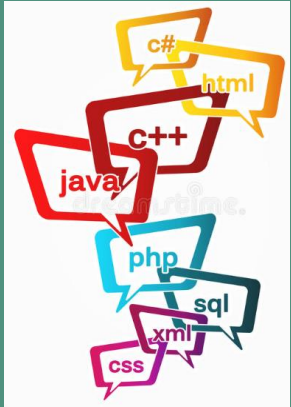


Internet Programming

Multimedia



Dan Brandon, Ph.D., PMP

Audio & Video

- In the not too distant past the power of PC's and the limited bandwidth to most Internet users made audio and video impractical
- With
 - the ever increasing power of PC's
 - the increase in bandwidth to a large number of users (faster modems, cable modems, satellite, fiber, etc.)
 - “steaming” transmission software
 - vector graphics
- Audio and video content are now practical



Linking to Media

- You may be able to link to different types of media (sound, video, etc.), and as long as the browser used by the client is equipped and configured for that media (without conflicting plug-ins), they may be able to experience it, but “controls” will be lacking:
 - <dl>
 - <dt>2. Approach to Earth (1 Mbyte)
 - <dt>3. Hawaiian Overflight (1.4 Mbyte)
 - <dt>4. Splashdown (845 Kbyte)
 - <dt>5. Swimming with the Dolphins (945 Kbyte)
 - <dt>6. Approach to the Mariana Trench (488 Kbyte)
 - <dt>7. A Journey through the Trench (262 Kbyte)
 - <dt>8. Back to the Surface...and Beyond (1.3 Mbyte)
 - </dl>

Direct Link to Music

[<http://facstaff.cbu.edu/dbrandon/Multimedia/ragtime.mid>]

- `<HTML><HEAD><TITLE>MUSIC</TITLE>`
- `</HEAD>`
- `<BODY>`
- `<H1>Music</H1>`
- `Click here for
music`
- `</BODY></HTML>`

The way these lesson examples work (or not) may change with browser and time.

Plug-ins and EMBED

- Plug-ins in older browsers are marked using the `embed` element

```
<embed src="url" type="mime-type"  
      width="value" height="value" />
```

where *url* is the location of the media file, *type* attribute provides the mime-type, and *width* and *height* attributes set the width and height of the media player

```
<EMBED SRC="Tada.wav" LOOP="true"></EMBED>
```

EMBED (con't)

- The `src`, `type`, `height`, and `width` attributes are generic attributes applied to `embed` element for any plug-in
- The following `embed` element adds attributes that are recognized by Apple's QuickTime Player to display the media player controls and prevent the playback from starting automatically:

```
<embed src="cp_overture.mp3"  
width="250" height="50"  
controller="yes" autoplay="no" />
```

EMBED (con't)

- Add the `embed` element to the end of the `audio` element as the last option for a browser that does not support HTML5 multimedia elements
- The browser will use the “application” defined for that type of media in the browsers preferences or options window (ie the Windows Media Player); the application may be built into the browser or may be a “plug-in” or “helper”
- Use of plug-ins has steadily declined since the widespread adoption of HTML5 standard

Simple Wave File

[<http://facstaff.cbu.edu/dbrandon/Multimedia/embed.html>]

- `<HTML>`
- `<HEAD><TITLE>EMBED</TITLE></HEAD>`
- `<BODY>`
- `<H1>Using the EMBED Element</H1>`
- `<EMBED SRC="Tada.wav" LOOP="true"></EMBED>`
- `</BODY>`
- `</HTML>`

EMBED (con't)

- For Netscape (prior to version 6+) you must use it's "LiveAudio" capability (which is a Java Applet) and start it
- The code below is one way to implement sound so that it works in both IE and Older Netscapes:

```
■ <SCRIPT LANGUAGE="JavaScript">
■ <!--
■   function startMusic()
■   {
■       if (document.layers)
■           document.theSound.play(false);
■   }
■ // -->
■ </SCRIPT>
■ </HEAD>
■ <EMBED SRC="myTune.mid" LOOP="true"
■       HIDDEN="true" NAME="theSound" MASTERSOUND>
■ <BODY ONLOAD="startMusic()">
```

BGSOUND Tag

[<http://facstaff.cbu.edu/dbrandon/Multimedia/bgsound.html>]

■ In IE, the BGSOUND tag may also be used:

- `<HTML><HEAD><TITLE>BGSOUND</TITLE>`
- `<BGSOUND ID = "audio"`
- `SRC = "debussy.mid"`
- `LOOP = "-1">`
- `</BGSOUND>`
- `<SCRIPT LANGUAGE = "JavaScript">`
-
- `</SCRIPT>`
- `</HEAD>`
- `<BODY>`
- `<H1>Background Music</H1>`
- `</BODY></HTML>`

MPEG

- Drives broke the \$1/MB in 1994
- By 2004 the industry was approaching \$1/GB, an improvement of 36 *thousand* times since 1989, and 10 million since the first computers
- Memory now cost under 1 cent per MB
- Currently the common formatting/compression method for audio is MP3 (Mpeg-1 audio layer 3)
- An 40 GB drive can hold up to 1000 CD's
- MP3 files can be “played” by a number of software products including the Windows Media Player
- Creating MP3 recordings takes two steps – “ripping” and “encoding”

Ripping & Encoding

- Ripping refers to the process of extracting the music from a media (as CD) into .wav files
 - Many software products are available for ripping/encoding (ie xingtech)
- Ideally this is done digitally, but in some situations it must be done in an analog manner (playing the CD thru the sound card and digitizing the output)
- Next the .wav files must be converted to MP3 format; the CD track database can also be encoded in MP3
- Most CD's can be digitally converted to MP3 at a rate of 160Kbits/sec with almost no loss of fidelity (the range of typical encoding is from 16 to 320Kbits/sec)
- At 160Kbits/sec, the compression ratio (from the .wav file) is about 9/1

MP3

- **What is MP3 (MPEG Layer 3)?**

MP3 is an audio file format

- In the past, audio files were quite large, which is why it hasn't been practical to store music on PCs
- An MP3 file is an audio file that has been compressed (anywhere from 1/5 to 1/20 of its original size) with little or no loss in sound quality
- That means a great sounding file, in a package small enough that it can be downloaded and/or stored on your computer

Recording

■ What is the difference between analog and digital recording?

The main difference is the speed with which you can record. Analog recordings will always be equal to the play speed, whereas digital recordings will be faster (depending on your drive, PC processor and compression type selected)

- The sound quality difference of analog recording compared with digital should be minimal
- Conversion errors are primarily dependent on your PC system
- Factors that will affect the sound quality are the soundcard and the required level of multimedia real-time performance

Napster

Existing Napster Members: sign in | redeem cards & codes



Napster has joined Rhapsody!

The #1 digital music subscription service is now even bigger.

Join America's largest community of music lovers just like you.
All the music you want on your phone, the web and everywhere in between. Try it free now!

Try It Free

Free for 14 days, then just \$9.99/month.

Build your own Music Profile on Rhapsody

Discovering and sharing music is easy when you create your own Music Profile. Connect with Facebook to see what your friends are listening to and follow other members to find new artists and playlists.

Browse ▾ My Music ▾



Jon Irwin▶ Jon Irwin's Radiof Connect

Fan of: Hitmakers, Adult Alternative, Hardcore Electronic, Post-Grunge, Adult Contemporary
I see a red door and I want it painted black

Profile

People

Library

Playlists

High Volume



U2

Skrillex

Eminem

Matchbox Twenty

Rob Thomas

Following 99 More

Playlists 56 More



Music for Airplanes



She Moves She

Four Tet

2:10 / 4:41

Mixer

Save as Playlist

Clear

Everything in its Right Place

Radiohead

The Wilhelm Scream

James Blunt

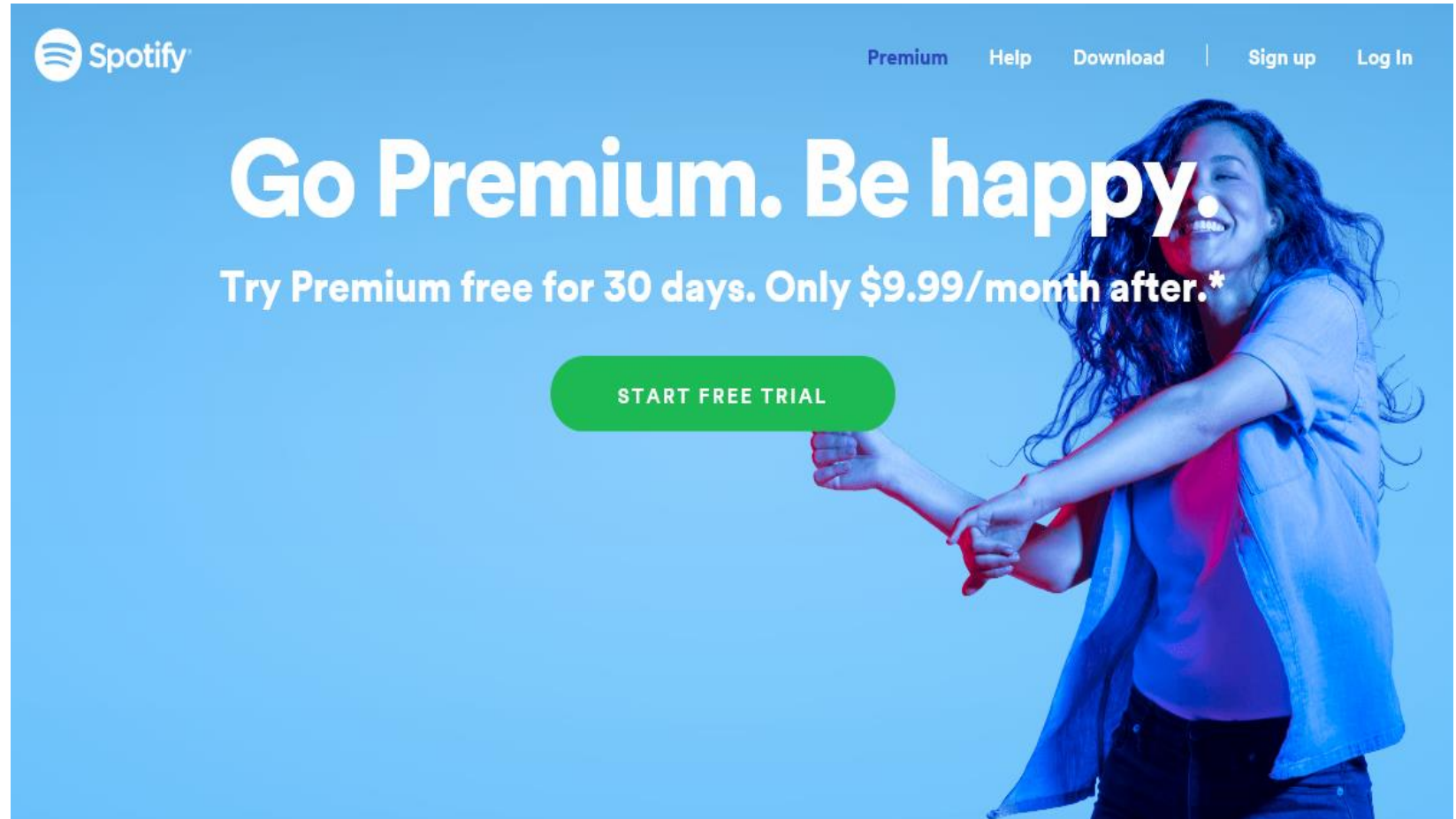
She Moves She

Four Tet

Skin Of The Night

James Blunt

Spotify

A promotional banner for Spotify Premium. The background is a solid blue color. On the right side, there is a photograph of a woman with long, dark, wavy hair, wearing a light blue button-down shirt over a white t-shirt, smiling and dancing. In the top left corner is the Spotify logo. In the top right corner, there are links for 'Premium', 'Help', 'Download', 'Sign up', and 'Log In'. The main text 'Go Premium. Be happy.' is in large white font. Below it, a smaller line of text says 'Try Premium free for 30 days. Only \$9.99/month after.*'. A green rounded rectangular button with the text 'START FREE TRIAL' is positioned in the center-left area.

 Spotify

[Premium](#) [Help](#) [Download](#) | [Sign up](#) [Log In](#)

Go Premium. Be happy.

Try Premium free for 30 days. Only \$9.99/month after.*

START FREE TRIAL

iTunes

[Store](#)[Mac](#)[iPod](#)[iPhone](#)[iPad](#)[iTunes](#)[Support](#)

[iTunes](#)[What's New](#)[What is iTunes](#)[What's on iTunes](#)[iTunes Charts](#)[How To](#)[Download iTunes](#)

Your entertainment. Everywhere.

With iTunes in the Cloud, the music, apps, and books you purchase automatically appear on all your devices. Or you can download only the stuff you want to just the devices you want.¹ It's all part of iCloud and iTunes 10.5.

**iTunes 10.5**
For Mac + PC

Free Download 



Automatic Downloads

Buy a song on one device and it automatically downloads to all your other devices — wirelessly and without syncing.

● ● ● ●

New iTunes in the Cloud.

Your entertainment on all your devices.

Now it's easy to have the great things you buy from the iTunes Store on your iPhone, iPad, iPod touch, Apple TV, and Mac or PC.

[Learn more >](#)



New iTunes Match

Your entire music collection in iCloud.

iTunes Match lets you store all your music in iCloud — even songs you've imported from CDs.²

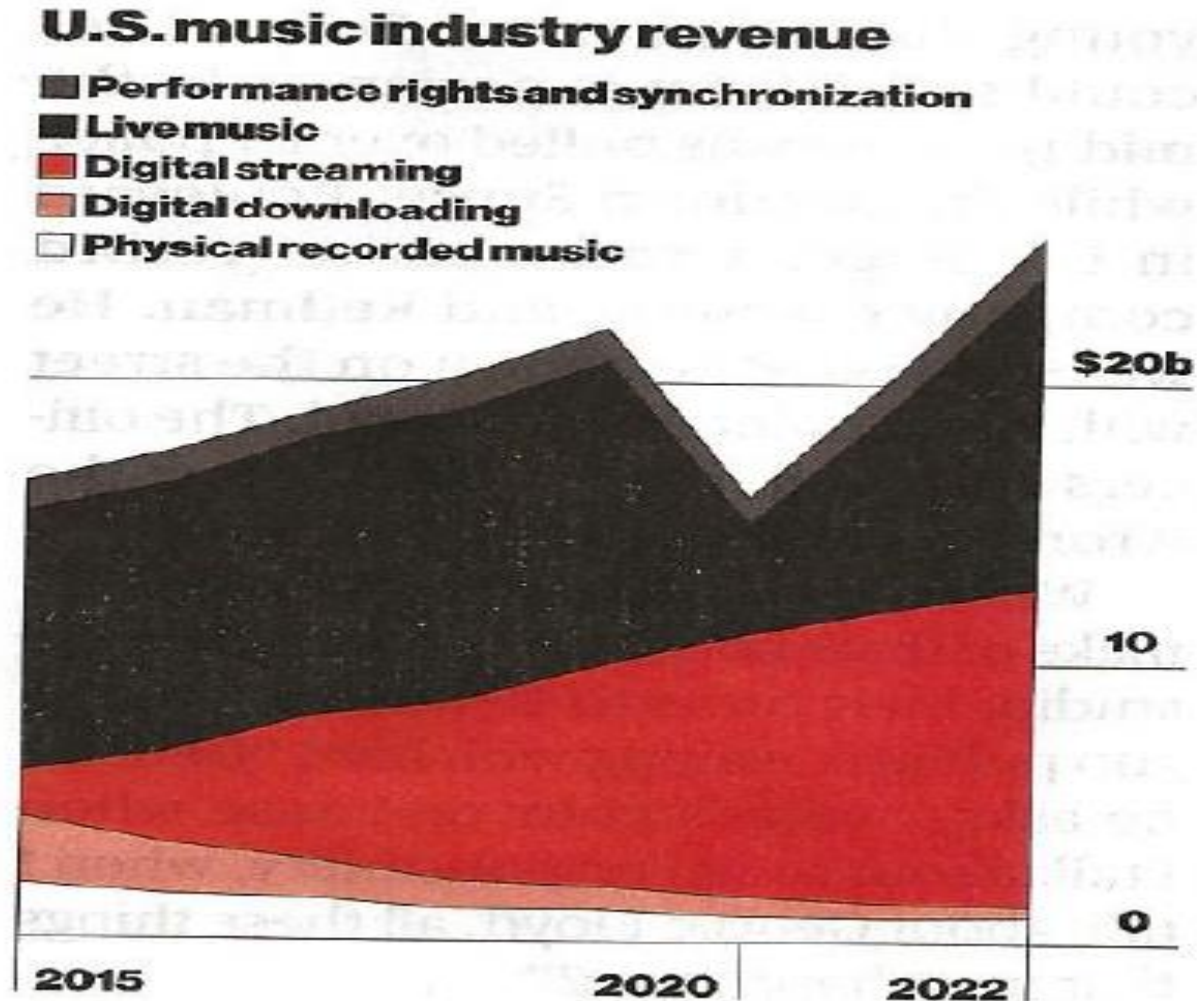
[Learn more >](#)



