

Ejercicio de Ventanas Emergentes en JavaScript:

1. Ventana de Alerta:

- Al cargar la página, muestra una ventana de alerta con el mensaje: "¡Bienvenido a nuestra página web!"

2. Ventana de Confirmación:

- Agrega un botón en la página con el texto "Confirmar".
- Al hacer clic en el botón, muestra una ventana de confirmación con el mensaje: "¿Estás seguro de continuar?"
- Si el usuario hace clic en "Aceptar", muestra una ventana de alerta con el mensaje: "Continuar con la operación".
- Si el usuario hace clic en "Cancelar", muestra una ventana de alerta con el mensaje: "Operación cancelada".

3. Ventana de Entrada (Prompt):

- Agrega otro botón en la página con el texto "Ingresar Nombre".
- Al hacer clic en el botón, muestra una ventana de entrada que solicita al usuario su nombre.
- Luego, muestra una ventana de alerta con un mensaje de saludo personalizado, por ejemplo: "¡Hola, [nombre]! Bienvenido de vuelta."

Ejercicio de Encabezado y Pie de Página con HTML, CSS y JavaScript:

Crear un sitio web con un encabezado y un pie de página que se carguen desde archivos HTML, CSS y JavaScript separados.

1. Encabezado:

- Crear un archivo HTML llamado **header.html** que contenga el encabezado del sitio web, como un logotipo y un menú de navegación.
- Estilizar el encabezado utilizando un archivo CSS llamado **header.css**.
- Opcionalmente, agregar interactividad al encabezado utilizando JavaScript en un archivo llamado **header.js**, por ejemplo, para desplegar un menú desplegable al hacer clic en un botón.

2. Pie de Página:

- Crear un archivo HTML llamado **footer.html** que contenga el pie de página del sitio web, como información de contacto y enlaces adicionales.
- Estilizar el pie de página utilizando un archivo CSS llamado **footer.css**.
- Opcionalmente, agregar interactividad al pie de página utilizando JavaScript en un archivo llamado **footer.js**, por ejemplo, para mostrar la fecha actual o un contador de visitas.

3. Página Principal:

- Crear un archivo HTML llamado **index.html** que sirva como la página principal del sitio web.
- Utilizar las etiquetas **<header>** y **<footer>** para incluir el encabezado y el pie de página respectivamente.
- Enlazar los archivos CSS y JavaScript correspondientes en el archivo HTML para aplicar estilos y funcionalidades al encabezado y al pie de página.

Ejercicio de Manipulación del DOM con Listas en HTML y JavaScript:

1. Lista de Elementos:

- Crea una página web con un título "Lista de Elementos".
- Incluye una lista desordenada (``) con tres elementos (``). Cada elemento debe contener un texto predeterminado, por ejemplo, "Elemento 1", "Elemento 2" y "Elemento 3", respectivamente.
- Junto a cada elemento de la lista, agrega un botón con el texto "Eliminar". Este botón será utilizado para eliminar el elemento correspondiente de la lista cuando se haga clic en él.

2. Agregar Elementos:

- Agrega un botón adicional debajo de la lista con el texto "Agregar Elemento".
- Cuando se haga clic en este botón, debe agregarse un nuevo elemento a la lista. Este nuevo elemento debe tener un número consecutivo al último elemento existente en la lista. Por ejemplo, si la lista tiene actualmente tres elementos, el nuevo elemento agregado deberá tener el texto "Elemento 4".
- Además, cada nuevo elemento de la lista debe tener un botón "Eliminar" asociado que permita eliminar ese elemento específico de la lista cuando se haga clic en él.

Ejercicio de Evento onload en HTML y JavaScript:

Crear una página web que muestre un mensaje de bienvenida cuando se cargue completamente. El mensaje de bienvenida debe aparecer automáticamente una vez que todos los elementos de la página estén listos para ser vistos por el usuario.

1. Mensaje de Bienvenida:

- Cuando la página se cargue completamente, mostrar un mensaje de bienvenida que incluya el nombre del usuario.
- Este mensaje debe aparecer en algún lugar visible de la página, como un encabezado (`<h1>`) o un párrafo (`<p>`).
- El mensaje debe incluir el texto "¡Bienvenido a nuestra página web!" seguido del nombre del usuario, por ejemplo, "¡Bienvenido, [nombre]!".

2. Captura del Nombre del Usuario:

- Antes de cargar completamente la página, pedir al usuario que ingrese su nombre en un cuadro de diálogo (**prompt**) al abrir la página.
- Utiliza JavaScript para capturar el nombre ingresado por el usuario y almacenarlo en una variable.

3. Evento onload:

- Utiliza el evento **onload** en el elemento `<body>` de la página para ejecutar una función de JavaScript cuando la página se cargue completamente.
- En esta función, muestra el mensaje de bienvenida utilizando el nombre del usuario ingresado anteriormente.

