

Ejercicios de eventos + objetos en JavaScript

En esta hoja tienes ejercicios donde debes combinar eventos del DOM con objetos y arrays de objetos en JavaScript. En el PDF solo va el HTML base; en tu archivo JS deberás crear los objetos y manejar los eventos usando los ids proporcionados.

Ejercicio 1 – Carrito de la compra (array de objetos producto)

Crea en JavaScript un array de objetos llamado 'productos', donde cada producto tenga las propiedades id, nombre y precio. Al hacer clic en un botón 'Añadir al carrito' de la tabla, debes buscar el producto por su id en el array y añadirlo a otro array llamado 'carrito'. Actualiza en pantalla el número de productos en el carrito y el precio total usando los elementos con id 'contador-carrito' y 'total-carrito'.

HTML base del ejercicio:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ejercicio 1 - Carrito de la compra</title>
</head>
<body>
    <h1>Ejercicio 1 - Carrito de la compra</h1>

    <table id="tabla-productos" border="1">
        <thead>
            <tr>
                <th>Nombre</th>
                <th>Precio</th>
                <th>Acción</th>
            </tr>
        </thead>
        <tbody>
            <!-- Relaciona estos ids con tus objetos de JS (mismo id numérico) -->
            <tr>
                <td>Producto 1</td>
                <td>10</td>
                <td><button class="btn-anadir-carrito" data-id="1">Añadir al carrito</button></td>
            </tr>
            <tr>
                <td>Producto 2</td>
                <td>15</td>
                <td><button class="btn-anadir-carrito" data-id="2">Añadir al carrito</button></td>
            </tr>
            <tr>
                <td>Producto 3</td>
                <td>20</td>
                <td><button class="btn-anadir-carrito" data-id="3">Añadir al carrito</button></td>
            </tr>
        </tbody>
    </table>

    <h2>Resumen del carrito</h2>
    <p>Productos en el carrito: <span id="contador-carrito">0</span></p>
    <p>Total: <span id="total-carrito">0</span> €</p>

    <!-- Maneja los eventos 'click' de los botones con clase 'btn-anadir-carrito'. -->
</body>
</html>
```

Ejercicio 2 – Gestor de tareas (objetos con estado)

Crea un array de objetos 'tareas', donde cada tarea tenga id, titulo y completada (true/false). Cuando el usuario escriba una nueva tarea y pulse el botón 'Añadir tarea', crea un nuevo objeto y añádelo al array. Pinta todas las tareas en el con id 'lista-tareas'. Cuando el usuario haga clic sobre el checkbox de una tarea, cambia la propiedad 'completada' del objeto correspondiente y actualiza

el contador de tareas completadas en el elemento con id 'contador-completadas'.

HTML base del ejercicio:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ejercicio 2 - Gestor de tareas</title>
</head>
<body>
    <h1>Ejercicio 2 - Gestor de tareas</h1>

    <input id="input-nueva-tarea" type="text" placeholder="Descripción de la tarea">
    <button id="btn-anadir-tarea">Añadir tarea</button>

    <h2>Listado de tareas</h2>
    <ul id="lista-tareas">
        <!-- Aquí pintarás las tareas desde tu array de objetos -->
    </ul>

    <p>Tareas completadas: <span id="contador-completadas">0</span></p>

    <!-- Maneja 'click' en btn-anadir-tarea y en los checkboxes que generes dinámicamente. -->
</body>
</html>
```

Ejercicio 3 – Filtro de libros por categoría (array de objetos)

Crea en JS un array de objetos 'libros' con id, título, autor y categoría. Cuando el usuario cambie la opción del con id 'filtro-categoría', muestra en el con id 'tbody-libros' solo los libros cuya categoría coincida. Si elige 'todas', muestra todos. Cada fila debe incluir un botón 'Ver detalle' que, al hacer clic, muestre la información completa del objeto libro en el con id 'detalle-libro'.

