



# HTML and CSS

8th Edition

## Chapter 6: Links



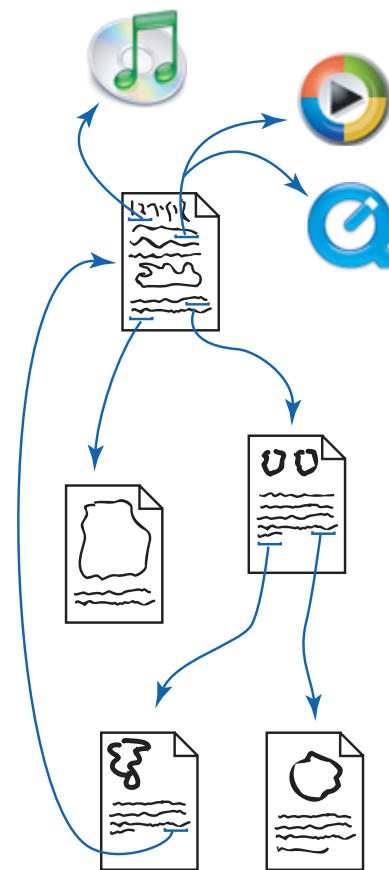
# Objectives

- Explain the importance of links to the web.
- Identify a link's destination and label.
- Create a link to another webpage.
- Use a target attribute to open a link in a new window or tab.
- Create anchors.
- Target specific anchors.



# Links

- Links are the lifeblood of the web.
- Without links, every page would just exist on its own, completely disconnected from all the others.
- Most links connect to other webpages or to specific locations on other webpages.





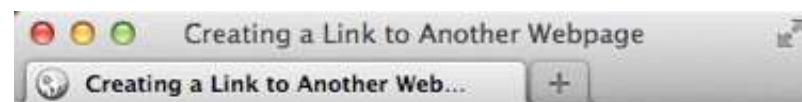
# Destinations and Labels

- A link has two main parts: a destination and a label.
- Destinations:
  - Defined by a URL.
  - Only visible to the visitor in browser's status bar.
- Label:
  - What visitors see or hear and then activate to reach the destination.
  - Can be text, image, or both.
  - Label text appears underlined and in blue.



# Creating a Link to Another Webpage

- If you have more than one webpage, you should create links between them.
- You can also link to pages on other sites, whether your own or someone else's.



## Cookie and Woody



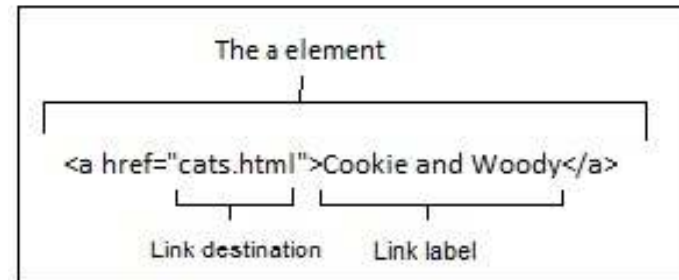
Generally considered the sweetest and yet most independent cats in the [Pioneer Valley](#), Cookie and Woody are consistently underestimated by their humble humans.

[www.fictitious-site.com/pioneer-valley.html](http://www.fictitious-site.com/pioneer-valley.html)



# To Create a Link to Another Webpage

- Type `<a href="page.html">`
  - `page.html` is the URL of the destination webpage.
- Type the label text.
  - You can also label it with an `img` element.
- Type `</a>` to complete the definition of the link.





# Linking Tips

- To link to someone else's website, use an absolute URL with the `http://`, server, full path, and file name.
- To link to the default file for a directory, specify the path, but omit the file name.
- To link to webpages on your site, use a relative URL.
- Use all lowercase letters for URLs, unless page or directory name has uppercase letters.



# How to Use (and Not Use) the Target Attribute

- Target attribute can open a link in a new window or tab, but it is considered bad practice.
  - Should be up to user, not HTML developer.
  - Risks confusing users when results don't open in current window.

