CO456
Web

- most materials adapted from ***Moseley (2006)***, Chapters 2, 3 & 4 -

Week 3

Recap (tables, images, forms)

… and some CSS …

Module schedule

Wk.	Lecture/subject area(s)	Practical	Reading (Moseley, 2007)
1	Introduction How the Web works	Internet/Web definitions and HTML report	Ch 1 (The way the Web works)
2	HTML 1 (Introductory - inc. lists and hyperlinks)	HTML	Ch 2 pp 24-36 (HTML)
3	HTML 2 (inc. tables, images and forms)	HTML	Ch 2 pp 36-48 (HTML) Ch 3 (XHTML and frames)
4	CSS 1 (Introduction and core CSS principles)	CSS – introductory styles, embedded styles.	Ch 4 pp 76-96.
5	CSS 2 (Positioning elements).	CSS– using IDs, classes and layout control.	Ch 4 pp 97-103.
6	CSS 3 (Advanced layout & navigation)	CSS – using CSS to produce button-like navigation from HTML list elements. (CW2a to be demonstrated).	Specialised articles.
7	JavaScript 1 (Fundamentals, variables)	JS – foundation constructs.	Ch 5 pp 108-116
8	Guided Learning Week	Consolidate Internet & W3 knowledge and HTML & CSS skills.	Review Ch 1 to Ch 4.
9	JavaScript 2 (Functions, branches, loops).	JS – calling functions.	Ch 5 pp 117-124.
10	JavaScript 3 (Objects and the DOM).	JS – manipulating the DOM.	Ch 6 126-139.
11	JavaScript 4 (Forms and validation). And DHTML	JS– validating user completed forms.	Ch 6 139-145, Ch 7.
12	HTML***5***, CSS***3***, - media, forms, gradients, SVG ('Edge') and other enhancements	Web frameworks taster session 1	See practical sheets for information sources
	Vacation		
13	Advanced HTML5, CSS3 & JS frameworks (e.g. jQuery, jQuery Mobile, Box2DWeb)	Web frameworks taster session 2	See practical sheets for information sources
14	Assignment workshop 1	Assignment workshop 1	N/A
15	Assignment workshop 2	Assignment workshop 2	N/A

Tables - in summary

- Covered last week
- Key tags are <table>, <tr>, <th> and <td>
- Common <table> attributes are
  - border="1"
  - width="n pixels/n%"
  - cellpadding – in pixels between inner cell wall and cell content (now superseded by CSS)
  - cellspacing – in pixels between cells (now superseded by CSS)
  - rules="all"
- Two useful <th> and <td> attributes for irregular tables are
  - colspan="n cols" – merges cells horizontally
  - rowspan="n rows" – merges cells vertically

Images - in summary

- Covered by independent study last week – brief review below
- The image tag, source, height (opt), width (opt), alternative text (opt)
  - <img src="url/path/file.ext" height="px/%" width="px/%" alt="My pic">
- Can also be used for background
  - <body background="duck.jpg>
- … and for hyperlinks
  - <img src= …>
- … and embedded as background styles in other tags
  - <form action="processor.php" method="post" style="background-image:url(duck.jpg)">

Forms - in summary

- Covered by independent study last week – brief review below
- The form tag with action and method attributes
  - <form action="processThisPlease.php" method="post">
- The input tag for collecting user data – many “type” values
  - <input type=“text/password/checkbox/radio/file/submit/button/reset/hidden”>
  - <body background=“duck.jpg>
- And other tags including
  - **Text areas** <textarea name=“comments” rows=“10” columns =“70” wrap=“wrap”>a message can go here</textarea>
  - **Drop down menus** <select><option value=“slct1”>Opt 1<option><option value=“slct2”>Opt 2<option><option value=“slct3”>Opt 3<option></select>

Screen capture of image and form example

Examples of embedding an image and a form - adapted from Moseley (2006)

This is an image with a hyperlink to a suitable site ...

log in now

Please enter your use name

Your password is ...

Choices?

Favourite colour green

My favourite fruit is: Apple

I live in a: House Flat Bedsit Caravan

My upload file is: Browse...

Leaving now?

Write over this in here ...

