11 Web Multimedia and Interactivity

Chapter Objectives

In this chapter, you will learn how to . . .

- Describe the purpose of plug-ins, helper applications, media containers, and codecs
- Describe the types of multimedia files used on the Web
- Configure hyperlinks to multimedia files
- Configure audio and video on a web page with HTML5 elements
- Configure a Flash animation on a web page
- Configure a Java applet on a web page
- Create an interactive drop down navigation menu with CSS
- Configure the CSS3 transform and transition properties
- Describe features and common uses of JavaScript
- Describe the purpose of HTML5 APIs such as geolocation, web storage, offline web applications, and canvas
- Describe features and common uses of Ajax
- Describe features and common uses of jQuery
- Locate Flash, Java applets, JavaScript, Ajax, and jQuery resources on the Web

Video and sounds on your web pages can make them more interesting and informative. In this chapter, you'll work with multimedia and interactive elements on web pages. Methods to add audio, video, and Flash are introduced. Sources of these media types, the HTML code required to make the media available on a web page, and suggested uses of the media are discussed.

You began to work with **interactivity** in Chapter 6 when you used CSS pseudo-classes to respond to mouse movements over hyperlinks. You'll expand your CSS skill set as you configure an interactive drop down navigation menu and explore CSS3 transition and transform properties. Adding the right touch of interactivity to a web page can make it engaging and compelling for your visitors.

Technologies commonly used to add interactivity to web pages include Flash, Java applets, JavaScript, Ajax, and jQuery. This chapter introduces you to these techniques. Each of these topics is explored more fully in other books; each technology could be the sole subject of an entire book or college course. As you read this chapter and try the examples, concentrate on learning the features and capabilities of each technology, rather than trying to master the details.

11.1 Plug-Ins, Containers, and Codecs

Web browsers are designed to display web pages and GIF, JPG, and PNG images, among others. When the **media** is not one of these types, the browser searches for a **plug-in** or **helper application** that is configured to display the file type. If it cannot find a plug-in or helper application (which runs in a separate window from the browser) on the visitor's computer, the web browser offers the visitor the option of saving the file to their computer. Several commonly used plug-ins are listed as follows:

- **Adobe Flash Player (http://www.adobe.com/products/flashplayer).** The Flash Player displays SWF files. These can contain audio, video, and animation, along with interactivity.
- **Adobe Shockwave Player (http://www.adobe.com/products/shockwaveplayer).** The Shockwave Player displays high-performance multimedia created using the Adobe Director application.
- **Adobe Reader (https://get.adobe.com/reader).** Adobe Reader is commonly used to display information stored in PDF format, such as printable brochures, documents, and white papers.
- **Java Runtime Environment (http://www.java.com/en/download/manual.jsp).** The Java Runtime Environment (JRE) is used to run applications and applets utilizing Java technology.
- **Windows Media Player (http://windows.microsoft.com/en-us/windows/download-windows-media-player).** The Windows Media Player plug-in plays streaming audio, video, animation, and multimedia presentations on the Web.
- **Apple QuickTime (http://www.apple.com/quicktime/download).** The Apple QuickTime plug-in displays QuickTime animation, music, audio, and video directly within the web page.

The plug-ins and helper applications listed previously have been used on the Web for many years. What is new about HTML5 audio and video is that it is native to the browser; no plug-ins are needed. When working with native HTML5 audio and video, you need to be aware of the **container** (which is designated by the file extension) and the **codec** (which is the algorithm used to compress the media). There is no single codec that is supported by popular browsers. For example, the H.264 codec requires licensing fees and is not supported by the

Firefox and Opera web browsers, which support royalty-free Vorbis and Thora codecs. See http://www.jwplayer.com/html5/ for more information of browser support of HTML5 video.

Explore Table 11.1 and Table 11.2, which list common media file extensions, the container file type, and a description with codec information (if applicable for HTML5).

Table 11.1 Common audio file types

Extension	Container	Description
.wav	Wave	Created by Microsoft; standard on the PC platform; also supported on the Mac platform.
.aiff and .aif	Audio Interchange	Popular audio file format on the Mac platform; also supported on the PC platform.
.mid	Musical Instrument Digital Interface (MIDI)	Contains instructions to recreate a musical sound rather than a digital recording of the sound itself; a limited number of types of sounds can be reproduced.
.au	Sun UNIX Sound File	Older type of sound file that generally has poorer sound quality than the newer audio file formats.
.mp3	MPEG-1 Audio Layer-3	Popular for music files because of the MP3 codec, which supports two channels and advanced compression.
.ogg	OGG	Open-source audio file format (see http://www.vorbis.com) that uses the Vorbis codec.
.m4a	MPEG-4 Audio	Audio-only MPEG-4 format that uses the Advanced Audio Coding (AAC) codec; supported by QuickTime, iTunes, and mobile devices such as the iPod and iPad.

Table 11.2 Common video file types

Extension	Container	Description
.mov	QuickTime	Created by Apple and initially used on the Mac platform, it is also supported by Windows.
.avi	Audio Video Interleaved	Microsoft's original standard video format for the PC platform.
.flv	Flash Video	Flash-compatible video file container; supports the H.264 codec.
.wmv	Windows Media Video	Streaming video technology developed by Microsoft; the Windows Media Player supports this file format.
.mpg	MPEG	Developed under the sponsorship of the Moving Picture Experts Group (MPEG) (http://www.chiariglione.org/mpeg); supported on both Windows and Mac platforms.
.m4v and .mp4	MPEG-4	MPEG-4 (MP4) codec and H.264 codec; played by QuickTime, iTunes, and mobile devices such as the iPod and iPad.
.3gp	3GPP Multimedia	H.264 codec; a standard for delivery of multimedia over third-generation, high-speed wireless networks.
.ogv or .ogg	OGG	Open-source video file format (see http://www.theora.org) that uses the Theora codec.
.webm	WebM	Open media file format (see http://www.webmproject.org) sponsored by Google; uses the VP8 video codec and Vorbis audio codec.

11.2 Getting Started with Audio and Video

As you read this chapter, you'll explore different ways to provide audio and video for your website visitors, including providing a hyperlink, the embed element, and the new HTML5 audio and video elements. We'll get started with the easiest method, which is coding a hyperlink.

Provide a Hyperlink

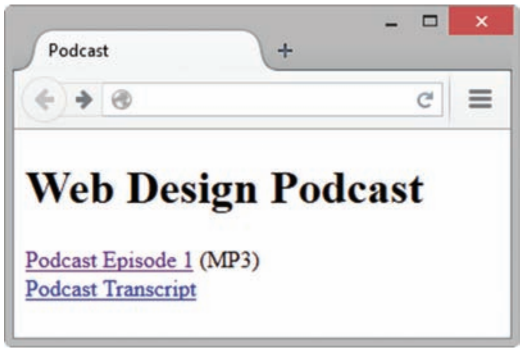
The easiest way to give your website visitors access to an audio or a video file is to create a simple hyperlink to the file. For example, the code to hyperlink to a sound file named WDFpodcast.mp3 is

```
<a href="WDFpodcast.mp3">Podcast Episode 1</a> (MP3)
```

When your website visitor clicks on the hyperlink, the plug-in for MP3 files that is installed on the computer (such as QuickTime) typically will display embedded in a new browser window or tab. Your web page visitor can then use the plug-in to play the sound. If your website visitor right-clicks on the hyperlink, the media file can be downloaded and saved.

Hands-On Practice 11.1

In this Hands-On Practice, you will create a web page similar to Figure 11.1 that contains an h1 tag and a hyperlink to an MP3 file. The web page will also provide a hyperlink to a text transcript of that file to provide for accessibility. It's useful to your web page visitors to also indicate the type of file (such as an MP3) and, optionally, the size of the file to be accessed.



Web Design Podcast

Podcast Episode 1 (MP3)
Podcast Transcript

Figure 11.1 The default MP3 player will launch in the browser when the visitor clicks on Podcast Episode 1

Copy the podcast.mp3 and podcast.txt files from the chapter11/starters folder in the student files and save them to a folder named podcast. Use the chapter2/template.html file as a starting point and create a web page with the heading "Web Design Podcast", a hyperlink to the MP3 file, and a hyperlink to the text transcript. Save your web page as podcast2.html and display it in a browser. Test your web page in different browsers, using different versions. When you click on the MP3 hyperlink, an audio player (whichever player or plug-in is configured for the browser) will launch to play the file. When you click on the hyperlink for the text transcript, the text will display in the browser. Compare your work to chapter11/11.1/index.html in the student files.

FAQ **How can I make a podcast?**

Podcasts are audio files on the web that may take the format of an audio blog, radio show, or interview. There are three steps in publishing a podcast:

1. **Record the podcast.** The Windows and Mac operating systems contain audio recording utilities. Apple's Quicktime Pro (available for both Windows and Mac) is a low-cost application that can be used to record audio. If you are using a Mac, another option is Apple's GarageBand, which is a pre-installed music application that offers a range of options for recording and editing audio. Audacity is a free cross-platform digital audio editor (available at http://sourceforge.net/projects/audacity for both Windows and Mac). You can use Audacity to record your voice for a podcast and mix in music loops to add interest. Once the WAV file is created, the LAME encoder (http://lame.sourceforge.net) or a similar application can be used to convert to an MP3 format.
2. **Upload the podcast.** Upload the MP3 to your website. If your web host does not permit MP3 files, an alternative is to upload to a site that accepts audio files at no cost such as Internet Archive (http://www.archive.org) or Ourmedia (http://ourmedia.org).
3. **Make the podcast available.** The most straightforward method is to code a hyperlink to the audio file. The hyperlink allows website visitors to access the podcast MP3 file, but does not make the podcast available for subscription. If you would like to provide a way for your visitors to subscribe to your podcasts, you will need to create an RSS feed. An **RSS feed** for a podcast is an XML file that lists information about your podcast. With a bit of patience, you can code your own RSS feed using a text editor (see Stephen Downes's website at http://www.downes.ca/cgi-bin/page.cgi?post=56 or Robin Good's website at http://www.masternewmedia.org/news/2006/03/09/how_to_create_a_rss.htm). However, a number of websites, such as FeedBurner (http://feedburner.google.com), and HostMyRss (http://hostmyrss.com/), provide a service that generates and hosts the RSS feed for you. After the RSS feed is uploaded to the Web (either your own or the RSS feed generator's site), you can code a link to the RSS file. Apple provides instructions for submitting your podcast to iTunes at http://apple.com/itunes/podcasts/specs.html. Web visitors using software such as Apple's iTunes or a free RSS feed reader website such as Feedreader (http:// feedreader.com) can locate and automatically download your podcast.

Working with Multimedia on the Web

More About Audio Files

Focus on Ethics

There are a number of ways that you can obtain audio files. You can record your own sounds, download sounds or music from a free site, record music from a CD, or purchase a DVD of sounds. There are some ethical issues related to using sounds and music created by others. You may only publish sounds or music that you have created yourself or for which you have obtained the rights (sometimes called a license) to publish. When you purchase a CD or DVD, you have not purchased the rights for publishing to the Web. Contact the owner of the copyright to request permission to use the music.

There are many sources of audio files on the Web. Some offer free files, such as Loopasonic (http://www.loopasonic.com) and FreeAudioClips.com (http://www.freeaudioclips.com).

Others, like SoundRangers (http://www.soundrangers.com), may offer one or two free sounds, but ultimately they are in the business of selling soundtracks and CDs. An interesting resource for free sound is the Flash Kit site (http://www.flashkit.com); click on the Sound Loops link. While this site is intended for Adobe Flash developers, the sound files can be used without Flash.

Audio files can be quite large and it is important to be aware of the amount of time required to download them for play. If you decide to use an audio file on a web page, make it as brief as possible. If you are recording your own audio files, be aware that the sampling rate and bit depth will affect the file size. A **sampling rate** is a value related to the number of digital sound samples taken per second when the sound is recorded. It is measured in kilohertz (kHz). Common sampling rates vary from 8 kHz (AM radio quality sound or sound effects) to 44.1 kHz (music CD quality sound). As you would expect, a sound recorded at 44.1 kHz has a much larger file size than a sound recorded at 8 kHz. Bit depth or resolution is another factor in audio file size. A sound recorded with 8-bit resolution (useful for a voice or other simple sounds) will have a smaller file size than a sound recorded using 16-bit resolution (music CD quality).

The Windows and Mac operating systems contain audio recording utilities. Audacity is a free cross-platform digital audio editor (available at http://sourceforge.net/projects/audacity for both Windows and Mac). Apple's Quicktime Pro (available for both Windows and Mac) is a low-cost application that can be used to record audio. If you are using a Mac, another option is Apple's GarageBand, which is a low-cost music application that offers a range of options for recording and editing audio.

More About Video Files

Focus on Ethics

Just as with audio files, there are a number of ways that you can obtain video files, including recording your own, downloading videos, purchasing a DVD that contains videos, or searching for video files on the Web. Be aware that there are ethical issues related to using videos that you did not create yourself. You must obtain the rights or license to publish videos created by other individuals before publishing them on your website.

Many digital cameras and smartphones have the capability to take still photographs as well as short MP4 movies. This can be an easy way to create short video clips. Digital video cameras and webcams record digital videos. Once you have created your video, software such as Adobe Premiere (http://www.adobe.com/products/premiere.html), Apple QuickTime Pro (http://www.apple.com/quicktime/pro), Apple iMovie (http://www.apple.com/ilife/imovie), or Nero Video (http://www.nero.com/enu/products/nero-video) can be used to edit and configure your video masterpiece. Many digital cameras and smartphones provide the ability record to videos and immediately upload them to YouTube to share with the world. You'll work with a YouTube video in Chapter 13.

Multimedia and Accessibility Issues

Focus on Accessibility

Provide alternate content for the media files you use on your website in transcript, caption, or printable PDF format. Provide a text transcript for audio files such as podcasts. Often, you can use the podcast script as the basis of the text transcript file that you create as a PDF and upload to your website. Provide captions for video files. Applications such as Media Access Generator (MAGpie) can add captioning to videos. See the National Center for Accessible Media's website (http://ncam.wgbh.org/invent_build/web_multimedia/tools-guidelines/magpie) for the most up-to-date information on the application. Apple QuickTime Pro includes a captioning function. View an example of a captioned video in the student files (chapter11/starters/sparkycaptioned.mov). When you upload a video to YouTube (http://www.youtube.com), captions can be automatically generated (although you'll probably want to make some corrections). You can also create a transcript or text captions for an existing YouTube video (see https://support.google.com/youtube/topic/3014331?rd=1).

