

RECOPIULATORIO DE ETIQUETAS Y ELEMENTOS PARA CONTENIDO MULTIMEDIA EN LA WEB

1. Imágenes en la Web

Las imágenes son esenciales para el diseño web. Su correcta implementación mejora la accesibilidad, la experiencia de usuario y la optimización del rendimiento.

1.1 Etiqueta ``

La etiqueta `` permite insertar imágenes en una página web.

Ejemplo básico:

```

```

- **src**: Ruta de la imagen.
- **alt**: Texto alternativo para accesibilidad y SEO.
- **width y height**: Permiten definir el tamaño de la imagen.

Ejemplo con atributos avanzados (HTML5):

```

```

- **loading="lazy"**: Optimización de carga diferida para mejorar el rendimiento.

1.2 Etiqueta `<picture>`

Permite definir múltiples versiones de una imagen para diferentes tamaños de pantalla.

Ejemplo:

```
<picture>
  <source media="(min-width: 800px)" srcset="imagen-grande.jpg">
  <source media="(min-width: 400px)" srcset="imagen-mediana.jpg">
  
</picture>
```

- **<source>**: Define imágenes alternativas según el tamaño de pantalla.
 - **srcset**: Permite seleccionar la mejor opción según la resolución.
-

2. Insertar Audio en una Web

La etiqueta **<audio>** permite la reproducción de archivos de audio en la web sin necesidad de complementos adicionales.

2.1 Etiqueta **<audio>**

Ejemplo básico:

```
<audio src="audio.mp3" controls></audio>
```

- **controls**: Muestra los controles de reproducción.

Ejemplo con múltiples formatos:

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
  <source src="audio.ogg" type="audio/ogg">
  <p>Tu navegador no soporta el audio.</p>
</audio>
```

- **<source>**: Permite incluir distintos formatos para compatibilidad con navegadores.
- **Mensaje alternativo** si el navegador no soporta la etiqueta.

Ejemplo con reproducción automática y sin controles:

```
<audio src="audio.mp3" autoplay loop></audio>
```

- **autoplay**: Inicia la reproducción automáticamente (puede ser bloqueado por navegadores).
 - **loop**: Reproduce el audio en bucle.
-

3. Insertar Video en una Web

El video es un elemento multimedia clave en la web, y HTML5 permite su integración sin necesidad de complementos externos.

3.1 Etiqueta **<video>**

Ejemplo básico:

```
<video src="video.mp4" controls width="600"></video>
```

- **controls**: Muestra los controles de reproducción.
- **width**: Define el ancho del video.

Ejemplo con múltiples formatos y imagen previa (**poster**):

```
<video controls poster="preview.jpg" width="600">  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogv" type="video/ogg">  
  <source src="video.webm" type="video/webm">  
  <p>Tu navegador no soporta el video.</p>  
</video>
```

- **poster**: Imagen que se muestra antes de la reproducción.
- **<source>**: Múltiples formatos para compatibilidad.

Ejemplo con reproducción automática y sin controles:

```
<video src="video.mp4" autoplay loop muted></video>
```

- **muted**: Se necesita en algunos navegadores para que **autoplay** funcione.
- **loop**: Repite el video de forma infinita.

3.2 Incrustar videos de YouTube o Vimeo

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/videoID"  
frameborder="0" allowfullscreen></iframe>
```

- **src**: URL del video en YouTube/Vimeo.
- **allowfullscreen**: Permite pantalla completa.

4. Compatibilidad y Formatos

Cada navegador tiene compatibilidad con ciertos formatos de imagen, audio y video. Aquí los más utilizados:

Multimedia **Formatos más compatibles**

Imágenes JPEG, PNG, GIF, SVG,
WebP

Audio MP3, OGG, WAV

Video MP4 (H.264), WebM, OGV

Se recomienda usar `<source>` para asegurar compatibilidad entre navegadores.

5. Recomendaciones para Optimización

- Usar **formatos optimizados** (WebP para imágenes, MP4/WebM para videos).
 - Carga diferida (**lazy loading**) en imágenes y videos pesados.
 - Comprimir **archivos multimedia** antes de subirlos.
 - Usar **alt** en imágenes para accesibilidad y SEO.
 - Evitar **autoplay** sin **muted**, ya que puede ser bloqueado.
-

Este recopilatorio abarca todos los elementos clave explicados en el documento.

¿Necesitas que amplíe algún punto o que lo resuma más? 😊

Aquí tienes un **recopilatorio completo de etiquetas y elementos** para implementar **animaciones y transiciones** en la web, basado en el documento y con ejemplos propios.

RECOPIULATORIO DE ANIMACIONES Y TRANSICIONES EN LA WEB

1. Formatos que soportan animaciones

Existen varios formatos que permiten incluir animaciones en la web.

1.1 GIF Animados (**.gif**)

- Son secuencias de imágenes que se reproducen en bucle.
- Soportan transparencias.

- Tienen un límite de **256 colores** (pueden perder calidad).
- Se insertan como imágenes comunes con ``:

```

```

💡 **Desventaja:** No permiten control avanzado sobre la animación.

