CM1102: Introduction to HTML

Martin Caminada

Cardiff University

HTML and CSS overview

- *HTML: Hypertext Markup Language*
  specifies the structure of the document
  (e.g. where do the paragraphs start and end, and what are the section titles)

- *CSS: Cascading Style Sheets*
  specifies how to display the HTML document
  (e.g. use 11pt font size for paragraphs and 18pt font size for section titles)

HTML and CSS are evolving.
In the current module we will use
HTML5 and CSS3 unless stated otherwise

An example HTML5 document

```
<!DOCTYPE html>

<html>

<head>
  <meta charset="utf-8">
  <title>An Example Page</title>
</head>

<body>
  <p>Hello world!<br/>How are you?</p>
</body>

</html>
```

<head> versus <body>

- <head> element:
  contains meta data about the document
  such as the character set, the title
  (to be displayed in the browser tab)
  the author, etc.

- <body> element:
  contains the actual contents of the document
  such as text, hyperlinks, images, tables, etc.

we now proceed with the elements
that can be used within the <body> element

Block Elements versus Inline Elements

- *Block Elements* start on a new line,
  and are preceded and followed by a blank line

  examples:

<p>	paragraph
<h1>	level 1 heading
<h6>	level 6 heading
<pre>	preserve white space and line breaks
<blockquote>	quotation (usually also indented)
<table>	table
<form>	form to be filled in by user
<div>	generic block element
<header>, <footer>, <nav>, <aside>, <section>, <article>	these display the same as <div> but are useful for describing the role of the text in the document (HTML5 semantic elements)

Block Elements versus Inline Elements

- *Inline Elements* continue on the same line

  examples:

<em>	emphasis (usually displayed in italics)
<strong>	strong emphasis (usually displayed in bold)
<cite>	citation (source of information, usually italics)
<code>	computer code (usually monotype spacing)
<q>	inline quote
<span>	generic inline element

more examples (inline explicit style elements):

<b>	boldface
<i>	italics
<small>	smaller than adjacent text (for “fine print”)
<sub>	subscript (like H_2O)
<sup>	superscript (like $E=MC^2$)

Empty Elements

- Empty elements do not have a separate closing tag

  examples:

<br/>	line break
<hr/>	horizontal line
<img/>	image

  please be aware that

  - In HTML4.01 empty elements *do not* end with a ‘/’ (so <br>, <hr>, <img>)
  - In XHTML empty elements *have to* end with a ‘/’ (so <br/>, <hr/>, <img/>)
  - In HTML5 empty elements can *optionally* end with a ‘/’

HTML Comments

- Comments allow you to specify text that will not be displayed by the browser.

  examples:

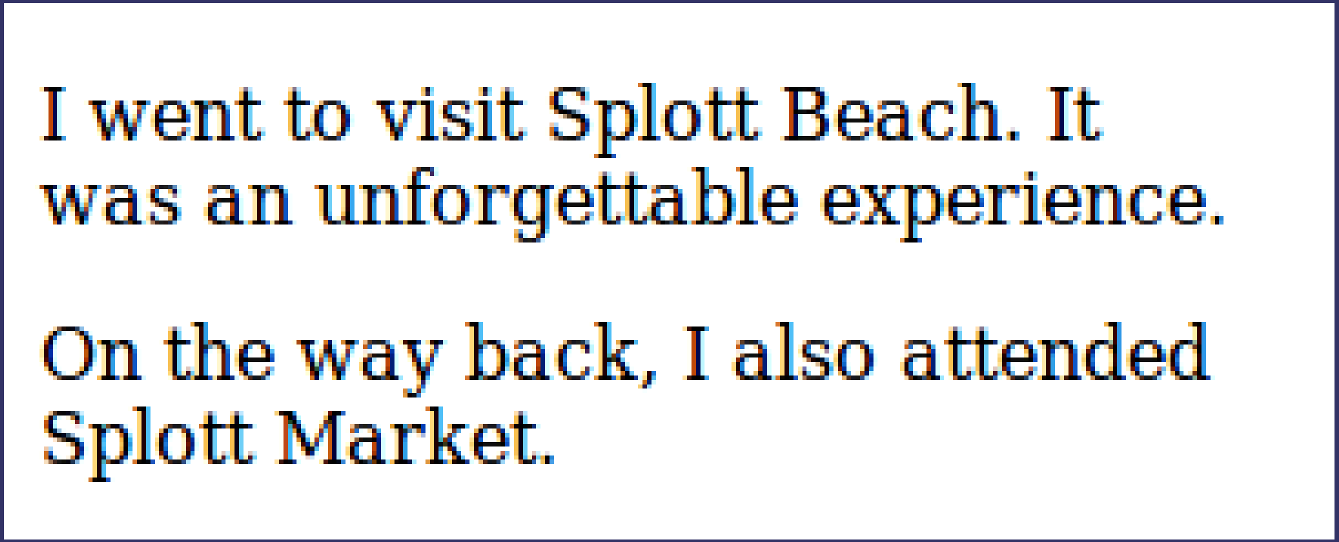
```
<!-- this is a HTML comment -->

<!-- this is a
multiple lines
HTML comment -->
```

Paragraphs

```
<p>I went to visit Splott Beach.
It was an unforgettable
experience.</p>

<p>On the way back,
I also attended Splott Market.</p>
```