4. Estilo CSS (opcional):

- Agrega estilos CSS para mejorar la apariencia del mensaje de bienvenida y hacerlo más atractivo visualmente.

EJERCICIO EVENTOS

1. **Crear una página web básica:** Crear una página web simple utilizando HTML. Deben incluir un encabezado (`<h1>`), un párrafo (`<p>`), y una lista (`` y ``).
2. **Agregar interactividad con JavaScript:** Agrega un botón (`<button>`) y un área de texto (`<textarea>`). Cuando se haga clic en el botón, debe mostrarse un mensaje en el área de texto.
3. **Validación de formulario:** Crea un formulario simple con campos como nombre, correo electrónico y contraseña. Utiliza JavaScript para validar que los campos no estén vacíos antes de enviar el formulario. Si algún campo está vacío, muestra un mensaje de error.

Ejercicio de uso de `<noscript>` en HTML:

Crear una página web que muestre un mensaje de advertencia cuando JavaScript está deshabilitado en el navegador del usuario.

1. **Mensaje de Advertencia:**
 - Utilizar la etiqueta `<noscript>` para envolver un mensaje de advertencia.
 - El mensaje debe indicar al usuario que JavaScript está deshabilitado en su navegador y que es necesario habilitarlo para utilizar todas las funcionalidades del sitio web.
2. **Estilo CSS (opcional):**
 - Agregar estilos CSS para mejorar la apariencia del mensaje de advertencia y hacerlo más visible para el usuario.

Ejercicio de uso de `try` y `catch` en JavaScript:

Crear una función en JavaScript que tome un número como argumento y devuelva el resultado de dividir 10 por ese número. Sin embargo, debemos manejar el caso en el que el usuario ingrese el valor 0 como argumento para evitar errores de división por cero.

1. **Función de División:**
 - Crear una función llamada `divisionPorDiez` que tome un parámetro `numero`.
 - Dentro de la función, utilizar un bloque `try` para intentar dividir 10 por el número proporcionado.
 - Si la división es posible, devolver el resultado.
 - Si se produce un error debido a una división por cero, capturar el error utilizando un bloque `catch` y devolver un mensaje de error personalizado, como "¡Error! No se puede dividir por cero".
2. **Prueba de la Función:**
 - Llamar a la función `divisionPorDiez` con varios valores, incluido el valor 0, para probar su funcionamiento.
 - Mostrar el resultado de cada llamada a la función en la consola del navegador.

Ejercicio de Cronómetro con HTML, CSS y JavaScript:

Crear una página web que simule un cronómetro simple con las siguientes características:

1. Interfaz de Usuario:

- Diseñar una interfaz de usuario simple que muestre el tiempo transcurrido del cronómetro.
- Incluir botones de control para iniciar, detener y reiniciar el cronómetro.

2. Funcionalidad del Cronómetro:

- Al cargar la página, el cronómetro debe estar detenido y mostrar un tiempo inicial de "00:00:00".
- Al hacer clic en el botón "Iniciar", el cronómetro comenzará a contar el tiempo.
- Al hacer clic en el botón "Detener", el cronómetro se detendrá y mostrará el tiempo transcurrido.
- Al hacer clic en el botón "Reiniciar", el cronómetro volverá a cero y se detendrá.

3. Estilo CSS:

- Diseñar el aspecto visual del cronómetro utilizando CSS para hacerlo atractivo y fácil de leer.

4. Funcionalidades Adicionales (opcional):

- Agregar un botón de vuelta para registrar vueltas o tiempos parciales.
- Permitir que el usuario configure el cronómetro para iniciar desde un tiempo específico.

Ejercicio de Animación con HTML y JavaScript:

Crear una página web que presente una animación de un objeto moviéndose de un lado a otro de la pantalla de manera continua.

1. Diseño del Objeto:

- Utilizar HTML para crear un elemento (por ejemplo, un `<div>`) que representará el objeto animado.
- Estilizar el objeto utilizando CSS para que sea visible y atractivo.

2. Animación de Desplazamiento:

- Utilizar JavaScript para definir una animación que haga que el objeto se mueva de un lado a otro de la pantalla de manera continua.
- La animación debe comenzar desde un extremo de la pantalla y moverse hacia el otro extremo en un bucle infinito.
- Experimentar con diferentes duraciones y estilos de animación para lograr el efecto deseado.

3. Estilo y Presentación:

- Agregar estilos adicionales utilizando CSS para mejorar la apariencia del objeto y hacerlo más atractivo visualmente.
- Considerar la inclusión de otros elementos de diseño, como fondos o efectos visuales, para complementar la animación principal.