SWINBURNE UNIVERSITY OF TECHNOLOGY

COS10011
Creating Web Applications

Lecture 2 – HTML (part 1)

Acknowledgement: The contents in this document was adopted from learning material prepared by Alan Colman (Convener, SUT)
Last Update: 1 Feb 2017 by Jeffery Sijore

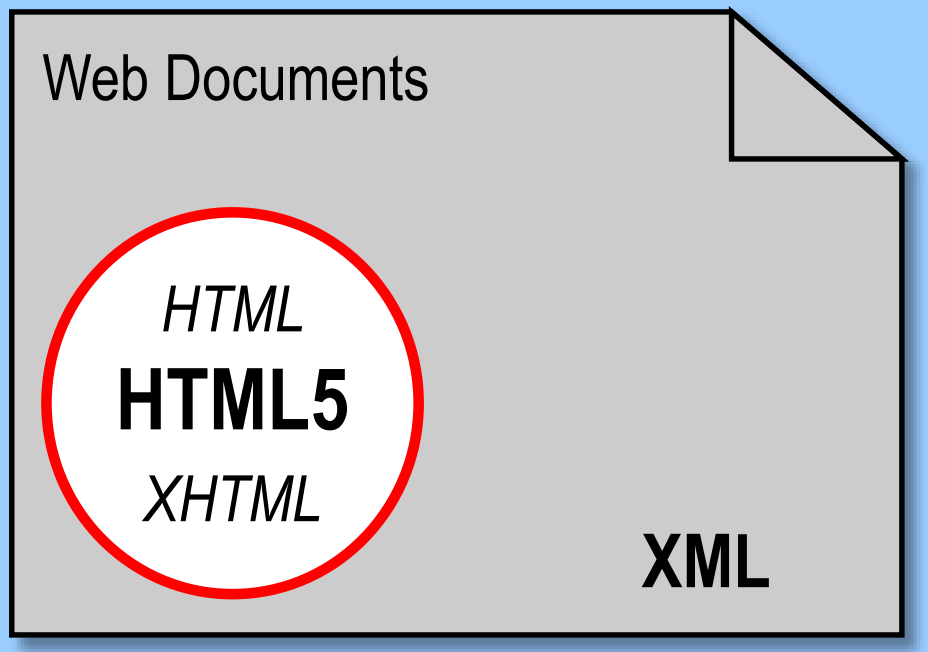
Unit of Study Outline

Internet Technologies: TCP/IP, URLs, URIs, DNS, MIME, SSL

Web Technologies: HTTP, HTTPS, Web Architectural Principles

Client Side Technologies:

Web Applications, Markup Languages



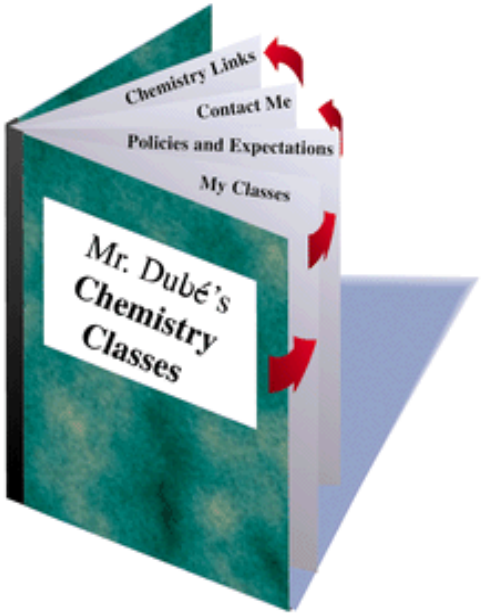
Lecture - overview

- **HTML Documents**
  - HTML and XML elements
  - HTML Head (meta information) and body (content)
- **HTML Body elements (page content)**
  - Headings and Paragraph
  - Phrase tags and Special Characters
  - Lists and Table
  - Image and Anchor
  - Form, Form Attributes and Form Elements
- **HTML Structure**

