

Homework - HTML Teaching Series

Perfect.

We now cover **HTML** `<audio>` and `<video>` properly — not just “how to play media”, but attributes, browser behavior, accessibility, formats, performance, and production rules.

These tags are **HTML-native media players** and are heavily used in **education platforms**, dashboards, blogs, and real products.

◆ HTML `<audio>` and `<video>`

(Native media embedding — attributes, behavior, real-world usage)

The image contains three side-by-side screenshots. The first screenshot shows a mobile device displaying an audio player with controls for play, volume, and seek. The second screenshot shows a code editor with an HTML file containing an `<audio>` tag and its source files. The third screenshot shows a video editing interface with options for poster images and other settings.

1 What `<audio>` and `<video>` Really Are (Mental Model)

`<audio>` and `<video>` embed **media players** directly into HTML, without plugins.

Important points:

- Browser provides **native controls**
- Media loads **separately** from HTML
- Media playback is **sandboxed**
- JavaScript can control playback, but HTML alone is sufficient

Think of them as:

“Built-in, accessible media players provided by the browser.”

2 Basic Syntax

◆ `<audio>` basic

html

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

◆ <video> basic

html

```
<video src="movie.mp4" controls></video>
```

Without `controls`, users **cannot play media unless JS is used.**

3 Core Attributes (COMMON TO BOTH)

These attributes apply to **both** `<audio>` and `<video>`.

◆ `controls` — User Controls (MOST IMPORTANT)

html

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

What it does

- Shows play / pause / volume / timeline
- Required for usability
- Required for accessibility

✗ Without controls → bad UX unless custom JS controls exist.

◆ `src` — Media Source

html

```
<video src="intro.mp4"></video>
```

Points:

- Can be local or external URL
 - Browser downloads media separately
 - Large files → performance impact
-

◆ `autoplay` — Automatic Playback (RESTRICTED)

html

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

Important browser rule

- Autoplay only works if muted
- Audio autoplay is blocked by most browsers

Used for:

- Background videos
- Silent previews

◆ muted — Start Without Sound

html

```
<video src="demo.mp4" muted></video>
```

Often combined with `autoplay`.

◆ loop — Replay Automatically

html

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

Used for:

- Background music
- Short clips
- Animations

◆ preload — Loading Hint

html

```
<audio src="song.mp3" preload="metadata"></audio>
```

Values:

- `none` → do not preload
- `metadata` → load duration/info only (recommended)

- `auto` → browser decides

Production rule:

Prefer `metadata` to save bandwidth.

◆ `controlslist` — Restrict Controls (Advanced)

html

```
<video controls controlslist="nodownload"></video>
```

Limits:

- Download button
- Playback speed
- Remote playback (browser-dependent)

Not fully standardized, but widely used.

4 `<video>`-Only Attributes

◆ `width` and `height` — Layout Stability

html

```
<video  
  src="lesson.mp4"  
  width="640"  
  height="360"  
  controls  
></video>
```

Why this matters:

- Prevents layout shift
 - Reserves space before video loads
 - Same reason as images
-

◆ `poster` — Thumbnail Image (VERY IMPORTANT)

html

```
<video  
    src="lesson.mp4"  
    poster="thumbnail.jpg"  
    controls  
></video>
```

What it does

- Shows image before playback
- Improves UX
- Avoids blank video box

Poster image **must have meaningful content.**

◆ `playsinline` — Mobile Behavior

html

```
<video playsinline></video>
```

Prevents full-screen takeover on mobile (especially iOS).

Used in:

- Embedded players
 - Social feeds
 - Dashboards
-

5 `<source>` — Multiple Formats (PRODUCTION STANDARD)

Browsers support **different codecs**.

You should provide **multiple formats**.

◆ Example: `<video>` with `<source>`

html

```
<video controls width="640" height="360">  
    <source src="video.mp4" type="video/mp4" />  
    <source src="video.webm" type="video/webm" />  
    Your browser does not support the video tag.  
</video>
```

Browser:

- Tries sources **top to bottom**
- Plays the first supported format

Same works for `<audio>`.

6 Accessibility (NON-NEGOTIABLE)

◆ Text fallback (MANDATORY)

html

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg" />
  Your browser does not support audio playback.
</audio>
```

◆ Captions & Subtitles (CRITICAL FOR VIDEO)

html

```
<video controls>
  <source src="lesson.mp4" type="video/mp4" />
  <track
    kind="subtitles"
    src="subtitles.vtt"
    srclang="en"
    label="English"
    default
  />
</video>
```

Why this matters:

- Deaf / hard-of-hearing users
- No-sound environments
- Accessibility compliance

7 Performance Considerations (Production-Level)

- Videos are **heavy**

- Always:
 - Set dimensions
 - Use `poster`
 - Avoid autoplay with sound
 - Use `preload="metadata"`
 - Consider lazy loading via JS if offscreen
-

8 Real-World Use Cases

`<audio>`

- Podcasts
- Voice messages
- Alerts
- Language learning apps

`<video>`

- Courses
- Tutorials
- Product demos
- Marketing pages
- Dashboards

HTML media tags are used even in React/Next apps.

9 Common Beginner Mistakes

✗ Autoplay with sound

Blocked by browsers.

✗ No controls

Unusable media.

✗ No poster for video

Bad UX.

✗ Large media without preload control

Performance issues.

10 Production Rules (MEMORIZE)

- Always include `controls`
- Always define video size
- Always provide fallback text
- Use `poster` for video
- Use captions for accessibility
- Avoid autoplay with sound

Senior developers review media usage carefully.

1 1 Interview Questions (Very Common)

1. Difference between `<audio>` and `<video>` ?
 2. Why does autoplay require `muted` ?
 3. Purpose of `poster` ?
 4. Why use `<source>` instead of `src` only?
 5. How do you make video accessible?
-

🧠 Mental Model to Lock In

HTML media tags embed players, not files.

Attributes control behavior, performance, and accessibility.

Media must be usable without JavaScript.

→ NEXT TOPIC (Remaining HTML)

We can continue with:

- HTML `<track>` (captions & subtitles deeper)
- HTML `<embed>` vs `<object>`
- HTML `<canvas>` (conceptual)
- HTML Accessibility Basics (ARIA intro)

Tell me which one you want next.