Basic HTML | Advanced HTML

Adding a touch of style

Dave Raggett

This is a short guide to styling your Web pages. It will show you how to use W3C's Cascading Style Sheets language (CSS) as well as alternatives using HTML itself. The route will steer you clear of most of the problems caused by differences between different brands and versions of browsers.

For style sheets to work, it is important that your markup be free of errors. A convenient way to automatically fix markup errors is to use the HTML Tidy utility. This also tidies the markup making it easier to read and easier to edit. I recommend you regularly run Tidy over any markup you are editing. Tidy is very effective at cleaning up markup created by authoring tools with sloppy habits.

The following will teach you how to:

- use the style element
- link to separate style sheets
- set page margins
- set left and right and first-line indents
- set the amount of whitespace above and below
- set the font type, style and size
- add borders and backgrounds
- set colors with named or numeric values
- add style for browsers that don't understand CSS

Getting started

Let's start with setting the color of the text and the background. You can do this by using the STYLE element to set style properties for the document's tags:

```
<style type="text/css">
  body { color: black; background: white; }
```

```
</style>
```

The style element is placed within the document head. Here is a template HTML file showing where the above style element fits. You can copy and paste this into your editor as a convenient way to experiment with CSS style sheets:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title> replace with your document's title </title>
<style type="text/css">
  body { color: black; background: white; }
</style>
</head>
<body>

replace with your document's content

</body>
</html>
```

The stuff between the <style> and </style> is written in special notation for style rules. Each rule starts with a tag name followed by a list of style properties bracketed by { and }. In this example, the rule matches the ***body*** tag. As you will see, the body tag provides the basis for setting the overall look and feel of your Web page.

Each style property starts with the property's name, then a colon and lastly the value for this property. When there is more than one style property in the list, you need to use a semicolon between each of them to delimit one property from the next. In this example, there are two properties - "color" which sets the color of the text, and "background" which sets the color of the page background. I recommend always adding the semicolon even after the last property.

Colors can be given as names or as numerical values, for instance `rgb(255, 204, 204)` which is a fleshy pink. The 3 numbers correspond to red, green and blue respectively in the range 0 to 255. You can also use a hexadecimal notation, the same color can also be written as `#FFCCCC`. More details on color are given in a later section.

Note that the style element must be placed in the document's head along with the title element. It shouldn't be placed within the body.

Linking to a separate style sheet

If you are likely to want to use the same styles for several Web pages it is worth considering using a separate style sheet which you then link from each page. You can do this as follows:

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

The ***link*** tag should be placed within the <head> ... </head> element. Here is an HTML file with a link to an external style sheet:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title> replace with your document's title </title>
<link type="text/css" rel="stylesheet" href="style.css">
</head>
<body>

replace with your document's content

</body>
</html>
```

The link element's ***rel*** attribute must be set to the value "stylesheet" to allow the browser to recognize that the ***href*** attribute gives the Web address (URL) for your style sheet. A simple stylesheet file might look like the following:

```
/* style.css - a simple style sheet */
body {
  margin-left: 10%; margin-right: 10%;
  color: black; background: white;
}
```

Note that the same HTML file can link to an external style sheet and also include a style element for additional style settings specific to this page. If you place the link element before the style element, you can use the latter to override the style settings in the linked style sheet.

Setting the page margins

Web pages look a lot nicer with bigger margins. You can set the left and right margins with the "margin-left" and "margin-right" properties, e.g.

```
<style type="text/css">
  body { margin-left: 10%; margin-right: 10%; }
</style>
```

This sets both margins to 10% of the window width, and the margins will scale when you resize the browser window.

Setting left and right indents

To make headings a little more distinctive, you can make them start within the margin set for the body, e.g.

```
<style type="text/css">
  body { margin-left: 10%; margin-right: 10%; }
  h1 { margin-left: -8%;}
  h2,h3,h4,h5,h6 { margin-left: -4%; }
</style>
```

This example has three style rules. One for the body, one for h1 (used for the most important headings) and one for the rest of the headings (h2, h3, h4, h5 and h6). The margins for the headings are additive to the margins for the body. Negative values are used to move the start of the headings to the left of the margin set for the body.

In the following sections, the examples of particular style rules will need to be placed within the style element in the document's head (if present) or in a linked style sheet.

Controlling the white space above and below

Browsers do a pretty good job for the white space above and below headings and paragraphs etc. Two reasons for taking control of this yourself are: when you want a lot of white space before a particular heading or paragraph, or when you need precise control for the general spacings.

