

CM0133 Internet Computing

2. Introduction to HTML

Objectives

Today we will look at

- Markup Languages and HTML
- Basic structure elements
- Block elements
- Inline descriptive elements
- Inline explicit style elements
- Lists
- Special characters

Markup languages

- Suppose we have a document containing only plain text
- We tag certain parts of the document to indicate **what they are** and **how they should be formatted**
 - This procedure is called **marking-up** the document
 - Tags are usually **paired**: e.g. `<title>My Memoirs</title>`
 - A pair of tags plus their content constitute an **element**
 - Un-paired tags are called **empty tags**

Markup languages

- Physical vs Semantic markup
 - physical refers to [appearance](#) (style) on the page
 - semantic refers to [structure and meaning](#)
- HTML is the [HyperText Markup Language](#)

Markup languages

- HTML is based on SGML (Standard Generalised Markup Language) which is more complex
- HTML places primary emphasis on [structure](#)
 - paragraphs, headings, lists, images, links,
- HTML places secondary emphasis on [style](#)
 - fonts, colours,
- HTML does not label the [meaning](#) of the text (XML)
- HTML has a fixed set of tags but is constantly evolving
 - newer versions are downward compatible

HTML - History

- 1991 - Tim Berners Lee publishes the first version of HTML and runs the first webserver
- 1993 - Mosaic: the first full-featured browser is made available
- 1996 to 2001 - The .com craze is in full swing. Anything can be bought online
- 2005 to present - The rise of rich internet applications
- 2008 - Run internet applications offline (Adobe AIR, Mozilla Prism, ...)

HTML - Versions

- A number of drafts published between 1991 and 1995
- Version 2.0 (1995) - Based on features developed in the Mosaic browser
- Version 3.2 (Jan 1997) - Extended based on the Netscape browser's visual markup elements
- Version 4.0 (Dec 1997) - Clean-up deprecating most of the visual markup elements
 - Comes in three flavours: Strict, Transitional, Frameset

XHTML

- XHTML is derived from HTML, but describes a valid XML document
 - All tags and attributes are in lower-case
 - All tags must be closed
 - All tags must be correctly nested
- XHTML 1.0 was derived from HTML 4.01
- XHTML 1.1 is a modularisation of XHTML 1.0

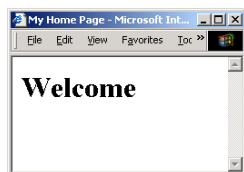
A basic document

- Every document should start with the following line

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

- There are three required elements, defined by the tags `<html>`, `<head>` and `<body>`

```
<html>
<head>
  <title>My Home Page</title>
</head>
<body>
  <h1>Welcome</h1>
</body>
</html>
```



Basic structure elements

- `<html>` and `</html>` must be the first and last tags
- The HEAD section
 - must come before the BODY section
 - contains **generic information** about the document
- Elements specified in the HEAD section include
 - **title, base, link, meta, script, style**
- The BODY section
 - contains the **content** of the document (text, images etc)
 - this content is structured by other tags

Block elements

- Block elements define sections of text, usually preceded by a blank line
- `<p></p>` - **paragraph**
- `<h1></h1>...<h6></h6>` - **headings**
- `<pre></pre>` - **preserve** (original format)
- `<blockquote></blockquote>` - **indented text**
- `<div></div>` - **division**
 - used to identify a section of the document that may be subject to special formatting (for example, using stylesheets).

Paragraphs

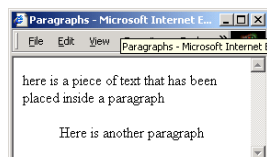
Paragraphs: `<p>...</p>`

- force a break between the enclosed text and the text surrounding it
- the tagged region of text may be subject to special formatting

`<p align="center">Here is another paragraph</p>`

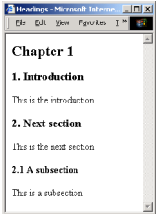
- **align** is an **attribute** of the paragraph tag
- **center** is the **value** of the align attribute

```
<p>here is a piece of
text that has been
placed inside a
paragraph</p>
<p align="center">Here
is another
paragraph</p>
```



