

ISTA 230



HTML & CSS - Random Bits

HTML

For the sake of completion, we'll be covering a mix of miscellaneous HTML, CSS, and web-related concepts that don't fit well elsewhere in the course material.

Blockquotes

The block quote element allows us to provide a more semantically specific HTML elements for long quotes. The opening blockquote element can also have a cite attribute, specifying where the quote was taken from.

HTML

```
<p>In 1949, John Von Neumann said:</p>
<blockquote cite="http://www.quotationspage.com/quote/22919.html">
  <p class='quote'>
    It would appear ...
  </p>
</blockquote>
<p>62 years later, it would appear he was correct...</p>
```

Browser

In 1949, John Von Neumann said:

It would appear that we have reached the limits of what it is possible to achieve with computer technology, although one should be careful with such statements, as they tend to sound pretty silly in 5 years.

62 years later, it would appear he was correct; his statement does sound silly.

Blockquotes can only contain block elements as its children and is typically indented by the browser automatically.

HTML

Quotes

The <q> element allows us to provide a semantically correct element for shorter quotes (i.e., one sentence or less). Note that while I did not specify quotation marks around the quote, the browser automatically does so when I use the <q> element.

HTML

```
<p>  
  In 1948, the chairman of the board for IBM said,  
  <q>I think there's a world market for about 5 computers.</q>  
  He was wrong.  
</p>
```

Browser

In 1948, the chairman of the board for IBM said, "I think there's a world market for about 5 computers."
He was wrong.

HTML

Citations

The `<cite>` element can be used to define the **title** of a work (e.g., a book, a movie, etc.). This element is italicized in most browsers. Note that the author's name (or additional information about the work) should NOT be included in the `<cite>` element.

HTML

```
<p>  
  <cite>The Walking Dead</cite>, by Robert Kirkman  
</p>
```

Browser

The Walking Dead, by Robert Kirkman

HTML

Subscript & Superscript

The <sub> and <sup> elements allow us to create subscript and superscript text respectively.

Subscript text is displayed below the normal text baseline. The font size for subscript text is often reduced to approximately 80% on some browsers as well. Subscript is most often used in mathematic and scientific formulas.

HTML

```
<p>
  The alchemist is dead and gone <br />
  He doth live no more <br />
  For what he though was H<sub>2</sub></sub>0 <br />
  Was H<sub>2</sub></sub>S<sub>O</sub><sub>4</sub>
</p>
```

Browser

The alchemist is dead and gone
He doth live no more
For what he though was H₂O
Was H₂SO₄

Superscript text is displayed above the normal text baseline. The font size for subscript text is often reduced to approximately 80% on some browsers as well. Superscript is used in mathematic and scientific formulas, as well as for ordinal suffixes (e.g., 1st, 2nd, etc.). It can also be used when creating footnotes/references.

HTML

```
<p>
  His grandmother lives on the 2<sup>nd</sup> floor,
  4<sup>th</sup> door on the left.
</p>
```

Browser

His grandmother lives on the 2nd floor, 4th door on the left.

HTML

Secondary Objects

Oftentimes, we want to include the contents of another file in our webpage. Most often, this is an image but there are other types of "secondary objects" that we can include as well.

Introduced in HTML 1.0, the `` element was the earliest implementation of embedding a secondary object in a webpage.

In HTML 3.2, the `<applet>` element was introduced. This allowed users to include Java applets in their webpages. While the idea of including third-party applications in a webpage was appealing, the implementation left something to be desired. The `<applet>` element was very specific in what it could include. It also had a number of accessibility shortcomings.

HTML

Objects

To resolve these issues, the W3C introduced the `<object>` element in HTML 4.0. This element was very generic in nature and was intended to allow users to embed images, audio files, video files, third-party applications (e.g., Java applets, Adobe Flash applications), and even other HTML files.

While the intentions were good, there is inconsistent support between browsers. Specifically, using `object` to include images and other HTML files will have mixed results.

HTML

Objects - Images

An object can be created as follows:

HTML

```
<object type="image/png" data="images/pic.png">  
</object>
```

The above code will include an image using the <object> element. Note the two required attributes: type (the type of file being embedded) and data (the URL to the file).

The above example will work in some browsers but others will not support embedding images this way. For cross-browser compatibility, it is best to just use the element for your images.

HTML

Objects - MP3 / Audio

The <object> element can be used to embed audio. If the browser has the ability to play the file, it will automatically display a set of controls for the user (play/pause button, etc.). As the web designer, you do not have to specify how you want these controls to look but you also do not have the option to change how they look.

HTML

```
<object type="audio/mpeg" data="music/song.mp3">  
</object>
```



HTML

Objects - Video

The <object> element can also be used to embed video. As with audio, if the browser has the ability to play the file, it will automatically display a set of controls for the user (play/pause button, etc.). As the web designer, you do not have to specify how you want these controls to look but you also do not have the option to change how they look.

