



Problema Lab

Módulo de Iniciación

MANIPULA EL DOM Y LA GESTIÓN DE EVENTOS

En este ejercicio, nos enfocaremos en familiarizarnos con los conceptos fundamentales de la manipulación del DOM y la gestión de eventos en JavaScript. Para empezar, seleccionaremos un proyecto que en nuestro caso consistirá en desarrollar **una aplicación para administrar tareas**. Haremos uso de un *template* (plantilla) HTML que proporcionamos, lo que nos permitirá centrarnos más en la lógica de JavaScript que en la estructura básica del HTML. A partir de aquí, empezaremos a explorar cómo interactuar con los elementos HTML, a modificar su estructura y contenido, y responder a eventos del usuario, como clics o entradas de teclado. Al final del ejercicio, habremos creado una interfaz interactiva que refleje nuestra comprensión y habilidades en la manipulación del DOM y en la gestión de eventos en JavaScript, todo ello sobre una aplicación práctica de la administración de tareas.

Objetivos de este ejercicio

El objetivo final es crear un administrador de tareas usando el template HTML y CSS proporcionado y desarrollando la funcionalidad en JavaScript.

Deberás poder crear, eliminar y marcar como completadas (o no) las tareas que aparezcan en la lista.

Descripción de la actividad

Se te proporcionará un archivo comprimido en el que se encuentra una carpeta *src* y un archivo *README.md*. En el directorio *src*, se encuentra toda la maqueta (HTML y CSS) del administrador de tareas y un archivo llamado ***script.js*** que es donde escribirás tu código. Para este lab te pediremos algo relativamente sencillo, ¡toma nota!

1. Abre el fichero **index.html** y familiarízate con los elementos que están dentro el **<article class="container">**.
2. Obtén los datos que se ingresan en el **<input id="taskInput">**.
3. Crea dos estructuras de datos: una llamada **task** con una propiedad llamada **name** para almacenar la información obtenida a través del input; otra llamada **id** para identificar; y otra llamada **completed** para saber su estado. La segunda estructura tiene como finalidad almacenar el listado de tareas llamada **tasks**.
4. Cuando el usuario haga clic en el **<button id="addTaskButton">**, se debe añadir la task al listado de tasks. Ten en cuenta que una task no puede ser añadida si el texto introducido está vacío o solo contiene espacios.

5. Añade el listado de tareas en la **<ul id="taskList">**. En el HTML tienes un template de cómo deberían de ser los **** del listado.
6. Añade la funcionalidad para que cuando se marque el **<input type="checkbox">**, la tarea cambie el estado **completed** acorde al estado del checkbox.
7. Cuando un usuario hace clic en el elemento que tiene la clase **"fa-trash"**, se debe quitar del listado la tarea en la que se ha clicado.

Puedes apoyarte en estas pautas que te damos a continuación:

- Cuando manipules el listado **tasks** (añadir o quitar elementos) recuerda que debes **rejecutar** lo que uses para pintar las tareas en el **DOM**, de lo contrario no verás los cambios en la página.
- Recuerda que cuando no recuerdes o no sepas cómo *trabaja* una función o evento, siempre puedes recurrir a la documentación oficial en [mozilla](https://developer.mozilla.org/).

Formato de entrega

Comparte en el campus tu ejercicio con todos los archivos que te fueron proporcionados para comprobar las funcionalidades que añadiste.

Criterios de corrección

Para autoevaluar tu práctica, toma como referencia el vídeo con la solución que ha realizado el profesor del lab. Como él indica, en este primer ejercicio lo importante es que hayas sabido manipular el DOM y la gestión de eventos. Recuerda que en el desarrollo diferentes personas pueden resolver un mismo problema de diferentes formas. Por lo tanto, mientras que el resultado de esta prueba sea correcto, no es necesario que el desarrollo del código sea el mismo. Lo cierto es que con este primer reto solo queremos que pongas a prueba tu capacidad resolutive necesaria en el mundo del desarrollo. Así que...

¡Manos a la obra y sin miedo!, que tú puedes con este reto.

