**Unidad 6: Eventos.**

**Ejercicios**

**INTRODUCCIÓN**

#### EJERCICIO 1: Ocultar texto

Agrega el código JavaScript al ejercicio 1 para que, al hacer clic en el button, desaparezca el texto del div con id="text".

#### EJERCICIO 2: Ocultar botón

Crea una copia del ejercicio anterior.

Modifica el código, para que, al hacer clic en el button, desaparezca el button.

#### EJERCICIO 3: Mover el balón por el campo

Agrega el código JavaScript al ejercicio 3 para que, al hacer clic en algún lugar del campo, el balón se mueva a ese lugar.

* El balón debe quedar centrado donde se ha hecho click, si es posible.
* El balón no debería salir del campo.
* Si la página se desplaza, el balón debería seguir correctamente colocado.

#### EJERCICIO 4: Crear un menú deslizante

Modifica el HTML/CSS del documento y agrega el código JavaScript al ejercicio 4.

Al hacer click sobre el texto Sweeties (click me)! , se debería mostrar/ocultar la lista que está debajo.

#### EJERCICIO 5: Agregar un botón para eliminar un mensaje

Agrega el código JavaScript al ejercicio 5 que inserte un botón de cierre en la esquina superior derecha de cada mensaje (div con class="pane").

Al hacer click sobre ese botón, se debe eliminar el div en que está el botón.

#### EJERCICIO 6: Carrusel

Agrega el código JavaScript al ejercicio 6 para que funcionen los botones “arrow prev” y “arrow next”. Para ello, basta con cambiar la propiedad margin-left del ul.

**DELEGACIÓN DE EVENTOS**

#### EJERCICIO 7: Agregar un botón para eliminar un mensaje (con delegación de eventos)

Modifica el ejercicio 5 para gestionar los eventos con el patrón de delegación de eventos, es decir, debe haber solamente un event listener en el contenedor de todos los div.

#### EJERCICIO 8: Mostrar/ocultar los nodos hijos con clicks

En este ejercicio tenemos un árbol. Se pide que al hacer clic sobre un texto, si tiene nodos hijos, los muestre/oculte.

La solución tiene dos partes.

1. Envuelve cada nodo de texto del árbol dentro de <span>.
2. Establece el manejador al nodo raíz del árbol y maneja los clics en aquellos nodos que sean <span>.

#### EJERCICIO 9: Tabla ordenable

Añade el JavaScript necesario para que, al hacer clic sobre los elementos <th>, la tabla se ordene por la columna correspondiente.

Cada <th> tiene un atributo que indica el tipo de datos de esa columna (sólo pueden ser "string" o "number"). La función de ordenamiento debe manejar el orden de acuerdo al tipo de datos.

Ojo! La tabla puede tener cualquier número de filas y de columnas.

**ACCIONES PREDETERMINADAS DEL NAVEGADOR**

#### EJERCICIO 10: Capturar enlaces en el elemento

Haz que, al hacer clic en los enlaces (<a>) que están dentro del elemento con id="contents", pregunten al usuario si realmente quiere ir a esa dirección, y si no quiere ir, no se irá.

Importante:

* Se debe utilizar delegación de eventos
* Dentro del <a> puede haber otra etiqueta (ej. <i>)

#### EJERCICIO 11: Galería de imágenes

Crea una galería de imágenes donde la imagen principal cambia al hacer clic en una miniatura.

Utiliza la delegación de eventos.

**EVENTOS DEL RATÓN**

#### EJERCICIO 12: Lista seleccionable

Modifica el archivo ej12.

Se trata de crear una lista donde los elementos son seleccionables, como en los administradores de archivos.

* Un clic en un elemento de la lista selecciona solo ese elemento (agrega la clase .selected), deselecciona todos los demás.
* Si se hace un clic con Ctrl (Cmd para Mac), el estado seleccionado/deseleccionado cambia para ese solo elemento, los otros elementos no se modifican.

Notas

PD: Para esta tarea, podemos suponer que los elementos de la lista son solo de texto. No hay etiquetas anidadas.

PPD: Evita la selección nativa del navegador del texto en los clics.

#### EJERCICIO 13: bombillaTradicional y bombillaW3C

Crear los siguientes scripts, partiendo del documento **combillaEnLinea.html**:

1. **bombillaTradicional**: que contendrá el resultado de transformar el modelo de eventos en línea al modelo de eventos tradicional.
2. **bombillaW3C**: que contendrá el resultado de transformar el modelo de eventos en línea o el modelo de eventos tradicional, al modelo de eventos del W3C.

