

## Ejercicio 1: Cambiar el contenido y estilo de un elemento

### Enunciado:

Tienes un `<h1>` con el id `"titulo"` en tu HTML. Cuando el usuario haga clic en un botón, el texto del título debe cambiar a `"¡Nuevo Título!"` y su color debe volverse rojo.

### Código HTML de referencia:

html

Copiar

Editar

```
<h1 id="titulo">Título Original</h1>
<button id="cambiarTitulo">Cambiar Título</button>
```

### Solución en JavaScript:

js

Copiar

Editar

```
// 1. Seleccionamos los elementos
let titulo = document.getElementById("titulo");
let boton = document.getElementById("cambiarTitulo");

// 2. Agregamos un evento al botón
boton.addEventListener("click", function() {
  titulo.textContent = "¡Nuevo Título!"; // Cambia el texto
  titulo.style.color = "red"; // Cambia el color
});
```



### Explicación paso a paso:

1. **Seleccionamos** el `<h1>` y el `<button>` usando `getElementById()`.
2. **Agregamos un evento** al botón con `addEventListener()`, que escucha los clics.
3. **Cuando el usuario hace clic**, el código dentro de la función se ejecuta:
  - Se cambia el contenido de `titulo` con `.textContent = "¡Nuevo Título!"`.
  - Se cambia el color de `titulo` con `.style.color = "red"`.

## Ejercicio 2: Crear y eliminar elementos dinámicamente

### Enunciado:

Cuando el usuario haga clic en un botón, se debe agregar un nuevo párrafo dentro de un `<div id="contenedor">`. Si el usuario hace clic en otro botón, el párrafo agregado debe eliminarse.

### Código HTML de referencia:

html

Copiar

Editar

```
<div id="contenedor"></div>
<button id="agregar">Agregar Párrafo</button>
<button id="eliminar">Eliminar Párrafo</button>
```

### Solución en JavaScript:

js

Copiar

Editar

```
// 1. Seleccionamos los elementos
let contenedor = document.getElementById("contenedor");
let btnAgregar = document.getElementById("agregar");
let btnEliminar = document.getElementById("eliminar");

// 2. Variable para almacenar el párrafo agregado
let nuevoParrafo;

// 3. Evento para agregar un párrafo
btnAgregar.addEventListener("click", function() {
    nuevoParrafo = document.createElement("p"); // Crear <p>
    nuevoParrafo.textContent = "Este es un párrafo nuevo"; // Añadir texto
    contenedor.appendChild(nuevoParrafo); // Insertar en el <div>
});

// 4. Evento para eliminar el párrafo
btnEliminar.addEventListener("click", function() {
    if (nuevoParrafo) { // Verificar que existe
        contenedor.removeChild(nuevoParrafo);
        nuevoParrafo = null; // Resetear variable
    }
});
```



## Explicación paso a paso:

1. **Seleccionamos** el `<div>` donde se agregará el párrafo y los botones.
2. **Creamos una variable** (`nuevoParrafo`) para almacenar el párrafo que se añadirá.
3. **Cuando se hace clic en el botón "Agregar":**
  - Se **crea** un nuevo elemento `<p>` con `document.createElement("p")`.
  - Se **le añade texto** con `.textContent = "Este es un párrafo nuevo"`.
  - Se **inserta dentro del div** con `.appendChild(nuevoParrafo)`.
4. **Cuando se hace clic en el botón "Eliminar":**
  - Se **verifica** si `nuevoParrafo` existe.
  - Se **elimina del div** con `.removeChild(nuevoParrafo)`.
  - Se **resetea la variable** a `null` para que pueda volver a crearse correctamente.

## Ejercicio 3: Cambiar imagen al pasar el mouse

### Enunciado:

Tienes una imagen en tu HTML. Cuando el usuario pase el mouse por encima, la imagen debe cambiar a otra. Cuando el usuario quite el mouse, debe volver a la imagen original.

### Código HTML de referencia:

html

Copiar

Editar

```

```

## Solución en JavaScript:

```
js Copiar Editar  
  
// Seleccionamos la imagen  
let imagen = document.getElementById("imagen");  
  
// Guardamos las URLs de las imágenes  
let imagenOriginal = "imagen1.jpg";  
let imagenNueva = "imagen2.jpg";  
  
// Evento cuando el mouse entra en la imagen  
imagen.addEventListener("mouseover", function() {  
    imagen.src = imagenNueva;  
});  
  
// Evento cuando el mouse sale de la imagen  
imagen.addEventListener("mouseout", function() {  
    imagen.src = imagenOriginal;  
});
```

## Explicación:

1. **Seleccionamos** la imagen con `getElementById()`.
2. **Definimos** las URLs de la imagen original y la nueva.
3. **Agregamos un evento** `mouseover` para cambiar la imagen cuando el mouse entra.
4. **Agregamos un evento** `mouseout` para restaurar la imagen original cuando el mouse sale.

## Ejercicio 4: Contador de clics

### Enunciado:

Tienes un botón y un párrafo. Cada vez que el usuario haga clic en el botón, el número en el párrafo debe aumentar en 1.

### Código HTML de referencia:

html

Copiar

Editar

```
<p>Clicks: <span id="contador">0</span></p>
<button id="incrementar">Incrementar</button>
```

### Solución en JavaScript:

js

Copiar

Editar

```
// Seleccionamos elementos
let contador = document.getElementById("contador");
let boton = document.getElementById("incrementar");

// Inicializamos el contador en 0
let clicks = 0;

// Evento de clic en el botón
boton.addEventListener("click", function() {
  clicks++; // Incrementa el contador
  contador.textContent = clicks; // Actualiza el número en la pantalla
});
```

### Explicación:

1. Seleccionamos el `<span>` donde se mostrará el número y el botón.
2. Creamos una variable `clicks` para llevar la cuenta.
3. Cada vez que se hace clic en el botón, la variable aumenta y se actualiza el texto en el `<span>`.


## Ejercicio 5: Mostrar un mensaje al presionar una tecla

### Enunciado:

Cuando el usuario presione cualquier tecla, debe aparecer un mensaje mostrando qué tecla presionó.

### Código HTML de referencia:

html


 Copiar

 Editar

```
<p>Presiona una tecla: <span id="teclaPresionada"></span></p>
```

### Solución en JavaScript:

js

 Copiar

 Editar

```
// Seleccionamos el span donde mostraremos la tecla
let spanTecla = document.getElementById("teclaPresionada");

// Evento cuando se presiona una tecla
document.addEventListener("keydown", function(event) {
    spanTecla.textContent = event.key; // Muestra la tecla presionada
});
```

### Explicación:

1. **Seleccionamos** el `<span>` donde mostraremos la tecla.
2. **Escuchamos el evento** `keydown` en el `document`, que se activa cuando cualquier tecla es presionada.
3. **Mostramos la tecla presionada** en el `<span>` usando `event.key`.