31268 Web Systems

Week 5

The Web and HCI

So where are we at?

Web page development
- *HTML/XHTML*
- *CSS: Cascading Style Sheets*

Web applications
- *Online media and file sharing*
- *Communications*
- *Search engines*

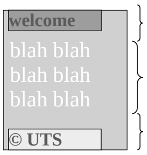
Human Computer Interaction
- *History of the Internet*
- *Security and encryption*

Problem with HTML

- HTML defines both **Structure** and **Presentation** of web pages
  - eg: Structural tags:
    <head> <p> <div> <a>
  - eg: Presentation tags:
    <h3> <center> <em>
- Best to separate structure from presentation.
  → Why?
  - more device types (eg: mobile)
  - different rendering of pages (eg: Printing)
  - accessibility (eg: Text to voice)

Structure and Presentation

- Structure:
  defines the components and areas on the page.
  eg: top of page, middle section, footer



- Presentation:
  defines how the information is presented ie: **style**
  eg: colour, size, background colour, font, etc.

<h1>**Large header**</h1>
<i>*Italic* <b>***Bold***</b></i>

Style Sheets and HTML Presentation

- Style Sheets move the presentation aspects (colour, font, etc.) out of HTML and into separate style sheets.
- Many presentational tags in HTML are deprecated
  →i.e. their usage is discouraged in new HTML and they will eventually be dropped from the HTML standard.
  - eg: <font> <center> <strong> <u>
  - see: http://www.html-reference.com/depreciated.asp for a list

Definition

From W3C:- *Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, spacing) to Web documents.*

It is how we achieve the outcome of separating style from structure.

CSS allows us to define the behaviour of each tag or container for our content.

Who's responsible for them?

• The W3C (www.w3.org) is responsible for defining many web standards and CSS is one of them.

• CSS 1 and CSS 2 have been released and version 3 is being finalised

• Most browsers follow the standards as best they can but there are subtle differences.

What does it look like?

A set of rules defining how elements are to be presented. The rules are set out as follows:

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

```
p { font-size: 12px;
}
```

What is each rule is applied to?

- 4 types of selectors available to us:

1. tag
   - this can be for any HTML tag
   - In stylesheet: p {color: yellow; }
   - In HTML: <p>this paragraph is yellow</p>
2. #id
   - this can be for a particular element of our page.
   - Starts with a # symbol
   - In stylesheet: #myYellow { color: yellow; }
   - In HTML: <p id="myYellow">this para is yellow</p>

What is each rule is applied to?

3. .class
   - allows us to apply a style to a given group of elements
   - In stylesheet: .yellowPara { color: yellow; }
   - In HTML: <p class="yellowPara">this para is yellow</p>

4. inline
   - Effectively hardcoding a style in an element without a stylesheet.
   - In HTML: <p style="color: yellow;">this para is yellow</p>

→you should avoid this technique where possible.

.

We need some more flexibility

- Sometimes we want to divide our page into sections with different styles. Use the following tags:

- <div>
  - apply style to elements that need to be separated from each other. Most browsers put a line break here.

- <span>
  - apply a different style to part of your content inline without any line breaks
  - e.g: this is **<span class="yellow">**inline**</span>** styles.

➔ this is inline styles

How to implement CSS

4 ways in which CSS can be included in your page:

1. Imported – adding the contents of an external CSS file to another set of CSS rules.
   - Stylesheet: @import url(/css/mystyle.css) 🚫
   - HTML: only within in <style> tags

2. Linked – an external CSS file included in the head of the page **** BEST WAY ****
   - Stylesheet: this just contains the styles.
   - HTML: <link rel="stylesheet"
     href="/css/mystyle.css"
     type="text/css"> ♥

Implement CSS (2)

3. Embedded – including the CSS as a section in the head of the page
   - HTML:
     ```
     <style>
       p { color: yellow;}
     </style>
     ```
4. Inline – style included as an attribute of a tag
   - HTML: <p style="color:yellow;">