Gift Cards

Apple TV

US Music Revenue



DRM

- **What is DRM (Digital Rights Management)?**

Most PC jukeboxes support three types of DRM technologies that include InterTrust's DRM, Windows Media Technologies DRM and IBM EMMS

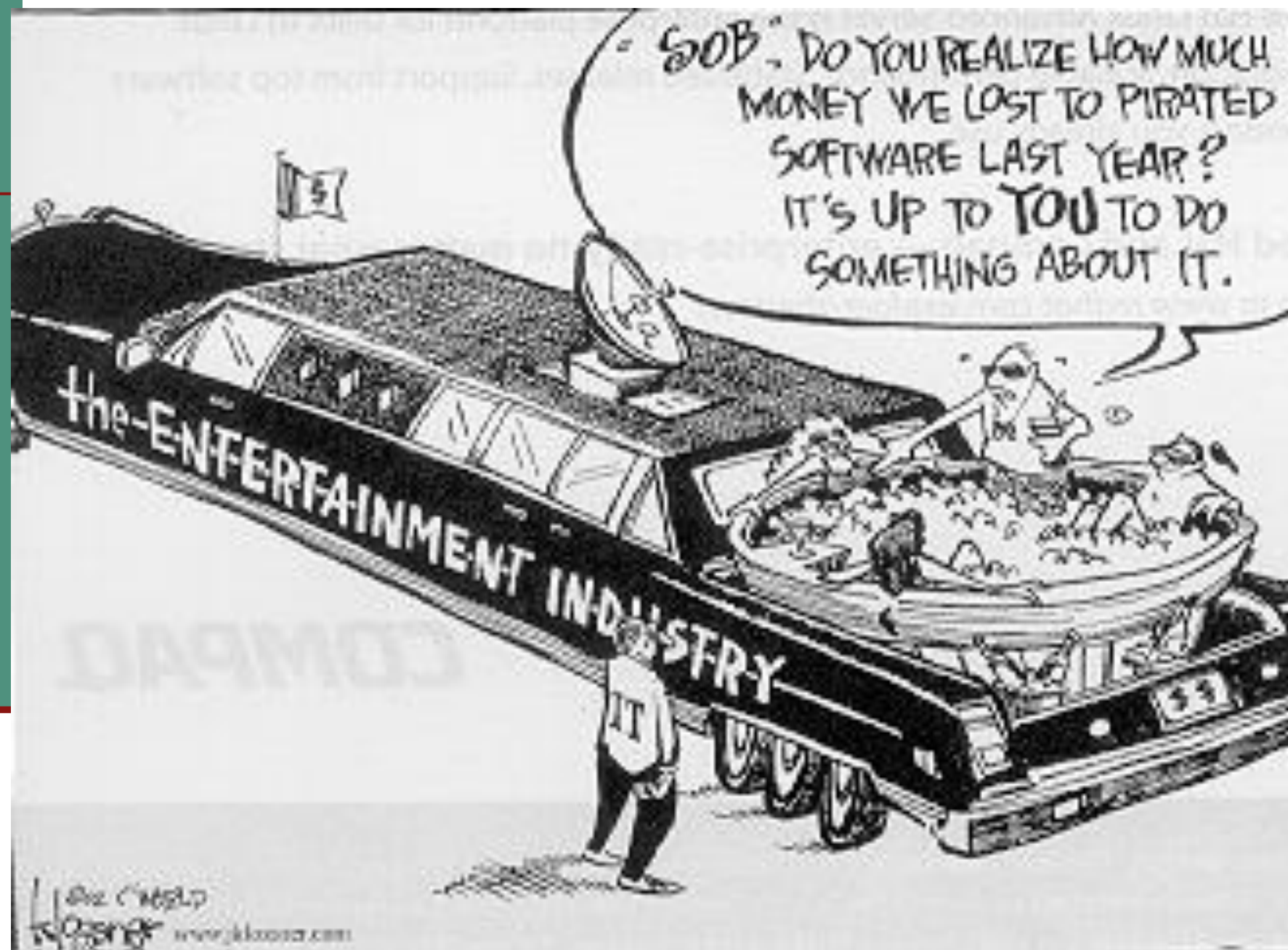
- InterTrust's sophisticated DRM technology is designed to protect and manage peoples' rights and interests in digital information
- The software platform enables new commercial models and markets by allowing people and organizations to define rules for using digital information, apply rules persistently after information is distributed and participate in a global system for digital commerce
- InterTrust's DRM platform is for all content providers who are developing a new spectrum of advanced Internet purchase and distribution models using the Web and other networks
- The Windows Media solution provides simple rights management capabilities for vending content over the Internet in an encrypted file format; this solution packages and encrypts the media files by locking them with a key
- Consumers need to obtain a separate license containing the key to play a packaged media file with the player
- Media files and licenses are stored and managed separately
- IBM's Electronic Music management system (EMMS) is a comprehensive e-business solution for the sale and secure downloading of music
- The EMMS copyright management solution for electronic music distribution features clearinghouse that tracks the sale of digital music content and manages access usage privileges as designated by the music owner

DRM (con't)

- **Why is DRM needed?**

The application of sophisticated DRM solutions in the electronic commerce environment is designed to rapidly expand the digital goods economy

- Digital rights management is needed across all digital content industries, and by all of the constituencies in these industries
- All parties want to get paid; artists and authors want to protect the integrity of their works
- Consumers want easy transparent access to good content but are concerned about protecting their privacy
- Producers, publishers and distributors want to structure and optimally manage their business models



Quicktime Movies

- `<EMBED SRC="myMovie.mov"
HEIGHT="320" WIDTH="240"
CONTROLLER="false" HREF="http or rtsp
content" TYPE=video/quicktime"
TARGET="myself"></EMBED>`

MPG or AVI Movies

[<http://facstaff.cbu.edu/dbrandon/Multimedia/splash.html>]

Using DYN SRC Attribute of IMG tag

- <HTML><HEAD>
- <TITLE>Dynamic Source</TITLE>
- <BGSOUND SRC="spinnin.mid" LOOP="-1">
- </HEAD>
- <BODY>
- <H1>Video via DYN SRC Property</H1>
- <IMG DYN SRC="splashDown.mpg"
- WIDTH="180" HEIGHT="135"
- LOOP="-1" ALT="Splash Down Under Ocean">
- </BODY></HTML>

Working with “Movies”

- Digital Video cameras and smartphones generally produce .mov files that need to be “edited” and converted to .avi files for web usage
- Apple’s QuickTime is free, but you need to buy QuickTime Pro to get the conversion utility to go from .mov to .avi
- Alternatively (and for free), you can go to www.radgametools.com and download “RAD Video Tools” to do the file conversion, editing, and make Windows friendly bundles that can play whether QuickTime is installed or not
- You can also convert individual frames of a .mov file into .jpg or gif files

Video Editing

- Grab videos from DVD camcorder
- Cut scenes
- Apply transitions, titles, and effects
- Save to new file (or load back into camcorder)
 - Screenblast Movie Studio
 - Microsoft Windows Movie Maker
 - Dazzle MovieStar
 - Ulead VideoStudio
 - WinDVD Creator Plus

Microsoft Live Movie Maker





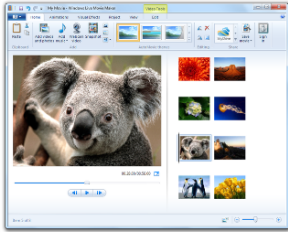
Windows Live Movie Maker

Lights, camera, action! Your very own movie studio. Whether you prefer Hollywood or the indie scene, you're the director with Movie Maker.

Quickly add photos and footage from your PC or camera into Movie Maker.

Free Download 


TESTED 21-NOV



Free downloads via [Soft32 Downloader](#). Additional [commercial offers](#) might be offered during the download process. The product may be available for download for free from the [manufacturer's website](#).

Windows Live Movie Maker's Review

Windows Live Movie Maker is a considerable step up from its forerunner, Windows Movie Maker, and if you're a Vista or Windows 7 user, then it's a free download. This software's main aim in life is to make life easier for the home video enthusiast. Because most of us own so much digital equipment these days, we're overloaded with photos and videos. It can sometimes be a royal pain trying to get them into a useful format to share with friends and family through various social networks. Although this is a fairly straightforward piece of editing kit, to be honest, the interface is a bit fiddly to use. The controls aren't necessarily in the most user-friendly place and you might need to do some scrolling around which is a tiny pain.

Pro's: The editing suite is pretty user-friendly and uses thumbnails in a handy timeline which makes it easy even for total beginners to edit their home movies.

Con's: The main drawback with this programme is that you're limited to exporting in the WMV Windows format which will frustrate some users.

Conclusion: The interface is not perfect, but it's a small moan really. Overall, if you can bear to put up with just exporting in WMV, then Windows Live Movie Maker makes it tons easier to upload films to the likes of You Tube, which is what it's all about, right?



★★★★★
by SOFT32 EDITORS

Name: Windows Live Movie Maker
Version: 15.4.3555
License: Free to try
Compatibility: Windows Vista

This site distributes software free of charge via Soft32 Downloader which is compatible with PCs running Windows XP, Vista, 7 or 8. Besides managing the download of your selected software, the download manager will make recommendations for commercial offers that you might be interested in. The additional software may include toolbars, browser add-ons and other types of software applications. You



4:21 PM
11/21/2013

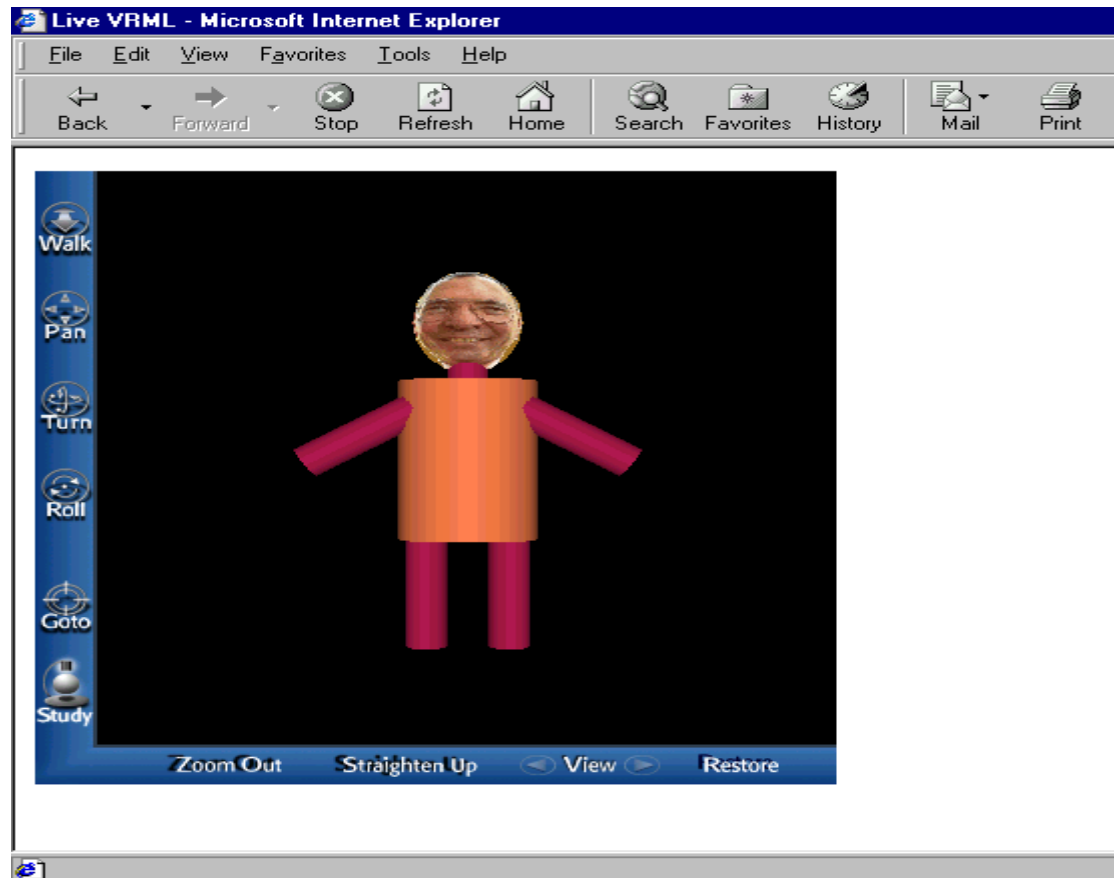
VRML

- Virtual Reality Modeling Language is a markup language used to describe 3D objects
- VRML, like HTML, is pure text and can be created with a plain editor like Wordpad or with special VRML editors; the file extension is “.wrl” (the files are called “worlds”)
- Many 3D modeling programs like CAD (computer aided design) programs can save their renderings in VRML format
- Netscape and IE have free plug-ins to display VRML files (not installed on CBU computers)

HTML for VRML

- <HTML>
- <HEAD><TITLE>VRML</TITLE></HEAD>
- <BODY>
- <EMBED SRC = "myVRMLFile.wrl"
- WIDTH = "400" HEIGHT = "400">
- </BODY>
- </HTML>

Vrml.html



YouTube Videos

- YouTube videos are easy to embed in a web page using YouTube's HTML5 video player
- You can link to a YouTube video
- Or you can **embed** a YouTube video
- Click the Share button below the YouTube video player to share it
- YouTube provides options to post a hypertext link to the video to a multitude of social media sites or to share the link via e-mail

YouTube Video (con't)

- To embed a video within a website, click Embed, which brings up a preview of the embedded player **and the HTML code** that needs to be added to the web page
- The general code for the embedded player is:

```
<iframe width="value"  
height="value" src="url"  
frameborder="0" allowfullscreen>  
</iframe>
```

YouTube Video (con't)

where,

- the `url` provides the link to the YouTube video
- `width` and `height` attributes define the dimensions of the player embedded on a web page
- `frameborder` attribute sets the width of the border around the player in pixels
- `allowfullscreen` attribute allows the user to play the video in full screen mode

Other Multimedia Products

- There are a number of special multimedia products
- For these products you need an editing product that allows you to create multimedia objects and drawings in the products special format
- The client needs the to install the “plug-in”, “helper”, or “viewer” on his PC and configure the browser to associate a particular file extension (and MIME type) with those types of files
- One product that used to be very popular is Macromedia’s (Adobe) “Flash” = it is no longer supported

OBJECT Element

- The `object` element is used to define browsers with plug-ins

```
<object attributes>
```

```
  parameters
```

```
</object>
```

where *attributes* define the object and *parameters* are values passed to the object controlling the object's appearance and actions

