

Ejercicios Eventos

Ejercicio 1: Fases de la propagación de eventos

Crea un documento HTML con tres elementos anidados: un div, un section y un button. Utiliza JavaScript para agregar manejadores de eventos a cada uno de ellos. Estos manejadores deben mostrar un mensaje en la consola con el nombre del elemento y la fase en la que se encuentra (captura o burbuja).

Investiga y responde:

1. ¿En qué orden se ejecutan los eventos en la fase de captura?
2. ¿Qué sucede si detienes la propagación de eventos con `stopPropagation()`?
3. ¿Cómo afecta la fase de burbuja al comportamiento general?

Ejercicio 2: Detener la propagación de eventos

Diseña un formulario con los siguientes elementos:

1. Un campo de texto para el nombre del usuario.
2. Un botón que diga "Enviar".

Agrega los siguientes manejadores:

- Un manejador en el form que muestre un mensaje en la consola indicando que el formulario ha sido enviado.
- Un manejador en el botón "Enviar" que detenga tanto la propagación como el envío del formulario.

Explica cómo cambiaría el comportamiento si `stopPropagation()` no se utilizara.

Ejercicio 3: Delegación de eventos

Crea una lista de tareas con los siguientes elementos:

1. Un contenedor `ul` con tres tareas iniciales como `li`.
2. Un botón que permita añadir nuevas tareas dinámicamente.

Utiliza delegación de eventos para que, al hacer clic en cualquier tarea (`li`), se muestre un mensaje en la consola con el texto de la tarea seleccionada.

Finalmente, añade funcionalidad para que al hacer clic en el botón, se agregue un nuevo elemento `li` a la lista, y este también pueda ser detectado por el manejador de eventos del contenedor `ul`.

Ejercicio 4: Prevenir el comportamiento predeterminado

Crea una página con los siguientes elementos:

1. Un enlace (<a>) que dirija a una página externa.
2. Un contenedor div que rodee al enlace.

Agrega manejadores de eventos que:

- Eviten que el navegador siga el enlace cuando el usuario haga clic en él.
 - Registren un mensaje en la consola al detectar un clic en el div.
- Prueba el comportamiento con y sin stopPropagation() en el manejador del enlace y explica qué ocurre.

Ejercicio 5: Eventos con captura y burbuja

Crea un documento con este HTML:

```
<div id="contenedor">
  <section id="seccion">
    <button id="boton">Haz clic aquí</button>
  </section>
</div>
```

En JavaScript:

1. Añade un manejador en la fase de captura para el div.
2. Añade un manejador en la fase de burbuja para el section.
3. Añade un manejador al button que muestre un mensaje en la consola.

Explica:

- ¿Cuál es el orden de ejecución de los manejadores?
- ¿Cómo cambia el comportamiento si detienes la propagación del evento en el manejador del button?

Ejercicio 6: Combinación de propagación y delegación

Crea el siguiente HTML:

```
<div id="contenedor">
  <button class="boton">Botón 1</button>
  <button class="boton">Botón 2</button>
</div>
```

Agrega los siguientes manejadores:

1. Un manejador en el div que registre un mensaje en la consola cuando se haga clic en cualquier botón.
2. Manejadores en cada botón que detengan la propagación del evento y muestren un mensaje en la consola con el texto del botón.

Explica qué sucede si:

- Quitas `stopPropagation()` de los manejadores de los botones.
- Agregas nuevos botones dinámicamente al contenedor `div`.