

HTML Audio/Video DOM

HTML Audio/Video Methods

Method	Description
addTextTrack()	Adds a new text track to the audio/video
canPlayType()	Checks if the browser can play the specified audio/video type
load()	Re-loads the audio/video element
play()	Starts playing the audio/video
pause()	Pauses the currently playing audio/video

HTML Audio/Video Properties

Property	Description
audioTracks	Returns an AudioTrackList object representing available audio tracks
autoplay	Sets or returns whether the audio/video should start playing as soon as it is loaded
buffered	Returns a TimeRanges object representing the buffered parts of the audio/video
controller	Returns the MediaController object representing the current media controller of the audio/video
controls	Sets or returns whether the audio/video should display controls (like play/pause etc.)
<code>crossOrigin</code>	Sets or returns the CORS settings of the audio/video
currentSrc	Returns the URL of the current audio/video
currentTime	Sets or returns the current playback position in the audio/video (in seconds)

HTML Audio/Video DOM

<u>defaultMuted</u>	Sets or returns whether the audio/video should be muted by default
<u>defaultPlaybackRate</u>	Sets or returns the default speed of the audio/video playback
<u>duration</u>	Returns the length of the current audio/video (in seconds)
<u>ended</u>	Returns whether the playback of the audio/video has ended or not
<u>error</u>	Returns a <code>MediaError</code> object representing the error state of the audio/video
<u>loop</u>	Sets or returns whether the audio/video should start over again when finished
<u>mediaGroup</u>	Sets or returns the group the audio/video belongs to (used to link multiple audio/video elements)
<u>muted</u>	Sets or returns whether the audio/video is muted or not
<u>networkState</u>	Returns the current network state of the audio/video
<u>paused</u>	Returns whether the audio/video is paused or not
<u>playbackRate</u>	Sets or returns the speed of the audio/video playback
<u>played</u>	Returns a <code>TimeRanges</code> object representing the played parts of the audio/video
<u>preload</u>	Sets or returns whether the audio/video should be loaded when the page loads
<u>readyState</u>	Returns the current ready state of the audio/video
<u>seekable</u>	Returns a <code>TimeRanges</code> object representing the seekable parts of the audio/video
<u>seeking</u>	Returns whether the user is currently seeking in the audio/video

HTML Audio/Video DOM

src	Sets or returns the current source of the audio/video element
startDate	Returns a Date object representing the current time offset
textTracks	Returns a TextTrackList object representing the available text tracks
videoTracks	Returns a VideoTrackList object representing the available video tracks
volume	Sets or returns the volume of the audio/video

HTML Audio/Video Events

Event	Description
abort	Fires when the loading of an audio/video is aborted
canplay	Fires when the browser can start playing the audio/video
canplaythrough	Fires when the browser can play through the audio/video without stopping for buffering
durationchange	Fires when the duration of the audio/video is changed
emptied	Fires when the current playlist is empty
ended	Fires when the current playlist is ended
error	Fires when an error occurred during the loading of an audio/video
loadeddata	Fires when the browser has loaded the current frame of the audio/video
loadedmetadata	Fires when the browser has loaded meta data for the audio/video
loadstart	Fires when the browser starts looking for the audio/video

HTML Audio/Video DOM

<u>pause</u>	Fires when the audio/video has been paused
<u>play</u>	Fires when the audio/video has been started or is no longer paused
<u>playing</u>	Fires when the audio/video is playing after having been paused or stopped for buffering
<u>progress</u>	Fires when the browser is downloading the audio/video
<u>ratechange</u>	Fires when the playing speed of the audio/video is changed
<u>seeked</u>	Fires when the user is finished moving/skipping to a new position in the audio/video
<u>seeking</u>	Fires when the user starts moving/skipping to a new position in the audio/video
<u>stalled</u>	Fires when the browser is trying to get media data, but data is not available
<u>suspend</u>	Fires when the browser is intentionally not getting media data
<u>timeupdate</u>	Fires when the current playback position has changed
<u>volumechange</u>	Fires when the volume has been changed
<u>waiting</u>	Fires when the video stops because it needs to buffer the next frame