

HTML5

VIDEO

HTML5 video allows us to **add video into our web pages** without the need for external plugins.

Background information

Containers and codecs

Video container formats are like zip files - they house other files. There are **three types of video containers formats.**

mgep-4 mpeg4v

Who: apple - patented

File format: .mpeg

WebM

Who: Google - free of patents

File format: .webm

Ogg

Who: Linux - free of patent

File format: .ogv

There are also **three different types of video codecs** (what's in the zip file)

h.264

Supported by Apple devices and browsers, IE9 native support, Andriod - heavily patented.

VP8 - WebM

Produced by Google - trying to be open source and competitive with h.264

Theora or “Ogg theora”

Not very good for compression

Browser support

<video> element support

Internet Explorer	9.0+
Firefox	3.5+
Safari	3.0+
Chrome	3.0+
Opera	10.5+
iPhone	1.0+
Andriod	2.0+

<video> codec support

Codecs	IE	Firefox	Safari	Chrome	Opera	iPhone	Android
Ogg Theora	No	3.5+	No	5.0+	10.5+	No	No
H.264 / MP4	9.0+	No	3.0+	No	No	3.0+	2.0+
WebM	9.0+	4.0+	No	6.0+	10.5+	No	2.3+

Mime types

Before creating any HTML5 video, you need to make sure that these mime types **are available on your server.**

```
AddType video/mp4 .mp4  
AddType video/webm .webm
```

New methods

The `<video>` element

The first step is to add the **<video>** element.

```
<video>
```

```
</video>
```

To create a <video>, you can use the **src attribute**. While this method is easy, it only allows you to source one video file format. For this reason, the “src” method is not recommended.

```
<video src="test.mp4">
```

```
</video>
```


The **controls attribute** is a boolean attribute that allows you to add browser controls to the video. These basic controls can be enhanced using JavaScript.

The controls attribute must always be included **if you want your users to be able to interact with the video.**

```
<video src="test.mp4" controls>
```

```
</video>
```

The **preload attribute** allows you to preload video.

```
<video src="test.mp4" controls preload>
```

```
</video>
```

The **autoplay attribute** allows you to autoplay video. This method is not recommended as it can cause accessibility issues - as well as being annoying!

```
<video src="test.mp4" controls autoplay>
```

```
</video>
```

The **loop attribute** allows you to loop (continuous play) video.


```
<video src="test.mp4" controls loop>
```

```
</video>
```

The **audio attribute** allows you to define if the video is muted or not. If the attribute is present, it must contain a value of “muted”. If the attribute is not present, the video’s audio will not be muted.

```
<video src="test.mp4" controls muted>
```

```
</video>
```

The **poster attribute** points to an image that is visible before the video is played. Ideally, it should be set to the same size as the video itself.

```
<video src="test.mp4" controls  
poster="poster.png">
```

```
</video>
```

The **width and height attributes** should be used where possible. These attributes allow browsers to set aside space for the video element. This improves page rendering speeds.

```
<video src="test.mp4" controls width="426"  
height="330">
```

```
</video>
```

The `<source>` element

The **<source> element** allows authors to specify multiple alternative media resources for media elements. The <source> element is child of video element.

The `<source>` element should be used **in preference to the “src” attribute** inside the `<video>` element. This allows browsers find the first available match and then plays only that video format.

```
<video controls poster="poster.png">  
  <source src="test.mp4">  
</video>
```

The **type attribute** defines the type of the media resource. This helps browsers determine if they can play the relevant media resource before fetching it. If specified, the value must be a valid MIME type.

```
<video controls poster="poster.png">  
  <source src="test.mp4" type="video/mp4">  
</video>
```

Ideally, we should always include the **two main different types of video** - so our video can be seen by the widest possible range of browsers.

```
<video controls poster="poster.png">  
  <source src="test.mp4" type="video/mp4">  
  <source src="test.webm" type="video/webm">  
</video>
```

The “.mp4” format **should always be placed first** so it plays in iPhone and iPad. These devices incorrectly render the first video they encounter.

Older browsers?

For browsers that do not support the `<video>` element, **the `<object>` or `<iframe>` elements can be used** to provide a Flash version of the video as a back-up.

The <object> and <iframe> elements **will be ignored by modern browsers** that support the <video> element

```
<video controls poster="poster.png">
  <source src="test.mp4" type="video/mp4">
  <source src="test.webm" type="video/webm">
  <iframe width="480" height="390"
src="http://www.youtube.com/embed/
hrwRG806f88" frameborder="0" allowfullscreen>
  </iframe>
</video>
```

You can also add additional backups, such as **text or images**.

```
<video controls poster="poster.png">
  <source src="test.mp4" type="video/mp4">
  <source src="test.webm" type="video/webm">
  <iframe width="480" height="390"
src="http://www.youtube.com/embed/
hrwRG806f88" frameborder="0" allowfullscreen>
  </iframe>
  <p>
    Text seen by older browsers.
  </p>
</video>
```

Using track

The `<track>` element

The **<track> element** defines text that you want to display along with the playing media file. Text may include subtitles, captions, descriptions, chapters or metadata.

The track element is a **void element**. It must be placed inside `<video>` or `<audio>` tags. The `<track>` element should appear after the `<source>` element.

```
<video controls>
  <source src="test.mp4" type="video/mp4">
  <source src="test.webm" type="video/webm">
  <track src="subtitles.srt"
        kind="subtitles"
        srclang="en"
        label="English_subs">
</video>
```

The **src attribute** is a required attribute that specifies the source address for the text file that contains the track data.

The value should be an **absolute or relative URL**. This means the files need to be put on a web server.

```
<video controls>
  <source src="test.mp4" type="video/mp4">
  <source src="test.webm" type="video/webm">
  <track src="subtitles.srt"
        kind="subtitles"
        srclang="en"
        label="English_subs">
</video>
```

The **kind attribute** defines the kind of track that is to be add. It may contain one of a number of values:

The **subtitles value** is used to display the dialogue being played in the video or audio file.

The **captions value** is a brief description accompanying the video being played. Used to inform users of some relevant information or even when the sound is not clear or inaudible.

The **srclang attribute** defines the language of the time-tracked data. This attribute must be included if the kind attribute is set to a value of subtitles.

The value of the srclang attribute must be a **valid BCP 47 language tag**. For instance, the value hi represents Hindi and en is used for English. There are about 8,000 language subtags available.



Russ Weakley

Max Design

Site: maxdesign.com.au

Twitter: twitter.com/russmaxdesign

Slideshare: slideshare.net/maxdesign

Linkedin: linkedin.com/in/russweakley