The "margin-top" property specifies the space above and the "margin-bottom" specifies the space below. To set these for all h2 headings you can write:

```
h2 { margin-top: 8em; margin-bottom: 3em; }
```

The em is a very useful unit as it scales with the size of the font. One em is the height of the font. By using em's you can preserve the general look of the Web page independently of the font size. This is much safer than alternatives such as pixels or points, which can cause problems for users who need large fonts to read the text.

Points are commonly used in word processing packages, e.g. 10pt text. Unfortunately the same point size is rendered differently on different browsers. What works fine for one browser will be illegible on another! Sticking with em's avoids these problems.

To specify the space above a particular heading, you should create a named style for the heading. You do this with the ***class*** attribute in the markup, e.g.

```
<h2 class="subsection">Getting started</h2>
```

The style rule is then written as:

```
h2.subsection { margin-top: 8em; margin-bottom: 3em; }
```

The rule starts with the tag name, a dot and then the value of the class attribute. Be careful to avoid placing a space before or after the dot. If you do the rule won't work. There are other ways to set the styles for a particular element but the class attribute is the most flexible.

When a heading is followed by a paragraph, the value for margin-bottom for the heading isn't added to the value for margin-top for the paragraph. Instead, the maximum of the two values is used for the spacing between the heading and paragraph. This subtlety applies to margin-top and margin-bottom regardless of which tags are involved.

First-line indent

Sometimes you may want to indent the first line of each paragraph. The following style rule emulates the traditional way paragraphs are rendered in novels:

```
p { text-indent: 2em; margin-top: 0; margin-bottom: 0; }
```

It indents the first line of each paragraph by 2 em's and suppresses the inter-paragraph spacing.

Controlling the font

This section explains how to set the font and size, and how to add italic, bold and other styles.

Font styles

The most common styles are to place text in italic or bold. Most browsers render the ***em*** tag in italic and the ***strong*** tag in bold. Let's assume you instead want em to appear in ***bold italic*** and strong in **bold uppercase**:

```
em { font-style: italic; font-weight: bold; }
strong { text-transform: uppercase;  font-weight: bold; }
```

If you feel so inclined, you can fold headings to lower case as follows:

```
h2 { text-transform: lowercase; }
```

Setting the font size

Most browsers use a larger font size for more important headings. If you override the default size, you run the risk of making the text too small to be legible, particularly if you use points. You are therefore recommended to specify font sizes in relative terms.

This example sets heading sizes in percentages relative to the size used for normal text:

```
h1 { font-size: 200%; }
h2 { font-size: 150%; }
h3 { font-size: 100%; }
```

Setting the font family

It is likely that your favorite font won't be available on all browsers. To get around this, you are allowed to list several fonts in preference order. There is a short list of generic font names which are guaranteed to be available, so you are recommended to end your list with one of these: serif, sans-serif, cursive, fantasy, or monospace, for instance:

```
body { font-family: Verdana, sans-serif; }
h1,h2 { font-family: Garamond, "Times New Roman", serif; }
```

In this example, important headings would preferably be shown in Garamond, failing that in Times New Roman, and if that is unavailable in the browsers default serif font. Paragraph text would appear in Verdana or if that is unavailable in the browser's default sans-serif font.

The legibility of different fonts generally depends more on the height of lower case letters than on the font size itself. Fonts like Verdana are much more legible than ones like "Times New Roman" and are therefore recommended for paragraph text.

Avoid problems with fonts and margins

My first rule is to avoid text at the body level that isn't wrapped in a block level element such as ***p***. For instance:

```
<h2>Spring in Wiltshire</h2>

Blossom on the trees, bird song and the sound of lambs
bleating in the fields.
```

The text following the heading runs the risk on some browsers of being rendered with the wrong font and margins. You are therefore advised to enclose all such text in a paragraph, e.g.

```
<h2>Spring in Wiltshire</h2>

<p>Blossom on the trees, bird song and the sound of lambs
bleating in the fields.</p>
```

My second rule is to set the font family for ***pre*** elements, as some browsers forget to use a fixed pitch font when you set the font size or other properties for pre.

```
pre { font-family: monospace; }
```

My third rule is to set the font family on headings, p and ul elements if you intend to set borders or backgrounds on elements such as div. This is a work-around for a bug where the browser forgets to use the inherited font family, instead switching to the default font as set by the browser preferences.

```
h1,h2,h3,h4,h5,p,ul { font-family: sans-serif; }
```

Adding borders and backgrounds

You can easily add a border around a heading, list, paragraph or a group of these enclosed with a ***div*** element. For instance:

```
div.box { border: solid; border-width: thin; width: 100% }
```

Note that without the "width" property some browsers will place the right margin too far to the right. This can then be used with markup such as:

```
<div class="box">
The content within this DIV element will be enclosed
in a box with a thin line around it.
</div>
```

There are a limited choice of border types: dotted, dashed, solid, double, groove, ridge, inset and outset. The border-width property sets the width. Its values include thin, medium and thick as well as a specified width e.g. 0.1em. The border-color property allows you to set the color.