HTML

```
<object type="video/mp4" data="videos/episode.mp4">  
</object>
```



HTML

Objects - Adobe Flash

Probably the most popular use of the <object> element is to embed Adobe Flash files.

HTML

```
<object type="application/x-shockwave-flash" data="flash/file.swf">  
  <param name="movie" value="flash/file.swf" />  
  <param name="wmode" value="transparent" />  
</object>
```

You'll note that in addition to our <object> element, we've also included two child <param> elements for our Flash object. The first provides greater compatibility with older browsers while the second tells the browser that the "wmode" for the application should be transparent. This causes the Flash application to be transparent, allowing background colors and images from the webpage to be seen "underneath" the <object>.

We do have the ability to resize the <object> element using CSS. Depending on the type of file we're embedding, this can have mixed results. However with most Flash applications, this will simply change the size of the application.

CSS

```
object  
{  
  width:200px;  
  height:200px;  
}
```

HTML

Objects - Other Websites - iFrame

We can technically use <object> to embed other HTML files in our page.

HTML

```
<object type="text/html" data="http://www.ista230.com">
</object>
```

While technically possible, this technique is not supported by most browsers. Instead, a better approach is to use the <iframe> element.

HTML

```
<iframe src="http://www.ista230.com"></iframe>
```

While well supported by almost every browser, the <iframe> is **NOT** valid XHTML 1.0 Strict. It is valid in HTML 5 though.

HTML

Objects - Accessibility & Nested Elements

While somewhat limited by cross-browser support, the <object> element was good in that it provided greater accessibility options for embedded content. Rather than just having an 'alt' attribute like the tag, we can nest 'fallback' elements within our object.

For example, we could provide an embedded video for 'How to make coffee'. If the user's browser didn't have the ability to display that video, it would try to render the children of that element instead. The first child could be another object containing an image representing the video's contents. If neither of those were supported by the user's browser (or if the user was using a screen reader), then the nested ordered list would be displayed next.

HTML

```
<object type="video/mpeg" data="howToMakeCoffee.mpg">
  <object type="image/png" data="coffee.png">
    <ol>
      <li>Place coffee grounds in filter</li>
      <li>Pour water into coffee maker</li>
      <li>Press 'On'</li>
    </ol>
  </object>
</object>
```

When using Adobe Flash, this technique is often used to include an <a> tag linking to where the user can download the Adobe Flash Player.

CSS

Cursors

The browser's internal CSS file decides how the user's cursor should appear when hovered over different elements. However, we have the ability to change this if we want to. We can do so using the 'cursor' property.

By default, the browser uses 'auto' as the value for the cursor property. For example:

CSS

```
div
{
  background-color:#333;
  cursor:auto;
}
```

If we wanted to trick our users into thinking all of our <div> elements were links, we could change the cursor to a 'pointer' by using the following property:

CSS

```
div
{
  background-color:#333;
  cursor:pointer;
}
```

CSS

Cursors

Below are all of the pre-defined properties for cursor:

- auto
- crosshair
- default
- X-resize
n s e w ne nw se sw
- help
- move
- pointer
- progress
- text
- wait

CSS

Cursors

In addition to the pre-defined cursor properties, we can also specify a custom cursor by pointing to a 'cur' file. We will want to specify a 'fallback' cursor property in addition to our custom cursor in case the file becomes unavailable or doesn't load quickly.

CSS

```
div
{
  background-color:#999;
  cursor:url('puppy.cur'), default;
}
```

While you have the ability to provide custom cursors, it is not usually a good idea. That said, you can use your best judgement to determine if your situation calls for it. If you do need to change the cursor, my advice would be to pick a cursor that introduces subtle changes.

CSS

cursor Property

CSS

```
div { cursor: auto; }
```

```
div { cursor: pointer; }
```

```
/*  
 * Specifies the type of cursor to be displayed  
 * when the mouse is over a given element. Supported  
 * values include auto, crosshair, default, n-resize,  
 * s-resize, e-resize, w-resize, ne-resize, nw-resize,  
 * se-resize, sw-resize, help, move, pointer, progress,  
 * text, wait, or a URL value.  
 *  
 *  
 * Default Value: auto  
 * Inherited: Yes  
 *  
 */
```

CSS

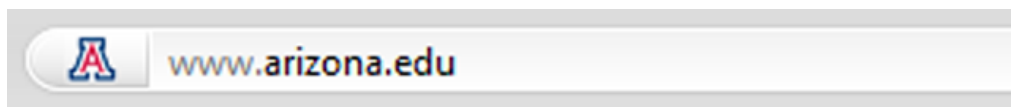
Favicon

Favicons are small icons that can be used to provide users with a small visual identifier for your site. This icon is used in the URL bar on many browser, often used in tabs on browsers that support them, and sometimes is used when 'bookmarking' a website.

Favicons can be specified using a <link> element within the <head> of your webpage.

HTML

```
<head>
<title>My Web Page</title>
...
<link rel="shortcut icon" href="/favicon.ico" />
</head>
```



For best results, it is recommended to use an icon that is 16 × 16 pixels.