CSS positioning

- CSS allows us to position HTML anywhere on a screen
  - Can be **relative** to current position
  - or **absolutely** placed – by pixel or percentage
  - You can also specify margins, borders, spacing, padding etc
  - → great for navigation bars without Javascript!
- Don't need to use <tables> or <frames> anymore
- Web 2.0 applications extensively use CSS
  - eg: Google maps →
  - eg: suckerfish menus → http://www.htmldog.com/articles/suckerfish

Cascading, The C in CSS

- Be aware that styles cascade in an order of precedence. This order is:

1. Browser Default settings (lowest!)
2. User settings in Browser
3. Linked External CSS
4. Imported CSS
5. Embedded CSS
6. Inline CSS
7. HTML Tag attributes (highest!)

precedence

Nesting of tags

- When tags are **nested** the styles for several tags may apply to a piece of content.
- You need to calculate the precedence. See http://www.w3.org/TR/CSS2/cascade.html
- Rule of thumb: inline before **id** before **class** & applied to **innermost** tag.
- eg:
  ```
  <div class="greenish">
  <p id="redish">Hello, World</p>
  </div>
  ```

?

⚠ **Microsoft IE treats precedence DIFFERENTLY from the rest!**

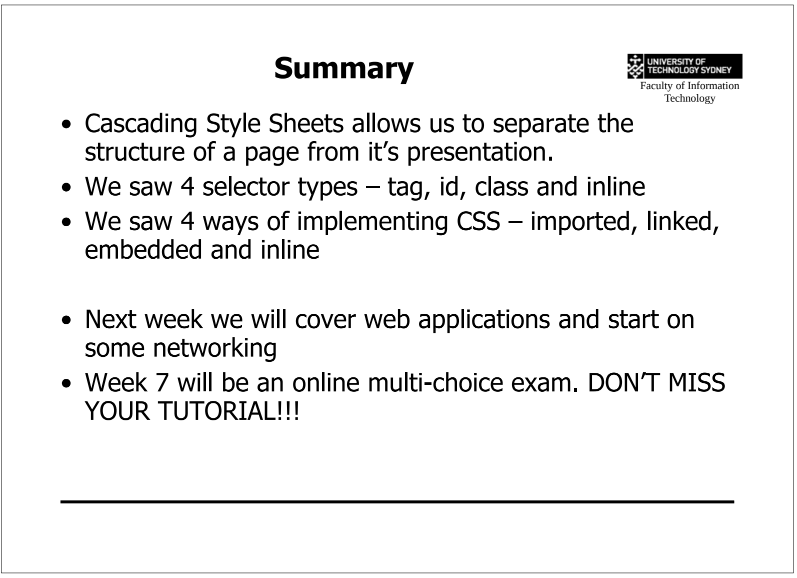
```
p {color: blue };
#redish { color: red};
.greenish { color: green}
```

Why use CSS?

- Some of the benefits of using CSS include:
  - A definite separation of style and structure
  - Simpler, cleaner code
  - Uniformity
  - The ability to define the look of several pages from one location
  - More power and flexibility than old methods
  - Better code reuse (makes your life easier)
- Problems: ⚠
  - Not all browsers follow the standards properly eg: Internet explorer 6
  - "Quirks mode" – where standards based browsers try to emulate broken Internet Explorer

Where to from here?

- We have only just scratched the surface of what CSS can achieve. Other references:
  - http://www.w3.org/Style/CSS - The official website
  - http://www.w3schools/css - good learning site
  - http://www.alistapart.com/topics/code/css/ a great set of articles on how you can push CSS to it's limits
  - http://meyerweb.com/eric/css/edge/ - shows off some of the fancier effects that can be achieved with CSS
  - http://www.glish.com/css/ - an excellent reference on how to achieve various different page layouts using CSS



Summary

- Cascading Style Sheets allows us to separate the structure of a page from it's presentation.
- We saw 4 selector types – tag, id, class and inline
- We saw 4 ways of implementing CSS – imported, linked, embedded and inline

- Next week we will cover web applications and start on some networking
- Week 7 will be an online multi-choice exam. DON'T MISS YOUR TUTORIAL!!!