## Syntax

```
<a target="_blank|_self|_parent|_top|framename">
```

## Attribute Values

Value	Description
_blank	Opens the linked document in a new window or tab
_self	Opens the linked document in the same frame as it was clicked (this is default)
_parent	Opens the linked document in the parent frame





# Creating and Linking to Anchors

- To link to a specific section of the webpage, create an anchor and reference the anchor in the link.
  - The href value that begins with # anchors to the element with the corresponding id.
  - You may apply an id to any element, as long as any given id exists only once in a page.
  - FAQ pages are most common use for anchor links.



# To Create an Anchor

- Place cursor in start tag of element you wish user to jump to.
- Type `id="anchor-name"`
  - Use anchor-name text to internally identify that section of the webpage .
  - Be sure a space exists between the element's name and the id.
    - Example: `<h2 id="features">`.



# To Create a Link to an Anchor

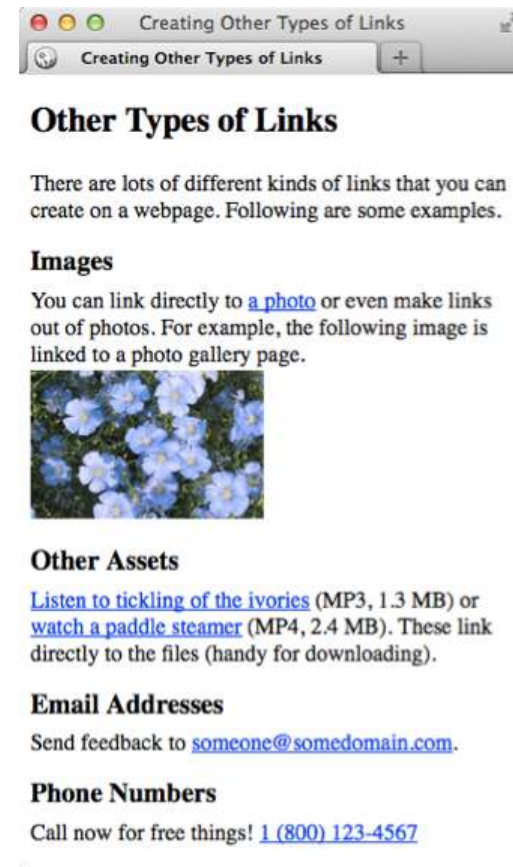
- Type `<a href="#anchor-name">`
  - “anchor-name” is the value of the destination’s id attribute.
- Type the label text.
- Type `</a>` to complete the definition of the link.





# Creating Other Types of Links

- Linking is not limited to other webpages.
- Create a link to any URL.
  - RSS feeds, files for download, and email addresses.
- All links look the same. Unless the link wraps around a photo.





# To Create Other Types of Links

- Type `<a href=“`
- Type the URL.
  - For a link to files on the web, type:  
<http://www.site.com/path/file.ext>.
  - For email address, type: `mailto:name@domain.com`.
  - For a phone number, type: `tel:+` (*not* preceded by `http://`) followed by country code and phone number without dashes.
- Type `">`.
- Type the label for the link.
- Type `</a>`.



# HTML and CSS

8th Edition

## Chapter 15: Lists



# Objectives

- Create ordered and unordered lists.
- Decide where to start list numbering.
- Change numbering of a list item in an ordered list.



# Lists

- **Three list types:**
  - **Ordered list:** ol for the parent, li for each list item
  - **Unordered list:** ul for the parent, li for each list item
  - **Description list:** dl for the parent, dt marks the term to describe, and dd marks the description of the term. This was known as a definition list before HTML5.





# Creating Ordered and Unordered Lists

- **Ordered lists:** Any list of items in which the order is meaningful:
  - Providing step-by-step instructions.
  - Creating outline of larger document.
  - Marking up breadcrumb navigation.
- **Unordered lists:** For when the order of list items isn't tied to the list's meaning.
  - Most common lists on web.
  - Unordered lists have solid round bullets by default.



