

HTML5 - Audio & Video

HTML5 features include native audio and video support without the need for Flash.

The HTML5 <audio> and <video> tags make it simple to add media to a website. You need to set **src** attribute to identify the media source and include a controls attribute so the user can play and pause the media.

Embedding Video

Here is the simplest form of embedding a video file in your webpage –

```
<video src = "foo.mp4" width = "300" height = "200" controls>
  Your browser does not support the <video> element.
</video>
```

The current HTML5 draft specification does not specify which video formats browsers should support in the video tag. But most commonly used video formats are –

- **Ogg** – Ogg files with Theora video codec and Vorbis audio codec.
- **mpeg4** – MPEG4 files with H.264 video codec and AAC audio codec.

You can use <source> tag to specify media along with media type and many other attributes. A video element allows multiple source elements and browser will use the first recognized format –

[Live Demo](#)

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <body>
```

```
    <video width = "300" height = "200" controls autoplay>
```

```
      <source src = "/html5/foo.ogg" type = "video/ogg" />
```

```
      <source src = "/html5/foo.mp4" type = "video/mp4" />
```

```
      Your browser does not support the <video> element.
```

```
    </video>
```

```
  </body>
```

```
</html>
```

This will produce the following result –



AD

Video Attribute Specification

The HTML5 video tag can have a number of attributes to control the look and feel and various functionalities of the control –

Sr.No.	Attribute & Description
1	autoplay This Boolean attribute if specified, the video will automatically begin to play back as soon as it can do so without stopping to finish loading the data.
2	autobuffer This Boolean attribute if specified, the video will automatically begin buffering even if it's not set to automatically play.
3	controls If this attribute is present, it will allow the user to control video playback, including volume, seeking, and pause/resume playback.
4	height This attribute specifies the height of the video's display area, in CSS pixels.
5	loop This Boolean attribute if specified, will allow video automatically seek back to the start after reaching at the end.
6	preload This attribute specifies that the video will be loaded at page load, and ready to run. Ignored if autoplay is present.
7	poster This is a URL of an image to show until the user plays or seeks.
8	src The URL of the video to embed. This is optional; you may instead use the <source> element within the video block to specify the video to embed.
9	width This attribute specifies the width of the video's display area, in CSS pixels.

Embedding Audio

HTML5 supports `<audio>` tag which is used to embed sound content in an HTML or XHTML document as follows.

```
<audio src = "foo.wav" controls autoplay>
  Your browser does not support the <audio> element.
</audio>
```

The current HTML5 draft specification does not specify which audio formats browsers should support in the audio tag. But most commonly used audio formats are **ogg**, **mp3** and **wav**.

You can use `<source>` tag to specify media along with media type and many other attributes. An audio element allows multiple source elements and browser will use the first recognized format –

[Live Demo](#)

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <body>
```

```
    <audio controls autoplay>
```

```
      <source src = "/html5/audio.ogg" type = "audio/ogg" />
```

```
      <source src = "/html5/audio.wav" type = "audio/wav" />
```

```
      Your browser does not support the <audio> element.
```

```
    </audio>
```

```
  </body>
```

```
</html>
```

This will produce the following result –



0:00 / 0:02

AD

Audio Attribute Specification

The HTML5 audio tag can have a number of attributes to control the look and feel and various functionalities of the control –

Sr.No.	Attribute & Description
1	autoplay This Boolean attribute if specified, the audio will automatically begin to play back as soon as it can do so without stopping to finish loading the data.
2	autobuffer This Boolean attribute if specified, the audio will automatically begin buffering even if it's not set to automatically play.
3	controls If this attribute is present, it will allow the user to control audio playback, including volume, seeking, and pause/resume playback.
4	loop This Boolean attribute if specified, will allow audio automatically seek back to the start after reaching at the end.
5	preload This attribute specifies that the audio will be loaded at page load, and ready to run. Ignored if autoplay is present.
6	src The URL of the audio to embed. This is optional; you may instead use the <source> element within the video block to specify the video to embed.

