

Ejercicios JSON

Ejercicio 1: Gestionar datos de una tienda online

Contexto: Eres el desarrollador de una tienda online. El sistema te entrega un archivo JSON con los productos disponibles, y necesitas implementarlo para mostrar la información en la web.

Requerimientos:

1. Dado un JSON con una lista de productos, extrae los nombres y precios de cada producto y muéstralos en una tabla en el DOM.
2. Añade un botón en cada fila de la tabla que permita marcar un producto como destacado. Al hacer clic, el fondo de esa fila debe cambiar de color.
3. Usa una función flecha como callback para manejar el evento del botón.

Datos JSON ejemplo:

```
[  
  { "id": 1, "nombre": "Camisa", "precio": 20.99 },  
  { "id": 2, "nombre": "Pantalón", "precio": 35.50 },  
  { "id": 3, "nombre": "Zapatos", "precio": 50.00 }  
]
```

Ejercicio 2: Sistema de puntuaciones de jugadores

Contexto: Estás desarrollando una aplicación para un torneo de videojuegos. Los datos de los jugadores se encuentran en formato JSON, y necesitas mostrar las puntuaciones y permitir ordenarlas.

Requerimientos:

1. Muestra en el DOM una lista con los nombres de los jugadores y sus puntuaciones extraídas de un JSON.
2. Añade dos botones: uno para ordenar los jugadores por puntuación (de mayor a menor) y otro para ordenarlos alfabéticamente por su nombre.
3. Implementa las funciones de ordenación usando `sort` y asegúrate de que las puntuaciones y nombres se actualicen en el DOM tras cada clic.

Datos JSON ejemplo:

```
[  
  { "nombre": "Ana", "puntuacion": 150 },  
  { "nombre": "Carlos", "puntuacion": 200 },  
  { "nombre": "Beatriz", "puntuacion": 180 }  
]
```

Ejercicio 3: Filtrar eventos deportivos

Contexto: Trabajas en un portal de eventos deportivos. Necesitas implementar un filtro dinámico para eventos según su categoría usando datos en JSON.

Requerimientos:

1. Dado un JSON con eventos deportivos, muestra todos los eventos en una lista al cargar la página.
2. Añade un formulario con un `<select>` que permita elegir una categoría (e.g., fútbol, baloncesto, tenis).
3. Cuando se seleccione una categoría, filtra los eventos y muestra solo los correspondientes en el DOM.
4. Usa `this` correctamente dentro de un callback para gestionar los eventos del formulario.

Datos JSON ejemplo:

```
[  
  { "id": 1, "evento": "Partido de fútbol", "categoria": "Fútbol" },  
  { "id": 2, "evento": "Final de baloncesto", "categoria": "Baloncesto" },  
  { "id": 3, "evento": "Torneo de tenis", "categoria": "Tenis" },  
  { "id": 4, "evento": "Amistoso de fútbol", "categoria": "Fútbol" }  
]
```

Ejercicio 4: Información del clima con JSON

Contexto: Estás desarrollando una aplicación del tiempo que recibe datos del clima en formato JSON. Debes procesar estos datos y mostrar la información relevante.

Requerimientos:

1. Dado un JSON con datos de clima, muestra en el DOM la ciudad, temperatura actual y una breve descripción del clima.
2. Añade un botón que, al hacer clic, actualice los datos del clima simulando una nueva llamada a un servidor. Usa `setTimeout` para simular la espera de los datos.
3. Implementa el manejo de eventos y el uso de funciones callback.

Datos JSON iniciales:

```
{  
  "ciudad": "Barcelona",  
  "temperatura": 20,  
  "descripcion": "Soleado"  
}
```

Nuevos datos al hacer clic:

```
{  
  "ciudad": "Barcelona",  
  "temperatura": 18,  
  "descripcion": "Parcialmente nublado"  
}
```

Ejercicio 5: Crear un gestor de tareas

Contexto: Estás desarrollando una aplicación de gestión de tareas. Las tareas están almacenadas en JSON y deben mostrarse en el navegador con opciones de interacción.

Requerimientos:

1. Muestra en el DOM las tareas almacenadas en un JSON, incluyendo su estado (**completada** o **pendiente**).
2. Añade un botón junto a cada tarea para marcarla como completada. Cambia su estilo al hacerlo.
3. Añade un botón que permita filtrar las tareas completadas y solo mostrar esas en el DOM.
4. Usa funciones flecha y callbacks para manejar los eventos.

Datos JSON ejemplo:

```
[  
  { "id": 1, "tarea": "Comprar pan", "estado": "pendiente" },  
  { "id": 2, "tarea": "Estudiar JavaScript", "estado": "completada" },  
  { "id": 3, "tarea": "Llamar a mamá", "estado": "pendiente" }  
]
```