PRÁCTICA 4.01 Eventos

Normas de entrega

- En cuanto al **código**:
  - en la **presentación interna**, importan los **comentarios**, la claridad del código, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como **autodocumentado**. No será necesario explicar que es un `if` un `for` pero sí su funcionalidad. Hay que comentar las cosas destacadas y, sobre todo, las **funciones** y **clases** empleadas. La ausencia de comentarios será penalizada,
  - en la **presentación externa**, importan las leyendas aclaratorias, información en pantalla y avisos de ejecución que favorezcan el uso de la aplicación,
  - todo el código debe estar situado dentro del evento `window.onload = () => {};` o a través del evento `document.addEventListener("DOMContentLoaded", () => {});` ,
  - si no se especifica lo contrario, la información resultante del programa debe aparecer en la consola del navegador `console.log(información)`,
  - los ejercicios deben realizarse usando ***JavaScript* ES6** y usar el modo estricto (`use strict`) No se podrá utilizar *jQuery* ni cualquier otra biblioteca (si no se especifica lo contrario en el enunciado),
  - para el nombre de **variables**, **constantes** y **funciones** se utilizará *lowerCamelCase,*
  - para la asignación de eventos se utilizará `addEventListener()` indicando sus tres parámetros en su definición,
  - se usarán las funcionalidades `import` y `export` para crear bibliotecas de funciones temáticas (no debe haber declaración de funciones ni objetos en el documento principal).
- En cuanto a la **entrega** de los archivos que componen los ejercicios:
  - todos los ejercicios en **una carpeta** (creando las **subcarpetas** necesarias para documentación anexa como imágenes o estilos) cuyo nombre queda a discreción del discente,
  - el nombre de los ficheros necesarios para resolver el ejercicio será el número de ejercicio que contenga,
  - el código contendrá ejemplos de ejecución, si procede, y
  - la carpeta será comprimida en formato `ZIP` y será subida a **Aules** de forma puntual.

Ejercicio 1 - Saludar

Realiza un programa con dos botones **Comenzar Saludos** y **Parar saludos**. Al hacer clic en el primero lanza un `setInterval` para que cada dos segundos genere un `<h1>` con el texto `¡Hola Feo!`. El botón `Parar saludos` parará la secuencia.

Ejercicio 2 - Colorines

Haz un programa que al hacer doble clic en la pantalla del navegador cambie el fondo a un color aleatorio. Puedes generar los colores bien en hexadecimal `#5a6f12` o en `RGB rgb(255,255,255)`.

Deberás crear una función que genere un color aleatorio válido. Si eliges colores en hexadecimal utiliza este *array*:

```
var letras = ["a","b","c","d","e","f","0","1","2","3","4","5","6","7","8","9"];
```

Ejercicio 3 - Localizador

Haz un programa que, mediante eventos y el uso del objeto `event`, muestre en todo momento la posición actual del ratón en pantalla. Para mostrar la posición, modificaremos de forma dinámica un elemento `HTML` (por ejemplo un `<p>`) que muestre la posición actual del ratón (coordenadas **x** e **y**). Como siempre, la salida debe estar debidamente formateada.

Ejercicio 4 - Acordeón

Crea un acordeón compuesto por seis elementos que al hacer clic sobre los impares se muestre la información que está situada en el elemento par inmediatamente inferior a él. Al volver a hacer clic, si el elemento par que contiene la información se ve en pantalla deberá ocultarse. **No se podrá utilizar** el atributo `id` de los elementos `HTML`.

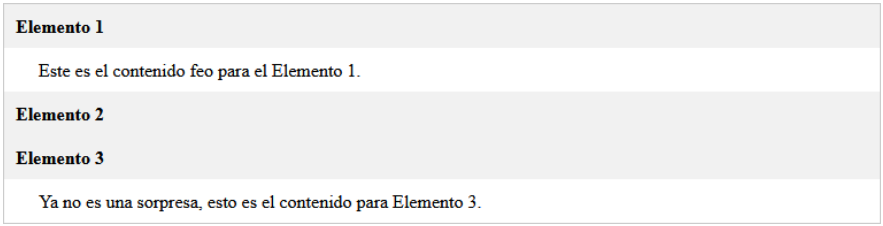

Figura 1: Esquema del acordeón.

Ejercicio 5 - Papelera de reciclaje

Escribe un programa que tenga una imagen de una bola de papel y una papelera vacía (o algo similar). Cuando se arrastre la bola de papel a la papelera vacía deberá cambiar la imagen de la papelera vacía a una papelera llena (o algo similar).