HTML Emulation of CSS Example

= Index DOT Css by **Brian Wilson** =

Example

1. <**html**>
2. <**head**>
3. <**title**>Document Title</**title**>
4. </**head**>

5. [Go To Analysis] <**body** BGCOLOR="#000000" TEXT="#80c0c0" LINK="#ff8080" VLINK="#ff0000" ALINK="#a05050">
6. [Go To Analysis] <**div** ALIGN="center">
7. [Go To Analysis] <**table** BGCOLOR="#000080" BORDER=3 BORDERCOLOR="#0000ff"><**tr**>
8. [Go To Analysis] <**td**><**h1**><**font** FACE="courier" SIZE=6><**font** SIZE=7>W</**font**>ELCOME TO MY HOME PAGE!</**font**></**h1**></**td**>
9. </**tr**></**table**>
10. </**div**>
11. <**br**><**br**>
12. [Go To Analysis] <**p**>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Hi there! If you are reading this then you have found my home page! Congratulations! I know it can be hard to find my pages, but I bet that you feel lucky now. Now that you are here, please take a look at my page of links to <**a** HREF="http://www.mysite.com/coolsites.html">cool sites</**a**> or sign my <**a** HREF="http://www.mysite.com/guestbook.html">guest book</**a**></**p**>
13. [Go To Analysis] <**div**>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <**b**><**font** SIZE=6 COLOR="#ffffff">M</**font**>y wonderful poetry</**b**> is available if you are REALLY bored. Why not give it a spin?</**div**>

14. [Go To Analysis] <**h2**><**font** SIZE=7 COLOR="#00ff00"> The Best Poetry I <**em**><**b**>NEVER</**b**></**em**> Wrote</**font**></**h2**>
15. <**ul**>
16. [Go To Analysis] <**li**>'There Once Was A Man From Nantucket' - <**font** FACE="'comic sans ms', fantasy" COLOR="#ffff00">Anonymous</**font**></**li**>
17. [Go To Analysis] <**li**>'Cool In Fur' - <**font** FACE="'comic sans ms', fantasy" COLOR="#ffff00">Harry B. Foot</**font**></**li**>
18. <**li**>And My All Time Fave:
19. <**ul**>
20. [Go To Analysis] <**li**><**font** SIZE=6><**i**> 'A Toast To My Toaster' -
21. [Go To Analysis] <**font** FACE="'comic sans ms', fantasy" COLOR="#ffff00">Me!</**font**></**i**></**font**></**li**>
22. </**ul**>
23. </**li**></**ul**>

24. [Go To Analysis] <**table** BGCOLOR="#000080"><**tr**>

```
25. [Go To Analysis]     <td WIDTH=50>&nbsp;</td>
26. [Go To Analysis]     <td><font COLOR="#00ff00">Brought to you by the letter
27. [Go To Analysis]       <font FACE="'comic sans ms', fantasy" COLOR="#ffff00">&quot;H&quot;
    </font> and <u>Joe Shmoe</u></font></td>
28.      </tr></table>

29. [Go To Analysis] <div><b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font SIZE=6
    COLOR="#ffffff">W</font>hen you are done looking through these masterpieces, I encourage you to
    visit my proud sponsor!! </b></div>

30. [Go To Analysis] <table BGCOLOR="#ffffff" BORDER=2 BORDERCOLOR="#0000ff"><tr>
31. [Go To Analysis]     <td WIDTH=5>&nbsp;</td>
32. [Go To Analysis]     <td><font SIZE=6>ADVERTISEMENT</font> Buy Navel Lint Contemplator!
    Its a browser and its a sandwich spread! Go to our <a HREF="http://www.navellint.com">home page</a>
    without delay! Why? Because shelf life is only 8 hours unless refrigerated. We know that makes it hard to
    browse, but it promotes frequent upgrading to the latest and greatest version. </td>
33.      </tr></table>

34. [Go To Analysis] <h6><font SIZE=1 COLOR="#ff0000">All standard disclaimers apply. Your life
    depends on NOT copying this document in any way. This tape will <a
    HREF="http://www.mysite.com/selfdestruct.html">self destruct</a> in 10 seconds...</font></h6>

35.     </body>
36.     </html>
```

Analysis

- **Line:** General
  CSS Properties/Issues: NA
  Description: We lose a lot here when we attempt to kludge CSS behavior with normal HTML. Most of the attempted solutions use FONT elements for text formatting and TABLE elements for layout control - with varying degrees of success. When using plain HTML, at some point you need to recognize that its display control REALLY is small compared to CSS.
- **Line:** 5
  CSS Properties/Issues: background, color, Pseudo-classes
  Description: We are using BODY element attributes to control hyperlink behavior and text/background appearance for the entire document. The lack of state dependent anchor pseudo-classes in HTML makes it difficult to create hyperlinks of varying appearance.
- **Line:** 6
  CSS Properties/Issues: background, font, font-variant, border
  Description: In the CSS examples we only use an H1 in this location. Because we are attempting to create a border on this content, we must resort to using a TABLE structure. The original H1 element was centered, so we encapsulate the TABLE in a center-aligned DIV element.
- **Line:** 7
  CSS Properties/Issues: background, font, font-variant, border
  Description: The opening TABLE tag here uses attributes to partially create the border and background effects used in the original CSS example. An arbitrary border pixel thickness is used here (keyword values not supported) while the 'dashed' border behavior in the CSS version is not possible. Problem: The solution of using the BORDERCOLOR attribute of the TABLE element has limited browser support.

