

## 1 AUDIO AND VIDEO

HTML5 and CSS3

## 4 HTML MULTIMEDIA—PAGE 1

- Multimedia on the web is sound, music, videos, movies, and animations
- It comes in many different formats, almost anything you can hear or see
  - E.g. images, music, sound, videos, records, films, animations, etc.
- Web pages can contain multimedia elements of all of these different types and formats

## 5 HTML MULTIMEDIA—PAGE 2

- Multimedia is displayed within a web page in a fashion similar to inline images:
  - Controls are displayed as part of the web page
  - To play a multimedia file, a browser used to need access to a plug-in or add on
  - Starting with HTML5, many browsers now include built-in support for audio and video files, removing the need for plug-ins

## 6 BROWSER SUPPORT

- The first web browsers had support for text only, limited to a single font in a single color
- Later came browsers with support for colors and fonts, and images
- Audio, video, and animation have been handled differently by major browsers
  - Different formats have been supported, and some formats require extra helper programs (plug-ins) to work
- HTML5 promises an easier future for multimedia

## 7 BANDWIDTH

- Bandwidth is a measure of the amount of data that can be sent through a communication pipeline each second
  - The developer should consider bandwidth when working with multimedia on a web site
- Today multimedia is much more accessible to Internet users because of *high-speed* internet connections and the *reduction of the file size* of multimedia clips

## 8 MULTIMEDIA AND COMPRESSION—PAGE 1

- Multimedia today is a major source of information on the web
- Uncompressed data required more storage and transmission bandwidth but out technologies have limited storage capacity and bandwidth

## 9 MULTIMEDIA AND COMPRESSION—2

- During the past years, compression technology has played an important role because it changes the way of working on multimedia (images, video, text, audio, speech, etc.)
- This is possible because digital multimedia have redundant data and irrelevant information

- For reducing redundancy, focus on removing replication from the source thereby reducing the size of the file
- 10  **EXPLORING DIGITAL AUDIO—PAGE 1**
- Every sound wave is composed of two components:
    - Amplitude—the “height” of the sound wave
      - Relates to sound’s volume
    - Frequency—the “speed” at which the sound wave moves
      - Relates to sound’s pitch
- 13  **EXPLORING DIGITAL AUDIO—PAGE 2**
- People hear sounds as a continuously varying “wave” signal
  - The sound wave must be converted from an *analog* format to *digital* format to store as a computer file
  - Digital recording measures sound’s amplitude at discrete moments in time
    - Each measurement is called a sample
    - The number of samples taken per second is called the sampling rate
- 15  **EXPLORING DIGITAL AUDIO—PAGE 3**
- Sampling resolution (also called bit depth) indicates the precision in measuring the sound within each sample, e.g. the number of bits used to store the sound
    - 8-bit, 16-bit, 32-bit
- 17  **AUDIO FILE FORMATS—PAGE 1**
- There are different sound file formats used for different operating systems
  - Different file formats provide varying levels of sound quality and file compression
- 18  **AUDIO FILE FORMATS—PAGE 2**
- Some of the audio file formats in HTML5 are:
    - MP3—Most popular format for downloading and storing music; compresses sound files to roughly one-tenth of original size; proprietary and not royalty-free
    - WAV—original format for Windows for uncompressed CD-quality sound; requires about 10 megabytes per minute, impractical for web audio except for sound clips
    - Ogg Vorbis—file compression that is open source; better sound quality than MP3 but supported by only a few portable players
- 19  **THE AUDIO ELEMENT—PAGE 1**
- The `<audio>` element is an HTML5 block that specifies a standard way to embed an audio file into a web page
    - Currently the only supported formats are MP3, WAV and OGG
    - One or more single-sided `<source>` elements within the `<audio>` block specify alternative audio files from which the browser chooses to play
      - Browser selects the first “recognized” format
- 21  **THE AUDIO ELEMENT—PAGE 2**

- Attributes for the <audio> element include:
  - controls—adds audio controls such as “play”, “pause” and “volume” to play audio (or controls="controls")
  - autoplay—causes the audio clip to start automatically once the web page is loaded (or autoplay="autoplay")
  - loop—automatically restarts the audio clip once it is finished playing (or loop="loop")

