

USTH MM2.1

Soft. Eng. for Interactive Media



Lecture #3.3 – Video and audio in HTML5

New tags: `<audio>` and `<video>`

- HTML5 treats audio and video as first-class citizens in terms of web content.
- `<audio>`
- `<video>`
- Allow audio/video to play natively in the browser.
- Good-bye plug-ins?

Audio on the web: a brief history

- Embedded MIDI files

```
<embed src="awesome.mp3" autostart="true"  
      loop="true" controller="true"></embed>
```

```
<object>  
<param name="src" value="simpsons.mp3">  
<param name="autoplay" value="false">  
<param name="controller" value="true">  
<embed src="awesome.mp3" autostart="false"  
      loop="false" controller="true"></embed>  
</object>
```

Plug-ins

- RealPlayer
- Windows Media
- QuickTime
- Flash
- And then came Apple... (iPhone / iPod with no Flash support) [2007]
 - H.264 codecs with richer capabilities
 - (Mobile) Safari

Video codecs

- Three relevant contemporary video formats related to the HTML5 `<video>` tag:
 - H.264
 - Theora
 - VP8

H.264

- High-quality codec, standardized in 2003 and created by the MPEG group.
- Supports several *profiles*, which allows a wide range of devices to be accommodated.
 - These profiles share a set of common features, but higher-end profiles offer additional options that improve quality.
 - It's possible to encode a video one time and embed multiple profiles so that it looks nice on various platforms.
- H.264 is a *de facto* standard because of support from Microsoft and Apple, which are licensees.

H.264

- Google's YouTube converted its videos to the H.264 codec so it could play on the iPhone, and Adobe's Flash Player supports it as well.
- However, it's not an open technology.
- It is patented, and its use is subject to licensing terms.
- Content producers must pay a royalty to encode videos using H.264, but these royalties do not apply to content that is made freely available to end users.

Theora

- Royalty-free codec developed by the xiph.org Foundation.
- Although content producers can create videos of similar quality with Theora, device manufacturers have been slow to adopt it.
 - Firefox, Chrome, and Opera: OK
 - Internet Explorer, Safari, and the iOS devices: not.
 - Apple and Microsoft are wary of “submarine patents”.

VP8

- Google's VP8 is a completely open, royalty-free codec with quality similar to H.264.
- Supported by Mozilla, Google Chrome, and Opera.
 - Microsoft's Internet Explorer 9 promises to support VP8 as long as the user has installed a codec already.
- Also supported in Adobe's Flash Player, making it an interesting alternative.
- Not supported in Safari or the iOS devices, which means that although this codec is free to use, content producers wanting to deliver video content to iPhones or iPads still need to use the H.264 codec.

Browser support for HTML5 video

Browser	Operating system	Latest stable release	Video formats supported			
			Theora	H.264 (MP4)	VP8 (WebM)	VP9 (WebM)
Android browser	Android	4.2.1 "Jelly Bean" (November 27, 2012; 15 months ago) [38][39]	2.3 ^[40]	3.0 ^[40]	2.3 ^[40]	No
Chromium	All supported	N/A	r18297 ^[41]	Manual install ^[note 1]	r47759 ^[43]	r172738 ^[44]
Google Chrome		33.0.1750.152 & 33.0.1750.154 (March 14, 2014; 18 days ago) ^[45] [z]	3.0 ^{[46][47]}	3.0 ^{[47][note 2]}	6.0 ^{[49][50]}	29.0 ^[note 3]
Internet Explorer	Windows	v11.0.9600.16521 (11.0.4) (11 March 2014; 21 days ago) [z]	Manual install ^[note 4]	9.0 ^[54]	Manual install ^[note 5]	
	Windows Phone	10.0 (November 21, 2012; 16 months ago) [z]	No	9.0 ^[57]	No	No
	Windows RT	10.0		10.0 ^[58]		
Konqueror	All supported	4.12.3 (4 March 2014; 28 days ago) [z] ^[59]		4.4 ^[note 6]		No
Mozilla Firefox	Windows 7+	28.0 (March 18, 2014; 14 days ago) ^[61] [z] ESR 24.4.0 (March 18, 2014; 14 days ago) ^[62] [z]	3.5 ^[63]	21.0 ^[note 7]	4.0 ^{[66][67]}	28.0 ^{[68][69]}
	Windows Vista			22.0 ^[70]		
	Linux			26.0 (using GStreamer) ^[note 8]		
	Android			17.0 ^[73]		
	All other supported			No		
Opera		16 for Android (September 18, 2013; 6 months ago) ^[74] [z] 12.0.22 for Symbian S60 (June 24, 2012; 20 months ago) ^[74] [z] 10.0 for Windows Mobile (March 16, 2010; 4 years ago) ^[75] [z]	No	11.50	15.0	16.0
	Windows, OS X	20.0.1387.82 (March 20, 2014; 12 days ago) ^[76] [z]	10.50 ^[78]	No	10.60 ^{[79][80]}	No
	Linux, FreeBSD	12.16 (July 4, 2013; 8 months ago) ^[77] [z]				
Safari	iOS	7.0.2 (February 25, 2014; 35 days ago) ^[81] [z]	No	3.1 ^{[82][83]}	No	No
	MacOS X	6.1.2 (February 25, 2014; 35 days ago) ^[81] [z]	Manual install ^[note 9]		Manual install ^[84]	
Web (previously Epiphany)	All supported	3.12 (March 26, 2014; 6 days ago) [z] ^[85]		2.28 ^[note 10]		No

