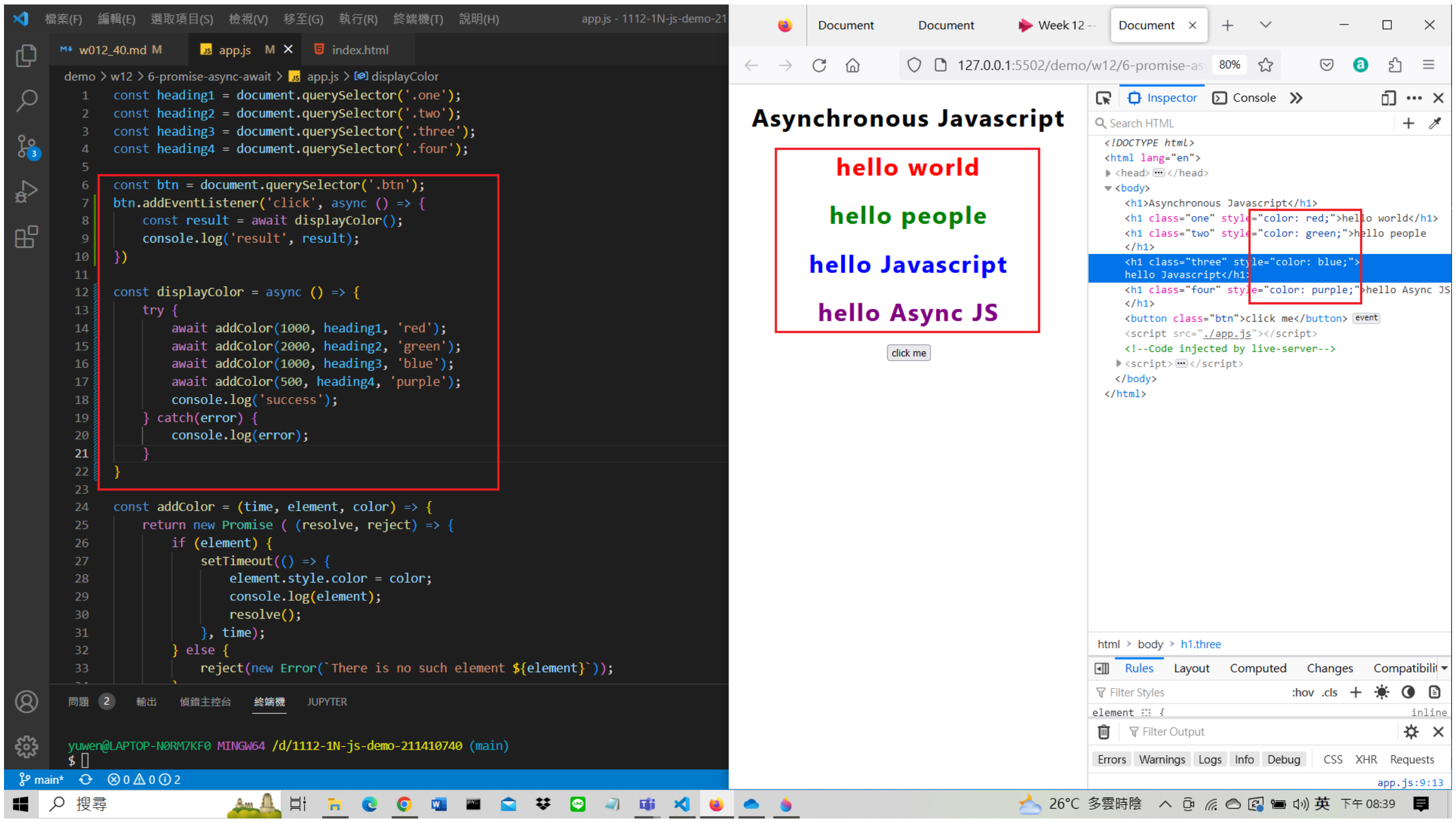
hello Async JS

```
<!DOCTYPE html>
<html lang="en">
  <head>…</head>
  <body>
    <h1>Asynchronous Javascript</h1>
    <h1 class="one" style="color: red;">hello world</h1>
    <h1 class="two" style="color: green;">hello people
    </h1>
    <h1 class="three" style="color: blue;">
    hello Javascript</h1>
    <h1 class="four" style="color: purple;">hello Async JS
    </h1>
    <button class="btn">click me</button>
    <script src="./app.js"></script>
    <!--Code injected by live-server-->
    <script>…</script>
  </body>
</html>
```

