

Зберігання даних у браузері (на стороні клієнта)

<https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

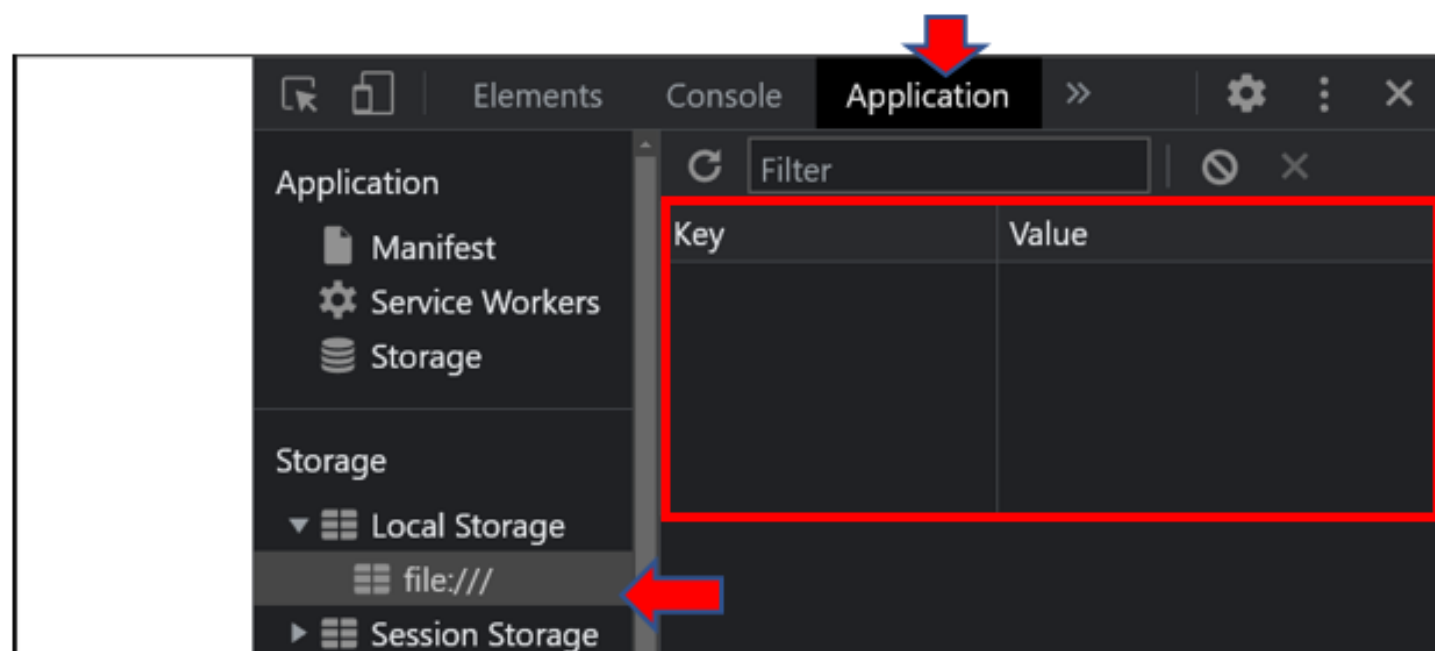
<https://uk.javascript.info/data-storage>

Основи збереження даних у Веб. Web storage API

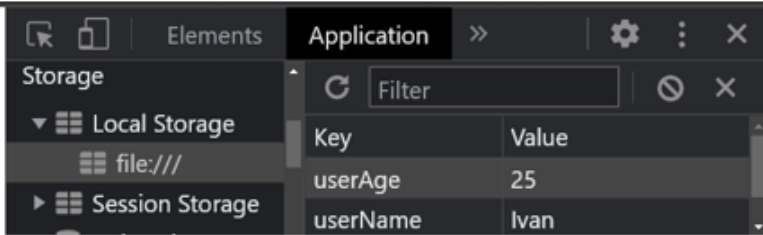
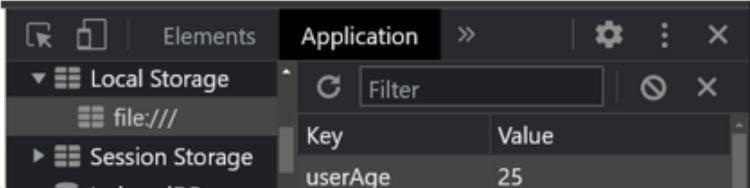
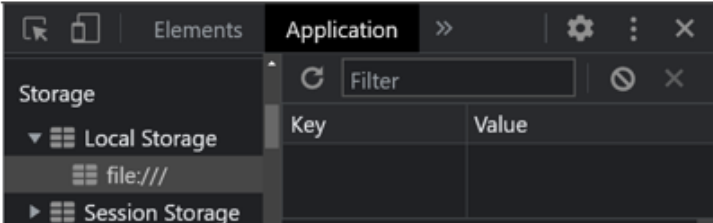
Web storage API забезпечуються механізми збереження даних на стороні клієнта.

