CSS: The language of style

The World Wide Web was originally envisaged as a way to present information simply, without elaborate formatting. As the web became popular, web browser manufacturers began adding HTML codes to control colours, typefaces and layout. However, with browser manufacturers inventing their own codes, the HTML language became polluted with incompatible formatting tags. Even worse, the structural role of HTML was becoming diluted with codes controlling presentation rather than structure.

To order to this chaos, a new approach was needed. HTML would be kept as a structural language, and new language developed to handle presentation. This was CSS, or Cascading Style Sheets. Style sheets would control formatting, colour, positioning and other presentational aspects, while HTML would define the ducument's underlying structure.

One of the most powerful features of CSS is the ability to control the appearance of any number of HTML files from a single CSS style sheet. This allows the formatting for an entire web site to be changed by changing the linked style sheet.

Style sheets are collections of rules. Each rule controls the appearance of a specific part of a web page. Every rule has the same basic structure. Here is an example:

```
h1 {font-size: 100px; color: #FF0000;}
```

Note the colour codes: use the web colour value from Photoshop's colour picker. The example above uses hexadecimal (base16) notation. You can also use decimal RGB notation, shown below:

```
h1 {font-size: 100px; color: rgb(255,0,0);}
```

The selector
The selector specifies what elements the rule will apply to. In the example above, the rule will apply to all the H1 (heading level 1) tags in the HTML document.

Following the selector are one or more declarations. Declarations describe the style that will be applied to the elements specified by the selector. Declarations take the form of two parts: a property and a value. Properties describe what we can style, such as colour, typeface and size. Values describe how they will be styled. In the example above, the property color makes it clear that we are talking about the size of the h1 element. The value 100px says that the size of the h1 will be 100 pixels.

A rule may have multiple declarations, as in the example.

Multiple selectors are also allowed, although we will not be investigating these in class.

Using style sheets
When using style sheets, one of the decisions you will have to make is where to put the rules. There are a number of options, but only two of these are commonly used.

Option 1: in a separate linked file
By placing your style sheet in a separate file, you need to only have a single style sheet for the entire web site. You do this by linking every page in your site to the css file, as follows:

```
<link href="global.css" rel="stylesheet">
```

In our examples, the file "global.css" contains the style rules that govern all pages linked to it. make sure only style rules, and not HTML tags,are in your .css file. The contents of the file will look like this:

```
body
{
      font-family: "lucida grande", verdana, geneva, sans-serif;
      color: #555555;
}


h1
{
      font-size: 100px;
      color: #111111;
```

3

```
        font-weight: normal;
        border-style: solid;
        border-width: 1px;
        border-color:#FF0000;
}
```
...and so forth, for all the rules you choose to define.

Option 2: within the <head> element
You can choose the put your style sheet in the <head> element of your HTML document, as follows:
```
<style>
h1
{
    color: #FF0000;
    font-size: 30px;
}
</style>
```

This is the simplest method, but means that each HTML document will need to have its own style sheet. Since having one style sheet to control an entire web site's appearance is one of the main advantages of CSS, this method is usually used only by beginners looking for a quick way to test CSS, or for one-of-a-kind pages that use different formatting than other pages in the site.

Selectors
CSS allows a number of different ways of specifying which rules are to be applied to which elements. The simplest is the element selector. Element selectors specify the style for all instances of an HTML tag. Any HTML tag can serve as an element selector. For example, if you wish all the paragraphs of text to be blue, you could use:
```
p {color: #000099;}
```
as the style rule.

However, by using only element selectors, you would have no way to make one paragraph on a blue and the remainder grey, since the element selector applies to all matching elements.

Getting specific: class and id selectors
One way of specifying which elements you wish to style is the class selector. First, define a class in your style sheet:
```
.highlight {background-color: #FFFF00;}
```

This creates a class called highlight. Note the leading dot. Class names are up to you to create - you can call the class anything, as they have no predefined meaning.

Then, use the class property in your HTML code to specify how to style that particular element
```
<p class="highlight">
```
A paragraph controlled by the "highlight" style`</p><p>` and now some text that looks like all the other paragraphs.`</p>` *Note the lack of the dot in the class name when used in your HTML code.*

Any element tagged as a member of the "highlight" class will be assigned the colour blue. `<hr class="highlight" />` for example, will create a blue horizontal rule.

To be even more specific, use id selectors. The syntax is similar. Define an id in your style sheet:
```
#menu {color: #FF0000;}
```
note the # sign

Then, in your HTML, tag an element with that id:
```
<p id="menu">
```
A paragraph controlled by the "menu" style`</p><p>` and now some text that looks like all the other paragraphs.`</p>`

Use class when more than one element will have that style applied to it. Use id when you want to identify a unique element on a page.

CSS basics: type

Some sample property/value pairs:

Controlling type
```
font-family: Arial, Helvetica, lucida, sans-serif;
font-size: 10px;
font-weight: bold;
```

Controlling colour
```
color: #FF0000;     You can look up the colour codes in Photoshop's colour picker.


background-color: #FFFFAA;     Changes the background colour of an element
```

Controlling borders
```
border-width: 4px;
border-style: dotted;  (some options are none/dotted/dashed/solid/double
```

CSS basics: positioning

CSS can be used to position elements, as well as style text. The normal way to do this is to tag the element with an id, and specify the position of that ID in a style sheet.
```
#mainimage {
position: absolute;
top: 200px;
left: 300px;
}
```
Then, in your HTML code, use: `<img id="mainimage" src="squid.jpg" alt="A squid">`

In CSS, px stands for pixels, and is measured from the top left corner of the browser window. In the example above, the image with the id mainimage will be positioned 200 pixels down from the top of the page, and 300 pixels to the left of the screen's right edge. (A sketch on a sheet of graph paper can be useful here.)

If you mix items for which you have specified positions with items you have not, strange things may happen.

Applying styles

To apply a style to a range of text, such as a phrase, use the span tag with an associated style.
This is `<span class="highlight">` some highlighted text`</span>` using the span tag with a style, class or id.

To apply a style to a range of other elements, such as multiple paragraphs of text, use the div tag. You can position elements by wrapping them in `<div>` tags, and using the positioning options shown above. A div creates a container in which text or images can be positioned as a single unit. In the example below, the div with the id of "content" would be positioned as a single block:

```
<div id="content">
<h2>Choose a section</h2>
     <ul>
     <li><a href="1350a/">Foundation computer 1350A: Thursdays</a></li>
     <li><a href="1350b/">Foundation computer 1350B: Fridays</a></li>
     </ul>
</div>
```

Resources

http://wendypeck.com/css101.html
https://developer.mozilla.org/en-US/docs/Learn/CSS