Week 3 Exercises

The aim of this week's exercises is to gain some experience in style Cascade, Inheritance and Specificity whilst formatting a given text

Time Required

These exercises should take you 1½ hours at the most, depending on your experience, and they are intended to be completed in class time.

Exercise 1: Installing the sample web page

I have created an html file for you with a blank stylesheet linked to it.

1. Open a browser and locate the file at **`http://www.headroomgate.co.uk/style/downloads/week3.zip`**
    a. Download and unzip this file into your week 3 folder.

In the zip file you will find **`ex1wk3.html`** and **`ex1wk3.css`**

2. Return to the browser and look at the html file. You will find a page with the heading 'Tablet Buying Guide'. The page is completely without any style declarations at the moment and therefore being rendered by the browser's default style. The rest of this evening's exercises will be involved with progressively adding style to the document in order to experience some of the issues of Cascade, Inheritance and Specificity, and text styling possibilities.

You might find it helpful to open the slides from this evening's lecture in order that you can remember the syntax for the various styling declarations discussed.

3. To do this, navigate to **`www.headroomgate.co.uk/style`**, select Week 3 and open the slides pdf by clicking on the icon.

Exercise 2: Styling the Body and the headings

The first selector you will style is the **body** tag.

1. Open the html file in Dreamweaver
    a. Click on the stylesheet in the Related Files Toolbar. As last week you will find that the stylesheet is currently empty except that the character set is declared as 'utf-8' and the standard comment is in place.

2. In the stylesheet
    a. Insert the **`body`** selector and the declarations that say you want a font-family (you can choose whichever font family you like, I chose my usual Arial, Helvetica, sans-serif)
    b. Set the font size to 0.9em
    c. Set a background colour of # CEFAF6.
    d. Save the file and take a look in the browser. You should see the text styled in the font of your choice, at a slightly smaller size and with a light blue background.

3. Return to the stylesheet
    a. Insert an ID below the **`body`** rule.
    b. Give the ID the name of **`container`** as you did last week.
    c. Continue to style the **`container ID`**
        i. With a width of 700px
        ii. Top and bottom margin of 20px
        iii. Auto setting for the left and right margins – can you recall the shorthand for these settings, if so then insert them.
        iv. Insert a padding of zero top and bottom, 10px left and right using the shorthand if you can.
        v. Use a background colour of #FFC
        vi. Finally give the ID a solid, black border of 1px.

4. Save the stylesheet and return to the browser, you should find that the display is now in a box centred in the viewport with a small margin at the top and bottom. The background of the central display should be a pale colour. The text probably feels a little too large currently, especially the headings.

5. Return to the style sheet
    a. Place these typography styles in the style sheet after the body rule and before the **`container ID`** rule.
    b. Set the **`h2`** to 1.3em
    c. The **`h3`** to 1.2em.
    d. To style the **`h1`** heading use the shorthand version for font styling,
    e. Type: **`font:bold 1.5em Verdana, Geneva, sans-serif;`** Notice that I have chosen a different font for the **`h1`** only.
    f. Save the file and return to the browser. When you refresh the display the headings should be starting to look a little more in balance with the paragraphs.

6. In order that you can see inheritance at work return to the stylesheet
    a. Set the colour value in the **`body`** selector to dark grey (**`#333`**).
    b. When you save the file and view the result in the browser you should see that all the text on the page is now dark grey. This is because all the selectors have inherited the colour value of the **`body`** selector (if this change is not apparent because the grey is too dark and so looks black, try using the colour **`#666`** which is a lighter grey).

c. If you prefer black then you can of course reset the colour now that you have seen the effect.
d. Style the **h1** in a similar way to last week.
e. Start by giving the **h1** a background colour of **#F00**. When you view the changes in the browser you will see that all the h1s have a red background. In actual fact you don't want the h1s that show the title of the tablets to be red but you will attend to that shortly, just ignore them for now.

Notice how the background colour does not extend to the edges of the display because of the padding in the div with the **ID** of **container**.

7. To overcome this return to the stylesheet
   a. Give the **h1**
      i. Top margin of zero,
      ii. Left and right margins set to minus 10px
      iii. Bottom margin set to 10px.
   b. Save the file and view the result in the browser.
   c. The effect is closer to filling the width, but the appearance is far from ideal.

8. Return to the stylesheet and give the h1 a padding of 6px top and bottom and 14px left and right to compensate for the **container** div's padding and then view the result in the browser you should see that the h1 now fills the space and looks better placed inside its coloured background.

9. Finally, return to the stylesheet and set the **h1** to a colour of white to cancel out the global colour set by the **body** rule.

Exercise 3: Formatting the tablets

Return to the stylesheet in Dreamweaver

1. Between the **h3** rule and the **container ID** rule
   a. Insert a class selector for the class '**tablet**'
   b. Set the class to a width of 90%
   c. Top/bottom margin of 5%
   d. Left/right margins of auto.
   e. Save the file and return to the browser
   f. You should see that the text for the three tablets have moved inwards on the page with the width reduced by 10% of the overall display width.