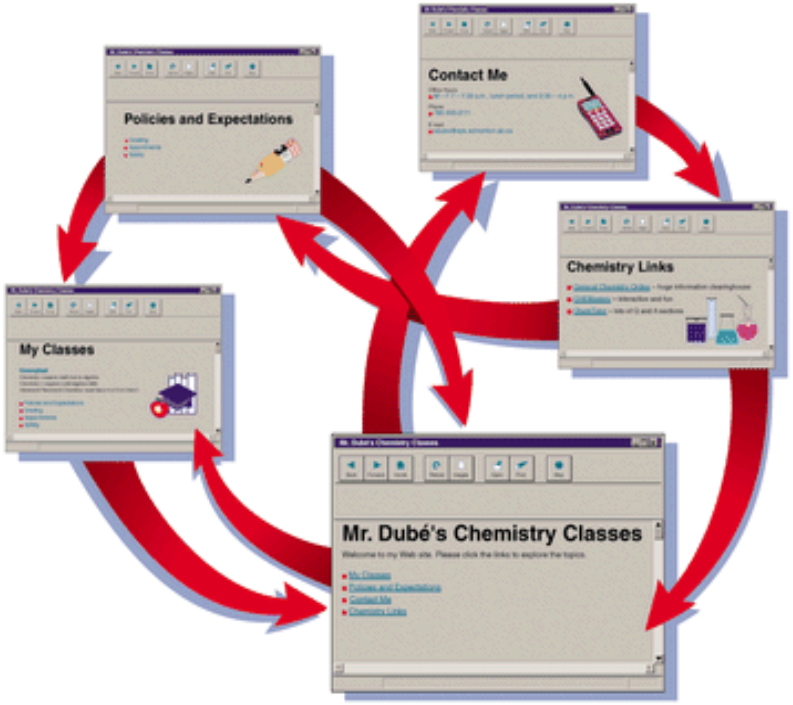
SWINBURNE UNIVERSITY OF TECHNOLOGY

Documents of the Web
HTML (and Others)

Linear vs Hypertext Documents (again)