檔案(F) 編輯(E) 選取項目(S) 檢視(V) 移至(G) 執行(R) 終端機(T) 說明(H)
app.js - 1112-1N-js-demo-21
w012_40.md M
app.js M
demo > w12 > 5-promise-demo > app.js > promise > <function>
問題 1 輸出 偵錯主控台 終端機 JUPYTER
main*
Document
Week 12 -- 1N
127.0.0.1:5502/demo/w12/5-promise-de
80%
click me
Inspector
Console
Search HTML
html
Rules Layout Computed Changes Compatibili
Filter Styles
Filter Output
Errors Warnings Logs Info Debug CSS XHR Requests
app.js:21:25
搜尋
27°C 多雲時陰
英 下午 08:15

```
cd7c486 yuwen1213       Thu May 4 20:20:40 2023 +0800   W12-P2: use promise to solve the cb hell problem
```

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



```javascript
const heading1 = document.querySelector('.one');
const heading2 = document.querySelector('.two');
const heading3 = document.querySelector('.three');
const heading4 = document.querySelector('.four');

const btn = document.querySelector('.btn');
btn.addEventListener('click', async () => {
    const result = await displayColor();
    console.log('result', result);
})

const displayColor = async () => {
    try {
        await addColor(1000, heading1, 'red');
        await addColor(2000, heading2, 'green');
        await addColor(1000, heading3, 'blue');
        await addColor(500, heading4, 'purple');
        console.log('success');
    } catch(error) {
        console.log(error);
    }
}

const addColor = (time, element, color) => {
    return new Promise ( (resolve, reject) => {
        if (element) {
            setTimeout(() => {
                element.style.color = color;
                console.log(element);
                resolve();
            }, time);
        } else {
            reject(new Error(`There is no such element ${element}`));
```

```
yuwen@LAPTOP-N0RM7KF0 MINGW64 /d/1112-1N-js-demo-211410740 (main)
$
```

