

# STAFF - BLOG POST

## ブラウザにデータを保存する localStorage（ローカルストレージ）の使 い方

Posted by [NAGAYA \(/blog/staff\)](#) on Oct 26th, 2017



[\(/blog/staff/local-storage-01\)](#)

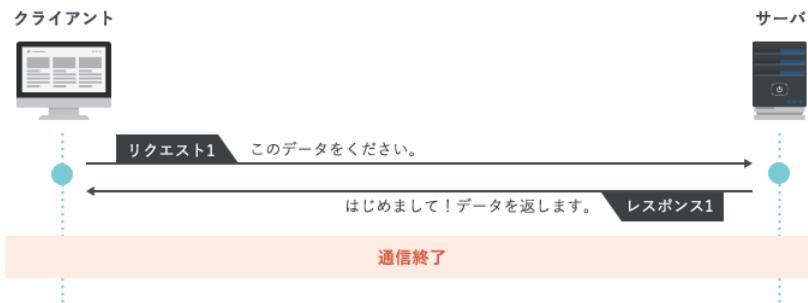
今回お届けするのは、「localStorage（ローカルストレージ）」について。  
HTML5から導入されたAPI、**Web Storage** の一種で、その仕組みはcookieによく似ています。

Web Storageと一体どのような違いがあるのか、今回はその概要と使い方を見ていきましょう。

### ブラウザにデータを保有してどうするの？

cookieもWeb Storageも **クライアント側（ブラウザ）にデータを保有する機能** ですが、なぜそのような仕組みが登場したのでしょうか？  
その答えは、HTTPが「**ステートレスな（=状態をもたない）プロトコル**」であることに関連しています。少しだけおさらいしてみましょう。

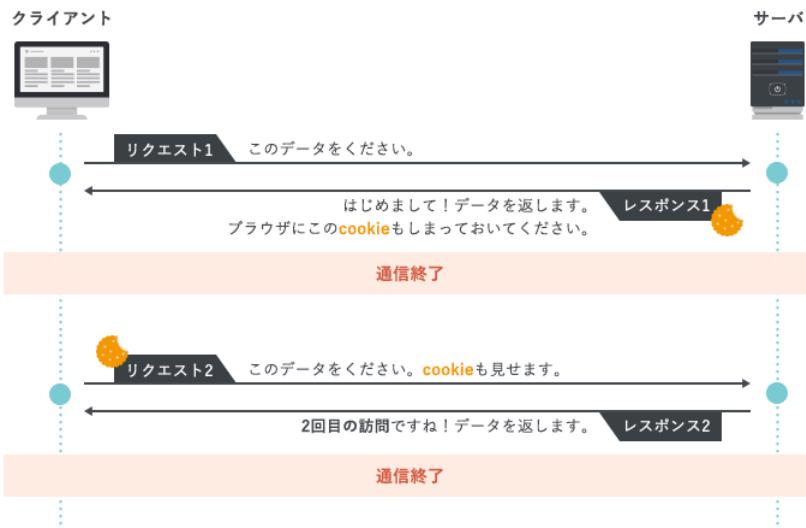
HTTP通信は「リクエストを投げる→レスポンスを返す」の一往復のやりとりで完結します。



ブラウザにデータを保存するlocalStorage（ローカルストレージ）の使い方 | 株式会社グランフェアズ  
全通信のひとつひとつが全く別のリクエストとして扱われるため、例えば **再訪問かどうか** を判断することは不可能で、同じクライアントからリクエストされたとしてもサーバ側にとっては「はじめてまして」の扱いになってしまいます。

これを解決するために従来から使われていたのが **cookie**。  
サーバ側がレスポンスと一緒に、入力情報や訪問回数などの情報をブラウザに保存させます。

そしてクライアント側が2回目以降のリクエストを行った場合には、このcookieも一緒に送信されます。サーバ側はこれを参照することで **状態に応じたレスポンスが可能になる** のです。