A nice effect is to paint the background of the box with a solid color or with a tiled image. To do this you use the background property. You can fill the box enclosing a div as follows:

```
div.color {
  background: rgb(204,204,255);
  padding: 0.5em;
  border: none;
}
```

Without an explicit definition for border property some browsers will only paint the background color under each character. The padding property introduces some space between the edges of the colored region and the text it contains.

You can set different values for padding on the left, top, right and bottom sides with the padding-left, padding-top, padding-right and padding-bottom properties, e.g. `padding-left: 1em`.

Suppose you only want borders on some of the sides. You can control the border properties for each of the sides independently using the border-left, border-top, border-right and border-bottom family of properties together with the appropriate suffix: style, width or color, e.g.

```
p.changed {
  padding-left: 0.2em;
  border-left: solid;
  border-right: none;
  border-top: none;
  border-bottom: none;
  border-left-width: thin;
  border-color: red;
}
```

which sets a red border down the left hand side only of any paragraph with the class "changed".

Setting Colors

Some examples for setting colors appeared in earlier sections. Here is a reminder:

```
body {
  color: black;
  background: white;
}
```

```
  strong { color: red }
```

This sets the default to black text on a white background, but renders strong elements in red. There are 16 standard color name, which are explained just below. You can also use decimal values for red, green and blue, where each value appears in the range 0 to 255, e.g. *rgb(255, 0, 0)* is the same as *red.* You can also used hex color values which start with the '#' characted followed by six hexadecimal digits. A two-way converter is included below which allows you to convert from RGB to hex color values.

Setting Link Colors

You can use CSS to set the color for hypertext links, with a different color for links that you have yet to follow, ones you have followed, and the active color for when the link is being clicked. You can even set the color for when the mouse pointer is hovering over the link.

```
  :link { color: rgb(0, 0, 153) }  /* for unvisited links */
  :visited { color: rgb(153, 0, 153) } /* for visited links */
  a:active { color: rgb(255, 0, 102) } /* when link is clicked */
  a:hover { color: rgb(0, 96, 255) } /* when mouse is over link */
```

Sometimes you may want to show hypertext links without them being underlined. You can do this by setting the *text-decoration* property to *none*, for example:

```
  a.plain { text-decoration: none }
```

Which would suppress underlining for a link such as:

```
This is <a class="plain" href="what.html">not underlined</a>
```

Most people when they see underlined text on a Web page, will expect it to be part of a hypertext link. As a result, you are advised to leave underlining on for hypertext links. A similar argument applies to the link colors, most people will interpret underlined blue text as hypertext links. You are advised to leave link colors alone, except when the color of the background would otherwise make the text hard to read.

Color Blindness

When using color, remember that 5 to 10% of men have some form of color blindness. This can cause difficulties distinguishing between red and green, or between yellow and blue. In rare cases, there is an inability to perceive any colors at all. You are recommended to avoid foreground/background color combinations that would make the text hard to read for people with color blindness.

Named colors

The standard set of color names is: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. These 16 colors are defined in HTML 3.2 and 4.01 and correspond to the basic VGA set on PCs. Most browsers accept a wider set of color names but use of these is not recommended.

Color names and sRGB values

■	black = "#000000"	■	green = "#008000"
■	silver = "#C0C0C0"	■	lime = "#00FF00"
■	gray = "#808080"	■	olive = "#808000"
□	white = "#FFFFFF"	■	yellow = "#FFFF00"
■	maroon = "#800000"	■	navy = "#000080"
■	red = "#FF0000"	■	blue = "#0000FF"
■	purple = "#800080"	■	teal = "#008080"
■	fuchsia = "#FF00FF"	■	aqua = "#00FFFF"

Thus, the color value "#800080" is the same as "purple".