You can leave a message here:

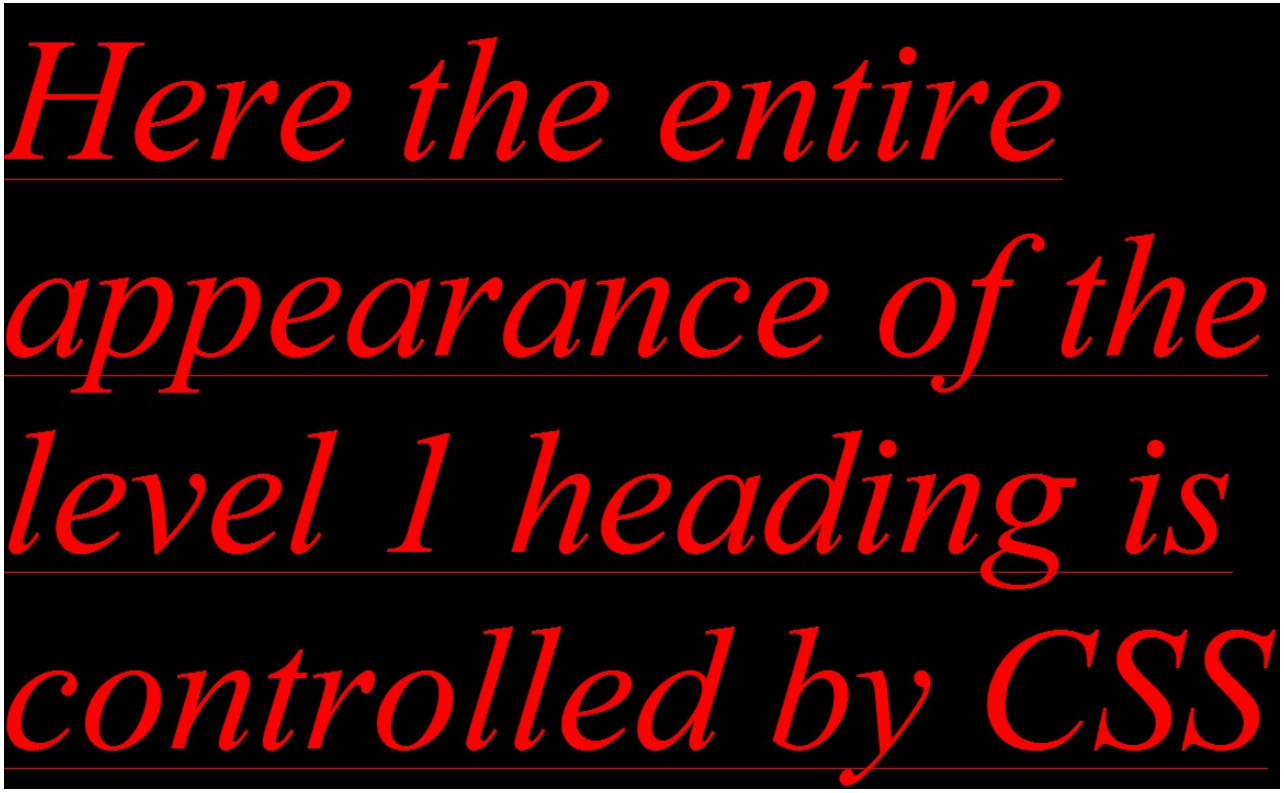
GO! STAR AGAIN!

CSS

- Original HTML was concerned purely with ***structural*** markup
- Early 1990s HTML expanded to include ***presentation*** markup
- Confusion of structure and presentational elements
  - Difficult to engineer, maintain and reuse
  - Poor search engine indexing
  - Inconsistent browser interpretation
- CSS1 standardised by the World Wide Web Consortium (W3C) in 1996
- CSS2, an extension of CSS1 w/o major changes, standardised in 1998

- An example: the html
  - The html: `<h1>The main heading</h1>`
  - The CSS to control the html: `h1 { color : red ; font : italic  10em Times, serif; text-decoration: underline; background: black; }`

