

# Ejercicios Unidad 5 – Eventos y formularios

Realiza cada ejercicio en una carpeta diferente dentro del directorio  
unidad 5

## 1. Ejercicio básico

Dado el siguiente HTML:

```
<table id="outside">
  <tr>
    <td id="t1">one</td>
  </tr>
  <tr>
    <td id="t2">two</td>
  </tr>
</table>
```

y el siguiente JavaScript:

```
function modifyText() {
  const t2 = document.getElementById("t2");
  const isNodeThree = t2.firstChild.nodeValue === "three";
  t2.firstChild.nodeValue = isNodeThree ? "two" : "three";
}

// Add event listener to table
const el = document.getElementById("outside");
el.addEventListener("click", modifyText, false);
```

1. Programar este código para que funcione con dos archivos (uno html y otro js) y explicar su funcionamiento ¿dónde debemos incluir la llamada al script? ¿por qué?
2. Añadir otro evento para que llame a la función consola que muestre un saludo por consola al pinchar la tabla
3. Añadir otro evento para que cambie el color del fondo de t1 a azul usando una función anónima
4. Crea las funciones:
  - a. *saludar1* que salude con alert al identificador de this

- b. `saludar2` que salude con alert al identificador de `e.target` (e debe pasarse como parámetro)
5. Crea los dos eventos correspondientes y pruébalos
6. ¿Cuál es la diferencia?
7. Crea una función `cambiaColor` para que cambie SOLO el color del texto de la fila pinchada de negro a azul o de azul a negro.
8. Modificar el ejercicio para asegurarnos de que los eventos se crean al cargarse la página usando `window.onload` y subiendo la llamada al script a la cabecera del HTML.

## 2. Sintaxis ES6

Añadir un evento a una imagen para que se agrande al hacer clic sobre ella usando la nueva sintaxis.

## 3. Cancelar eventos

1. Crea dos enlaces con href a otras páginas.
2. Cancela uno de los enlaces mediante `preventDefault()`

## 4. Tipo de evento de ratón

Dado el siguiente div:

```
<div id="contenidos" style="width:150px; height:60px; border:2px solid grey">  
  Sección de contenidos...  
</div>
```

Asociarle los siguientes eventos:

1. Cuando se pasa el ratón por encima se cambie el color del borde a verde
2. Cuando deja de pasarse el ratón por encima el borde pasará a ser de color gris.

Se usará una única función llamada **resalta** para los 2 eventos y se usará un **switch** para determinar el evento recibido

## 5. Propiedades de evento de teclado

Realizar una función para mostrar las diferentes propiedades del evento recibido al pulsar una tecla sobre un input.

Las propiedades se mostrarán del modo:

“la propiedad x tiene el valor y”

dentro de un div con identificador “info”.

Para ello usaremos el evento “onkeydown”.

[https://www.w3schools.com/jsref/obj\\_keyboardevent.asp](https://www.w3schools.com/jsref/obj_keyboardevent.asp)

## 6. Fases de un evento

Partiendo del siguiente código y sin modificarlo: añadir un evento en JavaScript para que al pinchar cualquier imagen de la galería se agrande a 500px. Probar su funcionamiento para demostrar la delegación de eventos modo “burbujeo” (por defecto).

ESTILOS:

```
<style>
  a img{
    width:200px;
  }
  .gallery-container{
    display: flex;
    flex-wrap: wrap;
    column-gap: 10px;
  }
</style>
```

CONTENEDOR:

```
<div class="gallery-container">
  <a href="#"> <p>foto1</p></a>
  <a href="#"> <p>foto2</p></a>
  <a href="#"> <p>foto3</p></a>
  <a href="#"> <p>foto4</p></a>
  <a href="#"> <p>foto5</p></a>
  <a href="#"> <p>foto6</p></a>
  <a href="#"> <p>foto7</p></a>
  <a href="#"> <p>foto8</p></a>
  <a href="#"> <p>foto9</p></a>
</div>
```

## 7. Eventos del ratón

Crear un script que informe al usuario en que zona de la pantalla ha pulsado el ratón. Las zonas definidas son las siguientes: izquierda arriba, izquierda abajo, derecha arriba y derecha abajo.

[https://www.w3schools.com/jsref/obj\\_mouseevent.asp](https://www.w3schools.com/jsref/obj_mouseevent.asp)

## 8. Eventos del ratón

Limita el número máximo de caracteres que puede contener un textArea a N caracteres.

