**Adding Video and Audio to your Web Pages**

In this chapter we are going to look at adding video and audio to our pages. Collectively, these are often referred to as ***rich media****.*

When it comes to showing Flash animations, video, and audio on your site, things get a little more complicated for two reasons:

* First, there are many different *formats* to learn about. Flash is fairly easy because there is just the one format for Flash animations, but there are many different formats for video, including AVI, Flash Video, MPEG, QuickTime, and Windows Media, and several formats for audio such as MP3 and WAV.
* Second, while browsers are built to display text and images, they do not automatically show the different formats of rich media. Instead, browsers rely on additional programs known as *players* or *plug - ins,* and these players do not play every format (just as a DVD player would not be able to play a VHS video or a vinyl record). So, your choice of which format to use might be restricted by how many people have the player installed.

Some plug - ins are installed by default when you get a new computer, and when they are not installed users can download them (usually for free) from the web sites of the companies who make them. For example, the Flash plug - in is usually installed on new computers, but it can also be downloaded for free from the Adobe web site.

**Embed Tag:**

The <embed> tag in HTML is used for embedding external application/plug -ins/interactive content which is generally multimedia content like audio or video into an HTML document. It is used as a container for embedding plug-ins such as flash animations. This tag is a new tag in HTML 5 and it requires only starting tag.

Syntax:

<embed src = “hello.swf” type=“application/x-shockwave-flash”>

Attributes of embed tag:

* src
* type
* height
* width

**Object Element:**

The object tag is use to embed the multimedia contents in web pages using HTML. With the help of object tag we can display pdf, content, video format and another webpage into your document.

In order to show rich media on your pages, you will need to learn a new element, the <object> element. But for now, you just need to know that the <object> element can tell the browser several things, including:

* The name of the file and where to find it
* The format of the file
* The type of plug - in used to play the file

Attributes:

* data
* type
* height
* width etc

Example:

<object width="400" height="400" data="helloworld.swf"></object>

**Param Element:**

Inside the <object> element, you can use the <param> element to pass extra information to the player (for example, you might want to tell a player to start playing automatically when the page loads, rather than waiting for the user to press a play button).

The <param> tag is used to define parameters for plugins embedded with an [<object>](https://www.w3schools.com/tags/tag_object.asp) element.

Attributes of param tag:

* name (controller, autoplay, width, height, src etc)
* value (true, false, 400, 200, hello.mp4 etc.)

Example:

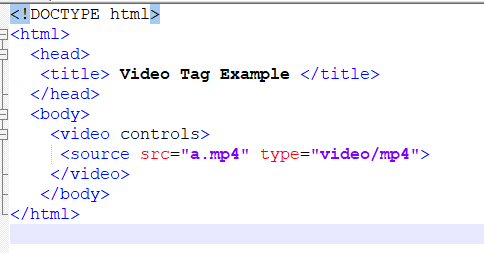
<object data="horse.wav">  
  <param name="autoplay" value="true">  
</object>

**Adding Videos to Your Site:**

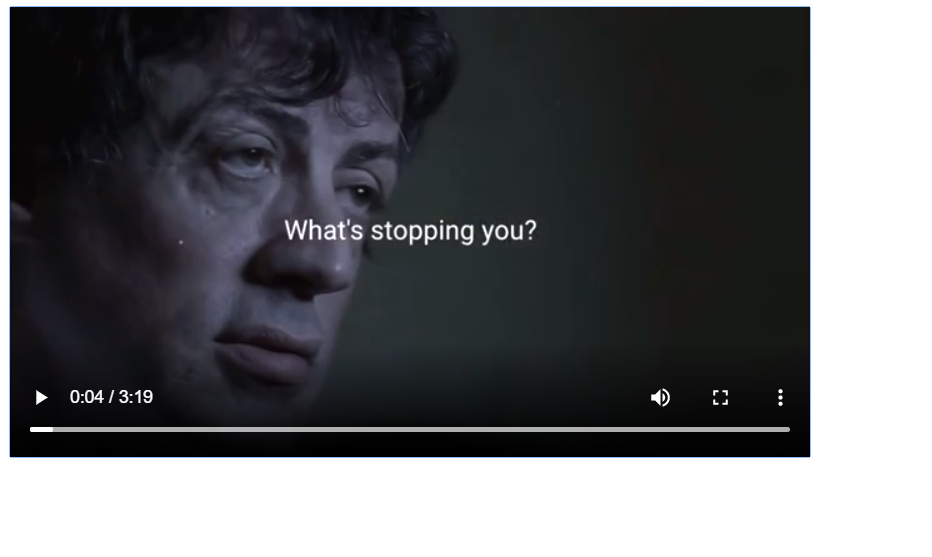
Before HTML 5, to play video files user need to install pug-ins like flash players which would result in taking more resources or storage. But nowadays most of the mobile platforms do not support flash players.