Headings

- Use headings to divide document into sections
- Six levels of importance
<h1>...<h6>



```
<html>
<head>
  <title>Headings</title>
</head>
<body>
  <h2>Chapter 1</h2>
  <h3>1. Introduction</h3>
  This is the introduction
  <h3>2. Next section</h3>
  This is the next section
  <h4>2.1 A subsection</h4>
  This is a subsection
</body>
</html>
```

Element relationships

- The elements marked by tags form a **hierarchy**
- The root element is **html** (marked by <html>...</html>)
- It usually has two children: **head** and **body**
 - each of these are further subdivided
- There are **rules** for which elements can contain other elements
 - e.g. headers cannot contain headers
 - see <http://www.w3.org/> for a full list of rules
- Elements must not overlap each other
 - we cannot have: <h1>...<a...> ... </h1>...
 - we can have: <h1>...<a...></h1>

Links

- The **link** (anchor) element <a>... provides hypertext links between
 1. different documents (using a URL)
 2. different parts of an individual document
- User selection of the link (hot spot) results in
 1. retrieval and display of the designated document
 2. movement to relevant part of same document

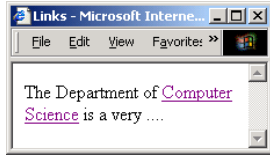
```
<body>
The Department of
<a href="http://www.cs.cf.ac.uk/index.html">
Computer Science</a> is a very ...
</body>
```

Link with URL

Example:

```
<body>
The Department of
<a href="http://www.cs.cf.ac.uk/index.html">
Computer Science</a> is a very ....
</body>
```

- The **href** attribute gives the URL of the target page
- The text between the tags is highlighted – selecting it activates the link



Relative addressing

- The previous example gave the full path name, known as the **absolute address**
- This is OK for URLs that are external to a website, but can be inflexible for web pages that are part of the website, because it ties all web pages (files) to a particular location (directory)
- A **relative address** specifies a URL relative to the directory of the page in which they are called (the parent page)
- This allows the location of a collection of web pages (that constitute a website) to be changed without having to edit all of the internal links

Relative addressing

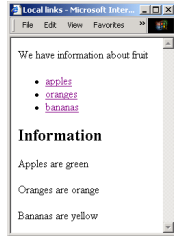
- The home page for my website is
`http://www.cs.cf.ac.uk/user/F.A.Twaroch/index.html`
- This page has several links to other pages:

```
<a href="research.html">Research</a>
<a href="pub.html">Publications</a>
<a href="McGurk/index.html">Stuff..</a>
<a href="../../index.html">Computer Science home</a>
```

- The 'root' directory for the link is assumed to be the directory containing the parent page of the link
- Clicking on the 'Research' link results in accessing
`http://www.cs.cf.ac.uk/user/F.A.Twaroch/research.html`

Local links

```
We have information about fruit
<ul>
  <li><a href="#apples">apples</a></li>
  <li><a href="#oranges">oranges</a></li>
  <li><a href="#bananas">bananas</a></li>
</ul>
<h2>Information</h2>
<p><a name="apples">
  Apples are green
</p>
<p><a name="oranges">
  Oranges are orange
</p>
<p><a name="bananas">
  Bananas are yellow
</p>
```



Images

- Images are included using the empty tag ``
- Example:
``
- The `src` attribute specifies the file containing the image
 - absolute or relative path names can be used (see notes for links)
- The `alt` attribute specifies the text to be displayed if the image is not viewed
 - some users choose not to display images (for faster download)
 - also used for compatibility with older browsers

Image attributes

- The `size` attributes control the size of the image
``
- The `align` attribute controls the vertical location of the image, relative to the line of text
 - `align="top"` top of image aligned with top of text
 - `align="middle"` centre of image aligned with centre of text
 - `align="bottom"` bottom of image aligned with baseline of text

Image attributes

- The **align** attribute also controls the horizontal location of the image, relative to the line of text
 - **align="left"** image aligned with left margin
 - **align="right"** image aligned with right margin
- The paragraph text flows around left or right aligned images - a feature of HTML 4
- To stop wrap around, use the clear attribute of the **break element** `
` (an empty tag)

```
  
<br clear="left">
```

