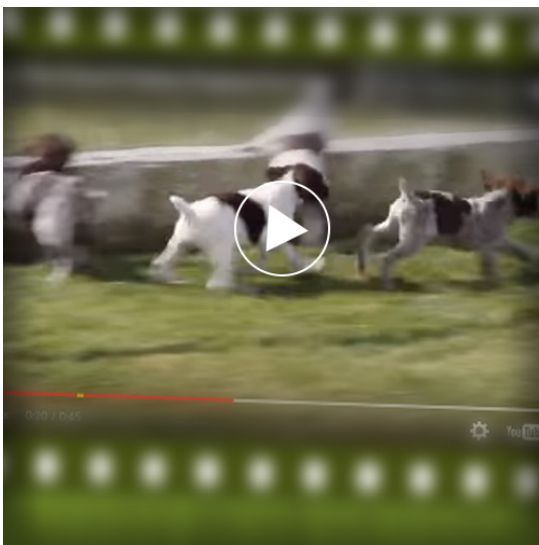




[Course](#) > [Modul...](#) > [5.3 Mul...](#) > Audio ...

Audio element

Audio element



`audio` and `video` are new HTML 5 elements that were highly anticipated. With HTML5 support for multimedia, this has become much easier, than previous methods.


Audio tag

You can use the `<audio>` tag to embed audio in your page.

1. `<audio src="sounds/flute.mp3">`
2. Your browser does not support the audio file.
3. `</audio>`

Any text within the `<audio>` tags will be displayed if the browser does not support the `audio` element. You should add such a message to provide better user experience for your page as it will be viewed in all types of devices and browsers.

The `audio` element has several attributes that can be used to configure audio playback. The following table lists the `audio` element's attributes:

Attribute	Description	Usage
<code>src</code>	Used to specify the URL of the audio file to embed. Values: <ul style="list-style-type: none"> absolute URL (file residing somewhere on the Web) relative URL (within your Web site) 	<code><audio src="sounds/flute.mp3"></code> <code></audio></code>
<code>controls</code>	Boolean attribute when specified provides controls for the user like play, pause, seek bar and volume 	<code><audio src="sounds/flute.mp3" controls></audio></code>
<code>loop</code>	Boolean attribute when specified loops media content	<code><audio src="sounds/flute.mp3" controls loop></audio></code>
<code>muted</code>	Boolean attribute when specified mutes media when playback begins	<code><audio src="sounds/flute.mp3" controls muted></audio></code>
<code>preload</code>	Allows author to communicate to the browser which settings will work best - audio should not be preloaded (none), only audio metadata is fetched (metadata), audio file can be downloaded when page loads (auto) values: none, metadata, auto	<code><audio src="sounds/flute.mp3" controls preload="auto"></audio></code>
<code>autoplay</code>	Boolean attribute when specified will automatically begin playing the source file as soon as it can without waiting for the entire audio file to finish downloading	<code><audio src="sounds/flute.mp3" controls autoplay></audio></code>

Here is an example code:

1. `<audio src="https://archive.org/download/Music_Calgary_PD_1/The.good.the.bad.and.the.ashy.mp3" controls loop muted preload="none">`
2. Your browser does not support the audio file.
3. `</audio>`

Output for the code above (try playing):

0:26 / 1:19

If you hit play and didn't hear anything, remember that we have added the `muted` attribute. So the audio will be muted when playback begins. Increase the volume to hear the music.

Audio file formats

Just like image file formats, not all audio file formats are supported by all browsers. You will want to use common audio file formats for browser compatibility ensuring the highest probability that your audio file will play.

The most common ones are MP3, WAV and Ogg.

[This page](#) under 'Browser Compatibility' lists the audio formats supported by the `audio` element and their browser support.

Here is some information regarding different types of audio formats and their compression techniques that can help you decide which audio format to choose apart from `audio` element and browser compatibility:

- There are three major groups of audio file formats - uncompressed (eg: WAV), lossless compressed (eg: MPEG-4, WMA Lossless) and lossy compressed (eg: Opus, MPC, AAC, WMA Lossy).
- In uncompressed audio file formats, no compression is applied to the audio file. The memory used for both sound and silence is the same though silence contains less information/data.

- In lossless compression, no data is lost but the file is compressed as silence is designed to take up very little space. Compared to uncompressed, lossless compression's compression ratio is approximately 2:1.
- Lossy compression provides the greatest compression by simplifying the data and removing some audio information resulting in some loss of quality. It is also the most popular. There are techniques in place to ensure that the parts of sound that is lost has little effect on quality. You can also select a range of compression rates. The larger the rate of compression, the bigger the loss in quality and smaller the file size.
- The audio format Ogg Opus has two parts to it. 'Ogg' is a digital container format. It is a specification that describes how different elements of data and metadata work in an audio file. However, it provides no information on how the data is compressed. So a program that opens a container file like 'Ogg' might not know how to decode it. 'Opus', the second part of the audio format, represents the encoding or decoding mechanism for that stream of audio. Opus is a lossy audio coding format.
- If you have an audio file in one format and wish to convert it to another, there are a lot of software applications available to help you do that.

Source element for multiple source files

The `source` element, also new in HTML5, serves the same purpose as the `src` attribute in an `audio` element. It is used to specify source files for the `audio` and `video` elements. Using the `source` element, you can specify multiple source files. The `<source>` tag is self-closing, therefore, it does not require a closing tag.

Example:

```
1. <audio controls>
2.   <source src="https://courses.edx.org/asset-
   v1:W3Cx+HTML5.0x+3T2016+type@asset+block@splash.wav" type="
   audio/wav">
3.   <source src="https://courses.edx.org/asset-
   v1:W3Cx+HTML5.0x+3T2016+type@asset+block@splash.mp3" type="
   audio/mpeg">
4.   Your browser does not support the audio element.
5. </audio>
```

Output for code above (try playing):

0:00 / 0:02

The advantage of providing multiple source files in different formats is that if the browser doesn't support the first format, it will try the second source file. The browser can select from the list based on its file format or codec support. In the code snippet example above, Internet Explorer does not support .wav files. So if you tried to play the file above in Internet Explorer, the browser would have tried to play .wav, failed and played the .mp3 version instead.

The following table lists the `<source>` element's attributes:

Attribute	Description	Usage
src	Specifies the URL of the media file	<code><source src="sounds/flute.mp3"></code>
type	<p>Specifies the internet media type, also known as the MIME type for the audio resource. A media type is an identifier for file formats and format contents transmitted over the internet like text and audio files.</p> <p>It consists of a type and a sub-type. Eg: "audio/mpeg" - audio is the type and mpeg is the subtype. It can also take optional parameters that can be specified after a semicolon - "audio/ogg; codecs=opus" means the audio is in the ogg format and uses the opus codec. If the browser supports the Ogg format but not the Opus codec, the audio file will not load.</p> <p>If the type attribute is not specified, the media type is retrieved from the server.</p>	<code><source src="sounds/flute.mp3" type="audio/mpeg"></code>

Knowledge check 5.3.1 (not graded)

0 points possible (ungraded)

How do you handle browsers that do not support the audio element?

- ☐ Don't do anything
- ☐ Add text in your audio element with a message stating that your browser does not support the audio element or audio format
- ☐ Do not include the 'autoplay' attribute so as to avoid an error being displayed
- ☐ Add multiple source elements

Submit

© All Rights Reserved