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



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") 	<pre>localStorage.setItem('userName', 'Ivan') localStorage.setItem('userAge', 25)</pre> <pre>> localStorage < Storage {userAge: '25', userName: 'Ivan', length: 2}</pre>
Зчитування простих даних	localStorage.getItem("key")	<pre>> localStorage.getItem('userName') < 'Ivan' > localStorage.getItem('userAge') < '25'</pre> <pre>localStorage.userAge '25' localStorage.userName 'Ivan'</pre>
Визначення ключа у заданій позиції	localStorage.key(index)	<pre>> localStorage.key(0) < 'userAge' > localStorage.key(1) < 'userName'</pre>
Видалення даних	localStorage.removeItem("key")	localStorage.removeItem('userAge') 
Очищення	localStorage.clear()	localStorage.clear() 

===== 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).

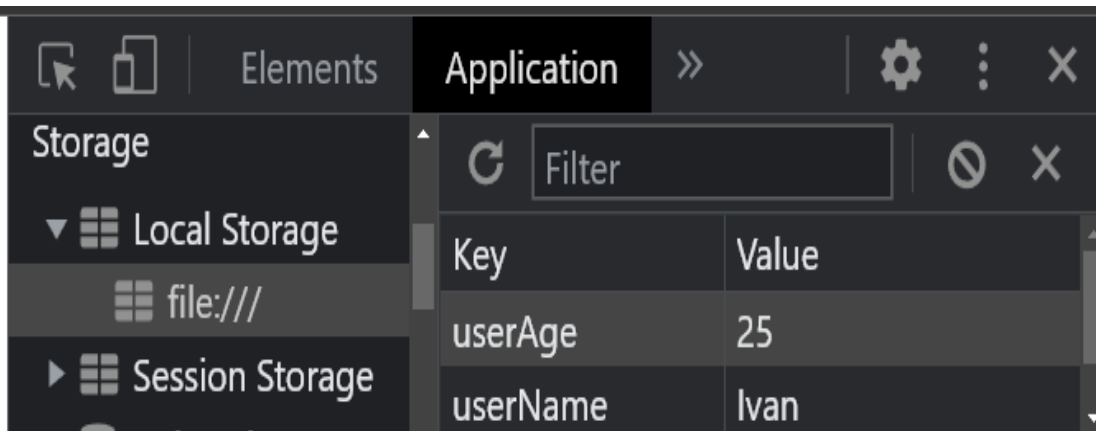
Задача 3. Зберігати у пам'яті назви товарів, які задає користувач і виводити при натисненні на відповідну кнопку

Звертання до даних як до властивостей `localStorage` (не рекомендується)

- можливі проблеми з іменами (наприклад, `length`)
- не відслідковується зміна даних (не генерується подія)

`localStorage.` властивість_користувача

```
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).";
```



```
localStorage.userAge  
'25'
```

```
localStorage.userName  
'Ivan'
```

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)	<pre>var user = { name: "Ivan", age: 27 }; localStorage.setItem("user", JSON.stringify(user));</pre>
Зчитування з localStorage і перетворення з JSON у об'єкт	JSON.parse(рядок_json_об'єкта)	<pre>var user = JSON.parse(localStorage.getItem("user"));</pre>