※ **cookie**はクライアント側でいくらでも変更可能

なので、ログイン情報などの機微な情報を取り扱う際には  
サーバ側でもつ「セッション」データと組みわるなど、セキュリティに考慮して実装されています。

## WEB STORAGEとCOOKIEの比較

利用目的がわかったところで、本題Web Storageのお話に戻りましょう。

先述のとおり、**Webサーバからクライアント側（ブラウザ）にデータを保有できる** 点はWeb Storageもcookieも同じです。その他のおおまかな共通点と相違点は下記のとおり。

### 共通点

- ◎ クライアント側で自由に書き換え可能
- ◎ 保存するデータは「キー」と「値」のペアで取り扱われる

### 相違点（Web Storageの特徴）

- ◎ Web Storageのほうが 取り扱えるデータ量が大きい
- ◎ Web Storageは JavaScriptによる制御がより簡単
- ◎ cookieは通信毎に自動送信・Web Storageは必要なデータを必要なときだけ送信

相違点3つ目「データ送信のタイミング」は、ケースにより使い分けられます。

例えば「初回アクセス時だけモーダルを表示させる」という処理をするとき、判定フラグをcookieに持たせると 以降の関係ない通信でもサーバとの間で判定フラグのやり取りが発生してしまいます。  
このような処理には、制御の簡単さからもWeb Storageに軍配があがると言えそうです。

また、Web Storageはさらに **localStorage** と **sessionStorage** の2種類に分かれ、**保持するデータの有効期限** に違いがあります。

ここまで内容を踏まえ、「cookie」「localStorage」「sessionStorage」の比較表を改めて記しました。目的に応じて利用しましょう。

有効期限	データ量	サーバへのデータ送信
<b>localStorage</b>	永続的に有効	データ利用時のみ
<b>sessionStorage</b>	ウィンドウ・タブを閉じるまで	データ利用時のみ

有効期限	データ量	サーバへのデータ送信
Cookie	指定期限まで	4KB リクエスト毎に自動送信

## LOCALSTORAGEを使ってみよう

ブラウザの対応状況は「Can I use...」を参照してみましょう。

最近のブラウザにはほぼ対応しており、IEも8以降であれば難なく利用できます。

[Can I use... Support tables for HTML5, CSS3, etc \(https://caniuse.com/#search=localStorage\)](https://caniuse.com/#search=localStorage).

### localStorageの利用例をご紹介



#### 選択内容の保存

トップページで選択した内容を保存しておき、遷移後の条件分岐処理に活用します。

☆今回グランフェアズの案件でlocalStorageを使ったのがこのような目的によるものでした。



#### サイト設定の保存

「[365 FLOWERS \(https://365flowers.kirin.co.jp/\)](https://365flowers.kirin.co.jp/)」では、音楽のミュート設定の保持にlocalStorageが使われています。



#### メモやtodoリストの作成

配列でデータを取り扱えば、こんな簡単なWebアプリらしきものも作れます！本記事を書くにあたり参考にしたいくつかの技術ブログでも、このようなサンプルが多く見られました。

参考：[ローカルストレージ\(LocalStorage\)の使い方をサンプル付きで説明 | Webliker \(http://webliker.info/how-to-use-localstorage/#LocalStorage\)](http://webliker.info/how-to-use-localstorage/#LocalStorage)

## データの保存

localStorageへのデータ保存はたったこれだけ。 `localStorage.setItem()` を使います。

```
localStorage.setItem('キー', '値');
```

### 配列を取り扱う場合

ひとつのキーに対して複数の値を保存したい場合に使うのがおなじみの「配列」。

ただしlocalStorageは 文字列形式で値が格納される ので、保存前に **JSON形式に変換** します。

```
var array = [];
var obj = {
  'キー1': '値1',
  'キー2': '値2'
};
array.push(obj);

var setjson = JSON.stringify(obj);
localStorage.setItem('キー', setjson);
```

これは、先述の[localStorageの利用例](#)に挙げた「メモやtodoリストの作成」にも使われています。

## データの取得

データの取得は `localStorage.getItem()` を使います。

```
localStorage.getItem('キー');
```

### 配列を取り扱う場合

JSON形式で取得した値は `JSON.parse()` してJavaScriptで使えるオブジェクトに戻しましょう。

```
var getjson = localStorage.getItem('キー');
var obj = JSON.parse(getjson);

// obj['キー1']などで取り出し可能に
```

