

ステップ1: localStorage基本操作の体験（localStorage API & JSON変換）（穴埋め版）

導入・問題提起

現在のタスク管理アプリでは、タスクを追加してもブラウザを更新すると全てのタスクが消えてしまいます。これは、データがコンピュータのメモリ上に一時的にしか保存されていないためです。

この問題を解決するために、ブラウザの `localStorage` という機能を使って、データを永続的に保存する方法を学びます。

準備：ベースアプリの動作確認

まずは、演習用ベースコードが正しく動作することを確認しましょう。

- フォルダを開く**： 演習用ベースコード フォルダをVS Codeで開いてください
- Live Serverで起動**： `index.html` を右クリック → 「Open with Live Server」
- 動作確認**：
 - タスクを数個追加してみる
 - チェックボックスをクリックして完了状態を変更
 - 削除ボタンで削除
 - ブラウザを更新 (F5)** → データが消えることを確認

“ **なぜデータが消えるのか？** 現在のアプリは、データをJavaScriptの変数（メモリ）にのみ保存しています。ブラウザを更新すると、メモリがリセットされるため、データが失われます。

Before（今の状況）

- タスクを追加しても、ブラウザを更新すると消えてしまう
- データがメモリ上にしか存在しない一時的な状態

このステップの目標

- localStorage APIの基本操作（ `setItem` , `getItem` , `removeItem` , `clear` ）が理解できる
- ブラウザの開発者ツールでlocalStorageを確認・操作できる
- `JSON.stringify()` と `JSON.parse()` の基本的な使い方が分かる

考えてみよう

手順1: 開発者ツールを開く

- ブラウザで **F12キー** を押すか、右クリック → 「検証」
- Application** タブ（Chromeの場合）または **Storage** タブ（Firefoxの場合）をクリック
- 左側のメニューから **Local Storage** → あなたのサイトのURL（例： `http://127.0.0.1:5500` ）をクリック

手順2: localStorageの基本操作を体験

Console タブに移動して、以下のコードを順番に実行してみましょう：

基本操作

指示：以下の穴埋めを完成させて、コンソールで実行し、各操作後にApplicationタブでlocalStorageの内容を確認してください。

```
// 1. データを保存する - "userName"というキーで"山田太郎"という値を保存
localStorage.____(____, ____);

// 2. データを取得する - "userName"キーの値を取得
const userName = localStorage.____(____);
console.log("保存されたユーザー名:", userName);

// 3. データを削除する - "userName"キーのデータを削除
localStorage.____(____);

// 4. 全てのデータを削除する - localStorage内の全データを削除
localStorage.____();
```

より詳しく：JSON形式での保存

localStorageは **文字列** しか保存できません。オブジェクトや配列を保存するには、JSON形式に変換する必要があります。

JSON操作

指示：タスクオブジェクトをlocalStorageに保存・復元してみましょう

```
// タスクオブジェクトを作成
const task = {
  id: 1,
  text: "localStorage学習",
  completed: false
};

// 1. オブジェクトをJSON文字列に変換し、"sampleTask"キーで保存
const taskJson = JSON.____(task);
localStorage.setItem("sampleTask", ____);

// 2. "sampleTask"キーでJSON文字列を取得し、オブジェクトに復元
const savedTaskJson = localStorage.____("sampleTask");
const restoredTask = JSON.____(savedTaskJson);

console.log("復元されたタスク:", restoredTask);
console.log("タスクのテキスト:", restoredTask.____);
console.log("完了状態:", restoredTask.____);
```

✨ 最小限ヒント

- localStorage APIは `localStorage.method()` の形で使用します
- データの保存： `____(キー, 値)`
- データの取得： `____(キー)`
- データの削除： `____(キー)`
- 全削除： `____()` （引数なし）
- JSON変換： `JSON.____(オブジェクト)` → 文字列に変換
- JSON復元： `JSON.____(文字列)` → オブジェクトに復元
- オブジェクトのプロパティアクセス： `オブジェクト.プロパティ名`

動作確認チェックリスト

基本操作の確認：

- ☐ 開発者ツールのConsoleタブで `localStorage.setItem("testKey", "testValue")` を実行後、ApplicationタブのLocal Storageに `testKey` と `testValue` が表示される。
- ☐ Consoleタブで `localStorage.getItem("testKey")` を実行すると、`"testValue"` が返される。
- ☐ Consoleタブで `localStorage.removeItem("testKey")` を実行後、ApplicationタブのLocal Storageから `testKey` が削除される。
- ☐ 別のキーと値（例： `localStorage.setItem("anotherKey", "anotherValue")` ）を保存後、Consoleタブで `localStorage.clear()` を実行すると、ApplicationタブのLocal Storageが空になる。

JSON操作の確認：

- ☐ 以下のコードをConsoleで実行した際、エラーなく期待通りに動作する。

```
const myObject = { name: "test", value: 123 };
localStorage.setItem("testObject", JSON.stringify(myObject));
const retrievedObject = JSON.parse(localStorage.getItem("testObject"));
console.log(retrievedObject.name); // "test" が表示される
console.log(retrievedObject.value); // 123 が表示される
```

- ☐ 「考えてみよう（基本操作）」のコードを実行し、各ステップでApplicationタブのLocal Storageの内容が期待通りに変化することを確認する。
- ☐ 「考えてみよう（JSON操作）」のコードを実行し、コンソールに復元されたタスクオブジェクトの情報が正しく表示されることを確認する。

解答例

- ▶ 解答例（完全なコード）

ポイント解説

| localStorage APIの特徴

- **永続性** : ブラウザを閉じて、PCを再起動してもデータが保持される
- **オリジン単位** : `http://localhost:5500` と `https://example.com` は別々のストレージ
- **容量制限** : 通常5MB程度（ブラウザにより異なる）
- **同期API** : データの読み書きが即座に完了（大量データは注意）

JSON変換の重要性

- localStorageは文字列のみ保存可能
- オブジェクトや配列は `JSON.stringify()` で文字列化が必要
- 取得時は `JSON.parse()` でオブジェクトに復元
- エラーハンドリング（次のステップで学習）も重要

補足資料への誘導

- 「補足資料1: localStorage API詳解」で、より深い仕様や他のストレージとの比較を段階的に学べます。

次のステップへ

ステップ1で学んだlocalStorage APIの基本操作を使って、実際のタスク管理アプリに **自動保存機能** を実装していきましょう！