## 22 THE AUDIO ELEMENT—PAGE 3

- Attributes for the <audio> element include (*con.*):
  - preload—specifies how audio clip should be preloaded by the browser when the web page first loads
    - auto—preload the entire clip (before playing)
    - metadata—preload just the descriptive data
    - none—do not preload audio clip

## 23 THE AUDIO ELEMENT—PAGE 4

- Format:
 

```
<audio id="id" controls[="controls"] autoplay[="autoplay"] >
  <source src="path/filename" type="MIME-type" />
  <source src="path/filename" type="MIME-type" />
  ...
  Text displayed if browser supports no listed format
</audio>
```

  - The *filenames* are the possible sources of the audio file
  - The type attribute in each source tag within the block specifies the “MIME-type” (the type of data in the file)

## 24 THE AUDIO ELEMENT—PAGE 5

- Example:
 

```
<audio id="myAudio" controls="controls" >
  <source src="myTune.mp3" type="audio/mp3" />
  <source src="myTune.ogg" type="audio/ogg" />
  <source src="myTune.wav" type="audio/wav" />
</audio>
```

## 25 THE MIME-TYPE—PAGE 1

- The MIME-type (multipurpose internet mail extension) is an Internet standard that specifies files based on their “natures” and “formats”
- Each MIME is indicated in the type attribute for an HTML element contains two parts:
  1. The type indicates categorization of the type, e.g. text, audio, video, etc.
  2. The subtype is specific to a particular file type, e.g. css, mp3, webm, etc.

26  **THE MIME-TYPE—PAGE 2**

- [Examples](https://www.thoughtco.com/mime-types-by-content-type-3469108) (there are over 600 of them for HTML) (<https://www.thoughtco.com/mime-types-by-content-type-3469108>):
  - text/css
  - audio/mp3

27  **THE MIME-TYPE—PAGE 3**

- All three MIME type attribute values for *audio* are shown in this example:

```
<audio id="myAudio" controls>
  <source src="myTune.mp3" type="audio/mp3" />
  <source src="myTune.wav" type="audio/wav" />
  <source src="myTune.ogg" type="audio/ogg" />
</audio>
```

29  **BROWSER SUPPORT FOR AUDIO ELEMENT**

<u>Browser</u>	<u>MP3</u>	<u>WAV</u>	<u>OGG</u>	
Edge/IE	YES	NO	NO	
Chrome	YES	YES	YES	
Firefox	YES	YES	YES	
Safari		YES	YES	NO
Opera	YES	YES	YES	

31  **THE EMBED ELEMENT—PAGE 1**

- Older browsers that do not support the HTML5 audio element rely instead on plugins to play embedded media
- To insert an embedded object such as a media player, you can nest the embed element within the audio element
- Browsers recognizing the audio element will attempt to load the audio clip that way

32  **THE EMBED ELEMENT—PAGE 2**

- Format:
 

```
<embed id="id" src="path/filename" type="MIME-type" width="width" height="height" />
```
- Example:
 

```
<audio>
  <embed src="overture.mp3" type="audio/mp3" width="250" height="10" />
</audio>
```

33  **MIME-TYPES FOR THE EMBED ELEMENT**

<u>Format</u>	<u>File extension</u>	<u>MIME-type</u>
AU	.au	audio/basic
MIDI	.mid	audio/mid
MP3	.mp3	audio/mpeg

Ogg Vorbis	.ogg	audio/ogg
RealAudio	.ra	audio/x-pn-realaudio
SND	.snd	audio/basic
Shockwave Flash		.swf audio/x-shockwave-flash
WAV	.wav	audio/wave

### 35 **EXPLORING DIGITAL VIDEO—PAGE 1**

- Digital video adds a visual element to a web page as well as provides information
- Video files are composed of a series of single images called frames
  - Many frames are sized to have width-to-height ratios or aspect ratios of 4:3
  - Theatrical releases typically have aspect ratios of 1.85:1 or 2.39:1

### 36 **EXPLORING DIGITAL VIDEO—PAGE 2**

- The number of frames shown in a specified period of time is the frame rate
- The video bit rate is the amount of data that has to be processed by the video player each second

### 37 **VIDEO FILE FORMATS—PAGE 2**

- Some video (compression) file formats in HTML5 are:
  - MP4—proprietary format developed by Apple; most popular format for video and supported both by HTML5 and Flash Players
  - Ogg Vorbis—file compression that is open source; better sound quality than MP3 but supported by only a few portable players
  - WebM—original format for Windows for uncompressed CD-quality sound; requires about 10 megabytes per minute, impractical for web audio except for sound clips