- Parameters of the object are defined using `param` element

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

where *name* is the name of the parameter and *value* is the parameter's value

HTML 5 Audio

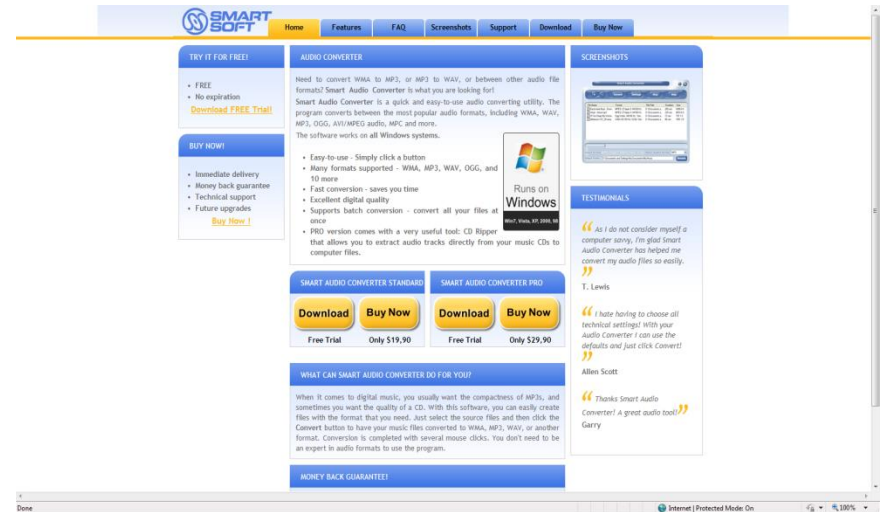
- HTML5 has the “audio” element:
 - `<audio src=“...” attributes></audio>`
 - Or
 - `<audio src=“...” attributes />`
- Attributes are:
 - `autoplay` – true/false
 - `controls` – true/false
 - `loop` – true/false
 - `preload` –none/metadata/auto
 - `onerror` - failure event

HTML 5 Audio (con't)

Attribute	Description
<code>autoplay</code>	Starts playing the media clip as soon as it is loaded by the browser
<code>controls</code>	Displays the player controls in the web page
<code>loop</code>	Automatically restarts the media clip when it is finished playing
<code>muted</code>	Specifies that the audio output should be muted
<code>preload="auto metadata none"</code>	Specifies whether the media clip should be preloaded by the browser, where <code>auto</code> preloads the entire clip, <code>metadata</code> preloads descriptive data about the clip, and <code>none</code> preloads nothing. <code>metadata</code> (to preload only descriptive data about the clip), or <code>none</code> (not to preload the media clip)
<code>src="url"</code>	Specifies the source of the media clip, where <code>url</code> is the location and name of the media file

OGG (Theora)

- Originally HTML5 browsers could only use the **OGG format** for audio & video, but that will change in time
- Many file converters are available
 - <http://www.smartaudioconverter.com/>



HTML 5 Audio (con't)

- HTML does not specify any one audio format
 - Developers can pick a format that meets the needs of their customers and clients
- Browsers and devices differ on support
 - Before choosing an audio format, determine whether the intended browser(s) will be able to play it
 - Currently (2019):

Browser	MP3	AAC	Ogg	WAV	FLAC	WebM
Chrome	✓	✓	✓	✓	✓	✓
Firefox	✓	✓	✓	✓	✓	✓
Microsoft Edge	✓	✓	✓	✓	✓	✓
Opera	✓	✓	✓	✓	✓	✓
Safari	✓	✓		✓	✓	

HTML 5 Audio (con't)

Format	Description	Codec	File Extension(s)	MIME Type
MP3	MPEG-1 Audio Layer 3 or MP3 is one of the most widely used audio types and is the standard format for digital audio players	MP3	.mp3	audio/mpeg
AAC	Advanced Audio Coding or AAC is the encoding standard for all Apple products, as well as YouTube and several gaming systems and mobile devices; AAC was introduced as the successor to MP3 with the goal of achieving better sound quality at similar compression ratios	AAC	.aac .mp4 .m4a	audio/mp4

HTML 5 Audio (con't)

Format	Description	Codec	File Extension(s)	MIME Type
OGG	A file compression format designed for web audio, Ogg is an open-source and royalty-free format; in general, Ogg provides better sound quality than MP3, especially at lower bit rates	Vorbis	.ogg	audio/ogg
WAV	The original audio format for Windows PCs, WAV is commonly used for storing uncompressed audio, making it impractical for all but the shortest audio clips	PCM	.wav	audio/wav

HTML 5 Audio (con't)

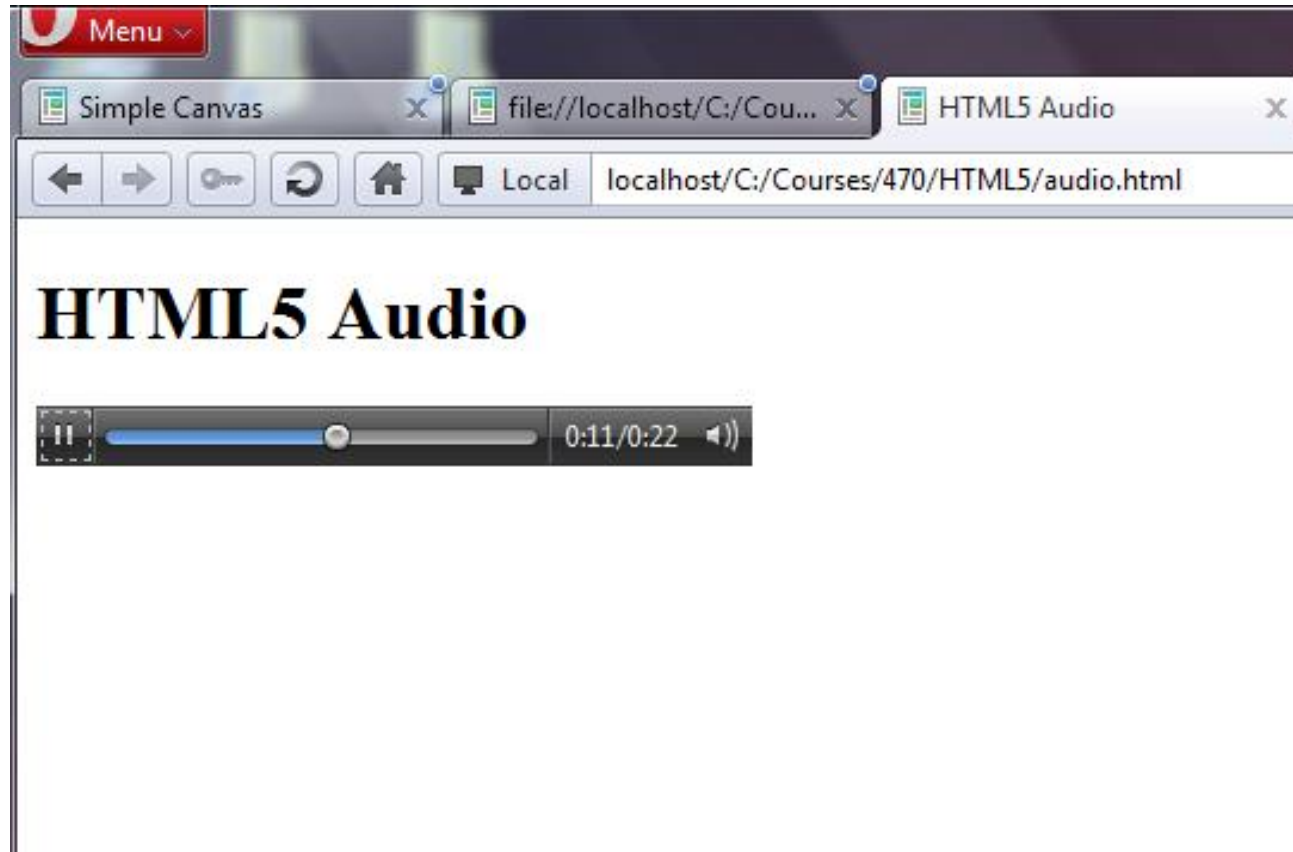
Format	Description	Codec	File Extension(s)	MIME Type
FLAC	Free Lossless Audio Codec or FLAC is a free open-source digital music format that offers bit-perfect copies of CD-quality music at half the size	FLAC	.flac	audio/flac
WebM	WebM is an open-source format introduced by Google to provide royalty-free audio and video to be used with the HTML audio and video elements	VP8 VP9	.webm	audio/webm

Audio Example

- <!DOCTYPE html>
- <html>
- <head>
- <meta charset="utf-8">
- <title>HTML5 Audio</title>
- </head>
- <body>
- <h1>HTML5 Audio</h1>
- <audio controls src="violin.ogg"></audio>
- </body>
- </html>

Audio Example (con't)

[<http://facstaff.cbu.edu/dbrandon/Multimedia/audio.html>]



HTML 5 Audio (con't)

- To support the most browsers, nest several `source` elements within a single `audio` element to provide several versions of the same media file

```
<audio>
```

```
<source src="url1" type="mime-type" />
```

```
<source src="url2" type="mime-type" />
```

```
... </audio>
```

where, *url1*, *url2*,... are the URLs for each audio file and *mime-type* specifies the audio format associated with each file

HTML 5 Audio (con't)

```
<p>Click the play button below to hear the musical overture  
for Burton Lane's <cite>Royal Wedding</cite>.</p>  
<audio controls>  
  <source src="cp_overture.mp3" type="audio/mp3" />  
  <source src="cp_overture.ogg" type="audio/ogg" />  
</audio>
```

displays the controls
for the audio player

two possible
sources for
the audio file

```
<audio controls>  
  <source src="cp_overture.mp3" type="audio/mp3" />  
  <source src="cp_overture.ogg" type="audio/ogg" />  
  <p><em>To play this audio clip, your browser needs to support HTML5.</em></p>  
</audio>  
</aside>
```

displays this paragraph
when the audio element
is not supported

HTML 5 Audio (con't)

- The appearance of a media player is determined by the browser itself
- CSS can be applied to set the width of the media player, add borders and drop shadows, and apply filters and transformations to the player's appearance

```
audio {  
  box-shadow: rgb(51, 51, 51) 8px 8px 15px;  
  display: block;  
  margin: 10px auto;  
  width: 90%;  
}
```

HTML5 Video

- A video file contains codecs for the following:
 - Audio
 - Video images
- The most popular video codec is H.264 used by YouTube and most commercial vendors
- Because H.264 is a commercial product, it is not royalty free

HTML5 Video (con't)

Codec	Description
H.264	Developed by the MPEG group, the H.264 codec is the industry standard for high-definition video streams, movie sharing websites such as YouTube, and video plug-ins
Theora	Theora is a royalty-free codec developed by the Xiph.org Foundation that produces video streams that can be used with almost any container
VP8	VP8 is an open-source royalty-free codec owned by Google for use in Google's WebM video format
VP9	VP9 is Google's successor to the VP8 codec, offering the same video quality as VP8 at half the download size

HTML5 Video (con't)

Format	Description	Video Codec	File Extension(s)	MIME Type
MPEG-4	MPEG-4 or MP4 is a widely used proprietary format developed by Apple based on the Apple QuickTime movie format	H.264	.mp4 .m4v	video/mp4
Ogg	Ogg is an open-source format developed by the Xiph.org Foundation using the Theora codec as an alternative to the MPEG-4 codec	Theora	.ogg	video/ogg
WebM	WebM is an open-source format introduced by Google to provide royalty-free video and audio to be used with the HTML5 video element	VP8 VP9	.webm	video/webm

HTML5 Video (con't)

- Current (2019) browser support:

Browser	MPEG-4	Ogg	WebM
Chrome	✓	✓	✓
Firefox	✓	✓	✓
Microsoft Edge	✓	✓	✓
Opera	✓	✓	✓
Safari	✓		

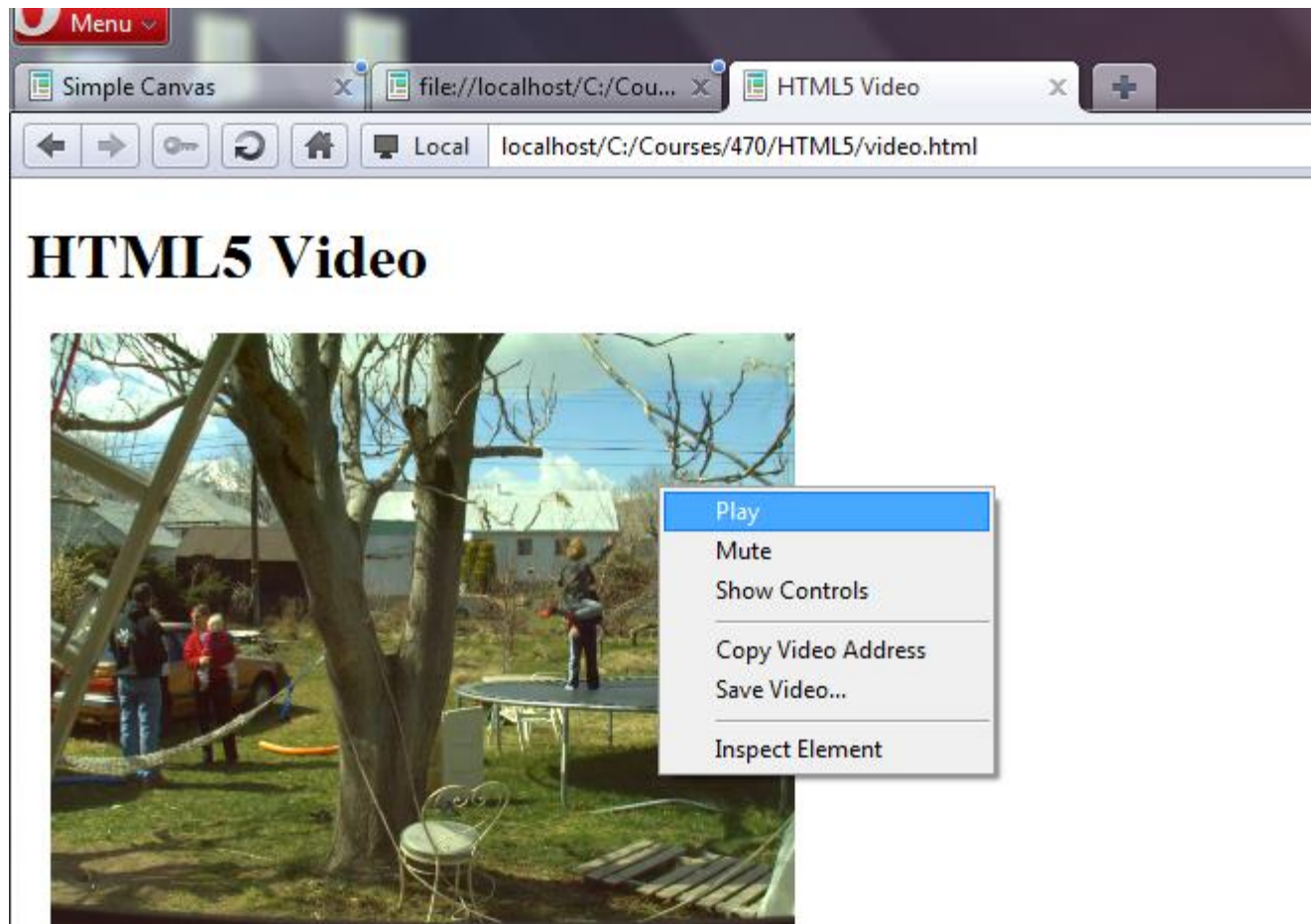
HTML5 Video

- HTML5 has the “video” element:
 - `<video height=“yy” width=“xx” src=“...”></video>`
- Attributes are:
 - autoplay – true/false
 - controls – true/false
 - height & width
 - loop – true/false
 - poster – URL for image if video is unavailable
 - preload –none/metadata/auto
 - onerror - failure event

Simple Video Tag

```
■ <!DOCTYPE html>
■ <html>
■     <head>
■         <meta charset="utf-8">
■         <title>HTML5 Video</title>
■     </head>
■     <body>
■         <h1>HTML5 Video</h1>
■         <video height="300" width="400" src="j0011_5.ogg"></video>
■     </body>
■ </html>
```