Browser Compatibility Issues

As you completed the Hands-On Practice, you may have encountered playback issues in various browsers. Playing audio and video files on the Web depends on the plug-ins installed in your visitor's web browsers. A page that works perfectly on your home computer may not work for all visitors; it depends on the configuration of the computer. Some visitors will not have the plug-ins properly installed. Some visitors may have file types associated with incorrect plug-ins or incorrectly installed plug-ins. Some visitors may be using low bandwidth and have to wait an overly long time for your media file to download. Are you detecting a pattern here? Sometimes media on the Web can be problematic.

In a response to these browser plug-in compatibility issues and in an effort to reduce reliance on a proprietary technology like Adobe Flash, HTML5 introduces new audio and video elements that are native to the browser. However, because HTML5 is not yet supported by commonly used browsers (such as older versions of Internet Explorer), web designers still need to provide a fallback option, such as providing a hyperlink to the media file or displaying a Flash version of the multimedia. You'll work with HTML5 audio and video later in this chapter, but first, let's explore Adobe Flash.

11.3 Adobe Flash

Adobe **Flash** is an application that can be used to add visual interest and interactivity to web pages with slide shows, animation, and multimedia effects. Flash animation can be interactive—it can be scripted, with a language called ActionScript, to respond to mouse clicks, accept information in text boxes, and invoke server-side scripting. Flash can also be used to play audio and video files. Flash multimedia files are stored in a **.swf** file extension and require the Flash Player browser plug-in.

HTML5 Embed Element

The **embed element** is a self-contained, or void, element whose purpose is to provide a container for external content (such as Flash) that requires a plug-in or player. Although used for many years to display Flash on web pages, the embed element was never an official W3C element until HTML5. One of the design principles of HTML5 is to "pave the cowpaths"—to smooth the way for valid use of techniques that, although supported by browsers, were not part of the official W3C standard. Figure 11.2 (also in the student files

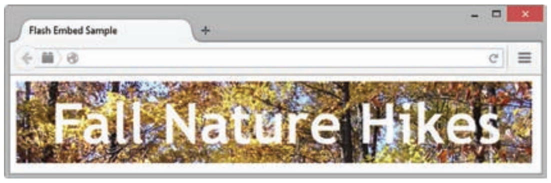


Figure 11.2 The embed element was used to configure the Flash media

at chapter11/flashembed.html) shows a web page using an embed element to display a Flash .swf file. The attributes of the embed element commonly used with Flash media are listed in Table 11.3.

Table 11.3 Embed element attributes

Attribute	Description and Value
`src`	File name of the Flash media (SWF file)
`height`	Specifies the height of the object area in pixels
`type`	The MIME type of the object; use `type="application/x-shockwave-flash"`
`width`	Specifies the width of the object area in pixels
`bgcolor`	Optional; hexadecimal value for the background color of the Flash
`quality`	Optional; describes the quality of the media, usually set to "high"
`title`	Optional; specifies a brief text description that may be displayed by browsers or assistive technologies
`wmode`	Optional; set to "transparent" to configure a transparent background in supporting browsers

The following code configures the Flash SWF file shown in Figure 11.2:

```
<embed type="application/x-shockwave-flash"
       src="fall5.swf"
       width="640"
       height="100"
       quality="high"
       title="Fall Nature Hikes">
```

Focus on Accessibility

Notice the value of the title attribute in the previous code. The descriptive text could be accessed by assistive technologies such as a screen reader.

Hands-On Practice 11.2

In this Hands-On Practice, you will launch a text editor and create a web page that displays a Flash slide show of photographs. Your web page will look like the one shown in Figure 11.3.

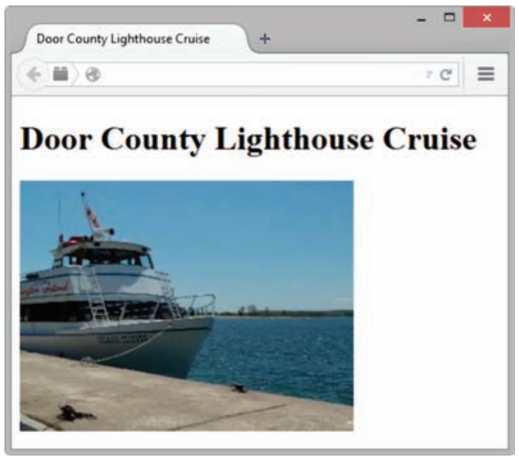


Figure 11.3 Flash slide show of images configured with the embed element

Create a folder called embed. Copy the lighthouse.swf file from the chapter11/starters folder in the student files and save it in your embed folder. Use the chapter2/template.html file as a starting point and create a web page with the heading "Door County Lighthouse Cruise" and an embed element to display a Flash file named lighthouse.swf that is 320 pixels wide and 240 pixels high. The code is

```
<embed type="application/x-shockwave-flash"
       src="lighthouse.swf" quality="high"
       width="320" height="240"
       title="Door County Lighthouse Cruise">
```

Save your web page as index.html in the embed folder and test it in a browser. Compare your work to chapter11/11.2/index.html in the student files.

Flash Resources

There are many sources of free Flash animation and Flash tutorials on the Web. In addition to resources at the Adobe site (http://adobe.com), the following websites contain tutorials and news about Flash:

- Flash Kit: http://flashkit.com
- ScriptOcean: http://www.scriptocean.com/flashn.html
- Kirupa: http://www.kirupa.com/developer/flash/index.htm

As you visit these and other Flash resource sites, keep in mind that some Flash media is copyrighted. Obtain permission from the creator of the media before using it on your site and follow any instructions for giving credit to the source. Some sites allow personal use of their Flash media for free, but require licenses for commercial use.

Focus on Ethics

FAQ What's Microsoft Silverlight?

Silverlight (http://www.silverlight.net) is a tool for delivering media experiences and rich interactive applications for the Web, which are displayed on web pages by the Silverlight plug-in.

FAQ What will happen if the browser my web page visitor uses does not support Flash?

If you used the code in this section to display Flash media on a web page and your visitor's browser does not support Flash, the browser typically will display a message about the need for a missing plug-in. The Adobe Flash Player plug-in for desktop browsers can be downloaded at http://www.adobe.com/products/flashplayer.html.

Although the Flash player is installed on most desktop web browsers, be aware that many users of mobile devices will not be able to view your Flash multimedia. There is no Flash support in the iPhone or iPad. A mobile Flash Player was previously available for Android devices, but Adobe has dropped support of Flash on Android. Mobile devices and modern desktop browsers support the new HTML5 video and audio elements, which are introduced next in this chapter. You'll also explore the potential of the HTML5 canvas element, CSS3 transform property, and CSS3 transition property later in this chapter.

Checkpoint 11.1

1. List three common web browser plug-ins and describe their use.
2. Describe the issues involved with adding media such as audio or video to a web page.
3. Describe a disadvantage of using Flash on a web page.

11.4 HTML5 Audio and Video Elements

The new HTML5 audio and video elements enable browsers to natively play media files without the need for browser plug-ins. When working with native HTML5 audio and video, you need to be aware of the **container** (which is designated by the file extension) and the **codec** (which is the algorithm used to compress the media). Refer to Table 11.1 and Table 11.2, which list common media file extensions, the container file type, and a description with codec information (if applicable for HTML5). Let's get started using the HTML5 audio element.

Audio Element

The new HTML5 **audio element** supports native play of audio files in the browser without the need for plug-ins or players. The audio element begins with the **`<audio>`** tag and ends with the `</audio>` tag. Table 11.4 lists the attributes of the audio element.

You'll need to supply multiple versions of the audio file because of the browser support of different codecs. Plan to supply audio files in at least two different containers, including OGG and MP3. It is typical to omit the `src` and `type` attributes from the audio tag and, instead, configure multiple versions of the audio file with the source element.

Table 11.4 Audio element attributes

Attribute	Value	Usage
`src`	File name	Optional; audio file name
`type`	MIME type	Optional; the MIME type of the audio file, such as audio/mpeg or audio/ogg
`autoplay`	`autoplay`	Optional; indicates whether audio should start playing automatically; use with caution
`controls`	`controls`	Optional; indicates whether controls should be displayed; recommended
`loop`	`loop`	Optional; indicates whether audio should be played over and over
`preload`	`none, metadata, auto`	Optional; values: `none` (no preload), `metadata` (only download media file metadata), and `auto` (download the media file)
`title`	Text description	Optional; specifies a brief text description that may be displayed by browsers or assistive technologies

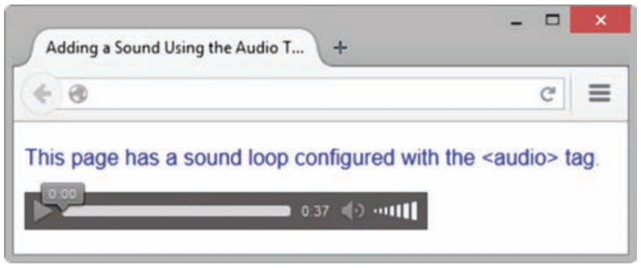
Source Element

The **source element** is a self-contained, or void, tag that specifies a media file and a MIME type. The `src` attribute identifies the file name of the media file. The `type` attribute indicates the MIME type of the file. Code `type="audio/mpeg"` for an MP3 file. Code `type="audio/ogg"` for audio files using the Vorbis codec. Configure a source element for each version of the audio file. Place the source element before the closing audio tag.

HTML5 Audio on a Web Page

The following code sample configures the web page shown in Figure 11.4 (see chapter11/audio.html in the student files) to display a controller for an audio file:

```
<audio controls="controls">
      <source src="soundloop.mp3" type="audio/mpeg">
      <source src="soundloop.ogg" type="audio/ogg">
      <a href="soundloop.mp3">Download the Audio File</a> (MP3)
</audio>
```

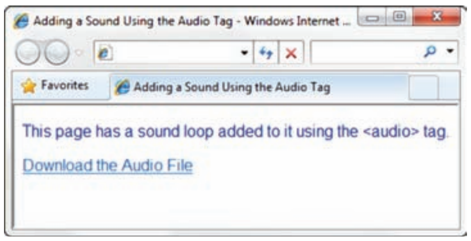


This page has a sound loop configured with the <audio> tag.

Figure 11.4 The Firefox browser supports the HTML5 audio element. Screenshots of Internet Explorer. Copyright by Microsoft Corporation. Used by Permission of Microsoft Corporation

Current versions of Safari, Chrome, Firefox, Microsoft Edge, and Opera support the HTML5 audio element. While Internet Explorer (version 9 and later) supports the audio element, earlier versions of Internet Explorer offer no support. The controls displayed by each browser are different.

Review the previous code and note the hyperlink placed between the second source element and the closing audio tag. Any HTML elements or text placed in this area is rendered by browsers that do not support the HTML5 audio element. This is referred to as fallback content; if the audio element is not supported, the MP3 version of the file is made available for download. Figure 11.5 shows a screen shot of Internet Explorer 8 displaying the web page.


Adding a Sound Using the Audio Tag - Windows Internet ...
Favorites
Adding a Sound Using the Audio Tag
This page has a sound loop added to it using the <audio> tag.

Download the Audio File

Figure 11.5 Internet Explorer 8 does not recognize the audio element. Screenshots of Internet Explorer. Copyright by Microsoft Corporation. Used by Permission of Microsoft Corporation

Hands-On Practice 11.3

In this Hands-On Practice, you will launch a text editor and create a web page (see Figure 11.6) that displays an audio control to play a podcast.

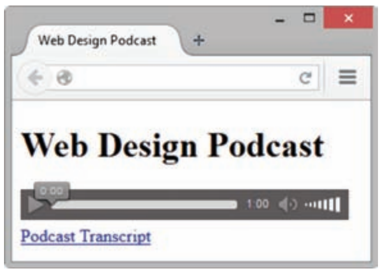


Figure 11.6 Using the audio element to provide access to a podcast

Copy the podcast.mp3, podcast.ogg, and podcast.txt files from the chapter11/starters folder in the student files and save them to a folder named audio5. Use the chapter2/template.html file as a starting point and create a web page with the heading "Web Design Podcast", an audio control (use the audio element and two source elements), and a hyperlink to the podcast.txt transcript file. Configure a hyperlink to the MP3 file as the fallback content. The code for the audio element is

```
<audio controls="controls">
    <source src="podcast.mp3" type="audio/mpeg">
    <source src="podcast.ogg" type="audio/ogg">
    <a href="podcast.mp3">Download the Podcast</a> (MP3)
</audio>
```

Save your web page as index.html in the audio5 folder and display it in a browser. Test your web page in different browsers, using different versions. Recall that Internet Explorer versions prior to Version 9 do not support the audio element, but they will display the fallback content. When you click on the hyperlink for the text transcript, the text will display in the browser. Compare your work to chapter11/11.3/index.html in the student files.

FAQ How can I convert an audio file to the Ogg Vorbis codec?

The open-source Audacity application supports Ogg Vorbis. See http://sourceforge.net/projects/audacity for download information. If you're looking for a free Web-based converter, you can upload and share an audio file at the Internet Archive (http://www.archive.org) and an OGG format file will automatically be generated.

Video Element

The new HTML5 **video element** supports native play of video files in the browser without the need for plug-ins or players. The video element begins with the **`<video>`** tag and ends with the `</video>` tag. Table 11.5 lists the attributes of the video element.

