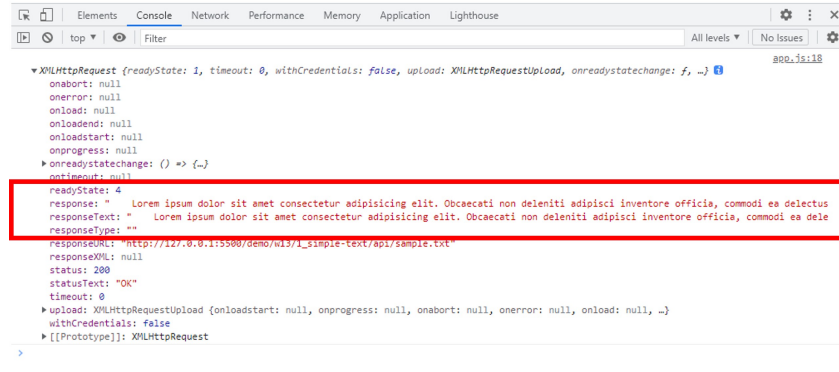


w13-p1: use xhr object to get sample.txt, and show it on webpage

AJAX -- Hsintal, 210410196

Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati non deleniti adipisci inventore officia, commodi ea delectus quod, placeat dolor voluptatum ipsa excepturi laborum. Reprehenderit nulla molestiae sit autem recusandae?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati non deleniti adipisci inventore officia, commodi ea delectus quod, placeat dolor voluptatum ipsa excepturi laborum. Reprehenderit nulla molestiae sit autem recusandae?



The screenshot shows the browser's developer console with the 'Network' tab selected. A red box highlights the details of an XMLHttpRequest. The 'readyState' is 4, indicating the request is complete. The 'responseText' is the Lorem ipsum text. The 'status' is 200, and the 'statusText' is 'OK'.

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

w13-p2: use xhr object to get people.json, and show it on webpage

AJAX -- Hsintal, 210410196

show json

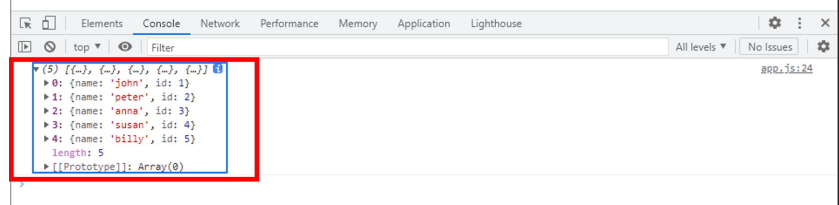
john

peter

anna

susan

billy



The screenshot shows the browser's developer console with the 'Network' tab selected. A red box highlights the details of an XMLHttpRequest. The 'readyState' is 4, indicating the request is complete. The 'responseText' is a JSON array of objects, each representing a person with a name and an id. The 'status' is 200, and the 'statusText' is 'OK'.

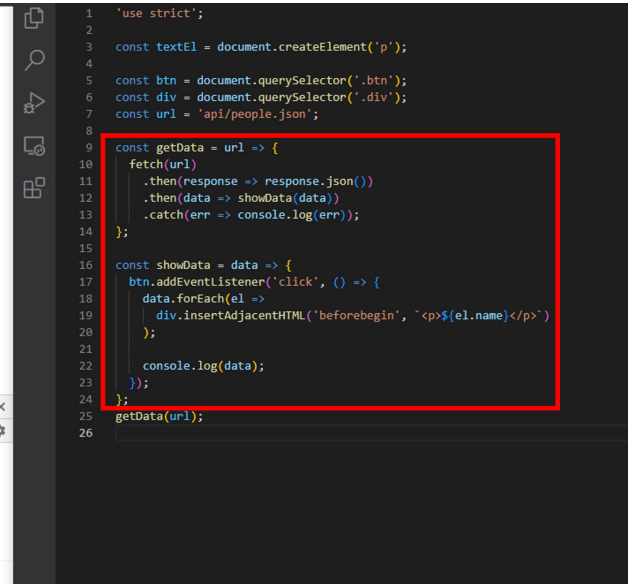
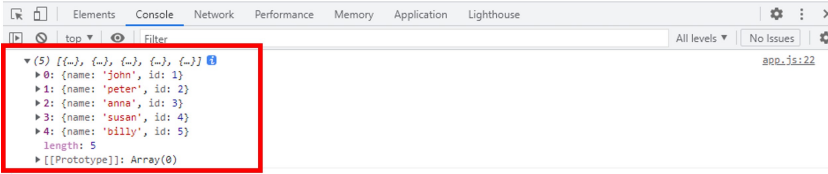
```
1 'use strict';
2
3 const textEl = document.createElement('p');
4
5 const btn = document.querySelector('.btn');
6 const div = document.querySelector('.div');
7 const url = 'api/people.json';
8
9 const getData = url => {
10   const request = new XMLHttpRequest();
11   request.open('GET', url);
12   request.send();
13
14   if (!request.readyState && request.status != 200)
15     console.log({ status: request.status, text: request.statusText });
16
17   btn.addEventListener('click', function () {
18     const data = JSON.parse(request.responseText);
19
20     data.forEach(el => {
21       div.insertAdjacentHTML('beforebegin', `<p>${el.name}</p>`);
22     });
23
24     console.log(data);
25   });
26 };
27
28 getData(url);
29
```

