

- 1 **HTML5 and CSS3**
AUDIO AND VIDEO
- 2
- 3 **Try It Out**
 - Add your name and the date to the existing web document
 - Add your name and the date to the two style sheet files
 - View the “jumbo.htm” web document in browser
- 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 different types and formats
- 5 **HTML Multimedia—Page 2**
 - Multimedia is displayed within a Web page in a fashion similar to an inline images:
 - Controls are displayed as part of the Web page
 - To play a multimedia file, a browser often will have access to a plug-in or add on
 - Starting with the widespread adoption of 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 the 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 Types and Compression**
- 9 **Converters**
- 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

11 Exploring Digital Audio

12 Exploring Digital Audio—Page 2

- People hear sounds as a continuously varying “wave” signal
- The sound wave must be converted to digital format to store as a computer file
- Digital recording measures the 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

13 Exploring Digital Audio

14 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

15 Exploring Digital Audio

16 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

17 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

18 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

19 The audio Element

20 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")
 - preload—specifies how audio 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 audio clip

21 The audio Element—Page 3

- Format:

```
<audio id="id" width="width" height="height" 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 video clip
- The type attribute in each source tag within the block specifies the “MIME-type” (the type of data in the file)

22 The audio Element—Page 4

- 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>
```

23 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 the categorization of the type, e.g. text, audio, video, etc.
 2. The subtype is specific to a particular file type
- [Examples](#) (there are over 600 of them for HTML):
 - text/css
 - audio/mp3

24 **The MIME-Type—Page 2**

- 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>
```

25 **Try It Out**

- Add an audio clip
- Add controls to the audio clip
- Link the "jumbo.htm" web document to the "clips.css" style sheet
- Add the stylesheet formatting
-

26 **Browser Support for audio Element**

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

27 **How Standards Proliferate**28 **The embed Element—Page 1**

- Older browsers that do not support the HTML5 audio element rely instead on plug-ins 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

29 **The embed Element—Page 2**

- Format:

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

- Example:

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

30 **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

31 **Try It Out**

- Add the embed object
- Add attributes for the QuickTime Player

32 **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

33 **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

34 **Exploring Digital Video**

35 **Video File Formats—Page 2**

- Some other video 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

36 **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 like “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

37 **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

38 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

39 The video Element—Page 4

- Format:

```
<video id="id" width="width" height="height" 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

40 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>
```

41 Try It Out

- Add the “Royal Wedding” video
- Add controls to view the video

42 The object Element—Page 1

- The object element was introduced in the specifications for HTML 4 for marking any kind of non-textual content
- Replaced the embed element which was 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.

43 **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

44 **The object Element—Page 3**

- Attributes for the <video> 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

45 **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" height="20" width="250">
```

```
  <param name="src" value="overture.mp3" />
```

```
</object>
```

- Notice that the filename is provided twice, once as an *attribute* for object and the second as a param

46 **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" width="value"
```

```
  height="value">
```

```
  <param name="movie" value="url" />
```

```
  parameters
```

```
</object>
```

- *url* is the location and filename of the SWF file
- *parameters* are additional <param> elements that manage the appearance and actions of the player

47 **Adding a Flash Player File —Page 2**

- Some additional <param>'s for the Flash Player include:

- bgcolor ("*#rrggbb*" or "*colorName*")—sets background color for the 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 when user right-clicks movie

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

48 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" />
```

49 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>
```

50 Try It Out

- Embed a Flash Player file
- Add parameters to the Flash Player
- Add the hypertext link

51 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" width="value"
        height="value"
```

```
        parameters />
```

```
</object>
```

52 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)

53 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

54 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=1 "
  />
  <param name="allowFullScreen" value="true" />
  <embed src="http://www.youtube.com/v/iaN8M0p?fs=1&rel=1"
    type="application/x-shockwave-flash"
    width="640" height="349"
    allowFullScreen="true" />
</object>
```

55 Embedding an Applet

56 Introducing Java

- Oak was developed by Sun Microsystems as an operating system intended to be used by common appliances and devices
- Oak was renamed Java in 1995
- Each Java program works with a Java Virtual Machine (JVM)

57 Applets and Java Virtual Machines

58 Applets

- Applets are displayed as embedded objects on a Web page
- Several libraries of Java applets are available
- The popularity of Java for web applications has declined more recently as more and more of its features can be duplicated with Flash or JavaScript
- The programming enhancements provided with HTML5 also will result in an even more reduced need for Java

59 Working with Java Applets

- To embed a Java applet, use the object element


```
<object type="application/x-java-          applet"
          width="value" height="value">
  <param name="code" value="url" />
  parameters
</object>
```

where the *width* and *height* attributes define the dimensions of the applet window, *url* specifies the location and filename of the Java file (usually the class file), and *parameters* represents the parameters associated with running the applet.

60 Inserting Java Parameters

61 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

62 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">
  <h1>Jumbo Popcorn</h1>
</object>
```

63 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" >
    <h1>Jumbo Popcorn</h1>
  </object>
</object>
```

64 Embedding an HTML File

- Web pages themselves can be embedded into a web page
- 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>
```

65

66 **The object Element and ActiveX**

- Internet Explorer supports a technology called ActiveX to play embedded media
- ActiveX employs reusable software components that can be run from within a variety of Windows programs
- Each ActiveX component is identified by a unique string of characters called the class id.

67 **The object Element and ActiveX**

68 **The codebase Attribute**

- When a browser encounters a plug-in or an ActiveX control that it doesn't recognize, it usually leaves a blank space where the embedded object normally would appear
- One way of dealing with this problem is to provide browsers with information about where a working version of the plug-in or control can be downloaded

69 **The codebase Attribute**

```
<object classid="clsid:02BF25D5-      8C17-4B23-BC80-D3488ABDDC6B"  
      codebase="http://www.apple.co m/qtactivex/qtplugin.cab">  
      parameters  
</object>
```