# HTML. Unit 8. Video and audio content.

## 1. Introduction

Now that we are comfortable with adding text and simple images to a web page, the next step is to start adding video and audio players to your HTML documents! In this unit we'll look at doing just that with the `video` and `audio` elements; we'll then finish off by looking at how to add captions to your videos, and as a bonus, how to add maps to your pages.

## 2. Video and audio on the web

Web developers have wanted to use video and audio on the Web for a long time, ever since the early 2000s when we started to have bandwidth fast enough to support any kind of video (video files are much larger than text or even images). In the early days, native web technologies such as HTML didn't have the ability to embed video and audio on the Web, so proprietary (or plugin-based) technologies like Flash (and later, Silverlight) became popular for handling such content. This kind of technology worked ok, but it had a number of problems, including not working well with HTML/CSS features, security and accessibility issues.

A native solution would solve much of this if implemented correctly. Fortunately, a few years later the HTML5 specification had such features added, with the `<video>` and `<audio>` elements, and some shiny new JavaScript APIs for controlling them. We'll not be looking at JavaScript here — just the basic foundations that can be achieved with HTML.

We won't be teaching you how to produce audio and video files — that requires a completely different skillset. For your own experimentation you can use some sample audio and video files as for example the ones available at "https://fernandoruizrico.com/test-audio-and-video/".

## 3. The <video> element

The `<video>` element allows you to embed a video very easily. In the same way as for the `<img>` element, the `src` (source) attribute contains a path to the video you want to embed (it works in exactly the same way). You can also allow the users to be able to control video and audio playback through the `controls` attribute to include the browser's own control interface. If this attribute is present, the browser will offer controls to allow the user to control video playback, including volume, seeking, and pause/resume playback. A really simple example looks like this:

```
<video src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm" controls></video>
```

You can insert a paragraph inside the video tag. This is called **fallback content**. It will be displayed if the browser accessing the page doesn't support the `<video>` element, allowing us to provide a fallback for older browsers. This can be anything you like; in this case, we suggest providing a direct link to the video file, so the user can at least access it some way regardless of what browser they are using:

```
<video src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm" controls>
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>
```

## 3.1. Width and height attributes

You can control the video size either with these attributes or with CSS. By setting the value in pixels of each attribute, you can choose to change either the width or the height of the video's display area. In both cases, videos maintain their native width-height ratio — known as the **aspect ratio**. If the aspect ratio is not maintained by the sizes you set, the video will grow to fill the space horizontally, and the unfilled space will just be given a solid background color by default. For example, to set the width of the video to `720px` and maintain the aspect ratio:

```
<video width="720"
       src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm"
       controls>
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>
```

### 3.1.1. Proposed exercise: Caminandes animated film

> *Create a web page with the previous example, and copy the code three times to show the same video three times in total. Change the width to set a different value for each video (1080, 720 and 480) and also change the src attribute accordingly. Finally, check the results in your browser (refresh your web page to assure that the browser is rendering the last changes in your code). Do not forget to include all necessary basic HTML tags and validate your code.*
>
> *Videos:*

> *– https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm*
>
> *– https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.webm*
>
> *– https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_480p.mp4*

```
<video width="1080"
       src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm"
       controls>
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>

<video width="720"
       src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.webm"
       controls>
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>

...
```

## 3.2. The poster attribute

You can set a URL for an image ( `poster` attribute) to be shown while the video is downloading. If this attribute isn't specified, nothing is displayed until the first frame is available, then the first frame is shown as the poster frame. This image can be easily set like this:

```
<video width="1080"
       src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm"
       poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/poster.jpg"
       controls>
```

```
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>
```

### 3.2.1. Proposed exercise: Caminandes animated film

> *Modify the code of the previous exercise, where you have the same video with three different sizes. Now you have to use the poster attribute to set three different images (you can find below the links to the posters, which are available at https://fernandoruizrico.com/test-audio-and-video/, or you can use any other images you like). When you finish, check the results in your browser (refresh your web page to assure that the browser is rendering the last changes in your code). Do not forget to include all necessary basic HTML tags and validate your code.*
>
> *Posters:*
>
> *– https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/poster.jpg*
>
> *– https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/mine.png*
>
> *– https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/share.png*

```
<video width="1080"
    src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm"
    poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/poster.jpg"
    controls>
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>

<video width="720"
    src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.webm"
```

```
        poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/mine.png"
        controls>
   <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
 </video>

 ...
```

