Основи збереження даних в Веб. Web storage API

Web storage API забезпечуються механізми збереження даних на стороні клієнта.

В основі лежать такі механізми (властивості об’єкта window)

|  |  |
| --- | --- |
| sessionStorage | забезпечує збереження даних для кожного домену на протязі сесії (поки відкрито браузер, навіть при перезавантаженні сторінки) |
| localStorage | забезпечує збереження даних навіть при перевідкритті браузера |

Використання localStorage

Дані зберігається у формі впорядкованих пар ("key", "value"). При цьому значення "value" обов’язково повинно бути значенням типу string.

|  |  |  |
| --- | --- | --- |
| Призначення | Загальна форма | Приклад |
| Збереження простих скалярних даних | localStorage.setItem("key", "value"); |  |
| Зчитування простих даних | localStorage.getItem("key") |  |
| Видалення даних | localStorage.removeItem("key") |  |
| Очищення | localStorage.clear() |  |
| Звертання даних як до властивостей localStorage |  | if (localStorage.clickcount) {     localStorage.clickcount = Number(localStorage.clickcount) + 1; } else {     localStorage.clickcount = 1; } document.getElementById("result").innerHTML = "You have clicked the button " + localStorage.clickcount + " time(s)."; |

Збереження об’єктів

Для збереження об’єктів необхідно перетворити їх у формат JSON. Для цього є відповідні методи

|  |  |  |
| --- | --- | --- |
| Призначення | Загальна форма | Приклад |
| Перетворення об’єкта у рядок з JSON і збереження у localStorage | JSON.stringify( об’єкт ) | var user = {  name: "Ivan",  age: 27  };  localStorage.setItem("user", JSON.stringify(user)); |
| Зчитування з localStorage і перетворення з JSON у об’єкт | JSON.parse( рядок\_json\_об’єкта ) | var user = JSON.parse(localStorage.getItem("user")); |

Приклад.

window.onload = function () {

if (localStorage.getItem("user"))

{

var user = JSON.parse(localStorage.getItem("user"));

}

else

{

var user = {

name: "Ivan",

age: 27

};

localStorage.setItem("user", JSON.stringify(user));

}

document.write(`${user.name} : ${user.age}`);

}

Використання sessionStorage

Даний тип збереження даних аналогічний до sessionStorage, але зберігає дані тільки під час однієї сесії (поки відкрито браузер, навіть при перезавантаженні сторінки)

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Збереження простих скалярних даних | sessionStorage.setItem("key", "value"); |  |
| Зчитування простих даних | sessionStorage.getItem("key") |  |
| Видалення даних | sessionStorage.removeItem("key") |  |
| Очищення | sessionStorage.clear() |  |

Приклад.

if (sessionStorage.clickcount) {  
    sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;  
} else {  
    sessionStorage.clickcount = 1;  
}  
document.getElementById("result").innerHTML = "You have clicked the button " +  
sessionStorage.clickcount + " time(s) in this session.";

Порівняння localStorage, sessionStorage, cookie

|  |  |  |
| --- | --- | --- |
| LocalStorage | SessionStorage | Cookie |
| * Stores data with no expiration date, and gets cleared only through JavaScript, or clearing the Browser cache / Locally Stored Data * Storage limit is the maximum amongst the three | * The sessionStorage object stores data only for a session, meaning that the data is stored until the browser (or tab) is closed. * Data is never transferred to the server. * Storage limit is larger than a cookie (at least 5MB). | * Stores data that has to be sent back to the server with subsequent requests. Its expiration varies based on the type and the expiration duration can be set from either server-side or client-side (normally from server-side). * Cookies are primarily for server-side reading (can also be read on client-side), localStorage and sessionStorage can only be read on client-side. * Size must be less than 4KB. * Cookies can be made secure by setting the httpOnly flag as true for that cookie. This prevents client-side access to that cookie |

Посилання

<https://www.w3schools.com/jsref/obj_storage.asp>

Storage Object

The Storage object of the Web Storage API provides access to the session storage or local storage for a particular domain. This allows you to read, add, modify, and delete stored data items.

Storage Object Properties and Methods

|  |  |
| --- | --- |
| **Property/Method** | **Description** |
| [key(*n*)](https://www.w3schools.com/jsref/met_storage_key.asp) | Returns the name of the *n*th key in the storage |
| [length](https://www.w3schools.com/jsref/prop_storage_length.asp) | Returns the number of data items stored in the Storage object |
| [getItem(*keyname*)](https://www.w3schools.com/jsref/met_storage_getitem.asp) | Returns the value of the specified key name |
| [setItem(*keyname*, *value*)](https://www.w3schools.com/jsref/met_storage_setitem.asp) | Adds that key to the storage, or update that key's value if it already exists |
| [removeItem(*keyname*)](https://www.w3schools.com/jsref/met_storage_removeitem.asp) | Removes that key from the storage |
| [clear()](https://www.w3schools.com/jsref/met_storage_clear.asp) | Empty all key out of the storage |

Related Pages for Web Storage API

|  |  |
| --- | --- |
| **Property** | **Description** |
| [window.localStorage](https://www.w3schools.com/jsref/prop_win_localstorage.asp) | Allows to save key/value pairs in a web browser. Stores the data with no expiration date |
| [window.sessionStorage](https://www.w3schools.com/jsref/prop_win_sessionstorage.asp) | Allows to save key/value pairs in a web browser. Stores the data for one session |

=========================== ПРИКЛАДИ ===================================================

Задача 1. Зберігати ім’я користувача при заході на сайт та кількість відвідувань.

Задача 2. Зберігати колір фону (можна змінювати за допомогою кнопок) та кількість змін (протягом одного сеансу і загальну кількість).

Задача 3. Зберігати у пам’яті назви товарів, які задає користувач і виводити при натисненні на відповідну кнопку