Asynchronous Javascript

hello world

hello people

hello Javascript

hello Async JS

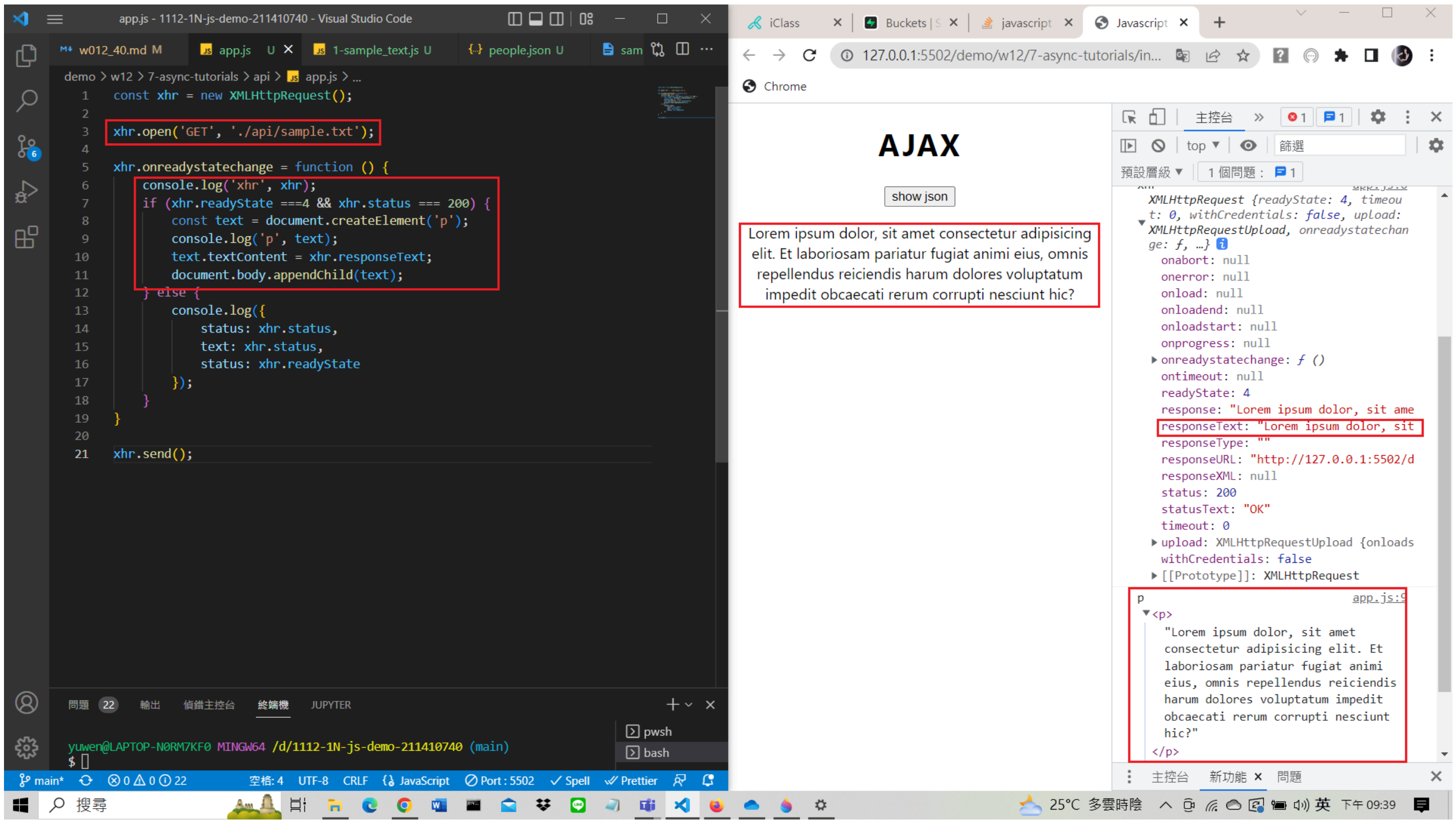
click me

```html
<!DOCTYPE html>
<html lang="en">
  <head>…</head>
  <body>
    <h1>Asynchronous Javascript</h1>
    <h1 class="one" style="color: red;">hello world</h1>
    <h1 class="two" style="color: green;">hello people
    </h1>
    <h1 class="three" style="color: blue;">
    hello Javascript</h1>
    <h1 class="four" style="color: purple;">hello Async JS
    </h1>
    <button class="btn">click me</button> event
    <script src="./app.js"></script>
    <!--Code injected by live-server-->
    <script>…</script>
  </body>
</html>
```

```
97754a7 yuwen1213       Thu May 4 20:42:09 2023 +0800   W12-P3: use async/await to solve the cb hell problem
```

# W12-P4: xhr, get sample.txt

## success reading



```
const xhr = new XMLHttpRequest();

xhr.open('GET', './api/sample.txt');

xhr.onreadystatechange = function () {
    console.log('xhr', xhr);
    if (xhr.readyState ===4 && xhr.status === 200) {
        const text = document.createElement('p');
        console.log('p', text);
        text.textContent = xhr.responseText;
        document.body.appendChild(text);
    } else {
        console.log({
            status: xhr.status,
            text: xhr.status,
            status: xhr.readyState
        });
    }
}

xhr.send();
```

```
yuwen@LAPTOP-N0RM7KF0 MINGW64 /d/1112-1N-js-demo-211410740 (main)
$
```

127.0.0.1:5502/demo/w12/7-async-tutorials/in...

