

### **Ejercitación 1** - Responder el siguiente cuestionario

- a. ¿Ventajas de HTML5?
- HTML5 aporta etiquetas semánticas (<header>, <nav>, <article>, <section>, <footer>), soporte nativo para multimedia (<video>, <audio>), APIs modernas (Canvas, Web Storage, Geolocation, Drag & Drop, WebSockets), mejor compatibilidad móvil y offline (Application Cache / Service Workers), validación de formularios en el cliente y mayor accesibilidad y SEO gracias a su semántica.
- b. ¿Por qué utilizarlo?
- Porque facilita el desarrollo moderno: permite integrar audio/video sin plugins, crear gráficos y juegos en el navegador, almacenar datos localmente, y comunicarse en tiempo real. Su semántica mejora la accesibilidad y el SEO; además está estandarizado y es compatible con navegadores actuales, por lo que tu sitio será más usable, ligero y mantenible (especialmente en dispositivos móviles).
- c. Nombre ventajas
- Etiquetas semánticas → mejor estructura y SEO.
  - Multimedia nativo (<video>, <audio>) → sin plugins.
  - Canvas y SVG → gráficos y juegos en el navegador.
  - APIs de almacenamiento (localStorage / sessionStorage) → datos en cliente.
  - Geolocation API → acceso a ubicación del usuario.
  - Form validation nativa → menos JS para validar.
  - Soporte mobile y responsive mejorado.
  - WebSockets → comunicación en tiempo real.
  - Service Workers / offline → PWA, funcionamiento sin conexión.
  - Mayor accesibilidad → mejores prácticas para lectores de pantalla.
  - Rendimiento y menor dependencia de librerías externas.

### **Ejercitación 2**

- a. ¿Qué formatos soporta ?
- Los formatos más comunes y sus tipos MIME son:
- MP3 — audio/mpeg (soporte universal en navegadores modernos).
  - AAC / MP4 audio — audio/mp4 o audio/aac (muy usado, buen soporte en Safari y móviles).
  - OGG (Vorbis) — audio/ogg (buena compatibilidad en Firefox y Chrome; menos en Safari/IE antiguos).
  - Opus — suele ir dentro de audio/ogg; codecs="opus" o audio/webm; codecs="opus" (excelente calidad a bajos bitrates; soportado en Chrome/Firefox).
  - WAV (PCM) — audio/wav o audio/x-wav (soporte general, sin compresión).
  - WebM audio — audio/webm (contiene Vorbis/Opus; soportado en Chrome/Firefox).
- b. Crear un elemento audio
- Ejemplo de src: <https://html5tutorial.info/media/vincent.mp3>

<audio **controls**>

<source src="https://html5tutorial.info/media/vincent.mp3" type="audio/mpeg">

Tu navegador no soporta el elemento de audio.

</audio>

Le agregué controls para que muestre los controles de pausa o por el estilo..

### **Ejercitación 3**

- a. Crear un elemento video

Ejemplo de src: [https://www.w3schools.com/html/mov\\_bbb.mp4/ogg](https://www.w3schools.com/html/mov_bbb.mp4/ogg)

```
<video controls width="640" height="360">
  <source src="https://www.w3schools.com/html/mov_bbb.mp4" type="video/mp4">
  Tu navegador no soporta el elemento <code>video</code>.
</video>
```

### **Ejercitación 4**

- a. Crear un formulario con un campo requerido.

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Ejercitación 4 - Formularios</title>
</head>
<body>
  <form id="formA">
    <label for="nombre">Nombre: </label>
    <input id="nombre" name="nombre" type="text" required placeholder="Ingresá tu
nombre">
    <button type="submit">Enviar</button>
  </form>
</body>
</html>
```

- b. Crear un formulario con un campo de tipo email y validar que funcione.

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Ejercitación 4 - Formularios</title>
</head>
<body>
  <form id="formB">
    <label for="email">Email: </label>
    <input id="email" name="email" type="email" required
placeholder="tu@ejemplo.com">
    <button type="submit">Enviar</button>
  </form>
</body>
<script>
  const form = document.getElementById("formB");
  form.addEventListener('submit', e => {
```

```

        e.preventDefault();
        const v = email.value.trim();
        if (!v) return console.log('El email es obligatorio');
        if (!email.checkValidity()) return console.log('Formato inválido');
        console.log('Email válido');
    });
</script>
</html>

