





JAVASCRIPT EN FORMULARIO:

* Ingresar los datos correspondientes
* Clic en reservar turno
* validar los datos del formulario
* vaya automáticamente al link de WhatsApp y salga esta frase remplazando los datos del usuario:
* ¡Hola [Nombre]! 💅✨

Gracias por reservar tu turno. Estos son los datos de tu cita:

📱 Teléfono: [Número de teléfono]

💖 Servicio: [Servicio solicitado]

🌟 Extra: [Servicio extra]

📅 Fecha: [Fecha elegida]

🕒 Hora: [Hora seleccionada]

Para confirmar tu turno, es necesario abonar una seña obligatoria de $2.000.

Podés hacerlo por transferencia o en efectivo (consultame el medio que prefieras).

Tu turno quedará reservado una vez recibido el comprobante 📩

¡Gracias por elegirnos! 💕

RESPUESTA:

**Validación en HTML (básica)**

Ya estás usando parte de esto en tu código con required, que evita que se envíe el formulario si un campo está vacío.

Ejemplos:

<input type="text" id="nombre" name="nombre" required minlength="3">

<input type="tel" id="telefono" name="telefono" required pattern="[0-9]{10}">

required: campo obligatorio.

minlength="3": mínimo 3 caracteres.

pattern="[0-9]{10}": exactamente 10 dígitos (ideal para teléfono).

1. **Captura el evento "submit" del formulario** document.querySelector("form").addEventListener("submit", function (e) {

document.querySelector("form"): selecciona el primer formulario de la página.

.addEventListener("submit", ...): escucha el evento de envío Al hacer clic.

function (e) { ... }: función que se ejecuta cuando el usuario intenta enviar el formulario.

e es el evento que podemos usar para **detener el envío si hay errores**.

1. **Obtiene los valores ingresados por el usuario**

const nombre = document.getElementById("nombre").value.trim();

const telefono = document.getElementById("telefono").value.trim();

const fecha = document.getElementById("fecha").value;

const hora = document.getElementById("hora").value;

getElementById("..."): busca los campos del formulario por su ID.

.value: obtiene el valor ingresado.

.trim(): quita los espacios al principio y al final (solo en nombre y teléfono).

1. Crea un array vacío para guardar los errores

let errores = [];

Se usará para guardar los mensajes de error detectados durante la validación.

1. **Validación del nombre**

if (nombre.length < 3) {

errores.push("El nombre debe tener al menos 3 letras.");

}

Si el nombre tiene menos de 3 letras, guarda ese mensaje de error en el array errores.

1. Validación del teléfono

if (!/^\d{10}$/.test(telefono)) {

errores.push("El número de teléfono debe tener 10 dígitos.");

}

/^\d{10}$/ es una **expresión regular** que verifica si el teléfono tiene exactamente 10 números.

! ... .test(telefono): si **no** cumple con esa condición, se guarda el error.

1. Validación de la fecha

const hoy = new Date().toISOString().split("T")[0];

if (fecha < hoy) {

errores.push("La fecha no puede ser anterior a hoy.");

}

new Date().toISOString() da la fecha actual en formato 2025-06-11T....

.split("T")[0]: se queda solo con la parte de la fecha (2025-06-11).

Si la fecha elegida es menor que la de hoy, se considera inválida.

push() **agrega elementos al final de un array**.

Sirve para ir acumulando datos, como en tu caso, **mensajes de error**.

1. Validación de la hora

if (!hora) {

errores.push("Debe ingresar una hora.");

}

Si el campo de hora está vacío (!hora), guarda un mensaje de error.

1. Mostrar errores o permitir el envío

if (errores.length > 0) {

e.preventDefault(); // Evita que el formulario se envíe

alert(errores.join("\n")); // Muestra los errores en una ventana emergente

}

Si hay al menos un error, **no se envía el formulario**.

alert(errores.join("\n")): muestra todos los errores en un solo mensaje, cada uno en una línea nueva.

.length te dice **cuántos elementos hay en un array** o **cuántos caracteres tiene un string**.

**1. Creación del mensaje personalizado**

const mensaje = `¡Hola ${nombre}! 💅✨

Gracias por reservar tu turno. Estos son los datos de tu cita:

📱 Teléfono: ${telefono}

💖 Servicio: ${servicio}

🌟 Extra: ${servicioExtra}

📅 Fecha: ${fecha}

🕒 Hora: ${hora}

Para confirmar tu turno, es necesario abonar una seña obligatoria de $2.000.

Podés hacerlo por transferencia o en efectivo (consultame el medio que prefieras).

Tu turno quedará reservado una vez recibido el comprobante 📩

¡Gracias por elegirnos! 💕`;

* Usamos **template strings** (comillas invertidas `) para incluir variables directamente dentro del texto.
* Las expresiones ${nombre}, ${telefono}, etc., se reemplazan por los datos que el usuario ingresó en el formulario.
* Así, **mensaje** contiene un texto listo para enviar por WhatsApp con emojis, formato y salto de líneas (\n).

**2. Codificar el mensaje para la URL**

const mensajeCodificado = encodeURIComponent(mensaje);

* encodeURIComponent(...) convierte el texto en un formato válido para URLs.
* Transforma espacios, saltos de línea y caracteres especiales (como emojis) en códigos seguros (por ejemplo, %20 para espacio).
* Esto asegura que la URL de WhatsApp no se corte o genere errores al incluir el mensaje.

**3. Preparar el número de WhatsApp y la URL**

const numeroBelenza = "543515223831"; // formato: prefijo de país + número sin "+"

const url = `https://wa.me/${numeroBelenza}?text=${mensajeCodificado}`;

* numeroBelenza es tu número de WhatsApp en formato internacional: **código del país + código de área + número**. En este caso, Argentina (54) + Córdoba (351) + número local.
* La URL https://wa.me/…?text=… es la forma oficial para abrir WhatsApp con un mensaje precargado.
* mensajeCodificado se añade al final como parámetro text.

**4. Abrir WhatsApp con el mensaje**

window.open(url, "\_blank");

* window.open(...) abre la URL en una nueva pestaña o ventana.
* Al ejecutar esto, el navegador redirecciona al usuario a WhatsApp (web o app) con el mensaje ya listo para enviar.
* El usuario solo debe presionar "Enviar".