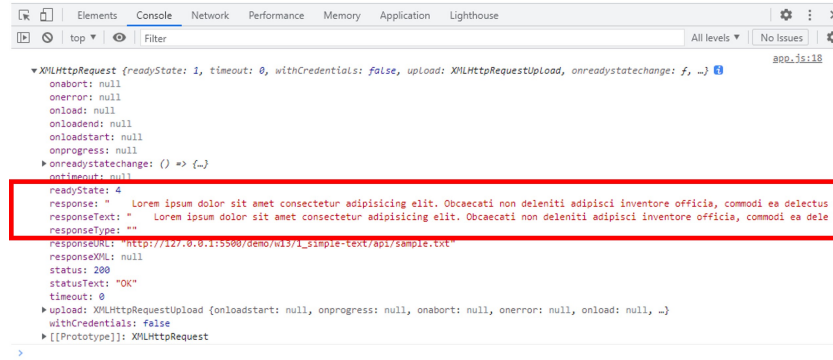


## w13-p1: use xhr object to get sample.txt, and show it on webpage

### 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?



```
XMLHttpRequest (readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: f, ...)
  abort: null
  onerror: null
  onload: null
  onloadend: null
  onloadstart: null
  onprogress: null
  onreadystatechange: () => { ... }
  response: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati non deleniti adipisci inventore officia, commodi ea delectus dolor voluptatum ipsa excepturi laborum. Reprehenderit nulla molestiae sit autem recusandae?"
  responseText: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati non deleniti adipisci inventore officia, commodi ea delectus dolor voluptatum ipsa excepturi laborum. Reprehenderit nulla molestiae sit autem recusandae?"
  responseType: ""
  responseURL: "http://127.0.0.1:5500/demo/w13/p1/sample-text/api/sample.txt"
  status: 200
  statusText: "OK"
  timeout: 0
  upload: XMLHttpRequestUpload {onloadstart: null, onprogress: null, onabort: null, onerror: null, onload: null, ...}
  withCredentials: false
  [[Prototype]]: XMLHttpRequest
```

```
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

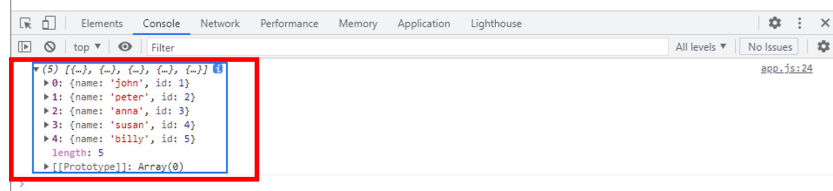
john

peter

anna

susan

billy



```
(5) [{"name": "john", id: 1}, {"name": "peter", id: 2}, {"name": "anna", id: 3}, {"name": "susan", id: 4}, {"name": "billy", id: 5}]
  length: 5
  [[Prototype]]: Array(0)
```

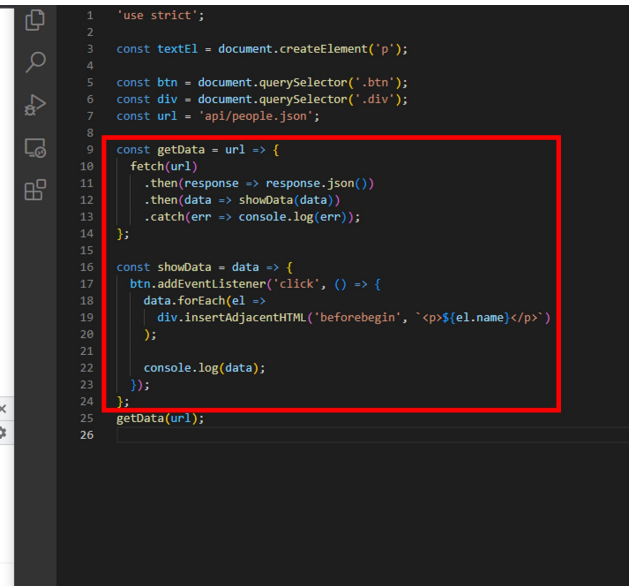
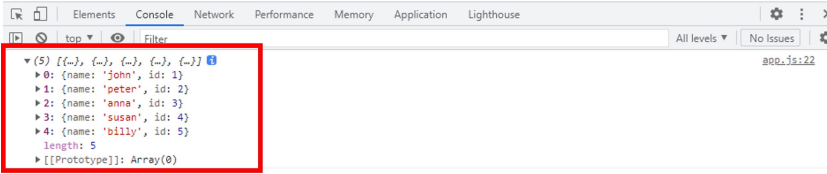
```
1 'use strict';
2
3 const textEl = document.createElement('p');
4
5 const btn = document.querySelector('.btn');
6 const div = document.querySelector('.div');
7 const url = 'api/people.json';
8
9 const getData = url => {
10   const request = new XMLHttpRequest();
11   request.open('GET', url);
12   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);
```

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

## AJAX -- Hsintal, 210410196

show json

john  
peter  
anna  
susan  
billy

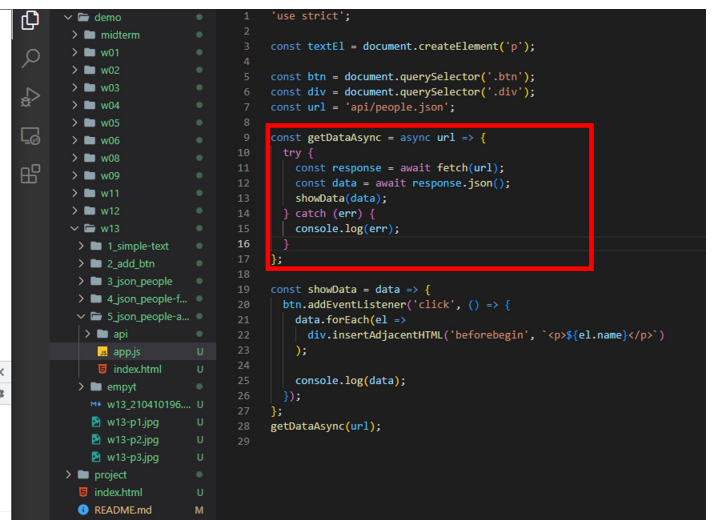
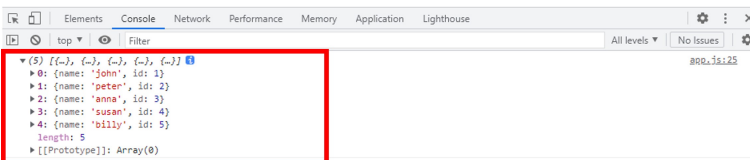


## w13-p4: use async/await to get people.json, and show it on webpage

### AJAX -- Hsintal, 210410196

show json by async/await

john  
peter  
anna  
susan  
billy



## w13-p5: w1-p5 (挑戰題)

### AJAX -- Hsintal, 210410196

show json by async/await

john  
peter  
anna  
susan  
billy

