



Web design and development II (CW2)

Class 12

Managing audio and video elements

<audio> </audio>

In HTML5, audio is integrated using `<audio> </audio>` to which different attributes and contents should be added.

Into the `<audio>` tags, it is mandatory to indicate what is the file to be played using `<source>` as well as the attributes `src=""` to define the file and `type=""` to indicate the file's format.

Attributes :

autoplay

Plays the audio file as soon as sufficient data has been downloaded. It is not a good practice to use autoplay in most circumstances as it bothers most of the users.

controls

Displays the control items such as volume, stop, the reading head, etc. Note that the audio player is supplied by the browser.

loop

Plays the audio file over and over again.

preload

Using different values, it indicates to the browser how to treat this file.

- **none**: the file is not put in cache.
- **metadata**: metadata only are downloaded.
- **auto**: the file is download even though the user may not play it.

src

Usually used in the `<source>` tag.

<source />

Many `<source>` tags can be used specifying different audio formats. In which case, the browser would use the first file supported.

A warning can also be added in case the audio would not be supported by the browser.

```
<audio src="file.mp3" controls> </audio>
```

```
<audio controls preload="auto" loop>
  <source src="file.ogg" type="audio/ogg" />
  <source src="file.mp3" type="audio/mpeg" />
  <source src="file.wav" type="audio/wav" />
  Your browser does not support the audio.
</audio>
```

<video> </video>

Before HTML5, video had to be played using a *plugin*. Now it is simply played with the HTML content using the `<video>` `</video>` tags.

The same attributes as for `<audio>` are used with `<video>`.

A specific attribute, `poster=""`, allows you to specify an image to be displayed before video starts, and, of course, height and width needs to be specified.

It uses `<source />` tag the same way as it is used for `<audio>` and also allows you to leave a warning for browsers not supporting video files.

```
<video width="320" height="240" poster="image.jpg"
controls autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support video.
</video>
```

How to create a video background

Video backgrounds are quite a trendy thing. Unfortunately, videos cannot be used as values in CSS background properties. However, there is nothing a creative mind can't achieve!

Procedure

All there is to do is to make the video full screen and positioned behind the content items. There are little tricks to be known.

We first have to make sure video will start playing automatically using the attribute ***autoplay***, that it will play over and over using ***loop***, and that it will be silent using ***muted***.

The ***playsinline*** attribute is used so the video is played inline and not launch full-screen with some devices (e.g. IOS prior to version 10).

Positioning the video perfectly centered needs a little trick. Notice that *top* and *left* positions are set to 50%, but transform: ***translateX*** and ***translateY*** are set to -50%.

Minimum width and height of 100% are used plus auto width and height as a fall-back.

Finally, a background image is used just in case the video would not work.

You will make sure to position the <video> at the very beginning of the code (within the <body> section) and code the rest of the page as usual.

HTML:

```
<video poster="image.jpg" playsinline autoplay muted preload="auto" loop>
  <source src="bg.mp4" type="video/mp4">
</video>
```

CSS:

```
video {
  position: fixed;
  top: 50%;
  left: 50%;
  min-width: 100%;
  min-height: 100%;
  width: auto;
  height: auto;
  z-index: -100;
  transform: translateX(-50%) translateY(-50%);
  background: url("image.jpg") no-repeat;
  background-size: cover;
}
```