CSS 101

WHY CSS?

A consistent system was needed to apply stylistic values to HTML elements. What CSS does is provide a way to attach styling like **`color:red`** to HTML elements like **`<h1>`**. It does this by defining a properties like **`color`** and **`font-size`** and a means to associate these with elements by targeting the elements' type, class, id or attributes.

Before the introduction of CSS this was done by special HTML elements that specified an attribute to be applied to the HTML within it. This resulted in huge amounts of work any time that styling was changed.

WHAT IS CSS?

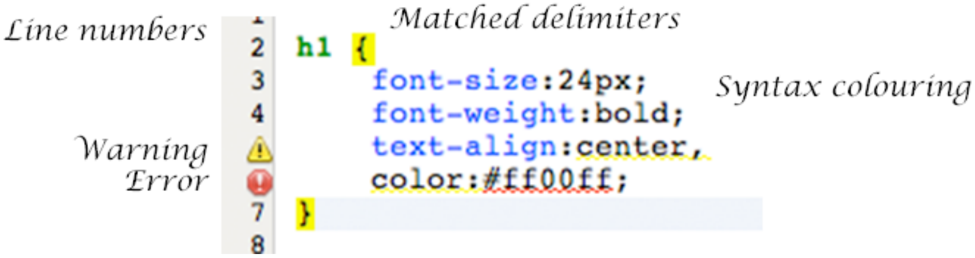
CSS stands for Cascading Style Sheets. It provides a way to customize the styling on HTML documents. It replaces the built in HTML styles such as **`font`** and **`width`** that were used in earlier web sites. CSS provides ways to set the layout, fonts, background colours and image, and many other visual aspects of the document or website.

WHAT IS A STYLE SHEET?

A CSS style sheet is a plain text file consisting of a number of rules that describe the styling of one or more HTML files.

No special editor is required to edit these files, although many editors such as NetBbeans, Coda and DreamWeaver provide assistance such as flagging errors and helping you chose the properties and values that make up a styling rule.

Editing CSS in Netbeans



```
h1 {
    font-size:24px;
    font-weight:bold;
    text-align:center,
    color:#ff00ff;
}
```

CSS may be applied to an HTML by including a style sheet from an external location, by including the style sheet directly in the header of the document. Style sheets may also be included by the reader as options in her browser or by the browser itself.

CSS CODE

The code in a style sheet is made up of rules of the following types

@ RULES

these start with an `@` character

the most common kinds are **`@import`** and **`@media`**

`@import` copies the content of one styelsheet into another

`@media` specifies which media types the enclosed styles apply to. These are used to support print style sheets, stylesheets for special readers and responsive stylesheets that have breakpoints set to sizes defined by the content or by screen types.A style sheet consists of one or more rules.

COMMENTS

Comments are coded like the multiline comment in C. **`/*`** starts a comment and **`*/`** ends the comment.

STYLING RULES

all other lines in the style sheet are styling rules.

styling rules consist of a selector and a number of property values eg:

```
p.callToAction {color:red; font-size:24px;}
```

this sets the foreground colour to **`red`** and font to **`24px`** for all paragraphs with class **`callToAction`**.

INLINE RULES

CSS may be applied directly on the HTML elements using the style tag.

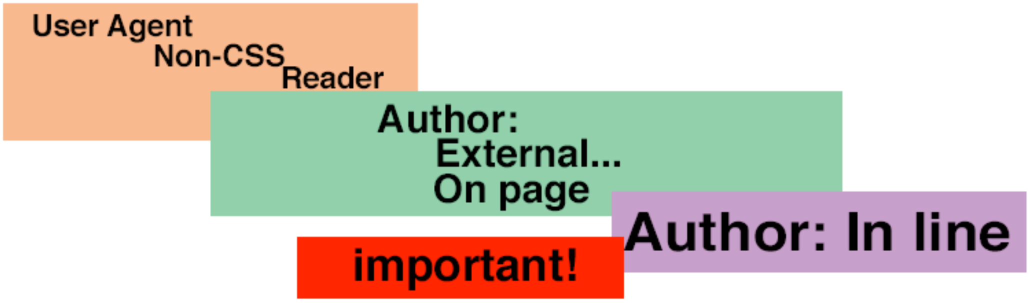
```
<h2 style="color:blue">Blue heading</h2>
```

This sets the color of this particular **`h2`** to blue, and overrides anything set in style sheets. This is a very powerful feature, and shouldn't be used except when there is no other means available or a very local change is exactly what's required. Styling this way means that you need to change the style everywhere that it occurs.

WHAT IS THE CASCADE?

The styles are said to "cascade" in that later styles override earlier styles. If more than one selector occurs in the style sheets that have the same priority the last one will be used. To understand what “last” means you need to know the order that style sheets are included. The order is based on the source of the rules.

The Cascade



User Agent - The browser, these styles describe the standard appearance for a particular browser. These are the first styles included and most page authors will override this using a set of styles called a "reset".

Non-CSS - HTML tags and attributes, these should not be present, but can be overridden by the author’s CSS if they are in the HTML.

Reader - The user of the browser may set a style sheet to be applied before those provided by the page author, these often relate to accesability. These are overridden by styles provided by the author unless they are marked **`important!`**

Author- The author of the page, this is the most common source. Styles can come from:

External style sheets - These are imported into the HTML document using the link tag eg.

```
<link rel="stylesheet" type="text/css" media="all"
href="/css/mystyle.css" />
```

Most of the styles in a website should be Author external styles.

`On page style sheets` - These are included in the **`<head>`** of the document. they provide a way to have styles that are used by only one document. They override anything included from external files. They are found inside a **`<style>`** tag.

