

Styles outline

Adapted from: <http://www.w3schools.com/css/default.asp>

Why use styles

When HTML was first developed, the purpose of tags was simply to place content and elements on a page, without specifying properties to change their appearance (formatting or styling). They were supposed to say "This is a header", "This is a paragraph", "This is a link", by using tags like `<h1>`, `<p>`, `<a>`, and so on. All considerations of styling were left to the browser (typically a white background and black text etc).

As you can imagine, from a visual point of view this would make things quite plain. During the 90's, in order to make HTML more versatile and visually appealing, the two main competing browsers, Netscape and Internet Explorer added new HTML tags and attributes (like the `` tag and the colour attribute) to the original HTML specification. This brought inconsistency to HTML. A situation occurred where certain things would work in one make of browser and not another. It also meant that the original idea of HTML as a tool just for placing elements on a page was being changed to include formatting.

To solve this problem, the World Wide Web Consortium (W3C) - the non profit, standard setting consortium responsible for standardising HTML - created *Styles*, employed through the use of Cascading Style Sheets (CSS), when HTML 4.0 was introduced.

Think of HTML and Styles like this:

- HTML is mark-up i.e. it places elements on a page
- Styles deal with considerations of aesthetics i.e. how things look

A major benefit of styles is that they give greater control of layout for the designer.

Style Sheets can save a Lot of Work

Styles define how HTML elements are displayed, just like the font tag and the colour attribute were used in HTML 3.2. Styles are normally saved in files external to your HTML documents. External style sheets enable you to change the appearance and layout of all the pages in a web site, just by editing a single CSS document. If you have ever tried to change the font or colour of all the headings in all your web pages, you will understand how CSS can save you a lot of work.

CSS is a breakthrough in web design because it allows developers to control the style and layout of multiple web pages all at once. As a web developer you can define a style for each HTML element and apply it to as many pages as you want. To make a global change, simply change the style, and all elements in the site are updated automatically.

Multiple Styles Will Cascade In to One

Style sheets can be created in different ways:

- Within an HTML tag (inline style)
- Within the <head> of an HTML document (internal style sheet)
- As an *external style sheet* (CSS). Even multiple external Style Sheets can be referenced inside a single HTML document.

Dreamweaver automatically creates styles in the <head> unless you create an external style sheet.

Cascading Order

The *cascade* means an order of priority. This defines what style will be used when there is more than one style for a specific element on the page. The 'cascade' is listed below. Number four has the highest priority:

1. Browser default
2. External Style Sheet
3. Internal Style Sheet
4. Inline Style

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style declared for the same element from the <head> tag, in an external style sheet, and from the browser (browser default).

To illustrate this, let us consider formatting properties for the default font of a web page.

1. If no style is specified then the *browser default* is used.
2. If a style for the default font is specified in an *external* style sheet then all web pages linked to that style sheet will use that style.
3. If, in one of those web pages a different default font is required then the style can be specified in an *internal* style sheet within the head tag. This will override the style defined in the external style sheet.
4. Finally, an *inline* style for the default font will override an internal style sheet. *Generally, inline styles are an inefficient use of styles. Their use is mostly appropriate for HTML emails.*

Different types of styles

Below is an outline of the three main types of styles

Tag selectors

This is a style that targets HTML tags in a web page. For instance, attributes such as font, colour and size can be specified for the <h1> tag. This will make every occurrence of that tag in your web page look a particular way.

Class selectors

This style is not dependant on any particular tag. It can be applied to any element in your HTML document by including the class name as an attribute of the HTML tag e.g. <p class="styleName"> or .

ID selectors

As with the class selector, this style is not dependant on any particular tag. It can be applied to any element in your HTML document by including the ID name (identifier) as an attribute of the HTML tag e.g. <p id="styleName"> or <div id="styleName">.

Class and ID selectors are essentially the same as each other except in the way they should be used:

- Class selectors can be applied more than once in a page.
- ID selectors are intended to be used only once in a page. They indicate that the object they are applied to is unique.

How styles are written

External style sheet

A link to the external style sheet appears in the head and looks like this:

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

href="mystyle.css" is file path to the style sheet.

Internal style sheet

An internal style sheet is written in the head and looks like this:

```
<head>

<style type="text/css">

hr {color: #CC0000}

p {margin-left: 20px}

body {background-image: url("images/back40.gif")}

</style>

</head>
```

Inline style

Inline styles are written in the body as part of the element they are affecting:

```
<p style="color: #CC0000; margin-left: 20px">
This is a paragraph
</p>
```

Grouping

You can group selectors. To do this, separate each selector with a comma. In the example below the header elements have been grouped. Each header element will be green:

```
h1,h2,h3,h4,h5,h6
{
color: #CC0000
}
```