COMP4021

Internet Computing

# HTML

**David Rossiter**

# HTML

- **HTML =Hypertext Mark-up Language.**
- HTML is a markup language (or tagging language)
- A tag indicates what an object is but not how it is displayed
- Separation of content and display style

```
<h1>Internet Computer</h1>
<p>The objectives of COMP 4021 are:
<ul>
<li> … </li><li> … </li>
</ul></p>
```

# Historical Overview

- First there was HTML
- Then there was HTML + CSS (=style sheets)
- Then there was XML (Mother of HTML)
- Then there was XHTML(=HTML, using XML style) and DHTML (Dynamic HTML: Dynamic web page; HTML + JavaScript)
- Now: HTML 5
  - Many parts of HTML 5 are supported by modern browsers
  - HTML 5 is a full W3C standard
- Now, W3C uses the standard name "HTML" to refer to the language (no "5" or "x" attached)

# Basic HTML Page Structure

```
<html>

<head>
<title>Title of the web page goes here</title>
</head>

<body>
...HTML tags go here...
</body>

</html>
```

- You may find some older pages using <xhtml>..</xhtml> instead of <html>..</html> but these days we stick to <html>..</html>
- With 'old style' HTML, some tags/sections could be omitted and the browser would still display something appropriate – this is considered to be bad
- These days you need to do everything 'properly'
- For example, if you start a tag, you must finish it, and so on

# HTML Tags

- HTML tags need matching start and end tags
- Basic idea:

```
<html_tag>text</html_tag>
```

- Tags are case sensitive; must have matching case; (usually lower case), so can't do this:
  `<p>paragraph text</P>`
- Some tags don't have an end tag, i.e.
  `<br />`
  `<hr />` ⇐ `/>` means 'end of the command'

# Headers

- Headers help indicate document structure
- Default web page display shown here

<h1>Header 1</h1>

<h2>Header 2</h2>

<h3>Header 3</h3>

<h4>Header 4</h4>

<h5>Header 5</h5>

<h6>Header 6</h6>

# HTML Text

- Examples of simple HTML text formatting:

This is bold
This is italic
This is <u>underline</u>
This is <tt>text</tt> (fixed width font)
This is <big>big</big>
This is <small>small</small>
This is <strike>strikethrough</strike>
This is <sub>Subscript</sub>

This is <sup>Superscript</sup>

# Special Characters

- Because < and > are used by HTML for denoting tags, special methods are required to visually generate these two characters, as well as other special characters

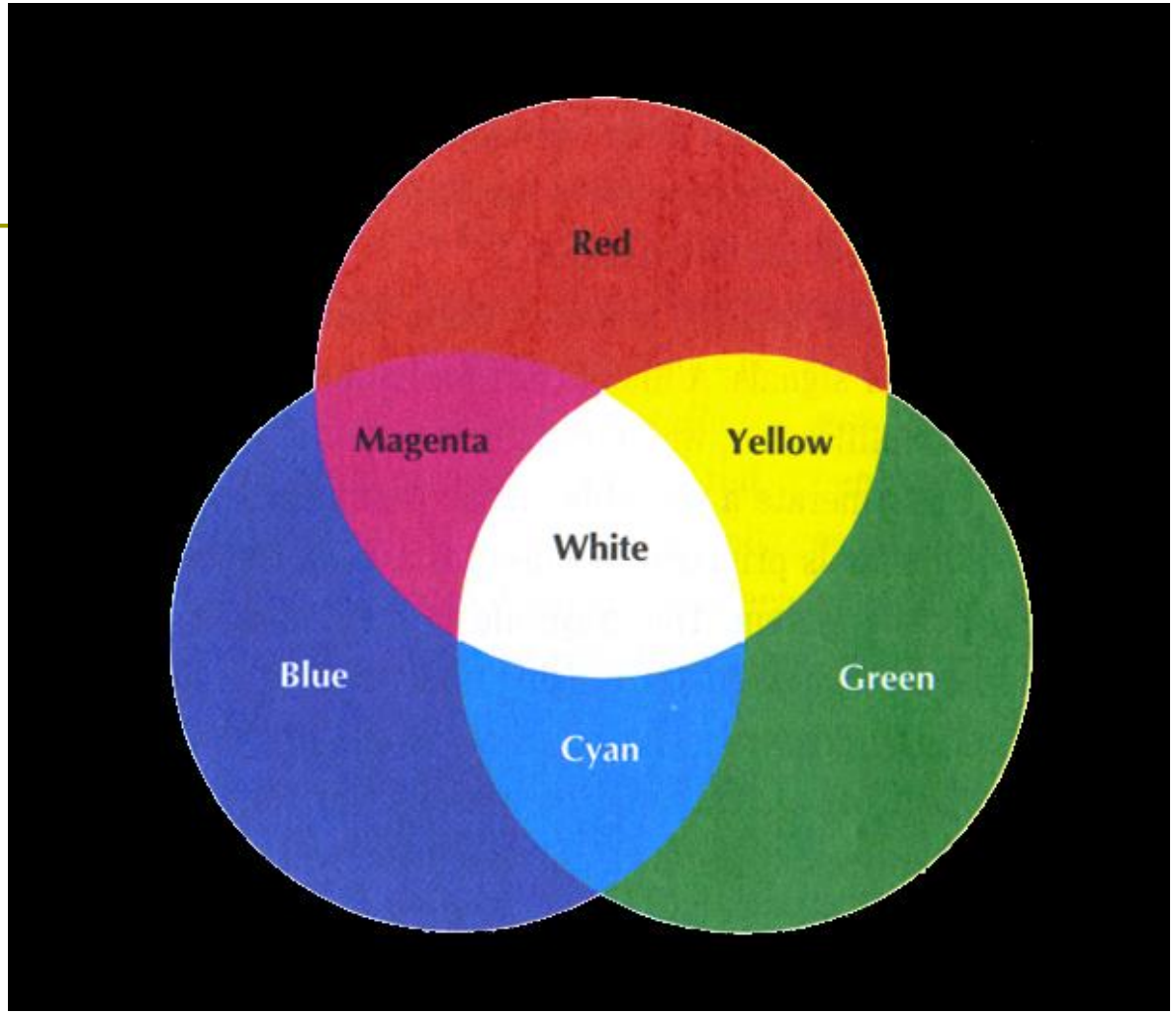
| Character | HTML Code |
|---|---|
| non-breaking space (meaning it won't get shrunk) |   |
| & | &amp; |
| < | &lt; |
| > | &gt; |
| " | &quot; |
| ® | &reg; |
| © | &copy; |
| ¼ | &frac14; |
| ½ | &frac12; |
| ± | &plusmn; |
| ÷ | &divide; |