## 3.3. Media file support in browsers

Usually some codecs are used to compress video and audio into manageable files, since raw audio and video are both exceedingly large. Each web browser supports an assortment of codecs, which are used to convert the compressed audio and video into binary data and back. Each codec offers its own positive and negative features affecting your decisions about which to use. In order to maximize the likelihood that your web site or app will work on a user's browser, you may need to provide each media file you use in multiple formats. If your site and the user's browser don't share a media format in common, your media simply won't play.

One additional thing to keep in mind: mobile browsers may support additional formats not supported by their desktop equivalents, just like they may not support all the same formats the desktop version does. On top of that, both desktop and mobile browsers *may* be designed to offload handling of media playback (either for all media or only for specific types it can't handle internally). This means media support is partly dependent on what software the user has installed.

So how do we do this? Take a look at the following updated example:

```
 <video width="1080"
        poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/mine.png"
        controls>
   <source
     src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm"
     type="video/webm">
   <source
     src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.mp4"
     type="video/mp4">
   <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
 </video>
```

Here we've taken the `src` attribute out of the actual `<video>` tag, and instead included separate `<source>` elements that point to their own sources. In this case the browser will go through the `<source>` elements and play the first one that it has the codec to support. First WebM is tried. If that can't be played, then MP4 is tried. Usually, including WebM and MP4 sources should be enough to play your video on most platforms and browsers these days. Also in the example, a fallback message is displayed if the video element isn't supported, but not if all sources fail.

Each `<source>` element also has a `type` attribute. This is optional, but it is advised that you include it. The `type` attribute contains the MIME type of the file, and browsers can use the `type` to immediately skip videos they don't understand. If `type` isn't included, browsers will load and try to play each file until they find one that works, which obviously takes time and is an unnecessary use of resources.

You can refer to the Mozilla's guide to media types and formats for help selecting the best containers and codecs for your needs, as well as to look up the right MIME types to specify for each.

### 3.3.1. Proposed exercise: Caminandes animated film

> *You have to modify the code of the previous exercise where you have three videos displaying the animated film "Caminandes", each one with a different size and a different poster. You have to insert now a couple of source elements so that all three videos contain links for both the webm and mp4 codecs (you can find the links in the example below and also at https://fernandoruizrico.com/test-audio-and-video/). Check the results in your browser and do not forget to add all necessary HTML tags and validate your code.*
>
> — Notice that the width of the video is set through the `width` attribute, and we have also used the `poster` attribute to set the image that will be displayed before the video is played:

```
<video width="1080"
       poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/mine.png"
```

```
        controls>
   <source
     src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.webm"
     type="video/webm">
   <source
     src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_1080p.mp4"
     type="video/mp4">
   <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
 </video>

 <video width="720"
        poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/share.png"
        controls>
   <source
     src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.webm"
     type="video/webm">
   <source
     src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.mp4"
     type="video/mp4">
   <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
 </video>

 ...
```

### 3.3.2. Proposed exercise: Sintel animated film

> *Create a new web page with the code below to show the "Sintel" animated film. After that, duplicate the code a couple of times to show three videos in total. Change the size of the videos to 1080, 720 and 480 and set the source elements accordingly (also remove the old `src` attribute, so that you get a similar code as the one shown in the example below). Finally change the poster image so that each video is displaying a different image when the page has just been loaded. You may find the links to both the videos and posters at [https://fernandoruizrico.com/test-audio-and-video/](https://fernandoruizrico.com/test-audio-and-video/), and also here:*

> *Videos (1080 px):*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/trailer/sintel_trailer-1080p.mp4*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/trailer/sintel_trailer-1080p.ogv*
>
> *Videos (720 px):*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/trailer/sintel_trailer-720p.mp4*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/trailer/sintel_trailer-720p.ogv*
>
> *Videos (480 px):*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/trailer/sintel_trailer-480p.mp4*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/trailer/sintel_trailer-480p.ogv*
>
> *Posters:*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/stills/poster.jpg*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/stills/scales.png*
>
> *– https://fernandoruizrico.com/examples/test-media/video/sintel/stills/dragon.png*

```
<video
  width="1080" controls
  poster="https://fernandoruizrico.com/examples/test-media/video/sintel/stills/scales.png">
  <source
    src="https://fernandoruizrico.com/examples/test-media/video/sintel/trailer/sintel_trailer-1080p.mp4"
    type="video/mp4">
  <source
    src="https://fernandoruizrico.com/examples/test-media/video/sintel/trailer/sintel_trailer-1080p.ogv"
    type="video/ogg">
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>

...
```

