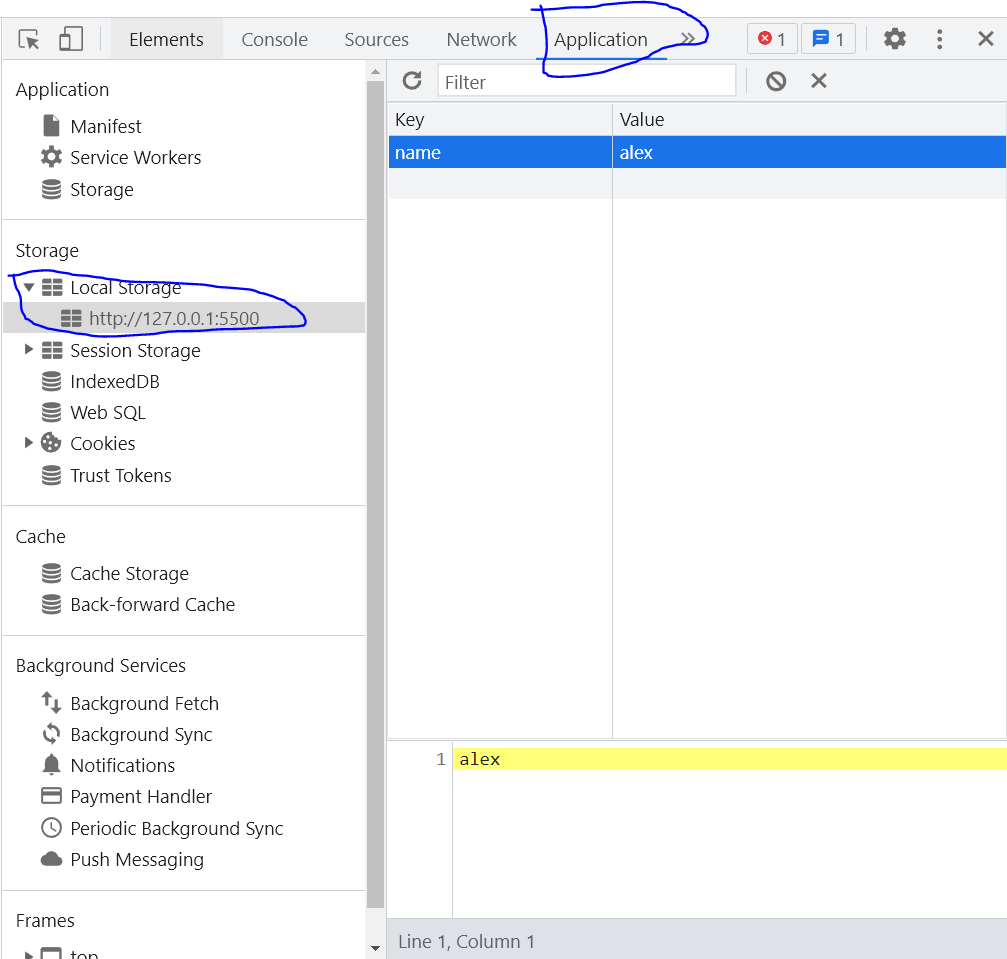
**Как сохранять данные в браузере или краткий гайд по LocalStorage**JavaScript\*Sandbox

Привет! В этой статье я бы хотел поговорить про LocalStorage, его методы и особенности. Статью можно посмотреть в формате видео по этой [ссылке](https://youtu.be/6VtL9SHRwCk). Ну что, приступим?И так, что же это такое. LocalStorage это свойство объекта window, предназначенное для хранения пар ключ/значение в браузере. В зависимости от браузера, мы можем сохранять до 5 мб данных. Но тогда возникает логичный вопрос, зачем нам localStorage, если данные мы можем сохранять данные в обычных переменных? Ответ очень прост, все что мы запишим в localStorage, останется там после перезагрузки страницы и даже после закрытия браузера. Теперь поговорим о том как мы можем взаимодействовать с localStorage.

**Запись в LocalStorage**

Чтобы сохранить данные можем использовать метод setItem(key, value) у localStorage. Этод метод принимает два аргумента: ключ по которому мы будем сохранять информацию и само значение, которое мы хотим сохранить. Например:

localStorage.setItem('name', 'Alex');

Для просмотра наших записей надо открыть DevTools там найти вкладку Application и там обязательно будет LocalStorage**Чтение из LocalStorage**

Теперь конечно же хочется получить эти данные. Для этого воспользуемся методом getItem(key). Результат можно вывести в консоль, или сохранить в переменную.console.log(localStorage.getItem('name'));

**Только строки**

Как видите, все просто, но есть один маленький нюанс. Значение, которое мы записываем должно быть строкой. Давайте попробуем добавить поле age со значением 5.

