

Documentación: Lista de Tareas en JavaScript

Variables Globales

javascript

```
let taskCounter = 0;
```

Propósito: Variable que cuenta el número total de tareas creadas. Inicia en 0 y se incrementa/decrementa según se agreguen o eliminen tareas.

javascript

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

Propósito: Obtiene una referencia al botón de "Agregar Tarea" del HTML usando su ID. Esta variable almacena el elemento del DOM para poder agregarle eventos.

Event Listener del Botón Principal

javascript

```
btn.addEventListener("click", function(){  
    // Aquí iría el código para agregar una tarea  
});
```

Propósito: Agrega un "escuchador de eventos" al botón. Cuando el usuario hace clic en el botón, se ejecuta la función anónima (sin nombre) que está dentro.

Función Principal: addTask()

Creación de Elementos del DOM

javascript

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

Propósito:

- Crea un nuevo elemento `` (item de lista) en memoria
- Le asigna clases de Bootstrap para styling: lista agrupada con flexbox para alinear contenido

javascript

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

Propósito:

- Crea un elemento `` que contendrá el texto de la tarea
- Le asigna la clase `mt-1` (margin-top-1) de Bootstrap para espaciado

javascript

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

Propósito:

- Crea un botón de eliminar
- Le asigna el texto "Eliminar"
- Le da clases de Bootstrap: `btn` (botón básico) y `btn-danger` (color rojo)

Obtención de Referencias del DOM

javascript

```
let taskInput = document.getElementById("taskInput");
```

Propósito: Obtiene referencia al campo de entrada de texto donde el usuario escribe la tarea.

javascript

```
let taskList = document.getElementById("taskList");
```

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

javascript

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

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

Validación de Entrada

javascript

```
if (taskInput.value == "") {  
    alert("Ingrese una Tarea");  
    taskInput.focus();  
    return;  
}
```

Propósito:

- Verifica si el campo de entrada está vacío
- Si está vacío: muestra una alerta, pone el cursor en el campo (`focus()`) y sale de la función (`return`)
- Previene agregar tareas vacías

Event Listener para Eliminar Tarea

javascript

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

Propósito:

- Agrega funcionalidad al botón eliminar
- `confirm()`: Muestra un diálogo de confirmación
- Si el usuario confirma:
 - Decrementa el contador de tareas
 - Actualiza el texto del contador en pantalla
 - Elimina el elemento `li` del DOM usando `removeChild()`

Construcción y Visualización de la Tarea

javascript

```
span.textContent = taskInput.value;
```

Propósito: Toma el texto que escribió el usuario en el campo de entrada y lo coloca dentro del `span`.

```
javascript
```

```
li.appendChild(span);
```

Propósito: Agrega el `span` (con el texto de la tarea) como hijo del elemento `li`.

```
javascript
```

```
taskCounter++;
```

Propósito: Incrementa el contador de tareas en 1.

```
javascript
```

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

Propósito: Agrega el botón de eliminar como hijo del elemento `li`.

```
javascript
```

```
countTask.textContent = "Cantidad de Tareas : " + taskCounter;
```

Propósito: Actualiza el texto del contador mostrado en pantalla con el nuevo número de tareas.

```
javascript
```

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

Propósito: Agrega todo el elemento `li` (que ya contiene el texto y el botón) a la lista de tareas en el DOM. **Aquí es donde la tarea se hace visible en pantalla.**

Limpieza del Formulario

```
javascript
```

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

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

Propósito:

- Limpia el campo de entrada (lo deja vacío)
- Pone el cursor de vuelta en el campo para facilitar agregar otra tarea



Flujo de Ejecución

1. **Usuario hace clic** → Se activa el event listener

2. **Validación** → Verifica que haya texto ingresado
 3. **Creación** → Se crean los elementos HTML en memoria
 4. **Configuración** → Se configuran propiedades y eventos
 5. **Ensamblaje** → Se conectan los elementos entre sí
 6. **Visualización** → Se agrega todo al DOM (se hace visible)
 7. **Limpieza** → Se prepara para la siguiente tarea
-

Clases CSS Utilizadas

- `list-group-item`: Estilo de Bootstrap para items de lista
 - `d-flex`: Display flex
 - `justify-content-between`: Espaciado entre elementos
 - `align-items-start`: Alineación vertical
 - `mt-1`: Margen superior pequeño
 - `btn`: Clase base para botones
 - `btn-danger`: Botón rojo de Bootstrap
-

Notas sobre el Código

- Hay duplicación en el event listener del botón eliminar
- Se mezclan concatenación de strings tradicional (`(+)`) con template literals (``${variable}``)
- El código funciona pero podría optimizarse eliminando la duplicación