## 3.4. Other <video> features

There are a number of other features you can include when displaying an HTML video. Take a look at our next example:

```
<video controls width="720"
       autoplay loop muted preload="auto"
       poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/poster.jpg">
  <source src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.webm" type="video/webm">
  <source src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.mp4" type="video/mp4">
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>
```

### 3.4.1. The autoplay attribute

A boolean attribute. If specified, the video automatically begins to play back as soon as it can do so without stopping to finish loading the data.

Important: You are advised not to use autoplaying video (or audio) on your sites, because users can find it really annoying (some browsers will not even allow the audio or video start playing right away). Keep in mind that sites that automatically play audio (or videos with an audio track) can be an unpleasant experience for users, so should be avoided when possible. If you must offer autoplay functionality, you should make it opt-in (requiring a user to specifically enable it). However, this can be useful when creating media elements whose source will be set at a later time, under user control. See our autoplay guide for additional information about how to properly use autoplay.

### 3.4.2. The loop attribute

Makes the video (or audio) start playing again whenever it finishes. This can also be annoying, so only use if really necessary.

### 3.4.3. The muted attribute

A boolean attribute that indicates the default setting of the audio contained in the video. If set, the audio will be initially silenced. Its default value is `false` , meaning that the audio will be played when the video is played.

### 3.4.4. The preload attribute

This enumerated attribute is intended to provide a hint to the browser about what the author thinks will lead to the best user experience with regards to what content is loaded before the video is played. It may have one of the following values:

- `none` : Indicates that the video should not be preloaded.
- `metadata` : Indicates that only video metadata (e.g. length) is fetched.
- `auto` : Indicates that the whole video file can be downloaded, even if the user is not expected to use it.

### 3.4.5. Proposed exercise: Additional attributes

> *Following the previous examples, create a new web page with three videos you like, and set different values of the following attributes:* `loop` *,* `muted` *and* `preload` *. You can also try the* `autoplay` *attribute, although keep in mind that some browsers may have this feature disabled. Finally check the results in your browser and validate the code.*

```html
<video controls width="720"
       loop muted
       poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/poster.jpg">
  <source src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.webm" type="video/webm">
  <source src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.mp4" type="video/mp4">
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>

<video controls width="720"
       loop preload="metadata"
       poster="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/stills/poster.jpg">
  <source src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.webm" type="video/webm">
  <source src="https://fernandoruizrico.com/examples/test-media/video/caminandes-llamigos/caminandes_llamigos_720p.mp4" type="video/mp4">
  <p>Your browser doesn't support HTML5 video. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
</video>

...
```

### 3.4.6. Proposed exercise: Top ten videos

> *Create a web page to show the top ten videos you most like. You must set at least the* `width` *and* `poster` *attributes and also the* `source` *element to assure all of them are*

> *displayed in the right way as the previous examples.*
>
> — You can find some sample videos at: https://senkorasic.com/testmedia/, https://archive.org/details/BigBuckBunny_124, https://archive.org/download/ElephantsDream/, https://tools.woolyss.com/html5-audio-video-tester/, https://test-videos.co.uk/, http://losplayer.com/?page_id=1880, https://dafftube.org/video-links/, https://github.com/mediaelement/mediaelement-files, http://techslides.com/sample-webm-ogg-and-mp4-video-files-for-html5, https://sample-videos.com/, http://samples.mplayerhq.hu/, https://www.clipcanvas.com/a/video-clip-formats-and-codec-samples, https://peach.blender.org/download/, https://download.blender.org/peach/bigbuckbunny_movies/, https://orange.blender.org/download/, https://github.com/mdn/learning-area/tree/master/html/multimedia-and-embedding/video-and-audio-content, etc.