Reading a linear document



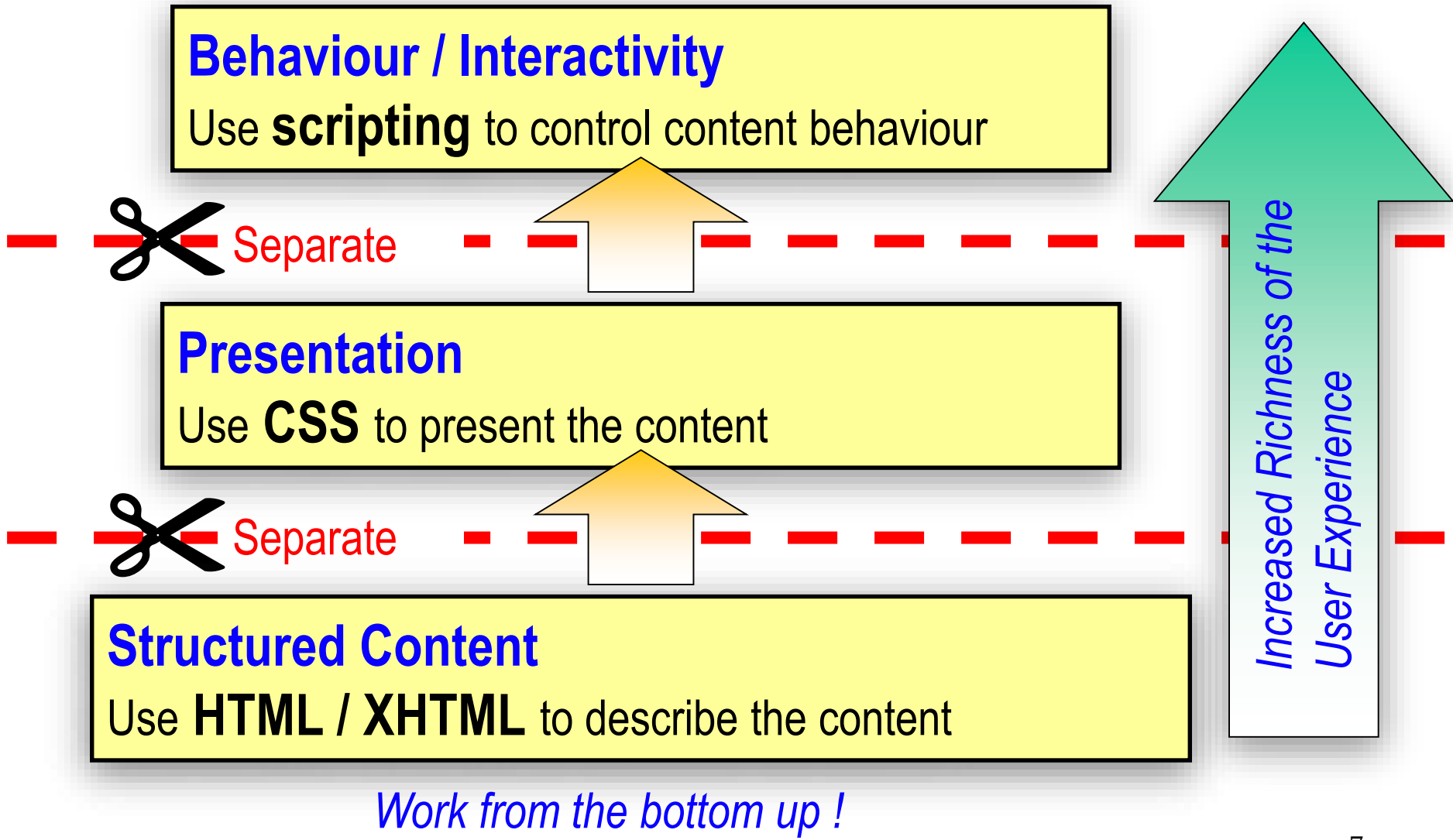
Reading a hypertext document

Web Documents

Web Pages (Web documents) are text files with

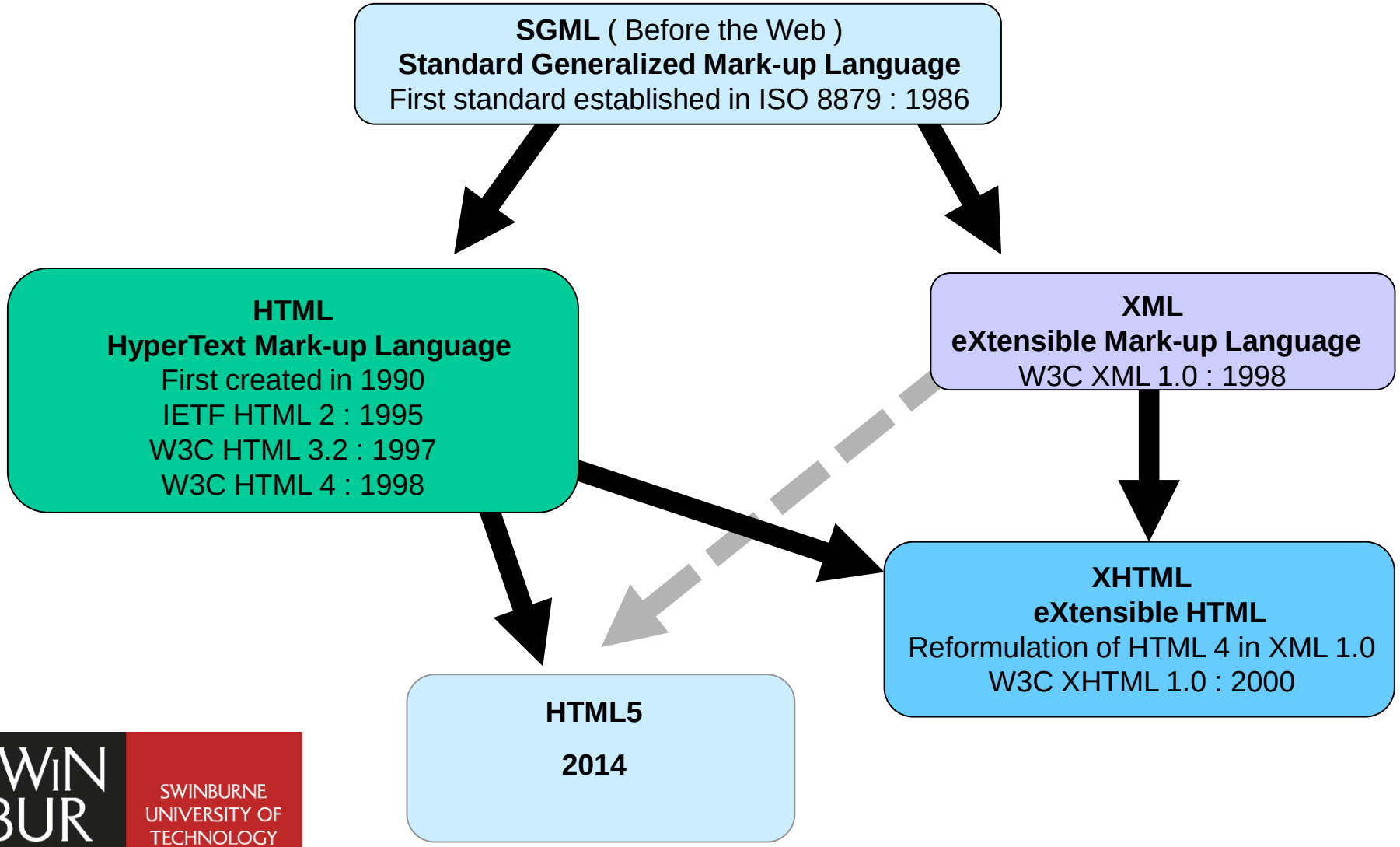
- ***HyperText Markup Language (HTML)*** or ***Extensible HyperText Markup Language (XHTML)*** used to mark-up page ***structure*** and ***content***
- ***Cascading Style Sheets (CSS)*** **applied** to HTML mark-up page ***presentation***
- ***Images / graphics*** and other ***media***, added to provide ***visual content*** and to ***enrich*** web pages
- ***JavaScript*** (for client-side scripting) to ***enhance*** web user ***interaction***.

