

|  |
| --- |
| **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Уральский государственный экономический университет»**  **(УрГЭУ)** |

**Контрольная работа**

по дисциплине **«Корпоративные информационные системы»**

**Тема: Локальное хранилище LOCALSTORAGE**

|  |  |  |
| --- | --- | --- |
| Институт цифровых технологий управления и информационной безопасности  Направление подготовки  *09.03.03 Прикладная информатика в экономике*  Направленность (профиль)  *Прикладная информатика в экономике*  Кафедра  *Информационных технологий и статистики*  Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Студент  *Галашов Игорь Алексеевич*  Группа З*ПИЭ-21-1* |

Екатеринбург

2024 г.

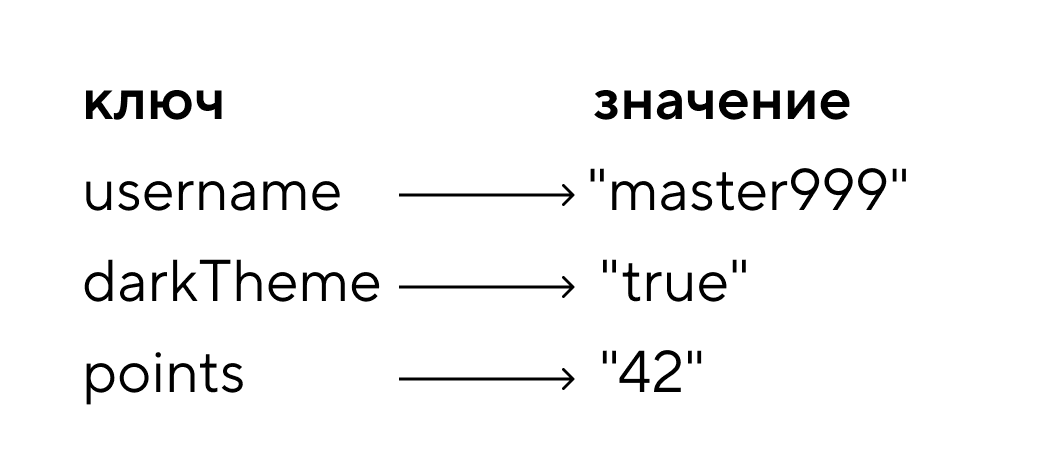
LOCALSTORAGE

localStorage — это место в браузере пользователя, в котором сайты могут сохранять разные данные. Это как ящик для хранения вещей, которые не исчезнут, даже если вы выключите компьютер или закроете браузер.

До localStorage разработчики часто использовали cookies, но они были не очень удобны: мало места и постоянная передача данных туда-сюда. LocalStorage появился, чтобы сделать процесс более простым и эффективным.

КАК ХРАНЯТСЯ ДАННЫЕ В LOCALSTORAGE

В localStorage данные хранятся в формате пары ключ-значение, причём и ключ, и значение всегда сохраняются в виде строк. Даже если вы сохраните числовое или логическое значение, оно автоматически преобразуется в строку.



Сохранение числа или булевой переменной:

localStorage.setItem('number', 123);

localStorage.setItem('boolean', true);

При извлечении в обоих случаях будет строка:

console.log(localStorage.getItem('number')); // "123"

console.log(localStorage.getItem('boolean')); // "true"

Если нужно сохранить более сложные структуры данных, например, объекты или массивы, используйте JSON.stringify() для их преобразования в строку перед сохранением и JSON.parse() для преобразования обратно после извлечения:

const user = {

name: "Alex",

age: 25

};

// Сохранение объекта в localStorage

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

// Извлечение объекта из localStorage

const retrievedUser = JSON.parse(localStorage.getItem('user'));

console.log(retrievedUser.name); // "Alex"

Таким образом, несмотря на то что в localStorage можно сохранять только строки, с помощью JSON вы можете эффективно работать с более сложными типами данных.

В КАКИХ СЛУЧАЯХ ПРИГОДИТСЯ LOCALSTORAGE

Сохранение пользовательских настроек

Например, если на сайте можно выбирать светлую или темную тему, чтобы каждый раз не переключать, можно использовать следующий код:

// Проверяем, установлена ли тема 'dark' в localStorage

if (localStorage.getItem('theme') === 'dark') {

// Если установлена темная тема, применяем ее к body документа

document.body.classList.add('dark-mode');

}

Сокращение запросов к серверу

Если не хочется каждый раз обращаться к серверу за данными, можно сохранить их в localStorage:

// Пытаемся получить данные из localStorage по ключу 'someData'

const cachedData = localStorage.getItem('someData');

// Проверяем, существуют ли данные в кэше

if (cachedData) {

// Если данные существуют, преобразуем их из строки в объект или массив и обрабатываем

processData(JSON.parse(cachedData));

} else {

// Если данных в кэше нет, делаем запрос к серверу за данными

fetchDataFromServer().then(data => {

// После получения данных сохраним их в localStorage для дальнейшего использования

localStorage.setItem('someData', JSON.stringify(data));

// Обрабатываем полученные с сервера данные

processData(data);

});

}

РАБОТА БЕЗ ИНТЕРНЕТА

Приложения, которые могут работать без интернета (например, список задач), могут использовать localStorage для хранения информации:

// Пытаемся получить список задач из localStorage. Если его нет, используем пустой массив.

const tasks = JSON.parse(localStorage.getItem('tasks')) || [];

// Функция для добавления задачи

function addTask(task) {

// Добавляем новую задачу в массив

tasks.push(task);

// Сохраняем обновленный список задач в localStorage

localStorage.setItem('tasks', JSON.stringify(tasks));

// Обновляем отображение списка задач на странице (предполагается, что функция renderTasks определена где-то еще)

renderTasks();

}

При создании сайта интернет магазин, для сохранения данных пользователя и переноса выбранного товара в корзину использовалось локальное хранилище, так-как JavaScript не работает с файловой системой пользователя. При обновлении и перезагрузки страницы данные сбрасываются, при использовании локального хранилища данные остаются.