**Tic-Tac-Toe Infinito**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Tic-Tac-Toe Infinito</title>

<style>

body { text-align: center; font-family: Arial, sans-serif; }

.container { display: flex; justify-content: center; align-items: center; flex-direction: column; height: 100vh; }

.board { display: grid; grid-template-columns: repeat(auto-fit, 50px); gap: 5px; margin: 20px auto; }

.cell { width: 50px; height: 50px; border: 1px solid black; display: flex; justify-content: center; align-items: center; font-size: 24px; cursor: pointer; }

</style>

</head>

<body>

<div class="container">

<h1>Tic-Tac-Toe Infinito</h1>

<div class="board" id="board"></div>

<p id="status">Turno de: X</p>

</div>

<script>

const board = document.getElementById("board");

const status = document.getElementById("status");

const size = 3;

let grid = Array(size).fill(null).map(() => Array(size).fill(""));

let turn = "X";

function createBoard() {

board.innerHTML = "";

board.style.gridTemplateColumns = `repeat(${grid.length}, 50px)`;

grid.forEach((row, r) => {

row.forEach((cell, c) => {

const div = document.createElement("div");

div.classList.add("cell");

div.textContent = cell;

div.onclick = () => makeMove(r, c);

board.appendChild(div);

});

});

}

function makeMove(r, c) {

if (grid[r][c] !== "") return;

grid[r][c] = turn;

if (checkWin(r, c)) {

status.textContent = `¡${turn} gana!`;

board.onclick = null;

return;

}

turn = turn === "X" ? "O" : "X";

status.textContent = `Turno de: ${turn}`;

expandBoardIfNeeded();

createBoard();

}

function expandBoardIfNeeded() {

if (grid.some(row => row.includes(""))) return;

grid.forEach(row => row.push(""));

grid.push(Array(grid[0].length).fill(""));

}

function checkWin(r, c) {

const directions = [[1, 0], [0, 1], [1, 1], [1, -1]];

return directions.some(([dr, dc]) => {

let count = 1;

for (let d of [-1, 1]) {

let nr = r + dr \* d, nc = c + dc \* d;

while (grid[nr] && grid[nr][nc] === turn) {

count++;

nr += dr \* d;

nc += dc \* d;

}

}

return count >= 3;

});

}

createBoard();

</script>

</body>

</html>

**TRES EN RAYA**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Tres en Raya</title>

<style>

body { display: flex; justify-content: center; align-items: center; height: 100vh; flex-direction: column; font-family: Arial, sans-serif; }

.board { display: grid; grid-template-columns: repeat(3, 100px); grid-gap: 5px; }

.cell { width: 100px; height: 100px; display: flex; justify-content: center; align-items: center; font-size: 36px; border: 2px solid black; cursor: pointer; }

.cell.taken { cursor: not-allowed; }

</style>

</head>

<body>

<h1>Tres en Raya</h1>

<div class="board" id="board"></div>

<p id="status">Turno de: X</p>

<button onclick="resetGame()">Reiniciar</button>

<script>

const board = document.getElementById("board");

const status = document.getElementById("status");

let cells = Array(9).fill(null);

let turn = "X";

function createBoard() {

board.innerHTML = "";

cells.forEach((\_, index) => {

const div = document.createElement("div");

div.classList.add("cell");

div.onclick = () => makeMove(index);

board.appendChild(div);

});

}

function makeMove(index) {

if (cells[index] || checkWinner()) return;

cells[index] = turn;

updateBoard();

if (checkWinner()) {

status.textContent = `¡${turn} gana!`;

return;

}

turn = turn === "X" ? "O" : "X";

status.textContent = `Turno de: ${turn}`;

}

function updateBoard() {

document.querySelectorAll(".cell").forEach((cell, index) => {

cell.textContent = cells[index];

if (cells[index]) cell.classList.add("taken");

});

}

function checkWinner() {

const winningPatterns = [

[0, 1, 2], [3, 4, 5], [6, 7, 8],

[0, 3, 6], [1, 4, 7], [2, 5, 8],

[0, 4, 8], [2, 4, 6]

];

return winningPatterns.some(pattern =>

pattern.every(index => cells[index] === turn)

);

}

function resetGame() {

cells.fill(null);

turn = "X";

status.textContent = "Turno de: X";

createBoard();

}

createBoard();

</script>

</body>

</html>

**Calculadora**  
  
<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Calculadora</title>

<style>

body { display: flex; justify-content: center; align-items: center; height: 100vh; flex-direction: column; font-family: Arial, sans-serif; }