Table 11.5 Video element attributes

Attribute	Value	Usage
`src`	File name	Optional; video file name
`type`	MIME type	Optional; the MIME type of the video file, such as video/mp4 or video/ogg
`autoplay`	`autoplay`	Optional; indicates whether video should start playing automatically; use with caution
`controls`	`controls`	Optional; indicates whether controls should be displayed; recommended
`height`	`number`	Optional; video height in pixels
`loop`	`loop`	Optional; indicates whether video should be played over and over
`poster`	File name	Optional; specifies an image to display if the browser cannot play the video
`preload`	`none, metadata, auto`	Optional; values: `none` (no preload), `metadata` (only download media file metadata), and `auto` (download the media file)
`title`	Text description	Optional; specifies a brief text description that may be displayed by browsers or assistive technologies
`width`	Number	Optional; video width in pixels

You'll need to supply multiple versions of the video file because of the browser support of different codecs. Plan to supply video files in at least two different containers, including MP4 and OGG (or OGV). It is typical to omit the `src` and `type` attributes from the video tag and, instead, configure multiple versions of the audio file with the source element.

Source Element

Recall from the previous section that the source element is a self-contained, or void, tag that specifies a media file and a MIME type. The `src` attribute identifies the file name of the media file. The `type` attribute indicates the MIME type of the file. Code `type="video/mp4"` for video files using the MP4 codec. Code `type="video/ogg"` for video files using the Theora codec. Configure a source element for each version of the video file. Place the source elements before the closing video tag.

HTML5 Video on a Web Page

The following code configures the web page shown in Figure 11.7 (chapter11/sparky2.html in the student files) with the native HTML5 browser controls to display and play a video:

```
<video controls="controls" poster="sparky.jpg" width="160"
height="150">
   <source src="sparky.m4v" type="video/mp4">
   <source src="sparky.ogv" type="video/ogg">
   <a href="sparky.mov">Sparky the Dog</a> (.mov)
</video>
```

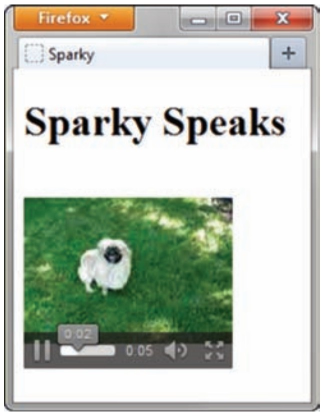


Figure 11.7 The Firefox browser. Screenshots of Mozilla Firefox. Courtesy of Mozilla Foundation

Current versions of Safari, Chrome, Firefox, Microsoft Edge, and Opera support the HTML5 video element. Internet Explorer 9 supports the video element, but

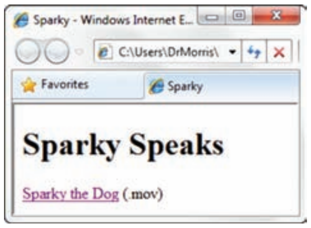


Figure 11.8 Internet Explorer 8 displays the fallback option.

earlier versions do not. The controls displayed by each browser are different. Review the code just given and note the anchor element placed between the second source element and the closing video tag. Any HTML elements or text placed in this area is rendered by browsers that do not support the HTML5 video element. This is referred to as fallback content. In this case, a hyperlink to a QuickTime (.mov) version of the file is supplied for the user to download. Another fallback option is to configure an embed element to play a Flash SWF version of the video. Figure 11.8 shows Internet Explorer 8 displaying the web page.

VideoNote
HTML5 Video

Hands-On Practice 11.4

In this Hands-On Practice, you will launch a text editor and create the web page in Figure 11.9, which displays a video control to play a movie.

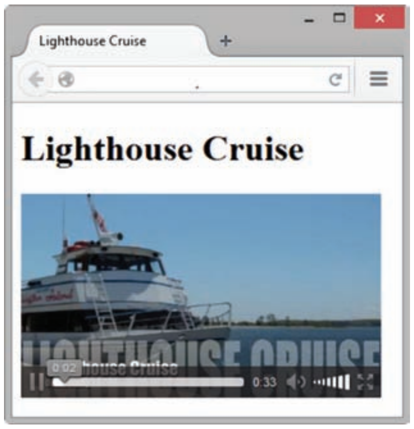


Figure 11.9 HTML5 video element. Screenshots of Internet Explorer. Copyright by Microsoft Corporation. Used by Permission of Microsoft Corporation

Copy the lighthouse.m4v, lighthouse.ogv, lighthouse.swf, and lighthouse.jpg files from the chapter11/starters folder in the student files and save them to a folder named video. Open chapter2/template.html in a text editor and configure a web page with the heading “Lighthouse Cruise” and a video control (use the video element and two source elements). While we could configure a hyperlink to a video file as fallback content, in this example, we’ll configure an embed element to display the Flash media file (lighthouse.swf) as fallback content. Configure the lighthouse.jpg file as a poster image, which will display if the browser supports the video element, but it cannot play any of the video files. The code for the video element is

```
<video controls="controls" poster="lighthouse.jpg" width="320"
height="240">
    <source src="lighthouse.m4v" type="video/mp4">
    <source src="lighthouse.ogv" type="video/ogg">
    <embed type="application/x-shockwave-flash"
      src="lighthouse.swf" quality="high" width="320" height="240"
      title="Door County Lighthouse Cruise">
</video>
```

Save your web page as index.html in the video folder and display it in a browser. Test your web page in different browsers. Compare your work to Figure 11.9 and chapter11/11.4/index.html in the student files.

FAQ **How can I convert a video file to the new codecs?**

You can use Firefogg (http://firefogg.org) to convert your video file to the Ogg Theora codec. Online-Convert offers free conversion to WebM (http://video.online-convert.com/convert-to-webm). The free, open-source Miro-VideoConverter (http://www.mirovideoconverter.com) can convert most video files to MP4, WebM, or OGG formats.

11.5 Multimedia Files and Copyright Law

Focus on Ethics

It is very easy to copy and download an image, audio, or video file from a website. It may be very tempting to place someone else's file in one of your own projects, but that may not be ethical or lawful. Only publish web pages, images, and other media that you have personally created or have obtained the rights or license to use. If another individual has created an image, sound, video, or document that you think would be useful on your own website, ask permission to use the material instead of simply taking it. All work (web pages, images, sounds, videos, and so on) is **copyrighted**, even if there is no copyright symbol and date on the material.

Be aware that there are times when students and educators can use portions of another's work and not be in violation of copyright law. This is called **fair use**. Fair use is the use of a copyrighted work for purposes such as criticism, reporting, teaching, scholarship, or research. The criteria used to determine fair use are as follows:

- The use must be educational rather than commercial.
- The nature of the work copied should be factual rather than creative.
- The amount copied must be as small a portion of the work as possible.
- The copy does not impede the marketability of the original work.

Visit the U.S. Copyright Office (http://copyright.gov) and Copyright Website (http://www.copyrightwebsite.com) for some additional information about copyright issues.

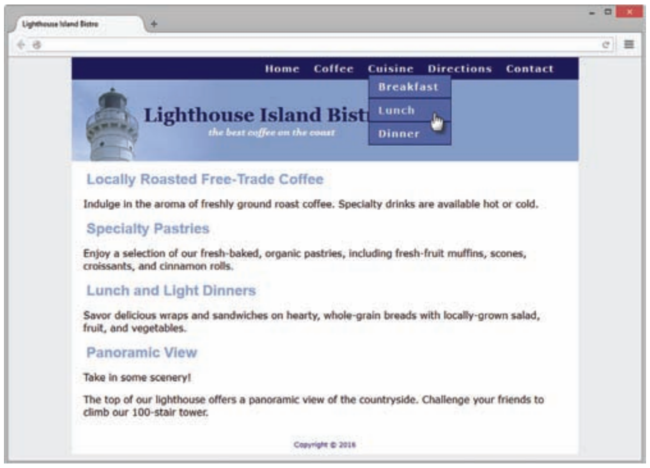
Some individuals may want to retain ownership of their work, but make it easy for others to use or adapt it. Creative Commons (http://creativecommons.org) provides a free service that allows authors and artists to register a type of copyright license called a **Creative Commons license**. There are several licenses to choose from, depending on the rights you wish to grant as the author. The Creative Commons license informs others exactly what they can and cannot do with the creative work.

11.6 CSS3 and Interactivity

CSS Drop Down Menu

Recall from Chapter 6 that the CSS `:hover` pseudo-class provides a way to configure styles to display when the web page visitor moves the mouse over an element. You'll use this basic interactivity, along with CSS positioning and display properties, to configure an interactive navigation menu with CSS and HTML in the next Hands-On Practice.

Hands-On Practice 11.5



Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and cinnamon rolls.

Lunch and Light Dinners

Savor delicious wraps and sandwiches on hearty, whole-grain breads with locally-grown salad, fruit, and vegetables.

Panoramic View

Take in some scenery!

The top of our lighthouse offers a panoramic view of the countryside. Challenge your friends to climb our 100-stair tower.

Copyright © 2016

Figure 11.10 An interactive navigation menu with CSS

In this Hands-On Practice you will configure a drop down menu that displays when a visitor hovers over the Cuisine navigation hyperlink as shown in Figure 11.10. The main menu has hyperlinks for Home, Coffee, Cuisine, Directions, and Contact. As shown in the site map (Figure 11.11) the Cuisine page has three subpages: Breakfast, Lunch, and Dinner. Create a folder named mybistro. Copy the files from the chapter11/bistro folder in the student files into your mybistro folder. You will modify the CSS and edit each page to configure a Cuisine submenu that provides hyperlinks to three pages (Breakfast, Lunch, and Dinner).

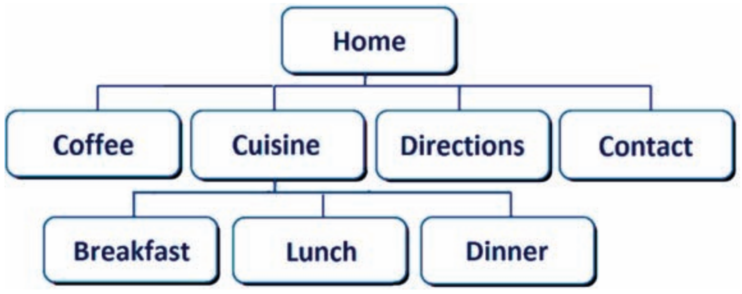


Figure 11.11 Site map

Task 1: Configure the HTML. Launch a text editor and open the index.html file. Modify the nav area to contain a new unordered list with hyperlinks to the Breakfast, Lunch, and Dinner pages. Configure a new ul element that is contained *within* the Cuisine li element. The new ul element will contain an li element for each meal. The HTML follows with the new code displayed in blue.

```
<nav>
<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="coffee.html">Coffee</a></li>
  <li><a href="cuisine.html">Cuisine</a>
    <ul>
      <li><a href="breakfast.html">Breakfast</a></li>
      <li><a href="lunch.html">Lunch</a></li>
      <li><a href="dinner.html">Dinner</a></li>
    </ul>
  </li>
  <li><a href="directions.html">Directions</a></li>
  <li><a href="contact.html">Contact</a></li>
</ul>
</nav>
```

Save the file and display it in a browser. Don't worry if the navigation area seems a bit garbled—you'll configure the submenu CSS in Step 2. Next, edit each page (coffee.html, cuisine.html, breakfast.html, lunch.html, dinner.html, directions.html and contact.html) and edit the nav area as you did in the index.html file.

Task 2: Configure the CSS. Launch a text editor and open the bistro.css file.

a. The submenu will be configured with absolute positioning. Recall from Chapter 7 that absolute positioning precisely specifies the location of an element outside of normal flow in relation to its first parent non-static element. The nav element's position is static by default so add the following declaration to the styles for the nav element selector:

```
position: relative;
```

b. The submenu that displays the hyperlinks for the Breakfast, Lunch, and Dinner pages is configured using a new ul element that is contained within the existing ul element in the nav area. Configure a descendant `nav ul ul` selector and code style declarations to use absolute positioning, #5564A0 background color, 0 padding, left text alignment and display set to none. The CSS follows:

```
nav ul ul { position: absolute;
            background-color: #5564A0;
            padding: 0;
            text-align: left;
            display: none; }
```

c. To style each li element within the submenu, use a descendant `nav ul ul li` selector and configure the li elements in the submenu with a border, block display, 8em width, 1em left padding, and 0 left margin. The CSS follows:

```
nav ul ul li { border: 1px solid #00005D;
            display: block;
            width: 8em;
            padding-left: 1em;
            margin-left: 0; }
```

d. Configure the submenu ul to display when the `:hover` is triggered for the li elements in the nav area. The CSS follows:

```
nav li:hover ul { display: block; }
```

Test your pages in a browser. The drop down menu should look similar to Figure 11.10. You can compare your work to the sample in the student files (chapter11/11.5/horizontal). An example of a web page with a vertical fly-out menu is available in the student files (chapter11/11.5/vertical).

CSS3 Transform Property

CSS transforms allow you to change the display of an element and provides functions to rotate, scale, skew, and reposition an element. The **`transform` property** is supported by current versions of modern browsers, including Internet Explorer (version 10 and later). Both two-dimensional (2D) and three-dimensional (3D) transforms are possible.

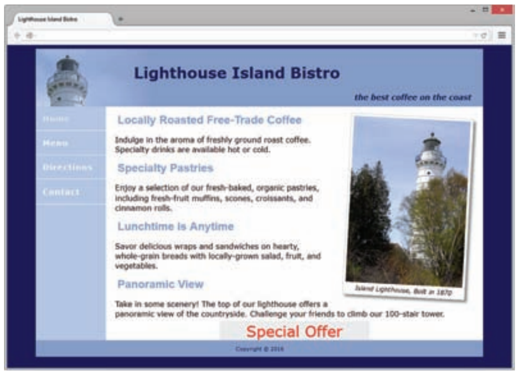
Table 11.6 lists commonly used 2D transform property function values and their purpose. See http://www.w3.org/TR/css3-transforms/#transform-property for a complete list. We'll focus on the rotate and scale transforms in this section.

Table 11.6 Values of the transform property

Value	Purpose
`rotate` (*degree*)	Rotates the element by the angle
`scale` (*number, number*)	Scales or resizes the element along the X and Y axis (X,Y)
`scaleX` (*number*)	Scales or resizes the element along the X axis
`scaleY` (*number*)	Scales or resizes the element along the Y axis
`skewX` (*number*)	Distorts the display of the element along the X axis
`skewY` (*number*)	Distorts the display of the element along the Y axis
`translate` (*number, number*)	Repositions the element along the X and Y axis (X,Y)
`translateX` (*number*)	Repositions the element along the X axis
`translateY` (*number*)	Repositions the element along the Y axis

CSS3 Rotate Transform

The **`rotate()` transform** function takes a value in degrees (like an angle in geometry). Rotate to the right with a positive value. Rotate to the left with a negative value. The rotation is around the origin, which, by default, is the middle of the element. The web page in Figure 11.12 demonstrates the use of the CSS3 transform property to slightly rotate the figure.



