

Problem 1

- (1) <script> 태그 뒤에 작성된 HTML 코드를 JS 코드가 인식하지 못해 생긴 문제이다.
- (2) 1. script 태그를 body태그 내부의 중간이나 끝에 작성한다.
2. script 태그에 async나 defer 속성을 활성화한다.

Problem 2

- (1) Title, Content, Post ID 체크박스들이 중복 체크된다.
- (2) Input 속성을 checkbox에서 radio로 바꾸고, name을 ~~로 설정한다.

Problem 3

- (1) document.querySelector(".post") 가 .post 클래스를 가진 첫번째 요소를 선택하기에 length 속성이 존재하지 않는다. 따라서 while (this.elements.length) 이 while문이 실행되지 않기에 삭제가 되지 않는다.
- (2) post-container를 선택해 자식들을 하나씩 지워준다.
- (3)

```
clear: function () {  
    const container = document.querySelector("#post-container");  
    while (container.firstChild) {  
        container.removeChild(container.firstChild);  
    }  
},
```

Problem 4

- (1) 자료형까지 고려하여 값을 비교하는 strict equality를 사용했는데, post.id는 Number 자료형이고, postid는 String 자료형이라서 항상 false가 나오게 된다.
- (2) 1. return data.posts.filter((post) => post.id === parseInt(postId));
2. return data.posts.filter((post) => post.id.toString() === postId);

Problem 5

- (1) `if (isNaN(parseInt(postId)) || parseInt(postId) <= 0)`
- (2) `alert("Post ID는 1이상 정수입니다.");`

Problem 6

- (1) `return data.users.find((user) => user.id === userId);`
- (2) `for (const i of data.users) {if (i.id === userId) { return i; } }`

Problem 7

- (1) index.html은 이 웹 페이지의 구조를 표현한다.
style.css는 index.html의 요소들의 스타일과 레이아웃을 기술한다.
script.js는 HTML DOM 요소를 조작해 검색 기능을 구현한다.
interface.js는 data.js의 데이터들을 가져오는 함수들이 정의 되어있다.
data.js는 user들의 정보와 글들의 정보를 저장한다
- (2) HTML 문서가 로드되면서 <head> 섹션의 style.css와 <body> 섹션의 data.js, interface.js, script.js 파일들이 순차적으로 로드된다.
사용자가 검색 입력창에 검색어를 입력하고 검색 버튼을 클릭하면, script.js에서 eventlistener가 동작해 검색어와 선택된 검색 옵션을 바탕으로 board.search() 메서드가 호출된다. 그렇게 하면 interface.js에서 데이터를 가져온다.
검색 결과는 board.render()를 통해 화면에 렌더링되어 사용자에게 표시된다.

Problem 8

(1) 우선 html 문서의 meta 태그에 viewport를 설정하고 css에 media query 구문을 사용한다.

(2)

☐ Title ☐ Content ☐ Post ID

Search

Hello World

This is my first post! Alice (alice@korea.ac.kr)

My Second Post

Hello world again! Alice (alice@korea.ac.kr)

I am too much talker

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Integer quis auctor elit sed vulputate. Lorem ipsum dolor sit amet. Quam nulla porttitor massa id. Metus vulputate eu scelerisque felis imperdiet proin fermentum leo vel. Tristique nulla aliquet enim tortor. Massa massa ultricies mi quis hendrerit dolor magna eget est. David (david@korea.ac.kr)

Coffee Chat

Let's talk about coffee. Bob (bob@korea.ac.kr)

Tech Trends

Latest in tech July 2024. Charlie (charlie@korea.ac.kr)

Morning Routines

How do you start your day? Charlie (charlie@korea.ac.kr)

Book Club

Discussing 'The Great Gatsby'. Charlie (charlie@korea.ac.kr)

Favorite Recipes

Share your best recipes. Bob (bob@korea.ac.kr)

Fitness Goal

Set fitness goals for the year. Charlie (charlie@korea.ac.kr)

Music Reviews

Reviewing the latest albums. Eve (eve@korea.ac.kr)

Problem 9

```
document.querySelector("#search-input").addEventListener("keydown", event => {
  if (event.code == "Enter") {
    const value = document.querySelector("#search-input").value;
    const option = document.querySelector(".search-option >
input:checked").value;
    board.search(value, option);
  }
});
```

Problem 10

- (1) href 속성은 span이 아니라 a태그에서 동작하기 때문이다.
- (2) a 태그를 만들고, 해당 a태그에 href속성을 넣는다. userSpan의 하위 요소로 넣는다.
userSpan의 innerText를 없애고 같은 것을 a태그에 넣는다.
- (3)

```
const userSpan = document.createElement("span");
userSpan.classList.add("post-user");
//userSpan.innerText = user.name + " (" + user.email + ")";
postDiv.appendChild(userSpan);

const emailLink = document.createElement("a");
emailLink.innerText = user.name + " (" + user.email + ")";
emailLink.setAttribute("href", "mailto:" + user.email);
userSpan.appendChild(emailLink);
```

Problem 11

```
.post-user a {
  color: #666;
  cursor: pointer;
  text-decoration: none;
}
```

Problem 12

```
button:hover {
  background-color: skyblue;
  border: 1px solid black;
  font-weight: bold;
  font-size: 20px;
  color: black;
}
```

Problem 13

```
@media (max-width: 799px) {
  .post-content {
    overflow: hidden;
    white-space: nowrap;
    text-overflow: ellipsis;
  }
}
```

Problem 14

this가 board를 가리키는 것이 아니게 되어 posts와 render를 참조할 수 없다.

Problem 15

(1) var은 function scope이지만, let은 block scope 라서 case 문 밖에서 posts 라는 변수에 접근하면 오류가 생긴다.

(2)

```
search: function (value, option) {  
  this.clear();  
  let posts;  
  switch (option) {  
    case "title":  
      posts = interface.getPostsByTitle(value); break;  
    case "content":  
      posts = interface.getPostsByContent(value); break;  
    case "id":  
      posts = interface.getPostsById(value); break;  
  }  
  this.posts = posts;  
  this.render();  
},
```