Screen capture of CSS style of <h1> tag



CSS – rules

- The basic rule
  - selector {property 1 : value 1 ; property 2 : value 2 ; …. }
- The selector is the identifier of the element followed by a list of paired property values in curly brackets
  - e.g. body {color : yellow }
- Values with multiple words must be enclosed in quotes
  - p {font-family : “sans serif”}
- Paired property values are separated with semicolons
  - p {text-align : center ; color : red }
- It is possible to apply the same properties to many selectors – in this case level 1 and 2 headings and paragraph tags
  - h1, h2, p { color : red ; font : italic 10em Times, serif; text-decoration: underline; background: black; }

CSS - classes

- You can create your own style rules AND you may also have more than one rule for the same element
  - p.right { text-align : right ; color : red ; font : italic 4em Times , serif ; background: yellow}
  - p.left { text-align: left; color : green; font : 3em Arial, sans-serif ; background : aqua}
- To use
  - This paragraph is right aligned
  - … and this paragraph is left aligned

This paragraph is right aligned

… and this paragraph is left aligned

- A useful technique is to use ***anonymous*** classes that may then be applied to multiple tag types – simply don't specify the tag
  - .left { text-align: left; color : green; font : 3em Arial, sans-serif ; background : aqua}
- To use
  - <h1 class="left">This heading is left aligned
  - … and this paragraph is left aligned

This heading is left aligned

.. and this paragraph is also left aligned

CSS – the “id” selector

- Similarly to classes - you can create styles for “identified” elements
  - p#bluepara { text-align : center; color : darkblue ; font: 3em Arial, sans-serif ; background : aqua}
  - p#greenpara { text-align : center; color : darkgreen ; font : 3em Arial, sans-serif ; background : lightgreen}
- To use
  - This is a BLUE paragraph
  - … and here is a GREEN one

This is a BLUE paragraph

… and here is a GREEN one

- The difference between class and id is that class may apply to many parts of one page but id can be used once only
- Although it is also possible to have anonymous identifiers as follows …
  - #myBlueStyle {color : darkblue ; background : aqua}
- … strictly, style rules say one should not apply this to more than one tag in the same page - however it does seem to work in IE Explorer …

- id selectors are particularly appropriate for styling layouts with the tag (see later)

CSS – pseudo-class selectors

- Some classes represent objects that can have different states/behaviour
- One example is the HTML anchor tag <a href="http://www..../my.html">
  - a:link {color : red}
  - a:active {color : coral}
  - a:visited {color : green; font-weight : bold}
  - a:hover {color : yellow}
- To use
  - This text will be red/coral/green depending if not visited/active/visited

Using CSS

Hayes D (2002) and Moseley (2006)

- THREE CSS TYPES –

1. linked = style properties are stored in a separate file that is referenced by a <link> tag within the <head> tags. This is a ***true*** "separation of content and presentation";
2. embedded = style properties are included in <style> tags at the top of pages;
3. inline = style properties are included in <other> HTML tags throughout a page.

- WHY "CASCADING"? – Describes the order of precedence – if style for an element is described in more than one place, web browser will cascade precedence (express or over-ride) to the style closest to the <tag> affected. Order of precedence is: inline > embedded > linked.

Cascading Style Sheets (examples)

Hayes D (2002) and Moseley (2006)

INLINE – Simplest but problematic - when updated, every occurrence must be changed. E.g. adding a style attribute to a <table> tag. Useful for “exceptions”.

```
<table style=“background:aqua” width=“100%”>
```

EMBEDDED – Better separation of presentation from content, only have to update at the top of a (each) web page. Useful for testing – CSS and HTML on same page

```
<head><style type="text/css">
  body {font-family: Trebuchet MS, Arial; color: blue; font-size: 12; background-color: Thistle}
  h1 {color: red; background-color: #00ff00} h2 {color: green}
  p {background-color:  rgb(250,0,255)}
</style></head>
```

LINKED – Best separation of presentation from content, only have to update in one file for all pages referenced using the <link> tag. E.g.

```
<head>
  <link rel=“stylesheet” href=“myGreatStyles.css” type=“text/css”>
</head>
```

CSS – more “tricks”

- Many more elements may be manipulated using CSS. Examples include - see Moseley (2006) Chapter 4 or an alternative text for more:
  - background images, colours and properties;
  - text colour, decoration, indentation, case, fonts;
  - borders, boxes, margins, padding;
  - lists – **very important** as form the basis of CSS button-like / image less navigation (see later)
- Later we will also look at using CSS for styling ***layout*** and ***navigation***.