






Documentación - Lista de Tareas JavaScript

Descripción General

Este código JavaScript implementa una aplicación de lista de tareas (To-Do List) que permite a los usuarios agregar y eliminar tareas dinámicamente. Utiliza manipulación del DOM y eventos para crear una interfaz interactiva.

Funcionalidades Principales

-  Agregar nuevas tareas
 -  Eliminar tareas existentes
 -  Contador automático de tareas
 -  Validación de entrada vacía
 -  Interfaz estilizada con Bootstrap
-

Documentación Línea por Línea

Inicialización de Variables

```
javascript  
  
let taskCounter = 0;
```

Propósito: Declara una variable para llevar el conteo total de tareas activas.

- **Tipo:** `number`
- **Valor inicial:** `0`
- **Función:** Se incrementa al agregar tareas y se decrementa al eliminarlas

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

Propósito: Obtiene la referencia del botón "Agregar Tarea" del HTML.

- **Método:** `getElementById()` - busca un elemento por su ID
 - **Elemento esperado:** Un botón con `id="btn-add-task"`
 - **Uso:** Para agregar el event listener de click
-

Event Listener Principal

javascript

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

Propósito: Registra un evento de click en el botón de agregar tarea.

- **Evento:** `click`
 - **Función:** Anónima que se ejecuta cada vez que se hace click en el botón
-

Creación de Elementos DOM

javascript

```
let li = document.createElement("li");  
li.className = "list-group-item d-flex justify-content-between align-items-start";
```

Propósito: Crea un elemento de lista (``) para la nueva tarea.

- **Elemento:** `` (list item)
- **Clases Bootstrap:**
 - `list-group-item`: Estilo de elemento de lista
 - `d-flex`: Display flex
 - `justify-content-between`: Espaciado entre elementos
 - `align-items-start`: Alineación vertical al inicio

javascript

```
let span = document.createElement("span");  
span.className = "mt-1";
```

Propósito: Crea un elemento `` para contener el texto de la tarea.

- **Elemento:** `` (contenedor inline)
- **Clase Bootstrap:** `mt-1` (margin-top pequeño)

javascript

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

Propósito: Obtiene referencia al elemento que muestra el contador de tareas.

- **Elemento esperado:** Elemento con `id="countTask"`

javascript

```
let deleteButton = document.createElement("button");
deleteButton.textContent = "Delete";
deleteButton.className = "btn btn-danger";
```

Propósito: Crea el botón de eliminar para cada tarea.

- **Elemento:** `<button>`
- **Texto:** "Delete"
- **Clases Bootstrap:**
 - `btn`: Estilo base de botón
 - `btn-danger`: Botón rojo (peligro)

Referencias a Elementos Existentes

javascript

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

Propósito: Obtiene referencias a elementos clave del HTML.

- **taskInput:** Campo de entrada donde el usuario escribe la tarea
- **taskList:** Contenedor donde se muestran todas las tareas

Validación de Entrada

javascript

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

Propósito: Valida que el usuario haya escrito algo antes de agregar la tarea.

- **Condición:** Verifica si el campo está vacío ("")
 - **Acciones si está vacío:**
 - Muestra alerta de error
 - Enfoca el campo de entrada
 - Termina la ejecución con (return)
-

Construcción de la Tarea

javascript

```
span.textContent = taskInput.value; // Lo que esta en la caja de entrada pasa al span  
li.appendChild(span); //Agrego el span dentro del li  
taskCounter++;  
li.appendChild(deleteButton);
```

Propósito: Construye la estructura de la nueva tarea.

1. **Línea 1:** Asigna el texto ingresado al span
 2. **Línea 2:** Agrega el span al elemento li
 3. **Línea 3:** Incrementa el contador de tareas
 4. **Línea 4:** Agrega el botón de eliminar al li
-

Actualización de la Interfaz

javascript

```
countTask.textContent = `CANTIDAD DE TAREAS: ${taskCounter}`; //TEMPLATE LITERAL  
taskList.appendChild(li);  
taskInput.value = "";  
taskInput.focus();
```

Propósito: Actualiza la interfaz con la nueva tarea.

1. **Línea 1:** Actualiza el contador usando template literal (ES6)
 2. **Línea 2:** Agrega la nueva tarea a la lista visible
 3. **Línea 3:** Limpia el campo de entrada
 4. **Línea 4:** Enfoca el campo para la siguiente entrada
-

Funcionalidad de Eliminación

javascript

```
deleteButton.addEventListener('click', function () {  
    if (confirm("¿Estás seguro de que quieres eliminar esta tarea?")){  
        taskCounter--;  
        countTask.textContent = "Number of tasks : " + taskCounter;  
        taskList.removeChild(li);  
    }  
})
```

Propósito: Maneja la eliminación de tareas individuales.

Flujo de eliminación:

1. **Event Listener:** Se activa al hacer click en "Delete"
2. **Confirmación:** Muestra diálogo de confirmación
3. **Si confirma:**
 - Decrementa el contador de tareas
 - Actualiza el texto del contador
 - Remueve el elemento li de la lista

Nota: Hay inconsistencia en el idioma (español en confirmación, inglés en contador)



Estructura HTML Esperada

Para que este código funcione correctamente, el HTML debe contener:

html

```
<!-- Campo de entrada para escribir tareas -->
<input type="text" id="taskInput" placeholder="Escribe tu tarea...">

<!-- Botón para agregar tareas -->
<button id="btn-add-task">Agregar Tarea</button>

<!-- Contenedor para mostrar el contador -->
<div id="countTask">CANTIDAD DE TAREAS: 0</div>

<!-- Lista donde aparecerán las tareas -->
<ul id="taskList" class="list-group"></ul>
```






Dependencias

Este código utiliza **Bootstrap** para el estilizado:





- `list-group-item`
 - `d-flex`, `justify-content-between`, `align-items-start`
 - `mt-1`
 - `btn`, `btn-danger`
-

Análisis de Mejoras Posibles

Fortalezas:

-  Funcionalidad completa básica
-  Validación de entrada
-  Confirmación antes de eliminar
-  Contador automático
-  Limpieza automática del campo

Áreas de Mejora:

-  **Consistencia de idioma** (mezcla español/inglés)
-  **Separación de responsabilidades** (función muy larga)
-  **Manejo de errores** más robusto
-  **Accesibilidad** (ARIA labels, navegación por teclado)

-  **Persistencia** (localStorage)
-

Ejemplo de Uso

1. Usuario escribe "Comprar leche" en el campo de entrada
 2. Hace click en "Agregar Tarea"
 3. Se crea una nueva tarea con botón "Delete"
 4. El contador se actualiza a "CANTIDAD DE TAREAS: 1"
 5. Para eliminar, hace click en "Delete" y confirma
 6. La tarea desaparece y el contador vuelve a 0
-

Resumen Técnico

- **Lenguaje:** JavaScript ES6+
- **Patrón:** Manipulación directa del DOM
- **Eventos:** Click listeners
- **Validación:** Verificación de campo vacío
- **UI Framework:** Bootstrap 4/5
- **Funciones principales:** Crear, mostrar y eliminar tareas