

Laboratorio Práctico: Desarrollar una Aplicación para Crear un Sistema de Gestión de Libros Usando Objetos

Tiempo estimado necesario: 30 minutos

Lo que aprenderás

En este laboratorio, explorarás conceptos como la recolección de datos del usuario a través de formularios HTML, el manejo de datos de entrada utilizando objetos de JavaScript, la manipulación de arreglos para gestionar datos de libros, la actualización dinámica para mostrar libros e implementar funcionalidades como agregar, editar y eliminar entradas de libros. Este ejercicio práctico proporciona a los aprendices una visión de conceptos fundamentales como métodos de arreglos, manejo de formularios e interacción del usuario, formando una comprensión básica de los principios del desarrollo web.

Objetivos de aprendizaje

Después de completar este laboratorio, podrás:

- **Implementar una interfaz de gestión de libros:** Crear una interfaz para recoger detalles del libro, incluyendo nombre, autor, descripción y número de páginas utilizando elementos de formulario HTML.
- **Almacenar y gestionar datos de libros:** Utilizar arreglos y objetos de JavaScript para almacenar y gestionar la información de los libros ingresada por los usuarios, permitiendo funcionalidades como agregar, editar y eliminar entradas de libros.
- **Visualización dinámica de la información del libro:** Actualizar dinámicamente la página web mostrando los detalles del libro ingresados en un formato estructurado, permitiendo a los usuarios ver una lista de libros añadidos.
- **Experiencia de usuario interactiva:** Facilitar la interacción del usuario permitiendo ediciones y eliminaciones de entradas de libros directamente desde la lista mostrada, mejorando la usabilidad del sistema de gestión de libros.

Requisitos previos

- Conocimientos básicos de HTML y comandos de Git.
- Comprensión básica de las variables de JavaScript y su alcance.
- Navegador web con consola (Chrome DevTools, Consola de Firefox, etc.).

Paso 1: Configuración del entorno

1. Primero, necesitas clonar tu repositorio principal en el **Entorno de Skills Network** que creaste en el primer laboratorio y donde has subido todos los archivos y carpetas de tus laboratorios anteriores. Sigue los pasos indicados:
 - o Para esto, haz clic en el terminal en la esquina superior derecha y luego selecciona **Nuevo Terminal**.

- o Realiza el comando `git clone` escribiendo el comando dado en el terminal.

```
git clone <github-repository-url>
```

Nota: Coloca tu propio enlace de repositorio de GitHub en lugar de `<github-repository-url>`.

- o El paso anterior clonará la carpeta de tu repositorio de GitHub dentro de la carpeta del proyecto en el explorador. También verás múltiples carpetas dentro de la carpeta clonada.

- o Ahora necesitas navegar dentro de la carpeta clonada. Para esto, escribe el comando dado en el terminal:

```
cd <repository-folder-name>
```

Nota: Escribe el nombre de tu carpeta clonada en lugar de `<repository-folder-name>`. Realiza `git clone` si has cerrado sesión en el **Entorno de Skills Network** y no puedes ver ningún archivo o carpeta después de iniciar sesión.

2. Ahora selecciona la carpeta **cloned Folder Name**, haz clic derecho sobre ella y selecciona **Nueva Carpeta**. Ingresar el nombre de la carpeta como **bookSystem**. Esto creará la carpeta para ti. Luego selecciona la carpeta **bookSystem**, haz clic derecho y selecciona **Nuevo Archivo**. Ingresar el nombre del archivo como **book_system.html** y haz clic en OK. Esto creará tu archivo HTML.

3. Ahora selecciona la carpeta **bookSystem** nuevamente, haz clic derecho y selecciona **Nuevo Archivo**. Ingresar el nombre del archivo como **book_system.js** y haz clic en OK. Esto creará tu archivo JavaScript.

