

Web Storage API

Web Storage API는 브라우저에서 데이터를 키-값 쌍으로 안전하게 저장할 수 있는 메커니즘을 제공하는 웹 API입니다. 일반적으로 쿠키보다 더 많은 데이터를 저장할 수 있으며(대개 5-10MB), 웹 애플리케이션에서 더 큰 양의 데이터를 클라이언트 측에 저장하기에 적합합니다.

Web Storage API	설명
LocalStorage	로컬 스토리지는 브라우저 세션이 끝나도 데이터가 사라지지 않는 저장소입니다. 웹 사이트의 여러 세션 간에 데이터를 지속적으로 저장하고 싶을 때 사용할 수 있습니다. 각 도메인에 대해 별도로 저장되며, 서버로 전송되지 않습니다.
SessionStorage	세션 스토리지는 브라우저 탭이나 창이 닫힐 때까지 데이터를 유지하는 저장소입니다. 한 페이지의 세션 동안만 정보를 저장하고 싶을 때 유용하며, 각 탭 또는 창마다 별도의 세션 스토리지가 있습니다.

local storage

LocalStorage는 HTML5 Web Storage API의 일부로, 웹 브라우저 내에 데이터를 키-값 쌍 형태로 저장할 수 있는 시스템입니다. 웹 애플리케이션은 사용자의 브라우저에 데이터를 영구적으로 저장할 수 있으며, 브라우저를 닫았다가 다시 열어도 저장된 데이터를 사용할 수 있습니다.

대부분의 브라우저에서 LocalStorage는 약 5MB 정도의 데이터를 저장할 수 있습니다. LocalStorage에 저장된 데이터는 만료 기간이 없으며, 사용자가 명시적으로 지우거나 브라우저의 캐시를 비우기 전까지 유지됩니다. 데이터는 도메인별로 분리되어 저장되므로, 다른 도메인의 LocalStorage 데이터에 접근할 수 없습니다.

- LocalStorage를 사용하는 기본적인 방법
저장하기: `localStorage.setItem('key', 'value');` 가져오기: `localStorage.getItem('key');` 삭제하기: `localStorage.removeItem('key');` 모두 삭제하기: `localStorage.clear();`

```
// LocalStorage 데이터 저장
localStorage.setItem('key', 'value');

// LocalStorage 데이터 가져오기
let data = localStorage.getItem('key');

// LocalStorage 데이터 제거
localStorage.removeItem('key');
```

session storage

SessionStorage는 HTML5 Web Storage API의 일부로, 웹 페이지의 세션 동안 데이터를 저장할 수 있게 해주는 저장소입니다. LocalStorage와 유사하게 작동하지만, 주요 차이점은 저장된 데이터의 생명주기와 범위에 있습니다.

SessionStorage에 저장된 데이터는 사용자가 브라우저 탭이나 창을 닫을 때까지만 유지됩니다. 즉, 브라우저 세션이 종료되면 저장된 데이터도 사라집니다. LocalStorage와 마찬가지로 대부분의 브라우저에서 SessionStorage는 약 5MB 정도의 데이터를 저장할 수 있습니다.

- SessionStorage 사용 방법
저장하기: `sessionStorage.setItem('key', 'value');` 가져오기: `sessionStorage.getItem('key');` 삭제하기: `sessionStorage.removeItem('key');` 모두 삭제하기: `sessionStorage.clear();`

```
// SessionStorage에 데이터 저장
sessionStorage.setItem('bakery', '모카빵');

// SessionStorage에서 데이터 가져오기
const bakery = sessionStorage.getItem('bakery');
console.log('SessionStorage에서 가져온 bakery:', bakery);

// 브라우저 세션이 끝나면(브라우저가 닫히거나 탭이 닫힘) SessionStorage의 데이터는 삭제됨
```

cookie

쿠키(Cookies)는 웹 스토리지의 한 형태이긴 하지만, Web Storage API에 포함되는 LocalStorage와 SessionStorage와는 다른 목적과 특성을 가진 데이터 저장 수단입니다. 쿠키와 Web Storage는 모두 클라이언트 측에서 데이터를 저장할 수 있게 해주지만, 몇 가지 중요한 차이점이 있습니다.

- 쿠키와 웹 스토리지 차이

분류	쿠키	Web Storage (LocalStorage & SessionStorage)
목적 및 사용 방식	서버와 클라이언트 간의 상태 정보 유지. HTTP 헤더를 통해 자동 전송됨.	클라이언트 측에서만 필요한 정보 저장. 서버로 자동 전송되지 않음.
용량	상대적으로 작음 (대개 4KB).	더 큰 데이터 저장 가능 (5-10MB).
수명	만료 날짜 설정 가능. 만료되면 자동 삭제.	LocalStorage: 영구적 저장. 수동 삭제 필요. SessionStorage: 탭/창 닫힐 때 삭제.
보안	HTTP 요청과 함께 전송되므로 CSRF 등의 보안 문제 가능성 있음.	서버로 전송되지 않아 쿠키보다 보안이 강화됨.

- 로컬에서 쿠키 확인하는 방법

로컬 파일 시스템에서 HTML 파일을 직접 열 때 보안 정책 때문에 쿠키 설정이 작동하지 않을 수 있습니다. 대부분의 브라우저는 보안상의 이유로 로컬 파일에서 실행되는 JavaScript에서 쿠키 설정을 허용하지 않습니다.

로컬에서 쿠키를 테스트하려면 웹 서버를 설정하고 해당 HTML 파일을 서버에서 제공해야 합니다. 가장 간단한 방법은 Node.js의 내장 모듈인 http-server를 사용하는 것입니다.

`npm install -g http-server` Nodejs 내장 모듈 서버 설치
`http-server` 실행

```
document.cookie = "username=John Doe; expires=Thu, 18 Dec 2023
12:00:00 UTC; path="/";

// 쿠키 생성 (유효 기간: 1시간)
document.cookie
  = "bread=소금빵; expires="
  + new Date(new Date().getTime() + 60 * 60 * 1000).toUTCString()
  + "; path="/";

// 쿠키에서 데이터 가져오기
const cookies = document.cookie.split(';');
let bread;
for (let cookie of cookies) {
  const [name, value] = cookie.trim().split('=');
  if (name === 'bread') {
    bread = decodeURIComponent(value);
    break;
  }
}

console.log('쿠키에서 가져온 bread:', bread);
```

- 브라우저에서 쿠키 확인 방법

대부분의 웹 브라우저에는 쿠키를 관리하는 도구가 내장되어 있습니다. 브라우저의 개발자 도구에서 'Application' 또는 'Storage' 탭을 확인하면 현재 웹사이트에 대한 쿠키 목록과 값을 볼 수 있습니다.