Right Click to Play



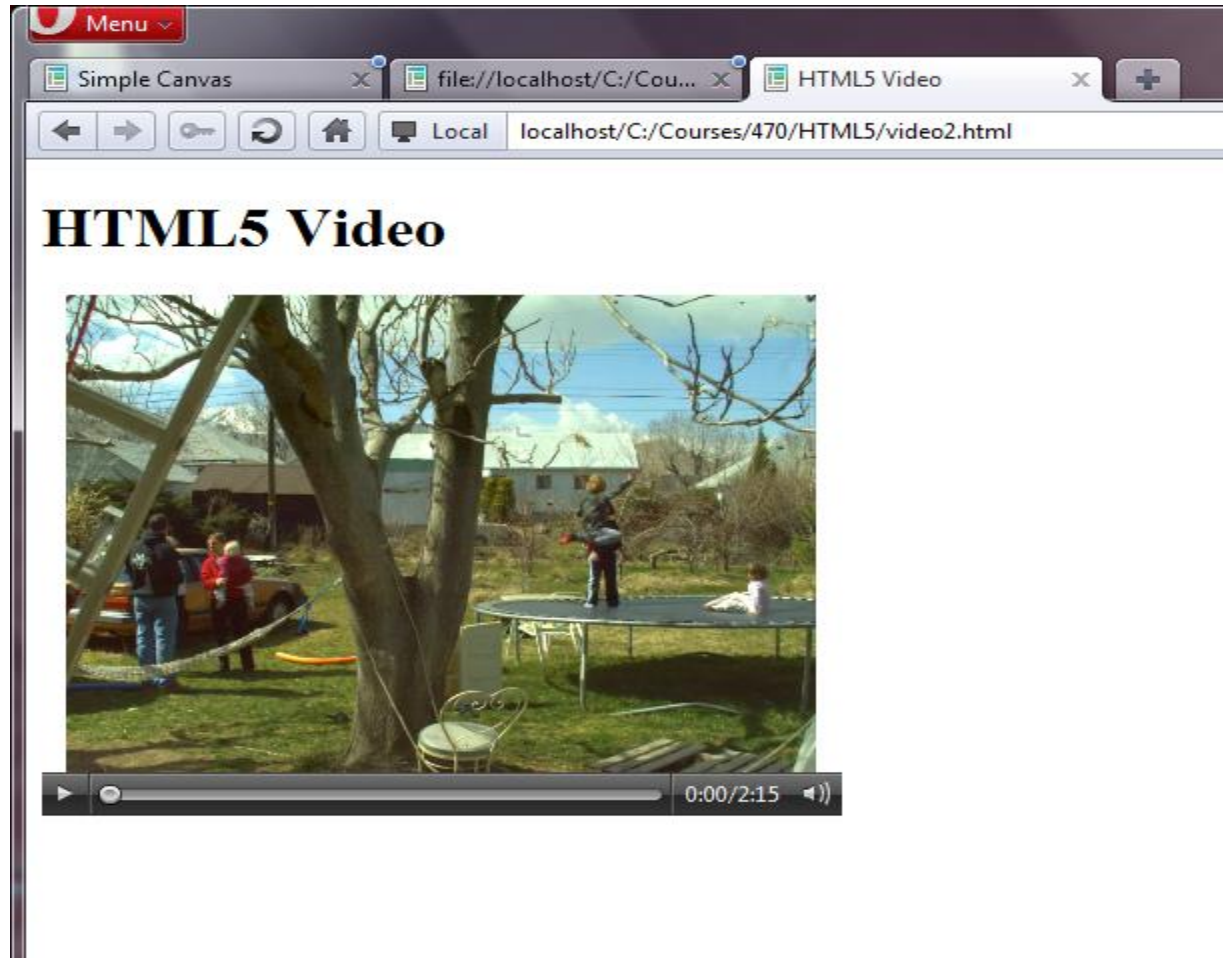
Adding Controls

- <!DOCTYPE html>
- <html>
- <head>
- <meta charset="utf-8">
- <title>HTML5 Video</title>
- </head>
- <body>
- <h1>HTML5 Video</h1>
- <video controls height="300" width="400"
src="j0011_5.ogg"></video>
- </body>
- </html>

Adding Controls (con't)

[<http://facstaff.cbu.edu/dbrandon/Multimedia/video2.html>]

[controls show when mouse over video]



HTML5 Video (con't)

- Multiple sources can be specified:

```
<video attributes>
```

```
<source src="url1" type="mime-type" />
```

```
<source src="url2" type="mime-type" />
```

```
... </video>
```

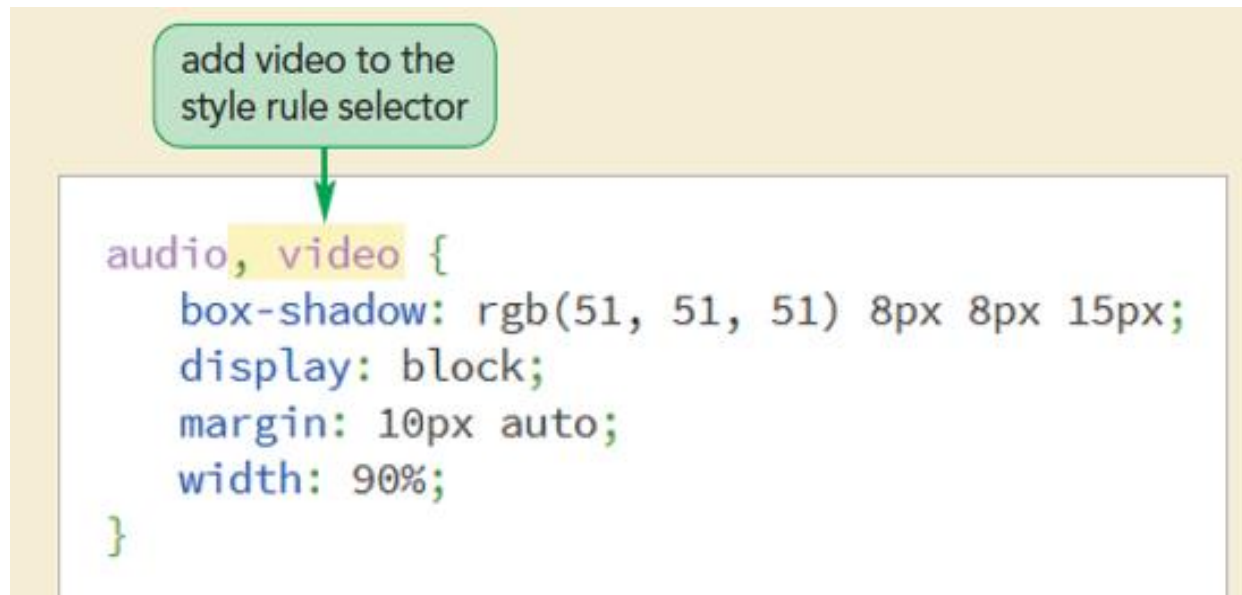
where *attributes* are HTML attributes that control the behaviour and appearance of the video playback, *url1*, *url2*,... are the possible sources of the video, and *mime-type* specifies the format associated with each video file

HTML5 Video (con't)



HTML5 Video (con't)

- CSS styling can be added:



HTML5 Video (con't)

- The `poster` attribute defines a video's preview image

```
<video poster="url">
```

```
...
```

```
</video>
```

where `url` points to an image file containing the preview image

- The `poster` attribute is used as a placeholder image that is displayed when a video is being downloaded

Making a Quality HTML 5 Video

- 1. **Use high quality video:** This means using video in the original format. It's important to be aware that video compression is lossy and when you use it will degrade the quality of your video.
- 2. **Use a Tripod:** For image stability and clarity, this is necessary. Do not use handheld shots unless there is no other choice. As for the camera, open use a higher f/stop setting (wider aperture) in low light situations, if your camera supports it. This means that more light will reach the sensor and improve the quality of your shots.
- 3. **Use an External Microphone:** Most cameras have an external input that you can make use of. A lavalier microphone could do the trick or if you're using the video for a podcast later, you could make use of a Rode, Blue Yeti or Samson Meteor microphone. Built in microphones should only be used as a last resort because they don't perform as well.
- 4. **Lighting:** It is crucial to have good lighting and to make use of reflectors (umbrellas), diffusion and scrims to shape the lighting if necessary. If the lighting is poor, the end result will be a lot of noise in your video. The effect will be magnified with compression.
- Ideally, it's wise to shoot in situations with a lot of natural light, but some situations will not allow for that. In those cases, using simple (and inexpensive) work lamps, like those found at construction sites could do the trick.
-

Making a Quality HTML 5 Video (con't)

- 5. **Shoot Imagery Which Is Easy to See At a Small Scale:** Since you're going to be working with smaller screens (especially with mobile) you need to take that into account and create video imagery that is easy to see. You will want to maximize the viewing space and if you use text, use it sparingly and at a large size. Too small and you run the risk of losing your audience.
-
- 6. **Video Conversion:** A simple solution for the video conversion is: <http://www.mirovideoconverter.com/>. This is a free video tool which you can use to convert your files into web formats.
-
- 7. **Make Sure Your Memory Card Supports Camera Movement (Panning) at High Resolution Settings:** If not, your pan will be choppy instead of fluid and the result will be unusable. As an example, the pan might work perfectly at 640x480 but become choppy at an HD setting.
-
- 8. **Shoot in Progressive Mode:** Interlaced video can have artifacts which will damage the quality of the image.
-

Making a Quality HTML 5 Video (con't)

- 9. **Shoot with a Shallow Depth of Field:** The result is closer objects stay in focus and shots behind the close objects become blurry. Outside of the nice effect, the blurry parts are easier to compress.
-
- 10. **Wear Appropriate Clothing:** While it's important to wear comfortable clothing, some clothing doesn't work well with video. Here are some guidelines:
 - a. Avoid very light or very dark colors (including navy) because these clothes could lose their detail on video
 - b. Avoid red or orange. Red can sometimes appear to "crawl" on video
 - c. Solid colors are good
 - d. Avoid checked patterns such as hounds-tooth, tight stripes, plaid or herringbone. These have a tendency to create Moire' patterns (rainbow effects) on video
 - e. Buttoned clothing is good (for a Lavalier or clip-on microphone)
 - f. If you wear glasses, be careful of frames that cause reflections.
 - g. Be careful with jewelry, especially dangly earrings or bracelets which could cause noise. Also, make sure there aren't any reflection issues.

HTML5 makes use of three major file types: MP4, WebM, and Ogg/Ogv. The MPEG-4 file type is generally encoded in H.264. Not all browsers support all three formats.

BROWSER	H264	H265	VP8	VP9	AAC	MP3	VORBIS	OPUS
Chrome for Desktop	30+	-	30+	30+	30+	30+	30+	33+
Internet Explorer for Windows	9+	-	-	-	9+	9+	-	-
Firefox for Desktop	22+ ¹	-	20+	28+	22+ ¹	22+ ¹	20+	20+
Safari for Mac	3+	-	-	-	3+	3+	-	-
Opera for Desktop	-	-	11+	16+	-	-	11+	12+
Safari for iOS	3+	-	-	-	3+	3+	-	-
Stock Browser for Android	2.3+	-	4.0+ ²	-	2.3+	2.3+	4.0+ ³	-
Chrome for Android	30+	- ⁴	30+ ²	-	30+	30+	30+	-
Internet Explorer for WinPhone	7.5+	-	-	-	7.5+	7.5+	-	-
Firefox for Android	20+	-	20+ ²	28+ ²	20+	-	20+	20+

1. Firefox supports H264/AAC/MP3 on OSX as of version 35.
2. Software decoding only, impacting performance and battery life.
3. `canPlayType()` not reporting on Vorbis, but audio is playing
4. Android Lollipop contains a software decoder that makes H265 play in Chrome.

Making a Quality HTML 5 Video (con't)

- With the different browsers, you will only be able to use two formats at most. The basic code would be:
 - `<video width="640" height="480" controls>`
 - `<source src=bucerias.mp4" type="video/mp4" />`
 - `<source src="bucerias.ogv" type="video/ogg" />`
 - `</video>`
- If you are testing this within video design software, it may not work. Always, upload the code to your server to test it. Do not trust the preview in web design software. The acid test is ALWAYS testing the code live.
- Here are some optional video controls:
 - • autoplay: With this attribute, the video will play back automatically without stopping to load the rest of the data.
 - • autobuffer: Here, the video will automatically buffer, even if it's not set to play automatically.
 - • controls: With this attribute you can control video playback, volume and pause or resume.
 - • loop: Here, the video will automatically loop back to the beginning after reaching the end.
 - • preload: With this attribute, the video will loaded at page load and will be ready to play. This is ignored if autoplay is used.

HTML5 Video Players

- HTML5 video players work within a browser with CSS and JavaScript files
- It presents a customizable player that can be adapted to the needs of business or organization
- For example, YouTube player that provides both the player and a hosting service for the video content

HTML5 Video Players (con't)

■ HTML5 includes video players

■ JWPlayer

- A popular commercial player that supports both HTML 5 and Flash video

■ VideoJS

- A free player that works with the popular WordPress HTML framework

■ Flowplayer

- Originally marketed as a Flash player, Flowplayer is a commercially licensed audio and video player, payable as a one-time fee for perpetual use

■ Kaltura Player

- A free open-source video player that can be customized with user-created player controls and skins to match the design of one's website

■ Wistia

- An HTML 5 video player focused on business needs and applications

References

- HTML5 Multimedia: Develop and Design by Ian Devlin
- Multimedia Demystified by Jennifer Coleman Dowling
- Multimedia Web Design and Development: Using Languages to Build Dynamic Web Pages by Theodor Richardson and Charles Thies

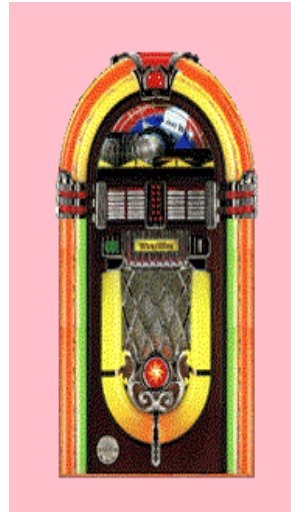
Project Two Deliverables

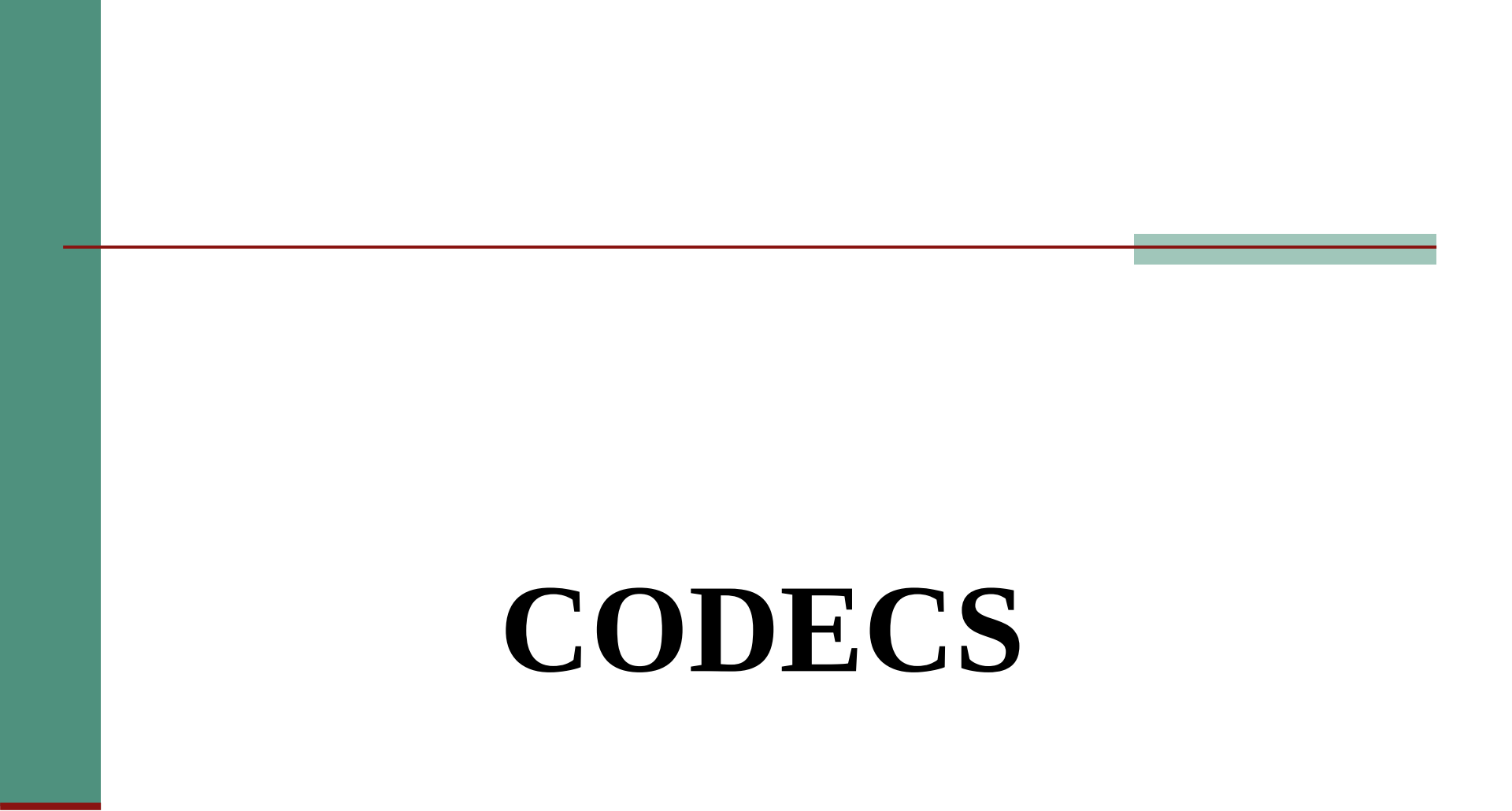
- Selected industry area web site
- Common external stylesheet for all pages
- Forms
- Multimedia 
- Rollovers
- JS validation of form data