HTML base del ejercicio:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ejercicio 3 - Filtro de libros</title>
</head>
<body>
    <h1>Ejercicio 3 - Filtro de libros por categoría</h1>

    <label for="filtro-categoría">Categoría:</label>
    <select id="filtro-categoría">
        <option value="todas">Todas</option>
        <option value="novela">Novela</option>
        <option value="ensayo">Ensaya</option>
        <option value="fantasia">Fantasía</option>
    </select>

    <table border="1">
        <thead>
            <tr>
                <th>Título</th>
                <th>Autor</th>
                <th>Categoría</th>
                <th>Acciones</th>
            </tr>
        </thead>
        <tbody id="tbody-libros">
            <!-- Rellena esta tabla a partir del array de objetos 'libros' -->
        </tbody>
    </table>

    <h2>Detalle del libro seleccionado</h2>
    <div id="detalle-libro">
        <!-- Muestra aquí el objeto libro seleccionado -->
    </div>

    <!-- Usa 'change' sobre filtro-categoría y 'click' en los botones de cada fila. -->
</body>
</html>
```

Ejercicio 4 – Alumnado y notas (métodos sobre objetos)

Crea un array de objetos 'alumnos'. Cada alumno tendrá id, nombre y un array de notas. Añade un método (o función externa) que calcule la nota media de un alumno a partir de su objeto. Al pulsar el botón 'Mostrar todo el alumnado', muestra todos los alumnos en la tabla con id 'tabla-alumnos'. Al pulsar el botón 'Mostrar solo aprobados', pinta solo los alumnos cuya media sea ≥ 5 . Usa los botones con id 'btn-todos' y 'btn-aprobados'.

HTML base del ejercicio:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ejercicio 4 - Alumnado y notas</title>
</head>
<body>
    <h1>Ejercicio 4 - Alumnado y notas</h1>

    <button id="btn-todos">Mostrar todo el alumnado</button>
    <button id="btn-aprobados">Mostrar solo aprobados</button>

    <table id="tabla-alumnos" border="1">
        <thead>
            <tr>
                <th>Nombre</th>
                <th>Notas</th>
                <th>Media</th>
            </tr>
        </thead>
        <tbody>
            <!-- Rellenar dinámicamente a partir del array de objetos 'alumnos' -->
        </tbody>
    </table>

    <!-- Maneja los 'click' de btn-todos y btn-aprobados para filtrar el array. -->
</body>
</html>
```

Ejercicio 5 – Personajes y ficha detallada (objeto seleccionado)

Crea un array de objetos 'personajes' de un juego, con propiedades como id, nombre, clase, vida y ataque. Rellena la lista con id 'lista-personajes' con un por personaje. Cuando el usuario haga clic en un elemento de la lista, busca el objeto correspondiente en el array y muestra toda su información en el panel con id 'ficha-personaje'. Resalta en la lista el personaje seleccionado añadiendo o quitando una clase CSS desde JS.

HTML base del ejercicio:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ejercicio 5 - Personajes</title>
    <style>
        .personaje-seleccionado {
            font-weight: bold;
            text-decoration: underline;
        }
    </style>
</head>
<body>
    <h1>Ejercicio 5 - Personajes de juego</h1>

    <section>
        <h2>Listado de personajes</h2>
        <ul id="lista-personajes">
            <!-- Rellenar con elementos <li> a partir del array de objetos 'personajes' -->
        </ul>
    </section>
```

```

<section>
  <h2>Ficha del personaje seleccionado</h2>
  <div id="ficha-personaje">
    <!-- Muestra aquí los datos del objeto personaje seleccionado -->
  </div>
</section>

