📋 Guía Completa: JavaScript de Formulario de Registro

🎯 ¿Qué hace este código?

Este código crea un **formulario interactivo** que:

1. Captura los datos que escribes (nombre, email, edad)
2. Los muestra en una tarjeta bonita
3. Limpia el formulario para otro registro

🔍 PASO 1: Capturar el Envío del Formulario

```
document.getElementById('formulario').addEventListener('submit', function(e) {
    e.preventDefault();
    console.log('¡Formulario enviado!');
```

¿Qué significa cada parte?

Código	Explicación	Analogía
document.getElementById('formulario')	Busca el formulario en tu página HTML	Como buscar un libro específico en una biblioteca
.addEventListener('submit', ...)	"Quédate escuchando, cuando envíen el formulario, avísame"	Como poner una alarma que suena cuando alguien toca el timbre
function(e)	La función que se ejecuta cuando ocurre el evento	Las instrucciones de qué hacer cuando suena la alarma
e.preventDefault()	Detiene la acción predeterminada (recargar la página)	Como decirle a alguien "espera, no hagas eso todavía"

🤔 ¿Por qué preventDefault()?

Sin él, la página se recargaría y perderías todo. Es como si llenases un vaso de agua y alguien lo tirara antes de que puedas beberlo.

📥 PASO 2: Obtener los Valores del Usuario

```
let nombre = document.getElementById('nombre').value;
let email = document.getElementById('email').value;
let edad = document.getElementById('edad').value;
```

Desglose línea por línea:

Línea 1: let nombre = document.getElementById('nombre').value;

Parte	Qué hace
let nombre	Crea una "caja" llamada nombre para guardar información
document.getElementById('nombre')	Encuentra el input con id="nombre" en el HTML
.value	Obtiene el texto que el usuario escribió dentro
=	Guarda ese texto en la caja nombre

Ejemplo práctico:

```
Si el usuario escribió "Juan" → nombre = "Juan"
Si el usuario escribió "María" → nombre = "María"
```

📝 PASO 3: Crear HTML Dinámico

I

```
let infoHTML = `
    <p><strong>📝 Nombre:</strong> ${nombre}</p>
    <p><strong>📧 Email:</strong> ${email}</p>
    <p><strong>🎂 Edad:</strong> ${edad} años</p>
    <p><strong>✅ Estado:</strong> Registrado exitosamente</p>
`;
```

🔑 **Conceptos clave:**

Template Literals (Plantillas de texto)

- Se escriben con **backticks** o comillas invertidas ` ` (no comillas simples ' ')
- Permiten escribir texto en múltiples líneas
- Permiten insertar variables con ${}

Interpolación con ${}:

```
let nombre = "Carlos";
let mensaje = `Hola ${nombre}`; // Resultado: "Hola Carlos"
```

Comparación con concatenación tradicional:

```
// ❌ Forma antigua (difícil de leer):
let infoHTML = '<p><strong>Nombre:</strong> ' + nombre + '</p>';

// ✅ Forma moderna (clara y fácil):
let infoHTML = `<p><strong>Nombre:</strong> ${nombre}</p>`;
```

📺 **PASO 4: Mostrar el HTML en la Página**

```
let contenedorInfo = document.getElementById('info-usuario');
contenedorInfo.innerHTML = infoHTML;
```

¿Qué está pasando?

Línea 1: Encuentra el contenedor donde queremos poner la información
Línea 2: Inserta el HTML que creamos dentro de ese contenedor

Visualización:

```
ANTES:
<div id="info-usuario"></div>  ← Vacío

DESPUÉS:
<div id="info-usuario">
    <p><strong>📝 Nombre:</strong> Juan</p>
    <p><strong>📧 Email:</strong> juan@email.com</p>
    ...
</div>  ← Lleno con información
```

👁️ PASO 5: Hacer Visible la Tarjeta

```
let tarjetaResultado = document.getElementById('tarjeta-resultado');
tarjetaResultado.classList.remove('tarjeta-hidden');
```

¿Cómo funciona?

classList es una lista de todas las clases CSS que tiene un elemento.

Método	Qué hace	Ejemplo
.add('clase')	Añade una clase	Hacer algo invisible
.remove('clase')	Quita una clase	Hacer algo visible
.toggle('clase')	Añade si no está, quita si está	Interruptor de luz

En este caso:

- La tarjeta tiene la clase tarjeta-hidden (CSS: display: none)
- Al quitarla, la tarjeta se vuelve visible
- Como quitar una sábana que cubría algo

🧹 PASO 6: Limpiar el Formulario

```
this.reset();
```

¿Qué es this?

En este contexto, this se refiere al **formulario** (el elemento que disparó el evento).

.reset() es un método que borra todos los campos del formulario, dejándolos como estaban al cargar la página.

Analogía: Es como borrar una pizarra después de usarla.

📊 Flujo Completo del Código

```
1. Usuario llena el formulario
   ↓
2. Usuario hace clic en "Registrar"
   ↓
3. Se dispara el evento 'submit'
   ↓
4. preventDefault() detiene la recarga
   ↓
5. Se obtienen los valores (nombre, email, edad)
   ↓
6. Se crea HTML con esos valores
   ↓
7. Se inserta el HTML en el contenedor
   ↓
8. Se muestra la tarjeta de resultado
   ↓
9. Se limpia el formulario
```

🎓 Conceptos JavaScript Aprendidos

Concepto	Para qué sirve	Nivel
addEventListener()	Escuchar eventos (clicks, envíos, etc.)	⭐ ⭐
preventDefault()	Detener comportamiento predeterminado	⭐ ⭐
getElementById()	Buscar elementos HTML por ID	⭐
.value	Obtener texto de un input	⭐
let	Crear variables	⭐
Template literals (` `)	Crear texto con variables de forma fácil	⭐ ⭐
${}	Insertar variables en template literals	⭐ ⭐
.innerHTML	Cambiar el contenido HTML de un elemento	⭐ ⭐
.classList.remove()	Quitar clases CSS	⭐ ⭐
this	Referirse al elemento que disparó el evento	⭐ ⭐ ⭐
.reset()	Limpiar un formulario	⭐