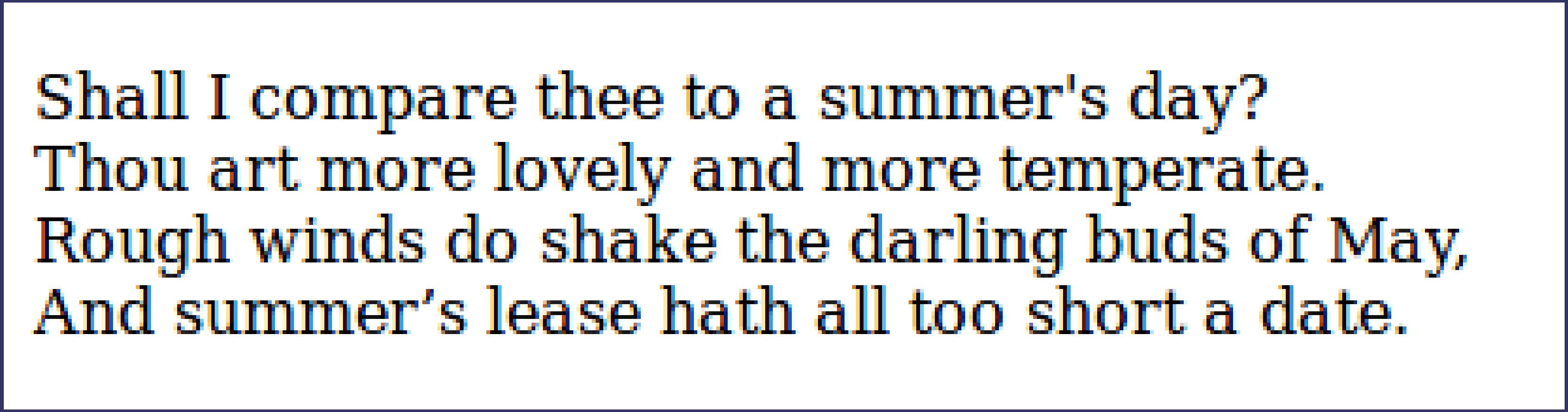
I went to visit Splott Beach. It was an unforgettable experience.

On the way back, I also attended Splott Market.

(please be aware of "whitespace collapse"
it doesn't matter how much white space or line breaks you insert in the HTML source)

Line Breaks

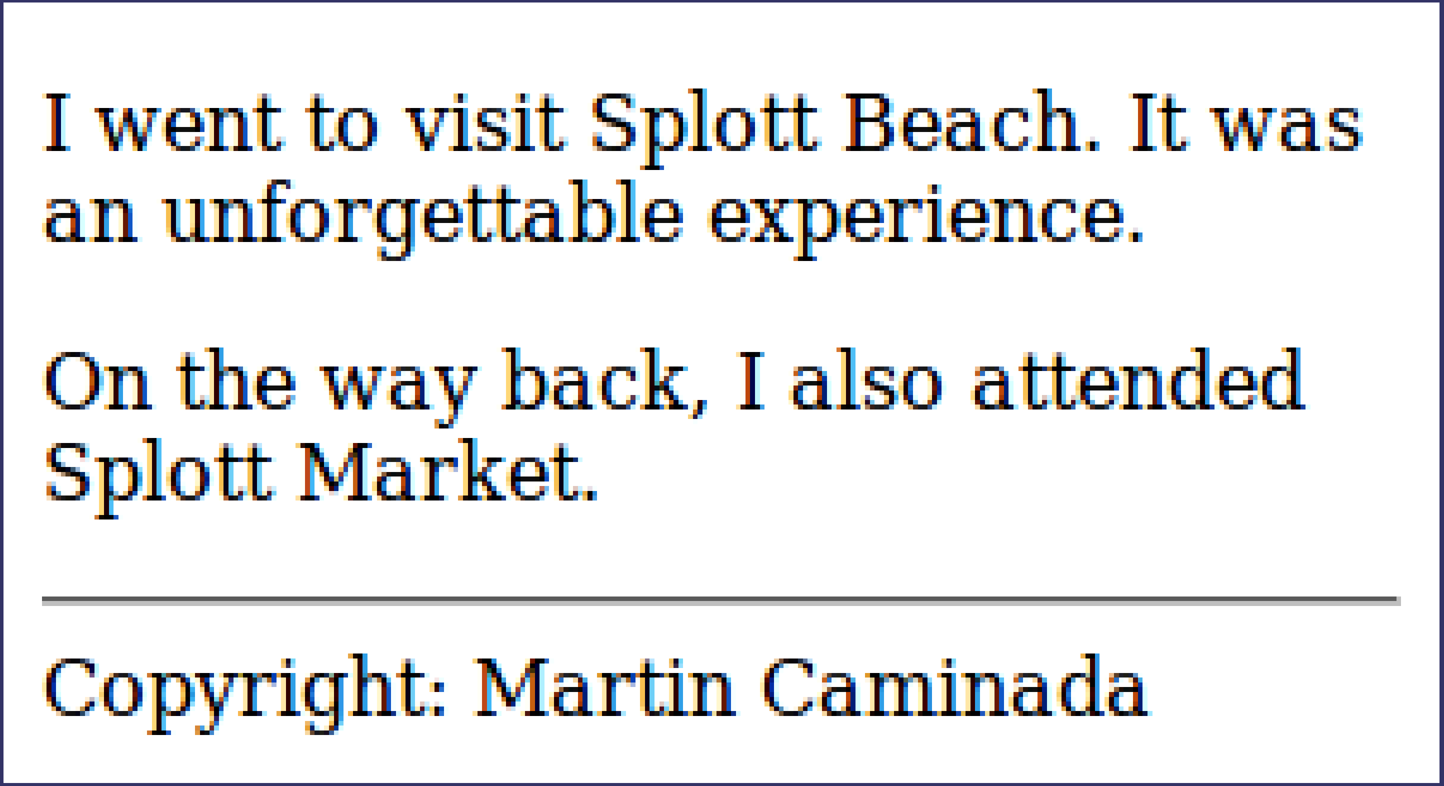
```
<p>Shall I compare thee to a summer's day? <br/>
Thou art more lovely and more temperate. <br/>
Rough winds do shake the darling buds of May, <br/>
And summer's lease hath all too short a date. </p>
```



Shall I compare thee to a summer's day?
Thou art more lovely and more temperate.
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date.

Horizontal Lines (1/2)

```
<p> I went to visit Splott Beach.
It was an unforgettable experience. </p>
<p> On the way back, I also attended
Splott Market. </p>
<hr/>
<footer> Copyright: Martin Caminada </footer>
```



I went to visit Splott Beach. It was an unforgettable experience.

On the way back, I also attended Splott Market.

