

Web Storage

개요

JavaScript의 Web Storage는 클라이언트 측에서 데이터를 저장할 수 있는 방법을 제공합니다. 이는 브라우저 내에 데이터를 저장하며, 서버와의 통신 없이 클라이언트에서 데이터를 영구적으로 혹은 세션 동안 유지할 수 있게 해줍니다. Web Storage는 두 가지 주요 저장소로 구성되어 있습니다: localStorage와 sessionStorage입니다.

localStorage

localStorage는 브라우저가 종료되더라도 데이터가 사라지지 않고 계속해서 남아 있는 저장소입니다. 이는 특정 웹사이트에 대한 데이터를 영구적으로 저장하는 데 유용합니다.

메서드

- `setItem(key, value)`: 주어진 키와 값을 저장합니다.
- `getItem(key)`: 주어진 키에 해당하는 값을 반환합니다.
- `removeItem(key)`: 주어진 키에 해당하는 데이터를 삭제합니다.
- `clear()`: 모든 데이터를 제거합니다.
- `key(index)`: 주어진 인덱스에 있는 키를 반환합니다.

sessionStorage

sessionStorage는 페이지 세션이 끝나면 데이터가 사라지는 저장소입니다. 이는 탭 또는 창을 닫을 때 데이터를 유지하지 않아야 하는 경우에 유용합니다.

메서드

- `setItem(key, value)`: 주어진 키와 값을 저장합니다.
- `getItem(key)`: 주어진 키에 해당하는 값을 반환합니다.
- `removeItem(key)`: 주어진 키에 해당하는 데이터를 삭제합니다.
- `clear()`: 모든 데이터를 제거합니다.
- `key(index)`: 주어진 인덱스에 있는 키를 반환합니다.

사용 예제

localStorage 사용 예제

```
localStorage.setItem('username', 'JohnDoe');  
  
console.log(localStorage.getItem('username')); // JohnDoe  
  
localStorage.removeItem('username');  
  
localStorage.clear();
```

sessionStorage 사용 예제

```
sessionStorage.setItem('sessionID', '123456');  
  
console.log(sessionStorage.getItem('sessionID')); // 123456  
  
sessionStorage.removeItem('sessionID');  
  
sessionStorage.clear();
```

주의 사항

Web Storage는 동일한 출처 정책(Same-Origin Policy)을 따르며, 이는 데이터가 저장되는 도메인, 프로토콜, 포트가 동일해야 함을 의미합니다. 또한, 저장 용량이 제한되어 있어 대량의 데이터를 저장하는 데 적합하지 않습니다.

JavaScript의 Web Storage는 간단하고 직관적인 방법으로 클라이언트 측 데이터를 저장하는 강력한 도구입니다.