### 38 **THE VIDEO ELEMENT—PAGE 1**

- The <video> element is an HTML5 block that specifies a standard way to embed a video file into a web page
  - Before HTML5, a video could only be played in a browser with a plug-in such as “Flash”
- One or more single-sided <source> elements within the <video> block specify alternative video files from which the browser chooses to play
  - Browser selects the first recognized format

### 39 **THE VIDEO ELEMENT—PAGE 2**

- Attributes for the <video> element include:
  - controls—adds video controls such as “play”, “pause” and “volume” to play video (or controls=“controls”)
  - autoplay—causes the video clip to start automatically once the web page is loaded (or autoplay=“autoplay”)
  - loop—automatically restarts the video clip once it is finished playing (or loop=“loop”)
  - preload—specifies how video clip should be preloaded by the browser when the

web page loads

- auto—preload the entire clip
- metadata—preload just the descriptive data
- none—do not preload video clip

#### 40 THE VIDEO ELEMENT—PAGE 3

- Attributes for the <video> element include (*con.*):
  - audio—used to mute the audio of the video clip (e.g. audio="muted")
  - It also is a good idea to specify the width and height attributes to keep the page from “flickering” while the video loads

#### 41 THE VIDEO ELEMENT—PAGE 4

- Format:

```
<video id="id" controls="controls" autoplay="autoplay">
  <source src="path/filename" type="MIME-type">
  <source src="path/filename" type="MIME-type">
```

...

*Text displayed if browser supports no listed format*

```
</video>
```

The src attribute names possible sources of the video clip

#### 42 THE VIDEO ELEMENT—PAGE 5

- Example:

```
<video id="myVideo" controls="controls">
  <source src="small.mp4" type="video/mp4">
  <source src="small.webm" type="video/webm">
  <source src="small.ogv" type="video/ogg">
  <source src="small.3gp" type="video/3gp">
</video>
```

#### 44 THE OBJECT ELEMENT—PAGE 1

- The object element was introduced in the specifications for HTML 4 for marking any kind of non-textual content
- A replacement for embed element which is widely supported though never part of the previous HTML specifications released by the W3C
- The object element can be used with almost any type of content, from sound and video clips to graphic images, PDF files and even the content of other web pages

#### 45 THE OBJECT ELEMENT—PAGE 2

- Format:

```
<object attributes>
  parameters
</object>
```

- *attributes* “define” the object
- *parameters* are values passed to the object that control its appearance and how users interact with the object

#### 46 THE OBJECT ELEMENT—PAGE 3

- Attributes for the <object> element include:
  - *data*—the source of the file used in the object (*data*="url")
  - *form*—in HTML5 the form to which the object belongs (*form*="name")
  - *type*—the MIME type of the data within the object (*type*="mime")
  - *usemap*—associates the object with a client-side image map (*usemap*="url")
  - Other attributes are a unique name, width and height

#### 47 THE OBJECT ELEMENT—PAGE 4

- For example to embed a QuickTime Player using the object element, define the source of the audio clip by adding the following param element:
 

```
<object data="overture.mp3" type="audio/mpeg">
  <param name="src" value="overture.mp3" />
</object>
```
- Notice that the filename is provided twice, once as a data attribute for the object and the second as a value attribute for the param

#### 48 ADDING A FLASH PLAYER FILE—PAGE 1

- Format to add a Flash player (.swf) video file using the object element:
 

```
<object data="url" type="application/x-shockwave-flash">
  <param name="movie" value="url" />
  parameters
</object>
```
- *url* is the location and filename of the SWF file
- The *parameters* are additional <param> elements that manage the appearance and actions of the player

#### 49 ADDING A FLASH PLAYER FILE —PAGE 2

- Some additional <param>'s for the Flash Player include:
  - *bgcolor* (“#rrggb” or “colorName”)—background color for Flash player, e.g.:
 

```
<param name="bgcolor" value="blue" />
```
  - *loop* (“true” or “false”)—plays in a continuous loop
 

```
<param name="loop" value="true" />
```
  - *menu* (“true” or “false”)—displays the full Flash pop-up menu when the user right-clicks the movie
 

```
<param name="menu" value="false" />
```

#### 50 ADDING A FLASH PLAYER FILE —PAGE 3

- Some additional <param>'s for the Flash Player (*con.*):