Lighthouse Island Bistro

the best coffee on the coast

Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and cinnamon rolls.

Lunchtime Is Anytime

Savor delicious wraps and sandwiches on hearty, whole-grain breads with locally-grown salad, fruit, and vegetables.

Panoramic View

Take in some scenery! The top of our lighthouse offers a panoramic view of the countryside. Challenge your friends to climb our 100-stair tower.

Special Offer

Copyright © 2016

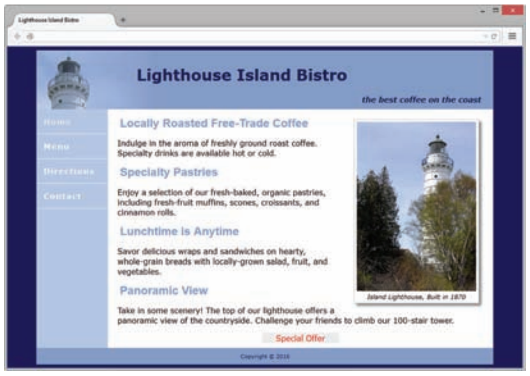
Figure 11.12 The transform property in action

CSS3 Scale Transform

The **`scale()` transform** function resizes an element in three different ways: along the X-axis, along the Y-axis, and along both the X- and Y-axes. Specify the amount of resizing using a number without units. For example, scale(1) does not change the element's size, scale(2) indicates the element should render two times as large, scale(3) indicates the element should render three times as large, and scale(0) indicates the element should not display.

Hands-On Practice 11.6

In this Hands-On Practice you will configure the rotation and scale transforms shown in Figure 11.12. Create a new folder named transform. Copy the lighthouseisland.jpg, lighthousemini.jpg, and lighthouselogo.jpg images from the chapter11/starters folder in the student files to your transform folder. Launch a text editor and open the starter.html file in the chapter11 folder. Save the file as index.html in your transform folder. Launch the file in a browser and it will look similar to Figure 11.13.



Lighthouse Island Bistro

the best coffee on the coast

Home
Menu
Directions
Contact

Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and cinnamon rolls.

Lunchtime is Anytime

Savor delicious wraps and sandwiches on hearty, whole-grain breads with locally-grown salad, fruit, and vegetables.

Panoramic View

Take in some scenery! The top of our lighthouse offers a panoramic view of the countryside. Challenge your friends to climb our 100-stair tower.

Island Lighthouse, Built in 1870

Special Offer

Copyright © 2016

Figure 11.13 Before the transform property

Open index.html in a text editor and view the embedded CSS.

1. Locate the figure element selector. You will add new style declarations to the figure element selector that will configure a three-degree rotation transform. The new CSS is shown in blue.

```
figure { float: right; margin: 10px; background-color: #FFF;
         padding: 5px; border: 1px solid #CCC;
         box-shadow: 5px 5px 5px #828282;
         transform: rotate(3deg); }
```

2. Locate the `#offer` selector. This configures the "Special Offer" div displayed above the page footer. You will add a style declaration to the `#offer` selector that configures the browser to display the element two times larger. The new CSS is shown in blue.

```
#offer { background-color: #eaeaea;
         width: 10em;
         margin: auto;
         text-align: center;
         transform: scale(2); }
```

Save the file and display it in a browser. You should see the figure displayed on a slight angle and the "Special Offer" text displayed in large text. Compare your work to Figure 11.12 and the sample in the student files (chapter11/11.6/index.html).

Explore Transforms

This section provided an overview of the rotate and scale transforms. Visit http://www.westciv.com/tools/transforms/index.html to generate the CSS for rotate, scale, translate, and skew transforms. Find out more about transforms at http://www.css3files.com/transform and http://developer.mozilla.org/en/CSS/Using_CSS_transforms.

CSS Transition Property

CSS3 **transitions** provide for changes in property values to display in a smoother manner over a specified time. Transitions are supported by current versions of most modern browsers, including Internet Explorer (version 10 and later). You can apply a transition to a variety of CSS properties including `color`, `background-color`, `border`, `font-size`, `font-weight`, `margin`, `padding`, `opacity`, and `text-shadow`. A full list of applicable properties is available at http://www.w3.org/TR/css3-transitions. When you configure a transition for a property, you need to configure values for the `transition-property`, `transition-duration`, `transition-timing-function`, and `transition-delay` properties. These can be combined in a single **`transition` property**. Table 11.7 lists the transition properties and their purpose. Table 11.8 lists commonly used transition-timing-function values and their purpose.

Table 11.7 CSS transition properties

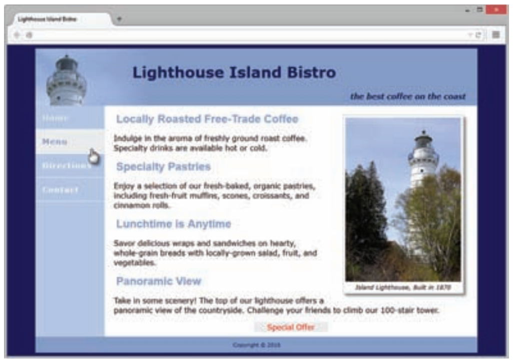
Property	Description
`transition-property`	Indicates the CSS property to which the transition applies
`transition-duration`	Indicates the length of time to apply the transition; default value 0 configures an immediate transition; a numeric value specifies time (usually in seconds)
`transition-timing-function`	Configures changes in the speed of the transition by describing how intermediate property values are calculated; common values include `ease` (default), `linear, ease-in, ease-out, ease-in-out`
`transition-delay`	Indicates the beginning of the transition; default value 0 configures no delay; a numeric value specifies time (usually in seconds)
`transition`	Shorthand property; list the value for `transition-property, transition-duration`, `transition-timing-function`, and `transition-delay` separated by spaces; default values can be omitted, but the first time unit applies to `transition-duration`

Table 11.8 Commonly used transition-timing-function values

Value	Purpose
`ease`	Default; transition effect begins slowly, speeds up, and ends slowly
`linear`	Transition effect has a constant speed
`ease-in`	Transition effect begins slowly and speeds up to a constant speed
`ease-out`	Transition effect begins at a constant speed and slows down
`ease-in-out`	Transition effect is slightly slower; Begins slowly, speeds up, and slows down

Hands-On Practice 11.7

Recall that the CSS `:hover` pseudo-class provides a way to configure styles to display when the web page visitor moves the mouse over an element. The change in display happens somewhat abruptly. Web designers can use a CSS transition to create a more gradual change to the hover state. You'll try this out in this Hands-On Practice when you configure a transition for the navigation hyperlinks on a web page.



Lighthouse Island Bistro

the best coffee on the coast

Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and cinnamon rolls.

Lunchtime is Anytime

Savor delicious wraps and sandwiches on hearty, whole-grain breads with locally-grown salad, fruit, and vegetables.

Panoramic View

Take in some scenery! The top of our lighthouse offers a panoramic view of the countryside. Challenge your friends to climb our 100-stair tower.

Figure 11.14 The transition in action

Create a new folder named transition. Copy the lighthouseisland.jpg, lighthousemini.jpg, and lighthouselogo.jpg images from the chapter11/starters folder in the student files to your transition folder. Launch a text editor and open the starter.html file in the chapter11 folder. Save the file as index.html in your transition folder. Open index.html in a browser and it will look similar to Figure 11.13. Place your mouse over one of the navigation hyperlinks and notice that the background color and text color change immediately.

Open index.html in a text editor and view the embedded CSS. Locate the `nav a:hover` selector and notice that the color and background-color properties are configured. You will add new style declarations to the `nav a` selector to cause a more gradual change in the background color when the user places the mouse over the hyperlink. The new CSS is shown in blue.

```
nav a { text-decoration: none; display: block; padding: 15px;
        transition: background-color 2s linear; }
```

Save the file and display it in a browser. Place your mouse over one of the navigation hyperlinks and notice that while the text color changes immediately, the background color changes in a more gradual manner—the transition is working! Compare your work to Figure 11.14 and the student files (chapter11/11.7/index.html).

Explore Transitions

If you'd like more control over the transition than what is provided by the values listed in Table 11.8, explore using the cubic-bezier value for the transition-timing-function. A Bezier curve is a mathematically defined curve often used in graphic applications to describe motion. Explore the following resources:

- http://www.the-art-of-web.com/css/timing-function
- http://roblaplaca.com/blog/2011/03/11/understanding-css-cubic-bezier
- http://cubic-bezier.com

Practice with Transitions

Hands-On Practice 11.8

In this Hands-On Practice you will use CSS `positioning`, `opacity`, and `transition` properties to configure an interactive image gallery with CSS and HTML. This is a slightly different version of the image gallery than the web page you created in Hands-On Practice 6.8.

Figure 11.15 shows the initial display of the gallery (see the student files chapter11/11.8/index.html) with a semi-opaque placeholder image. When you place the mouse over

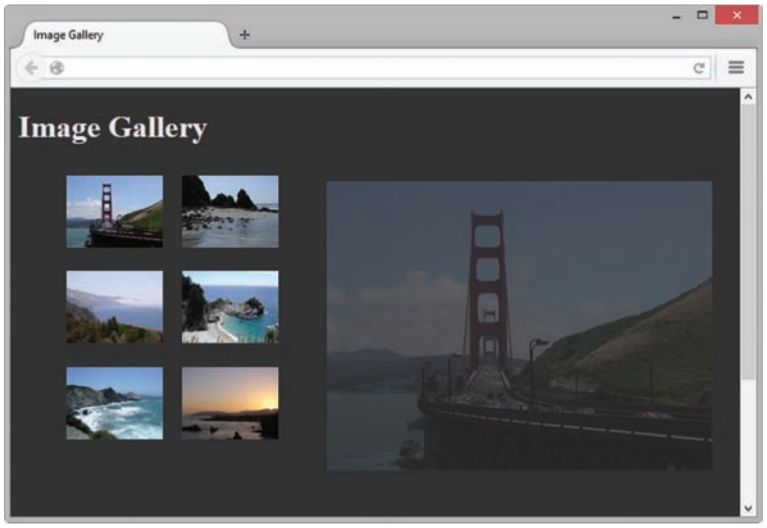


Figure 11.15 Initial display of the gallery

a thumbnail image, the larger version of that image is gradually displayed along with a caption (see Figure 11.16). If you click the thumbnail, the image will display in its own browser window.

Create a new folder called gallery2. Copy all the images from the chapter11/starters/gallery folder in the student files to the new gallery2 folder.

Launch a text editor and modify the chapter2/template.html file to configure a web page as indicated:

1. Configure the text, "Image Gallery", within an h1 element and within the title element.
2. Code a div element assigned to the id named `gallery`. This div will contain a placeholder figure element and an unordered list that contains the thumbnail images.
3. Configure a figure element within the div. The figure element will contain a placeholder img element that displays photo1.jpg.
4. Configure an unordered list within the div. Code six li elements, one for each thumbnail image. The thumbnail images will function as image links with a `:hover` pseudo-class that causes the larger image to display on the page. We'll make this all happen by configuring an anchor element containing both the thumbnail image and a span element that comprises the larger image along with descriptive text. An example of the first li element is

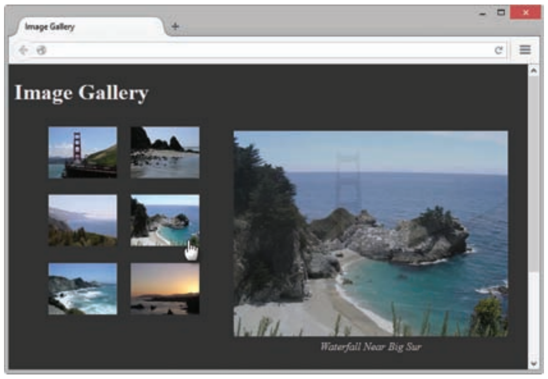


Figure 11.16 The new photo gradually displays

```
<li><a href="photo1.jpg"><img src="photo1thumb.jpg" width="100"
 height="75" alt="Golden Gate Bridge">
 <span><img src="photo1.jpg" width="400" height="300"
 alt="Golden Gate Bridge"><br>Golden Gate Bridge</span></a>
</li>
```

5. Configure all six li elements in a similar manner. Substitute the actual name of each image file for the href and src values in the code. Write your own descriptive text for each image. Use photo2.jpg and photo2thumb.jpg in the second li element. Use photo3.jpg and photo3thumb.jpg in the third li element, and so on for all six images. Save the file as index.html in the gallery2 folder. Display your page in a browser. You'll see the placeholder image followed by an unordered list with the thumbnail images, the larger images, and the descriptive text.
6. Now, let's add CSS. Open your file in a text editor and code a style element in the head section. Configure embedded CSS as follows:
   a. Configure the body element selector with a dark background color (#333333) and a light gray text color (#eaeaea).
   b. Configure the `gallery` id selector. Set `position` to `relative`. This does not change the location of the gallery but sets the stage to use absolute positioning on the span element relative to its container (`#gallery`) instead of relative to the entire web page document.

c. Configure the figure element selector. Set `position` to `absolute, left` to 280px, `text-align` to center, and `opacity` to .25. This will cause the figure to initially be semi-opaque.

d. Configure the unordered list within the `#gallery` with a width of 300 pixels and no list marker.

e. Configure the list item elements within the `#gallery` with inline display, left float, and 10 pixels of padding.

f. Configure the img elements within the `#gallery` to not display a border.

g. Configure anchor elements within the `#gallery` with no underline, #eaeaea text color, and italic text.

h. Configure span elements within the `#gallery`. Set `position` to `absolute`, `left` to `-1000px` (which causes them not to display initially in the browser viewport), and `opacity` to 0. Also configure a three second `ease-in-out transition`.

```
#gallery span { position: absolute; left: -1000px; opacity: 0;
                transition: opacity 3s ease-in-out; }
```

i. Configure the span elements within the `#gallery` to display when the web visitor hovers the mouse over the thumbnail image link. Set `position` to `absolute`, `top` to 16px, `left` to 320px, centered text, and `opacity` to 1.

```
#gallery a:hover span { position: absolute; top: 16px; left: 320px;
                        text-align: center; opacity: 1; }
```

Save your file in the gallery2 folder and display it in a browser. Compare your work to Figure 11.15, Figure 11.16, and the student files (chapter11/11.8/index.html).