Copyright: Martin Caminada

Horizontal Lines (2/2)

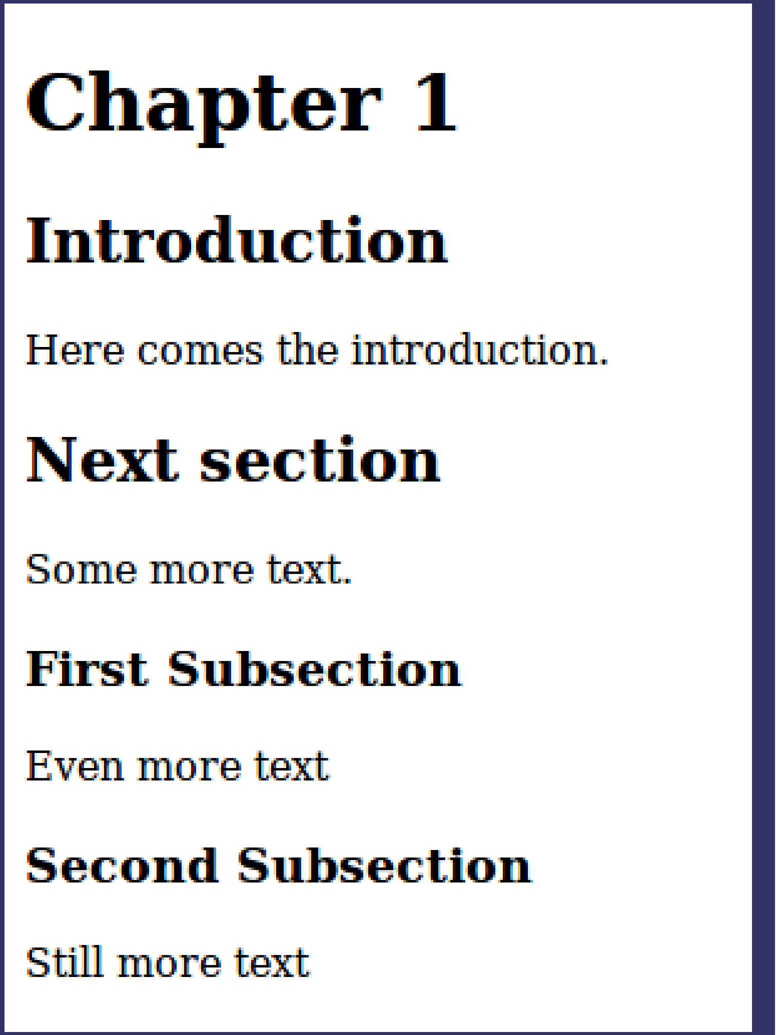
WARNING:

- In HTML 4.01 <hr> represents a horizontal rule
- In HTML5 <hr/> represents a thematic break (so it is defined in semantic terms instead of in terms of explicit layout)
- In HTML5 <hr/> is still displayed as a horizontal rule by most browsers, even though this is not required by the HTML5 standard
- If you want to play safe, use CSS (introduced next week) to explicitly define what <hr/> should display

Headings (1/2)

- Use headings to give titles to the different (sub)sections of the document
- Six levels of headings available: <h1> … <h6>

```
<h1> Chapter 1 </h1>
<h2> Introduction </h2>
   <p> Here comes the
   introduction. </p>
<h2> Next section </h2>
   <p> Some more text. </p>
<h3> First Subsection </h3>
   <p> Even more text </p>
<h3> Second Subsection </h3>
   <p> Still more text </p>
```



Chapter 1

Introduction

Here comes the introduction.

Next section

Some more text.

First Subsection

Even more text

Second Subsection

Still more text

Headings (2/2)

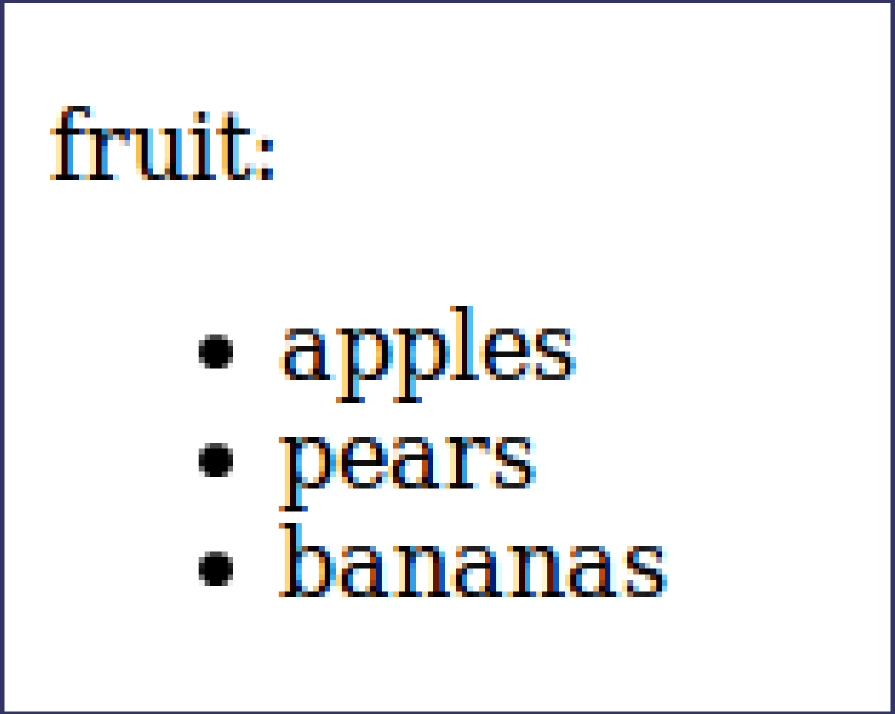
Headings are important:

- Search engines use the headings to index the structure and content of your web pages
- Users skim your pages by its headings. It is important to use headings to show the document structure (especially for large documents)
- Use <h1> for the main heading (like chapter), <h2> for the next level heading (like section), <h3> for the level after that (like subsection) etc
- Do *not* use headings if you just want to display something in big or bold. Headings are for indicating the *structure* of the document.