```

- c. Crear un formulario con un campo de tipo fecha y validar que funcione.

```

<!doctype html>
<html lang="es">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width,initial-scale=1" />
<title>Validación Fecha</title>
<style>
.err{color:#c00} .ok{color:green}
input.invalid{outline:2px solid #f99}
</style>
</head>
<body>
<form id="formDate">
  <label for="fecha">Fecha (no puede ser futura)</label><br>
  <input id="fecha" name="fecha" type="date" required>
  <div id="msg" aria-live="polite"></div><br>
  <button type="submit">Enviar</button>
</form>

<script>
const form = document.getElementById('formDate');
const fecha = document.getElementById('fecha');
const msg = document.getElementById('msg');

form.addEventListener('submit', e => {
  e.preventDefault(); // quitar si querés enviar realmente
  msg.textContent = "";
  fecha.classList.remove('invalid');

  if (!fecha.value) {
    return show('La fecha es obligatoria', true);
  }
  const hoy = new Date();
  const sel = new Date(fecha.value + 'T00:00:00');
  if (sel > hoy) return show('La fecha no puede ser futura', true);

  show('Fecha válida', false);
});

```

```

    fecha.addEventListener('input', () => { msg.textContent="";
    fecha.classList.remove('invalid'); });

```

```

    function show(text, isError){
        msg.textContent = text;
        msg.className = isError ? 'err' : 'ok';
        fecha.classList.toggle('invalid', isError);
        if (isError) fecha.focus();
    }
</script>
</body>
</html>

```

- d. Crear un formulario con un campo de tipo color y validar que funcione.

```

<form id="formColor">
    <label for="color">Elegí un color (requerido)</label><br>
    <input id="color" name="color" type="color" required>
    <div id="msg" style="margin-top:6px;min-height:1.2em;"></div><br>
    <button type="submit">Enviar</button>
</form>

```

```

<script>
    const form = document.getElementById('formColor');
    const color = document.getElementById('color');
    const msg = document.getElementById('msg');

    form.addEventListener('submit', e => {
        e.preventDefault(); // quita esta línea si querés que el formulario se envíe
        realmente
        if (!color.checkValidity()) {
            msg.textContent = 'Por favor elegí un color.';
            msg.style.color = '#c00';
            color.focus();
            return;
        }
        msg.textContent = 'Color válido: ' + color.value;
        msg.style.color = 'green';
    });
</script>

```

- e. Crear un formulario con un campo de tipo number y validar que funcione. Además configurar valores mínimos y máximos permitidos.

```

<form id="formNumber">
    <label for="cantidad">Cantidad (entre 1 y 10)</label><br>
    <input id="cantidad" name="cantidad" type="number" required min="1" max="10"
    step="1" placeholder="1">
    <div id="msgNum" style="margin-top:6px;min-height:1.2em;"></div><br>

```

```
<button type="submit">Enviar</button>
</form>
```

```
<script>
```

```
const form = document.getElementById('formNumber');
const cantidad = document.getElementById('cantidad');
const msg = document.getElementById('msgNum');
```

```
form.addEventListener('submit', e => {
  e.preventDefault(); // quitar si querés que el formulario se envíe realmente
  msg.textContent = "";
  cantidad.classList.remove('invalid');
```

```
  // requerido
```

```
  if (!cantidad.value) {
    msg.textContent = 'La cantidad es obligatoria.';
    msg.style.color = '#c00';
    cantidad.focus();
    return;
  }
```

```
  // validación nativa de min/max
```

```
  if (!cantidad.checkValidity()) {
    if (cantidad.validity.rangeUnderflow) msg.textContent = `El valor debe ser ≥
    ${cantidad.min}.`;
    else if (cantidad.validity.rangeOverflow) msg.textContent = `El valor debe ser ≤
    ${cantidad.max}.`;
    else msg.textContent = 'Valor inválido.';
    msg.style.color = '#c00';
    cantidad.classList.add('invalid');
    cantidad.focus();
    return;
  }
```

```
  // todo ok
```

```
  msg.textContent = 'Cantidad válida: ' + cantidad.value;
  msg.style.color = 'green';
});
```

```
  // limpiar mensaje al escribir
```

```
  cantidad.addEventListener('input', () => { msg.textContent="";
  cantidad.classList.remove('invalid'); });
</script>
```