Для этого напишем:localStorage.setItem('age', 5);

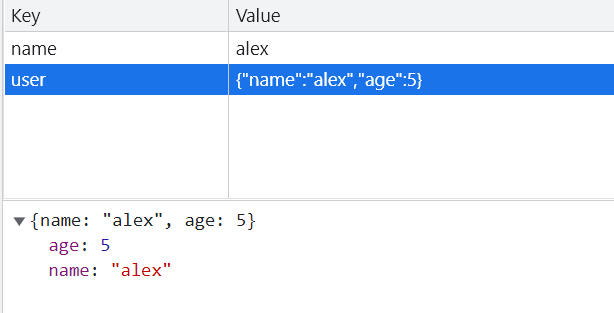
Сохраняем файл. И что мы видим? Ошибок никаких нет, а если перейти во вкладку Application, то мы увидим что запись прошла успешно. Это конечно же так, но при записи пятерка стала строкой, то есть была автоматически приведена к типу string. Это не то чтобы плохо, но может привести к неожиданному поведению, к примеру при строгом сравнении. Я бы советовал привести строку к числу, это можно сделать вот так:const age = +localStorage.getItem('age');

Если с примитивами все просто, то что на счет объектов? Создадим объект user и попробуем записать его в LocalStorage:

const user = {  
 name: 'alex',   
 age: 5  
};  
  
localStorage.setItem('user', user);

Переходим во вкладку Application и видим [objectObject].И есть одна небольшая проблемка у этой строки не будет полей, которые мы определили в объекте user. Это просто обычная строка. И в исходный вид это уже никак не вернуть. Для того, чтобы записать объект в localStorage надо сделать его строкой. У нас как раз есть прекрасный формат под это, и имя его json. Для преобразования объекта в строку нужно использовать JSON.stringify:

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

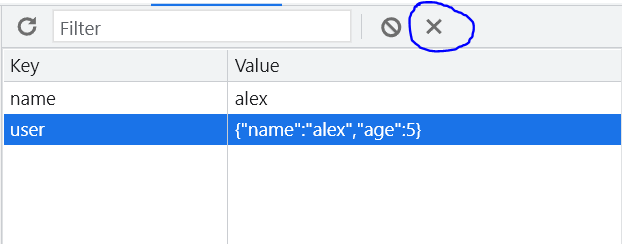
Теперь все записалось корректно:Чтобы получить объект в первозданном состоянии используем метод

parse у JSON:

JSON.parse(localStorage.getItem('user'));

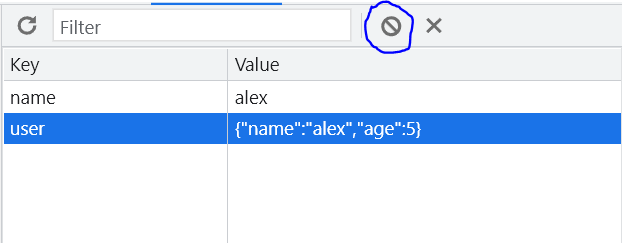
Полученый результат можем записать в консоль, или сохранить в переменную.

**Удаление из LocalStorage**

Мы научились записывать и получать элементы, теперь надо научится их удалять. Для этого будем использовать метод  removeItem(key) который удалит поле с ключом key.localStorage.removeItem('user');Так же во вкладке Application мы можем выделить нужный нам рядок и нажать на крестик, что тоже приведет к удалению.**Очистка LocalStorage**

Чтобы очистить все поля мы можем использовать метод clear():

localStorage.clear();

Вариант из Application:Я рекомендую так делать, только тогда, когда вы на 200% уверены что все данные заносили только вы, и после удаления чего либо ваш сайт не начнет вести себя некорректно.

**Событие storage**

Если у вас есть потребность, или желание прослушивать изменение в localStorage, то можно просто добавить слушатель событий на 'storage':

window.addEventListener('storage', (event) => console.log(event));

При каждом изменении localStorage генерируется событие storage. Оно отрабатывает на всех вкладках, на которых открыт наш сайт, кроме той, где мы создали это событие. В event хранятся название поля которое мы затронули, старое значение, новое значение, url документа и [storageArea](https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/storage/StorageArea). Если интересно, крайне рекомендую переписать этот кусок кода себе, открыть две вкладки вашего локалхоста и посмотреть как это работает.Также, надо держать в голове что самый "топовый" браузер, то бишь IE вызывает событие 'storage' даже если данные не изменились, что является некорректным поведением. Поэтому если вам не пофиг на екплоер и вы не хотите каких-то нежданчиков, то надо явно сравнивать новое и старое значение, которые мы можем получить из event.

**Итоги**

Особенности: Благодаря localStorage мы можем хранить пары ключ - значение в браузере

Данные остаются в хранилище даже после выключения компьютера

Данные привязаны к конкретному браузеру и источнику

И ключ и значения должны быть строками

Методы:

setItem(key, value) – записать пару ключ/значение

getItem(key) – получить данные по ключу

keyremoveItem(key) – удалить значение по ключу

keyclear() - очистить хранилище

Tags:[javascript](https://habr.com/en/search/?target_type=posts&order=relevance&q=%5Bjavascript%5D)[localstorage](https://habr.com/en/search/?target_type=posts&order=relevance&q=%5Blocalstorage%5D)Hubs:[JavaScript](https://habr.com/en/hub/javascript/)