2. Return to the stylesheet
   a. Give the **tablet** class:
   b. 1px black border
   c. A background colour of white
   d. Text colour of #090
   e. View the result in the browser and you will see all three tablets set to the availability key of green to show that they are currently available; the default.
   f. You will also see that the heading is out of line with the border of the div.
   g. Go back to the stylesheet and give the **tablet** class
      i. Top padding of 6px
      ii. Left and right padding of 10px
      iii. Bottom padding of zero.

h. Insert a descendent selector for the **`h1`** inside the **`tablet`** class
    i. Give it a top margin of -6px,
    ii. Bottom margin of 6px
    iii. Left/right margins of -10px (can you remember the shorthand for this?).

i. View the result in the browser. You should see that the h1 is now sitting inside the border.
j. Finish off the **`h1`** inside the **`tablet`** class by giving it a background colour of black.
k. Make the font size 1em and give it a variant of small caps.
l. Before you view the result take a moment to work out what the presentation of the page will now be like.

Did that surprise you? The change in size of the **`h1`** is due to the descendent selector declaration overriding the global **`h1`** size setting and making the font the same size as that defined by the **`body`** rule.

3. Return to the stylesheet
    a. Set the size of h2 inside the tablet to 1em.
    b. Size the paragraphs and list items to 0.9em.
    c. This will give the overall presentation a more balanced appearance.
    d. You could achieve the styling of the paragraphs and the list items by grouping the two descendent selectors thus: **`.tablet p, .tablet ul li {…};`**

4. Finally before you leave the tablet
    a. Set the paragraphs and list items in the tablet to be aligned as ‘justify’.
    b. You can do this in the grouped descendent selectors you just created.

Exercise 4: Looking at line-height

Return to the stylesheet

1. After the descendent selector rule you just created
    a. Insert a class selector for the **`key`** class
    b. Indent the text by 10px
    c. Save the stylesheet and return to the browser and before you refresh the view scroll down to the bottom of the page where you will find the Key.
    d. When you refresh the view you should see all the text, including the title, move over to the right by 10px.
    e. Return to the stylesheet and add a further declaration that sets the line height to 0.2.
    f. When you view the changes in the browser you should see that the vertical space between the lines of text and the heading in the Key has been reduced quite significantly.

Exercise 5: Word and letter spacing

This exercise looks at the company name at the very bottom of the page.

1. In the stylesheet insert a class selector with the name of **`company`**
    a. Place it after the key style rule you just completed.
    b. This rule is going to reduce the space between the letters of the company name by first setting the letter spacing to -0.5px.
    c. Save the stylesheet and view the change in the browser.
    d. The spacing between the letters should now be quite significantly reduced;
    e. Try setting different values to the declaration – what happens when you set it to minus 1 and then to plus 1 (no plus sign required just remove the minus sign)?

f. Leave the value set at whatever gives you a satisfactory result, then set the font style to italics and view the result.
g. You might want to adjust the letter spacing when the italics have been applied. It is all a matter of personal taste.

One point you might have considered is how the style rule got applied only to one word. If you look at the HTML you will find that the class of company is an attribute of the span tag which surrounds just that one word plus its TM symbol. You may well see that in fact the TM symbol has not been italicised, if this is the case it is probably because the PC does not have a font for superscript characters.

Turning now to the paragraph that contains the company name; this is called trade.

2. Insert an ID selector below the company rule you just created
   a. Give it the name of **`trade`**.
   b. Give the paragraph a width of 100%
   c. A background colour of #F00.
   d. Save the changes and view them in the browser.

   e. You should see that the paragraph has a background colour of red and despite saying 100% width it does not extend to the edge of the container div as you might expect. This is because the container div has a padding setting of 14px left and right, more about that in a moment.

   f. Look particularly at that background colour; it does not extend beyond the ascenders and descenders of the text, this is because the line height is currently the browser's default setting of 1; you will change that in a moment.

3. To see the font declaration shorthand at work you will change fonts for this rule
   a. Make the font bold,
   b. Size 1.1em
   c. Line height of 1.5.
   d. Make the font Verdana, Geneva, sans-serif.
   e. This declaration is:

```
font:bold 1.1em/1.5 Verdana, Geneva, sans-serif;
```

Taking this one bit at a time, first the font is set to bold, then the size is 1.1em and the line height is 1.5, notice the forward slash that conjoins these two settings. Finally you set the font. Note you are only changing the font by way of demonstration it could just have been the default Arial declaration, or some other font family.

Save the changes and view them in the browser. You should see that the font is now 10% bigger and the vertical space around the text has grown as a result of the line height. Try a few changes in these settings (no negative values allowed) to understand what is happening.