# Ordered Lists

## Ordered List



## Changing a light bulb

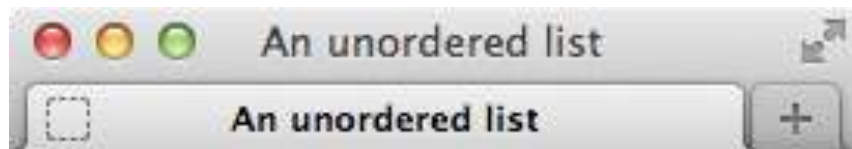
1. Make sure you have unplugged the lamp from the wall socket.
2. Unscrew the old bulb.
3. Get the new bulb out of the package.
4. Check the wattage to make sure it's correct.
5. Screw in the new bulb.
6. Plug in the lamp and turn it on!

```
...  
<body>  
  
<h1>Changing a light bulb</h1>  
  
<ol>  
  <li>Make sure you have unplugged the  
  → lamp from the wall socket.</li>  
  <li>Unscrew the old bulb.</li>  
  <li>Get the new bulb out of the  
  → package.</li>  
  <li>Check the wattage to make sure  
  → it's correct.</li>  
  <li>Screw in the new bulb.</li>  
  <li>Plug in the lamp and turn it  
  → on!</li>  
</ol>  
  
</body>  
</html>
```



# Unordered Lists

## Unordered List



## Product Features

- One-click page layout.
- Spell-checker for 327 languages.
- Image retouching tool.
- Unlimited undos and redos.

```
...  
<body>  
  
<h1>Product Features</h1>  
<ul>  
  <li>One-click page layout.</li>  
  <li>Spell-checker for 327 languages.  
  → </li>  
  <li>Image retouching tool.</li>  
  <li>Unlimited undos and redos.</li>  
</ul>  
  
</body>  
</html>
```



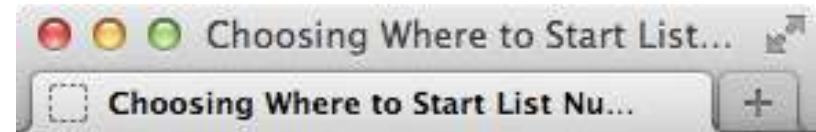
# To Create Lists

- Type `<ol>` for an ordered list or `<ul>` for an unordered list.
  - Optional attributes for ordered lists: start, type, and reversed.
- Type `<li>` to begin the first list item.
  - Optional value attribute for ordered list.
- Add content to be included in list item.
- Type `</li>` to complete each list item.
- Repeat steps for each new list item.
- Type `</ol>` or `</ul>` to complete list.



# Choosing Where to Start List Numbering

- You can start an ordered list's numbering somewhere other than 1.



## **Changing a light bulb (with a few steps missing)**

2. Unscrew the old bulb.
5. Screw in the new bulb.
6. Plug in the lamp and turn it on!



# List Numbering

- **To specify the initial value of an entire list's numbering scheme:**
  - Within the ol start tag, type *start="n"*
    - *n* represents the initial value for the list.
- **To change the numbering of a given list item in an ordered list:**
  - In the desired li item, type *value="n"*
    - *n* represents the value for this list item.

```
...  
<body>  
  
<h1 class="hdg">Changing a light bulb (with a  
→ few steps missing)</h1>  
  
<ol start="2">  
  <li>Unscrew the old bulb.</li>  
  <li value="5">Screw in the new bulb.  
  → </li>  
  <li>Plug in the lamp and turn it on!  
  → </li>  
</ol>  
  
</body>  
</html>
```



# HTML and CSS

8th Edition

## Chapter 18: Tables



# Objectives

- Structure tables.
- Span columns and rows.
- Span a cell across two or more columns.
- Span a cell across two or more rows.





# Tables

- **Tables:** Information presented with one or more column or row headers, along with the tabular data itself.
- **Tabular data:** Includes financial or survey data, event calendars, bus schedules, or television programming schedules.