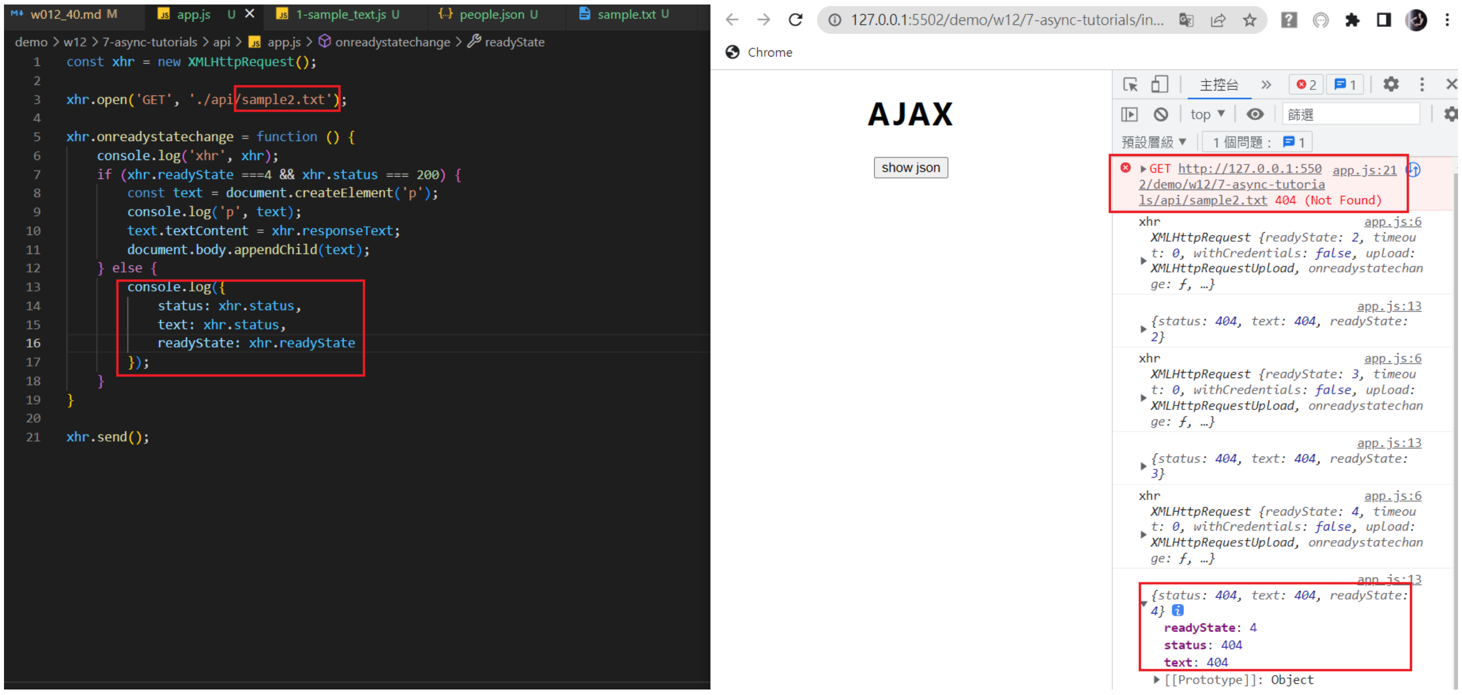
AJAX

show json

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Et laboriosam pariatur fugiat animi eius, omnis repellendus reiciendis harum dolores voluptatum impedit obcaecati rerum corrupti nesciunt hic?

```
XMLHttpRequest {readyState: 4, timeout: 0, withCredentials: false, upload:
XMLHttpRequestUpload, onreadystatechange: f, …}
  onabort: null
  onerror: null
  onload: null
  onloadend: null
  onloadstart: null
  onprogress: null
  onreadystatechange: f ()
  ontimeout: null
  readyState: 4
  response: "Lorem ipsum dolor, sit ame
  responseText: "Lorem ipsum dolor, sit
  responseType: ""
  responseURL: "http://127.0.0.1:5502/d
  responseXML: null
  status: 200
  statusText: "OK"
  timeout: 0
  upload: XMLHttpRequestUpload {onloads
  withCredentials: false
  [[Prototype]]: XMLHttpRequest
p                                          app.js:9
<p>
  "Lorem ipsum dolor, sit amet
  consectetur adipisicing elit. Et
  laboriosam pariatur fugiat animi
  eius, omnis repellendus reiciendis
  harum dolores voluptatum impedit
  obcaecati rerum corrupti nesciunt
  hic?"
</p>
```

