

Guía práctica: Uso de LocalStorage con tarjetas dinámicas

¿Qué es LocalStorage?

LocalStorage es una característica de HTML5 que permite almacenar datos de manera local en el navegador del usuario. Los datos almacenados persisten incluso después de cerrar el navegador, a diferencia de sessionStorage.

Es muy útil para guardar configuraciones, formularios, preferencias del usuario o información dinámica como en este ejemplo.

Ejemplo: Aplicación de tarjetas dinámicas

Vamos a crear una pequeña aplicación donde el usuario puede crear 'artículos' o 'tarjetas' que se almacenan localmente. Cada tarjeta contiene un título, subtítulo y descripción. Los datos se guardan en LocalStorage y se muestran automáticamente al recargar la página.

HTML del formulario y contenedor

```
<form id="formulario">
  <input type="text" id="inputTitulo" placeholder="Título">
  <input type="text" id="inputSubtitulo" placeholder="Subtítulo">
  <textarea id="textarea" placeholder="Descripción"></textarea>
  <button id="btnCrear">Crear</button>
</form>
<div id="contenedorTarjetas"></div>
```

JavaScript completo

Este es el código JavaScript que permite agregar, mostrar y eliminar tarjetas, utilizando LocalStorage.

```
//Traemos el form
let titulo = document.querySelector("#inputTitulo");
let subtitulo = document.querySelector("#inputSubtitulo");
let descripcion = document.querySelector("#textarea");
let btn = document.querySelector("#btnCrear");

//traemos la tarjeta
let contenedorTarjetas = document.querySelector("#contenedorTarjetas");
```

Guía práctica: Uso de LocalStorage con tarjetas dinámicas

```
// Recuperar si ya hay artículos guardados
let articulos = JSON.parse(localStorage.getItem("articulos")) || [];

btn.addEventListener("click", function(event){
    event.preventDefault();

    if(typeof (Storage) !== "undefined"){
        console.log("Local storage ready")
    } else {
        alert("No anda el Local Storage");
    }

    let nuevoArticulo = {
        titulo: titulo.value,
        subtitulo: subtitulo.value,
        descripcion: descripcion.value
    };

    if(titulo.value === "" || subtitulo.value === "" || descripcion.value === ""){
        alert("Debes completar todos los campos")
    } else {
        articulos.push(nuevoArticulo);
        localStorage.setItem("articulos", JSON.stringify(articulos));
    }

    mostrarArticulos();

    titulo.value = "";
    subtitulo.value = "";
    descripcion.value = "";
});

function mostrarArticulos() {
    contenedorTarjetas.innerHTML = "";
    articulos.forEach((articulo, index) => {
        let tarjeta = document.createElement("div");
        tarjeta.classList.add("layout__card");
        tarjeta.innerHTML = `
            <div class="header">
                <h2 class="card__subtitle">\${articulo.subtitulo}</h2>
            </div>
            <h1 class="card__title">\${articulo.titulo}</h1>
            <p class="card__text">\${articulo.descripcion}</p>
            <div class="botones">
                <button class="card__btn-change">Ver más</button>
                <button class="card__btn-delete" data-index="\${index}">Eliminar</button>
            </div>
        `;
        contenedorTarjetas.appendChild(tarjeta);
    });
}
```

Guía práctica: Uso de LocalStorage con tarjetas dinámicas

```
let botonesEliminar = document.querySelectorAll(".card__btn-delete");
botonesEliminar.forEach(boton => {
  boton.addEventListener("click", () => {
    let index = boton.getAttribute("data-index");
    articulos.splice(index, 1);
    localStorage.setItem("articulos", JSON.stringify(articulos));
    mostrarArticulos();
  });
});

window.addEventListener("DOMContentLoaded", mostrarArticulos);
```

Resumen

- Usamos `localStorage.setItem()` y `getItem()` para guardar y recuperar datos.
- Convertimos objetos con `JSON.stringify()` y `JSON.parse()`.
- Usamos `DOMContentLoaded` para mostrar los datos al cargar.
- Cada tarjeta puede eliminarse, actualizando el LocalStorage.

Este patrón se puede aplicar a muchas situaciones: tareas pendientes, notas, preferencias, etc.