.calculator { display: grid; grid-template-columns: repeat(4, 60px); gap: 10px; background: #333; padding: 20px; border-radius: 10px; }

.display { grid-column: span 4; height: 50px; text-align: right; font-size: 40px; padding: 10px; background: #222; color: white; border-radius: 5px; }

button { width: 60px; height: 60px; font-size: 20px; cursor: pointer; border: none; border-radius: 5px; }

button.operator { background: orange; color: white; }

button.equal { background: green; color: white; }

button.clear { background: red; color: white; width: 270px; align-items: center;}

</style>

</head>

<body>

<h1>Calculadora</h1>

<div class="calculator">

<div class="display" id="display">0</div>

<button onclick="appendToDisplay('7')">7</button>

<button onclick="appendToDisplay('8')">8</button>

<button onclick="appendToDisplay('9')">9</button>

<button class="operator" onclick="appendToDisplay('/')">/</button>

<button onclick="appendToDisplay('4')">4</button>

<button onclick="appendToDisplay('5')">5</button>

<button onclick="appendToDisplay('6')">6</button>

<button class="operator" onclick="appendToDisplay('\*')">\*</button>

<button onclick="appendToDisplay('1')">1</button>

<button onclick="appendToDisplay('2')">2</button>

<button onclick="appendToDisplay('3')">3</button>

<button class="operator" onclick="appendToDisplay('-')">-</button>

<button onclick="appendToDisplay('0')">0</button>

<button onclick="appendToDisplay('.')">.</button>

<button class="equal" onclick="calculateResult()">=</button>

<button class="operator" onclick="appendToDisplay('+')">+</button>

<button class="clear" onclick="clearDisplay()">C</button>

</div>

<script>

let display = document.getElementById("display");

function appendToDisplay(value) {

if (display.textContent === "0") {

display.textContent = value;

} else {

display.textContent += value;

}

}

function clearDisplay() {

display.textContent = "0";

}

function calculateResult() {

try {

let numero = parseFloat(display.textContent);

if (numero % 1 !== 0) { // Verifica si el número tiene decimales

display.textContent = eval(display.textContent).toFixed(3);

} else {

display.textContent = eval(display.textContent);

}

} catch {

display.textContent = "Error";

}

}

</script>

</body>

</html>

**CARRUSEL IMÁGENES**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Carrusel de Imágenes</title>

<style>

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.carrusel {

position: relative;

width: 100%;

max-width: 400px;

height: 400px; /\* Altura fija de 400px \*/

overflow: hidden;

border-radius: 10px;

}

.carrusel-images {

display: flex;

transition: transform 0.5s ease-in-out;

height: 100%; /\* Asegura que las imágenes no sobrepasen la altura \*/

border-radius: 10px;

}

.carrusel-images img {

width: 100%;

height: 100%;

object-fit: contain; /\* Mantiene la imagen proporcional dentro del contenedor \*/

border-radius: 10px;

}

.boton {

position: absolute;

top: 50%;

transform: translateY(-50%);

background-color: rgba(0, 0, 0, 0.5);

color: white;

border: none;

padding: 15px;

cursor: pointer;

font-size: 24px;

}

.boton-prev {

left: 10px;

}

.boton-next {

right: 10px;

}

/\* Estilo adicional para hacer el carrusel más centrado \*/

.carrusel-container {

display: flex;

justify-content: center;

align-items: center;

width: 100%;

height: 100%;

}

</style>

</head>

<body>

<div class="carrusel-container">

<div class="carrusel">

<div class="carrusel-images">

<img src="imagen1.jpg" alt="Imagen 1">

<img src="imagen2.jpg" alt="Imagen 2">

<img src="imagen3.jpg" alt="Imagen 3">

</div>

<button class="boton boton-prev" id="prevBtn">&#10094;</button>

<button class="boton boton-next" id="nextBtn">&#10095;</button>

</div>

</div>

<script>

let currentIndex = 0;

const images = document.querySelectorAll('.carrusel-images img');

const totalImages = images.length;

const updateCarousel = () => {

const offset = -currentIndex \* 100;

document.querySelector('.carrusel-images').style.transform = `translateX(${offset}%)`;

};

document.getElementById('prevBtn').addEventListener('click', () => {

currentIndex = (currentIndex > 0) ? currentIndex - 1 : totalImages - 1;

updateCarousel();