## fail reading



```
const xhr = new XMLHttpRequest();

xhr.open('GET', './api/sample2.txt');

xhr.onreadystatechange = function () {
    console.log('xhr', xhr);
    if (xhr.readyState ===4 && xhr.status === 200) {
        const text = document.createElement('p');
        console.log('p', text);
        text.textContent = xhr.responseText;
        document.body.appendChild(text);
    } else {
        console.log({
            status: xhr.status,
            text: xhr.status,
            readyState: xhr.readyState
        });
    }
}

xhr.send();
```

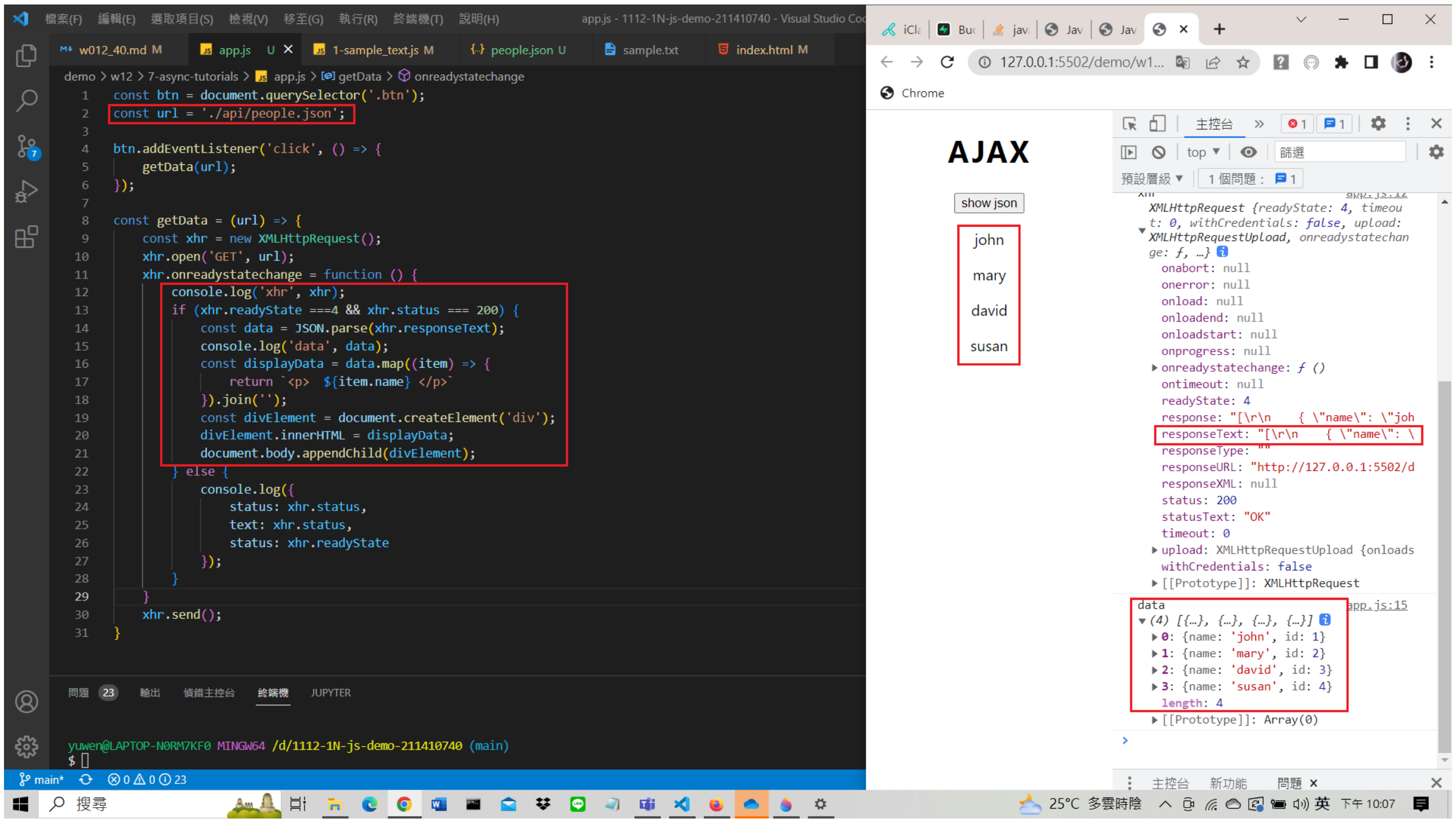
127.0.0.1:5502/demo/w12/7-async-tutorials/in...