- **Line:** 8
  CSS Properties/Issues: background, font, font-variant, border, Pseudo-elements
  Description: We are able to simulate the 'font-variant' and 'font-size' properties with simple HTML. Small caps is achieved by varying the font size on already capitalized text. The 'line-height' portion of the 'font' property is not controllable in HTML. Notice we are able to retain the <H1> heading level of the content here.
- **Line:** 12
  CSS Properties/Issues: margin-left, Pseudo-elements
  Description: The CSS version of this P element gives a 'margin-left' value to the first line. This can be simulated fairly easily in HTML using multiple non-breaking spaces (&nbsp; or &#160;) This is a fairly portable and safe solution.
- **Line:** 13
  CSS Properties/Issues: font-weight, margin-left, font-size, color, Pseudo-elements
  Description: The original CSS examples for this section employ 'first-line' and 'first-letter' pseudo-elements. The first-letter portion can be manually determined, but the first line is harder to pin down unless explicitly stated. Our 'font-weight' and 'margin-left' properties used for the first line have fair equivalents in HTML: the <b> element and non-breaking spaces (see previous analysis point.) HTML only has an on/off toggle for applying bold to text, so we are lucky that the original case only used this method as well. The styling on the first-letter is again achieved through attributes to the FONT element.
- **Line:** 14
  CSS Properties/Issues: background, color, line-height, font-size, font-weight, Contextual Selectors
  Description: The original CSS phrase for the H2 element is not easy to duplicate in HTML. Line-height and backgrounds for regular elements are not controllable in HTML, while the explicit font-size originally used can only be approximated with the FONT element (which is also used to create the text coloring here.) The nested EM element uses the font-weight property. As mentioned in the previous analysis point, HTML only has one bold setting, not the 9 that are possible under CSS. We use <B> here, and while this gradient range is not as rich as what CSS allows, it gives enough visual distinction from surrounding content to be satisfactory.
- **Line:** 16 / 17 / 21 / 27
  CSS Properties/Issues: font-family, color
  Description: We are lucky in these examples that the only properties originally used control text color and font family, because they are easily controlled currently with the FONT element in HTML.
- **Line:** 20
  CSS Properties/Issues: font-size, font-style
  Description: Again we lucked out in regards to the original example. The original case used the 'font-size' and 'font-style' properties with values that are easily converted to HTML. These properties, as well as most of the others, allow other values that are not as easily converted to plain HTML. This is where the real power of CSS over HTML for specifying layout information becomes apparent.
- **Line:** 24
  CSS Properties/Issues: background
  Description: We again resort to the use of tables to achieve the desired layout characteristics here. The original container in the CSS examples was the BLOCKQUOTE element, which we lose here in order to create the desired formatting. This is THE most frequent tradeoff when using HTML to layout content - loss of structural meaning. Using tables we DO achieve, however, the ability to manipulate background colors, borders and a small degree of margin control.
- **Line:** 25
  CSS Properties/Issues: margin-left
  Description: Here we use an initial table cell in the row to create a fixed-width left margin of 50 pixels. Our original unit measure was centimeters, so the value used is approximate only.
- **Line:** 26
  CSS Properties/Issues: color
  Description: We assign the text color information for the original BLOCKQUOTE element here at the cell level using the FONT element.

- **Line:** 27
  CSS Properties/Issues: font-family, color, padding, border
  Description: Our original CSS version contains complex padding and border properties to the 'Joe Shmoe' text section. Neither of these properties are fully possible with ordinary HTML even when resorting to complex table structures. Rather than make the document even more complex, we settle for a simple underlining effect.
- **Line:** 29
  CSS Properties/Issues: background, font-weight, margin-left, font-size, color
  Description: Originally we had a background on this element in the CSS version, but that is not possible with HTML. We ignore it in this case. We do have the advantage again of being able to kludge 'first-letter' property effects using the FONT element, and our 'margin-left' and 'font-weight' properties similarly degrade nicely to non-breaking spaces and <B> elements respectively.
- **Line:** 30
  CSS Properties/Issues: border, background
  Description: We go back to the well again with the over-used solution of HTML tables to create equivalent display effects. This time we are not so lucky. Many of the original properties used do not degrade well to HTML layout attempts. Our original example had numerous display properties assigned to the P element, but we are going to lose some: font line-height and generic font names disappear. Precise four-side margin control is also out. We also lose element background image placement and positioning too. What we end up salvaging is the border-style, -size and -color behaviors at the TABLE level.
- **Line:** 31
  CSS Properties/Issues: margin
  Description: We use the initial cell of the row to again create a left margin effect. Fortunately, the units originally used and those possible with HTML are compatible in this example.
- **Line:** 32
  CSS Properties/Issues: font, font-variant
  Description: Our 'ADVERTISEMENT:' banner originally had many additional style property assignments (font, font-variant, border), most of which can not be achieved using ordinary HTML. We resort to just making the text big in this case.
- **Line:** 34
  CSS Properties/Issues: font-size, color
  Description: FONT attributes are used here as in many of the other parts of this example to achieve equivalent display characteristics of the original CSS properties.