

Локальное хранилище данных на языке программирования JavaScript

Кочитов Михаил Евгеньевич

Приамурский государственный университет им. Шолом-Алейхема

студент

Аннотация

В данной статье рассматривается объект веб-хранилища localStorage, который позволяет сохранять данные в клиенте браузера при обновлении страницы и закрытии самого браузера. Также в статье будут рассмотрены функции работы объекта localStorage, реализованные на языке программирования JavaScript.

Ключевые слова: локальное хранилище данных, JavaScript, клиент браузера, localStorage

Local data storage in the JavaScript programming language

Kochitov Mikhail Evgenevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the localStorage web storage object, which allows data to persist in the browser client when the page is refreshed and the browser itself is closed. Also, the article will consider the functions of the localStorage object, implemented in the JavaScript programming language.

Keywords: local data store, JavaScript, browser client, localStorage

Веб-сайты содержат много информации и данных, которые надо где то хранить, но возникает потребность в тех данных, которые надо сохранить, даже если был закрыт браузер. Такими данными могут являться некоторые настройки или информация об авторизованном пользователе и так далее. Объект веб-хранилища localStorage, написанный на языке программирования JavaScript позволяет хранить данные любого типа в клиенте браузера с поддержкой загрузки сохраненных данных при последующем открытии веб-браузера без использования сервера и базы данных.

В статье О.Ю. Лазаревой и Д.Д. Ложкиной рассматривается обзор графических JavaScript-библиотек для создания интерактивных электронных изданий [1]. Рассматривая статью К.А. Тезика можно увидеть защиту Web-сайтов с помощью сценариев на языке программирования JavaScript [2]. В.Н.Белов, А.И. Ковалёв и С.А. Новиков в своей статье рассмотрели реализацию элемента пользовательского интерфейса Slider с помощью JavaScript [3]. В статье К.Янчека, Е.Койчевой, Р.В. Хазанкина и

А.М.Морозова рассматривается интерпретатор скриптового языка JavaScript для симулятора обобщенных сетей [4]. Р.Жуйков, Д.Мельник, Р.Бучацкий, В.Варданян, В.Иванишин и Е.Шарыгин рассматривают в своей статье методы динамической и предварительной оптимизации программ на языке JavaScript [5].

Целью статьи является рассмотрение объекта веб-хранилища `localStorage`, который позволяет сохранять данные в клиенте браузера при обновлении страницы или закрытии браузера. Также в статье будут рассмотрены функции, позволяющие работать с локальным хранилищем данных объекта `localStorage`, который реализован на языке программирования JavaScript.

Для реализации локального хранилища данных и использования объекта `localStorage` понадобится только веб-браузер без необходимости использования локального веб-сервера, так как `localStorage` добавлен в JavaScript, поэтому с ним можно работать с любым веб-браузером. Теперь создадим сам файл JavaScript и в нем распишем код добавления данных в объект `localStorage`

```
localStorage.setItem('number', 123)
localStorage.setItem('string', "Hello")
```

Рисунок 1 – Функция `setItem()` объекта `localStorage`, позволяющая добавлять данные

На рисунке 1 представлена функция `setItem()`, позволяющая добавлять данные в объект `localStorage` для дальнейшего их хранения при обновлении веб-страницы или закрытии браузера. В саму функцию передается два параметра: первый – ключ, второе – значение. На примере функция `setItem` добавляет число 123 в ключ «number» и строку «Hello» в ключ «string».

Теперь напишем следующую функцию объекта `localStorage`, которая имеет возможность считывать сохраненные данные

```
var var1 = localStorage.getItem('number')
var var2 = localStorage.getItem('string')
alert(var1 + " " + var2)
```

Рисунок 2 – Функция `getItem()` объекта `localStorage`, позволяющая получать сохраненные данные

На рисунке 2 продемонстрирована функция `getItem()`, позволяющая получать сохраненные данные. В переменные `var1` и `var2` заносятся значения из ключей «number» и «string» объекта `localStorage`, которые были добавлены предыдущей функцией `setItem()` (См. Рис. 1). Функция `alert()` дает возможность отобразить в самом браузере всплывающее окно с содержимым

этих переменных `var1` и `var2`, то есть полученных сохраненных данных объекта `localStorage`.

Далее в браузере проверим результат получения сохраненных данных из локального хранилища `localStorage` во всплывающем окне.



Рисунок 3 – Всплывающее окно с отображением полученных данных из объекта `localStorage`

На рисунке 3 продемонстрировано всплывающее окно, вызванное функцией `alert()`, в котором содержится информация о полученных данных из объекта `localStorage`, которые были сохранены.

Теперь рассмотрим функцию удаления определенных данных из объекта `localStorage`

```
localStorage.removeItem('string')
```

Рисунок 4 – Функция `removeItem()`, которая удаляет указанные данные из объекта `localStorage`

На рисунке 4 показана функция `removeItem()`, позволяющая удалять определенные данные из локального хранилища `localStorage`. В функцию передается один параметр это ключ, который был ранее добавлен. В данном случае из объекта `localStorage` производим удаление ключа «string» с записанным в него значением «Hello».

