How-To: Insert CSS into webpages

For some of you, this may sound like a real lame article. But you ought to give it a chance because I don't write about stuff you'll never use! I avoid marginal, nerd centric ivory tower blather like a cat avoids water.

I've read hundreds (sadly, this is true) of those 2-inch thick nerd books (I don't get out much) and I can't stand it when people waste paper on that kind of generally useless stuff. So read on, eager student, and you will be well on your way to building your own killer web sites! ☺

The wise people who created CSS came up with 3 basic ways for you to use CSS in your web pages:

1. With an external file that you link to in your web page:

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

or using the import method:

```
<style type="text/css" media="all">
   @import "myCSSfile.css";
</style>
```

Why use the import method versus the link method when using external style sheets? Use the import method if you want to support really old browsers like Netscape 4.0.

Let me explain: Netscape can't handle most CSS beyond font settings and colors, if it finds any other types of CSS it could cause good old Netscape 4 to crash in some occasions or at the very least mangle your page.

Netscape 4 does not understand the @import method of linking to a style sheet like the newer browsers can, so Netscape just ignores it. You can use this to hide the fancy CCS code in the external style sheet by using the @import method so all the good browsers can reference it while keeping Netscape 4 out of the picture.

Netscape 4.0 is pretty much dead (that is really a good thing) so I personally don't worry much about it. But for some of you, it may be of concern so I thought it should be mentioned.

2. By creating a CSS block in the web page itself; typically inserted at the top of the web page in between the <head> and </head> tags:

```
<head>
   <style type="text/css">
      p { padding-bottom: 12px; }
   </style>
</head>
```

3. By inserting the CSS code right on the tag itself:

```
<p style="padding-bottom:12px;">Your Text</p>
```

So some of you may be asking why have the 3 methods of including the CSS in a web page? The answer is: flexibility and laziness! Ok I'm kidding about the laziness, replace it with 'precision'. So what the heck does that mean?

I think the easiest way to explain to you what's going on, is by giving you real examples that demonstrate the differences. Wait a second, don't fall asleep … the examples are short and I think that once you finish, you will see how easy it really is!

Another reason that you want to continue reading this article is that you will gain a good understanding of some fundamental (and practical) CSS principles – remember that the difference between people who are really good at what they do, and those who are not so good, is in the mastery of the basics. Lets get down on it!

Method 1: Create a separate CSS file and link it to your web page(s)

The heading for this part of the article hints at why you would want to go through the trouble of creating a separate CSS file instead of just typing in the CSS code in the web page itself. Can you guess? Keep trying … times up! Did you get it? I could quote you some nerd centric description that describes the advantage; the problem is that only nerds who already know would understand!

In a nutshell: by keeping the CSS code in its own file, you can link that CSS file to as many web pages as you want. This has two major advantages:

1. You will have much less code in all your HTML pages – makes the pages neater and easier to manage and makes the web pages a little faster on the download. (Although this point is really minor in most cases, and is really over blown in my opinion by some people)
2. It can potentially reduce the amount of work you have to do in a big way. Why you ask? Simple; lets say you have 50 web pages where you've set the text to be black and the headline text (text in between the <h3> tags for example) to blue. Then one day you decide you want to change the color of the text. Since the CSS that controls the text color for the 50 pages is in one CSS file, you can easily change the color of your text in all 50 pages by changing one line in the CSS file!

If on the other hand you had decided to include all your font color information in each page, you would have had to change all 50 pages. This would have been even worse if you had been using font tags, or CSS right on each tag, you would have to change the color settings/code on all the <p> and <h3> tags in all 50 pages! I can tell you from experience that that sucks big time!

The rule: If you are going to have more than one web page with the same stylistic properties (that look the same in some way) you should create a separate CSS file and link your web pages to it.

Method 2: Create a CSS block in the web page itself

The Rule: Use this method if you want to override the CSS you have in a linked CSS file or if you only have a one-page web site.

Now that we covered the first method of putting all your CSS code in a separate file and linking to it, the other methods are easy to describe.

CSS stands for (is the acronym for): 'Cascading Style Sheets.' I think the words 'style sheets' in CSS are self-describing … we know what 'style' in style sheets mean. But what is the meaning of 'cascading' in CSS?

The cascading effect in CSS

The word 'cascading' in CSS describes a cascading mechanism; that is to say that the CSS code on the page itself will override the CSS code in a separate linked file. And subsequently, CSS declared 'inline' on the tag itself would override all other CSS.

So let's look a practical example; let's say you have a web site with 50 pages where the layout and fonts are the same on all 50 pages. Wisely you put the CSS information that sets the layout and font choices in a separate style sheet, but for a particular page you need to change the color of some of the text and add a border around a paragraph. This is a perfect example where you might want to place a little CSS in the page itself since the color and border will be unique to that page. Is this all sinking in? ☺

Method 3: Embed the CSS right on the tags themselves (called inline CSS)

The Rule: Use this method on a unique element/tag that you want to affect with CSS.

An example can be with a special heading on the page where you want to have a little more padding than you typically do for a heading. Instead of creating a class elsewhere that will only be used on this one occasion, it makes sense to me to just include the CSS inline. I have to stress that inline CSS is something you should rarely if ever use because it can get messy quick.

Courtesy: https://www.csstutorial.net/css_misc_inserting.php

Modified: 2021.10.04.10.40.PM

Dököll Solutions, Inc.