Image file formats

- GIF – Graphics Interchange Format (.gif)
 - 256 colours adapted to image
 - compressed (not good for photos)
 - options for transparency and animation (GIF89A)
- JPEG – Joint Photographic Experts Group (.jpg or .jpeg)
 - sophisticated compression
 - image quality can be chosen (good for photos)
- PNG – Portable Network Graphics (.png)
 - non-proprietary GIF (with better colour quality)
- X-Bitmap (.xbm)
 - black and white (transparent)
- X-Pixmap (.xpm)
 - 8 bits per pixel (colour)

Links with images

- A link element can include an image instead of text
 - both images and text can be included if required

```
<body>  
Enter my world of cats <a href="cats.html"></a>  
</body>
```



Large images via Thumbnails

- Large and high resolution images take a noticeable amount of time to load into the page.
- If you have an image that the user may or may not be interested in seeing, then provide a thumbnail version (which takes up very little disk space) as a link.
- When the user clicks on the thumbnail, the large image is loaded by the browser.
- This is an **external** file, as opposed to an **inline** file that is automatically loaded, e.g.:

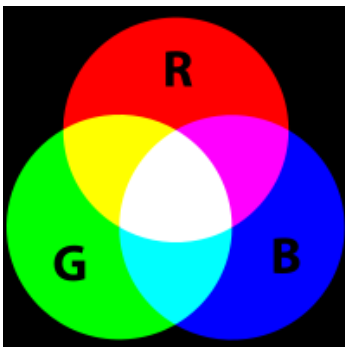
```
<a href="external.jpg" target="_blank">  
</a>
```

Colour

- We can specify the colour of text, the background of the whole page and the background of various parts of the page (e.g. the cells of a table – see later).
- Colours are specified with hexadecimal numbers for the red, green and blue primary colours, preceded by a "#".
- Each colour component has a value between 00 and ff (0 - 255 decimal)
- To set the background colour of a web page

```
<body bgColor="#994422">
```

Colour – RGB Model



- #ff0000 (red),
- #00ff00 (green)
- #0000ff (blue)
- #ffff00 (yellow)
- ...
- #3395ab (a pastel blue)

Colour

- To set the colour of all text on a page

```
<body text="#994422">
```

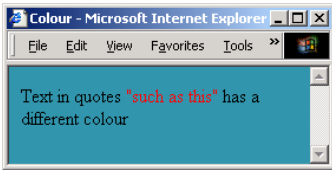
- In the body element, the colour of link text can be controlled with the following attributes:
 - `link` for an un-visited link
 - `vlink` for a visited link
 - `alink` for a link that is currently selected by the mouse
- Example

```
<body text="#000000" link="#0000ff">
```

Colour

- To set the colour of an individual piece of text use the `font` element (or preferably stylesheets – see later)

```
<body bgcolor="#3395ab">  
Text in quotes <font color="#ff0000">"such as  
this"</font> has a different colour  
</body>
```



Colour names

- Netscape and Internet Explorer allow textual names for colours instead of hexadecimal
- This is OK provided the page is not looked at by a browser that does not support colour names
- Some HTML text books give the available colour names and their hexadecimal equivalents
- For example

```
<body bgcolor="gray" text="black" link="blue">
```

Background patterns

- To give the background of your web page a pattern (rather than a uniform colour), use the background attribute of the body element to provide the name of a file containing an image that can be tiled
- HOWEVER, be careful in your choice of background pattern – some are very disconcerting and can make reading the page an unpleasant experience

```
<body background="tileimage.gif">
```

Question

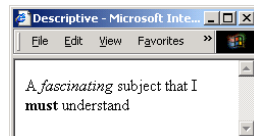
- **Warning:** Not all browsers support the latest features of HTML. Even when they do there are differences in the resulting appearance.
- What does this mean to you as a developer?