# Structuring Tables

- **Table element** is made up of rows of cells. Each row (tr) contains header (th) or data (td) cells, or both.
- **Table caption** helps visitors better understand the table. **Optional.**
- **Scope attribute** informs screen readers and assistive devices that a th is the header.  
Recommended.
  - For a table column (when scope="col"),
  - For a table row (when scope="row")
- Browsers can display tables as wide as page space.



# Structuring Tables: thead, tbody, and tfoot

- **thead** marks a row or rows of headers as the table head section.
- **tbody** surrounds all the data rows.
- **tfoot** marks row or rows as table foot section.
  - Use tfoot for column calculations, or to repeat thead headings for a long table, such as in a train schedule.
- The thead, tfoot, and tbody elements don't affect the layout and are not required.
  - Except: tbody required whenever a thead or tfoot is included.
  - Can also target styles with all three of them.



# A Basic Table Example

```

<table>
  <caption> Quarterly Financials for 1962-1964 (in Thousands) </caption>

  <tr>
    <th scope="col">1962</th>
    <th scope="col">1963</th>
    <th scope="col">1964</th>
  </tr>

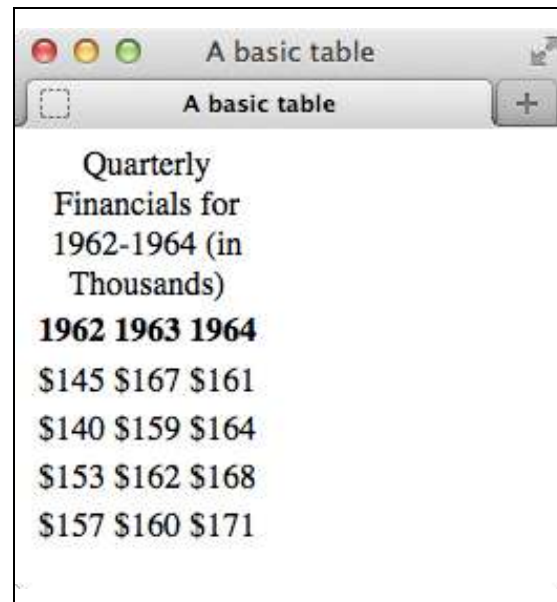
  <tr>
    <td>$145</td>
    <td>$167</td>
    <td>$161</td>
  </tr>

  <tr>
    <td>$140</td>
    <td>$159</td>
    <td>$164</td>
  </tr>

  <tr>
    <td>$153</td>
    <td>$162</td>
    <td>$168</td>
  </tr>

  <tr>
    <td>$157</td>
    <td>$160</td>
    <td>$171</td>
  </tr>
</table>

```



## Default:

- th text is bold.
- th and caption text is centered.
- Table is only as wide as its content.



# To Structure a Table

1. Type `<table>`.
2. If desired, type `<caption>caption content</caption>`.
3. If desired, before first `tr` element of section, type `<thead>`, `<tbody>`, or `<tfoot>`, as appropriate.
4. Type `<tr>` to define beginning of a row.
5. Type `<th scope="scopetype">` to begin a header cell, or type `<td>` to define the beginning of a data cell.
6. Type the contents of the cell.



# To Structure a Table

7. Type `</th>` to complete header cell, or `</td>` to complete a data cell.
8. Repeat steps 5–7 for each cell in the row.
9. Type `</tr>` to complete the row.
10. Repeat steps 4–9 for each row in the section.
11. If you started a section in step 3, close the section with `</thead>`, `</tbody>`, or `</tfoot>`, as appropriate.
12. Repeat steps 3–11 for each section.
13. To finish the table, type `</table>`.



# Spanning Columns and Rows

- **Colspan attribute:**  
Span a th or td across more than one column.
- **Rowspan attribute:**  
Span a th or td across more than one row.
- Number you assign to the attributes specifies the number of cells they span.