Source: http://en.wikipedia.org/wiki/HTML5_video#Browser_support

Audio codecs

- Most relevant contemporary audio codecs:
 - AAC
 - MP3
 - Vorbis (OGG)

Advanced Audio Coding (AAC)

- The audio format that Apple uses in its iTunes Store.
 - Designed to have better audio quality than MP3s for around the same file size
 - Also offers multiple audio profiles similar to H.264.
- Moreover, like H.264, it's not a free codec and does have associated licensing fees.
- All Apple products play AAC files.
 - So does Adobe's Flash Player and the open source VLC player.

MP3

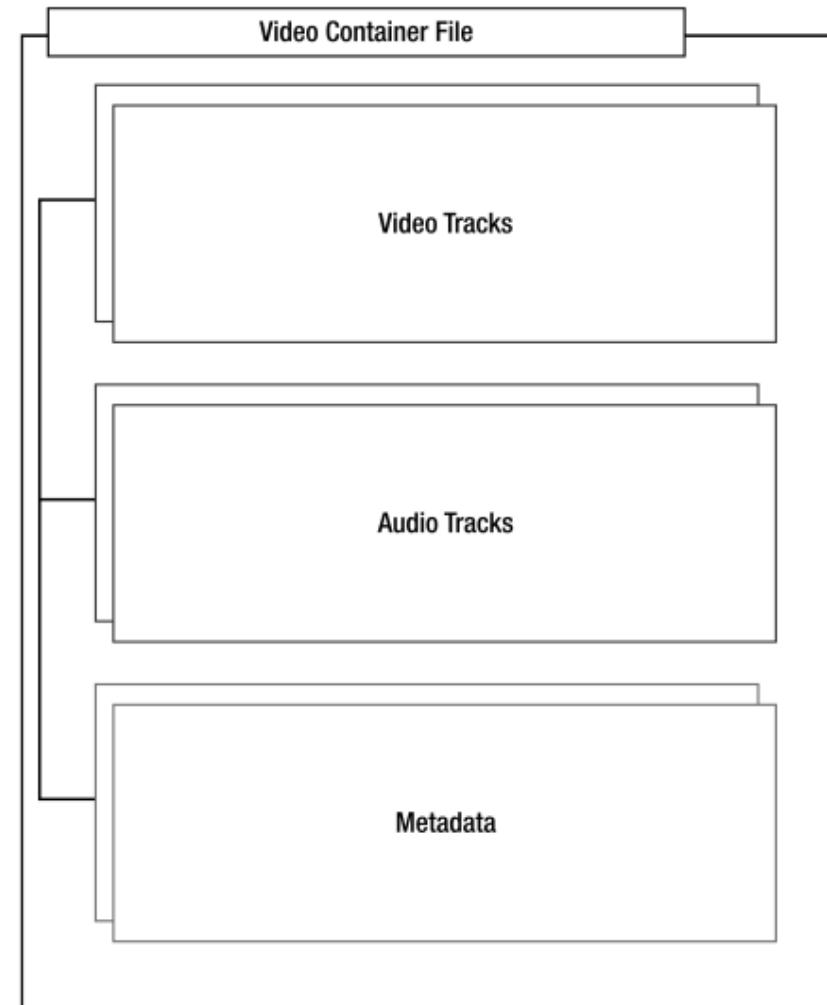
- The MP3 format, although extremely common and popular, isn't supported in Firefox and Opera because it's also patent-encumbered.
- It is supported in Safari and Google Chrome.