Unordered Lists

<ul> ... </ul>	unordered list element
<li> ... </li>	list item (use inside of <ul> ... </ul>) each item is preceded by a bullet (unless specified otherwise in CSS)

```
<p>fruit:</p>
<ul>
  <li> apples  </li>
  <li> pears   </li>
  <li> bananas </li>
</ul>
```



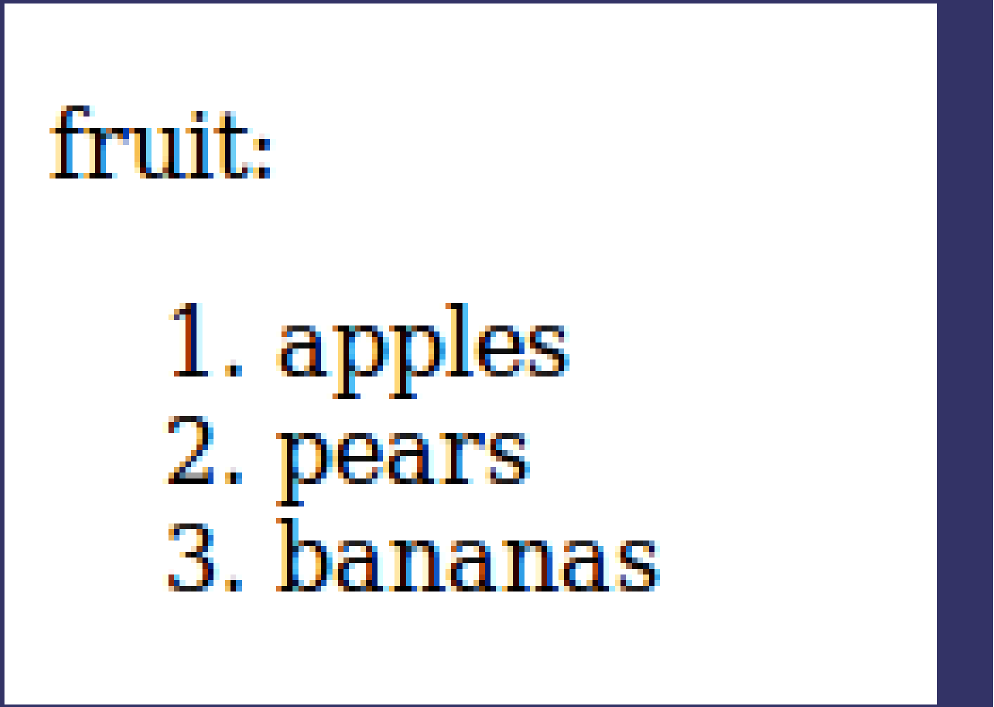
fruit:

- apples
- pears
- bananas

Ordered Lists

<ol> … </ol>	ordered list element
<li> … </li>	list item (use inside of <ol> … </ol>) each item is preceded by a number (unless specified otherwise in CSS)

```
<p>fruit:</p>
<ol>
  <li> apples  </li>
  <li> pears   </li>
  <li> bananas </li>
</ol>
```



fruit:

1. apples
2. pears
3. bananas

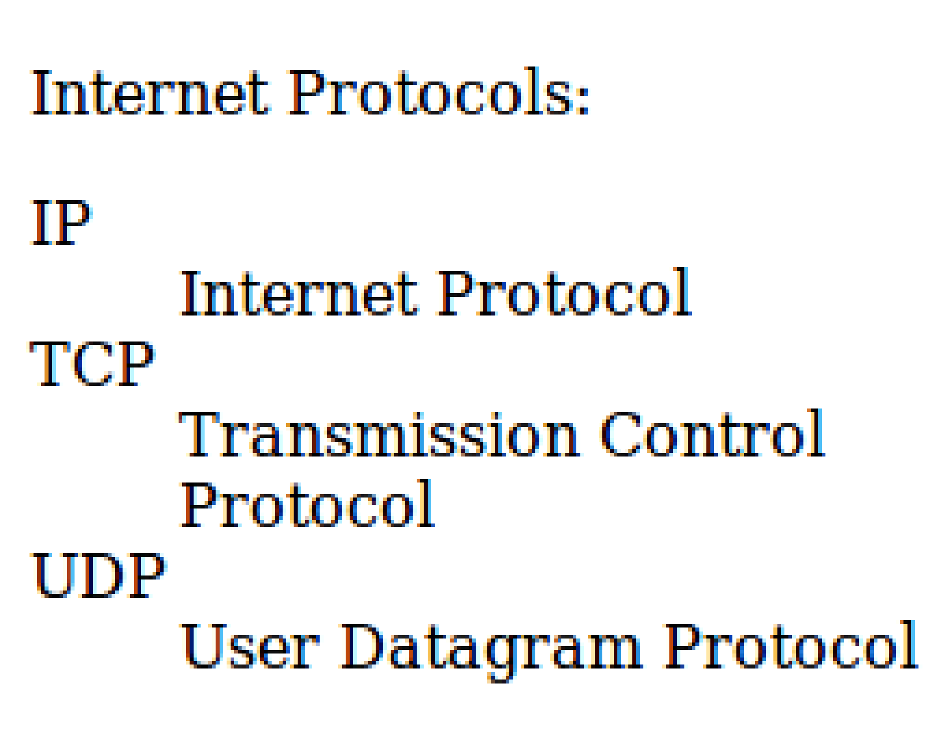
Description Lists

<dl> … </dl> description list element

<dt> … </dt> definition term
(this is what you are describing)

<dd> … </dd> definition data
(this is the description of the term)

```
<p>Internet Protocols:</p>
<dl>
  <dt> IP </dt>
      <dd> Internet
         Protocol </dd>
  <dt> TCP </dt>
      <dd> Transmission Control
         Protocol </dd>
  <dt> UDP </dt>
      <dd> User Datagram
         Protocol </dd>
</dl>
```