Ayuda: onkeypress, return false/true;

## 9. Detección de llamador de evento

En esta página tienes dos bloques div a modo de botones, como ves en este código HTML de abajo. Debes escribir una única función Javascript (llamada pulsar) para manejar los eventos click de ambos botones. Cuando pulses con el ratón en cualquier botón deberá aparecer una ventana alert indicando el id del botón donde se ha pulsado.

```
<head>
  <style>
    .boton {
      display: block;
      width: 100px;
      cursor: pointer;
      margin: 10px;
      border: 1px solid black
    }
  </style>
</head>

<body>
  <div id="boton1" class="boton">Pulsar aquí</div>
  <div id="boton2" class="boton">Pulsar aquí</div>
</body>
```

## 10. Bloques ocultos

En esta página existen tres bloques ocultos. Debes crear un evento de teclado para que actúe cuando el usuario pulsa una tecla. Si pulsa la tecla 1 se abre el bloque con id ficha1, si pulsa 2 se abre el bloque ficha2 y si 3 se abre el de ficha3.

*Los bloques no se ven hasta que se pulsa la tecla de cada uno. Solo hay un bloque visible cada vez. Puedes usar las clases definidas y el objeto classList.*

```
<head>
  <style>
    .oculto {
      display: none;
    }
    .visible {
      display: block;
    }
  </style>
</head>
```

```
<body>
  <div class="oculto" id="ficha1">
    Hola, soy el bloque 1</div>
  <div class="oculto" id="ficha2">
    Hola, soy el bloque 2</div>
  <div class="oculto" id="ficha3">
    Hola, soy el bloque 3</div>
</body>
```

## 11. Scroll

Escribe un script que responda al evento scroll y que muestre un botón o bloque div en el fondo de la pantalla cuando deje de verse la línea superior del documento.

*El botón se queda quieto en el fondo de la pantalla. Recuerda que el alto de la ventana del navegador viene dado por **window.innerHeight** mientras que el desplazamiento vertical de la ventana está en **window.scrollY***

## 12. Input numérico

En este input solo deben aceptar valores que sean numéricos. Diseña un manejador de eventos para evitar que en el control se anote caracteres que no sean números.

Pista: puedes usar `event.preventDefault()`;

## 13. Ocultar secciones

A partir de la página web proporcionada en *ej13.html*, realizar el código JavaScript para que:

- A. Cuando se pinche sobre el primer enlace, se oculte su sección relacionada
- B. Cuando se vuelva a pinchar sobre el mismo enlace, se muestre otra vez esa sección de contenidos
- C. Completar el resto de enlaces de la página para que su comportamiento sea idéntico al del primer enlace
- D. Cuando una sección se oculte, debe cambiar el mensaje del enlace asociado.

## 14. Eventos de ratón teclado

Completar el código JavaScript proporcionado en *ej14.html* para que:

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 página:

## Ratón

Navegador [326, 432]

Página [326, 88]

Para mostrar los mensajes, utilizar la función `muestraInformacion()` deduciendo su funcionamiento a partir de su código fuente.

Al pulsar cualquier tecla, el mensaje mostrado debe cambiar para indicar el nuevo evento y su información asociada:

## Teclado

Carácter [a]

Código [97]

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, el color de fondo debe ser azul (#CCE6FF). Al volver a mover el ratón, el color de fondo vuelve a ser blanco.

## Ratón

Navegador [606, 457]

Página [606, 113]

## Teclado

Carácter [c]

Código [99]

## 15. Cuadro de diálogo

Agrega un botón a una página HTML que abra un cuadro de diálogo (intenta que sea a partir de un elemento `div` que tiene un estilo que parece flotar sobre la página). Al hacer clic en cualquier lugar fuera del cuadro de diálogo (pero no en el cuadro de diálogo en sí) debería cerrar el cuadro de diálogo.

## 16. Juego de coches

Crea un juego de carreras con los dos coches a partir del archivo ej16.html. Cuando se presiona el botón de “empezar carrera”, los dos coches deben moverse por la pantalla hasta que uno de ellos llegue al final de la pantalla. Cuando uno de los bloques llega al final, debe mostrar alerta “¡ganador!”.

## 17. Validación con blur

Agrega un detector de eventos de desenfoque a un elemento de entrada que verifique que el valor ingresado en el elemento sea una cadena de al menos nueve caracteres. Agrega un borde rojo al elemento si no lo está.

## 18. Validar formulario

Construir un validador para un formulario de registro que, dado el HTML que se proporciona en *ej18.html*, valide cada uno de los campos cuando el usuario ha terminado de introducir datos en cada uno de ellos, es decir, al perder el foco.

Si el campo no cumple las restricciones, se mostrará una alerta al usuario, pero se le permitirá seguir introduciendo datos en el resto de campos.

En el momento en el que el usuario envíe el formulario (evento submit), se validarán todos los campos y el formulario no se enviará si alguno de los campos no es válido. Las restricciones a cumplir son las siguientes:

- A. El nombre, email y comentarios son campos obligatorios.
- B. El campo email debe ser una dirección de email válida.
- C. El texto introducido en el campo de comentarios no debe exceder los 50 caracteres.
- D. El password debe tener una longitud mínima de 6 caracteres, y contener al menos una letra minúscula, una letra mayúscula y un dígito.

Hacer la validación y la muestra de información al usuario lo más completa posible.