4. Crea la estructura básica del archivo HTML añadiendo el siguiente contenido de código.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Sistema de Gestión de Libros</title>
</head>
<body>
  <h1>Sistema de Gestión de Libros</h1>
  <label for="bookName">Nombre del Libro:</label><br>
  <input type="text" id="bookName"><br><br>
  <label for="authorName">Nombre del Autor:</label><br>
  <textarea id="authorName" rows="4"></textarea><br><br>
  <label for="bookDescription">Descripción del Libro:</label><br>
  <textarea id="bookDescription" rows="4"></textarea><br><br>
  <label for="pageNumber">Número de Páginas:</label><br>
  <input type="number" id="pageNumber"><br><br>
  <button onClick="addBook()">Agregar Libro</button>
  <h2>Libros</h2>
  <div id="books">
    <!-- los libros se mostrarán aquí -->
  </div>
  <script src="./book_system.js"></script>
</body>
</html>
```

- **Estructura HTML:** El código crea una estructura HTML básica para una interfaz de Sistema de Gestión de Libros. Consiste en varios elementos HTML, como `<input>`, `<textarea>` y `<button>` utilizados para recopilar información sobre los libros, incluyendo el nombre del libro, el nombre del autor, la descripción del libro y el número de páginas. La interfaz proporciona campos de entrada etiquetados y un botón para agregar libros al sistema.
- **Manejo de la entrada del usuario:** Cada campo de entrada (nombre del libro, nombre del autor, descripción y número de páginas) está asociado con un atributo ID, lo que permite que JavaScript recupere sus valores. Cuando se hace clic en el botón “Agregar Libro”, se activa la función `addBook()` (no incluida en el código proporcionado) para recopilar los detalles del libro ingresados desde los campos de entrada.
- **Sección de visualización de libros:** El código incluye una sección `<div id="books">` destinada a mostrar los libros añadidos. Sin embargo, carece de la funcionalidad de JavaScript para poblar y mostrar los libros. Tal como está, esta sección está indicada como el área donde se mostrarán los libros, pero no hay lógica de JavaScript proporcionada para mostrar los libros añadidos en esta sección.
- Para incluir el archivo `js` en **book_system.html**, se utiliza una etiqueta de `script` en el archivo HTML antes de la etiqueta `</body>`.

Nota: Después de pegar el código, guarda tu archivo.

Paso 2: Definición de variables y funciones

1. En **book_system.js**, declara una variable de arreglo vacía para libros.

```
let books = [];
```

2. Crear la función **addBook** para agregar detalles del libro en el sistema de gestión. Para esto, incluye el código proporcionado a continuación después del código anterior.

```
function addBook() {
  const bookName = document.getElementById('bookName').value;
  const authorName = document.getElementById('authorName').value;
  const bookDescription = document.getElementById('bookDescription').value;
  const pagesNumber = parseInt(document.getElementById('pagesNumber').value);
  if (bookName && authorName && bookDescription && !isNaN(pagesNumber)) {
    const book = {
      name: bookName,
      authorName: authorName,
      bookDescription: bookDescription,
      pagesNumber: pagesNumber
    };
    books.push(book);
    showbooks();
    clearInputs();
  } else {
    alert('Por favor, complete todos los campos correctamente.');
```

- **Recolección de datos y validación:** La función recupera los valores ingresados por el usuario en los campos de entrada HTML para el nombre del libro, nombre del autor, descripción del libro y número de páginas. Utiliza `document.getElementById('elementId').value` para acceder a estos valores. El código luego verifica si todos los campos han sido completados (`bookName`, `authorName`, `bookDescription`) y asegura que `pagesNumber` sea un número válido (usando `isNaN(pagesNumber)`). Si algún campo está vacío o si `pagesNumber` no es un número válido, se muestra una alerta que solicita al usuario que complete todos los campos correctamente.
- **Creación de un objeto libro:** Tras una validación exitosa, la función crea un objeto libro que contiene propiedades como `name`, `authorName`, `bookDescription` y `pagesNumber`, almacenando los datos ingresados por el usuario. Este objeto libro se añade al array `books`, efectivamente agregando el nuevo libro a la colección.
- **Mostrar y limpiar:** Después de agregar el libro, se llama a la función `showbooks()` para actualizar la visualización con los detalles del libro recién agregado. Además, se invoca la función `clearInputs()` para restablecer todos los campos de entrada, proporcionando una interfaz limpia para que el usuario agregue otro libro sin necesidad de limpiar manualmente las entradas anteriores.

Paso 3: Definir función para mostrar libros

1. Crea una función `showbooks` para mostrar los detalles de los libros en el sistema de gestión. Para esto, incluye el código proporcionado en el archivo `js` después del código anterior.

