

Documentación - Lista de Tareas JavaScript

Variables Globales

```
javascript
```

```
let taskCounter = 0;
```

Propósito: Inicializa un contador global para llevar el registro del número total de tareas creadas.

```
javascript
```

```
let btn = document.getElementById('btn-add-task');
```

Propósito: Obtiene la referencia al botón HTML que tiene el ID 'btn-add-task' y la almacena en la variable `btn`.

Event Listener Principal

```
javascript
```

```
btn.addEventListener("click", function(){
```

Propósito: Agrega un escuchador de eventos al botón que ejecutará la función cuando se haga clic en él.

Creación de Elementos DOM

```
javascript
```

```
let li = document.createElement("li");
```

Propósito: Crea un nuevo elemento de lista (``) que contendrá cada tarea individual.

```
javascript
```

```
li.className = "list-group-item d-flex justify-content-between align-items-start";
```

Propósito: Asigna clases CSS de Bootstrap al elemento `` para darle estilo y comportamiento flexbox.

```
javascript
```

```
let span = document.createElement("span");
```

Propósito: Crea un elemento `` que contendrá el texto de la tarea.

```
javascript
```

```
span.className = "mt-1";
```

Propósito: Asigna la clase CSS `mt-1` (margin-top: 1) al span para espaciado superior.

Referencias a Elementos Existentes

```
javascript
```

```
let countTask = document.getElementById("countTask")
```

Propósito: Obtiene la referencia al elemento HTML que mostrará el contador de tareas.

```
javascript
```

```
let deleteButton = document.createElement("button");
```

Propósito: Crea un botón que permitirá eliminar cada tarea individual.

```
javascript
```

```
deleteButton.textContent = "Delete";
```

Propósito: Establece el texto visible del botón de eliminar.

```
javascript
```

```
deleteButton.className = "btn btn-danger";
```

Propósito: Asigna clases CSS de Bootstrap para darle estilo de botón rojo al botón de eliminar.

```
javascript
```

```
let taskInput = document.getElementById("taskInput"); // For task entries
```

Propósito: Obtiene la referencia al campo de entrada donde el usuario escribe las nuevas tareas.

```
javascript
```

```
let taskList = document.getElementById("taskList"); // For listing tasks
```

Propósito: Obtiene la referencia al contenedor donde se mostrarán todas las tareas.

Validación de Entrada

javascript

```
if (taskInput.value == "") {  
    alert("Please, write a task!");  
    taskInput.focus();  
    return;  
}
```

Propósito:

- Verifica si el campo de entrada está vacío
- Si está vacío, muestra una alerta al usuario
- Pone el foco en el campo de entrada
- Sale de la función sin crear la tarea

Construcción y Agregado de la Tarea

javascript

```
span.textContent = taskInput.value; // Lo que esta en la caja de entrada pasa al span
```

Propósito: Transfiere el texto ingresado por el usuario al elemento span que mostrará la tarea.

javascript

```
li.appendChild(span); //Agrego el span dentro del Li
```

Propósito: Inserta el span (que contiene el texto de la tarea) dentro del elemento de lista.

javascript

```
taskCounter++;
```

Propósito: Incrementa el contador global de tareas en 1.

javascript

```
li.appendChild(deleteButton);
```

Propósito: Agrega el botón de eliminar al elemento de lista, colocándolo después del texto de la tarea.

javascript

```
countTask.textContent = `Cantidad de Tareas : ${taskCounter} `; // Template Literals
```

Propósito: Actualiza el contador visual usando template literals para mostrar el número actual de tareas.

```
javascript
```

```
taskList.appendChild(li);
```

Propósito: Agrega el elemento de lista completo (con texto y botón) al contenedor principal de tareas.

Limpieza de la Interfaz

```
javascript
```

```
taskInput.value = "";
```

Propósito: Limpia el campo de entrada para que el usuario pueda escribir una nueva tarea.

```
javascript
```

```
taskInput.focus();
```

Propósito: Pone el foco en el campo de entrada para mejorar la experiencia del usuario.

Funcionalidad de Eliminación

```
javascript
```

```
deleteButton.addEventListener('click', function () {
```

Propósito: Agrega un escuchador de eventos al botón de eliminar de cada tarea individual.

```
javascript
```

```
if(confirm("¿Estás seguro de que quieres eliminar esta tarea?")){
```

Propósito: Muestra un diálogo de confirmación antes de eliminar la tarea para evitar eliminaciones accidentales.

```
javascript
```

```
taskCounter--;
```

Propósito: Decrementa el contador global de tareas en 1.

```
javascript
```

```
countTask.textContent = "Number of tasks : " + taskCounter;
```

Propósito: Actualiza el contador visual usando concatenación de strings (nota: hay inconsistencia de idioma con la línea anterior).

```
javascript  
  
taskList.removeChild(li);
```

Propósito: Elimina el elemento de lista específico del DOM, removiendo visualmente la tarea de la interfaz.

Estructura General del Código

Este código implementa una aplicación de lista de tareas con las siguientes características:

✅ Funcionalidades Principales:

- Agregar nuevas tareas
- Validar entrada vacía
- Contar tareas activas
- Eliminar tareas individuales con confirmación
- Interfaz responsive con Bootstrap

⚠️ Observaciones de Mejora:

1. **Inconsistencia de idioma:** Mezcla español e inglés en mensajes
2. **Declaración de variables:** Las variables del DOM se declaran dentro del event listener (podrían ser globales)
3. **Reutilización:** El código podría ser más modular para facilitar mantenimiento

🔧 Dependencias:

- **Bootstrap CSS:** Para estilos de los elementos
- **HTML:** Debe tener elementos con IDs específicos:
 - `btn-add-task`: Botón para agregar tareas
 - `taskInput`: Campo de entrada de texto
 - `taskList`: Contenedor de la lista
 - `countTask`: Elemento para mostrar el contador