Vorbis (OGG)

- This open source royalty-free format is supported by Firefox, Opera, and Chrome.
- You'll find it used with the Theora and VP8 video codecs as well.
- Vorbis files have very good audio quality but are not widely supported by hardware music players.

Containers and codecs

- Containers are like an envelope that contains audio streams, video streams, and sometimes additional metadata such as subtitles.
 - These audio and video streams need to be encoded, and that's where codecs come in.



Combining codecs and containers

- Video and audio codecs need to be packaged together for distribution and playback.
- A *container* is a metadata file that identifies and interleaves audio or video files.
 - A container doesn't actually contain any information about how the information it contains is encoded.
 - Essentially, a container “wraps” audio and video streams.

Combining codecs and containers

- Containers can often hold any combination of encoded media, but we'll see these combinations when it comes to working with video on the Web:
 - The OGG container, with Theora video and Vorbis audio, which will work in Firefox, Chrome, and Opera.
 - The MP4 container, with H.264 video and AAC audio, which will work in Safari and Chrome.
 - It will also play through Adobe Flash Player and on iPhones, iPods, and iPads.
 - The WebM container, using VP8 video and Vorbis audio, which will work in Firefox, Chrome, Opera, and Adobe Flash Player.

Working with audio

- The following slides discuss how to embed audio files in a page.

Audio Example #1

- Sound samples

- [Live version](#)
- http://media.pragprog.com/titles/bhh5/code/html5_audio/audio.html
- Two codecs:
 - OGG
 - type="audio/ogg"
 - MP3
 - type="audio/mpeg"

Samples:

Drums



Guitar



Organ



Bass



Audio Example #2

```
<script type="text/javascript" charset="utf-8">
$(function(){
  var canPlayAudioFiles = !!(<document.createElement('audio').canPlayType);

  if(canPlayAudioFiles){
    $(".sample a").hide();

    var element = $("<p><input type='button' value='Play all' /></p>")
    element.click(function(){
      $("audio").each(function(){
        this.play();
      })
    });

    $("body").append(element);
  }
});
</script>
```

- [html5_audio/advanced_audio.html](#)
- [Live version](#)
- http://media.pragprog.com/titles/bhh5/code/html5_audio/advanced_audio.html
- Same 2 codecs as before
- JavaScript code for browsers that don't support the <audio> tag

Exploring the HTML5 audio API

- Control functions

Function	Behavior
<code>load()</code>	Loads the media file and prepares it for playback. Normally does not need to be called unless the element itself is dynamically created. Useful for loading in advance of actual playback.
<code>play()</code>	Loads (if necessary) and plays the media file. Plays from the beginning unless the media is already paused at another position.
<code>pause()</code>	Pauses playback if currently active.
<code>canPlayType(type)</code>	Tests to see whether the video element can play a hypothetical file of the given MIME type.

Source: [Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development](#), by Peter Lubbers, Brian Albers, Frank Salim

Exploring the HTML5 audio API

- Read-only media attributes

Read-only attribute	Value
<code>duration</code>	The duration of the full media clip, in seconds. If the full duration is not known, NaN is returned.
<code>paused</code>	Returns <code>true</code> if the media clip is currently paused. Defaults to <code>true</code> if the clip has not started playing.
<code>ended</code>	Returns <code>true</code> if the media clip has finished playing.
<code>startTime</code>	Returns the earliest possible value for playback start time. This will usually be 0.0 unless the media clip is streamed and earlier content has left the buffer.
<code>error</code>	An error code, if an error has occurred.
<code>currentSrc</code>	Returns the string representing the file that is currently being displayed or loaded. This will match the source element selected by the browser.