Hence, to let user to play video files in any platform without depending on plug-ins the solution is “video” tag. As, almost all browser support <video> tag.

Example/Source Code:



Output:



Attributes of video tag:

* autoplay (to make your video play use autoplay in conjunction as: playsinline autoplay muted)
* controls
* loop
* muted
* preload (it is ignored if autoplay is present) – Currently not working
* src
* poster (thumbnail)
* width
* height

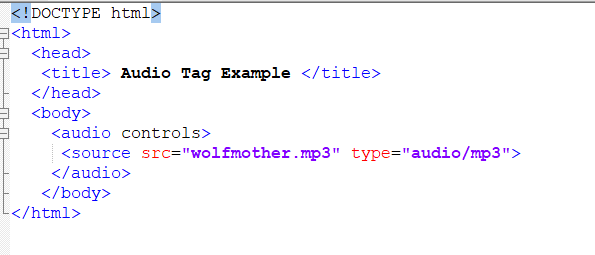
**Adding Audio to Your Site:**

The <audio> tag defines sound, such as music or other audio streams. Some supported audio file formats are MP3, WAV, OGG, midi etc. Audio tag is an inline element.

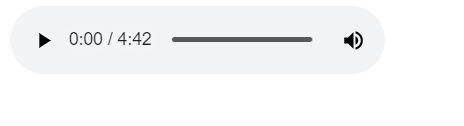
Attributes of audio**:**

* controls
* autoplay
* loop
* muted
* “preload” has 3 properties:
  + auto: icons of the audio file get displayed automatically
  + none: first icon does not gets displayed but after playing that audio file it displays the available icons.
  + metadata: First browser imports all the information of the file (file you want to display) and then imports that audio file.

Example/Source Code:



Output:



**Automatically Playing Audio and Video When a Page Loads:**

The autoplay attribute is a boolean attribute. When present, the audio will automatically start playing as soon as it can do so without stopping.

Example:

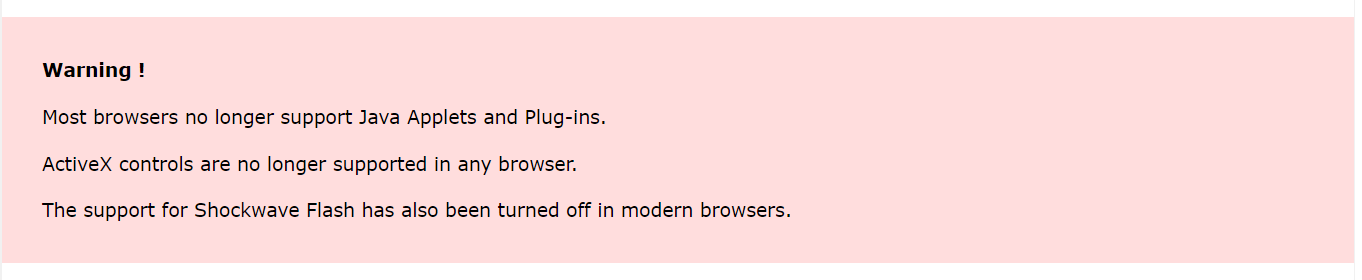
<audio controls autoplay>

<source src = ”wolfmother.mp3” type = “ audio/mp3”>

</audio>

**------------<Below topics are optional>-----------**

**A Closer Look at the <object> and <param> Elements**:

Now that you have seen several examples of how to add rich media to your web pages, we ’ll finish off the chapter with a closer look at the <object> and <param> elements in the same way that we have covered the syntax of other elements in previous chapters.