Build your webpages using the correct tools



Work from the bottom up !

Mark-up Languages and the Web



Which HTML?

Timeline	HTML	XHTML
1995	2.0	
1997	3.2	
1997-1998	4.0 Strict, Transitional, Frameset)	
1999	4.01	
2000		1.0
2001		1.1
2008	5.0 (Draft)	
2009		2.0 (abandoned , incompatible with 1.x)
2014	5.0	
2016	5.1	

XML – eXtensible Markup Language

- XML widely used for marking up ***any structured data,*** is
  - Human readable / Machine understandable
  - Device-independent and application-independent
  - Plain text
- XML is hierarchy of data elements:
  - A "**parent**" element contains the "**children**" elements
  - Children elements of the same parent element are called "**siblings**"

XML – Simple structured data

Any structured data can be marked up with XML:

Document type

Element start tag

Element attribute

Element content

Element end tag

```
<?xml version="1.0" encoding="utf-8"?>
<booklist>
   <book stock_level="5" >
        <name> ... </name>
        <author> ... </author>
        <publisher> ... </publisher>
        <ISBN> ... </ISBN>
        <date> ... </date>
   </book>
   <book>
        ...
   </book>
</booklist>
```

Element must be:

- **Nested (no overlapping tags)**
- **Closed**

HTML: Document Structure

- HTML is closely related to XML. The data elements ("markup") represent content on a Web page.
- In this unit we write HTML so that it is compliant with XML rules
- Markup is used to create the **structure** of a document.
  - HTML elements can be "**containers**" for other elements, which can also contain other elements, and so on.
  - This kind of **nesting** for elements creates a **tree** or **hierarchy**.

You need to know the basic tree structure of web pages.

First HTML5 Example

```
<!DOCTYPE html >
<html lang="en" >
<head>
  <meta charset="utf-8" />
  <title>First HTML Example</title>
</head>
<body>
  <h1>Welcome to HTML!</h1>
  <hr />
  <p>HTML is <em>really</em> easy. It is just simple text with
   <strong>meaning</strong>.</p>
  <p>In fact, we can just keep adding text and keep typing and adding
  more characters and more typing and just go on and on.</p>
</body>
</html>
```

doctype standard

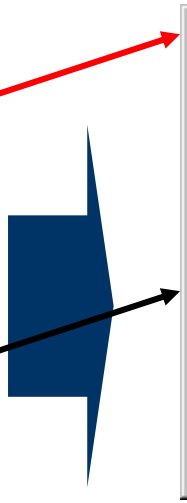
Character encoding

HTML5 will also accept <hr> but it is not XML!