## TV Schedule

TIME	MON	TUE	WED
8 pm	Staring Contest	Celebrity Hoedown	
9 pm	Hardy, Har, Har	What's for Lunch?	Screamfest Movie of the Weak
10 pm	Healers, Wheelers & Dealers	It's a Crime	



# Spanning Columns and Rows

```
<tbody> <!-- table body -->
  <tr>
    <th scope="row">8 pm</th>
    <td>Staring Contest</td>
    <td colspan="2">Celebrity
      → Hoedown</td>
  </tr>
  <tr>
    <th scope="row">9 pm</th>
    <td>Hardy, Har, Har</td>
    <td>What's for Lunch?</td>
    <td rowspan="2">Screamfest Movie
      → of the Weak</td>
  </tr>
  <tr>
    <th scope="row">10 pm</th>
    <td>Healers, Wheelers &
      → Dealers</td>
    <td>It's a Crime</td>
  </tr>
</tbody>
```



## TV Schedule

TIME	MON	TUE	WED
8 pm	Staring Contest	Celebrity Hoedown	
9 pm	Hardy, Har, Har	What's for Lunch?	Screamfest Movie of the Weak
10 pm	Healers, Wheelers & Dealers	It's a Crime	





# To Span a Cell Across Two or More Columns

- Type `<th` followed by a space for a header, or `<td` followed by a space for a data cell.
- Type `colspan="n">`.
  - *n* equals number of columns to span.
- Type the cell's contents.
- Type `</th>` or `</td>`, depending on whether you started a header cell or data cell.
- Complete the rest of the table.



# To Span a Cell Across Two or More Rows

- Type `<th` followed by a space for a header, or `<td` followed by a space for a data cell.
- Type `rowspan="n">`.
  - *n* equals the number of rows to span.
- Type the cell's contents.
- Type `</th>` or `</td>`, depending on whether you started a header cell or data cell.
- Complete the rest of the table.



# HTML and CSS

8th Edition

## Chapter 17: Video and Audio



# Objectives

- Employ video and audio in a webpage.
- Manage how media files function in different browsers.
- Use video attributes and audio attributes.
- Include controls and autoplay in a video.
- Add loop, preload, and specify a poster image.
- Establish multiple video and audio sources.
- Create video and audio fallbacks.



# Video, Audio, and Other MultiMedia

- Prior to HTML5, third-party plug-ins were the only method of adding multimedia to your webpages.
  - Adobe Flash Player or Apple’s QuickTime.
- HTML5 introduced native multimedia—where the browser takes care of it all.
  - Not all HTML5-capable browsers support the same video and audio formats.



# Third-Party Plug-ins and Going Native

- **Third-party plug-ins** allowed audio and video before HTML5, but there were problems:
  - Code for embedding a Flash video in one browser didn't necessarily work in another.
  - More importantly, browsers would slow down or crash.
- **Native multimedia** brings improved performance and stability, but there are drawbacks:
  - Media needs to be in more than one format to be playable by HTML5-capable browsers.
  - Embedding audio and video files means that the URLs to the source files are available to anyone who wants them.



# Video File Formats (FYI)

- Three video codecs supported by HTML5:
  - **Ogg Theora** uses either .ogg or .ogv file extension.
    - Supported by Firefox 3.5+, Chrome 4+, Opera 10.5+, and Firefox for Android.
  - **MP4 (H.264)** uses the .mp4 or .m4v file extension.
    - Supported by Safari 3.2+, Chrome 4-?, Internet Explorer 9+, iOS, Android 2.1+, Chrome for Android, Firefox for Android, and Opera Mobile 11+.
  - **WebM** uses the .webm file extension.
    - Supported by Firefox 4+, Chrome 6+, Opera 10.6+, Android 2.3+, Chrome for Android, Firefox for Android, and Opera Mobile 14.



# What's a Codec?

- **Codec** is a computer program that uses a compression algorithm to encode and decode a digital stream of data, making it more suitable for playback.
  - A codec aims to maintain highest audio and video quality it can with a smaller file size.





