Web Storage and Cookies in JavaScript

Web storage and cookies are used to store data on the client side. JavaScript provides APIs for both web storage (local storage and session storage) and cookies.

Local Storage

Local storage allows you to store key-value pairs in the browser with no expiration date. This means the data persists even after the browser is closed and reopened.

Syntax

- **Set Item**: `localStorage.setItem(key, value)`
- **Get Item**: `localStorage.getItem(key)`
- **Remove Item**: `localStorage.removeItem(key)`
- **Clear Storage**: `localStorage.clear()`

Example

```
// Set item
localStorage.setItem("username", "john_doe");

// Get item
let username = localStorage.getItem("username");
console.log(username); // Output: john_doe

// Remove item
localStorage.removeItem("username");

// Clear storage
localStorage.clear();
```

Session Storage

Session storage is similar to local storage, but the data is cleared when the page session ends. A page session lasts as long as the browser is open, and survives over page reloads and restores.

Syntax

- **Set Item**: `sessionStorage.setItem(key, value)`
- **Get Item**: `sessionStorage.getItem(key)`
- **Remove Item**: `sessionStorage.removeItem(key)`
- **Clear Storage**: `sessionStorage.clear()`

Example

```
// Set item
sessionStorage.setItem('username', 'john_doe');

// Get item
let username = sessionStorage.getItem('username');
console.log(username); // Output: john_doe

// Remove item
sessionStorage.removeItem('username');

// Clear storage
sessionStorage.clear();
```

Cookies

Cookies are data sent to and stored on the client-side that are sent back to the server with each request. They have expiration dates and can be used for session management, user tracking, and personalization.

Syntax

- **Set Cookie**: document.cookie = "key=value; expires=date; path=/"
- **Get Cookie**: document.cookie
- **Delete Cookie**: document.cookie = "key=value; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/"

Example

`Set Cookie`

- **key**: The name of the cookie.
- **value**: The value of the cookie.
- **expires**: The expiration date in GMT format. If not set, the cookie will expire at the end of the session (when the browser closes).
- **path**: The path within the domain that the cookie is valid for. If not specified, defaults to the path of the current document.

```
// Set cookie
document.cookie =
  "username=john_doe; expires=Fri, 31 Dec 2024 23:59:59 GMT; path=/";
```

`Get Cookie` To retrieve a cookie, you can use the **document.cookie** property, which returns all cookies in a single string. You can then parse this string to find the specific cookie you're interested in.

```
// Get cookie
function getCookie(name) {
  let cookieArr = document.cookie.split(";");
  for (let i = 0; i < cookieArr.length; i++) {
    let cookiePair = cookieArr[i].split("=");

    if (name == cookiePair[0].trim()) {
      return decodeURIComponent(cookiePair[1]);
    }
  }
  return null;
}

let username = getCookie("username");
console.log(username); // Output: john_doe
```

`Delete Cookie` To delete a cookie, you can set its expiration date to a past date.

```
// Delete cookie
document.cookie =
  "username=john_doe; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/";
```