Homework

[<http://facstaff.cbu.edu/dbrandon/DansJukebox/DansJukebox.html>]

- Textbook Chapter 8
- Appendices on **Codecs, Text Tracks, Animations/Transition, & Flash →**
- Create a web page for your own jukebox
- Have a selection of songs (via drop down or radio buttons) that can be played (you can use .wav, .mid, or .mp3 [maybe] file format)
 - Note that the CBU webmaster policy may not allow you to place full MP3 files in your web root
- Email URL to instructor





CODECS

Codec

- **Codec:** Computer program that encodes and decodes streams of data
- Codecs compress data to transmit it in fast and efficient manner
- Codecs decompress data when it is to be read or played back
- The compression method can be either lossy or lossless

Compression

- **Lossy compression:** Nonessential data is removed in order to achieve a smaller file size
- Example:
 - An audio file might be compressed by removing sounds that the human ear can barely hear
- The more the file is compressed, the more the content is lost
- Data removed during compression cannot be recovered

Compression (con't)

- **Lossless compression:** Data is compressed by removing redundant information
- Example:
 - AAAABBBBBCCCCCCC requires 15 characters of information, which can be rewritten using 6 characters as 4A5B6C
- Lossless compression cannot achieve the same level of compression as with lossy compression

Compression (con't)

- Most codecs involve some combination of lossy and lossless techniques
- Codecs are placed within a container that handles the packaging, transportation and presentation of data
- **Container** is the file format identified by a file extension

Containers

- The web supports a multitude of container and codec combinations
- Not all containers and codecs are equally supported
- Example
 - Google Chrome uses the WebM container for video content, compressing that data with the VP8 codec; however, that combination of container and codec is not supported by any Apple device

Plug-ins

- **Media player:** Decodes and plays multimedia content stored within a container file
- **Plug-in:** Software program accessed by a browser to provide a feature or capability not native to the browser
- A plug-in either opens in its own external window or runs within the web page as an **embedded object**

Plug-ins (con't)

- Problems with the plug-in approach for delivery of multimedia content
 - Plug-ins require users to install a separate application in addition to their web browsers
 - A common plug-in is not available across all browsers, operating systems, and devices
 - HTML documents that support multiple plug-ins are difficult to create and maintain
 - Plug-ins consume valuable system resources, resulting in slow and unreliable performance
 - Plug-ins are a security risk with some of the most prominent Internet attacks working their way into browsers via a plug-in



ADDING TEXT TRACK TO VIDEOS

Text Tracks

- A text track that needs to be read or recited to visually impaired users can be added to a media clip
- Allows audio and video content to be accessible to all users
- Text tracks are added to an audio or video clip using `track` element

Text Tracks (con't)

```
<track kind="type" src="url"  
label="text" srclang="lang" />
```

where,

- the `kind` attribute defines the track type
- the `src` attribute references a file containing the track text
- the `label` attribute gives the track name
- the `srclang` attribute indicates the language of the track

Text Tracks (con't)

Kind Value	Description
<code>captions</code>	Brief text descriptions synced to specified time points within the media clip; designed for hearing impaired users
<code>chapters</code>	Chapter titles used by the media player to navigate the user to specific time points within the media clip
<code>descriptions</code>	Longer descriptions synced to specified time points within the media clip; designed for visually impaired users
<code>subtitles</code>	(the default) Translation of dialog from the media clip; the language of the subtitle must be specified in the <code>srclang</code> attribute
<code>metadata</code>	Metadata content used by external scripts accessing the media file

Text Tracks (con't)

- Tracks are stored as simple text files written in **Web Video Text Tracks** or **WebVTT** language
- Format of a WebVTT file

WEBVTT

cue1

cue2

...

where *cue1*, *cue2*,... are cues matched with specific time intervals within a media clip

Text Tracks (con't)

- List of cues is separated by a single blank line after a cue text
- White space is not ignored in WebVTT files
- General form of a cue

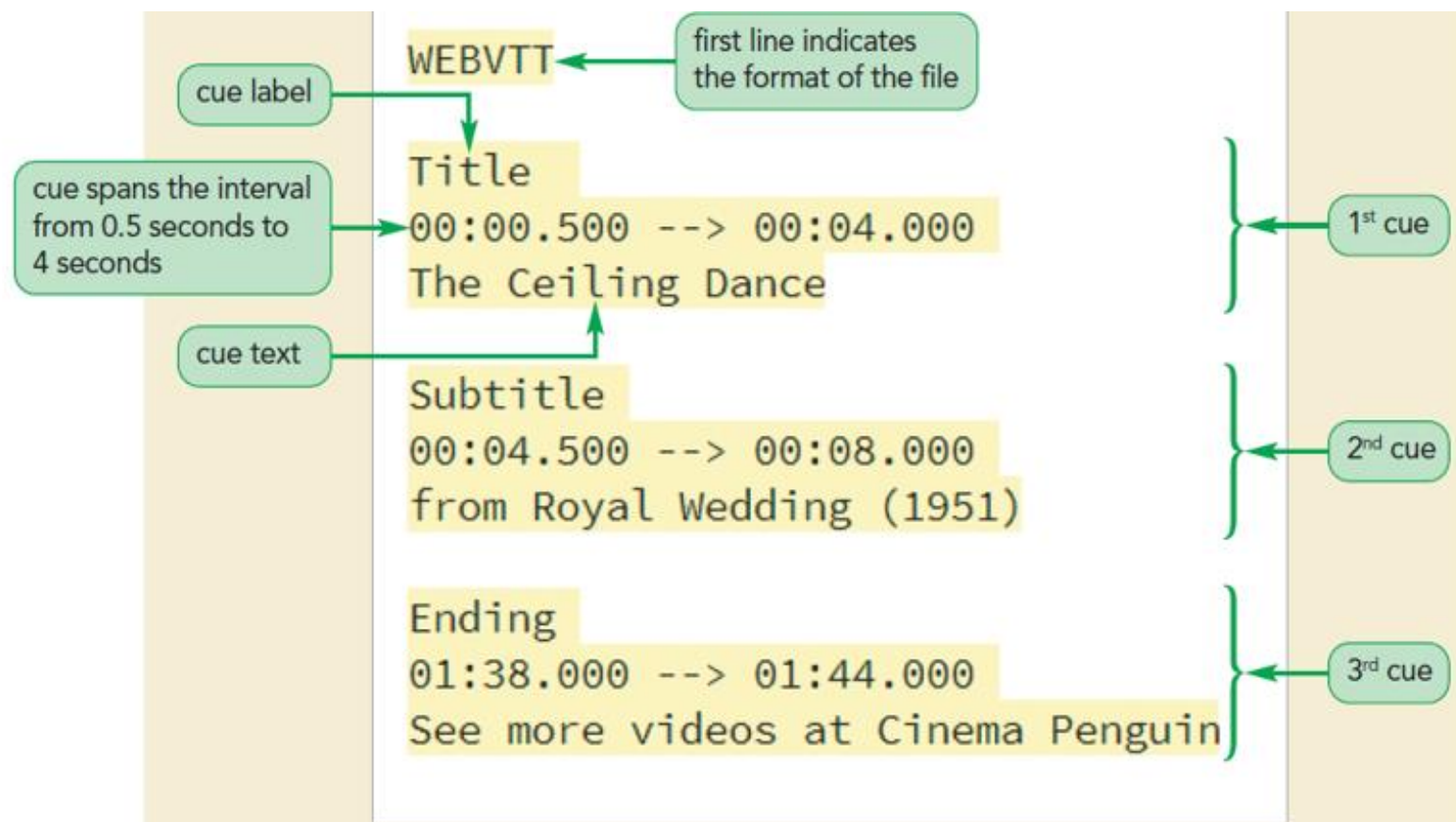
label

start --> stop

cue text

where *label* is the name assigned to the cue, *start* and *stop* define the time interval, and *cue text* is the text of the cue

Text Tracks (con't)



Text Tracks (con't)

- Size and position of a cue text can be set using cue settings directly after the cue's time interval

setting1:value1 setting2:value2 ...

where *setting1*, *setting2*, ... define the size and position of the cue text and *value1*, *value2*, ... are the setting values

- There is no space between the setting name and value

Text Tracks (con't)

Cue Setting	Description
<u>align</u> : <i>value</i>	Sets the horizontal alignment of the text within the cue, where <i>value</i> is <i>start</i> (left-aligned), <i>middle</i> (center-aligned), or <i>end</i> (right-aligned)
<i>line</i> : <i>value</i>	Sets the vertical position of the cue within the video window, where <i>value</i> ranges from 0% (top) to 100% (bottom)
<i>position</i> : <i>value</i>	Sets the horizontal position of the cue within the video window, where <i>value</i> ranges from 0% (left) to 100% (right)
<i>size</i> : <i>value</i>	Sets the width of the cue as a percentage of the width of the video window
<i>vertical</i> : <i>type</i>	Displays the cue text vertically rather than horizontally where <i>type</i> is <i>rl</i> (writing direction is right to left) or <i>lr</i> (writing direction is left to right)

Text Tracks (con't)

places the Title and Subtitle cues near the top of the video window with the text centered

```
Title
00:00:00.500 --> 00:00:04.000 line:5% align:middle
The Ceiling Dance

Subtitle
00:00:04.500 --> 00:00:08.000 line:5% align:middle
from Royal Wedding (1951)

Ending
00:01:38.000 --> 00:01:44.000 line:80% position:95% align:end
See more videos at Cinema Penguin
```

places the Ending cue near the bottom-right corner of the video window with the text right-aligned

Text Tracks (con't)

- `cue` pseudo-element to format the appearance of the cues appearing within a media clip

```
::cue {  
  styles  
}
```

- Styles for the `cue` pseudo-element are limited to background, color, font, opacity, outline, text-decoration, text-shadow, visibility, **and** white-space properties

Text Tracks (con't)

- Format specific cues or text strings within a cue using the following markup tags:
 - `<i></i>` for italicized text
 - `<b></b>` for bold-faced text
 - `<u></u>` for underlined text
 - `<span></span>` to mark spans of text
 - `<ruby></ruby>` to mark ruby text
 - `<rt></rt>` to mark ruby text

Text Tracks (con't)

- WebVTT supports tags that are not part of the HTML library
- `<c></c>` tag is used to mark text strings belonging to a particular class
`<c.classname></c>`
- `<v></v>` tag is used for captions that distinguish between one voice and another
`<v name></v>`

Text Tracks (con't)

The diagram shows a WEBVTT text track with three entries: Title, Subtitle, and Ending. Annotations highlight specific markup tags and their visual effects.

```
WEBVTT

Title
00:00.500 --> 00:04.000 line:5% align:middle
<c.Main>The Ceiling Dance</c>

Subtitle
00:04.500 --> 00:08.000 line:5% align:middle
from Royal Wedding (1951)

Ending
01:38.000 --> 01:44.000 line:80% position:95% align:end
See more videos at <i>Cinema Penguin</i>
```

Annotations:

- A green box labeled "markup tag for the Main class" with an arrow pointing to the `<c.Main>` tag in the Title line.
- A green box labeled "displays the website title in italics" with an arrow pointing to the `<i>` tag in the Ending line.



ANIMATIONS & TRANSITIONS

Transitions

- **Transition:** Change in an object's style from the initial state to the ending state, usually in response to an event initiated by the user or the browser
- It slows down the change from one color to another and provides intermediate styles
- To create transition, employ the following transition style:

`transition: property duration;`

where *property* is a property of the object that changes between the initial and end states and *duration* is the transition time in seconds or milliseconds

Transitions (con't)

- Varying speed of transition is defined using

`transition: property duration
timing-function;`

where *timing-function* is one of the following keywords:

- `ease`: (the default) Transition occurs more rapidly at the beginning and slows down near the end
- `ease-in`: Transition starts slowly and maintains a constant rate until the finish

Transitions (con't)

timing-function keywords (continued)

- `ease-out`: Transition starts at a constant rate and then slows down toward the finish
- `ease-in-out`: Transition starts slowly, reaches a constant rate, and then slows down toward the finish
- `linear`: Transition is applied at a constant rate throughout the duration

Transitions (con't)

- Timing function can be visualized as a graph
- It shows the progress of transition vs. duration
- The graphical representation of the timing function is the basis of another measure of transition timing using

`cubic-bezier(n, n, n, n)`

Where the *n* parameter values define the shape of the timing curve

Transitions (con't)

- Transition does not need to start immediately after the event that triggers it
- Start of the transition can be delayed by adding *delay* value to the following:

```
transition: property duration  
timing-function delay;
```

where *delay* is measured in seconds or milliseconds

Transitions (con't)

- The transition property can be added to slow down the transition from initial to end state
- Limitations of transition
 - It can only be run when a CSS property is being changed, such as during the hover event
 - It is run once and cannot be looped for repetition
 - Initial and end states of the transition can be defined but not the styles of intermediate states
- Animation is created to overcome the limitations

Transitions (con't)

initial state displays the
hypertext links in white
with a small text shadow

end state displays the
hypertext links in light
orange with a larger font
and a larger text shadow

```
/* Transition Styles */  
  
nav#topLinks a {  
    color: rgb(255, 255, 255);  
    font-size: 1em;  
    letter-spacing: 0em;  
    text-shadow: rgba(0, 0, 0, 1) 1px -1px 1px;  
}  
  
nav#topLinks a:hover {  
    color: rgb(255, 183, 25);  
    font-size: 3em;  
    letter-spacing: 0.1em;  
    text-shadow: rgba(0, 0, 0, 0.5) 15px -3px 8px;  
}
```

Animation

- **Key frame:** Sequence of changing images to create illusive movement for animation
- CSS replaces the concept of key frame images with key frame styles that are applied in rapid succession to a page object
- To define a sequence of key frames in CSS, apply the following `@keyframes` rule

```
@keyframes name {  
  keyframe1 {styles;}  
  keyframe2 {styles;}  
  ... }
```

Animation (con't)

where,

- `name` provides the name or title of the animated sequence
- `keyframe1`, `keyframe2`, and so on defines the progress of individual key frames
- `styles` are the styles applied within each key frame

Animation (con't)

- Key frames animation is applied to an object using `animation-name` and `animation-duration` properties

`animation-name: keyframes;`

`animation-duration: times;`

where *keyframes* is a comma-separated list of animations applied to the object using the names from the *@keyframes* rule and *times* are the lengths of each animation expressed in seconds or milliseconds

Animation (con't)