# Adding A Video to Your Webpage

- **Video element** adds a video to your webpage.
  - Browsers will figure out your video's dimensions when it loads the file and display it at that size, or you can set them yourself.





# To Add a Video to Your Webpage

- Obtain your video resource.
- Type `<video src="myVideo.ext"></video>`
  - *myVideo.ext* is location, name, and extension of video file.





# Exploring Video Attributes

- Attributes that can be use with video element:
  - **Src:** Specifies URL to video file.
  - **Autoplay:** Automatically starts playing video.
  - **Controls:** Adds the browser's default control set to the video.
  - **Muted:** Mutes video's audio (not supported by any browser).
  - **Loop:** Plays the video in a loop.
  - **Poster:** Specifies image file to display when it loads.



# Exploring Video Attributes

- Attributes that can be use with video element:
  - **Width:** Width of the video in pixels.
  - **Height:** Height of the video in pixels.
  - **Preload:** Hints to browser how much video to load. Takes three different values:
    - **None:** Doesn't load anything.
    - **Metadata:** Loads only the video's metadata (e.g., length and dimensions).
    - **Auto:** Lets the browser decide what to do (this is the default setting).



# Adding Controls and Autoplay to Your Video

- **Controls attribute** informs the browser to add a set of default controls to the video.
- Each browser has its own default controls.
- Boolean attributes, such as controls, don't need to have a value specified for them.





# Video Controls in IE10 and Opera





# When a Video Format Isn't Supported

- If the browser you use doesn't support the video file format you're using, it will show its controls bar and either an empty, white rectangle or the poster image.
- Safari does not support WebM.





# To Add Controls and Autoplay to a Video

- **To add controls to a video:**
  - Type `<video src="my-video.ext" controls></video>`
    - *my-video.ext* points to your video file.
- **To add autoplay to a video:**
  - Type `<video src="my-video.ext" autoplay controls></video>`
    - *my-video.ext* points to your video file.





# Looping a Video and Specifying a Poster Image

- **Loop attribute** is use with autoplay to set video to play automatically and continuously until stopped.
- **Poster image** specifies image to display on loading.





# To Loop a Video and Specify a Poster Image

- **To add autoplay and loop a video:**
  - Type `<video src="my-video.ext" autoplay loop></video>`, where *my-video.ext* points to your video file.
- **To specify a poster image for a video:**
  - Type `<video src="my-video.ext" controls poster="my-poster.jpg"></video>`, where *my-video.ext* points to your video file and *my-poster.jpg* points to the image that you want to use as the poster image.



# Preventing a Video from Preloading

- **To instruct the browser to not preload a video:**
  - Type `<video src="my-video.ext" preload="none" controls></video>`
    - *my-video.ext* points to your video file.





# Using Video with Multiple Sources and a Text Fallback

- To support all HTML5-capable browsers, supply video in at least two different formats.
- **Source element** allows you to define more than one source for a media element.
  - Browsers will load the first file format referenced in a source element that they support and ignore the rest.



# To Specify Two Different Video Sources with a Fallback

- Obtain your video sources (two this time).
- Type `<video controls>` to open video element.
- Type `<source src="my-video.mp4" type="video/mp4">`
  - *my-video.mp4* points to your MP4 video source file.
- Type `<source src="my-video.webm" type="video/webm">`
  - *my-video.webm* points to your WebM video source file.
- Create a fallback link or message for older browsers.
- Type `</video>` to close the video element.



# Audio File Formats (FYI)

- Main audio codecs:
  - **Ogg Vorbis**: .ogg file extension. Supported by Firefox 3.5+, Chrome 5+, and Opera 10.5+.
  - **MP3**: mp3 file extension. Supported by Safari 5+, Chrome 6+, IE9+, and iOS.
  - **WAV**: .wav file extension. Supported by Firefox 3.6+, Safari 5+, Chrome 8+, and Opera 10.5+.
  - **AAC**: .aac file extension. Supported by Safari 3+, IE 9+, iOS 3+, and Android 2+.
  - **MP4**: .mp4 file extension. Supported by Safari 3+, Chrome 5+, IE 9+, iOS 3+, and Android 2+.
  - **Opus**: .opus file extension. Only supported by Firefox.