Hexadecimal color values

Values like "#FF9999" represent colors as hexadecimal numbers for red, green and blue. The first two characters following the # give the number for red, the next two for green and the last two for blue. These numbers are always in the range 0 to 255 decimal. If you know the values in decimal, you can convert to hexadecimal using a calculator, like the one that comes as part of Microsoft Windows.

Enter RGB or Hex value and
press appropriate button to
convert

red: 255 Hex color value

green: 255 #FFFFFF

blue: 255 HEX? RGB?

Browser safe colors

New computers support thousands or millions of colors, but many older color systems can only show up to 256 colors at a time. To cope with this, these browsers make do with colors from a fixed palette. The effect of this is often visible as a speckling of colors as the browser tries to approximate the true color at any point in the image. This problem will gradually go away as older computers are replaced by newer models.

Most browsers support the same so called "browser safe" palette. This uses 6 evenly spaced gradations in red, green and blue and their combinations. If you select image colors from this palette, you can avoid the speckling effect. This is particularly useful for background areas of images.

If the browser is using the browser safe palette, the page background uses the nearest color in the palette. If you set the page background to a color which isn't in the browser safe palette, you run the risk that the background will have different colors depending on whether the computer is using indexed or true-color.

These are constructed from colors where red, green and blue are restricted to the values:

RGB	00	51	102	153	204	255
Hex	00	33	66	99	CC	FF

Here is a table of the browser safe colors and their hex values. My thanks to Bob Stein for this arrangement.

FFF FFF	CCC CCC	999 999	666 666	333 333	000 000	FFC C00	FF9 900	FF6 600	FF3 300						
99C C00					CC9 900	FFC C33	FFC C66	FF9 966	FF6 633	CC3 300					CC0 033
CCF F00	CCF F33	333 300	666 600	999 900	CCC C00	FFF F00	CC9 933	CC6 633	330 000	660 000	990 000	CC0 000	FF0 000	FF3 366	FF0 033
99F F00	CCF F66	99C C33	666 633	999 933	CCC C33	FFF F33	996 600	993 300	663 333	993 333	CC3 333	FF3 333	CC3 366	FF6 699	FF0 066
66F F00	99F F66	66C C33	669 900	999 966	CCC C66	FFF F66	996 633	663 300	996 666	CC6 666	FF6 666	990 033	CC3 399	FF6 6CC	FF0 099

33F F00	66F F33	339 900	66C C00	99F F33	CCC C99	FFF F99	CC9 966	CC6 600	CC9 999	FF9 999	FF3 399	CC0 066	990 066	FF3 3CC	FF0 0CC
00C C00	33C C00	336 600	669 933	99C C66	CCF F99	FFF FCC	FFC C99	FF9 933	FFC CCC	FF9 9CC	CC6 699	993 366	660 033	CC0 099	330 033
33C C33	66C C66	00F F00	33F F33	66F F66	99F F99	CCF FCC				CC9 9CC	996 699	993 399	990 099	663 366	660 066
006 600	336 633	009 900	339 933	669 966	99C C99				FFC CFF	FF9 9FF	FF6 6FF	FF3 3FF	FF0 0FF	CC6 6CC	CC3 3CC
003 300	00C C33	006 633	339 966	66C C99	99F FCC	CCF FFF	339 9FF	99C CFF	CCC CFF	CC9 9FF	996 6CC	663 399	330 066	990 0CC	CC0 0CC
00F F33	33F F66	009 933	00C C66	33F F99	99F FFF	99C CCC	006 6CC	669 9CC	999 9FF	999 9CC	993 3FF	660 0CC	660 099	CC3 3FF	CC0 0FF
00F F66	66F F99	33C C66	009 966	66F FFF	66C CCC	669 999	003 366	336 699	666 6FF	666 6CC	666 699	330 099	993 3CC	CC6 6FF	990 0FF
00F F99	66F FCC	33C C99	33F FFF	33C CCC	339 999	336 666	006 699	003 399	333 3FF	333 3CC	333 399	333 366	663 3CC	996 6FF	660 0FF
00F FCC	33F FCC	00F FFF	00C CCC	009 999	006 666	003 333	339 9CC	336 6CC	000 0FF	000 0CC	000 099	000 066	000 033	663 3FF	330 0FF
00C C99					009 9CC	33C CFF	66C CFF	669 9FF	336 6FF	003 3CC					330 0CC
						00C CFF	009 9FF	006 6FF	003 3FF						

Color swatches for the browser safe palette are available free for many popular graphics packages, from www.visibone.com.