#### EJERCICIO 14: calculadora

Partiendo del documento **calculadora.html**, añadir a nuestra calculadora dos botones más x3 y 1/x. Habrá que realizar la gestión de eventos correspondiente para que funcionen correctamente las dos operaciones nuevas.

También se añadirá un **campo informativo** mediante la etiqueta <h3> con contenido inicial "Información sobre el número". El contenido del campo se debe actualizar cada vez que se hace un cálculo en la calculadora. Su contenido debe ser "Información: el resultado de la operación *x2 | x3 | 1/x* es *número*."

#### EJERCICIO 15: mostrarInfo

A partir de la página web proporcionada, completar el código JavaScript para que se comporte de la siguiente forma:

1. Cuando se haga clic sobre el primer enlace, se oculte su sección relacionada.
2. Cuando se vuelva a hacer clic sobre el mismo enlace, se muestre otra vez esa sección de contenidos.
3. Completar el ejercicio con el resto de enlaces de la página para que su comportamiento sea idéntico al del primer enlace
4. Cuando una sección se oculte, debe cambiar el mensaje del enlace asociado

Si es necesario, modifica el HTML.

#### EJERCICIO 16: resaltarCeldaMouseOver

A partir del archivo HTML proporcionado, crear un script que resalte la celda sobre la que está el ratón.

Para resaltar la celda, se le asigna/desasigna la clase **highlight.**

El ejercicio se debe hacer utilizando mouseover y mouseout.

#### EJERCICIO 17: resaltarCeldaMouseEnter

A partir del archivo HTML proporcionado, crear un script que resalte la celda sobre la que está el ratón.

Para resaltar la celda, se le asigna/desasigna la clase **highlight.**

El ejercicio se debe hacer utilizando mouseenter y mouseleave.

#### EJERCICIO 18: zonasPulsar

A partir del archivo HTML proporcionado, crear un script que informe al usuario en qué zona de la pantalla ha hecho clic el ratón. Las zonas posibles son las siguientes: izquierda arriba, izquierda abajo, derecha arriba y derecha abajo, todas relativas al tamaño de la ventana del navegador.

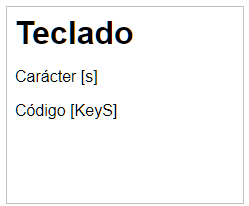
#### EJERCICIO 19: informacionEventos

Partiendo del archivo HTML proporcionado, crear un script que haga lo siguiente:

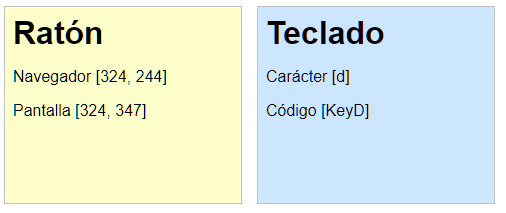
* Al mover el ratón en cualquier punto de la ventana del navegador, se muestre la posición del puntero respecto del navegador y respecto de la pantalla:



* Al pulsar cualquier tecla, se debe mostrar el valor de la tecla pulsada y su código asociado:



* Añadir la siguiente característica al script: cuando se pulsa un botón del ratón, el color de fondo del cuadro de mensaje debe ser amarillo (#FFFFCC) y cuando se pulsa una tecla del teclado, el color de fondo debe ser azul (#CCE6FF). Al volver a mover el ratón, el color de fondo vuelve a ser blanco.



#### EJERCICIO 20: parejas

Vas a crear un juego de memorizar parejas. Consiste en encontrar parejas en un tablero de 12 cartas, que contiene 6 cartas repetidas de los personajes de los Simpson. El juego consistirá en lo siguiente:

* La aplicación deberá tener una tabla con 3 filas y cuatro columnas de un color. Además, habrá un cuadro de texto con el valor 0 pero no modificable y un botón que permita reiniciar el juego.
* Cuando el usuario haga clic sobre una celda, se mostrará una imagen.
* Cuando el usuario haga clic sobre otra celda, se mostrará otra imagen.
* Si las dos imágenes son iguales, se mantendrá visibles y se añadirá 1 al cuadro de texto.
* Si las dos imágenes son diferentes, se ocultarán mostrando nuevamente el color inicial.
* Cuando se hayan completado las 6 parejas, se mostrará el mensaje: ¡¡GANASTE!!
* El botón de reiniciar el juego, cambia el orden de las cartas para volver a jugar. El siguiente método te puede ayudar, permite ordenar un array aleatoriamente:

array.sort(() => 0.5 - Math.random());