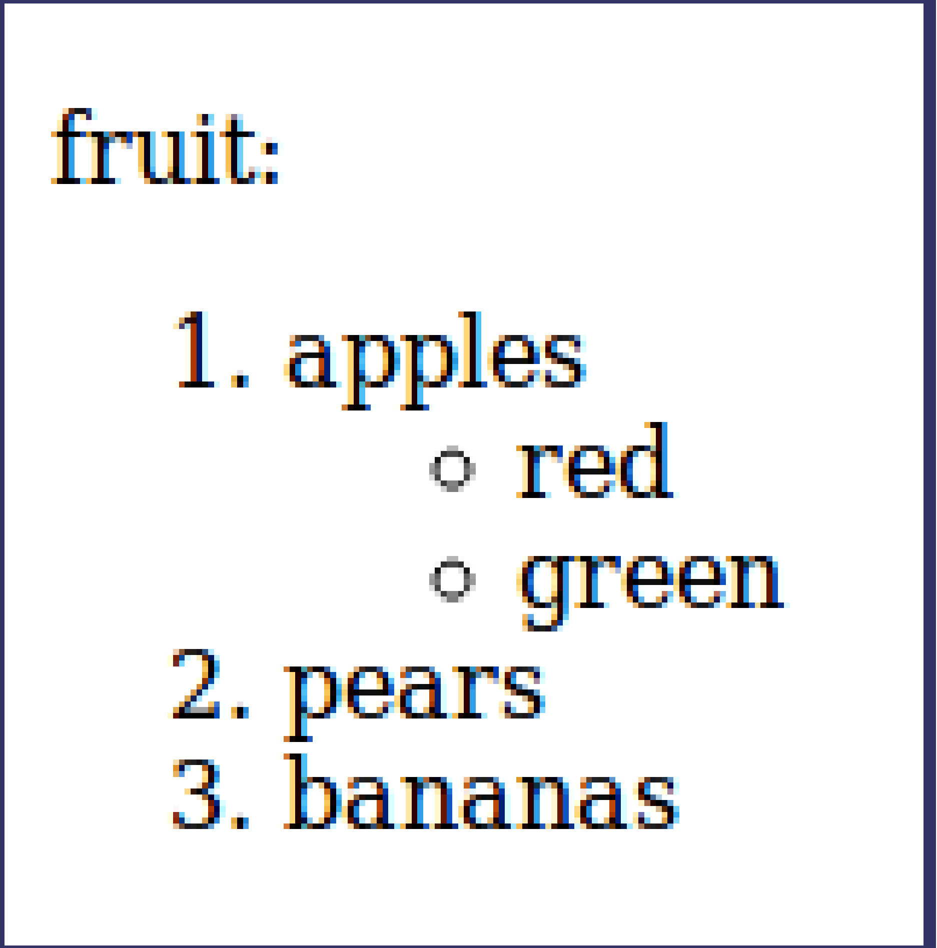
Internet Protocols:

IP
    Internet Protocol
TCP
    Transmission Control
    Protocol
UDP
    User Datagram Protocol

Nested Lists

A list may contain another list
(this has to be done inside of a list item)

```
<p> fruit: </p>
<ol>
  <li> apples
   <ul>
       <li> red   </li>
       <li> green </li>
   </ul> </li>
  <li> pears </li>
  <li> bananas </li>
</ol>
```



fruit:

1. apples
   - ○ red
   - ○ green
2. pears
3. bananas

Images (1/3)

- The <img/> element includes an image into a page (this is an empty element, so no closing tag)

```
<img src="uni_logo.png" alt="Cardiff Uni Logo"/>
```

Required attributes:

- *src* attribute
  specifies the file containing the image
- *alt* attribute
  specifies the text to be displayed
  if the image is not displayed

Images (2/3)

- Use the height and width attributes to specify the size of the image in pixels

```
<img src="uni_logo.png" alt="Cardiff Uni Logo"
height="100" width="100" />
```

Advantage: browser already knows how much space to allocate to the image (so no need to resize after the image is loaded

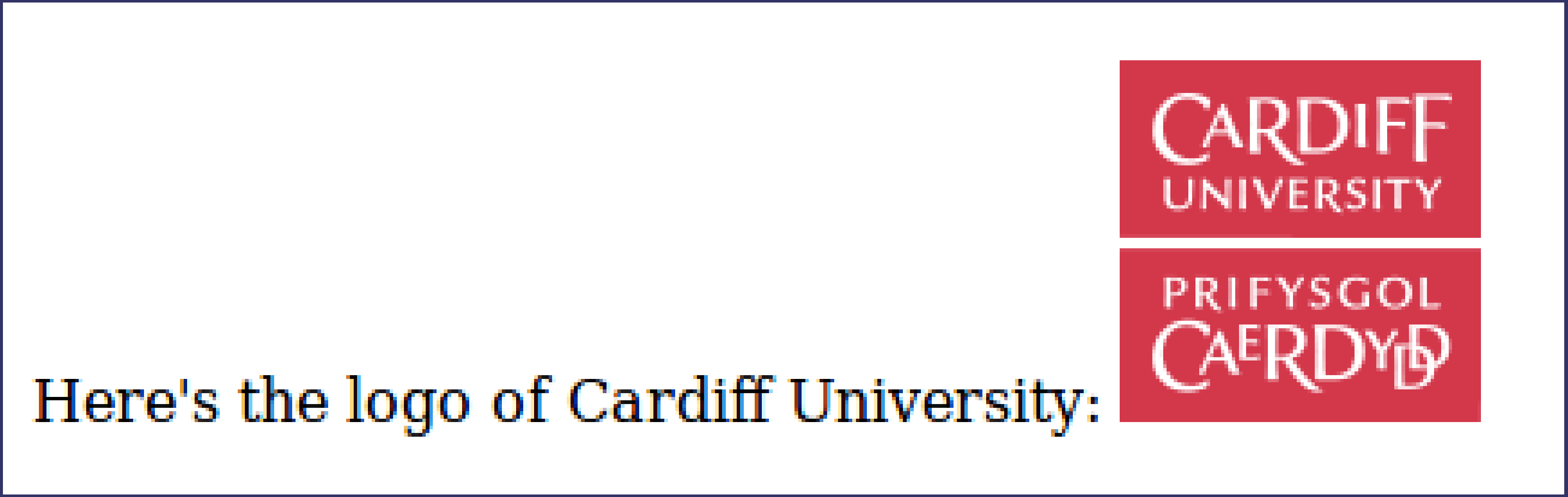
It is possible to use the height and width attributes to display the image bigger or smaller than its actual size. This is usually not a good idea (why?)