1.2 Flash (.swf) [OBSOLETO]

- Era el estándar en animaciones antes de HTML5.
- Integraba imagen, video y sonido.
- **No es compatible con dispositivos Apple** y la mayoría de navegadores han dejado de soportarlo.

Ejemplo (⚠️ No recomendado):

```
<object type="application/x-shockwave-flash" data="animacion.swf" width="600" height="400"></object>
```

💡 **Alternativa:** Usar **HTML5 + CSS3 + JavaScript**.

2. Animaciones en HTML5 y CSS3

2.1 @keyframes: Definir la animación

- `@keyframes` permite establecer una **secuencia de movimiento** con distintos estados.

```
@keyframes mover {  
  from { left: 0px; }  
  to { left: 200px; }  
}
```

- **from** → Estado inicial

- **to** → Estado final
(También se pueden usar porcentajes: 0%, 50%, 100%...)
-

2.2 Aplicar la animación con **animation**

- Para aplicar la animación a un elemento se usa **animation**.

```
div {  
  position: relative;  
  width: 100px;  
  height: 100px;  
  background: red;  
  animation: mover 2s infinite alternate;  
}
```

Explicación de los valores usados:

- **mover** → Nombre de la animación.
 - **2s** → Duración total (2 segundos).
 - **infinite** → Se repite infinitamente.
 - **alternate** → La animación se invierte en cada repetición.
-

2.3 Propiedades de **animation**

Propiedad	Descripción
animation-name	Nombre de la animación (@keyframes).
animation-duration	Tiempo que dura la animación.

`animation-timing-function` Aceleración de la animación (`ease`, `linear`, `ease-in`, etc.).

`animation-delay` Retraso antes de comenzar.

`animation-iteration-count` Número de repeticiones (`infinite` para repetir siempre).

`animation-direction` Dirección de la animación (`normal`, `reverse`, `alternate`).

`animation-fill-mode` Controla cómo se mantiene el estilo antes y después de la animación.

📌 Ejemplo con todas las propiedades:

```
div {  
  width: 100px;  
  height: 100px;  
  background: blue;  
  animation: ejemplo 3s ease-in-out 1s infinite alternate;  
}  
  
@keyframes ejemplo {  
  from { transform: scale(1); }  
  to { transform: scale(1.5); }  
}
```

- ♦ La caja crecerá y disminuirá con un retraso de 1s.
-

3. Transiciones en CSS3

Las **transiciones** (**transition**) permiten que los cambios en las propiedades CSS ocurran de forma animada.

3.1 Ejemplo básico (**hover**)

```
button {  
  
    background: blue;  
  
    color: white;  
  
    padding: 10px;  
  
    transition: background 0.5s ease-in-out;  
  
}  
  
button:hover {  
  
    background: red;  
  
}
```

- ♦ Al pasar el mouse, el color cambiará de azul a rojo suavemente.

3.2 Propiedades de **transition**

Propiedad	Descripción
transition-property	Propiedad que se animará (all , background , width , etc.).
transition-duration	Duración de la transición (segundos o milisegundos).
transition-timing-function	Ritmo de la animación (ease , linear , etc.).

transition-delay

Retraso antes de que inicie la transición.

📌 Ejemplo con todas las propiedades:

```
div {  
  width: 100px;  
  height: 100px;  
  background: blue;  
  transition: width 2s ease-in-out 1s;  
}  
  
div:hover {  
  width: 300px;  
}
```

- ♦ La caja se agranda con suavidad tras 1s al pasar el cursor.

3.3 transition vs animation

Característica	transition	animation
Se activa con evento (hover, click)	✅ Sí	❌ No
Repetición automática	❌ No	✅ Sí (con <code>infinite</code>)
Mayor control sobre el movimiento	❌ No	✅ Sí (<code>@keyframes</code>)

Requiere `@keyframes`

✗ No

✓ Sí

💡 Si necesitas un cambio simple, usa `transition`. Para animaciones más complejas, usa `animation`.

4. Ejemplo Completo: Animación + Transición

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

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

  <title>Animación y Transición</title>

  <style>

    .caja {

      width: 100px;

      height: 100px;

      background: blue;

      position: relative;

      transition: background 1s ease-in-out;

      animation: mover 3s infinite alternate;

    }

    .caja:hover {

      background: red;

    }

    @keyframes mover {

      from { left: 0px; }
```

```
        to { left: 200px; }  
    }  
</style>  
</head>  
<body>  
    <div class="caja"></div>  
</body>  
</html>
```

Explicación:

- **Al pasar el mouse**, la caja cambia de azul a rojo (con **transition**).
- **Se mueve automáticamente** de izquierda a derecha (con **animation**).

◆ Conclusión

- ✓ **Usar transiciones** para cambios suaves y simples.
- ✓ **Usar animaciones** para movimientos más complejos.
- ✓ **Evitar Flash**, ya que está obsoleto.
- ✓ **Optimizar animaciones** para mejorar el rendimiento web.

Este recopilatorio resume los puntos clave del documento. ¿Te gustaría que agregara algo más o lo haga más resumido? 😊