## 4. The <audio> element

The **HTML `<audio>` element** is used to embed sound content in documents. It may contain one or more audio sources, represented using the `src` attribute or the `<source>` element: the browser will choose the most suitable one. It works just like the `<video>` element, with a few small differences as outlined below. A typical example might look like so:

```
<figure>
    <figcaption>Trance 2. Instrumental Background Music.</figcaption>
    <audio
        controls
        src="https://fernandoruizrico.com/examples/test-media/audio/Twisterium-Trance2.wav">
        <p>Your browser doesn't support HTML5 audio. Click <a href="https://fernandoruizrico.com/test-audio-and-video/">here</a> instead.</p>
    </audio>
</figure>

<figure>
    <figcaption>Space battle. Space Fantasy Spot Effect.</figcaption>
    <audio
        controls
        src="http://bbcsfx.acropolis.org.uk/assets/07042219.wav">
        <p>Your browser doesn't support HTML5 audio. Click <a
```

```
href="http://bbcsfx.acropolis.org.uk">here</a> instead.</p>
      </audio>
  </figure>
```

The above example shows simple usage of the `<audio>` element. In a similar manner to the `<img>` and `<video>` elements, we include a path to the media we want to embed inside the `src` attribute. We can include other attributes to specify information such as whether we want it to loop, whether we want to show the browser's default audio controls, etc.

As with the `<video>` tag, the content inside the opening and closing `<audio></audio>` tags is shown as a fallback in browsers that don't support the element.

This takes up less space than a video player, as there is no visual component — you just need to display controls to play the audio. Other differences from HTML video are as follows:

- The `<audio>` element doesn't support the `width/height` attributes — again, there is no visual component, so there is nothing to assign a width or height to.
- It also doesn't support the `poster` attribute — again, no visual component.

### 4.1. Proposed exercise: Sample audios

> *Following the previous example, create a web page to insert at least ten figures containing audio samples. Update the caption of each figure to show a brief description of each audio file, and check the results in your browser. Do not forget to add all necessary HTML basic tags and validate your code.*
>
> — You can find thousands of audio samples at: [http://bbcsfx.acropolis.org.uk/](http://bbcsfx.acropolis.org.uk/)

## 5. The <iframe> element

By now you should really be getting the hang of embedding things into your web pages, including images, video and audio. At this point we will like to take somewhat of a sideways step, looking at an element that allow you to embed a wide variety of content types into your webpages: the `<iframe>` element.

## 5.1. Embed a video from Youtube

In this section we are going to jump straight into an active learning example to immediately give you a real idea of just what embedding technologies are useful for. The online world is very familiar with Youtube, but many people don't know about some of the sharing facilities it has available. Let's look at how Youtube allows us to embed a video in any page we like using an `<iframe>` :

1. First, go to Youtube and find a video you like.
2. Below the video, you'll find a *Share* button — select this to display the sharing options.
3. Select the *Embed* button and you'll be given some `<iframe>` code — copy this.

```
<figure>
    <figcaption>Nyan Cat [original].</figcaption>
    <iframe width="560" height="315" src="https://www.youtube.com/embed/QH2-TGUlwu4" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
</figure>
```

### 5.1.1. Proposed exercise: Top ten Youtube videos

> *Following the previous example, create a web page to insert at least ten figures containing videos from Youtube. Update the caption of each figure to show a brief description of each video, and check the results in your browser. Do not forget to add all necessary HTML basic tags and validate your code.*

## 5.2. Embed a map from Google Maps

You will also find quite interesting embedding a Google Map. Let's see how to do this with an example:

1. Go to Google Maps and find a map you like.
2. Click on the "Hamburger Menu" (three horizontal lines) in the top left of the user interface.
3. Select the *Share or embed map* option.
4. Select the *Embed map* option, which will give you some `<iframe>` code — copy this.

```
<figure>
    <figcaption>IES San Vicente.</figcaption>
    <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3126.6024
0.5317505844743932!3d38.4044398796498!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3
width="600" height="450" frameborder="0" style="border:0;" allowfullscreen="" ari
</figure>
```

### 5.2.1. Proposed exercise: Top ten locations

> *Following the previous example, create a web page to insert at least ten figures containing locations from Google Maps. Update the caption of each figure to show a brief description of each location, and check the results in your browser. Do not forget to add all necessary HTML basic tags and validate your code.*