});

document.getElementById('nextBtn').addEventListener('click', () => {

currentIndex = (currentIndex < totalImages - 1) ? currentIndex + 1 : 0;

updateCarousel();

});

// Opcional: Transición automática cada 3 segundos

setInterval(() => {

currentIndex = (currentIndex < totalImages - 1) ? currentIndex + 1 : 0;

updateCarousel();

}, 3000);

</script>

</body>

</html>

**ADIVINA NUMERO**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Juego Adivina el Número</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.game-container {

background-color: white;

padding: 20px;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

text-align: center;

width: 300px;

}

input[type="number"] {

padding: 10px;

font-size: 16px;

margin-top: 10px;

width: 60px;

border: 1px solid #ddd;

border-radius: 5px;

text-align: center;

}

button {

padding: 10px 15px;

font-size: 16px;

background-color: #28a745;

color: white;

border: none;

border-radius: 5px;

margin-top: 10px;

cursor: pointer;

}

button:hover {

background-color: #218838;

}

.result {

margin-top: 20px;

font-size: 18px;

font-weight: bold;

}

.reset-button {

background-color: #dc3545;

margin-top: 10px;

}

.reset-button:hover {

background-color: #c82333;

}

</style>

</head>

<body>

<div class="game-container">

<h1>Adivina el Número</h1>

<p>He elegido un número entre 1 y 100. ¡Intenta adivinarlo!</p>

<input type="number" id="guess" placeholder="" min="1" max="100">

<br>

<button onclick="checkGuess()">Adivinar</button>

<button class="reset-button" onclick="resetGame()">Resetear</button>

<div id="result" class="result"></div>

</div>

<script>

let randomNumber;

let attempts;

function resetGame() {

// Generar un nuevo número aleatorio y reiniciar los intentos

randomNumber = Math.floor(Math.random() \* 100) + 1;

attempts = 0;

// Limpiar el campo de texto y el resultado

document.getElementById('guess').value = '';

document.getElementById('result').textContent = '';

}

function checkGuess() {

const userGuess = parseInt(document.getElementById('guess').value);

const resultDiv = document.getElementById('result');

// Verificar si el usuario ha introducido un número válido

if (isNaN(userGuess) || userGuess < 1 || userGuess > 100) {

resultDiv.textContent = "Por favor, introduce un número entre 1 y 100.";

resultDiv.style.color = "red";

return;

}

attempts++; // Incrementar los intentos

// Comprobar si el número adivinado es correcto

if (userGuess === randomNumber) {

resultDiv.textContent = `¡Felicidades! Has adivinado el número ${randomNumber} en ${attempts} intentos.`;

resultDiv.style.color = "green";

} else if (userGuess < randomNumber) {

resultDiv.textContent = "El número es mayor. ¡Sigue intentando!";

resultDiv.style.color = "orange";

} else {

resultDiv.textContent = "El número es menor. ¡Sigue intentando!";

resultDiv.style.color = "orange";

}

}

// Inicializar el juego cuando la página carga

window.onload = resetGame;

</script>

</body>

</html>

**CONTADOR DE CLIKS**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Contador de Clics</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

}

.counter-container {

background-color: white;

padding: 20px;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

text-align: center;

width: 300px;

}

button {

padding: 10px 15px;

font-size: 16px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.count-display {

margin-top: 20px;

font-size: 24px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="counter-container">

<h1>Contador de Clics</h1>

<button onclick="incrementCounter()">Haz clic aquí</button>

<div class="count-display" id="count">Clics: 0</div>

</div>

<script>

// Inicializamos el contador

let clickCount = 0;

function incrementCounter() {

clickCount++;

document.getElementById('count').textContent = `Clics: ${clickCount}`;

}

</script>

</body>

</html>

**RELOJ DIGITAL**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Reloj Digital</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

}

.clock {

background-color: #333;

color: white;

font-size: 80px;

font-weight: bold;

padding: 20px;

border-radius: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

</style>

</head>

<body>

<div class="clock" id="clock">

00:00:00

</div>

<script>

function updateClock() {

const now = new Date();

const hours = String(now.getHours()).padStart(2, '0');

const minutes = String(now.getMinutes()).padStart(2, '0');

const seconds = String(now.getSeconds()).padStart(2, '0');

const timeString = `${hours}:${minutes}:${seconds}`;

document.getElementById('clock').textContent = timeString;

}

// Actualizar el reloj cada segundo

setInterval(updateClock, 1000);

// Inicializar el reloj

updateClock();