- play (“true” or “false”)—starts movie automatically when page loads  
`<param name="play" value="true" />`
- quality (“low” or “autolow” or “autohigh” or “medium” or “high” or “best”)— sets playback quality; low values favor speed over quality, while high values favor quality over speed  
`<param name="quality" value="high" />`

#### 51 **ADDING A FLASH PLAYER FILE—PAGE 4**

- Example:

```
<object data="rwdance.swf" type="application/x-shockwave-flash" width="280"
      height="239">
  <param name="movie" value="rwdance.swf " />
  <param name="quality" value="high" />
  <param name="menu" value="false" />
</object>
```

#### 53 **EMBEDDING VIDEOS FROM YOUTUBE—PAGE 1**

- YouTube videos are embedded using either the YouTube Shockwave Flash player file (.swf) or the user’s device and playing preferences, an HTML5 video player:

```
<object width="value" height="value">
  <param name="movie" value="url" />
  parameters
  <embed src="url" type="application/x-shockwave-flash" parameters />
</object>
```

#### 54 **EMBEDDING VIDEOS FROM YOUTUBE—PAGE 2**

- Some `<param>`’s for the YouTube player include:
  - autoplay (= 0 or 1)—indicates whether to play video automatically (1) or wait for user to press play button (0—default)
  - controls (= 0 or 1)—removes player controls (0) or displays them (1—default)
  - fs (= 0 or 1)—enables the use to play full screen (1) or not (0—default)
  - hd (= 0 or 1)—causes the high-definition feature of the video to play (1) or not (0—default)

#### 55 **EMBEDDING VIDEOS FROM YOUTUBE—PAGE 3**

- Some `<param>`’s for the YouTube player (*con.*):
  - loop (= 0 or 1)—set video to repeat (1) or to play only once (0—default)
  - playlist (= *list*)—plays a list of videos, in which *list* is the comma-delimited list
  - rel (= 0 or 1)—indicates whether to play a *list* of related YouTube videos (1—default) or disable the feature (0)
  - start (= *value*)—starts video playback *value* seconds into the video

#### 56 **EMBEDDING VIDEOS FROM YOUTUBE—PAGE 4**



- Example:

```
<object width="640" height="349">
  <param name="movie"
    value="http://www.youtube.com/v/iaN8M0p?fs=1&rel=0" />
  <param name="allowFullScreen" value="true" />
  <embed src="http://www.youtube.com/v/iaN8M0p?fs=1&rel=0"
    type="application/x-shockwave-flash"
    width="640" height="349"
    allowFullScreen="true" />
</object>
```

### 57 EMBEDDING VIDEOS FROM YOUTUBE—PAGE 5

- Use iframe tag:
- To easily:
  1. Go to YouTube and find the video
  2. Click the 'Share' button below the video
  3. Click the 'Embed' button next to the link
  4. Copy the entire iframe code and paste it into the HTML document

### 58 EMBEDDING VIDEOS FROM YOUTUBE—PAGE 6

- Example:

```
<iframe width="640" height="349"
  src="https://www.youtube.com/embed/ac6o8PXthzQ"
  frameborder="0"
  allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;
  picture-in-picture"
  allowfullscreen>
</iframe>
```

### 60 EMBEDDING INLINE IMAGES—PAGE 1

- The object element can be used in place of the img tag for embedding inline images
- Two advantages to this approach are:
  - Markup tags can be used for the alt attribute text
  - Users also can be provided with different formats of the same graphic image (if first one cannot be found???)

### 61 EMBEDDING INLINE IMAGES—PAGE 2

- Format:

```
<object data="url" type="image/jpeg" width="value" height="value">
</object>
```

- Example 1:

```
<object data="jplogo.jpg" type="image/jpeg" width="300" height="200">
</object>
```

## 62 EMBEDDING INLINE IMAGES—PAGE 3

- Example 2:

```
<object data="jplogo.png" type="image/png" width="300" height="200">  
  <object data="jplogo.jpg" type="image/jpeg" >  
  </object>  
</object>
```

## 64 EMBEDDING AN HTML FILE

- HTML code blocks can be embedded into web pages
- Format:

```
<object data="url" type="text/html" width="value" height="value">  
</object>
```

- Example:

```
<object data="secondPage.htm" type="text/html" width="600" height="400">  
</object>
```