```
function showbooks() {
  const booksDiv = books.map((book, index) => `<h1>Número de libro: ${index + 1}</h1>
  <p><strong>Nombre del libro: </strong> ${book.name}</p>
  <p><strong>Nombre del autor:</strong> ${book.authorName}</p>
  <p><strong>Descripción del libro:</strong> ${book.bookDescription}</p>
  <p><strong>Número de páginas:</strong> ${book.pagesNumber} página(s)</p>
  <button onclick="editbook(${index})">Editar</button>`
  );
  document.getElementById('books').innerHTML = booksDiv.join('');
```

- La función `map()` itera a través del arreglo de libros, creando un nuevo arreglo de elementos HTML o cadenas basadas en la información del libro presente en cada elemento del arreglo.
- Generando información del libro:
 - Para cada libro en el arreglo de libros, crea una representación HTML estructurada.
 - Construye una cadena HTML que incluye:
 - Un encabezado que muestra el número del libro `<h1>Número de libro: ${index + 1}</h1>`.
 - Párrafos `<p>` que contienen detalles sobre el libro como nombre, nombre del autor, descripción del libro y el número de páginas.
 - Un botón para editar `<button onclick="editbook(${index})">Editar</button>` un libro específico. Este botón está configurado para llamar a la función `editbook()`, pasando el índice del libro como parámetro.
- El método `join('')` se emplea para concatenar todos los elementos HTML generados para cada libro en una sola cadena. Esta representación de cadena de elementos HTML permite que el contenido se inserte como un solo bloque de HTML.
- La propiedad `innerHTML` del elemento HTML con el ID `'books'` se establece en la cadena HTML generada (`booksDiv`). Esta acción reemplaza el contenido existente dentro del elemento `'books'` con la nueva estructura HTML creada que representa los libros.

2. Agrega la función `editbook` para llenar los campos del formulario con los datos del libro seleccionado:

```
function editbook(index) {
  const book = books[index];
  document.getElementById('bookName').value = book.name;
  document.getElementById('authorName').value = book.authorName;
  document.getElementById('bookDescription').value = book.bookDescription;
  document.getElementById('pagesNumber').value = book.pagesNumber;
  books.splice(index, 1); // Eliminar entrada antigua
  showbooks(); // Actualizar lista
}
```

- La función editbook permite a los usuarios editar los detalles de un libro completando previamente el formulario con sus datos existentes.
- `const book = books[index];` Recupera el libro del arreglo books usando el index dado.
- **Llenar Campos del Formulario:**
 - Establece el nombre del libro en el campo de entrada `bookName`.
 - Establece el nombre del autor en el campo de entrada `authorName`.
 - Establece la descripción del libro en el campo de entrada `bookDescription`.
 - Establece el número de páginas en el campo de entrada `pagesNumber`.
- `books.splice(index, 1);` Elimina el libro seleccionado del arreglo usando `splice(index, 1)` para evitar entradas duplicadas al guardar cambios.
- Llama a `showbooks()` para actualizar la lista de libros y reflejar los cambios.

Paso 4: Definiendo la función para limpiar libros

1. Crea una función **clearInputs** para limpiar los detalles del libro en el sistema de gestión. Para esto, incluye el código dado en el archivo **book_system.js** después del código anterior.

```
function clearInputs() {
  document.getElementById('bookName').value = '';
  document.getElementById('authorName').value = '';
  document.getElementById('bookDescription').value = '';
  document.getElementById('pagesNumber').value = '';
}
```

- El código anterior limpia los detalles del libro asignando un valor vacío a los elementos HTML utilizando los ID's.

Paso 5: Verifica la salida

1. Para ver tu página HTML, haz clic derecho en el archivo **book_system.html** después de seleccionarlo, luego selecciona “Abrir con Live Server”.
2. El servidor debería iniciarse en el puerto 5500, indicado por una notificación en la esquina inferior derecha.
3. Haz clic en el botón de Skills Network a la izquierda (consulta el número 1), se abrirá la “Caja de herramientas de Skills Network”. Luego haz clic en **Lanzar Aplicación** (consulta el número 2). Desde allí, ingresa el número de puerto como 5500 en el número 3 y haz clic en este botón .
4. Se abrirá tu navegador predeterminado donde verás el nombre de la carpeta **cloned-folder-name**. Haz clic en esa carpeta **cloned-folder-name** y luego haz clic en el nombre de la carpeta **bookSystem**. Verás archivos relacionados con esta carpeta donde volverás a hacer clic en el archivo **book_system.html** como se muestra a continuación.
5. Se abrirá la página HTML como se muestra a continuación.
6. Ahora ingresa los detalles y luego haz clic en el botón **Agregar Libro** y luego verás los detalles de los libros. En la captura de pantalla dada, las entradas se han enviado dos veces sin refrescar la página.

Nota: Después de pegar el código, guarda tu archivo. Puedes usar cualquier método de salida para guardar. Si haces más ediciones a tu código, simplemente actualiza tu navegador que está ejecutándose en el número de puerto 5500. Esto elimina la necesidad de relanzar la aplicación repetidamente.

Paso 6: Realizar comandos de Git

1. Realiza `git add` para agregar los últimos archivos y carpetas escribiendo el comando dado en la terminal en el entorno de git.