Property	Description
<code>animation-name = <i>keyframes</i></code>	Assigns the <i>keyframes</i> animation to the object
<code>animation-duration = <i>time</i></code>	Sets the length of the animation in seconds or milliseconds (default = 0s)
<code>animation-timing-function = ease ease-in ease-out ease-in-out linear cubic-bezier(<i>n,n,n,n</i>)</code>	Defines the default timing between key frames in the animation (default = ease)
<code>animation-delay = <i>time</i></code>	Sets the delay time in seconds and milliseconds before animation is started (default = 0s)
<code>animation-iteration-count = <i>value</i> infinite</code>	Specifies the number of times the animation is played, where <i>value</i> is an integer and infinite repeats the animation without stopping (default = 1)

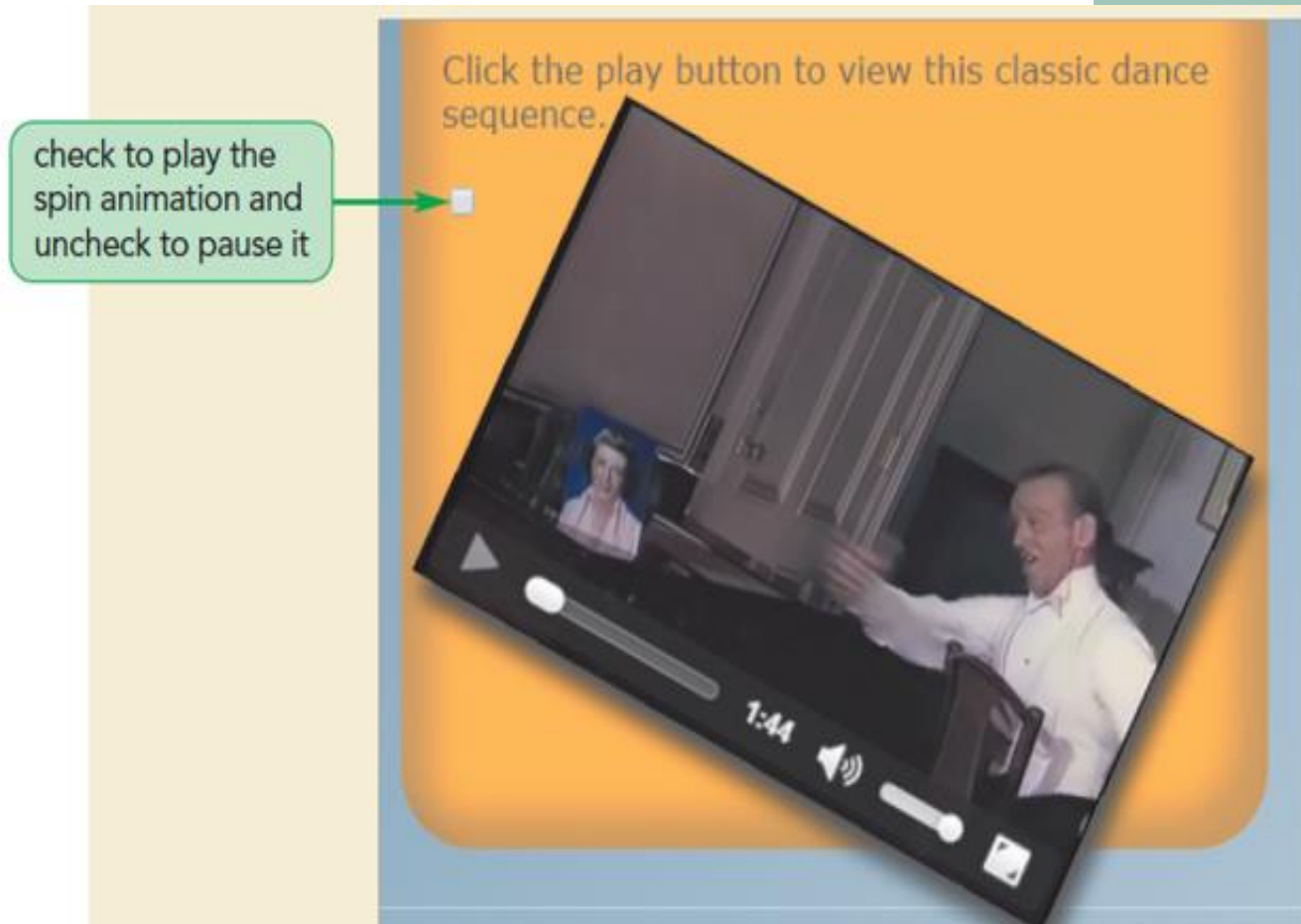
Animation (con't)

Property	Description
<code>animation-direction = normal reverse alternate alternate-reverse</code>	Defines the direction of the animation, where <code>normal</code> plays the animation as defined in the <code>@keyframes</code> rule, <code>reverse</code> reverses the order, <code>alternate</code> plays the animation in the normal direction followed by the reverse direction (for multiple iterations), and <code>alternate-reverse</code> plays the animation in reverse direction followed by normal direction (default= <code>normal</code>)
<code>animation-fill-mode = none backwards forwards both</code>	Defines what styles from the animation are applied to the object outside the time it is running, where <code>none</code> does not apply any styles, <code>backwards</code> applies the styles from the first key frame, <code>forwards</code> applies the styles from the last key frame, and <code>both</code> applies styles in both directions (default= <code>none</code>)
<code>animation-play-state = running paused</code>	Defines whether the animation is running or paused (default = <code>running</code>)

Animation (con't)

- Animation can have two states of operation
 - Play or pause
- Check box can be used to control animation
- Selecting the check box will play the animation
- Unselecting the check box will pause the animation

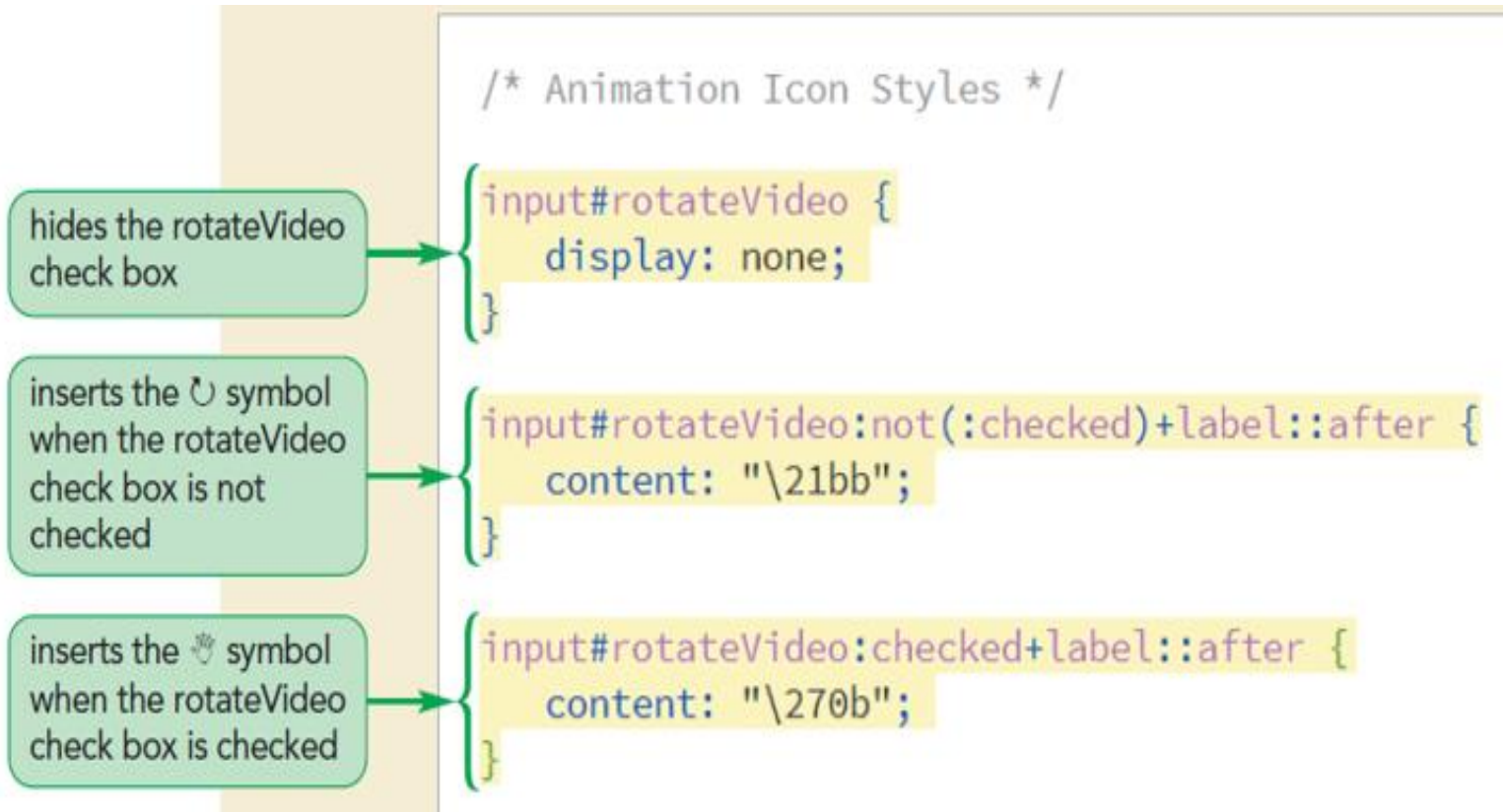
Animation (con't)



Animation (con't)

- Check box can be replaced with more attractive icons
- Display symbol ↺ to run the animation
 - Unicode value of \21bb
- Display symbol 🖐 to pause the animation
 - Unicode values of \270b

Animation (con't)





FLASH

Macromedia Flash

- Flash “movies” are interactive vector graphics and animation for web sites
- In Flash, you create a movie by drawing (or importing) artwork, arranging it on a “stage”, and animating it with a “timeline” (path of movement in time and space).
 - You can make the movie interactive by making it respond to events and to change in specified ways; sound can also be added

Flash (con't)

- Flash had become the technology of choice for creating rich, interactive, multimedia content for the web **before HTML5 was available**
- Three factors that had greatly contributed to the technology's popularity are:
 - The [.swf](#) format can compress both images and sounds
 - The format allows for the inclusion/use of vector formats, scaleable graphics with very small file sizes
 - The Flash player's streaming capabilities, which allow preloading images and sounds into the visitor's browser cache, and animation to stream while playing, rather than making the visitor wait for it to fully download
- **Flash is no longer supported by Adobe**

FLASH

- The most-used plug-in for video playback is Adobe Flash player embedded using the following `object` element:

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

FLASH (con't)

Name	Value(s)	Description
bgcolor	<i>color value</i>	Sets the background color of the player
flashvar	<i>text</i>	Contains text values that are passed to the player as variables to control the behavior and content of the movie
id	<i>text</i>	Identifies the movie so that it can be referenced
loop	true false	Plays the movie in a continuous loop
menu	true false	Displays a popup menu when a user right-clicks the player
name	<i>text</i>	Names the movie so that it can be referenced
play	true false	Starts the player when the page loads

FLASH (con't)

Name	Value(s)	Description
quality	low autolow autohigh medium high best	Sets the playback quality of the movie; low values favor playback speed over display quality; high values favor display quality over playback speed
scale	showall noborder exactfit	Defines how the movie clip is scaled within the defined space; a value of <code>showall</code> makes the entire clip visible in the specified area without distortion; a value of <code>noborder</code> scales the movie to fill the specified area without distortion but possibly with some cropping; a value of <code>exactfit</code> makes the entire movie visible in the specified area without trying to preserve the original aspect ratio
wmode	window opaque transparent	Sets the appearance of the player against the page background; a value of <code>window</code> causes the movie to play within its own window; a value of <code>opaque</code> hides everything behind the player; a value of <code>transparent</code> allows the page background to show through transparent colors in the player

Flash (con't)

- When one has built the movie, it can be “published” where by the movie file (*.swf) and corresponding HTML snippet is generated (browser needs Flash “player” installed)
- Movies can also be published in other formats such as animated gif's
- A trial version of Flash can be downloaded at:
 - www.macromedia.com/software/flash/trial/

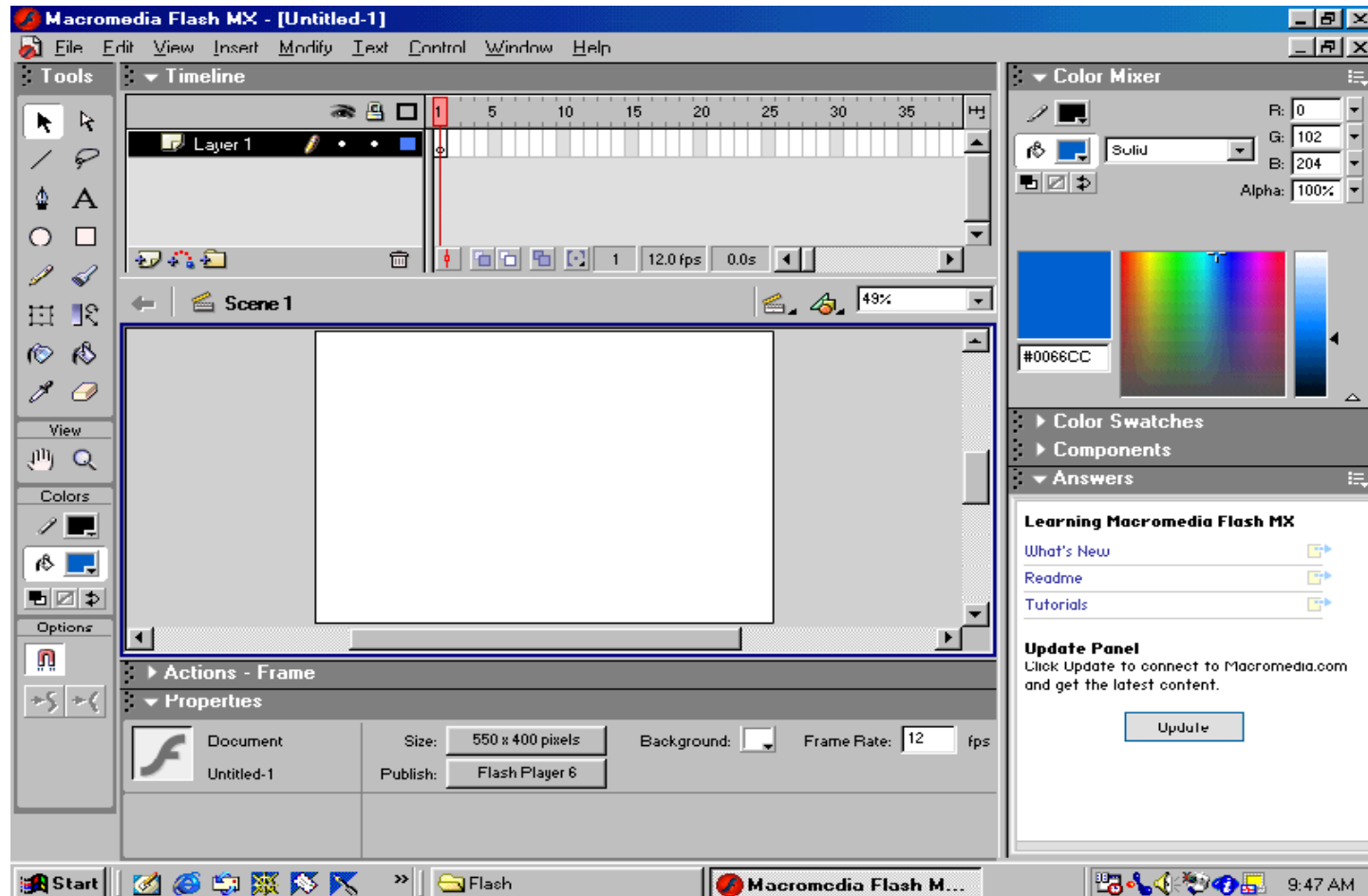
Flash Concepts

- Movie – a flash “document” (.fla) that can be published in a variety of formats; it is composed of frames along a “timeline”
 - Vector (.swf)
 - Gif (.gif)
 - Executable (.exe)
- Stage – the composing/drawing area showing the layer(s) of each frame
- Frame – a single moment in a timeline; a frame is composed of superimposed “layers”
- Layer – an x-y drawing surface