AJAX

show json

```
GET http://127.0.0.1:5502/demo/w12/7-async-tutorials/api/sample2.txt 404 (Not Found)   app.js:21
xhr                                                   app.js:6
XMLHttpRequest {readyState: 2, timeout: 0, withCredentials: false, upload:
XMLHttpRequestUpload, onreadystatechange: f, …}
                                                      app.js:13
{status: 404, text: 404, readyState: 2}
xhr                                                   app.js:6
XMLHttpRequest {readyState: 3, timeout: 0, withCredentials: false, upload:
XMLHttpRequestUpload, onreadystatechange: f, …}
                                                      app.js:13
{status: 404, text: 404, readyState: 3}
xhr                                                   app.js:6
XMLHttpRequest {readyState: 4, timeout: 0, withCredentials: false, upload:
XMLHttpRequestUpload, onreadystatechange: f, …}
                                                      app.js:13
{status: 404, text: 404, readyState: 4}
  readyState: 4
  status: 404
  text: 404
  [[Prototype]]: Object
```

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



```js
const btn = document.querySelector('.btn');
const url = './api/people.json';

btn.addEventListener('click', () => {
    getData(url);
});

const getData = (url) => {
    const xhr = new XMLHttpRequest();
    xhr.open('GET', url);
    xhr.onreadystatechange = function () {
        console.log('xhr', xhr);
        if (xhr.readyState ===4 && xhr.status === 200) {
            const data = JSON.parse(xhr.responseText);
            console.log('data', data);
            const displayData = data.map((item) => {
                return `<p>  ${item.name} </p>`
            }).join('');
            const divElement = document.createElement('div');
            divElement.innerHTML = displayData;
            document.body.appendChild(divElement);
        } else {
            console.log({
                status: xhr.status,
                text: xhr.status,
                status: xhr.readyState
            });
        }
    }
    xhr.send();
}
```

```
yuwen@LAPTOP-N0RM7KF0 MINGW64 /d/1112-1N-js-demo-211410740 (main)
$
```

AJAX

show json

john

mary

david

susan

```
XMLHttpRequest {readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: ƒ, …}
  onabort: null
  onerror: null
  onload: null
  onloadend: null
  onloadstart: null
  onprogress: null
▸ onreadystatechange: ƒ ()
  ontimeout: null
  readyState: 4
  response: "[\r\n    { \"name\": \"joh
  responseText: "[\r\n    { \"name\": \
  responseType: ""
  responseURL: "http://127.0.0.1:5502/d
  responseXML: null
  status: 200
  statusText: "OK"
  timeout: 0
▸ upload: XMLHttpRequestUpload {onloads
  withCredentials: false
▸ [[Prototype]]: XMLHttpRequest
data                                          app.js:15
▾ (4) [{…}, {…}, {…}, {…}]
  ▸ 0: {name: 'john', id: 1}
  ▸ 1: {name: 'mary', id: 2}
  ▸ 2: {name: 'david', id: 3}
  ▸ 3: {name: 'susan', id: 4}
    length: 4
  ▸ [[Prototype]]: Array(0)
```

```
25e09bb yuwen1213       Thu May 4 22:10:29 2023 +0800   W12-P5: xhr, get people.json, and show names in browser
```

# W12-logs

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2023-05-03"
```

```
25e09bb yuwen1213       Thu May 4 22:10:29 2023 +0800   W12-P5: xhr, get people.json, and show names in browser
6963295 yuwen1213       Thu May 4 21:47:23 2023 +0800   W12-P4: xhr, get sample.txt
97754a7 yuwen1213       Thu May 4 20:42:09 2023 +0800   W12-P3: use async/await to solve the cb hell problem
cd7c486 yuwen1213       Thu May 4 20:20:40 2023 +0800   W12-P2: use promise to solve the cb hell problem
24ea441 yuwen1213       Thu May 4 19:27:48 2023 +0800   W12-P1: call back hell DOM demo
```