<video/>

Adriano Flach
Matheus Bortolon

Intro

Uma alternativa ao Flash para a reprodução de mídia

Criada com o intuito de padronizar a reprodução de streaming de vídeo em todos os browsers, como já acontece com a tag <image />.

Incorporada à especificação do HTML na versão 5.

Tecnologia recente suportada apenas nas ultimas versões dos principais navegadores conhecidos.

Dependente do codec suportado no client onde é executando o streaming.

Browsers suportados

Element	![Chrome]	![Internet Explorer]	![Firefox]	![Safari]	![Opera]
<video>	4.0	9.0	3.5	4.0	10.5

Format	MIME-type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

Formatos suportados

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES from Firefox 21 from Firefox 30 for Linux	YES	YES
Safari	YES	NO	NO
Opera	YES From Opera 25	YES	YES

Múltiplos sources

```html
<video poster="movie.jpg" controls>
    <source src="movie.webm" type='video/webm; codecs="vp8.0, vorbis"'>
    <source src="movie.ogv" type='video/ogg; codecs="theora, vorbis"'>
    <source src="movie.mp4" type='video/mp4; codecs="avc1.4D401E, mp4a.40.2"'>
    <p>This is fallback content</p>
</video>
```

É possível informar opções com formatos diferentes

É importante informar o filetype
   para que o navegador escolha somente o video que é possivel reproduzir

Caso não tenha um formato suportado
   será apresentada a tag de texto que estiver dentro de vídeo

Atributos

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	*pixels*	Sets the height of the video player
loop	loop	Specifies that the video will start over again, every time it is finished

Atributos

Attribute		Value	Description
muted		muted	Specifies that the audio output of the video should be muted
poster		*URL*	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload		auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
src		*URL*	Specifies the URL of the video file
width		*pixels*	Sets the width of the video player

Eventos

```javascript
var vid = document.getElementById("myVideo");
vid.onerror = function() {
      alert("Error! Something went wrong");
};
```

É possível interceptar eventos do vídeo

Eventos

Event name	Dispatched when...
`loadstart`	The user agent begins looking for media data, as part of the resource selection algorithm.
`progress`	The user agent is fetching media data.
`suspend`	The user agent is intentionally not currently fetching media data, but does not have the entire media resource downloaded.
`abort`	The user agent stops fetching the media data before it is completely downloaded, but not due to an error.
`error`	An error occurs while fetching the media data.
`emptied`	A media element whose `networkState` was previously not in the `NETWORK_EMPTY` state has just switched to that state (either because of a fatal error during load that's about to be reported, or because the `load()` method was invoked while the resource selection algorithm was already running).
`stalled`	The user agent is trying to fetch media data, but data is unexpectedly not forthcoming.

Eventos

Event name	Dispatched when...
`play`	Playback has begun. Fired after the `play()` method has returned, or when the `autoplay` attribute has caused playback to begin.
`pause`	Playback has been paused. Fired after the `pause()` method has returned.
`loadedmetadata`	The user agent has just determined the duration and dimensions of the media resource
`loadeddata`	The user agent can render the media data at the current playback position for the first time.
`waiting`	Playback has stopped because the next frame is not available, but the user agent expects that frame to become available in due course.
`playing`	Playback has started.
`canplay`	The user agent can resume playback of the media data, but estimates that if playback were to be started now, the media resource could not be rendered at the current playback rate up to its end without having to stop for further buffering of content.

Eventos

Event name	Dispatched when...
`canplaythrough`	The user agent estimates that if playback were to be started now, the media resource could be rendered at the current playback rate all the way to its end without having to stop for further buffering.
`seeking`	The `seeking` IDL attribute changed to true and the seek operation is taking long enough that the user agent has time to fire the event.
`seeked`	The `seeking` IDL attribute changed to false.
`timeupdate`	The current playback position changed as part of normal playback or in an especially interesting way, for example discontinuously.
`ended`	Playback has stopped because the end of the media resource was reached.
`ratechange`	Either the `defaultPlaybackRate` or the `playbackRate` attribute has just been updated.
`durationchange`	The `duration` attribute has just been updated.
`volumechange`	Either the `volume` attribute or the `muted` attribute has changed. Fired after the relevant attribute's setter has returned.

Eventos

Referências

http://www.w3.org/wiki/HTML/Elements/video
http://www.w3schools.com/tags/tag_video.asp
http://www.w3.org/2010/05/video/mediaevents.html
http://www.w3schools.com/tags/av_event_error.asp
http://www.w3schools.com/tags/ref_av_dom.asp