

CSS Notes

Read a CSS Tutorial

All of the web sites with HTML tutorials listed in the [Lesson 1 Notes](#) also have tutorials on CSS. Choose one of them to read about CSS before going on to the rest of Lesson 2. Here are links to the CSS tutorials:

- [HTML Dog](#)
- [tizag.com](#)
- [sitepoint.com](#) Combines HTML and CSS in the beginning tutorial.
- [W3 Schools](#)

There are a number of relevant books available on UVU's subscription to Safari Books Online. (You can get to Safari Books Online by going to www.uvu.edu/library and then clicking on the **eBooks** link in the **Quick Links** section. If you are off-campus you will need to put in your UVID and password.)

Ways to refer to styles

There are three basic ways to define and refer to styles: inline styles, embedded style sheets, and external style sheets.

Inline styles refers to using style attributes in elements. They're easy to add to an HTML document, but updating and maintaining them is tedious and is an error-prone process.

Embedded or internal style sheets are made by using a style element in the head section of the document. Updating styles in an embedded style sheet is easier than updating inline styles because you can change the style in one place, and all the elements that make use of that style will be updated. An embedded style sheet can be useful for making a self-contained HTML document that is only a single file.

External style sheets are very useful for sharing styles between documents. You can use a link element in the head section of the document to refer to a file that has styles defined in it. A disadvantage of external style sheets is that they require the browser to send an additional request to the server (besides the original request for the HTML document). However, that's a minor disadvantage.

Advantages of CSS

One of the biggest advantages of CSS is that you can define a style in one place and use it in many places. If you want to change the appearance of a number of elements that share a style, you only have to change the style definition. That makes it easier to maintain documents and easier to give various elements a consistent appearance. This advantage is relevant for embedded and external style sheets, but not for inline styles.

Another advantage of CSS is that there are more options for changing the appearance document elements than there are in HTML. This advantage is relevant for any of the three ways of referring to styles.

Finally, with an external stylesheet, the HTML document and the CSS styles are separate and can be changed independently. That means that someone who knows how to edit HTML documents but is not a designer can easily modify the contents of a document without fear of causing problems with the appearance. Likewise a designer can easily change the appearance without changing the contents.

CSS Properties

There are a lot of properties in CSS (especially CSS3). Here are some of the properties that are most relevant for this class. You should be sure to understand them.

- font-family, font-size
- background-color, background-image
- color
- height, width
- box model properties: padding, border, margin (including the -right, -top, -left, and -bottom properties)
- float, clear, display, visibility
- position, left, right, top, z-index
- text-align, text-decoration, text-indent, vertical-align

Make sure you understand the concept of the box model. Box model properties are important for Web page design and for using JavaScript with HTML documents.

Here are a few links to tutorials about the box model:

- http://www.w3schools.com/css/css_boxmodel.asp W3 Schools
- <http://learn.shayhowe.com/html-css/box-model>
- <http://www.addedbytes.com/articles/for-beginners/the-box-model-for-beginners/>

ID Selectors

The authors of a textbook we used to use in this class recommended not using ID selectors in documents. Their reasoning was that IDs should be unique, so there's no point in referring to an ID with a selector and you should use a different kind of selector or an inline style.

The point that the authors seem to have missed is that ID selectors can be used to apply a style to all elements within the element with the selected ID. Take a look at the following sample page:

[idSelector.html](#)

Links on that page appear in three different styles: green and sans-serif, gray and serif, and in the default style. Without using ID selectors or classes, there would be no easy way to have one set of links appear in one style, and another set in a different style. You could get the same effect by using classes, but each tag would have to have a class attribute, so the HTML would be longer and more difficult to maintain.

Dynamic HTML

Dynamic HTML (DHTML) refers to Web pages where CSS and JavaScript are used together to create visual effects such as moving an element or changing colors of elements. We will look at DHTML and animation later in the term. One of the most important CSS properties for DHTML is the `position` property, so be sure to understand how it works. In particular, make sure you understand the difference between the four values of the `position` element: `static`, `absolute`, `fixed`, and `relative`.

CSS Positioning Video

Here is a link to a video about using CSS to position HTML elements:

CSS Positioning

CS 3550
Dr. Brian Durney

- [Video \(6:29\)](#)
- [PowerPoint slides](#)
- [Script](#)

The video shows screen shots of browser windows with some HTML documents in them to demonstrate different kinds of positioning. Below are links to the documents so that you can see how they work in a browser window. (Each link opens in a new window.)

[none.html](#) No CSS positioning is used with the L image.

[static.html](#) The L image is positioned using **static** positioning.

[relative.html](#) The L image is positioned using **relative** positioning.

[absolute.html](#) The L image is positioned using **absolute** positioning. The div that contains the L image is *not* positioned with CSS, so the L image is positioned with respect to the body of the document.

[absolute2.html](#) The L image is positioned using **absolute** positioning. The div that contains the L image *is* positioned with CSS, so the L image is positioned with respect to that div.

[fixed.html](#) The L image is positioned using **fixed** positioning. Scroll the window to see how fixed positioning differs from absolute positioning.