

Developing Mobile Websites

Video

Lesson 1, Activity 2: Optimizing the Video Experience for Mobile
Optimizing for Mobile

As the percentage of traffic to our Web sites from users on mobile devices grows, so too, of course, does the percentage of users viewing video from mobile. Before we delve into file formats, HTML5 code, or JavaScript, let's spend a few minutes on content creation: how should the video footage you shoot be different if your intended audience will be watching from a 3" screen?

First, keep in mind the constraints of the device. The small screen size on mobile devices means on-screen text or captions will be difficult to read if too small; be sure to make text big. Ditto with in-video links: it's often better to present hyperlinks on the Web page, above or under the video player, rather than in the video itself. A small screen also means wide, far-away shots will be difficult to view: zoom in on the interesting stuff.

Similarly, visitors on mobile devices frequently will hear the audio from your video on small, tinny speakers: be sure to make the audio as clear as possible.

Most importantly, test your video: does it look good on your smartphone? Of course, testing the video for optimal viewing is a best practice regardless of the intended viewing device - but testing on a phone should be a given.

Embedding from YouTube or Vimeo

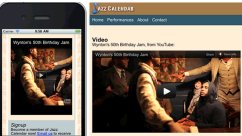
YouTube, Vimeo, and other video sites offer mobile-friendly embedding of videos. Conveniently, the same embed code works, chameleon-like, on desktops and mobile: where a user viewing our page on a PC or Mac desktop might see the Flash version of the video, a user viewing the page on an Android or iPhone would see a non-Flash version.

As a strategy, publishing your videos to YouTube, Vimeo, or some other video site makes a lot of sense: you increase the visibility of the videos (since users can find it via the video site itself, not just on your site); YouTube or Vimeo is hosting your video for you; and embedding solves many technical, how-do-I-do-this-for-mobile questions.

As a quick demonstration, we'll select a YouTube video appropriate for our *Jazz Calendar* site; this [Wynton Marsalis video](#) looks good. We'll choose "Embed" under the "Share" option:



and paste the iframe code into [Videos/Demos/youtube.html](#).



Note that we scale the iframe element, in [Videos/Demos/youtube.css](#), to fit the containing element:

```
#main #maincontent iframe {
  max-width:100%;
  margin-bottom:4%;
}
```

Hosting Your Own Videos

Housing your videos on YouTube or Vimeo isn't always a perfect option. Worries over intellectual property rights (others can embed my videos), loss of traffic (visitors view my videos on YouTube instead of my site), professionalism (worries that visitors might think my site less professional if videos are embedded), and other concerns might dictate that you host your videos on your own site.

The HTML5 video tag, along with appropriate choices for video file format, makes relatively easy the process of presenting videos to be playable regardless of the viewing device.

First, let's look at the video tag:



The width and height attributes set the width and height, of course; we'll also set the CSS max-width of the video element to ensure a responsive handling of width. The presence of the controls attribute dictates that the player will show controls ("play", "pause", etc.) The poster attribute is the image to show on the player before video playing starts.

The video tag encloses a series of source tags, which specify alternative video files from which the browser may choose, depending on the media types or codecs the browser supports.

In the example above, we offer browsers three options to pick from: .mp4, .webm, and .ogg versions of our video. Note that we're using open-source videos from the [Video for Everybody](#) site - we kindly appreciate their sharing.

Last, the content inside the video tag but outside the source tags is what users will see if their browser does not support the video tag.

Video Formats

Specifications about video file formats is a bit beyond the scope of this course. What we really need is a list of best formats to use and some information on how to convert to those formats.

Flash video, often the format in which embedded video is presented for desktop browsers, generally won't work on mobile devices. Instead, we'll concentrate on three formats:

| Video File Formats | | |
|--------------------|----------------|---|
| Format | File Extension | Notes |
| MPEG 4 | .mp4 | Based on Apple older QuickTime format; supported natively in all browsers |
| Ogg | .ogg, .ogv | Open source; unencumbered by patents; supported natively by Firefox >3.5, Opera >10.50, Chrome >3.0 |
| WebM | .webm | New (2010) format; supported natively by Chrome, Firefox, and Opera |

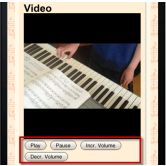
Thus, to publish video - specifically, video able to be viewed on different desktop and smartphone browsers - we:

1. Take our original video file (the raw file from the camera).
2. Convert it to each of the file formats above.
3. Publish with the video tag.

The user sees a video player with their browser - whether it be Safari on an iPhone, Internet Explorer on a Windows PC desktop, or anything else - picking the video format that works best.

Controlling the Video through JavaScript

Open [Videos/Demos/video.html](#), in a desktop browser and also, if possible, on a smartphone:



Code Sample:

```
Video/Demos/video.html

---- CODE OMITTED ----
Video

Your browser does not support the video tag

Play
Pause
Incr. Volume
Decr. Volume
---- CODE OMITTED ----
```

Note that the video is presented using the same code we reviewed above, with the "big_buck_bunny" video (in various formats) coming from an external website - of course, these might be video files hosted on your own server.