See the following resources for more examples of CSS transforms and transitions:

- CSS Transitions 101:
  http://www.webdesignerdepot.com/2010/01/css-transitions-101
- Using CSS Transitions:
  https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Using_CSS_transitions
- CSS: Animation Using CSS Transforms:
  http://www.the-art-of-web.com/css/css-animation/

11.7 Java

Java is an object-oriented programming (OOP) language developed by Sun Microsystems, which was later acquired by Oracle. An object-oriented program consists of a group of cooperating objects that exchange messages for the purpose of achieving a common objective. Java is not the same language as JavaScript. It is more powerful and much more flexible than JavaScript. Java can be used to develop both stand-alone executable applications and applets that are invoked by web pages.

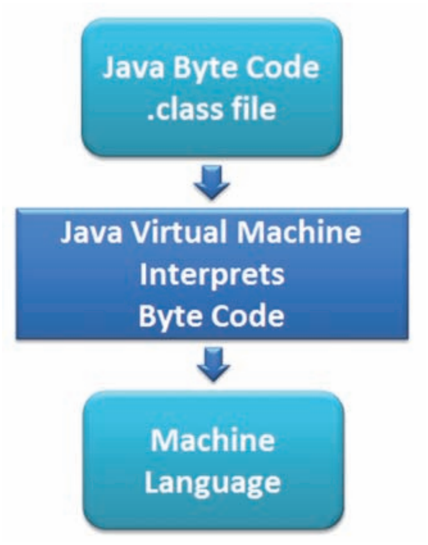
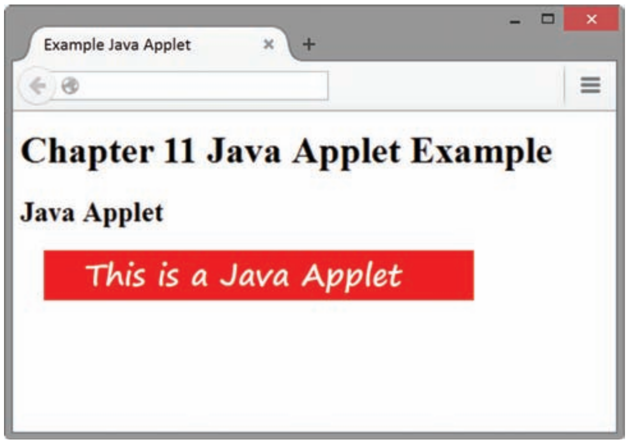


Figure 11.17 The Java Virtual Machine interprets the byte code into machine language

Java applets are platform independent, which means that they can be written and run on any platform: Mac, UNIX, Linux, and Windows. Java applets are compiled (translated from the English-like Java statements to an encoded form) and saved as **.class** files, which contain byte code. The byte code is interpreted by the **Java Virtual Machine (JVM)** in the web browser. The JVM interprets the byte code into the proper machine language for the operating system. The applet is then executed and appears on the web page. See Figure 11.17 for a diagram that shows this process. When a Java applet loads, the area reserved for it on the web page displays an empty rectangular area until the applet begins to execute.

Online games are a popular use of Java applets; try some classic Java applet games at Java on the Brain (http://www.javaonthebrain.com/brain.html). Java applets can also be used to create image and text effects, such as the example applet shown in Figure 11.18. While text effects and games are fun, Java applets may be used may be used in business applications for functions such as financial calculations and data visualization. As a web developer, your usual role will not be that of a Java software developer—that is, you should not be expected to write Java applets. However, you could be asked to work with a Java developer to place his or her applet on your website. Whether you obtain an applet from a coworker or find one on a free site, you need to code HTML to display the applet.



Chapter 11 Java Applet Example

Java Applet

This is a Java Applet

Figure 11.18 A Java applet that displays a text message

Adding a Java Applet to a Web Page

A while back, the now-obsolete applet element was typically used along with the param element to configure a Java applet on a web page. The W3C recommends using the multipurpose object element to configure a Java applet on a web page. The **object element** is a multipurpose container tag for adding various types of objects to a web page. The **`<object>`** tag specifies the beginning of an applet area in the body of a web page. Its closing tag, `</object>`, specifies the ending of an applet area in the body of a web page. The object element has a number of attributes that are used with Java applets, as described in Table 11.9.

Table 11.9 Object element attributes used with Java applets

Attribute	Value
type	Specifies the MIME type of the Java applet: application/x-java-applet
height	Specifies the height of the applet area in pixels
width	Specifies the width of the applet area in pixels
title	Optional; brief text description that may be displayed by browsers or assistive technologies

The software developer who creates an applet determines the parameter values and names required by a specific Java applet. Therefore, expect each applet to require different parameters. Other parameter names and expected values will be provided in the applet documentation. The parameters will be different, depending on the function of the applet. One parameter might be used to set a background color; another parameter could be used to contain a person's name. Parameters are configured with the **param element**, which is a void element with two attributes: name and value. All the **`<param>`** tags for the object appear before the closing `</object>` tag. The `code` parameter indicates the name of the Java applet.

Whether you obtain an applet from a free website or from a coworker, each applet should have some accompanying documentation that indicates what parameter it expects. Documentation for the example applet appears in Table 11.10.

Table 11.10 Documentation for example applet

Parameter Name	Parameter Value
code	example.class
message	The text to be displayed by the Java apple
textColor	The color of the text to be displayed by the Java applet; uses a hexadecimal color value
backColor	The background color of the Java applet area; uses a hexadecimal color value

The following code configures appropriate object and param elements to display a Java applet named example.class that is 610 pixels wide and 30 pixels high, has a white background (backColor parameter), has red text (textColor), and displays a text message parameter.

```
<object type="application/x-java-applet" width="610" height="30"
title="This Java Applet displays a message">
  <param name="code" value="example.class">
  <param name="textColor" value="#FF0000">
  <param name="message" value="This is a Java Applet">
  <param name="backColor" value="#FFFFFF">
  Java Applets can be used to display text, manipulate graphics, play
games, and more.
  Visit <a href="http://download.oracle.com/javase/tutorial/">Oracle</a>
for more information.
</object>
```

Java Applet Resources

There are many resources for free and commercial Java applets on the Web, including Java on the Brain (http://www.javaonthebrain.com) and Java Applet Archive (http://www.echoecho.com/freeapplets.htm).

Focus on Ethics

As you visit these and other Java resource sites, keep in mind that some Java applets are copyrighted. Be sure to obtain permission from the creator of the applet before using it on your site. There may be some requirements for giving credit to the creator either by name or by linking to their website. Follow the instructions provided with the applet. Some applets are free for use on personal websites, but require licenses for commercial use.

FAQ **Can you show an example of using the obsolete applet element?**

Sure! Although the applet element is obsolete, you'll probably run across web pages that still use it. An example of using the applet element to display a Java applet is

```
<applet code="example.class" width="610" height="30"
  alt="Displays a message that describes uses of Java applets.">
  <param name="textColor" value="#FF0000">
  <param name="message" value="This is a Java Applet">
  <param name="backColor" value="#FFFFFF">
  Java Applets can be used to display text, manipulate graphics, play
games, and more.
  <a href="http://download.oracle.com/javase/tutorial/">Oracle</a>
</applet>
```

Checkpoint 11.2

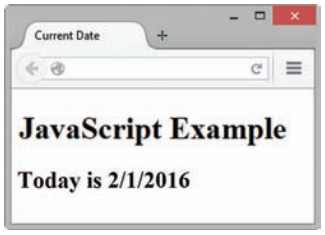
1. Describe a benefit of using the new HTML5 audio and video elements.
2. What is the purpose of the transform property?
3. Describe a disadvantage of using Java applets on web pages.

11.8 JavaScript

Although some interactivity on web pages can be achieved with CSS, JavaScript powers much of the interactivity on the Web. **JavaScript**, developed initially by Brendan Eich at Netscape, is an object-based, **client-side scripting** language interpreted by a web browser. JavaScript is considered to be **object-based** because it's used to manipulate the objects associated with a web page document: the browser window, the document itself, and elements such as forms, images, and hyperlinks.

JavaScript is not the same as the Java programming language. Unlike Java, JavaScript cannot be used to write stand-alone programs that can run outside of a web browser.

JavaScript statements can be placed in a separate file (with a .js extension) that is accessed by a web browser or within an HTML script element. The purpose of the **script element** is to either contain scripting statements or to indicate a file that contains scripting statements. Some JavaScript also can be coded within the HTML. In all cases, the web browser interprets the JavaScript statements. Because JavaScript is interpreted by a browser, it is considered to be a client-side scripting language.



JavaScript Example

Today is 2/1/2016

Figure 11.19 JavaScript in action

JavaScript can be used to respond to events such as moving the mouse, clicking a button, and loading a web page. This technology is also often utilized to edit and verify information on HTML form controls such as text boxes, check boxes, and radio buttons. Other uses for JavaScript include pop-up windows, image slideshows, animation, date manipulation, and calculations. Figure 11.19 shows a web page (found in the student files at chapter11/date.html) that uses JavaScript to determine and display the current date. The JavaScript statements are enclosed within an HTML script element and coded directly in the .html file. The code sample is below:

```
<h2>Today is
<script>
var myDate = new Date()
var month = myDate.getMonth() + 1
var day = myDate.getDate()
var year = myDate.getFullYear()
document.write(month + "/" + day + "/" + year)
</script>
</h2>
```

There is an introduction to coding JavaScript in Chapter 14. An important part of working with JavaScript is manipulating the **Document Object Model (DOM)**. The DOM defines every object and element on a web page. Its hierarchical structure can be used to access page elements and apply styles to page elements. A portion of a basic DOM that is common to most browsers is shown in Figure 11.20.

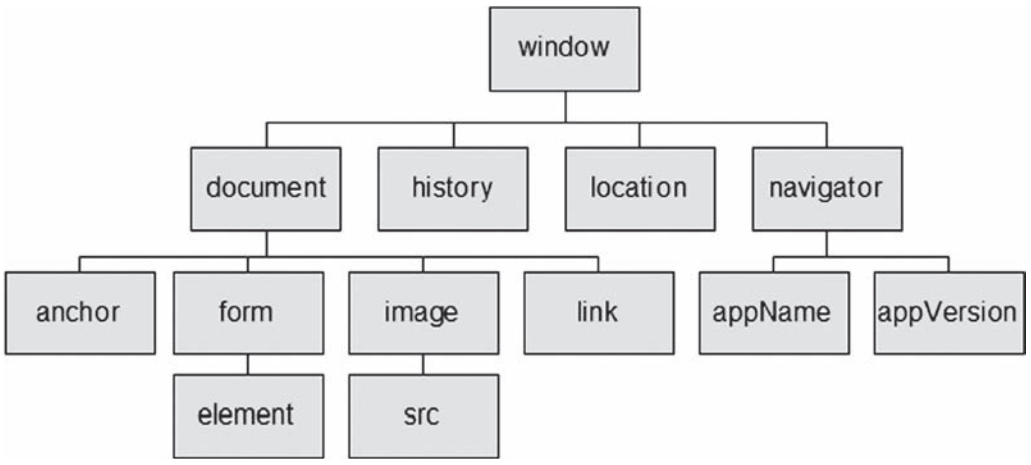


Figure 11.20 The Document Object Model (DOM)

JavaScript Resources

There is a lot to learn about JavaScript, but there are many free resources for JavaScript code and JavaScript tutorials on the Web. Here are a few sites that offer free tutorials:

- JavaScript Tutorial: http://echoecho.com/javascript.htm
- Mozilla Developer Network JavaScript Guide: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide
- JavaScript Tutorial: http://www.w3schools.com/JS

Once you are comfortable with HTML and CSS, the JavaScript language is a good technology to learn as you continue your studies. Try some of the resources listed and get your feet wet. See Chapter 14 for a more detailed introduction to JavaScript. The next section introduces Ajax, a technology that uses JavaScript.

11.9 Ajax

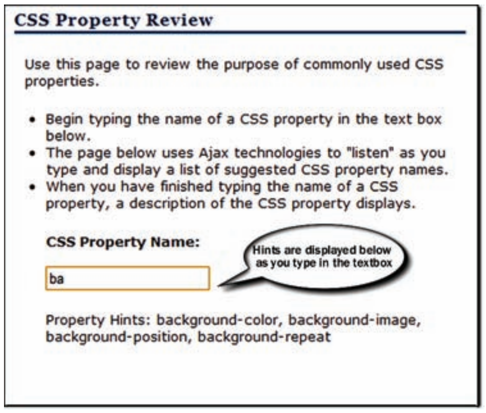
Ajax is not a single technology, but a combination of different technologies. Ajax stands for Asynchronous JavaScript and XML. These technologies are not new, but recently have been used together to provide a better experience for web visitors and create interactive web applications. Jesse James Garrett of Adaptive Path (http://adaptivepath.com/ideas/ajax-new-approach-web-applications) is credited with coining the term "Ajax". The technologies utilized in Ajax are as follows:

- Standards-based HTML and CSS
- The Document Object Model
- XML (and the related XSLT technology)
- Asynchronous data retrieval using XMLHttpRequest
- JavaScript

Some of these technologies may be unfamiliar to you. That's okay at this point in your web development career. You're currently creating a strong foundation in HTML and CSS and may decide to continue your studies in the future and learn additional web technologies. Right now, it's enough to know that these technologies exist and what they can be used for.

Ajax is part of the **Web 2.0** movement—the transition of the Web from isolated static websites to a platform that uses technology to provide rich interfaces and social networking opportunities for people. See "What Is Web 2.0" (http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html) by Tim O'Reilly, who was instrumental in the creation of the term "Web 2.0".

