

## Basic Cascading Style Sheets (CSS)

(© LSS 2010, last updated Jan 2013)

CSS is a technology which is used to make HTML files display in a particular style. HTML was not originally intended to describe exactly how web pages should be displayed, but as time went on, was used to do so. CSS enables a separation between the HTML file and the style-defining file. With HTML5, many of the options for changing the display of HTML that had previously been available have been removed, making CSS the primary way of altering how HTML 5 documents are displayed.

CSS files are particularly useful for applying one set of styles to many HTML pages, giving a whole website the same *look and feel*.

This document contains some elements of CSS. For more information, there are many resources available, for example <http://www.w3schools.com/css/default.asp> from w3schools, or any of the textbooks on CSS (e.g. E.A. Meyer: CSS: The Definitive Guide, O'Reilly 2006).

### CSS Cascades

CSS is *cascading* in the sense that styles defined in one place can be over-ridden by styles defined in other places. Styles may be defined

1. By using the browser's default style
2. By using an external style sheet
3. By using an internal style sheet (inside the <head> ... <head> section of the HTML file)
4. By using inline styles (inside an HTML element)

These *cascade*: that is something defined in 1 above will be over-ridden by something defined in 2 above, which in turn would be over-ridden by something in 3 above, etc.

### CSS Syntax

The CSS file is made up of a number of *rules*, where each rule consists of a *selector* and one or more *declarations*.

The selector is normally an HTML element (like h1, for the top level header, or ul for an un-numbered list), and is the HTML element whose style you want to define. Each declaration is a property (of that HTML element), followed by the value for that property.

The syntax used in CSS files (external style sheets) is different from that of HTML. For example, the text

```
h1 {color:green; font-size:13px;}
```

would mean that top level headings would be rendered in green in 13 point text. (The font type, and any other properties, would be unchanged).

In essence, the syntax is

*HTML\_element {declaration ; declaration; ... ;}*

Where each declaration consists of a property and a value for that property, separated by a `:` (as in `color: green`). CSS files support comments: these are introduced by `/*` and terminated with `*/`, as in `/* this is a comment */`. Note also that one can use a comma-separated list of selectors (for example `h1, h2, h3, h4`) before the `{`.

To use a particular CSS file from an HTML file, use the `<link>` tag within the header:

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

The syntax used inside an HTML file (internal style sheets) is similar: normally the CSS material is placed inside the head of the HTML file, inside a `<style>` element:

```
<style>
  body { background-color: black }
  h1   { background-color: red }
  h2   { color: green }
  p    { color: green; font-style: italic }
</style>
```

(Note that earlier versions of HTML required additional information in the `link` or `style` tags [`type="text/css"`]: HTML 5 defaults to CSS, so this should no longer be required.)

Where the CSS material is inside the actual tag (inline styles), it is written using the `style` property:

```
<h3 style="color:blue; font-style: italic;"> The heading </h3>
```

### Some CSS styles

CSS sections (whether external files, or inline sections in headers) are largely simply lists of how the styles for different HTML elements should be drawn. Different elements have different sets of properties. Thus, for example, the property `background-color` can be used for the `body` tag, or for any of the heading tags, or indeed, for many other tags. Similarly, properties that refer to text can be used in any HTML element that contains text, for example, `<p>`, `<h1>`, `<ul>`, `<li>` etc.

Here are some examples.

```
p {font-family:"Times New Roman",Georgia,Serif ; color red}
```

sets the font type of the text of a paragraph to Times New Roman, or if that is unavailable to Georgia, or, failing that to an available sans-serif font, and sets the text colour to red.

CSS can be used to customize both how links will be styled (in terms of the font, colour etc.), and also how they will appear and how they will alter depending on whether they are moused-over, or visited (these are known as pseudo-classes):

```
a:visited {color:#00FF00;} /* visited link */
a:hover {color:#FF00FF;} /* mouse over link */
a:active {color:#0000FF;} /* selected link */
```

Note `a:hover` must come after `a:link` and `a:visited` and that `a:active` MUST come after `a:hover`, and note also that other properties apart from the colour can be set.

```
canvas { border: 2px solid blue; }
```

This makes the border of the rectangle defined by the `canvas` tag 2 pixels wide, solid, and blue. The default style is that there is no border.

### Some additions: id selectors, class selectors and contextual selectors

So far, the way we have used CSS are such that they apply to all the HTML elements in their scope. However, it is possible to make the same HTML element have different styles applied to them. This is done by using an *id*, *class* or *contextual selector*.

#### Id selectors

Inside the HTML element we add the tag `id=something`, and then in the CSS we use

```
#something {declarations}
```

The CSS style will only be applied to tags with the `id=something` qualifier.

#### Class selectors

Inside the HTML element we add `class="someclass"`, and then in the CSS we write

```
.someclass {declarations}
```

The style will only be applied to elements with the appropriate class qualifier. We can refine this further: if the CSS is, for example,

```
li.someclass {declarations}
```

then the style will only be applied to `<li>` tags that have the class qualifier.

#### Contextual selectors

We can apply CSS styles to tags which occur only *inside* other tags, for example, a `<b>` tag occurring inside a `<li>` tag:

```
li b {color: #990000}
```

will change the colour of `<b>` text only when it occurs inside an `<li>` (i.e. "bold" text that occurs inside a list element).

## <div> and <span>

These two tags are primarily used in conjunction with CSS files. <div> and <span> allow the writer of an HTML file to define divisions of a document. <div> defines a division of a document (and puts a newline before and after it), and is primarily used to block a set of elements of a document together, and apply some specific styles to them. <span> is used simply to define some piece of text specifically so that it can have a style applied to it. For example, in the CSS file one might write

```
div.classnumber1 {color: blue}
```

and in the HTML file one would write

```
<div class="classnumber1"> ... </div>
```

The effect would be to make the text inside the division blue.

Using <span> one might write

```
span#aside {font-size:50%;}
```

and in the HTML file write

```
<span id=aside> ... </span>
```

with the result that the text inside the <span> ... </span> brackets would be much smaller.

## To conclude

There is an example of the use of CSS at [http://www.cs.stir.ac.uk/~lss/CSC9B2/example\\_ns.html](http://www.cs.stir.ac.uk/~lss/CSC9B2/example_ns.html) and this uses the file <http://www.cs.stir.ac.uk/~lss/CSC9B2/somestyles.css> . Download these and experiment.

There's an excellent website demonstrating CSS effects at [www.csszengarden.com](http://www.csszengarden.com).

There are a many different CSS selectors, and an even greater number of CSS properties. Together they allow the designer to control many aspects of the HTML file, and can be used in such a way as to provide a consistency of design across a whole set of web pages. A very useful annotated reference may be found at

[http://www.w3schools.com/css/css\\_reference\\_atoz.asp](http://www.w3schools.com/css/css_reference_atoz.asp)

and for those who would like to try out other techniques this page is strongly recommended.