Далее напишем следующую функцию объекта `localStorage`, которая показывает количество добавленных данных в собственно само локальное хранилище данных `localStorage`

```
var count = localStorage.length  
alert("Count = " + count)
```

Рисунок 5 – Функция `length`, получающая количество добавленных и сохраненных данных объекта `localStorage`

На рисунке 5 показана функция `length`, которая заносит в переменную `count` текущее количество добавленных сохраненных данных объекта `localStorage` и вызывает с помощью функция `alert()` всплывающее окно в браузере

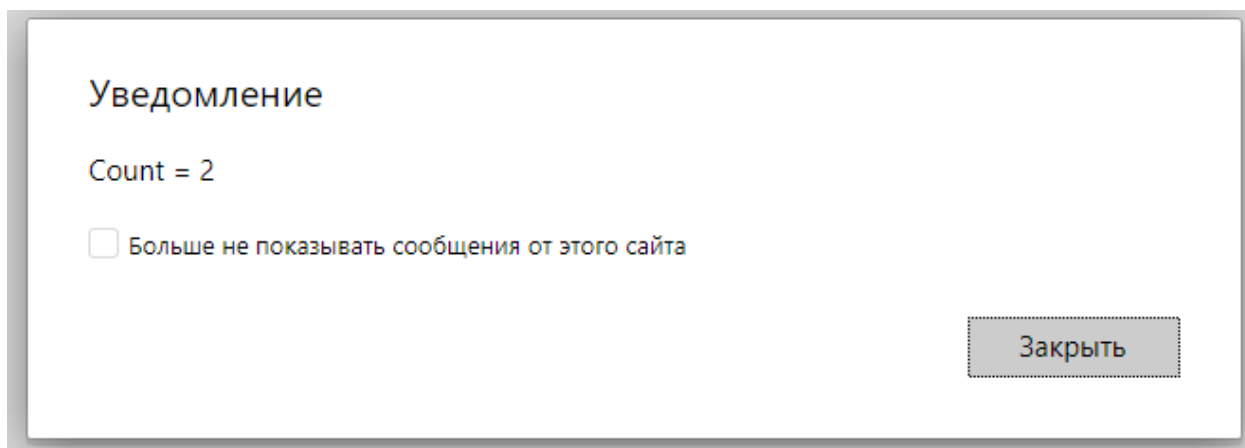


Рисунок 6 – Всплывающее окно с отображением количества добавленных данных объекта `localStorage`

На рисунке 6 изображено всплывающее окно, в котором содержится информация о количестве добавленных данных объекта `localStorage`. В данном случае их 2, так как в `localStorage` содержатся данные: «number» и «string».

Далее рассмотрим и напишем последнюю функцию объекта `localStorage`, которая полностью очищает все данные из объекта `localStorage` и делает его пустым

```
localStorage.clear()
```

Рисунок 7 – Функция `clear()`, очищающая все данные из объекта `localStorage`

На рисунке 7 показана функция `clear()`, которая полностью удаляет все сохраненные данные из локального хранилища объекта `localStorage`. Данная функция используется в том случае, когда уже не будет необходимости хранить данные в локальном хранилище браузера.

Таким образом, был рассмотрен объект веб-хранилища `localStorage`, позволяющий хранить данные в клиенте браузера при обновлении страницы и закрытии самого браузера. Также были рассмотрены функции объекта `localStorage` для работы с локальным хранилищем данных на языке программирования JavaScript.

Библиографический список

1. Лазарева О.Ю., Ложкина Д.Д. Обзор графических JavaScript-библиотек

- для создания интерактивных электронных изданий // Вестник МГУП имени Ивана Федорова. 2016. № 1. С. 76-78.
2. Тезик К.А. Защита Web-сайтов с помощью сценариев на языке программирования JavaScript // В сборнике: Инфокоммуникации и информационная безопасность: состояние, проблемы и пути решения. Сборник научных статей по материалам III Всероссийской научно-практической конференции: в 2 частях. Ответственный редактор: В.Г. Андронов. 2016. С. 167-172.
 3. Белов В.Н., Ковалёв А.И., Новиков С.А. Реализация элемента пользовательского интерфейса slider с помощью JavaScript // Современные наукоемкие технологии. 2016. № 10-2. С. 224-228.
 4. Янчек К., Койчева Е., Хазанкин Р.В., Морозов А.М. Интерпретатор скриптового языка JavaScript для симулятора обобщенных сетей // Вестник Уфимского государственного авиационного технического университета. 2009. Т. 13. № 2. С. 160-166.
 5. Жуйков Р., Мельник Д., Бучацкий Р., Варданын В., Иванишин В., Шарыгин Е. Методы динамической и предварительной оптимизации программ на языке JavaScript // Труды Института системного программирования РАН. 2014. Т. 26. № 1. С. 297-314.
 6. Работа с localStorage на примерах URL: <https://myrusakov.ru/js-localstorage.html> (дата обращения 31.01.2021)