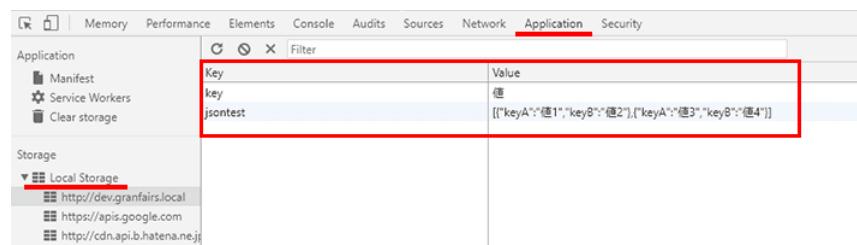
### データの削除

```
// キーを指定して削除
localStorage.removeItem('キー');

// 初期化
localStorage.clear()
```

### localStorageの内容をChrome上で確認する

Chromeでは、デベロッパーツールの「Application」>Storage内「Local Storage」から該当のURLを選択することで、そのWebサイトで利用されているlocalStorageのデータを確認することができます。



(おまけ)

グランフェアズサイトがちょっとだけ秋仕様になりました！[トップページ](#)と[ブログ](#)に期間限定で紅葉が降っています。

【2017/12/1追記】現在は終了しています。



おまけで、何を降らせようかな？と他にも色々試行錯誤していたものをここに供養しておきます。切り替えて遊んでください(ง'̀-'́)ง

▼選択中のオブジェクトは【なし】です



実はこれにはlocalStorageを簡単に使用しています。ボタンをクリック後オブジェクトのIDが保持されるので、リロードしたりブラウザを開き直したりしても選んだオブジェクトを降らせられますよ！

以上、localStorageのご紹介でした。

プラグイン「JQuery-Snowfall」の使い方はこちらの記事で紹介しています↓  
[Merry Christmas☆JS初心者がグランフェアズサイトに雪を降らせてみた！\(/blog/staff/js-snowfall\)](#).

ツイート

シェア 8

29

[◀ PREVIOUS POST \(/BLOG/STAFF/FS-COMMON-LAYOUT\)](#)  NAGAYA (/BLOG/STAFF)  [\(/BLOG/STAFF/GA-COPY-VIEW\)](#)

## LATEST POST IN THIS CATEGORY

- Jan 23rd, 2020 [【Axureで王ック作り】デバイスごとに表示を切り替える \(/blog/staff/axure-device-adaptive\)](#)
- Dec 26th, 2019 [初めましてのご挨拶とスマートオブジェクトについて \(/blog/staff/photoshop-smartobject\)](#)
- Sep 19th, 2019 [旦帰り社員旅行 ラフティング＆BBQの段【ヤマトブルブルの巻】 \(/blog/staff/company-trip-2019\)](#)
- Jul 25th, 2019 [Webフォントをサーバにアップロードして使うには？注意点や実装方法まとめ \(/blog/staff/webfont-by-selfhosting\)](#)
- Jun 27th, 2019 [入社3ヶ月。今までのこと、これから の目標 \(/blog/staff/history-and-goals\)](#)

## CATEGORY

-  [BLOG TOP \(/BLOG\)](#)
-  [NEWS \(/BLOG/NEWS\)](#)
-  [CEO - BLOG \(/BLOG/CEO\)](#)
-  [867 - BLOG \(/BLOG/CTO\)](#)
-  STAFF - BLOG (/BLOG/STAFF)

SITE SEARCH

 LET'S TALK! (/CONTACT/)



株式会社グランフェアズ

〒461-0004 愛知県名古屋市東区葵1-26-8 葵ビル5F

 MAP

TEL : 052-990-6908

&gt; 会社概要 (/about/overview/) &gt; 個人情報保護方針 (/privacy/)

 (<https://www.facebook.com/granfairs>)

Copyright © GRANFAIRS inc. All Rights Reserved.