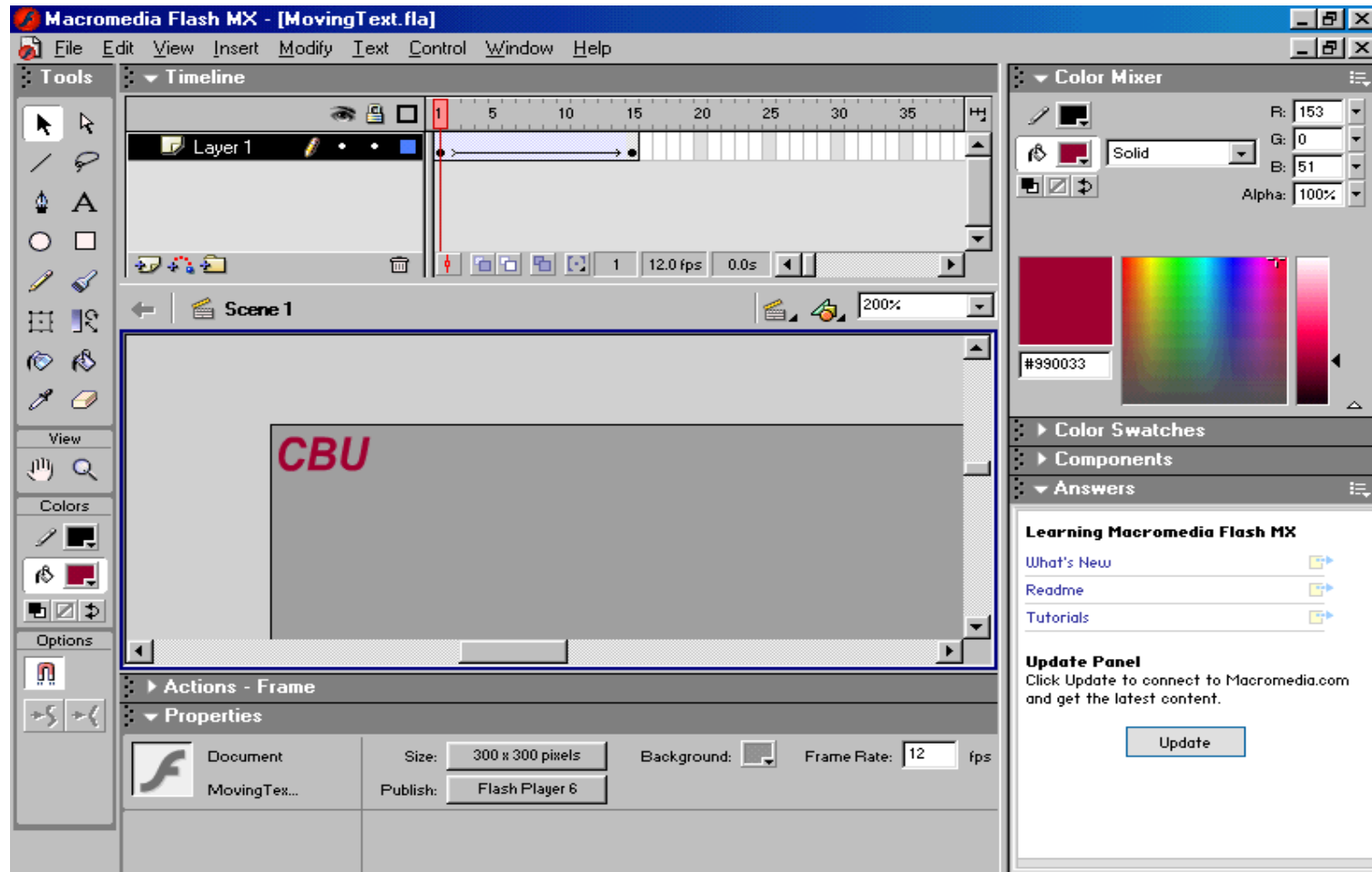
Key Flash Features

- Vector graphics – uses geometric formulas to draw curves that define images instead of bitmaps; reduces file sizes of images
- Tweening – the process of automatically creating the frames in between “key frames”

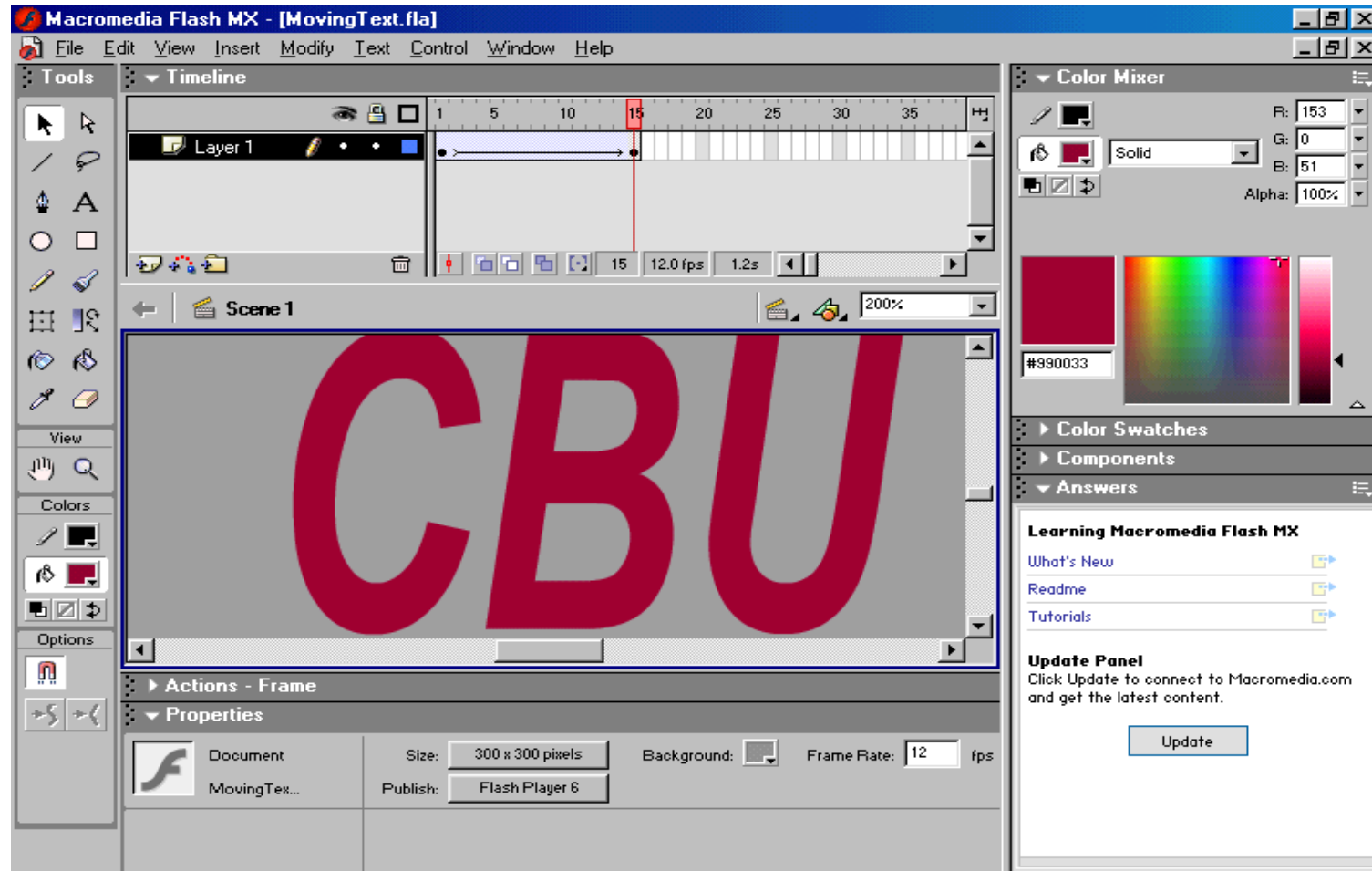
Flash MX Workspace



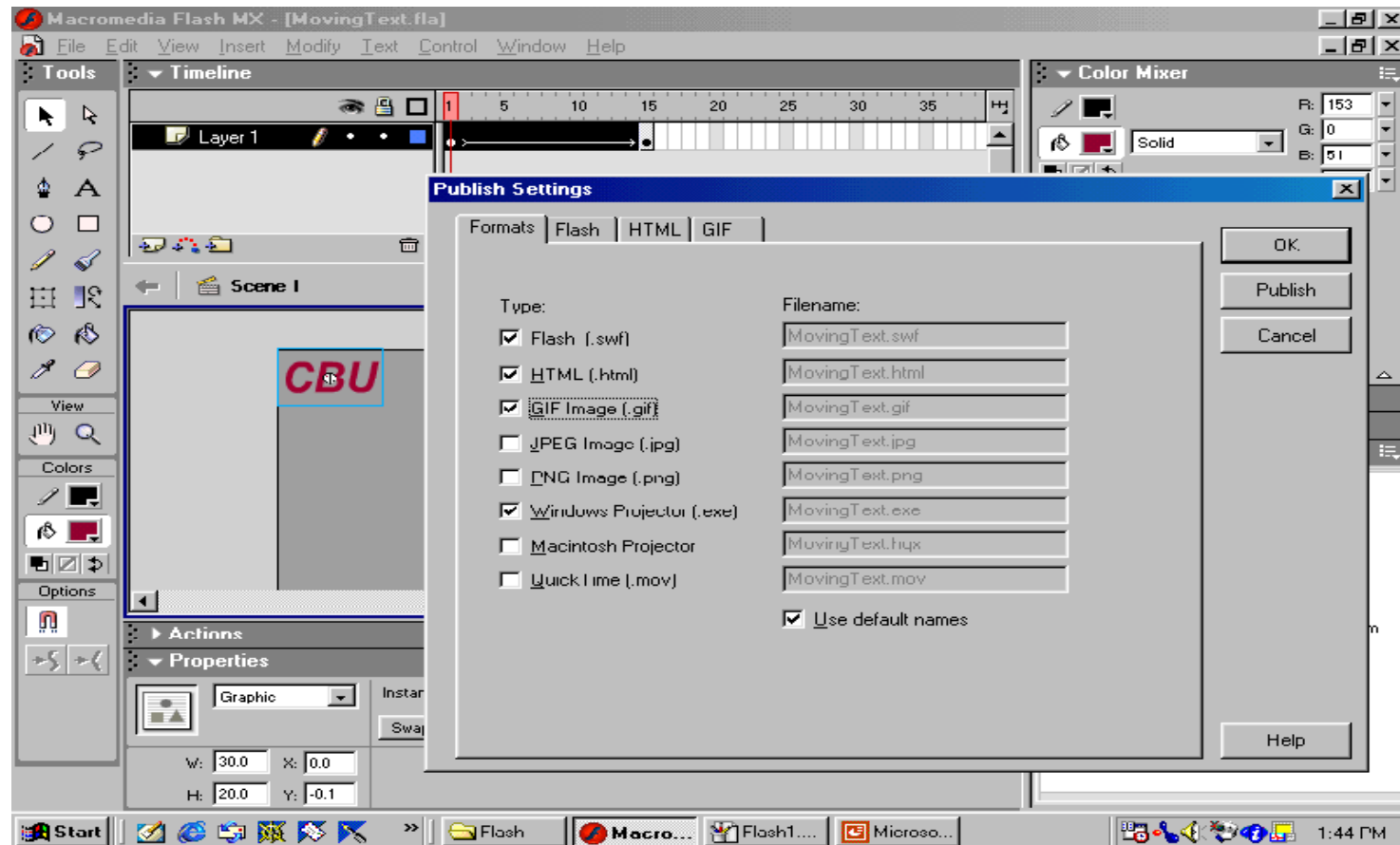
First Key Frame in a Scene



Last Key Frame in a Scene [with motion tween]