Images (3/3)

```
<p>Here's the logo of Cardiff University:

<img src="uni_logo.png" alt="Cardiff Uni Logo"
Height="100" width="100" />

</p>
```



Here's the logo of Cardiff University:

Image File Formats

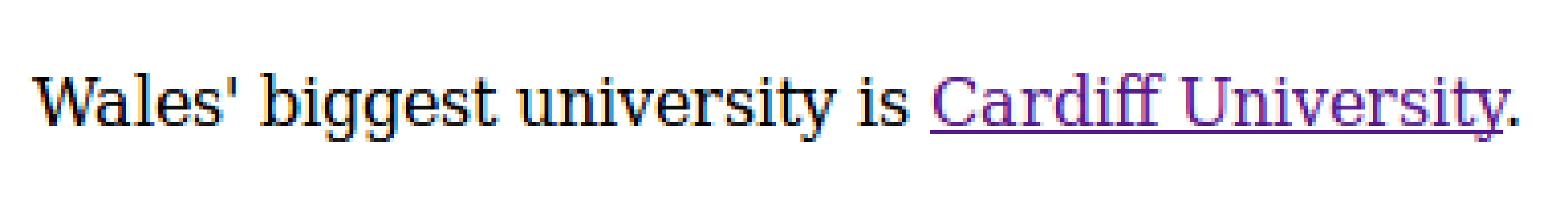
- GIF – Graphics Interchange Format (.gif)
  - Pixel based
  - Up to 256 colours (8 bits, not good for photos)
  - Lossless compression (good for drawings)
  - Options for transparency and animation (GIF89A)
- PNG – Portable Network Format (.png)
  - Pixel based
  - up to 24 bit colours
- JPEG – Joint Photographic Experts Group(.jpg/.jpeg)
  - Fractal based (sophisticated lossy compression)
  - Image quality depends on level of compression
  - Suitable for photos, less for drawings
- SVG – Scalable Vector Graphics (.svg)
  - Vector based
  - No loss of quality when resizing

Links

The *anchor* element (<a> … </a>) provides hypertext links between

- different documents (using a URL)
- different parts of the same document (using id attribute)

```
<p> Wales' biggest university is
<a href="http://www.cardiff.ac.uk">Cardiff University</a>.
</p>
```



Wales' biggest university is Cardiff University.

Absolute versus Relative Addressing

- Absolute address:
  `href="http://www.cs.cf.ac.uk/index.html"`
  (suitable for links to external sites)

- Relative address:
  `href="index.html" or href="../index.html"`
  (suitable for links to pages on the same site)

Relative addresses are related to the location of the page that uses them.

Advantage relative addresses:
you could move the entire website to a different domain name without having to change the links.

Examples Relative Addressing

Suppose the page
`http://users.cs.cf.ac.uk/CaminadaM/index.html`
contains a link
`<a href="publications.html">`
Clicking on this link results in accessing
`http://users.cs.cf.ac.uk/CaminadaM/publications.html`

Suppose the page
`http://users.cs.cf.ac.uk/CaminadaM/publications.html`
contains a link
`<a href="publications/JAIR_ABA_LP.pdf">`
Clicking on this link results in accessing
`http://users.cs.cf.ac.uk/CaminadaM/publications/JAIR_ABA_LP.pdf`

Internal Links

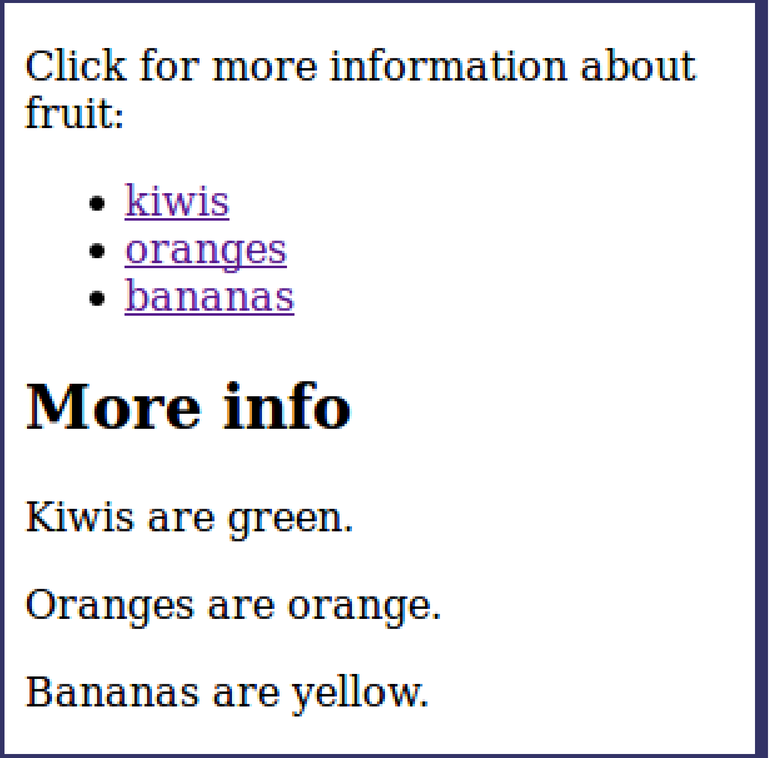
A link can also point to
a location on the same page

```
<p>Click for more information about fruit:</p>

<ul>
  <li> <a href="#kiwis">kiwis</a> </li>
  <li> <a href="#oranges">oranges</a> </li>
  <li> <a href="#bananas">bananas</a> </li>
</ul>

<h2>More info</h2>

<p id="kiwis"> Kiwis are green. </p>
<p id="oranges"> Oranges are orange. </p>
<p id="bananas"> Bananas are yellow. </p>
```



Click for more information about fruit:

- kiwis
- oranges
- bananas

More info

Kiwis are green.

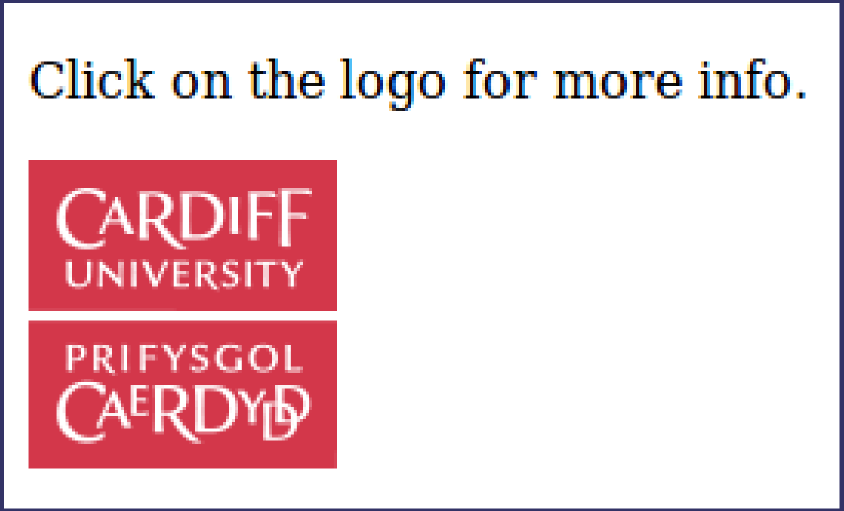
Oranges are orange.