</script>

</body>

</html>

**PIEDRA PAPEL TIJERAS**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Juego de Piedra, Papel o Tijeras</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

flex-direction: column;

}

.game-container {

text-align: center;

}

.choice-buttons button {

padding: 20px;

font-size: 20px;

margin: 10px;

cursor: pointer;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

}

.choice-buttons button:hover {

background-color: #0056b3;

}

.result {

margin-top: 20px;

font-size: 24px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="game-container">

<h1>Juego de Piedra, Papel o Tijeras</h1>

<div class="choice-buttons">

<button onclick="playGame('piedra')">Piedra</button>

<button onclick="playGame('papel')">Papel</button>

<button onclick="playGame('tijeras')">Tijeras</button>

</div>

<div class="result" id="result"></div>

</div>

<script>

function playGame(userChoice) {

const choices = ['piedra', 'papel', 'tijeras'];

const computerChoice = choices[Math.floor(Math.random() \* 3)];

let result = '';

if (userChoice === computerChoice) {

result = '¡Es un empate!';

} else if (

(userChoice === 'piedra' && computerChoice === 'tijeras') ||

(userChoice === 'papel' && computerChoice === 'piedra') ||

(userChoice === 'tijeras' && computerChoice === 'papel')

) {

result = `¡Ganaste! La computadora eligió ${computerChoice}.`;

} else {

result = `¡Perdiste! La computadora eligió ${computerChoice}.`;

}

document.getElementById('result').textContent = result;

}

</script>

</body>

</html>

**CONTADOR PALABRAS**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Contador de Palabras</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

flex-direction: column;

}

.input-container {

margin-bottom: 20px;

text-align: center;

}

textarea {

width: 80%;

height: 100px;

padding: 10px;

font-size: 16px;

margin: 10px 0;

border-radius: 5px;

border: 1px solid #ccc;

}

.result {

font-size: 20px;

font-weight: bold;

}

button {

padding: 10px 20px;

font-size: 16px;

cursor: pointer;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

}

button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<h1>Contador de Palabras</h1>

<div class="input-container">

<textarea id="textInput" placeholder="Escribe tu oración aquí..."></textarea><br>

<button onclick="countWords()">Contar Palabras</button>

</div>

<div class="result" id="wordCountResult">Número de palabras: 0</div>

<script>

function countWords() {

const text = document.getElementById('textInput').value.trim();

if (text === "") {

document.getElementById('wordCountResult').textContent = "Número de palabras: 0";

} else {

const wordCount = text.split(/\s+/).length; // Divide el texto por espacios y cuenta las palabras

document.getElementById('wordCountResult').textContent = `Número de palabras: ${wordCount}`;

}

}

</script>

</body>

</html>

**CONVERSOR UNIDADES**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Conversor de Unidades</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

flex-direction: column;

}

.container {

width: 400px;

padding: 20px;

background-color: white;

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

text-align: center;

}

select, input {

width: 80%;

padding: 10px;

font-size: 16px;

margin: 10px 0;

border-radius: 5px;

border: 1px solid #ccc;

}

button {

width: 100%;

padding: 10px;

font-size: 16px;

cursor: pointer;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

}

button:hover {

background-color: #0056b3;

}

.result {

margin-top: 15px;

font-size: 18px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="container">

<h1>Conversor de Unidades</h1>

<label for="category">Selecciona categoría:</label>

<select id="category" onchange="updateUnits()">

<option value="distance">Distancia</option>

<option value="weight">Peso</option>

<option value="temperature">Temperatura</option>

<option value="currency">Moneda</option>

</select>

<br>

<label for="unitFrom">De:</label>

<br>

<select id="unitFrom"></select>

<br>

<label for="unitTo">A:</label>

<br>

<select id="unitTo"></select>

<br>

<label for="value">Valor:</label>

<br>

<input type="number" id="value" placeholder="Introduce un valor">

<br>

<br>

<button onclick="convert()">Convertir</button>

<div class="result" id="result"></div>

</div>

<script>

// Opciones de unidades para cada categoría

const units = {

distance: ['Kilómetros', 'Metros', 'Centímetros', 'Milímetros', 'Millas', 'Pies', 'Pulgadas'],

weight: ['Kilogramos', 'Gramos', 'Libras', 'Onzas'],

temperature: ['Celsius', 'Fahrenheit', 'Kelvin'],

currency: ['Dólar', 'Euro']