<!-- Escucha 'click' en la lista y usa event.target + dataset o el texto para localizar el objeto. -->
</body>
</html>

```

Ejercicio 6 – Contadores de clics (objeto como estado global)

Crea en JS un objeto 'contadorClics' con propiedades para cada botón (por ejemplo, rojo, verde y azul). Cada vez que el usuario pulse un botón, incrementa la propiedad correspondiente en el objeto y actualiza el texto de los elementos span con id 'clics-rojo', 'clics-verde' y 'clics-azul'. Añade también un botón 'Reset' que ponga a 0 todas las propiedades del objeto y actualice la vista.

HTML base del ejercicio:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejercicio 6 - Contadores de clics</title>
</head>
<body>
  <h1>Ejercicio 6 - Contadores de clics</h1>

  <button id="btn-rojo">Botón rojo</button>
  <button id="btn-verde">Botón verde</button>
  <button id="btn-azul">Botón azul</button>
  <button id="btn-reset">Reset</button>

  <p>Clics botón rojo: <span id="clics-rojo">0</span></p>
  <p>Clics botón verde: <span id="clics-verde">0</span></p>
  <p>Clics botón azul: <span id="clics-azul">0</span></p>

  <!-- Usa 'click' en cada botón para modificar el objeto contadorClics y refrescar los spans. -->
</body>
</html>

```

Ejercicio 7 – Gestor de notas rápidas (array de objetos + eventos de borrado)

Crea un array de objetos 'notas', donde cada nota tenga id, texto y fechaCreacion. Cuando el usuario escriba en el textarea y pulse 'Guardar nota', crea un nuevo objeto nota y guárdalo en el array. Pinta todas las notas en el contenedor con id 'contenedor-notas' mostrando su texto y fecha. Cada nota debe tener un botón 'Eliminar' que, al hacer clic, borre el objeto correspondiente del array y vuelva a repintar el listado.

HTML base del ejercicio:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejercicio 7 - Gestor de notas rápidas</title>
</head>
<body>
  <h1>Ejercicio 7 - Gestor de notas rápidas</h1>

  <textarea id="texto-nota" rows="4" cols="40" placeholder="Escribe tu nota aquí..."></textarea><br>
  <button id="btn-guardar-nota">Guardar nota</button>

  <h2>Mis notas</h2>
  <div id="contenedor-notas">
    <!-- Aquí se mostrarán las notas creadas a partir del array de objetos 'notas' -->
  </div>

```

```

<!-- Maneja 'click' en btn-guardar-nota y en los botones 'Eliminar' generados dinámicamente. -->
</body>
</html>

```

Ejercicio 8 – Mini inventario de inventario/stock (array de objetos + formularios)

Crea un array de objetos 'inventario', con propiedades id, nombre y stock. El formulario permite añadir unidades a un producto existente (por id) o crear uno nuevo si no existe. Cuando se envíe el formulario con id 'form-inventario', evita el comportamiento por defecto, actualiza el array y repinta la tabla con id 'tabla-inventario'. Muestra mensajes de éxito o error en el párrafo con id 'mensaje-inventario'.

HTML base del ejercicio:

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ejercicio 8 - Mini inventario</title>
</head>
<body>
    <h1>Ejercicio 8 - Mini inventario</h1>

    <form id="form-inventario">
        <label for="id-producto">ID producto:</label>
        <input id="id-producto" type="number">

        <label for="nombre-producto">Nombre:</label>
        <input id="nombre-producto" type="text">

        <label for="unidades-producto">Unidades a añadir:</label>
        <input id="unidades-producto" type="number">

        <button id="btn-actualizar-inventario" type="submit">Actualizar inventario</button>
    </form>

    <p id="mensaje-inventario"></p>

    <table id="tabla-inventario" border="1">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nombre</th>
                <th>Stock</th>
            </tr>
        </thead>
        <tbody>
            <!-- Rellena esta tabla a partir del array de objetos 'inventario' -->
        </tbody>
    </table>

    <!-- Maneja el evento 'submit' del formulario para modificar el array inventario y repintar la tabla. -->
</body>
</html>

```