

Topic Notes: Cascading Style Sheets

With so many style attributes and the ability to specify them on nearly every HTML element, we can see how it can be quite a bit of work to keep track of and manage all of the style attributes in all but the simplest documents.

Suppose we want to have all of the level 3 headers in a particular font and color. And our document has dozens or even hundreds of these. We would need to remember to add the appropriate style to each `<h3>` tag. As the document changes over time, new headers may be added and those would also need to have the same style attribute.

See `manyh3s.html` example.

It's bad enough to have to create all of this in the first place, but now what if we decide to change the style. Perhaps we want a different color or font size applied to all instances of the level 3 headers. That involves some tedious editing.

The style Element

Fortunately, there are mechanisms in HTML to help in this situation. The first we will consider is the `<style> ... </style>` element.

The `<style>` element is always placed in the `<head>` element of an HTML document. An example:

```
<style type="text/css">
h2 {color:red;}
</style>
```

Here, all `<h2>` elements in the document will have their style property for `color` set to `red`, unless that property is overridden by a `style` specified within an individual `<h2>` tag.

See `manyh3sbetter.html` example.

Now, suppose we decide we want to change something about the style for all of our level 3 headers in the above document. We change it in one place and all instances are changed.

We can do this for any of our HTML elements.

Cascading Style Sheets

The `<style>` element helps us to manage styles within an HTML document, but what if we want to maintain a consistent set of styles across a web site consisting of many HTML documents? We

could insert the same `<style>` element in the header of each document. However, this would mean that style information would need to be stored in each document – again we would need to remember to insert it into all of the site’s documents, and update all of the documents when a style change is made.

Another mechanism comes to the rescue here – we place our style information in a separate document called a *style sheet*, which we then link from all of the documents that are intended to use those styles.

A style sheet is linked to a document using the `<link>` tag, which is also always placed within the document’s `<head>` element.

For example:

```
<link rel="stylesheet" type="text/css" href="h3style.css" />
```

This will load the styles defined in the file `h3style.css` in the same directory on the server as the document. As with anchor tags, the destination in the `href` attribute can be a full URL.

See `manyh3scss.html` example.

More Control

So we have a variety of ways we can assign style attributes to our HTML elements. We now consider some ways to make use of them.

First, what if we do not want to change the style of *all* elements of a given type? Perhaps there are two types of level 3 headers, some of which should be displayed in one style, the remainder of which should be displayed in another. The straightforward style definition we saw earlier applies to all elements of the type.

We can define a `class` for each group of elements that should share a style:

```
<h3 class="onekind">Header Type 1</h3>  
<h3 class="anotherkind">Header Type 2</h3>
```

and in our style definition, we define the styles for `<h3>`’s of classes `onekind` and `anotherkind` as follows:

```
.onekind, h3 {  
    font-size: 18pt;  
    ....  
}  
.anotherkind, h3 {  
    font-size: 15pt;  
    ....  
}
```