In line styles - styles provided for a particular HTML element. These are the highest priority styles. These are found inside **`style=`** attributes.

important! - individual declarations can be flagged as important. These override any other non-important rules. Reader important! declarations override Author important! ones, this is the opposite of normal order. These should be avoided, except for testing.

CSS STYLING RULES

These are what most people think of when you use the term CSS.

RULE SYNTAX

a styling rule has the format:

```
selector {property:value;..}
```

The **`property:value`** pair is called a declaration and there may be several of these in a single rule eg:

```
        h1 {

             font-size:24px;

             font-weight:bold;

             text-align:center;

             color:#ff00ff;

        }
```

This rule has a selector of **`h1`** which says that it applies to all heading 1 tags in the document.

It contains 4 declarations, the first applies font size **`24px`**, for 24 pixels.

the second a bold font weight,

the third centers the text,

and the forth sets its color to rgb(255,0,255) or magenta.

this could also be written rgba(255,0,255,1) with transparency, 1 is opaque

There are hundreds of properties and many of these have several keyword parameters or support sizes that may be specified in pixels, ems or as percentage. Some properties such as color have their own special forms for there values.

sizes defined by the content or by screen types.

CSS styling rules start with a selector and define a collection of style declarations that specify values to be assigned to properties of the HTML elements that match the selector.

SELECTORS

Selectors provide a way to identify a group of HTML elements, such as all paragraphs, or all articles with a class of ‘blog’. The most frequently used selectors are:

ELEMENT SELECTOR

these are simply the name of the elements, eg. **`h1, p, div`**, or **`article`**.

ID SELECTOR

these start with a hash symbol and specify the value set on the element in its **`id=`** attribute, eg. **`#top-menu`** selects **`<div id="top-menu">`**

CLASS SELECTOR

these start with a period and match any value set in the **`class=`** attribute of the element. eg. **`.error`** will match element **`<p class="form error">`** and **`<span class="error">`**

OTHER SELECTORS

there are several other kinds of selectors that I'm not covering here. These include Attribute, General, Pseudo-class and Pseudo-element selectors. These selectors are not as frequently used as the the element, id and class selectors.

COMBINING SELECTORS

several selectors can be combined together to get more complex element matching

GROUPING SELECTORS

two or more selectors separated by commas form a group, eg. **`h1, h2, h3`** matches all heading 1 2 and 3 elements.

DESCENDANT SELECTORS

two selectors separated by blanks select elements that are descendants of the first selector, eg. **`form h2`** selects heading 2 elements contained within a form element

CHILD SELECTORS

two selectors separated by the greater than symbol select elements that are immediate children, eg. **`form>h2`** selects heading 2 elements that are immediately within the form, but not those that have some other element in between

SIBLING SELECTORS

two selectors separated by a plus select the adjacent sibling of the the first selector, eg. **`h1 + p`** selects paragraphs that immediately follow a heading 1

CHAINING CLASS AND ID SELECTORS

multiple class and id selectors can be chained together without any separator. These classes and ids must all be present to match. eg. **`p#notice.urgent`** will select paragraphs with **`id="notice"`** and **`class="urgent"`**.

LET'S BE SPECIFIC

CSS ranks selectors by how specific they are. It calculates a value for "specificity" and uses the most specific selector that matches the element. This calculation is based on the number of each type of selector used. Four numbers are used:

inline - 0 if not inline, 1 if inline

id - number of ids in the selector

class - number of classes in the selector

element - number of elements in the selector

a value of 1,0,0,0 is higher than 0,1,1,1

This seems complicated, but the rule is simple: inline > id > class > element. If they are the same type the count matters.

PROPERTIES AND VALUES

The actual styling is done by setting values for properties.

PROPERTIES

Properties describe various styling attributes of an element, things such as the color, font attributes, background or layout. Property names are the left side of the style declarations that make up a rule. eg. **`color:red`** sets the property foreground color to red. Most property names are obvious, some require a bit of design knowledge and some are strange, eg. the **`line-height`** controls leading, but **`line-height`** defines the full height of the line, not just the extra space provided by strips of lead.

some common properties are **`color`**, **`background-color`**, **`font-size`**, **`font-family`**, **`border`**, **`margin`**, **`padding`**, **`float`**, **`display`**, **`overflow`**, **`width`**, **`height`**, **`z-index`**, and **`position`**.

VALUES

values appear to the right of the colon in a style declaration. The kind of values that may be provided depend on the property. Most property values are straight forward. Some properties take several different kinds of value. eg. **`color:red`**, **`color:#ff0000`**, and **`color:rgb(255,0,0)`** are equivalent.

there are quite a few properties that accept values describing a size or distance. These values may be specified in absolute terms such as **`in`**, **`cm`**, **`pt`** and relative units such as **`em`**, `%`, and **`ex.`** **`px`** is strange, it’s officially relative, but you should treat it as absolute most of the time. There is a value called pixel density that defines how many physical pixels make up a logical pixel.

CSS VERSIONS

CSS 2.1

CSS 2.1 is the version of CSS that has been in use since 2004, finally becoming the W3C recommendation in June 2011. It is supported by all modern browsers and is safe to use.

CSS 3

CSS 3 isn’t a single large specification like CSS 2 was. It consists of a number of smaller modules. Some such as the media queries are stable enough to use at present.

there are “polyfills” available that make many CSS3 features work in older browsers.

OTHER RESOURCES

BOOKS

The best book on CSS is the latest edition of Eric Meyer’s “Cascading Style Sheets, The Definitive Guide”. At present it is not up do date on CSS3, but it provides the best information on CSS basics and CSS2.

ONLINE

css-tricks.com - very useful when things get difficult

w3schools.com - lots of information, not always the most accurate

jigsaw.w3.org/css-validator/ - CSS validation service

csszengarden.com - wonderful examples, great for inspiration