Bananas are yellow.

Links with Images

A link (anchor) element can contain an image instead of text (or even both)

```
<p>Click on the logo for more info.</p>
<a href="http://www.cardiff.ac.uk">
    <img src="uni_logo.png" alt="Cardiff University logo"/>
</a>
```



Click on the logo for more info.

Thumbnails

Display a small image on a page,
and give users the option of displaying
a bigger version by clicking on it.

```
<a href= "big_version.jpg" target="_blank">
    <img src="small_version.jpg" alt="photo"
    width="80" height="80" />
</a>
```

`target="_blank"` *opens the image*
(or document) in a new browser window or tab

Tables

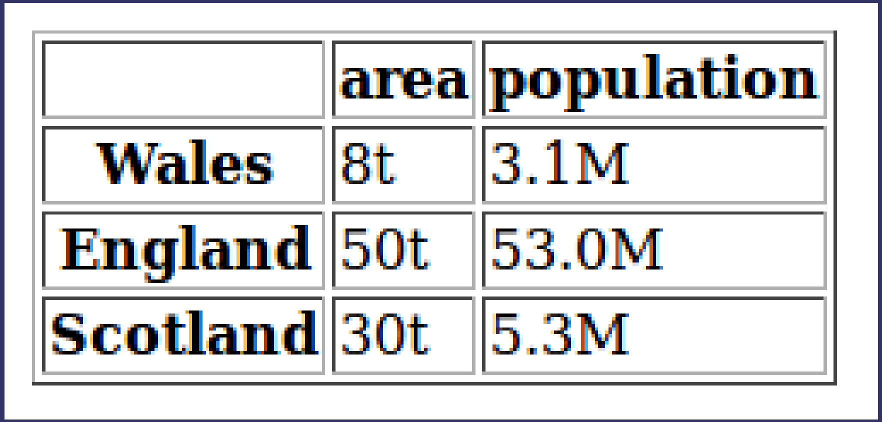
- Tables provide a means of organising the layout of data
- A table is divided into *rows* and *columns*; these specify the *cells* of the table
- Cells can contain text, links, images, etc.
- Tables used to be applied for the layout of the web page itself, but nowadays CSS is used instead

Tables

<table>	main element
<tr>	table row (inside <table> element)
<th>	table header (inside <tr> element)
<td>	table data (inside <tr> element)

Tables

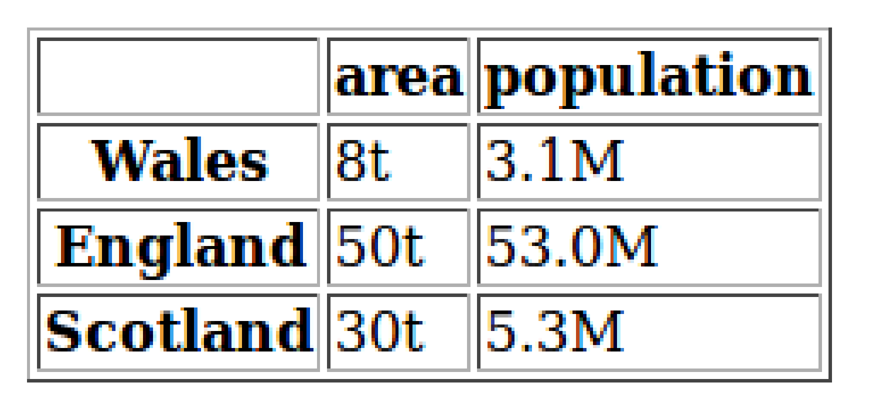
```
<table border="1">
 <tr>
  <th> </th>
  <th> area </th>
  <th> population </th>
 </tr>
 <tr>
  <th> Wales </th>
  <td> 8t </td>
  <td> 3.1M </td>
 </tr>
 <tr>
  <th> England </th>
  <td> 50t </td>
  <td> 53.0M </td>
 </tr>
 <tr>
  <th> Scotland </th>
  <td> 30t </td>
  <td> 5.3M </td>
 </tr>
</table>
```



	area	population
Wales	8t	3.1M
England	50t	53.0M
Scotland	30t	5.3M

Tables

```
<table border="1">
 <tr>
  <th> </th>
  <th scope="col"> area </th>
  <th scope="col"> population </th>
 </tr>
 <tr>
  <th scope="row"> Wales </th>
  <td> 8t </td>
  <td> 3.1M </td>
 </tr>
 <tr>
  <th scope="row"> England </th>
  <td> 50t </td>
  <td> 53.0M </td>
 </tr>
 <tr>
  <th scope="row"> Scotland </th>
  <td> 30t </td>
  <td> 5.3M </td>
 </tr>
</table>
```

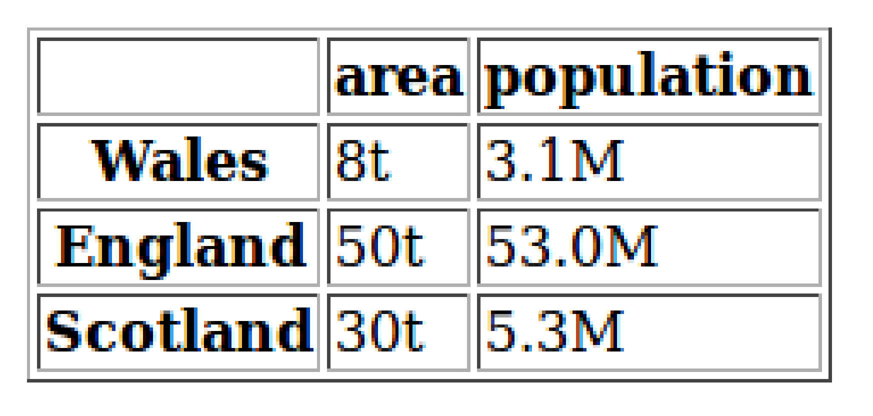


	area	**population**
Wales	8t	3.1M
England	50t	53.0M
Scotland	30t	5.3M

Use *scope* attribute to indicate whether a table header is for a row or for a column. This doesn’t affect the way the page is displayed, but can improve accessibility (like screen readers)