# Adding an Audio File with Controls to Your Webpage

- **Audio element** adds audio controls to your webpage.
- **To add an audio file with controls to your webpage:**
  - Obtain your audio file.
  - Type `<audio src="my-audio.ext" controls></audio>`
    - *my-audio.ext* is location, name, and extension of the audio file.



# Audio Controls in Different Browsers

Firefox

Internet Explorer 10







# Exploring Audio Attributes

- **Src:** Specifies URL to audio file.
- **Autoplay:** Automatically starts audio.
- **Controls:** Adds browser's default control set to audio.
- **Muted:** Mutes audio.
- **Loop:** Plays audio in a loop.
- **Preload:** Hints how much audio to load. Has three different values: None, Metadata, and Auto.



# Autoplaying, Looping, and Preloading Audio

- **Autoplay attribute** makes an audio file start playing when the page loads.
- **Loop attribute** indicates that you want the audio to play in a loop.
- Request that the browser preload the audio file in different ways by using the preload attribute values.



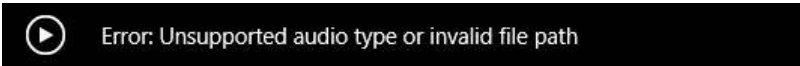
# To Loop and Preload Audio

- **To play an audio file in a loop:**
  - Type `<audio src="my-audio.ext" loop controls></audio>`
    - *my-audio.ext* points to your audio file.
- **To ask the browser to preload only the audio's metadata:**
  - Type `<audio src="my-audio.ext" preload="metadata" controls></audio>`
    - *my-audio.ext* points to your audio file.



# When an Audio Format Isn't Supported

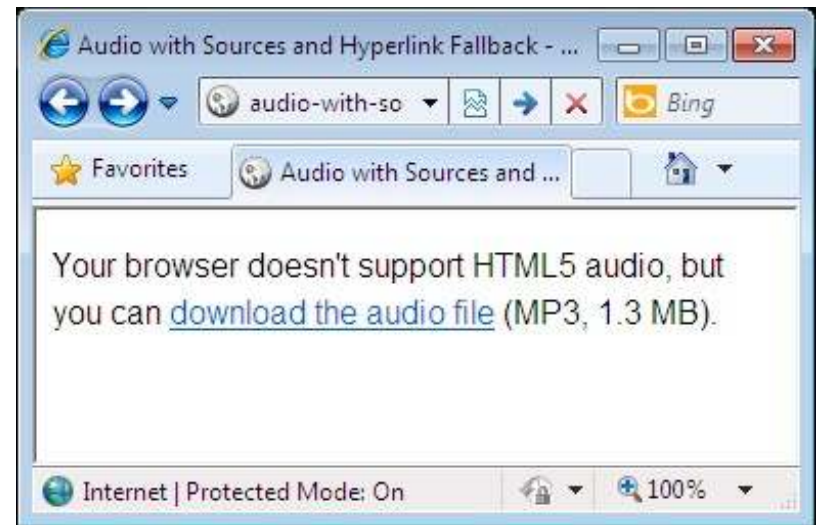
- IE10 when it doesn't support the audio file format you specify in src.





# Providing Multiple Audio Sources with Fallback

- **Source element** provides audio in more than one format.
  - Browsers that don't support the audio element display the fallback.
- **Type Attribute** helps the browser decide if it can play the audio file.





# To Specify Two Different Audio Sources with Fallback

- Obtain your audio files.
- Type `<audio controls>` to open the audio element.
- Type `<source src="my-audio.ogg" type="audio/ogg">`
- Type `<source src="my-audio.mp3" type="audio/mp3">`
- (Optionally, create fallback message or audio download link for browsers that don't support HTML5 audio.
- Type `</audio>` to close the audio element.