

Web Design Practical 2: Cascading Style Sheets

AIM: TO LEARN TO CONTROL THE FORMATTING OF WEB PAGES USING CSS

This sheet contains one checkpoint. Deadline: Thursday 23rd March

FURTHER DOCUMENTATION

There is a summary of CSS on the Web Design page on the Resources page in the References section of the CSCU9A2 website (<http://www.cs.stir.ac.uk/courses/CSCU9A2/resources.php>). For more detail, consult Chapters 5 and 6 of *Creating a Website: the Missing Manual*. See also the W3Schools tutorial on CSS at <http://www.w3schools.com/css/>.

IMAGES AND ALIGNMENT

Using TextPad, open the home page that you prepared and saved in your Web folder during the previous practical. If you followed the instructions for that practical closely, your home page is probably in a file called `index.html`. (You can open this file by launching TextPad from the **Start** menu, and then dragging the icon for your home page onto the TextPad window.)

Remember that the URL for this page will have the format:

`http://www.students.stir.ac.uk/~qjx001/`

Add some images to your page. Use the `` tag. You can find images on the web or choose some from our collection at <http://www.cs.stir.ac.uk/~sbj/images/>.

Remember that the HTML image tag syntax is:

```

```

and that the system is unforgiving of errors in the URL or filename. Note that you should **always** use the `alt` attribute. The `alt="..."` attribute within the `img` tag gives a short message that pops up when the mouse moves over an image. This is what a text-only browser would display – and is what a visually impaired person reading a web page with a screen reader would hear.

You should start by using the full URLs of the images; then try copying the images to your web folder and using relative path names in the `img` tag. If you are unsure please ask.

Try putting several images on the same line. Watch the browser rearrange them as you make the window narrower. Try forcing them onto separate lines using a line break tag `
`.

INTRODUCING CSS

You will find it useful to refer the document on CSS on the Resources page in the Reference section of the CSCU9A2 website. (<http://www.cs.stir.ac.uk/courses/CSCU9A2/resources.php>)

Copy the folder **CSS** from **Wide\Groups\CSCU9A2** to your Web folder. Now take a look at its contents. There are two subfolders: `resume` and `PessimistReviews`. These are examples taken from Creating a Website: the Missing Manual.

Take a look at the `resume` example first. The folder contains an unstyled HTML file, `resume_NoStyle.html`, a stylesheet `resume.css`, and a styled version of the first HTML file, `resume_WithStyle.html`. View the HTML files in a browser to see the effect of applying the stylesheet, and view all three files in TextPad so see how this effect is achieved. Consult the CSS reference document for an explanation of the CSS syntax. Ask a demonstrator if there is anything you do not understand.

Next, take a look at the second example, `PessimistReviews`, which contains an unstyled HTML page and two alternative styled versions. The first stylesheet demonstrates the use of *class selectors* to differentiate between different elements of the same type. This allows paragraph elements with the class “`byline`” to be styled differently from paragraph elements with the style “`review`”. The second stylesheet demonstrates the use of *divisions*.

WRITE YOUR OWN CSS

Next you will practice writing your stylesheets and applying them to your own HTML documents. First put together an HTML document containing a variety of elements, including tables, lists (both ordered and unordered), links (internal and external), etc. If you are starting with the home page you created earlier, take a copy of it. Call this **document1.html**. Currently, this document will be formatted according to the default rules of the browser you are using.

To explore the “cascade” of CSS: First, create an external CSS file (call it **demo1.css**), and set it up so that both ordered and unordered lists are in italic font in red, in 14 point text. Within the head of `document1.html`, add a link to your stylesheet. Next, add an internal CSS section to the header of `document1.html` so that ordered lists are in italic font in blue. Check that this works. Lastly, add some CSS to one single unordered list somewhere within `document1.html` so that the text in this list is in green. Check that this works.

For the next task, start with a fairly long web page. This could be a new copy of your home page, but add enough content to it to make it around at least 100 lines long (not counting blank lines). I’ll assume that this document is called **document2.html**.

Introduce three *divisions* in `document2.html`, with different class names. Using a new external CSS file (call it **demo2.css**), make the second level headings in these red, green and blue. Note: To style a specific element, say `h2`, within the `div` of a particular class, say `class1`, you can use rules that look like this:

```
div.class1 h2
{
    color:red
}
```

If some of the divisions in `document2.html` do not contain second level headings, then add some to test your CSS works as intended. Alter the fonts in each division so that they are from the same family, but of different sizes. Using a `span` tag, make some of the text in each division bold face.

Using *contextual selectors* and *id selectors*, create an unordered list with several elements where the different elements are formatted in three different ways.

Exercise:

1. Using a new CSS file (call it **mystyle.css**), alter the web page that you originally created (your home page in **index.html**) so that it uses CSS to make the page more visually appealing. Choose, for example, contrasting colours for text and backgrounds, and colours that make it clear which links have been visited, make the headings visually different from the main text, but not in a garish way. Remember that using too many different fonts tends to make a page look jumbled and difficult to read.
2. Do the exact opposite: create a new CSS file (call it **badstyle.css**) which makes your home page as *unappealing* as possible first: break all the rules you can think of (except those relating to content) for display. Use CSS and the `div` and `span` tags to break the document up and have lots of different formatting present.

CHECKPOINT [CSS]

Please demonstrate all the HTML and CSS pages you created (**demo1.css** with **document1.html**; **demo2.css** with **document2.html**; **mystyle.css** and **badstyle.css** with the **two** corresponding styled versions of your home page). Attempt to *validate* your pages on the W3C web site (<http://validator.w3.org/>) *before* you demonstrate your pages.