HTML: Document Structure

- ***The simple basic structure of HTML documents:***

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>...</title>
    </head>
    <body>
        ... body content
        goes here ...
    </body>
</html>
```

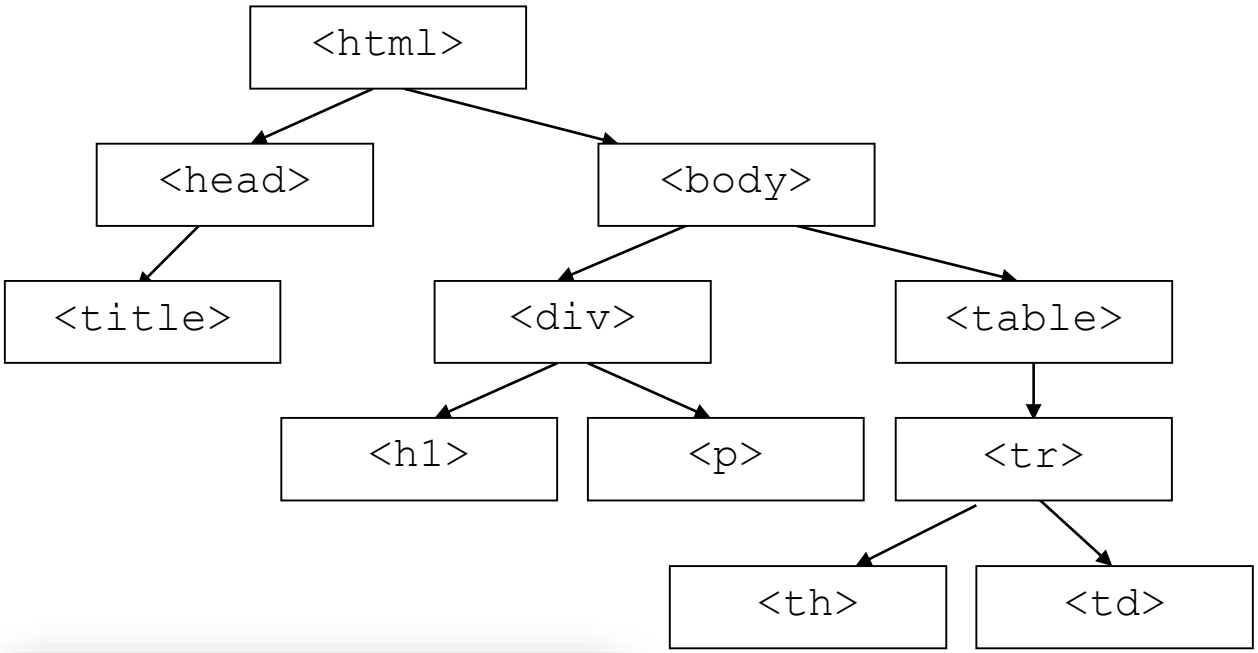


You need to know the basic structure of web pages.

HTML: Doc Structure – Tree View

- The "**root**" element of any html document, is the **`html`** element, which usually contains only two children **`head`** and **`body`**
  - The **`head`** then contains the **`title`**, and other 'head' elements.
  - The **`body`** can contain many other elements

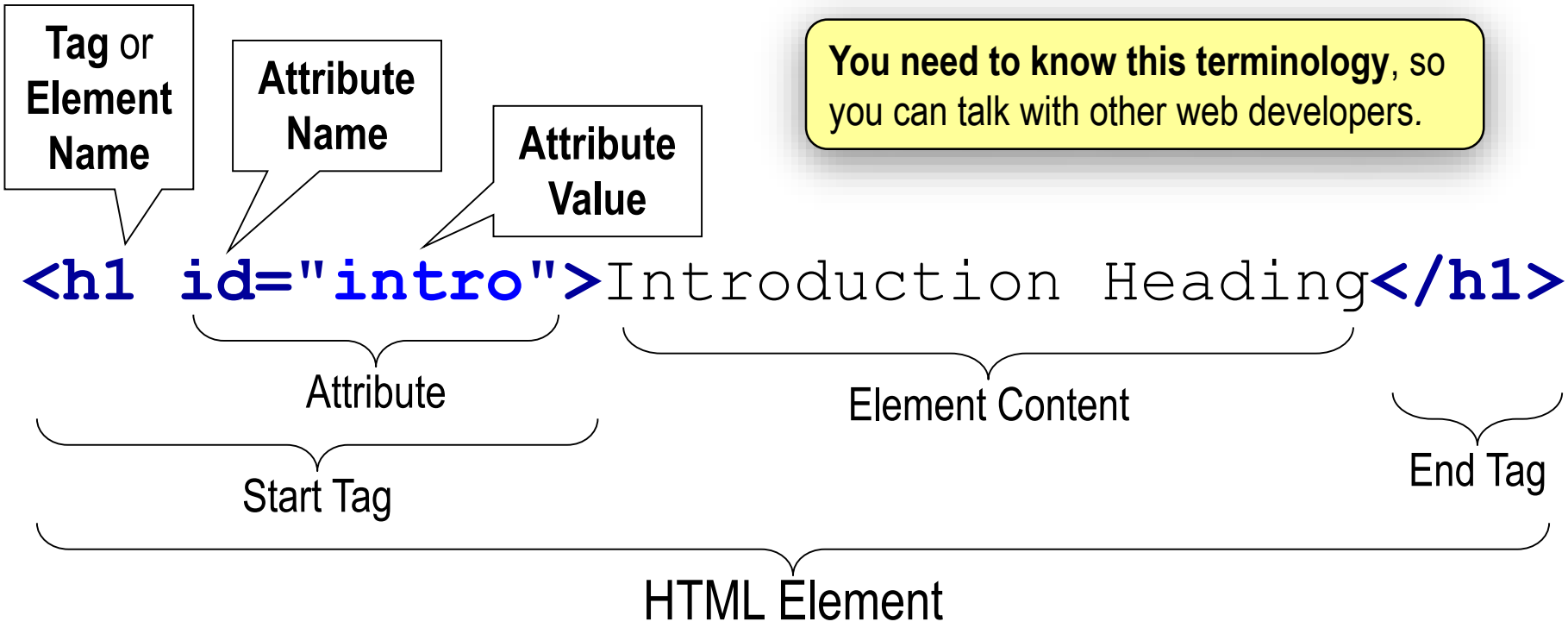
```
<html lang="en">
<head>
  <title>...</title>
</head>
<body>
  <div>
    <h1>...</h1>
    <p>...</p>
  </div>
  <table>
    <tr>
        <th>...</th>
        <td>...</td>
    </tr>
  </table>
</body>
</html>
```



You need to know the basic tree structure of web pages.

HTML Elements

- A HTML **element structure** includes: **start tag**, **tag name**, an attribute name (eg. **id**) with an **attribute value** (eg. **"intro"**), the **element content** (the text affected by the tag meaning), and finally the **end tag** of the element.



HTML Elements

- **Elements** begin with a **start tag** and
  usually finish with an **end tag**
- For example:

```
<h1>This is a major heading</h1>
<p>This is a paragraph</p>
<em>This is text that is emphasised</em>
<strong>This is really important text</strong>
```

- A **tag pair** fully encloses the **element contents**
- **Elements** *might* ***contain*** *other elements*

```
<p>content .. <em> .. Content .. </em> ..content</p>
```

  (i.e. elements might be nested)

HTML Element Attributes

Start tag can contain **attributes**

attribute *name*=*"attribute value"*

- **attributes** *modify* the meaning of a tag.

```
<input type="text" … />
<input type="submit" … />
```

Here the attribute "type" provides different "states" for input elements.

- **attribute values *should* be surrounded in a pair of quotes**
  - Either double "…" or single '…',
    eg. `id="intro"` or `id='intro'`

HTML Elements

End tag rules of HTML elements:

- Some **end tags** are optional.
  - In HTML it is ***good practice*** to: ***'close all tags'***
  - **Not optional in XHTML:** ***'must close all tags'***
- Some tags are not-paired – do not have an **end tag**
  - These are ***void elements*** - they do not have any 'element content' and hence are ***self-closing elements***
    (**Note:** *void elements* were formerly termed *empty elements*)
    There are not many ***void elements*** in HTML.
    See list of void elements later …

To be XML compliant ***void elements must self-close***:
- the **start tag** ***must*** finish with **/>** syntax

HTML Elements

Elements are either:

block-level elements or

inline elements.

- **Block-level** elements, like **`<h1>`** headings and **`<p>`** paragraphs, are usually presented by the browser with ***line breaks*** to separate them visually from other elements.
- **Inline** elements, like **`<strong>`** and **`<em>`** occur inline inside block-level elements, and usually do not introduce any “visual” breaks.

HTML5: Template