Ajax is a web development technique for creating interactive web applications. Recall the client/server model discussed in Chapters 1 and 9. The browser makes a request to the server (often triggered by clicking a link or a submit button), and the server returns an entire new web page for the browser to display. Ajax pushes more of the processing on the client (browser) using JavaScript and XML and often uses behind-the-scenes asynchronous requests to the server to refresh a portion of the browser display instead of the entire web page. The key is that when using Ajax technology, JavaScript code (which runs on the client computer within the confines of the browser) can communicate directly with the server, exchanging data and modifying parts of the web page display without reloading the entire web page. For example, as soon as a website visitor types a zip code into a form, the value could be looked up in a zip code database and the city/state automatically populated using Ajax—and all of this takes place while the visitor is entering the form information before they click the submit button. The result is that the visitor perceives the web page as being more responsive and has a more interactive experience. See CSS Property Review (http://webdevfoundations.net/css) for an example of Ajax in action. As shown in Figure 11.21, hints are provided as you type the name of a CSS property without refreshing the page.


CSS Property Review

Use this page to review the purpose of commonly used CSS properties.

- Begin typing the name of a CSS property in the text box below.
- The page below uses Ajax technologies to "listen" as you type and display a list of suggested CSS property names.
- When you have finished typing the name of a CSS property, a description of the CSS property displays.

CSS Property Name:

ba

Property Hints: background-color, background-image, background-position, background-repeat

Figure 11.21 Ajax technologies are used to update the page as the visitor types

Developers are using Ajax to support the web applications that are part of Web 2.0, including Flickr photo sharing (http://www.flickr.com) and Google e-mail (http://gmail.google.com).

Ajax Resources

Although you'll probably want to become familiar with scripting before you tackle Ajax, there are many resources and articles available. Some helpful sites are listed here:

- Getting Started with Ajax: http://www.alistapart.com/articles/gettingstartedwithajax
- Ajax Tutorial: http://www.tizag.com/ajaxTutorial

11.10 jQuery

You're already aware that JavaScript is a client-side scripting language that adds interactivity and functionality to web pages. Web developers need to configure the same type of common interactive features (such as slideshows, form validation, and animation) on web pages. One approach is for each person to write their own JavaScript code and test it in a wide variety of browsers and operating systems. As you might guess, this can become quite time consuming. John Resig developed the free, open-source **jQuery** JavaScript library to simplify client-side scripting.

An **application programming interface (API)** is a protocol that allows software components to communicate—interacting and sharing data. The jQuery API can be used to configure many interactive features, including:

- image slideshows
- animation (moving, hiding, fading)
- event handling (mouse movements and mouse clicking)
- document manipulation
- Ajax

Many web developers and designers have found that jQuery is easier to learn and work with than writing their own JavaScript, although a basic understanding of JavaScript is needed to be efficient when using jQuery. An advantage of the jQuery library is its compatibility with all current browsers.

Figure 11.22 jQuery is used to configure a slideshow

jQuery is often used on popular websites, such as Amazon, Google, and Twitter. Because jQuery is an open source library, anyone can extend the jQuery library by writing a new plugin that provides a new or enhanced interactive feature. For example, the jQuery Cycle plugin (http://jquery.malsup.com/cycle) supports a variety of transition effects. Figure 11.22 (see http://webdevfoundations.net/jQuery) shows an example of using jQuery and the Cycle plugin to create an image slideshow. Chapter 14 includes a brief introduction to working with jQuery.

jQuery Resources

There are many free resources and tutorials that can help you learn about jQuery. If you'd like to find out more about jQuery, visit the following resources:

- jQuery: http://jquery.com/
- jQuery Documentation: http://docs.jquery.com/
- How jQuery Works: http://learn.jquery.com/about-jquery/how-jquery-works/

11.11 HTML5 APIs

You've already been introduced to the term, application programming interface (API), which is a protocol that allows software components to communicate—interacting and sharing data. A variety of APIs that are intended to work with HTML5, CSS, and JavaScript are currently under development and in the W3C approval process. We'll explore some of the new APIs in this section, including geolocation, web storage, offline web applications, and two-dimensional drawing.

Geolocation

The **geolocation** API (http://www.w3.org/TR/geolocation-API/) allows your web page visitors to share their geographic location. The browser will first confirm that your visitor wants to share their location. Then, their location may be determined by the IP address, wireless network connection, local cell tower, or GPS hardware depending on the type of device and browser. JavaScript is used to work with the latitude and longitude coordinates provided by the browser. See http://webdevfoundations.net/geo and http://html5demos.com/geo for examples of geolocation in action.

Web Storage

Web developers have traditionally used the JavaScript cookie object to store information in key-value pairs on the client (the website visitor's computer). The **web storage** API (http://www.w3.org/TR/webstorage) provides two new ways to store information on the client side: local storage and session storage. An advantage to using web storage is the increase in the amount of data that can be stored (5MB per domain). The **localStorage** object stores data without an expiration date. The **sessionStorage** object stores data only for the duration of the current browser session. JavaScript is used to work with the values stored in the localStorage and sessionStorage objects. Visit http://webdevfoundations.net/storage and http://html5demos.com/storage for examples of web storage.

Offline Web Applications

An **offline web application** (http://www.w3.org/TR/html5/browsers.html#offline) enables website visitors to view documents and access web applications even when they are not connected to the Internet. You've heard of native applications (apps) for mobile phones. A native app must be built and distributed specifically for the platform it will be used on. If your client would like a native mobile app for both an iPhone and an Android, you'll need to create two different apps! In contrast, a web application (app) can be written with HTML, CSS, and JavaScript and can run in any browser—as long as you are online. An offline web application takes this one step further and stores the HTML, CSS, and JavaScript files on the visitor's device for use offline, even when the device is not connected to the Internet.

The key to an offline web application is the **manifest** file, which is a plain text file having the file extension .appcache. The manifest provides information about the web app in three sections:

- Cache—lists every resource (web pages, CSS, JavaScript, images, etc.) that is associated with the web app
- Fallback—lists fallback files to display when a visitor tries to access a file that is not cached
- Network—lists files that are only available when the visitor has an Internet connection

The web browser reads the list of URLs from the manifest file, downloads the resources, stores them locally, and automatically updates the local files when the resources change. If the visitor tries to access the app without an Internet connection, the web browser will render the local files and follow the fallback and network procedures. Try out a demonstration of an offline web application at http://html5demos.com/offlineapp and http://www.w3schools.com/html/html5_app_cache.asp.

Drawing with the Canvas Element

The HTML5 **canvas element** is a container for dynamic graphics. The canvas element begins with the **`<canvas>`** tag and ends with the `</canvas>` tag. The canvas element is configured with the Canvas 2D Context API (http://www.w3.org/TR/2dcontext2), which provides a way to dynamically draw and transform lines, shapes, images, and text on web pages. If that wasn't enough, the canvas API also provides for interaction with actions taken by the user, like moving the mouse. The canvas offers methods for two-dimensional (2D) bitmap drawing, including lines, strokes, arcs, fills, gradients, images, and text. However, instead of drawing visually using a graphics application, you draw programmatically by writing JavaScript statements. A very basic example of using JavaScript to draw within the canvas element is shown in Figure 11.23 (see chapter11/canvas.html in the student files). The code is

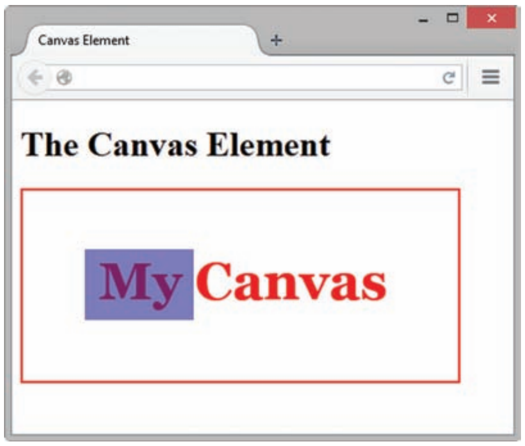


Figure 11.23 The canvas element

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<title>Canvas Element</title>
<meta charset="utf-8">
<style>
canvas { border: 2px solid red; }
</style>
<script type="text/javascript">
function drawMe() {
  var canvas = document.getElementById("myCanvas");
  if (canvas.getContext) {
    var ctx = canvas.getContext("2d");
    ctx.fillStyle = "rgb(255, 0, 0)";
    ctx.font = "bold 3em Georgia";
    ctx.fillText("My Canvas", 70, 100);
    ctx.fillStyle = "rgba(0, 0, 200, 0.50)";
    ctx.fillRect(57, 54, 100, 65);
 }
}
</script>
</head>
<body onload="drawMe()">
<h1>The Canvas Element</h1>
<canvas id="myCanvas" width="400" height="175">
My Canvas</canvas>
</body>
</html>
```

If some of the code looks like a foreign language to you, don't worry: JavaScript IS a different language than CSS and HTML; it has its own syntax and rules. Here's a quick overview of the code:

- The red outline was created by applying CSS to the canvas selector.
- The JavaScript function drawMe() is invoked when the browser loads the page. JavaScript looks for a canvas element assigned to the "myCanvas" id. JavaScript tests for browser support of canvas and, if true, performs the following actions:
  - Sets the canvas context to 2D.
  - Draws the "My Canvas" text.
  - Uses the fillStyle attribute to set the drawing color to red.
  - Uses the font attribute to configure font weight, font size, and font family.
  - Uses the fillText method to specify the text to display, followed by the x-value (pixels in from the left) and y-value (pixels down from the top).
  - Draws the rectangle.
  - Uses the fillStyle attribute to set the drawing color to blue with 50% opacity.
  - Uses the fillRect method to set the x-value (pixels in from the left), y-value (pixels down from the top), width, and height of the rectangle.

The promise of the canvas element is that it can be used to provide interactions as sophisticated as those developed with Adobe Flash. At the time this was written, all modern browsers (Internet Explorer version 9 and later) support the canvas element. Experience virtuoso examples of the canvas element in action at http://www.canvasdemos.com.

HTML5 API Resources

This section provided a brief overview of several of the new HTML5 APIs. Visit the following resources for more information, tutorials, and demos.

- http://html5demos.com
- http://www.creativebloq.com/html5/developer-s-guide-html5-apis-1122923
- http://www.ibm.com/developerworks/library/wa-html5fundamentals3
- http://platform.html5.org

Checkpoint 11.3

1. What are two uses of JavaScript?
2. Describe two technologies used in Ajax.
3. What is the purpose of the HTML5 canvas element?

11.12 Accessibility and Multimedia/Interactivity

Multimedia and interactivity can help to create a compelling, engaging experience for your website visitors. Please keep in mind that not every web visitor will be able to experience these features, so incorporate the following into your website:

Focus on Accessibility

- Provide links to free downloads for the plug-ins used by your multimedia.
- Text descriptions and equivalent content (such as captions) of audio and video will provide access for those with hearing challenges and will also assist visitors using mobile devices or slow Internet connections.
- When you work with multimedia developers and programmers to create Flash animation or Java applets for your site, request features that provide accessibility, such as keyboard access, text descriptions, and so on. If you use Flash or a Java applet for site navigation, be sure that it can be accessed with a keyboard and/or provide plain-text navigation links in the footer section of the pages. Adobe provides useful resources for web developers on their Accessibility web page (http://www.adobe.com/resources/accessibility).
- WCAG 2.0 Guideline 2.3.1 recommends that a web page not contain any item that flashes more than three times per second. This is to prevent optically induced seizures. You may need to work with your multimedia developer to ensure that dynamic effects perform within a safe range.
- If you use JavaScript, be aware that some visitors may have JavaScript disabled or are unable to manipulate the mouse. In order to be in compliance with Section 508

of the Rehabilitation Act, your site must be functional at a basic level, even if your visitor's browser does not support JavaScript. A site using Ajax to redisplay a portion of the browser window may have issues when accessed using an assistive technology or text browser. The importance of testing cannot be overemphasized. W3C has developed ARIA (Accessible Rich Internet Applications), which is a protocol that supports accessibility for scripted and dynamic content, such as the web applications created using Ajax. See WAI-ARIA Overview (http://www.w3.org/WAI/intro/aria.php) for more information about ARIA.

When you design with multimedia/interactivity accessibility in mind, you help those visitors who have physical challenges, as well as those who are using low bandwidth or who may be missing plug-ins on their browser. As a last resort, consider creating a separate text-only version of the page if the multimedia and/or interactivity used on a page cannot comply with accessibility guidelines.

Chapter Summary

This chapter introduced technologies to add media and interactivity to web pages. HTML techniques used to configure sound and video were discussed. Adobe Flash, Java applets, JavaScript, Ajax, and the HTML5 canvas element were introduced. You configured an interactive CSS menu, an interactive CSS image gallery and explored the new CSS3 transition and transform properties. Accessibility and copyright issues related to these technologies were addressed. Visit the textbook website at http://www.webdevfoundations.net for examples, the links listed in this chapter, and updated information.

Key Terms

.aiff
.au
.avi
.class
.flv
.m4a
.m4v
.mid
.mov
.mp3
.mp4
.mpg
.ogg
.ogv
.swf
.wav
.wmv
`<audio>`
`<canvas>`
`<object>`
`<param>`
`<script>`
`<source>`
`<video>`
Ajax
application programming interface (API)
audio element
canvas element
client-side scripting
codec
container
copyright
Creative Commons license
Document Object Model (DOM)
embed element
fair use
Flash
geolocation
helper application
interactivity
Java
Java applets
Java Virtual Machine (JVM)
JavaScript
jQuery
`localStorage`
manifest
media
object-based
object element
offline web application
param element
plug-ins
podcast
`rotate()` transform
RSS feed
sampling rate
`scale()` transform
script element
sessionStorage
source element
`transform` property
`transition` property
video element
Web 2.0
web storage

Review Questions

Multiple Choice

1. Which property provides a way for you to rotate, scale, skew, or move an element?

a. display
b. transition
c. transform
d. relative

2. Which code provides a hyperlink to an audio file called hello.mp3?

a. `<object data="hello.mp3"></object>`
b. `<a href="hello.mp3">Hello (Audio File)</a>`
c. `<object data="hello.mp3"></object>`
d. `<link src="hello.mp3">`

3. What type of files are .wav, .aiff, .mid, and .au?
a. audio files
b. video files
c. both audio and video files
d. image files

4. Which of the following should you do to provide for usability and accessibility?
a. Use video and sound whenever possible.
b. Supply text descriptions of audio and video files that appear on your web pages.
c. Never use audio and video files.
d. none of the above

5. What happens when a browser does not support the video or audio element?
a. The computer crashes.
b. The web page does not display.
c. The fallback content, if it exists, will display.
d. The browser closes.

6. Which of the following statement is true about JavaScript?
a. JavaScript is an object-based scripting language.
b. JavaScript is an easy form of Java.
c. JavaScript was created by Microsoft.
d. JavaScript is an obsolete technology.

7. Which of the following is true of Java applets?
a. Java applets are contained in files with the .class extension.
b. Java applets are not copyrighted.
c. Java applets must be saved in a different folder than web pages.
d. Java is an advanced form of JavaScript.

8. Which of the following is an open-source video codec?
a. Theora
b. Vorbis
c. MP3
d. Flash

9. A file that contains a Flash animation uses the ____________________ file extension.
a. .class
b. .swf
c. .mp3
d. .flash

10. Which of the following can describe Ajax?
a. It is an object-based scripting language.
b. It is the same as Web 2.0.
c. It is a web development technique for creating interactive web applications.
d. It is an obsolete technology.

Fill in the Blank

11. A(n) ____________________ is a protocol that allows software components to communicate—interacting and sharing data.

12. Use of a copyrighted work for purposes such as criticism, reporting, teaching, scholarship, or research is called ____________________.

13. The file extensions .webm, .ogv, and .m4v indicate types of ____________________ files.

14. When displaying a Java applet, the browser invokes the ____________________ to interpret the bytecode into the appropriate machine language.

15. The ____________________ defines every object and element on a web page.

Short Answer

16. List at least two reasons not to use audio or video on a web page.

17. Describe a type of copyright license that empowers the author/artist to grant some, but not all, rights for using his or her work.

Apply Your Knowledge

1. **Predict the Result.** Draw and write a brief description of the web page that will be created with the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head><title>CircleSoft Designs</title>
<meta charset="utf-8">
<style>
body { background-color: #FFFFCC; color: #330000;
       font-family: Arial,Helvetica,sans-serif; }
.wrapper { width: 750px; }
</style>
</head>
<body>
<div class="wrapper">
<h1>CircleSoft Design</h1>
<div><strong>CircleSoft Designs will </strong>
<ul>
  <li>work with you to create a Web presence that fits your
company</li>
  <li>listen to you and answer your questions</li>
  <li>utilize the most appropriate technology for your website</li>
</ul>
<p><a href="podcast.mp3" title="CircleSoft Client
Testimonial">Listen to what our clients say</a>
</p>
</div>
</div>
</body>
</html>
```