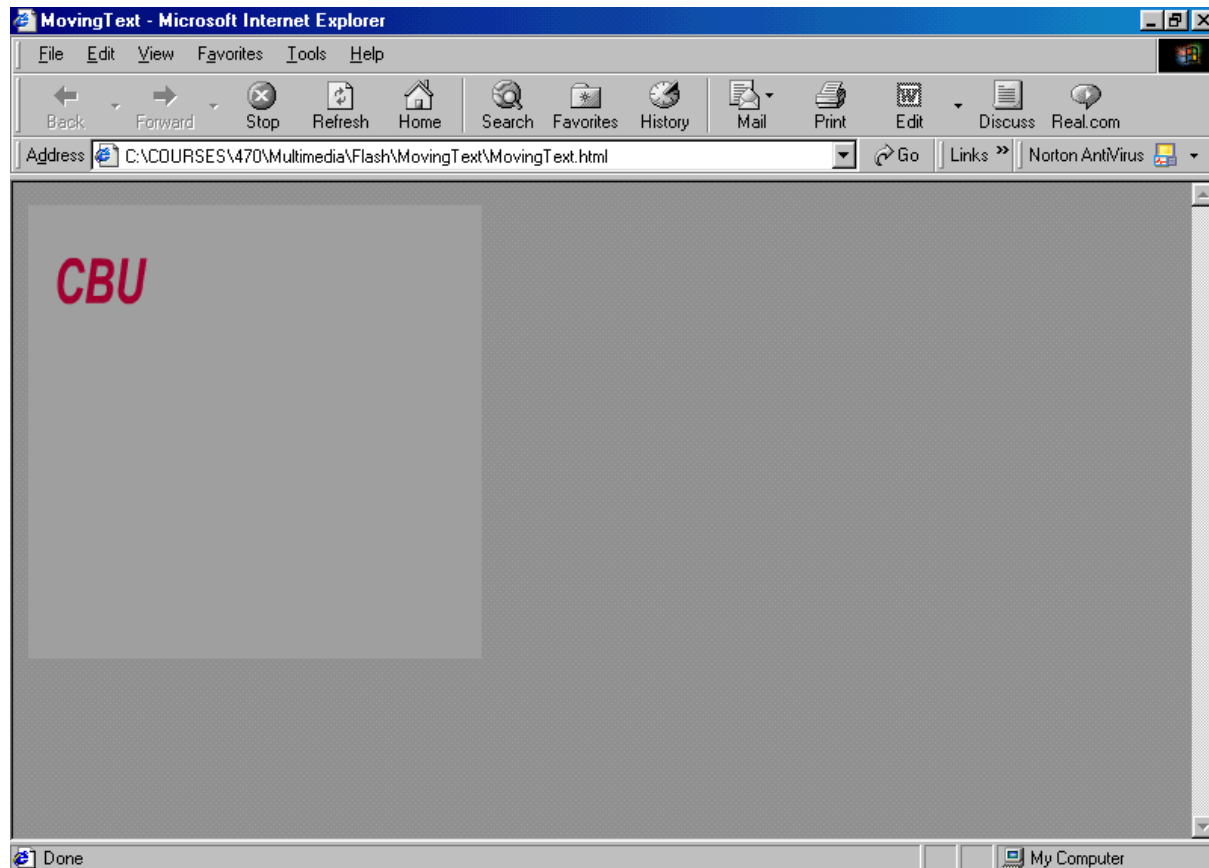
Publishing Format



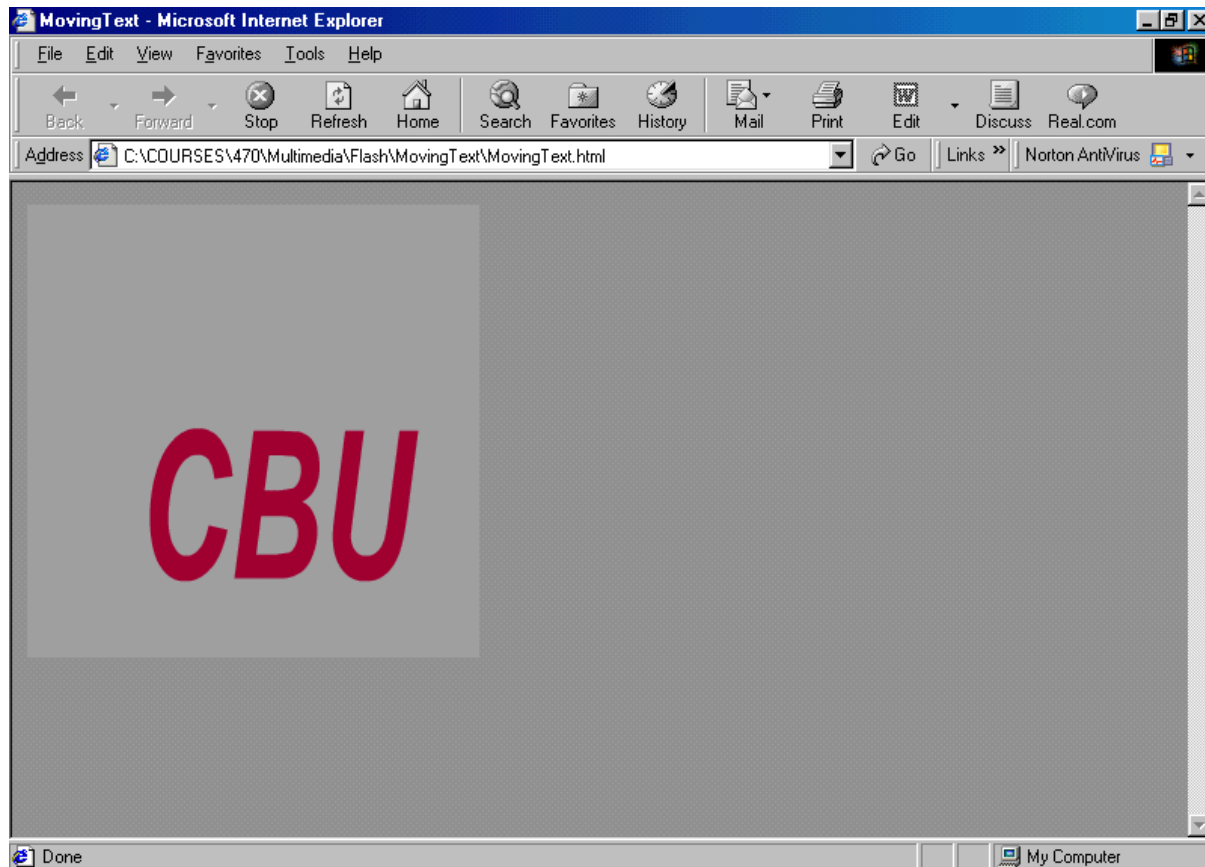
- <HTML>
- <HEAD>
- <meta http-equiv=Content-Type content="text/html; charset=ISO-8859-1">
- <TITLE>MovingText</TITLE>
- </HEAD>
- <BODY bgcolor="#999999">
- <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
- codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash
- .cab#version=6,0,0,0"
- WIDTH="300" HEIGHT="300" id="MovingText" ALIGN="">
- <PARAM NAME=movie VALUE="MovingText.swf">
- <PARAM NAME=quality VALUE=high>
- <PARAM NAME=bgcolor VALUE=#999999>
- <EMBED src="MovingText.swf" quality=high bgcolor=#999999
- WIDTH="300" HEIGHT="300" NAME="MovingText" ALIGN=""
- TYPE="application/x-shockwave-flash"
- PLUGINSOURCE="http://www.macromedia.com/go/getflashplayer">
- </EMBED>
- </OBJECT>
- </BODY>
- </HTML>

MovingText.html – works in
Both most browsers !!

Start of Animation

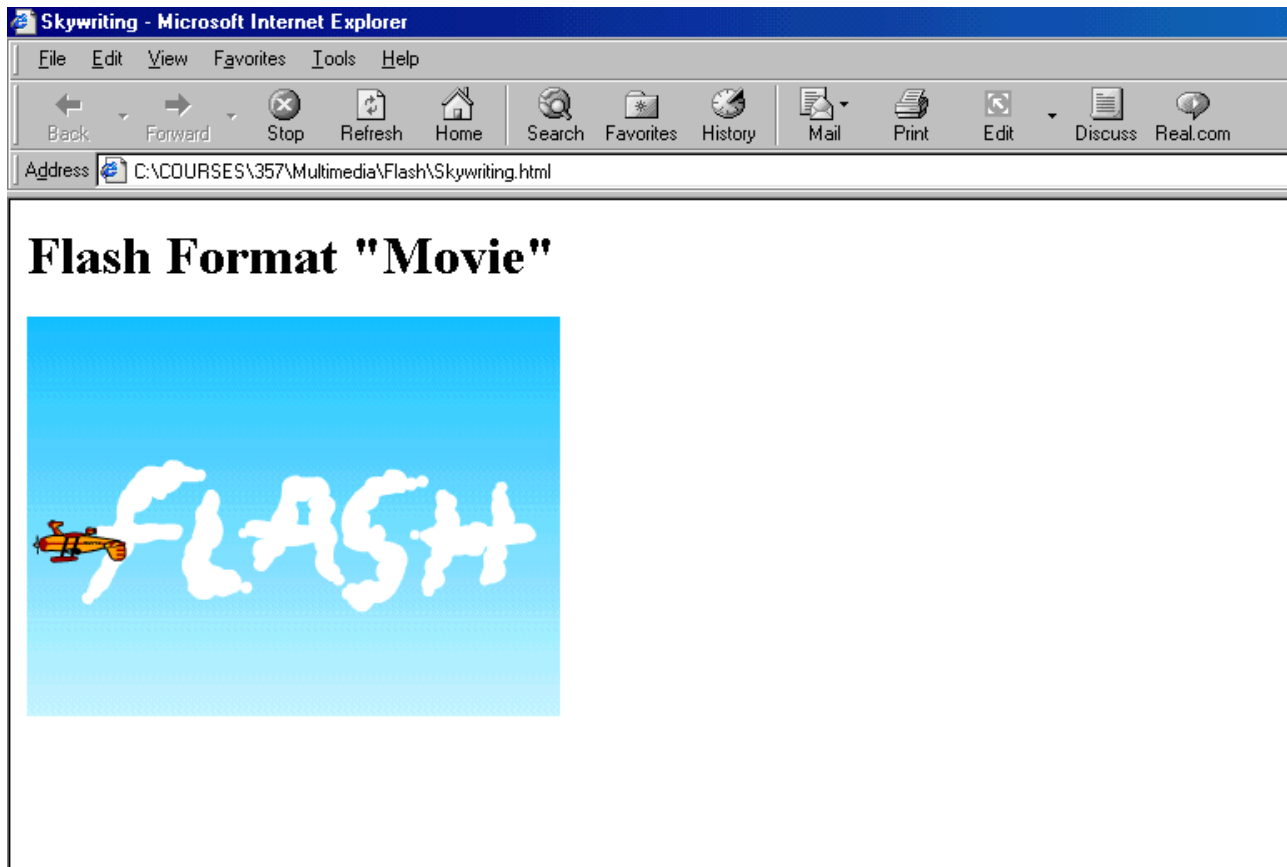


End of Animation



This Paints Quickly

[<http://facstaff.cbu.edu/dbrandon/Flash/Skywriting.html>]



HTML for Flash Movie

[Flash Product can generate HTML]

```
■ <HTML>
■ <HEAD>
■ <TITLE>Skywriting</TITLE>
■ </HEAD>
■ <BODY bgcolor="#FFFFFF">
■ <H1>Flash Format "Movie"</H1>
■ <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
■     codebase="http://active.macromedia.com/flash2/cabs/
■         swflash.cab#version=4,0,0,0"
■     ID=Skywriting WIDTH=320 HEIGHT=240>
■     <PARAM NAME=movie VALUE="Skywriting.swf">
■     <PARAM NAME=quality VALUE=high> <PARAM NAME=bgcolor VALUE=#FFFFFF>
■     <EMBED src="Skywriting.swf" quality=high bgcolor=#FFFFFF
■         WIDTH=320 HEIGHT=240 TYPE="application/x-shockwave-flash"
■         PLUGINSOURCE="http://www.macromedia.com/
■         shockwave/download/index.cgi?
■         Pl_Prod_Version=ShockwaveFlash">
■     </EMBED>
■ </OBJECT>
■ </BODY>
■ </HTML>
```

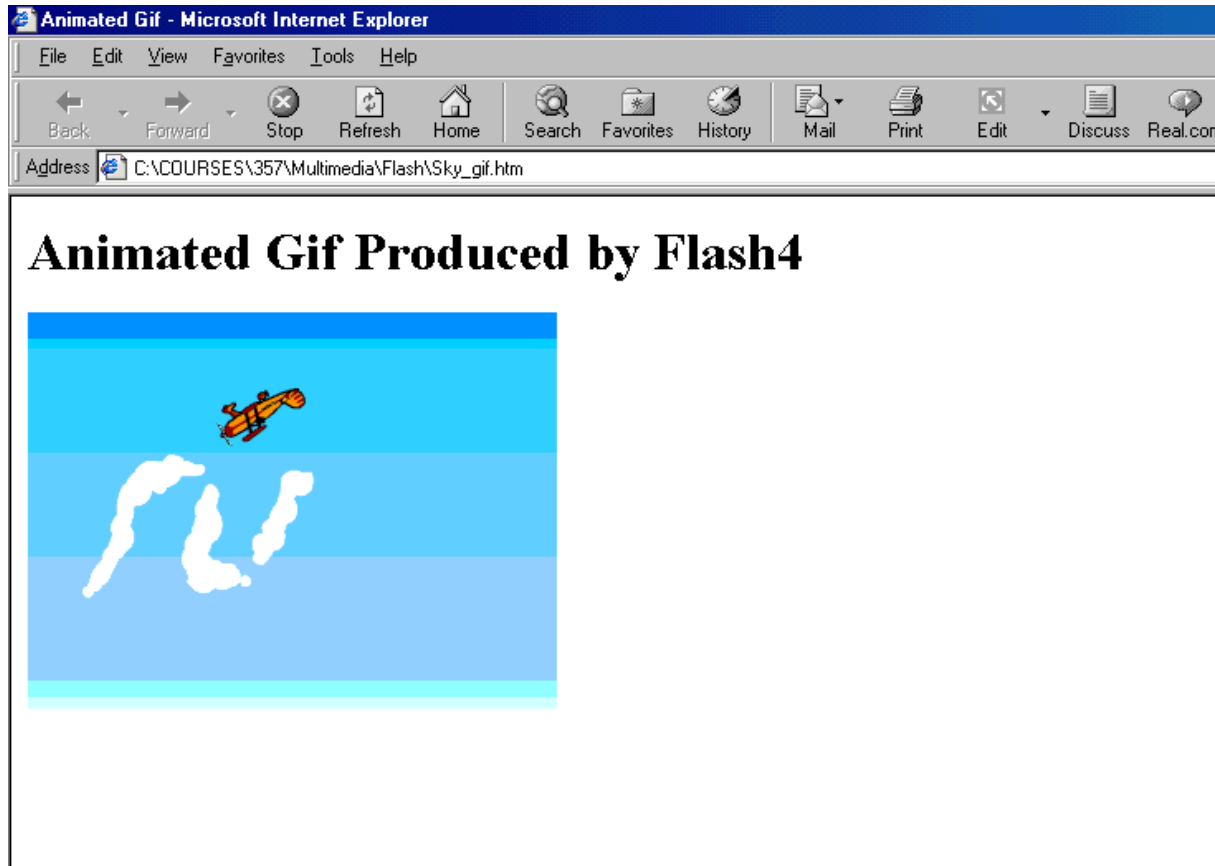
Also instructs browser
to download flash
player, if not loaded

HTML for Animated Gif Files

- `<HTML>`
- `<HEAD>`
- `<TITLE>Animated Gif</TITLE></HEAD>`
- `<BODY>`
- `<H1>Animated Gif Produced by
Flash4</H1>`
- `<IMG ALIGN=top Src="Skywriting.gif">`
- `</BODY>`
- `</HTML>`

This Paints Slowly

[http://facstaff.cbu.edu/dbrandon/Flash/Sky_gif.htm]



HTML5 Multimedia

- Using proprietary media such as Flash may limit your market for E-Commerce type applications
- If you want a wide (and world wide) audience, you need to write JS to determine the browser and version; if the user has an old version of a browser he probably has an older PC that will not support modern media (so serve up a plain HTML page to that type of user)
- HTML5 now has its own drawing and multimedia capability, but not yet fully supported for all browsers and media formats

Macromedia FLEX

[HTML5 will likely dominate this market in time]

ProductsSolutionsLearningHelpDownloadsCompanyStore

My AdobeMy ordersMy cartSign in

Flex

OverviewTech specsFAQShowcaseExtend

One codebase,
multiple devices

Build high-performance applications in less time for iOS,
Android, BlackBerry Tablet OS, and the desktop.

Download

What is Flex?

Flex is a powerful, open source application framework that allows you to easily build mobile applications for iOS, Android™, and BlackBerry® Tablet OS devices, as well as traditional applications for browser and desktop using the same programming model, tool, and codebase.

Enterprise-class programming model

Use constructs like strong typing, inheritance, and interfaces to program more efficiently. Extensive mobile and traditional components help speed development. Flex applications can access device capabilities such as GPS, accelerometer, camera, and local database.

Cross-platform and native experience

Create applications that run consistently across Android, BlackBerry Tablet OS, and iOS devices, as well as inside the browser and on traditional desktop computers. Although cross platform, with Flex you get an uncompromised native experience on each platform.

End-to-end tooling

Build Flex applications more easily using Adobe® Flash® Builder® software, an enterprise-class Eclipse™ based IDE. Productivity features include on-device debugging and mobile simulators for testing across screen sizes and resolutions.

Server integration

Integrate with all major back ends including Java™, Spring, Hibernate, PHP, Ruby, .NET, Adobe ColdFusion®, and SAP using industry standards such as REST, SOAP, JSON, JMS, and AMF.

Getting started

Build Flex and ActionScript® applications with Adobe Flash Builder software.

Learn more >

Flash Builder 4.5 Premium

Rapidly develop high-performing mobile, web, and desktop applications

Upgrade from US \$49.00

BuyTry

Includes complimentary Flash Builder 4.6 update

Do even more with...

CSS 3 Web Premium

Redefine the extraordinary in web design and development

Upgrade from US \$399.00

BuyTry

or call 800-585-0774

Add Lightroom for 30% off. Hurry — ends soon.

Are you an education customer?

Using PHP and Flex?
LEARN ABOUT NEW
FLASH BUILDER 4.5 FOR PHP

Download Adobe Flex SDK

SWiSH



If you want to create wild and groovy Flash™ animations without using Flash™, then SWiSH is the tool for you! SWiSH is so easy to use, you'll be producing complex animations with text, images, graphics and sound in no time. SWiSH has over 150 built-in effects like Explode, Vortex, 3D Spin, Snake and many more. SWiSH has tools for creating lines, rectangles, ellipses, Bezier curves, motion paths, sprites and rollover buttons, all in an easy-to-use interface.

If you are over battling with complicated, expensive programs, use SWiSH. SWiSH takes the drama out of dramatic effects.

SWiSH exports the SWF file format used by Macromedia Flash™, so your animation will play on any machine that has the Flash™ Player installed. According to a [recent survey by NPD Research](#) that's over 96% of web users, or 334 million people!

SWiSH

- The animation created is called a [Movie](#). Within each Movie is a collection of [Scenes](#). Each Scene has a Timeline consisting of multiple [Frames](#).
- During the [Timeline](#) of each Scene, you place [objects](#) (e.g. text, images, etc.) to which you apply [Effects](#). These Effects will start and stop at particular Frames and can be controlled by inserting [Actions](#) and [Events](#).
 - In Scenes, Events occur when the Movie reaches a certain Frame. For objects, Events occur when you interact with an object using the mouse, such as rolling the mouse cursor over the object or clicking on it.
 - An Event can trigger more than one Action. For example, when the mouse rolls over an object, the Movie can be stopped (with a Stop Action) and the browser can be told to load an URL into another Frame (with the Goto URL Action).
 - Actions are operations that are triggered by Events. Actions can alter the playing of the Movie, start or stop sounds, load other Movies or web pages, or communicate with the host browser or player.