```
git add --a
```

Asegúrate de que la terminal tenga la ruta como sigue:

2. Luego realiza `git commit` en la terminal. Al realizar `git commit`, la terminal puede mostrar un mensaje para configurar tu `git config --global` para `user.name` y `user.email`. Si es así, también necesitas realizar el comando `git config` para `user.name` y `user.email` como se indica.

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

Nota: Reemplaza los datos entre comillas con tus propios detalles.

Luego realiza el comando de commit como se indica:

```
git commit -m "mensaje"
```

3. Luego realiza `git push` simplemente escribiendo el comando dado en la terminal.

```
git push origin
```

- Después del comando de push, el sistema te pedirá que ingreses tu nombre de usuario y contraseña. Ingresa el nombre de usuario de tu cuenta de GitHub y la contraseña que creaste en el primer laboratorio. Después de ingresar las credenciales, todas tus últimas carpetas y archivos se enviarán a tu repositorio de GitHub.

Tarea de práctica

1. En esta tarea de práctica, necesitas crear un botón dinámico en el momento en que los detalles ingresados por el usuario se mostrarán como salida después de hacer clic en el botón **Agregar libro**.

2. Necesitas crear un botón de eliminar como se muestra a continuación. Para esto, debes crear una función **deletebook**, que puede incluir el siguiente código parcial.

```
<button onclick=".....">Eliminar</button>
```

3. La función **deletebook** se llamará cuando el usuario haga clic en el botón de eliminar. Esta función también eliminará los detalles del libro en el sistema de gestión para ese ID particular.

Nota: El ID particular es el número de índice del array donde se almacenan los detalles del libro en particular.

4. Para eliminar los detalles del libro, puedes usar el método de array `splice()` dentro de la función **deletebook**. Este método se utiliza para modificar el array de libros, eliminando un solo elemento a partir del índice especificado.

```
books.splice(index, 1);
```

5. Esta función elimina la entrada del libro en el índice dado del array de libros. Después de la eliminación, llama a la función `showBooks` para refrescar la lista de libros mostrada.

► Haz clic aquí para el código de solución

Resumen

- Interfaz de gestión de libros:** El código establece una interfaz web simple para gestionar libros. Incluye campos de entrada para el nombre del libro, el nombre del autor, la descripción del libro y el número de páginas. Los usuarios pueden añadir libros utilizando el botón "Agregar Libro".
- Visualización dinámica de libros:** Al añadir libros, la función `showbooks()` genera dinámicamente contenido HTML para mostrar la lista de libros añadidos. Formatea y estructura los detalles del libro utilizando la función `map()` de JavaScript para crear elementos HTML para cada libro, incluyendo botones para editar y eliminar entradas individuales de libros.
- Funcionalidad de manejo de libros:** El código proporciona funcionalidades para editar y eliminar entradas de libros. La función `editbook()` permite a los usuarios editar los detalles del libro llenando los campos de entrada con la información del libro seleccionado. Mientras tanto, la función `deletebook()` elimina una entrada de libro de la lista mostrada cuando se hace clic en el botón de eliminar correspondiente. Ambas acciones de edición y eliminación se manejan actualizando la lista de libros dinámicamente utilizando la función `showbooks()`, que refresca la lista mostrada después de cualquier cambio.

© IBM Corporation. Todos los derechos reservados.