2. **Fill in the Missing Code.** This web page should display a Flash media file named slideshow.swf that is 213 pixels wide and 163 pixels high. Some HTML attributes, indicated by "_", are missing. Fill in the missing code.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Fill in the Missing Code</title>
<meta charset="utf-8">
</head>
<body>
<h1>Properties</h1>
<p>Visual Tour of Our Properties</p><br>
<embed type="application/x-shockwave-flash"
       "_"="slideshow.swf" quality="high"
       "_"="213" "_"="163"
      title="slideshow">
</body>
</html>
```

3. **Find the Error.** The purpose of the following web page is to display a video. The video does not display on the Safari browser. Why?

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Find the Error</title>
<meta charset="utf-8">
</head>
<body>
<video controls="controls" width="160" height="150">
      <source src="sparky.webm" type="video/webm">
      <p>You are missing a great video.</p>
</video>
</body>
</html>
```

Hands-On Exercises

1. Write the HTML for a hyperlink to a video called sparky.mov on a web page.
2. Write the HTML to allow a web visitor to control the playback of an audio file named lesson1.mp3.
3. Write the HTML to play a video on a web page. The video source files are prime.m4v, prime.ogv, and prime.webm. The dimensions of the video are 213 pixels wide by 163 pixels high.
4. Write the HTML to add a Flash file called banner.swf to a web page. The effect needs an area that is 468 pixels wide and 60 pixels high.
5. Create a web page about your favorite movie or music CD that plays an audio file (use Windows Sound Recorder or a similar program to record your voice) and includes your review and recommendation. Remember to consider accessibility and provide a transcript of your audio file. Choose to either include the transcript text directly on the page or provide access to it with a hyperlink. Place an e-mail link to yourself on the web page. Save the page as audio11.html.
6. Create a web page about your favorite movie or music CD that plays a video file (use a digital camera, smartphone, or one of the applications listed in this chapter to record yourself) and includes your review and recommendation. Remember to consider accessibility and either provide a transcript of your video file or caption the video file. Place an e-mail link to yourself on the web page. Save the page as video11.html.
7. Visit the textbook website at http://webdevfoundations.net/flashcs5 and follow the instructions to create a Flash logo banner. Use your last name as the company name in your logo banner. Configure a web page to display the logo banner. Place an e-mail link to yourself on the web page. Save the page as flash11.html.

Web Research

1. This chapter mentioned some software applications that can be used to create and edit media files. With those as a starting point, search for more applications on the Web. Create a web page that lists at least five media authoring applications. Organize your web page with a list that provides the name of the software application, the URL,

a brief description, and the price. Place your name in an e-mail link on the web page. Your web page should include a hyperlink to a music audio file. Include an audio file (soundloop.mp3) from this chapter, record your own, or find an appropriate sound file on the Web. Place your name in an e-mail link on the web page.

2. Issues related to copyright law were discussed in this chapter. With the resources provided as a starting point, search for additional information related to copyright law and the Web. Create a web page that provides five helpful facts about copyright law and the Web. Provide the URLs of the websites you used as resources. Place a media console on the page to allow visitors to play an audio file while they read your web page. Include an audio file (soundloop.mp3) from this chapter, record your own, or find an appropriate sound file on the Web. Place your name in an e-mail link on the web page.

3. Choose one method of web interactivity discussed in this chapter: JavaScript, Java applets, Flash, Ajax, or jQuery. Use the resources listed in the chapter as a starting point, but also search the Web for additional resources on the interactivity method you have chosen. Create a web page that lists at least five useful resources along with a brief description of each. Organize your web page with a list that provides the name of the site, the URL, a brief description of what is offered, and a recommended page (such as a tutorial or free script) for each resource. Place your name in an e-mail link on the web page.

4. Choose one method of web interactivity discussed in this chapter: JavaScript, Java applets, Flash, or jQuery. Use the resources listed in the chapter as a starting point, but also search the Web for additional resources on the interactivity method you have chosen. Find either a tutorial or a free download that uses the method of web interactivity you are researching. Create a web page that uses the code or download that you found. Describe the effect and list the URL of the resource on the web page. Place your name in an e-mail link on the web page.

Focus on Web Design

1. There are web design usability and accessibility issues associated with Ajax. Visit the following sites to become aware of these issues:
   - Ajax Usability Mistakes: http://ajaxian.com/archives/ajax-usability-mistakes
   - Ajax and Accessibility: http://www.standards-schmandards.com/2005/ajax-and-accessibility
   - Flash, Ajax, Usability, and SEO: http://www.clickz.com/clickz/column/1702189/flash-ajax-usability-seo

   Write a one-page report that describes Ajax usability issues that web designers should be mindful of. Cite the URLs of the resources you used.

2. In a 2007 interview (http://www.guardian.co.uk/technology/2007/apr/05/adobe.newmedia), Mark Anders, the senior principal scientist at Adobe, recommended Flash as “a great platform for building the next generation of rich Internet applications.” Although Flash is well-supported in standard desktop browsers, mobile devices do not support Flash. In 2011 Adobe dropped development of the Flash player for mobile browsers (http://blogs.adobe.com/conversations/2011/11/flash-focus.html). After you review the sources listed, decide on your own opinion of Flash and when, as a designer, you would recommend its use. Write a one-page paper that persuasively presents your opinion. Cite the URLs of your resources.

WEBSITE CASE STUDY

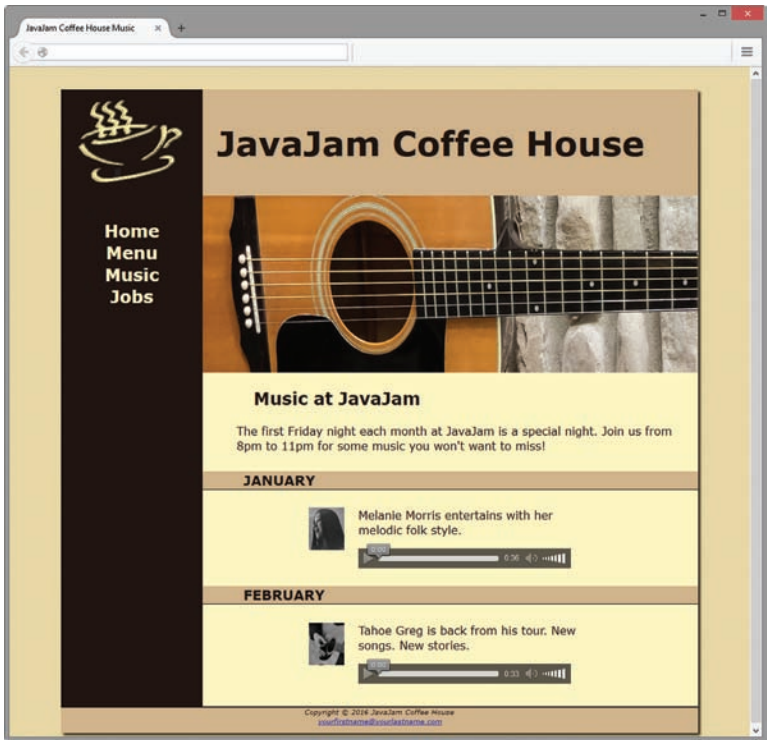
Adding Multimedia

Each of the following case studies continues throughout most of the text. This chapter adds media and interactivity to the websites.

JavaJam Coffee House

See Chapter 2 for an introduction to the JavaJam Coffee House case study. Figure 2.30 shows a site map for the JavaJam website. Use the Chapter 9 JavaJam website as a starting point for this case study. You have three tasks in this case study:

1. Create a new folder for this JavaJam case study.
2. Modify the style sheet (javajam.css) to configure style rules for an audio element.
3. Configure the Music page (music.html) to play audio files. Figure 11.24 shows the Music page with the audio players.



Music at JavaJam

The first Friday night each month at JavaJam is a special night. Join us from 8pm to 11pm for some music you won't want to miss!

Melanie Morris entertains with her melodic folk style.

Tahoe Greg is back from his tour. New songs. New stories.

Copyright © 2016 JavaJam Coffee House
yourfirstname@yourlastname.com

Figure 11.24 The audio element on the Music page

Hands-On Practice Case Study

Task 1: Create a Folder. Create a folder called javajam11. Copy all the files from your Chapter 9 javajam9 folder into the javajam11 folder. Copy melanie.mp3, melanie.ogg, greg.mp3, and greg.ogg from the chapter11/casestudystarters folder in the student files and save them to your javajam11 folder.

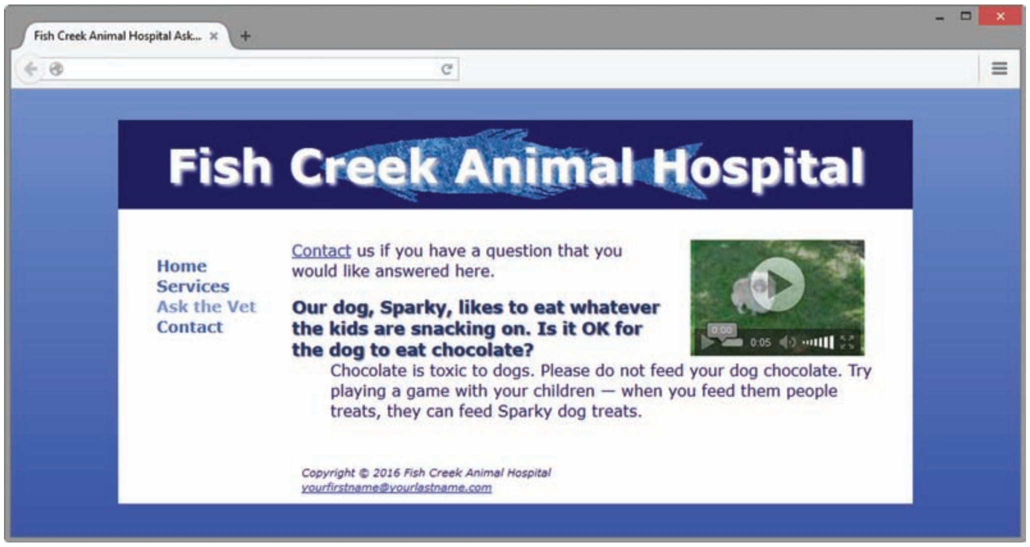
Task 2: Configure the CSS. Modify the external style sheet (javajam.css). Open javajam.css in a text editor. Create an audio element selector with block display and 1em of top margin. Save the javajam.css file.

Task 3: **Update the Music Page.** Open music.html in a text editor. Modify music.html so that two HTML5 audio controls display (see Figure 11.24). Refer to Hands-On Practice 11.3 when you create the audio control. Configure an audio control within the div about Melanie to play the melanie.mp3 or melanie.ogg file and provide a hyperlink to the melanie.mp3 file as a fallback if the audio element is not supported. Configure an audio control within the div about Greg to play the greg.mp3 or greg.ogg file and provide a hyperlink to the greg.mp3 file as a fallback if the audio element is not supported. Save the web page. Check your HTML syntax using the W3C validator (http://validator.w3.org). Correct and retest if necessary. Display the page in different browsers and play the audio files.

