A partir de este boletín el manejo de los eventos se realizará, única y exclusivamente, mediante el método addEventListener().

1. Crea una página web con un botón que capture los eventos click, mouseover y mouseout de un párrafo. Añade otro botón en el que al hacer clic se vayan eliminando los eventos capturados desde el primer botón. Puedes optar por crear los elementos HTML desde cero o por recuperarlos desde JS.

2. Dado el siguiente fragmento de código:

```
<script>
        var persona = {
            name: ['Rafael', 'Nadal'],
            age: 33,
            categoria: 'masculina',
            aficiones: ['tenis', 'restaurantes'],
            bio: function() {
                  alert(this.name[0] + ' ' + this.name[1] + ' tiene ' +
this.age + ' años y le gusta el ' + this.aficiones[0] + ' y el ' +
this.aficiones[1] + '.');
            },
            saludo: function() {
                        alert('Hola me llamo ' + this.name[0] + ' ' +
this.name[1] + '.');
            }
        };
    </script>
```

   a) Prueba el código a ver si funciona.
   b) Crea un par de botones y captura sus eventos click para que cada uno de ellos ejecute uno de los métodos del objeto.
   c) ¿Puedes sustituirlas por funciones flecha?
   d) Guarda el JSON en el almacenamiento del cliente y recupéralo en una página diferente.

3. Realiza el ejercicio del enlace: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/Image_gallery

A partir de aquí, los ejercicios se realizarán sobre el mismo archivo HTML que tenéis en Moodle. Aunque no lo diga explícitamente el enunciado, valora siempre que sea posible la opción de definir un solo manejador para el elemento contenedor y desde él actuar sobre todos sus hijos.

4. Cuando el usuario haga clic en cualquier botón dentro del div#contenedor-botones, cambiar el color de fondo de ese botón.

   Usa event.target para detectar qué botón fue presionado.

5. Cuando el usuario presione una tecla dentro del input#campo-texto, mostrar en el p#info-tecla el código de la tecla (event.key).

Usa keydown y la propiedad event.key.

6. En el div#listas, hay varias listas <ul>. Cuando el usuario haga clic en cualquier elemento <li>, mostrar en p#resultado-lista a qué <ul> pertenece ese <li>.

   Usa event.currentTarget para identificar la lista padre.

7. Dentro del div#textos, hay varios párrafos <p>. Cada vez que se haga clic en un párrafo, debe aumentar un contador de clics asociado a ese párrafo y mostrarse al lado del texto.

   Usa dataset para almacenar el conteo dentro de los párrafos.

8. Cuando el usuario mueva el mouse dentro del div#area-mouse, actualizar en p#posicion-mouse las coordenadas del cursor (clientX, clientY).

   Usa mousemove y event.clientX, event.clientY.

9. Cada vez que el usuario presione Enter en input#campo-texto, agregar el texto a la ul#lista-dinamica.

   Usa keydown y appendChild().

10. Cuando el usuario pase el mouse sobre cualquier li dentro de ul#lista-resaltable, cambiar su color de fondo. Al salir del elemento, volver al color original.

    Usa mouseover y mouseout en la lista padre con event.target.

11. Si el usuario hace doble clic en un li de ul#lista-eliminable, eliminar ese li.

    Usa dblclick y removeChild().

12. Permitir al usuario arrastrar y soltar elementos de ul#lista-drag para reordenarlos.

    Usa los eventos dragstart, dragover, drop y la propiedad event.dataTransfer.

13. Si el usuario presiona la tecla "r", "g" o "b", cambiar el fondo de la página a rojo, verde o azul respectivamente.

    Usa keydown y document.body.style.backgroundColor.