# Using Colours on the Web

- For handling colours on the web usually RGB or English words are used (this applies to all web display technologies, such as HTML, Flash, applets, and so on)
- For the RGB system a combination of some red plus some green plus some blue is used to create any colour
- For example: `#FF8800` means maximum red, half green and no blue, to create one single colour, written in hexadecimal
- Examples:

```
<p style="color:green">green text!</p>
<p style="color:#00ff00">green text!</p>
```

# RGB Colour




- ***RGB = Red Green Blue***
  - **000000 = black, FFFFFF = white**

# Example Colours using RGB

| *Name* | *Example* | *Hex code* | *Name* | *Example* | *Hex code* |
|---|---|---|---|---|---|
| black | | #000000 | silver | | #C0C0C0 |
| grey | | #808080 | white | | #FFFFFF |
| maroon | | #800000 | red | | #FF0000 |
| purple | | #800080 | fuchsia | | #FF00FF |
| green | | #008000 | lime | | #00FF00 |
| olive | | #808000 | yellow | | #FFFF00 |
| navy | | #000080 | blue | | #0000FF |
| teal | | #008080 | aqua | | #00FFFF |

# Expressing Dimension in HTML

- For the relevant HTML tags which have some kind of length, the magnitude can usually be expressed in two ways
  - as a percentage of the length of the parent 'thing'
  - as an exact pixel length
- For example, the horizontal rule tag <hr /> shows a line across the sc
- Examples:

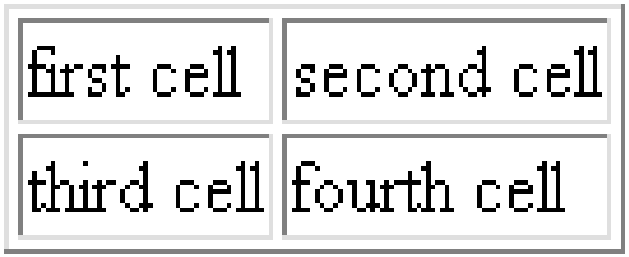
```
<hr align="left" width="200">
<hr align="left" width="70%">
```

# Tables

**A simple table:**

```
<table border=1
   columns=2>
  <tr>
  <td>first cell</td>
  <td>second cell</td>
  </tr>
  <tr>
  <td>third cell</td>
  <td>fourth cell</td>
  </tr>
</table>
```

- Tables are the most common way to get a basic visual structure on a web page
- Using HTML lists and layers are two other ways



# HTML Lists

- Definition list

```
<dl>
<dt>definition
   term</dt>
<dd>definition
   data</dd>
</dl>
```

definition term
    definition data

- Unnumbered list

```
<ul>
<li>item 1</li>
<li>item 2</li>
</ul>
```

- item 1
- item 2

- Ordered list

```
<ol>
<li>item 1</li>
<li>item 2</li>
</ol>
```

1. item 1
2. item 2

# Absolute and Relative Links

- **Example absolute link:**

```
click <a href="http://www.sitename.com/">here</a>
```

- **Example relative link:**
  - Links can point to a position within the *same* page
  - Define a target reference (i.e., position) in the page:

```
<h2 id="part5">Part 5 goes here</h2>
```

  - Clicking the following link will go to “part5” object:

```
go to <a href="#part5">part 5</a>
```

- **Absolute + relative link:**
  - Access to a object within another page:

http://www.sitename.com/pagename.html#part5

# Take Home Message

- HTML is one of the GREAT inventions for information exchange and one of the cornerstones of the Web
- The power of tagging is amazingly powerful
  - Microsoft Office supports HTML as native file format
  - Clear and consistent syntax for defining “objects”
    - Object name/class: <h1 …>, etc.
    - Object attributes: <h1 background=….>
  - Objects can be nested freely

# Take Home Message

- HTML is too big to remember; look up the web, e.g., search “HTML table” on Google or Bing
  - HTML has about 130 tags and 190 unique attributes (as of Feb 2015)
- Try to create a web page yourself:
  - Use a plain text editor to create an HTML page
  - Use an HTML Rich Text Editor (search on Google/Bing)
  - Use WORD, “save as HTML”
- You are expected to remember the tags discussed in the lectures (html, body, list, table, anchor, etc.)
- Most browsers are very tolerant to HTML mistakes (they just ignore things they do not understand), but try to conform to standard (e.g., adding end tags)