Fish Creek Animal Hospital

See Chapter 2 for an introduction to the Fish Creek Animal Hospital case study. Figure 2.34 shows a site map for the Fish Creek website. Use the Chapter 9 Fish Creek website as a starting point for this case study. You have three tasks in this case study:

1. Create a new folder for this Fish Creek case study.
2. Modify fishcreek.css to configure the placement of a video.
3. Add a video to the Ask the Vet page (askvet.html). See Figure 11.25 for a sample screenshot.


Fish Creek Animal Hospital Ask...

Fish Creek Animal Hospital

Home
Services
Ask the Vet
Contact

Contact us if you have a question that you would like answered here.

Our dog, Sparky, likes to eat whatever the kids are snacking on. Is it OK for the dog to eat chocolate?

Chocolate is toxic to dogs. Please do not feed your dog chocolate. Try playing a game with your children — when you feed them people treats, they can feed Sparky dog treats.

Copyright © 2016 Fish Creek Animal Hospital
yourfirstname@yourlastname.com

Figure 11.25 Fish Creek Ask the Vet page with HTML5 video element

Hands-On Practice Case Study

Task 1: Create a Folder. Create a folder called fishcreek11. Copy all the files from your Chapter 9 fishcreek9 folder into the fishcreek11 folder. Copy sparky.webm, sparky.mov, sparky.m4v, sparky.ogv, and sparky.jpg from the chapter11/casestudystarters folder in the student files and save them to your fishcreek11 folder.

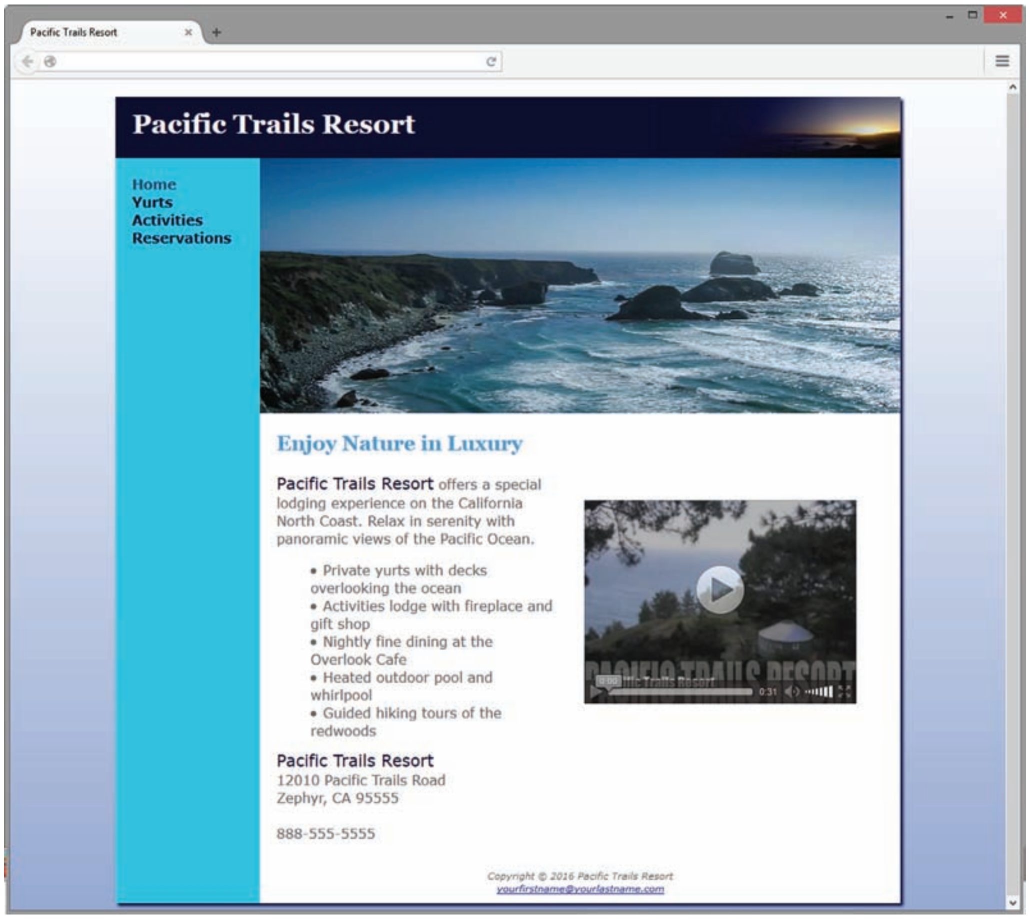
Task 2: Configure the CSS. Modify the external style sheet (fishcreek.css). Open fishcreek.css in a text editor. Edit the CSS and code a new video element selector that floats to the right and has 20 pixel left and right margins. Save the fishcreek.css file.

Task 3: Configure a Video on the Ask the Vet Page. Launch a text editor and edit the Ask the Vet page (askvet.html). Configure the Sparky video (sparky.mov, sparky.m4v, sparky.ogv, sparky.webm, and sparky.jpg files) to display to the right of the paragraph and description list. Use the HTML5 video and source elements. Configure a 120 pixel height, 200 pixel width, and a hyperlink to the sparky.mov file as fallback content in browsers that do not support the video element. Save your web page and test it using several browsers.

Pacific Trails Resort

See Chapter 2 for an introduction to the Pacific Trails Resort case study. Figure 2.38 shows a site map for the Pacific Trails website. Use the Chapter 9 Pacific Trails website as a starting point for this case study. You have three tasks in this case study:

1. Create a new folder for this Pacific Trails case study.
2. Modify pacific.css to configure the placement of a video.
3. Add a video to the home page (index.html). See Figure 11.26 for a sample screenshot.



Enjoy Nature in Luxury

Pacific Trails Resort offers a special lodging experience on the California North Coast. Relax in serenity with panoramic views of the Pacific Ocean.

- Private yurts with decks overlooking the ocean
- Activities lodge with fireplace and gift shop
- Nightly fine dining at the Overlook Cafe
- Heated outdoor pool and whirlpool
- Guided hiking tours of the redwoods

Pacific Trails Resort
12010 Pacific Trails Road
Zephyr, CA 95555

888-555-5555

Copyright © 2016 Pacific Trails Resort
yourfirstname@yourlastname.com

Figure 11.26 Pacific Trails Resort home page

Hands-On Practice Case Study

Task 1: Create a Folder. Create a folder called pacific11. Copy all of the files from your Chapter 9 pacific9 folder into the pacific11 folder. Copy the following files from the chapter11/casestudystarters folder in the student files and save them in your pacific11 folder: pacific.mp4, pacific.ogv, pacific.jpg, and pacific.swf.

Task 2: Configure the CSS. Modify the external style sheet (pacific.css). Open pacific.css in a text editor. Edit the CSS and configure a new video element selector and a new embed element selector with style declarations to float to the right with margin set to 2em. Save the pacific.css file.

Task 3: Configure the Video. Launch a text editor and open the home page (index.html). Code the HTML5 video control below the h2 element and above the paragraph. Configure the video, source, and embed elements to work with the following files: pacific.mp4, pacific.ogv, pacific.swf, and pacific.jpg. The dimensions of the video are 320 pixels wide by 240 pixels high. Save the file. Check your HTML syntax using the W3C validator (http://validator.w3.org). Launch a browser and test your new Home page (index.html). It should look similar to Figure 11.26.

Path of Light Yoga Studio

See Chapter 2 for an introduction to the Path of Light Yoga Studio case study. Figure 2.42 shows a site map for the Path of Light Yoga Studio website. Use the Chapter 9 Path of Light Yoga Studio website as a starting point for this case study. You have three tasks:

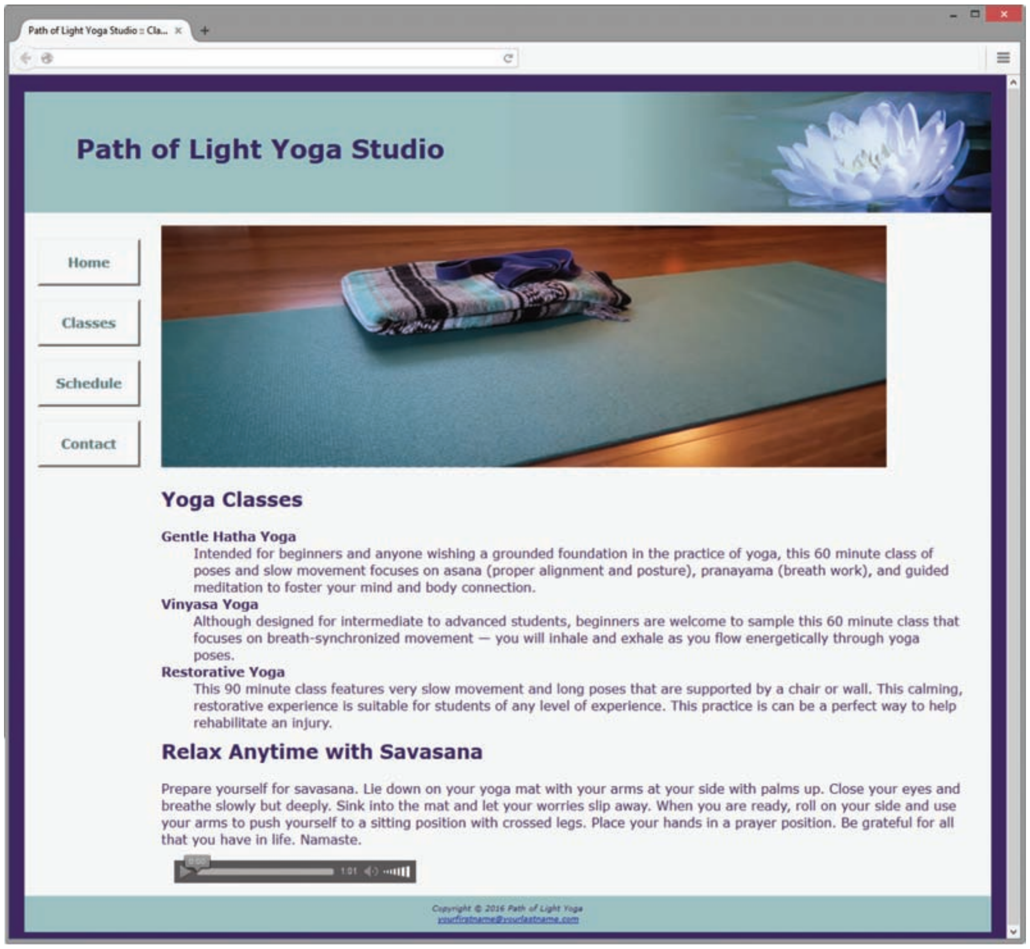
1. Create a new folder for this Path of Light Yoga Studio case study.
2. Modify yoga.css to configure the placement of an audio element.
3. Configure the Classes page (classes.html) to display an audio control.

Hands-On Practice Case Study

Task 1: Create a Folder. Create a folder called yoga11. Copy all of the files from your Chapter 9 yoga9 folder into the yoga11 folder. Copy the savasana.mp3 and savasana.ogg files from the chapter11/casestudystarters folder in the student files and save them in your yoga11 folder.

Task 2: Configure the CSS. Modify the external style sheet (yoga.css). Open yoga.css in a text editor. Edit the CSS and code a new audio element selector with block display and 1em margins. Save the yoga.css file.

Task 3: Configure the Audio. Open the Classes page (classes.html) in a text editor. Modify classes.html so that a heading, a paragraph, and an HTML5 audio control displays below the description list area on the page (see Figure 11.27). Use an h2 element to display the text “Relax Anytime with Savasana”. Add a paragraph that contains the following text: “Prepare yourself for savasana. Lie down on your yoga mat with your arms at your side with palms up. Close your eyes and breathe slowly but deeply. Sink into the mat and let your worries slip away. When you are ready, roll on your side and use your arms to push yourself to a sitting position with crossed legs. Place your hands in a prayer position. Be grateful for all that you have in life. Namaste.”
Refer to Hands-On Practice 11.3 when you create the audio control. Configure the audio and source elements to work with the following files: savasana.mp3 and savasana.ogg. Configure a hyperlink to the savasana.mp3 file as a fallback if the audio element is not supported. Save the file. Check your HTML syntax using the W3C validator (http://validator.w3.org). Correct and retest if necessary. Launch a browser and test your new Classes page (classes.html). It should look similar to Figure 11.27.



Path of Light Yoga Studio

Yoga Classes

Gentle Hatha Yoga
Intended for beginners and anyone wishing a grounded foundation in the practice of yoga, this 60 minute class of poses and slow movement focuses on asana (proper alignment and posture), pranayama (breath work), and guided meditation to foster your mind and body connection.

Vinyasa Yoga
Although designed for intermediate to advanced students, beginners are welcome to sample this 60 minute class that focuses on breath-synchronized movement — you will inhale and exhale as you flow energetically through yoga poses.

Restorative Yoga
This 90 minute class features very slow movement and long poses that are supported by a chair or wall. This calming, restorative experience is suitable for students of any level of experience. This practice is can be a perfect way to help rehabilitate an injury.

Relax Anytime with Savasana

Prepare yourself for savasana. Lie down on your yoga mat with your arms at your side with palms up. Close your eyes and breathe slowly but deeply. Sink into the mat and let your worries slip away. When you are ready, roll on your side and use your arms to push yourself to a sitting position with crossed legs. Place your hands in a prayer position. Be grateful for all that you have in life. Namaste.

Copyright © 2016 Path of Light Yoga
yourfirstname@yourlastname.com

Figure 11.27 The classes page displays an HTML5 audio control

Web Project

See Chapter 5 for an introduction to the Web Project case study. Review the goals of your website and determine whether the use of media or interactivity would add value to your site. If so, you will add either media and/or interactivity to your project site. Check with your instructor regarding the required use of any specific media or technology that supports interactivity in your web project.

Select one or more from the following:

1. Media: Choose one of the examples from the chapter, record your own audio or media file, or search the Web for royalty-free media.
2. Flash: Choose one of the examples from the chapter, create your own SWF file, or search the Web for additional SWF files.
3. CSS image gallery: Create or locate images that relate to your web project. Using Hands-On Practice 11.8 as an example, configure a CSS image gallery.
4. Decide where to apply the media and/or interactive technology to your site. Modify, save the page(s), and test in various browsers.