Tables

```
<table border="1">
 <tr>
  <th> </th>
  <th id="area"> area </th>
  <th id="pop"> population </th>
 </tr>
 <tr>
  <th id="wales"> Wales </th>
  <td headers="wales area"> 8t </td>
  <td headers="wales pop"> 3.1M </td>
 </tr>
 <tr>
  <th id="england"> England </th>
  <td headers="england area"> 50t </td>
  <td headers="england pop"> 53.0M </td>
 </tr>
 <tr>
  <th id="scotland"> Scotland </th>
  <td headers="scotland area"> 30t </td>
  <td headers="scotland pop"> 5.3M </td>
 </tr>
</table>
```

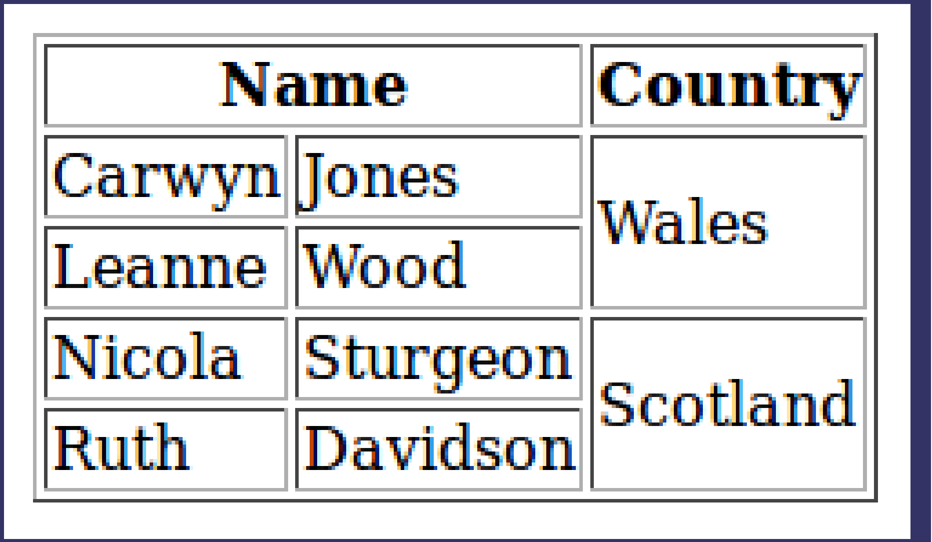


	area	**population**
Wales	8t	3.1M
England	50t	53.0M
Scotland	30t	5.3M

Alternatively, indicate for each data cell which headers it relates to

Tables

```
<table border="1">
  <tr>
    <th colspan="2"> Name </th>
    <th> Country </th >
  </tr>
  <tr>
    <td> Carwyn </td>
    <td> Jones </td>
    <td rowspan="2"> Wales </td>
  </tr>
  <tr>
    <td> Leanne </td>
    <td> Wood </td>
  </tr>
  <tr>
    <td> Nicola </td>
    <td> Sturgeon </td>
    <td rowspan="2"> Scotland </td>
  </tr>
  <tr>
    <td> Ruth </td>
    <td> Davidson </td>
</table>
```



Name		Country
Carwyn	Jones	Wales
Leanne	Wood	
Nicola	Sturgeon	Scotland
Ruth	Davidson	

Cells can span multiple columns and rows using the *colspan* and *rowspan* attributes

Tables

Other table elements:

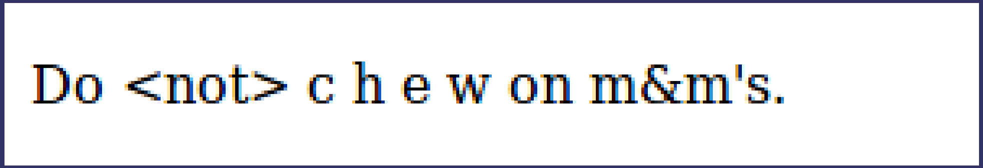
- the <caption> element puts a title above the table
- table header row can be wrapped in <thead>
  table footer row can be wrapped in <tfoot>
  main part of table can be wrapped in <tbody>
- <thead>, <tfoot> and <tbody> are only for semantic annotation and for applying CSS to rows
- <colgroup> supports applying CSS to several columns
  its attribute *span* indicates how many columns
- <col/> is an empty element, used inside of <colgroup> to apply CSS to one or more columns (using attribute *span*)

<table border="1"> is old way
of setting a border of width 1 around the cells;
nowadays people tend to use CSS instead

HTML Entity References

- Some characters (such as <, >, " and &) have a special meaning in HTML
- To prevent them from being interpreted as HTML code, they must be written as the following entity references:
  &lt;   &gt;   &quot;   &amp;
- To include a non-breaking blank space, use &nbsp;

```
Do &lt;not&gt;
c&nbsp;h&nbsp;e&nbsp;w
on m&amp;m’s.
```



Do <not> c h e w on m&m's.

HTML Entity References

- Similar notation can also be used for non-ASCII characters:

£	&pound;	ñ	&ntilde;
€	&euro;	ß	&szlig;
è	&egrave;	ç	&ccedil
é	&eacute;	α	&alpha;
ê	&ecirc;	β	&beta;
ë	&euml;	γ	&gamma;

- It is also possible to use the Unicode code points:

£	&#163;	ñ	&#241;
€	&#8364;	ß	&#223;

(etc)

More on HTML encodings of special characters:

- Appendix F of “Beginning HTML and CSS”
- http://dev.w3.org/html5/html-author/charref

More Information on HTML

- “Beginning HTML and CSS” (chapters 1-5)
- http://www.w3schools.com (when you want to know more about a particular HTML element)

HTML Validation

Browsers are not compilers!
They will try to interpret
whatever code you throw at them.

How to find out whether
your HTML code is correct:

http://validator.w3.org/

(please distinguish *warnings* from *errors*)