Source: [Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development](#), by Peter Lubbers, Brian Albers, Frank Salim

Exploring the HTML5 audio API

- Scriptable attribute values

Attribute	Value
<code>autoplay</code>	Sets the media clip to play upon creation or queries whether it is set to <code>autoplay</code> .
<code>loop</code>	Returns <code>true</code> if the clip will restart upon ending or sets the clip to loop (or not loop).
<code>currentTime</code>	Returns the current time in seconds that has elapsed since the beginning of the playback. Sets <code>currentTime</code> to seek to a specific position in the clip playback.
<code>controls</code>	Shows or hides the user controls, or queries whether they are currently visible.
<code>volume</code>	Sets the audio volume to a relative value between 0.0 and 1.0, or queries the value of the same.
<code>muted</code>	Mutes or unmutes the audio, or determines the current mute state.
<code>autobuffer</code>	Tells the player whether or not to attempt to load the media file before playback is initiated. If the media is set for auto-playback, this attribute is ignored.

Source: [Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development](#), by Peter Lubbers, Brian Albers, Frank Salim

Exploring the HTML5 audio API

- Intro example

Exercise: make the ‘Play All’ button behave as a “toggle” between ‘Play All’ and ‘Pause All’

```
var element = $("<p><input type='button' value='Play all' /></p>")
element.click(function(){
    $("audio").each(function(){
        this.play();
    })
});

$("body").append(element);
```


Embedding video

- The following slides discuss how to embed video files in a page.

Video Example

Photoshop tricks

- [Live version](#)

[http://
media.pragprog.com/
titles/bhh5/code/
html5video/index.html](http://media.pragprog.com/titles/bhh5/code/html5video/index.html)

- 3 codecs:

- H.264 (mp4)
- Theora (ogv)
- VP8 (webm)

- Flash fallback

- Links for
downloads

```
<video controls>
  <source src="video/h264/01_blur.mp4">
  <source src="video/theora/01_blur.ogv">
  <source src="video/webm/01_blur.webm">

  <object width="640" height="480" type="application/x-shockwave-flash"
    data="swf/flowplayer-3.2.2.swf">
    <param name="movie" value="swf/flowplayer-3.2.2.swf" />
    <param name="allowfullscreen" value="true" />
    <param name="flashvars"
      value='config={"clip":{"url":"../video/h264/01_blur.mp4",
        "autoPlay":false,
        "autoBuffering":true
      }}' />
    
  </object>

  <p>Your browser does not support the video tag.</p>
</video>
```

Limitations of HTML5 video

- HTML5 video has no provisions for streaming the video files.
- There's no way to manage rights.
- The process of encoding videos is costly and time-consuming.
 - The need to encode in multiple formats makes HTML5 video much less attractive

Exploring the HTML5 video API

- Additional video attributes

Attribute	Value
poster	The URL of an image file used to represent the video content before it has loaded. Think “movie poster.” This attribute can be read or altered to change the poster.
width, height	Read or set the visual display size. This may cause centering, letterboxing, or pillaring if the set width does not match the size of the video itself.
videoWidth, videoHeight	Return the intrinsic or natural width and height of the video. They cannot be set.

Source: [Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development](#), by Peter Lubbers, Brian Albers, Frank Salim

Exploring the HTML5 video API

- Intro example
- Play button doesn't work on Safari (or Opera)

```
<div id="time">
  00:00:00
</div>

<script type="text/javascript">
var video = $("video");
video.bind("timeupdate", function(){
  $("#time").html( this.currentTime );
});
var button = $("<input type='button' value='Play'>");
button.insertAfter(video);
</script>
```

Using the HTML5 video API

- Advanced example :Video Timeline (combining <video> and <canvas>)



Learn more about it

- <http://www.html5video.org/>
- **Players:**
 - <http://videojs.com/>
 - <http://sublimevideo.net/>

- [Book:](#)

