

CSCU9B2 Practical 2: Cascading Style Sheets

Aim: To learn to control the formatting of web pages using Cascading Style Sheets (CSS)

Please register your practical attendance: Go to the GROUPS\CSCU9B2 folder in your Computer folder and double-click on the Register icon. Ask a demonstrator if you need help or something goes wrong.

This sheet contains one checkpoint (see end of sheet).

Further Documentation

There is a short summary of CSS on the CSCU9B2 module **Resources** web page and the lecture slides should also be a handy reference point. 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/>.

Examples of CSS

To get you started on using CSS, we will look at a couple of examples.

Copy the folder **CSS** from **Wide\Groups\CSCU9B2** 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* (<div> tag).

Reminder: ID and CLASS Selectors

One thing that makes CSS rather powerful is its use of the HTML5 ability to identify particular instances of tags using the **id** and **class** selectors. Intelligent usage of this enables a single external CSS file to be used to create a single “look and feel” for a complete web site consisting of multiple pages. Doing this is beyond the scope of the course, but the methods we describe are those used.

More or less every tag can have an id and /or a class selector. The idea essentially is:

id: a tag can be labelled with an id, enabling a specific set of formatting rules to be applied to it. The id is specific to a particular tag instance.

class: a tag can be labelled with a class, enabling a specific set of formatting rules to be applied to it. Many tag instances can belong to the same class.

For example, the CSS tag

```
#idname {...}
```

will apply the formatting to the single tag with the id *idname*, and the tag

```
.classname {...}
```

will apply the formatting to all tags with the class *classname*. You can also restrict which tags have a particular set of formatting used by writing something like

```
p#idname {...}
```

or

```
p.classname {...}
```

to restrict the application to a specific tag (in this case <p>).

Writing Your own CSS

Now you will practice writing your own stylesheets and applying them to your own HTML documents. Either start from your **index.html** page from the first practical, or 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 (index.html) you created earlier, take a copy of it (so that you don't over-write it). Call this **document1.html**. Currently, this document will be formatted according to the default rules of the browser you are using.

Explore the “cascade” of CSS

To start with, create a separate CSS file (call it **demo1.css**), and add style rules to it so that both ordered and unordered lists are in italic font in red, with 14 point text. Check that this works.

Now add an internal (embedded) CSS style sheet to the head of document1.html so that ordered lists are in italic font in blue. Check that this works.

Now add some CSS (inline style) to one single unordered list somewhere within document1.html so that the text in this list is in green. Check that this works.

Finally, using *contextual selectors* and *id selectors* (either in demo1.css or in your embedded style sheet) make two list items in one particular list (ordered or unordered) have different formatting from each other and all the other items in the list.

Using Divisions

For the next task, start with a fairly long web page. This could be a new copy of your home page, but you might have to add further content to it to carry out the following tasks (add content as you need it). I'll assume that this new 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 divisions red, green and blue, respectively. (If some of the divisions in document2.html do not contain second level headings, then add some!)

Alter the fonts in each division so that they are from the same family, but of different sizes.

Using **span** tags, make some of the text in each division bold face.

Images and Alignment

Now edit a web page containing some images: document1.html or document2.html may already contain `` tags and will be fine for this exercise. If not, you can use another HTML file, or add some image tags (more than one) to either of the existing documents. 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:

```
<img src= "http://www.cs.stir.ac.uk/fred.gif " alt= "Picture of F. Bloggs">
```

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 (sometimes) 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, using your browser, you should try copying the images to your home (web) folder (navigate to the page showing the image and then right-click on the image you want and chose “Save target as...” or “Save image as...”) and using simple relative pathnames in the **img** tag. If you are unsure please ask.

Try putting several images on the same line ie `` tags one after another. Watch the browser rearrange them as you make the window narrower. Try forcing them onto separate lines using a line break tag `
`. Put some text on either side of an image, and try the inline style **vertical-align** with values of "top"/ "middle" / “bottom” in the `` tag to see how this affects the placement of the image relative to the text.

With another image, try adding the inline style **float**, with values of “left” or “right” and see what happens.

Styling your Home Page

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.

CHECKPOINT [CSS]

Please demonstrate all the HTML and CSS pages you created (**demo1.css** with **document1.html**; **demo2.css** with **document2.html**; **mystyle.css** with the corresponding styled version of your home page, **index.html**). Attempt to *validate* your pages on the W3C web site (<http://validator.w3.org/>) *before* you demonstrate your pages. Remember that CSS style sheets can be validated at: **jigsaw.w3.org/css-validator/**