

LocalStorage, sessionStorage

Объекты веб-хранилища **localStorage** и **sessionStorage** позволяют хранить пары ключ/значение в браузере. Что в них важно – данные, которые в них записаны, сохраняются после обновления страницы (в случае **sessionStorage**) и даже после перезапуска браузера (при использовании **localStorage**).

Доступ к данным производится по **ключу**: вы сохраняете данные с каким-то ключом, а затем можете получить их по этому ключу или удалить. При этом разрешено сохранять только **строки**. Если попытаться прочитать ключ, для которого **не были записаны данные**, то результатом будет **null**. Это можно использовать для того, чтобы выполнить однократное сохранение данных. Можно перезаписывать данные, хранящиеся под определенным ключом

Объекты хранилища **localStorage** и **sessionStorage** предоставляют одинаковые методы и свойства:

- **setItem(key, value)** – сохранить пару ключ/значение.
- **getItem(key)** – получить данные по ключу key. Если не найдено, то **null**
- **removeItem(key)** – удалить данные с ключом key.
- **clear()** – удалить всё.
- **key(index)** – получить ключ на заданной позиции.
- **length** – количество элементов в хранилище.

В браузере **Chrome** в настройках инструментов см. **Application - Storage - LocalStorage**

```
localStorage.setItem("name", "Jhon");
localStorage.setItem("age", 26);
console.log('localStorage.getItem("name"):', localStorage.getItem("name")); // localStorage.getItem("name"): Jhon
console.log('typeof localStorage.getItem("name"):', typeof localStorage.getItem("name")); // typeof localStorage.getItem("name"): string
localStorage.clear();
```

localStorage хранит в себе только **строки**:

```
const person = {
  name: "John",
  age: 26,
  profesion: "Programmer",
  isMarried: true,
};
const fruitsArray = ["apple", "banana", "plum", "cherry", "grape"];
localStorage.setItem("objectPerson", person); // [object Object]
localStorage.setItem("array", fruitsArray); // apple,banana,plum,cherry,grape
```

Поэтому для преобразования в строку **объектов** или **массивов** используют **JSON** формат:

```
// Сохранение массива
const fruitsArray = ["apple", "banana", "plum", "cherry", "grape"];
localStorage.setItem("fruits", JSON.stringify(fruitsArray)); // ["apple","banana","plum","cherry","grape"]
```

// Сохранение объекта

```
const person = {
  name: "John",
  age: 26,
  profesion: "Programmer",
  isMarried: true,
};
localStorage.setItem("person", JSON.stringify(person)); // {"name":"John","age":26,"profesion":"Programmer","isMarried":true}
```

Обратное преобразование:

// Получение массива и объекта

```
const arrayFruits = JSON.parse(localStorage.getItem("fruits"));
console.log("arrayFruits: ", arrayFruits); // arrayFruits: (5) ['apple', 'banana', 'plum', 'cherry', 'grape']
```

```
const personFromStorage = JSON.parse(localStorage.getItem("person"));
console.log("personFromStorage: ", personFromStorage); // personFromStorage: {name: 'John', age: 26, profesion: 'Programmer', isMarried: true}
```