};

const conversionRates = {

currency: {

'Dólar': {

'Euro': 0.92,

'Dólar': 1

},

'Euro': {

'Dólar': 1.09,

'Euro': 1

}

},

distance: {

'Kilómetros': {

'Metros': 1000,

'Centímetros': 100000,

'Milímetros': 1000000,

'Millas': 0.621371,

'Pies': 3280.84,

'Pulgadas': 39370.1

},

'Metros': {

'Kilómetros': 0.001,

'Centímetros': 100,

'Milímetros': 1000,

'Millas': 0.000621371,

'Pies': 3.28084,

'Pulgadas': 39.3701

},

},

weight: {

'Kilogramos': {

'Gramos': 1000,

'Libras': 2.20462,

'Onzas': 35.274

},

'Gramos': {

'Kilogramos': 0.001,

'Libras': 0.00220462,

'Onzas': 0.035274

},

'Libras': {

'Kilogramos': 0.453592,

'Gramos': 453.592,

'Onzas': 16

},

},

temperature: {

'Celsius': {

'Fahrenheit': (value) => (value \* 9/5) + 32,

'Kelvin': (value) => value + 273.15

},

'Fahrenheit': {

'Celsius': (value) => (value - 32) \* 5/9,

'Kelvin': (value) => (value - 32) \* 5/9 + 273.15

},

'Kelvin': {

'Celsius': (value) => value - 273.15,

'Fahrenheit': (value) => (value - 273.15) \* 9/5 + 32

}

}

};

// Función para actualizar las unidades de conversión según la categoría seleccionada

function updateUnits() {

const category = document.getElementById('category').value;

const unitFromSelect = document.getElementById('unitFrom');

const unitToSelect = document.getElementById('unitTo');

const unitsList = units[category];

unitFromSelect.innerHTML = '';

unitToSelect.innerHTML = '';

unitsList.forEach(unit => {

const optionFrom = document.createElement('option');

optionFrom.value = unit;

optionFrom.textContent = unit;

unitFromSelect.appendChild(optionFrom);

const optionTo = document.createElement('option');

optionTo.value = unit;

optionTo.textContent = unit;

unitToSelect.appendChild(optionTo);

});

}

// Función para convertir las unidades

function convert() {

const category = document.getElementById('category').value;

const value = parseFloat(document.getElementById('value').value);

const unitFrom = document.getElementById('unitFrom').value;

const unitTo = document.getElementById('unitTo').value;

let result = 0;

if (!value || isNaN(value)) {

document.getElementById('result').textContent = 'Introduce un valor válido';

return;

}

if (category === 'currency') {

result = value \* conversionRates.currency[unitFrom][unitTo];

} else if (category === 'distance') {

result = value \* conversionRates.distance[unitFrom][unitTo];

} else if (category === 'weight') {

result = value \* conversionRates.weight[unitFrom][unitTo];

} else if (category === 'temperature') {

result = conversionRates.temperature[unitFrom][unitTo](value);

}

document.getElementById('result').textContent = `${value} ${unitFrom} = ${result.toFixed(2)} ${unitTo}`;

}

// Inicializar unidades por defecto

updateUnits();

</script>

</body>

</html>

**CALCULAR EDAD**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Calculadora de Edad</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f7f7f7;

flex-direction: column;

}

.container {

width: 400px;

padding: 20px;

background-color: white;

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

text-align: center;

}

input, button {

width: 80%;

padding: 10px;

font-size: 16px;

margin: 10px 0;

border-radius: 5px;

border: 1px solid #ccc;

}

button {

background-color: #007bff;

color: white;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

.result {

margin-top: 20px;

font-size: 18px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="container">

<h1>Calculadora de Edad</h1>

<label for="dob">Fecha de Nacimiento:</label>

<input type="date" id="dob" />

<button onclick="calculateAge()">Calcular Edad</button>

<div class="result" id="result"></div>

</div>

<script>

function calculateAge() {

const dob = document.getElementById('dob').value;

if (!dob) {

document.getElementById('result').textContent = 'Por favor, ingresa una fecha válida.';

return;

}

const birthDate = new Date(dob);

const today = new Date();

let age = today.getFullYear() - birthDate.getFullYear();

let month = today.getMonth() - birthDate.getMonth();

let day = today.getDate() - birthDate.getDate();

if (month < 0 || (month === 0 && day < 0)) {

age--;

month = (month + 12) % 12;

day = (day + 30) % 30; // Aprox. días en un mes

}

document.getElementById('result').textContent = `Edad: ${age} años, ${month} meses y ${day} días.`;

}

</script>

</body>

</html>