

Formatos de Imagen

Ejercicios

- Instalar una aplicación de edición de imágenes y crear los diferentes formatos
- Crear un documento HTML que incluya una galería de imágenes y al ampliarse presente los diferentes tamaños
- Crear un documento HTML con los diferentes formatos y ver la diferencia de tiempos de carga en la página.

Formatos de video

Ejercicios

Descarga un video desde alguna de estas plataformas:

- [Coverr.co](https://coverr.co)
- <https://pixabay.com/es/videos/>
- <https://www.videvo.net/>
- Genera un documento HTML donde
- Convierte los videos en los distintos formatos (MP4, OGG, WebM)
- Modifica los archivos con el fin de realizar una conversión de buena calidad pero que pese lo menos posible.

Elemento <video> en HTML5

- **Width y height**, definen las dimensiones del reproductor de vídeo.
- **Src**, indica la fuente del vídeo. Este atributo puede ser reemplazado por el elemento <source> y su propio atributo src para declarar varias fuentes con diferentes formatos. En el siguiente ejemplo el navegador leerá la etiqueta <source> y decidirá qué archivo reproducir de acuerdo a los formatos que soporte.
- **Controls**, muestra los controles de vídeo que nos ofrece el navegador. Cuando se incluye el atributo el navegador activará su propia interface de control del vídeo. De esta forma el usuario podrá reproducir el vídeo, pausarlo, etc.
- **Autoplay**, al incluir este atributo, el navegador comienza a reproducir el vídeo automáticamente.
- **Loop**, al incluir este atributo, el navegador reproduce nuevamente el vídeo cuando llega a su final.
- **Poster**, provee una imagen que será mostrada mientras el usuario espera a que cargue el vídeo.
- **Preload**, puede recibir tres valores:
 - **none**: el vídeo no se cachea.
 - **metadata**: recomienda al navegador que capture información acerca de la fuente (dimensión, duración, etc.).
 - **auto**: descarga del archivo lo más pronto posible.

Ejercicios

Mediantes los videos generados en el ejercicio anterior crea un documento HTML donde:

- Incluye los vieos en los formatos mp4, ogg, webm
- Indica los valores correspondientes en el atributo type

Formatos de audio

- <https://www.zedge.net/>
- <https://www.elongsound.com/>

Atributos elemento <audio>

- **Src**, indica la URL del archivo de audio. Este atributo puede ser reemplazado por el elemento <source> y su propio atributo src para declarar varias fuentes con diferentes formatos. En el siguiente ejemplo el navegador leerá la etiqueta <source> y decidirá qué archivo reproducir de acuerdo a los formatos que soporte.
- **Controls**, muestra los controles de audio que nos ofrece el navegador. Cuando se incluye el atributo el navegador activará su propia interface de control del audio. De esta forma el usuario podrá reproducir el audio, pararlo, etc.
- **Autoplay**, al incluir este atributo, el navegador reproduce el audio automáticamente.
- **Loop**, al incluir este atributo, el navegador reproduce nuevamente el audio cuando llega a su fin.
- **Preload**, puede recibir tres valores:
- **none**: el audio no se cachea.
- **metadata**: recomienda al navegador que capture información acerca de la fuente (duración).
- **auto**: descarga el archivo lo más pronto posible.

Ejercicio

- Crea un documento HTML e incluye los audios que generaste en el ejercicio anterior en dos formatos diferentes.

Métodos JavaScript

- **play()**: comienza a reproducir el medio desde el inicio, siempre que el medio no haya sido pausado previamente.
- **pause()**: pausa la reproducción.
- **load()**: carga el archivo del medio. Es útil para cargar el medio anticipadamente.
- **canPlayType(formato)**: para comprobar si el formato del archivo se soporta por el navegador

Propiedades HTML5

- **paused**: tiene valor "true" si la reproducción del medio está actualmente pausada o no ha comenzado.
- **ended**: tiene valor "true" si la reproducción del medio ha finalizado porque se llegó al final.
- **duration**: contendrá la duración del medio en segundos.
- **currentTime**: Esta propiedad de lectura/escritura contendrá un valor para informar sobre la posición en segundos en la cual el medio está siendo reproducido, o especifica una nueva posición donde continuar reproduciendo.
- **error**: tiene el valor del error ocurrido.
- **muted**: para desactivar (true) o activar (false) el sonido del vídeo.
- **volume**: para indicar el volumen del vídeo. El valor del volumen debe ser un número entre 0 y 1, por ejemplo si indicamos medio.volume = 0.3 estaríamos configurando el volumen al 30%.

Eventos

Existen diferentes eventos que nos permiten conocer la situación del medio:

- **progress**: progreso de la descarga del medio. La información estará disponible a través del atributo buffered.

- **canplaythrough:** se dispara cuando el medio completo puede ser reproducido sin interrupción.
- **ended:** se dispara cuando el reproductor llega al final del medio.
- **pause:** se dispara cuando el reproductor es pausado.
- **play:** se dispara cuando el medio comienza a ser reproducido.
- **error:** se dispara cuando ocurre un error.

Ejercicio

Se trata de utilizar lo aprendido hasta hora y crear un documento HTML en el que simule un reproductor de video. Tienes que contener lo siguiente:

- Botones play/pause/stop
- Botones de volumen
- Botón mute
- Botones de rebobinar y avanzar: se adelantará o retrasará 5 segundos.
- Selector para elegir entre los diferentes formatos de videos