Handling Media Events

The HTML5 audio and video tag can have a number of attributes to control various functionalities of the control using JavaScript –

S.No.	Event & Description
1	abort This event is generated when playback is aborted.
2	canplay This event is generated when enough data is available that the media can be played.
3	ended This event is generated when playback completes.
4	error This event is generated when an error occurs.
5	loadeddata This event is generated when the first frame of the media has finished loading.
6	loadstart This event is generated when loading of the media begins.
7	pause This event is generated when playback is paused.
8	play This event is generated when playback starts or resumes.
9	progress This event is generated periodically to inform the progress of the downloading the media.
10	ratechange This event is generated when the playback speed changes.
11	seeked

	This event is generated when a seek operation completes.
12	seeking This event is generated when a seek operation begins.
13	suspend This event is generated when loading of the media is suspended.
14	volumechange This event is generated when the audio volume changes.
15	waiting This event is generated when the requested operation (such as playback) is delayed pending the completion of another operation (such as a seek).

Following is the example which allows to play the given video –

[Live Demo](#)

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <script type = "text/javascript">
```

```
      function PlayVideo() {
```

```
        var v = document.getElementsByTagName("video")[0];
```

```
        v.play();
```

```
      }
```

```
    </script>
```

```
  </head>
```

```
  <body>
```

```
    <form>
```

```
      <video width = "300" height = "200" src = "/html5/foo.mp4">
```

```
      Your browser does not support the video element.
```

```
    </video>
```

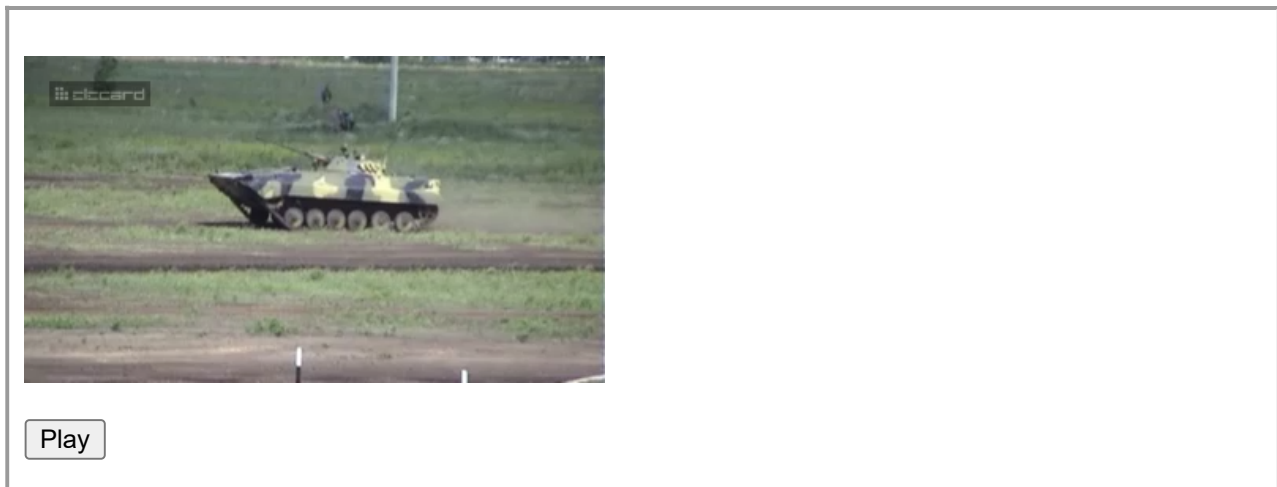
```
    <br />
```

```
    <input type = "button" onclick = "PlayVideo();" value = "Play"/>
```

```
  </form>
```

```
</body>  
</html>
```

This will produce the following result –



Configuring Servers for Media Type

Most servers don't by default serve Ogg or mp4 media with the correct MIME types, so you'll likely need to add the appropriate configuration for this.

```
AddType audio/ogg .oga  
AddType audio/wav .wav  
AddType video/ogg .ogv .ogg  
AddType video/mp4 .mp4
```