4. Returning to the stylesheet and still in the **`trade ID`** rule
   a. Give the font a colour of #FFC
   b. Apply a word spacing value of 10px.
   c. View the result in the browser and you will see that the space between each word in the paragraph has been increased by an extra 10px.
   d. Again try this with different values to see the effect, negative values can be quite spectacular but equally very useful if applied carefully, i.e. only very small values.

You have now styled the font of the paragraph but the placing is still unattractive.

5. Start to modify this by giving the **trade ID** rule
    a. A left margin of -10px and that will move the paragraph over to the left of the container div.
    b. if you then apply a bottom margin of zero the paragraph will move down to the bottom of the container div.
    c. There is still a space off to the right and the text inside the paragraph is still set very tight as in there may be space from the line height but the text is hard over to then left
    d. Give the **trade** ID rule a padding setting of 10px all round and the result should be more pleasing.
    e. Notice how the padding is applied in addition to the line height which makes the text look a tad too far over to the left, especially when seen in relation to the Key above.
    f. To correct this you could just increase the left padding value, to push the text over to the right.

Exercise 6: Formatting the Tablets

Returning to the tablets; currently these all show with green text which the Key tells you means that the items are currently available. Each of the tablets is also showing the text 'this month's special offer' with a lower case 't' for this. The Key says that items on offer are coloured blue, so you need to do some styling to make this happen easily.

1. Open the stylesheet if it is not already open and locate the **h3** rule.
    a. Immediately below that start a new rule for the class **offer**.
    b. In this rule insert a declaration that sets the first letter of the element to a capital letter. This needs a pseudo-element of first-letter and a declaration of text-transform to capitalize:

```
.offer:first-letter {
      text-transform: capitalize;
}
```

2. Save the changes and view the result in Firefox. You should see that the lower case 't' has now become a capital letter.

Spelling!

This is one of those occasions where the use of American English can cause a problem – the word capitalize must have a 'z' not an 's'.

The reality is of course that you don't want to always see this text. If it is not on offer or is discontinued then you don't want to see it. The default display is green which means the tablet is available, but it is not on offer.

3. To start to manage this you first need to make the default contain a rule in the stylesheet that the offer text does not show.
    a. Create a new class rule for **offer**
    b. Set it to display none:

```
.offer {
      display:none;
}
```

When you save this change and view it in the browser the text has disappeared. This completes the default behaviour, now you should set a rule for the tablet which is on offer. If you look at the HTML of the exercise you will see that the first tablet has no ID, so is available by default, whereas the second tablet has an ID of offer which means that it is on offer and should be styled accordingly. Finally the third tablet is discontinued and therefore has an ID of discontinued. Now you will set the styling to respond to these IDs.

4. Back in the stylesheet and below the **`offer`** class you just created
   a. Insert an **`ID`** selector of **`offer`** and give it a colour of #00F.
   b. Save the change and view it in the browser.
   c. The second tablet (the one with the ID of **`offer`**) has now turned blue, which is correct colouring as per the Key.
   d. However, the text that says it is a special offer is not showing.
   e. To change that you can add a descendent selector to the ID selector such that it styles the offer class when it is a child of the offer ID.
   f. This probably sounds very confusing but this is the style rule:

```
#offer, #offer .offer {
      color:#00F;
      display:block;
}
```

To look at that in detail: the style is to be applied to the element which has the ID of offer, which contains the details of the second tablet. In addition the rule is applied to the class of offer which is a child of the ID offer. The declaration is setting the colour to be #00F (the blue) and the display setting to be block which is the opposite to none, i.e. display the element.

When you save this change and look at the result in the browser you will see the text ‘This month's special offer’ displayed in blue inside the second tablet’s display. This is a classic example of a style conflict being used for a specific effect; you set the class not to display as a default and override that setting as display block when the conditions require it. This kind of styling is mostly used when the content of the elements are unknown because they are being supplied ‘on the fly’ at runtime, i.e. coming from a database as the result of a query when the visitor accesses the page.

One final change to make before you leave is to set the display of the discontinued tablet.

a. Return to the stylesheet and below the last style rule insert an ID selector with the name of **`discontinued`**.
b. Set the text decoration to line-through
c. Give it a colour of dark blue (#006).
d. When you view these changes in the browser the third tablet should be showing as discontinued because the details have been struck through and the display turned to dark blue.

In summary

Have you noticed how the colour of text in the h1 of the tablets never changes even though the body text and the h2 headings do; and in the discontinued tablet the text is struck through. Can you explain why the colour of the h1 does not change even though the text and the h2 do? This is inheritance at work; the base colour for all text has been set in the body style rule, i.e. it is #333, or whatever you finally decided to make it. The h2 and h3 style rules do not contain any colour settings so they inherit those of the body. The h1 however does contain the colour declaration of white and it retains that throughout the stylesheet because it does not get overruled. The body rule colour declaration does get overruled when the colour is changed by the tablet class and by the two IDs offer and discontinued, and this affects the h2 because it does not have its own specified colour.

This evening’s exercises have demonstrated the cascade and inheritance whilst also giving you an example of every setting that you can change to style and colour text.