New to this example are four buttons allowing the user to play or pause the video and to increase or decrease the volume. We accomplish this with a mix of HTML and JavaScript. Each button tag has an onclick handler; when clicked, the play() or pause() function is called or the volume property incremented/decremented, depending on which button was clicked. We find the video element upon which to act by referencing its id (vid) through document.getElementById. Note that, as smartphones like the iPhone will open the video in a new video-player window, these buttons won't be terribly useful on an iPhone.

Converting Video Formats

There exist a variety of free, open-source, and/or relatively inexpensive options for converting raw, from-the-camera video files into HTML5-appropriate file formats:

- [Miro Video Converter](#), is a free, downloadable video converter for PC and Mac, converting to mp4, ogg, and webm formats.
- [video online convert.com](#), offers free, online conversion from and to a variety of formats;
- [Fireform](#), is a free Firefox add-on, offering conversion to ogg and webm formats.
- [HandBrake](#) is an open-source, GPL-licensed, multiplatform, multithreaded video transcoder, available for MacOS X, Linux and Windows.

A note: if taking video with an iPhone, be careful about trimming the video right from iPhoto: the modified version of the .mov file produced by iPhoto is a file that Miro was unable to edit - be sure to use the original raw video file

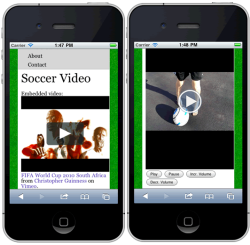
Further Reading

[Dive Into HTML 5](#), is a great resource for more information on HTML5 in general; [Videos on the Web](#) is, in particular, an excellent source of information on creating, saving, and publishing video in HTML5 and, thus, for mobile viewing.

Lesson 1, Activity 4: Soccer Video

Duration: 20 to 30 minutes.

In this exercise, you will add several videos to an interior page on the *Pickup Soccer* site. The following screen shots shows the page on an iPhone - the screen capture on the left is the top of the page and the screen capture on the right is the bottom of the page:



1. Open [Video/Exercises/video/index.html](#) in your file editor.
2. Find and embed a video from YouTube or Vimeo; test to make sure the video works on both desktop and mobile browsers; be sure to address flexible width for the embedded element, especially on mobile.
3. Use the video tag to display the included video (the "Big Buck Bunny" video) as a video player underneath the embedded video; use [Video/Exercises/video/images/poster.png](#) as the poster image for the video.
4. Add buttons below the video tag to allow the user to play, pause, and control the volume for the video.
5. Be sure to address the width of the video element, especially on mobile.
6. Test your work in both desktop and mobile browsers.

Challenge

If your current setup allows and you have access to raw video footage, convert your own video file to .mp4, .ogv, and .webm formats; use these files in the video tag for your solution. Use any of the options listed above - Handbrake, Miro, Firefogg, video.online-convert.com - or another video-conversion method. Be sure to test your video on a variety of devices.

Solution:

[Video/Solutions/video/index.html](#)

```

---- CODE OMITTED ----
<h1>Soccer Video</h1>
<p>Embedded video:</p>
<iframe src="http://player.vimeo.com/video/11828030?title=0&amp;byline=0&amp;portrait=0" width="400" height="225" frameborder="0" webkitAllowFullScreen mozallowfullscreen allowFullScreen"></iframe><p><a href="http://
<p>Hosted video:</p>
<video width="400" height="225" controls poster="images/poster.png" id="vid">
  <source type="video/mp4" src="http://clips.vorwaerts-gmbh.de/big_buck_bunny.mp4"></source>
  <source type="video/webm" src="http://clips.vorwaerts-gmbh.de/big_buck_bunny.webm"></source>
  <source type="video/ogg" src="http://clips.vorwaerts-gmbh.de/big_buck_bunny.ogv"></source>
</video>
<p>Your browser does not support the video tag</p>
<div>
  <button onclick="document.getElementById('vid').play()">Play</button>
  <button onclick="document.getElementById('vid').pause()">Pause</button>
  <button onclick="document.getElementById('vid').volume+=0.1">Incr. Volume</button>
  <button onclick="document.getElementById('vid').volume-=0.1">Decr. Volume</button>
</div>
---- CODE OMITTED ----

```

We choose a FIFA 2010 World Cup video from Vimeo to embed, pasting the Vimeo-supplied `iframe` code into our page; we set the `iframe` element to have `max-width: 100%` in [Video/Solutions/video/index.html](#).

We use the `video` tag to present the video, supplying three different versions (.mp4, .ogv, and .webm) of the same video clip to work on a wide variety of browsers. We use `button` tags, with an `onclick` handler, to allow the user to control the video on desktop browsers.