w13-p3: use fetch api to get people.json, and show it on webpage

AJAX -- Hsintal, 210410196

show json

john
peter
anna
susan
billy

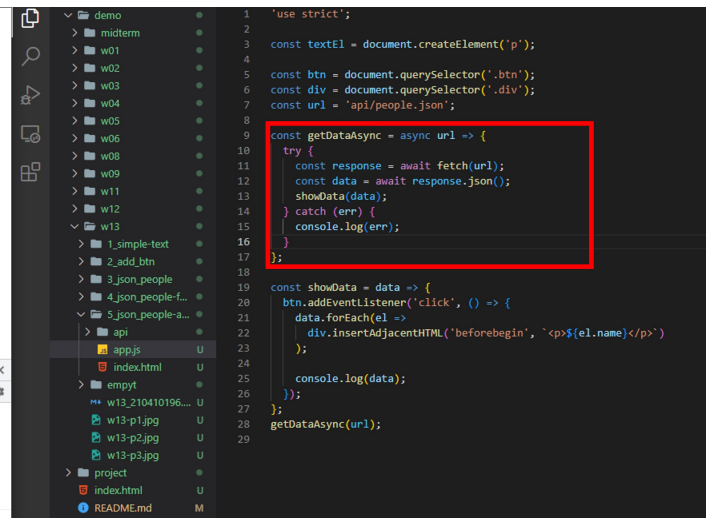
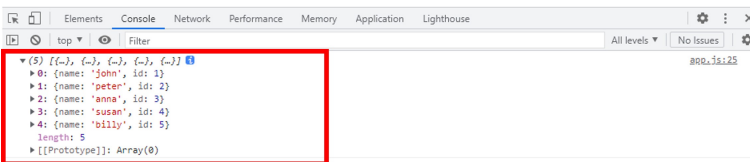


w13-p4: use async/await to get people.json, and show it on webpage

AJAX -- Hsintal, 210410196

show json by async/await

john
peter
anna
susan
billy



w13-p5: w1-p5 (挑戰題)

```
demo > w13 > 6_midterm_p2_210410196 > app.js > ...
1 'use strict';
2
3 const textEl = document.createElement('p');
4
5 const btn = document.querySelector('.btn');
6 const main = document.querySelector('#main');
7 const url = 'api/dataset.json';
8
9 const dataArray = [];
10
11 function formatMoney(number) {
12   return '$' + number.toFixed(2).replace(/\d(?=(\d{3})+\.)/g, '$&,');
13 }
14
15 const getDataAsync = async url => {
16   try {
17     const response = await fetch(url);
18     const data = await response.json();
19     data.map(el => dataArray.push(el));
20     showData(data);
21   } catch (err) {
22     console.log(err);
23   }
24 };
25
26 const showData = data => {
27   let allData = data
28   .map(
29     el =>
30     `<div class="person"><strong>${el.name}</strong> ${formatMoney(
31       el.money
32     )}</div>`
33   )
34   .join('');
35
36   btn.addEventListener('click', () => {
37     main.innerHTML = `<h2><strong>Person</strong> Wealth</h2> ${allData}`;
38   });
39
40   console.log(dataArray);
41 };
42
43 getDataAsync(url);
```

AJAX -- Hsintal, 210410196

show json by 6_midterm_p2_210410196

Person	Wealth
Ellen Cruz	\$561,575,400.00
Raul Velasco	\$41,338,300.00
Konsta Tuominen	\$112,175,400.00
Billy Wang	\$123,754,000.00

Elements

Console

Network

Performance

Memory

Application

Lighthouse

top

Filter

All levels

No Issues

(4) [(-), (-), (-), (-)]

0: {name: 'Ellen Cruz', money: 561575400}

1: {name: 'Raul Velasco', money: 41338300}

2: {name: 'Konsta Tuominen', money: 112175400}

3: {name: 'Billy Wang', money: 123754000}

length: 4

[[Prototype]]: Array(0)