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}`);
}
```


Перегляд ключів localStorage

<p>З використанням властивості-кількість значень <code>localStorage.length</code></p> <p>та можливості отримання назви властивості за індексом</p> <p><code>localStorage.key(index)</code></p>	<pre>for(let i=0; i<localStorage.length; i++) { let key = localStorage.key(i); alert(`\${key}: \${localStorage.getItem(key)}`); }</pre>						
<p>З використанням циклу <code>for..in</code> перебираємо усі властивості</p> <p><code>for(let key in localStorage) { ... }</code></p> <p>та пропускаємо його стандартні властивості</p> <p><code>localStorage.key(index)</code></p>	<pre>for(let key in localStorage) { if (!localStorage.hasOwnProperty(key)) { continue; // пропускати ("setItem", "getItem",...) } alert(`\${key}: \${localStorage.getItem(key)}`); }</pre>						
<p>Отримуємо список ключів <code>Object.keys</code> у цьому списку будуть тільки ключі, що додані об'єкту, а не ключі з прототипу</p>	<pre>let keys = Object.keys(localStorage) //Отримуємо ключі for(let key of keys) { alert(`\${key}: \${localStorage.getItem(key)}`) }</pre> <div><div><div>Elements</div><div>Application</div><div>Storage</div><div>Local Storage</div><div>file:///</div><div>Session Storage</div></div><div><div>Filter</div><table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>userAge</td><td>25</td></tr><tr><td>userName</td><td>Ivan</td></tr></tbody></table></div></div> <div><code>Object.keys(localStorage)</code> <code>▶ (2) ['userAge', 'userName']</code></div>	Key	Value	userAge	25	userName	Ivan
Key	Value						
userAge	25						
userName	Ivan						

Приклад. Реалізувати ToDo list з можливістю збереження

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

Методи для роботи з `sessionStorage` аналогічні

 (Ctrl) ▾

Призначення	Загальна форма
Збереження простих скалярних даних	<code>sessionStorage.setItem("key", "value");</code>
Зчитування простих даних	<code>sessionStorage.getItem("key")</code>
Визначення ключа у заданій позиції	<code>sessionStorage.key(index)</code>
Видалення даних	<code>sessionStorage.removeItem("key")</code>
Очищення	<code>sessionStorage.clear()</code>

Приклад. Звертання до даних як до властивостей `sessionStorage` (**не рекомендується**)

```
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.";
```

Подія – зміна значення

Подія **StorageEvent** (нащадок від Event) генерується при зміні даних *localStorage* або *sessionStorage*

Деякі властивості **StorageEvent**

key	значення ключа, який було оновлено (null, якщо зміни пов'язані з викликом методу clear)
newValue	нове значення (null, якщо ключ було видалено)
oldValue	переднє значення ключа (null, якщо ключ додано вперше)
url	url документа, де відбулись зміни
storageArea	об'єкт для якого відбулись зміни (об'єкт localStorage або sessionStorage)

Додавання обробника зміни об'єктів **localStorage** або **sessionStorage**

Загальна форма	<code>window.onstorage = функція_обробник</code> або <code>window.addEventListener("storage", функція_обробник)</code>
Приклад	<code>window.onstorage = event => { alert(event.key + ':' + event.newValue + " в " + event.url) }</code>

```
<script>  
function onLoad() {  
  let locale = localStorage.getItem('lang')  
  if (!locale) {  
    locale = 'ua'  
    localStorage.setItem('lang', locale)  
  }  
  document.getElementById('lang').innerText = locale  
}  
  
function onChangeStorage(event) {  
  document.getElementById('lang').innerText = localStorage.getItem('lang')  
}  
  
function onSelectorChange() {  
  const locale = document.getElementById('localeSelector').value  
  localStorage.setItem('lang', locale)  
  window.dispatchEvent(new StorageEvent('storage', {  
    key: 'lang',  
    newValue: locale,  
  }))  
}  
  
window.onload = function () {  
  onLoad()  
  document.getElementById('localeSelector').onchange = onSelectorChange  
  window.addEventListener('storage', onChangeStorage)  
}  
</script>
```

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    <script>
      function onLoad() {
        let locale = localStorage.getItem('lang')
        if (!locale) {
          locale = 'ua'
          localStorage.setItem('lang', locale)
        }
        document.getElementById('lang').innerText = locale
      }

      function onChangeStorage(event) {
        document.getElementById('lang').innerText = localStorage.getItem('lang')
      }

      function onSelectorChange() {
        const locale = document.getElementById('localeSelector').value
        localStorage.setItem('lang', locale)
        window.dispatchEvent(new StorageEvent('storage'), {
          key: 'lang',
          newValue: locale,
        })
      }

      window.onload = function () {
        onLoad()
        document.getElementById('localeSelector').onchange = onSelectorChange
        window.addEventListener('storage', onChangeStorage)
      }
    </script>
  </head>

```

Locale: ua

Українська ▼
 Обрати мову
 Українська
 Англійська

```

<body>
  <div>Locale: <span id="lang"></span></div>

  <select id="localeSelector">
    <option value="">Обрати мову</option>
    <option value="ua">Українська</option>
    <option value="en">Англійська</option>
  </select>
</body>
</html>

```