**<object> element attributes: -**

1. **The archive attribute: (Not supported in HTML5)**

A space-separated list of URIs for archives of resources for the object.

1. **The border Attribute (Deprecated): (Not supported in HTML5)**

The border attribute specifies the width of the border to appear around the object; the value is specified in pixels. However, it is deprecated and you should use the border property in CSS instead.

1. **The classid Attribute: (Not supported in HTML5)**

The URI of the object's implementation. It can be used together with, or in place of, the data attribute.

The following is an example of a classid to embed a player to show a QuickTime movie:

classid = ”clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B”

1. **The codebase Attribute: (Not supported in HTML5)**

If the browser does not have the required plug - in (or other code) required to display the file specified in the <object> element, then the codebase attribute specifies the base URL where a plug - in can be downloaded.

The idea is that if the browser cannot display the file it will offer the user the option to download the necessary files. For example, if you were working with Java, it might look like this:

codebase = ”http://www.example.org/javaclasses/”

1. **The codetype Attribute: (Not supported in HTML5)**

The codetype attribute specifies the MIME type of the application or code required to play the file (not to be confused with the MIME type of the file itself). For example, if you wanted to embed an Excel spreadsheet into a web page, you would have a codetype attribute like this:

<object data=”sales.xls” codetype=”s”>

Browsers can use the codetype attribute to skip over unsupported media types without having to download unnecessary objects. Appendix H covers MIME types.

1. **The declare Attribute: (Not supported in HTML5)**

The declare attribute is used to declare an object without instantiating it. It saves you from having to download an object until it is actually required.

It is a Boolean attribute, and while it does not need a value in HTML, all attributes in XHTML require a value, so you would use the following:

declare = ”declare”

1. **The data Attribute:**

If the object has a file to process or play, then the data attribute specifies the URL for that file. For example, here is a URL to an MP3:

data = ”http://www.example.com/mp3s/newsong.mp3”

This is similar to the src attribute that is used on the <img> element, and the value can be a relative URL.

1. **The height and width Attributes:**

The height and width attributes specify the height and width of an object. The values should be in pixels or a percentage of the containing element. They are treated just like height and width attributes of the <img> element. The use of these attributes should make the page load faster because the browser can lay out the rest of the page without completely loading the object.

1. **The hspace and vspace attributes (Deprecated): (Not supported in HTML5)**

The hspace and vspace attributes specify the amount of white space that should appear around an object, just as when they are used with the <img> element. They have been replaced by the margin and border properties of CSS.

1. **The name Attribute (Deprecated): (Not supported in HTML5)**

The name attribute provides a name that can be used to refer to the object, in particular for use in scripts. It has been replaced by the id attribute in XHTML.

1. **The standby Attribute:**

The standby attribute specifies a text string that will be used when the object is loading:

standby = ”Trailer for Harry Potter 27 is loading”

The value should be a meaningful description of the object that is loading.

1. **The tabindex Attribute:**

The tabindex attribute indicates the tab index of the object within a page.

**<param> element attributes: -**

The <param> element is used to pass parameters to an object. The kinds of parameters an object requires depend upon what the object does; for example, if an object has to load a Flash MP3 player into the page, you will probably need to specify where the MP3 file can be found. Alternatively, if you are adding a video to a page, your object might allow you to tell it whether to automatically play the video when the page loads, or whether to wait for the user to press a play button in order for it to start.

1. **The name and value Attributes: (Supported in HTML5)**

The name and value attributes act as a name/value pair (rather like attributes themselves). The name attribute provides a name for the parameter you are passing to the application, while the value gives the value of the parameter.

Here are a couple of examples, taken from a QuickTime movie. The first parameter indicates the source of the file being loaded to play, while the second indicates that the movie should start playing automatically as it is loading (without the user having to start it):

<param name=”src” value=”movieTrailer.mov”>

<param name=”autoplay” value=”true”>

1. **The valuetype Attribute: (Not supported in HTML5)**

If your object accepts parameters, then the valuetype attribute indicates whether the parameter will be a file, URL, or indeed another object. The table that follows shows the possible values.

1. **The type Attribute: (Not supported in HTML5)**

This specifies the MIME type of the content being used in the object. For example, you might want to specify that you were passing an MP3 file, in which case you would use the value attribute like so:

value=”audio/mpeg”