Inline descriptive elements

Descriptive elements affect the appearance of text depending on how the text is described

- `` **emphasis**, usually with italics
- `` **strong**, usually with bold
- `<cite></cite>` **citation**, usually in italics
- `<code></code>` usually results in monotype spacing

```
<body>
A <em>fascinating</em>
subject that I
<strong>must</strong>
understand
</body>
```



Inline explicit style elements

- `<boldface></boldface>`
- `<big></big>` bigger font than surrounding text
- `<small></small>` smaller font than surrounding text
- `<i></i>` italics
- `<s></s>` strikethrough
- `` subscripts
- `` superscripts
- `` delimits text for stylesheet control
- `<div></div>` delimits blocks of text for stylesheet control

Inline explicit style elements

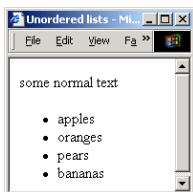
`` attributes

- **face** - name of font (must be installed)
 - "arial", "times", "verdana", "helvetica"
- **size** - absolute size (1-7), or relative to previous text
 - "2", "5", "7", "+1", "-2"...
- **color** - hexadecimal RGB, or a named color
 - "3399dd", "blue", "red"
- **weight** - boldness from 100, 200, ..., 900
 - "100", "300", "900"
- e.g.

```
<font face="arial" size="+1" color="pink" weight="300">
```

Unordered lists

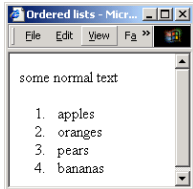
- Unordered lists `...`
- `...` for the list elements
- each item has a **bullet**



```
some normal text
<ul>
<li>apples</li>
<li>oranges</li>
<li>pears</li>
<li>bananas</li>
</ul>
```

Ordered lists

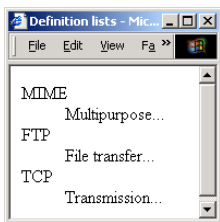
- Ordered lists `...`
- `...` for the list elements
- each item has a **number**



```
some normal text
<ol>
<li>apples</li>
<li>oranges</li>
<li>pears</li>
<li>bananas</li>
</ol>
```

Definition (glossary) lists

- `<dl></dl>` The enclosing tags
- `<dt></dt>` The definition term
- `<dd></dd>` The definition



```
<dl>
<dt>MIME</dt>
<dd>
  Multipurpose...
</dd>
<dt>FTP</dt>
<dd>
  File transfer...
</dd>
<dt>TCP</dt>
<dd>
  Transmission...
</dd>
</dl>
```

Nested lists

- A list may contain another list
- The inner list is **nested** inside the outer list



```
<body>
<ol>
<li>apples</li>
  <ul>
    <li>red</li>
    <li>green</li>
  </ul>
<li>oranges</li>
<li>pears</li>
<li>bananas</li>
</ol>
</body>
```

Comments

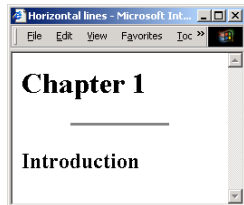
- Comments are delimited by `<!--` and `-->`
`<!-- this is a comment -->`

- Comments may span multiple lines

```
<body>
<!--
  this is
  a comment
-->
</body>
```

Horizontal lines

- To insert a horizontal line to divide up parts of a document we use the empty tag `<hr>`
- Attributes: `align`, `size` (in pixels), `width` (in pixels or percentage), `noshade`

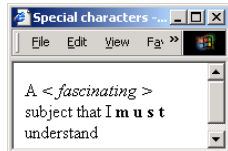


```
<body>
<h1>Chapter 1</h1>
<hr align="center" size="3" width="50%" noshade>
<h2>Introduction</h2>
</body>
```

Special characters

- Some characters such as `<`, `>`, `"` and `&` have special meanings.
- To prevent them being interpreted as HTML code, they must be written as follows: `<`; `>`; `"`; `&`
- Blank space is normally ignored in HTML. To include a space in your document use: ` `

```
<body>
A <em> &lt;
fascinating &gt; </em>
subject that I
<strong>m&nbsp;u&nbsp;
s&nbsp;t</strong>
understand
</body>
```



Structure & Styling

- Separation of Structure & Styling
 - Easier to update documents
 - Easier to change the styling
 - Styles are attached to elements, not an integral part of the document
 - Allows for improved machine-readability
 - Better indexing & searching of documents
 - Faster parsing & smaller parser size

Summary

Today we looked at

- HTML a Markup Language
- Basic structure elements
- Block elements
- Inline descriptive elements
- Inline explicit style elements
- Lists
- Special characters

Outlook

Next time we will look at

- More HTML
- Images
- Links
- Tables, Frames
- Use of color
- Web page design