```
<!DOCTYPE html>
<!- First HTML5 Example -->
<html lang="en">
<head>
 <meta charset="utf-8" />
  <meta name="description" content="
   [description of what the doc is about] " />
  <meta name="keywords" content="
       [keywords description the document]" />
  <meta name="author"   content="[your name]" />
  <title> … </title>
</head>
<body>
   …
</body>
</html>
```

xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en"

This Lecture: Web Page CONTENT and STRUCTURE

Replace the *[italicized text]* with your code.

Do not forget to **validate** your code.

Note we will be using XML compliant HTML5 code

Demo

HTML Doc: <!DOCTYPE> declaration

- must be the very first item in the HTML document
- is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.
- In HTML 4.01, the <!DOCTYPE> declaration refers to a DTD. The DTD specifies the rules for the markup language, so that the browsers render the content correctly.
- HTML5 does not require a reference to a DTD.

HTML Doc: <!DOCTYPE> declaration

- **HTML 5**

  <!DOCTYPE html>

- **XHTML 1.0 Strict**

  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

- **XHTML 1.0 Transitional**

  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

- **Other / earlier versions**
  - **XHTML 1.0 Frameset, HTML 4.01 Strict, HTML 4.01 Transitional**

HTML Doc: HTML Tag

- represents the root of an HTML document.
- is the container for all other HTML elements
- In HTML 5,

```
<html lang="en">
…
</html>
```

- In XHTML,

```
<html xmlns="http://www.w3.org/1999/xhtml"
   lang="en" xml:lang="en">
…
</html>
```

HTML Doc: Void Element Structure

- Void elements are elements with no end tag
- In HTML5

```
<meta charset=utf-8>
<hr>
<br>
<img … >
```

- In XHTML, all void elements must be properly closed

```
<meta charset="utf-8" />
<hr />
<br />
<img … />
```

It is best practice to code html5 so it is XML compliant

HTML5: Head and Title Tag

- **<head>…</head>** is the container for all the head elements.
  - **must** include a **title** for the document, and can include scripts, styles, meta information, and more.
- **<title>…</title>** defines the title for the HTML document:

```
<head>
  <title>HTML 5 Page</title>
</head>
```

HTML5: Meta Tag

- Describe metadata within an HTML document
- Place as part of the <head> element content

```
<head>
  <meta charset="utf-8"/>
  <meta name="description"
     content="Web development" />
  <meta name="keywords"
     content="HTML,CSS,JavaScript"/>
  <meta name="author"
     content="Your Name" />
  …
</head>
```

HTML5: Comment Tag

- **<!-- .. -->** used to insert comments or explanation in the source code
- It is not displayed in the browsers.
- It is use to "hide" scripts from browsers without support for it, so they don't show them as plain text:

```
<!- First HTML5 Example -->
```

Lecture - overview

- HTML Documents
  - HTML and XML elements
  - HTML Head (meta information) and body (content)
- **HTML Body elements (page content)**
  - Headings and Paragraph
  - Phrase tags and Special Characters
  - Lists and Table
  - Image and Anchor
  - Form, Form Attributes and Form Elements
- HTML Structure

'Containers' for text, images etc.

Allow content to be set out on the page into meaningful blocks. (Next week)

HTML: Quick Start Elements

- ***Here is a quick start list of some common elements to get you going.***

headings: `<h1>…</h1>` `<h2>…</h2>` to `<h6>…</h6>`

paragraphs: `<p>…</p>`

line breaks: `<br />`

horizontal rule: `<hr />`

inline image: `<img src="url" alt="" height="" width="" />`

strong or *emphasized* text: `<strong>…</strong>` and `<em>…</em>`

unordered / ordered list: `<ul>…</ul>` and `<ol>…</ol>`

list items: `<li>…</li>`

hypertext link: `<a href="url">…</a>`

The URL value can be relative or start with `http://`, `ftp://`, `mailto:` etc.

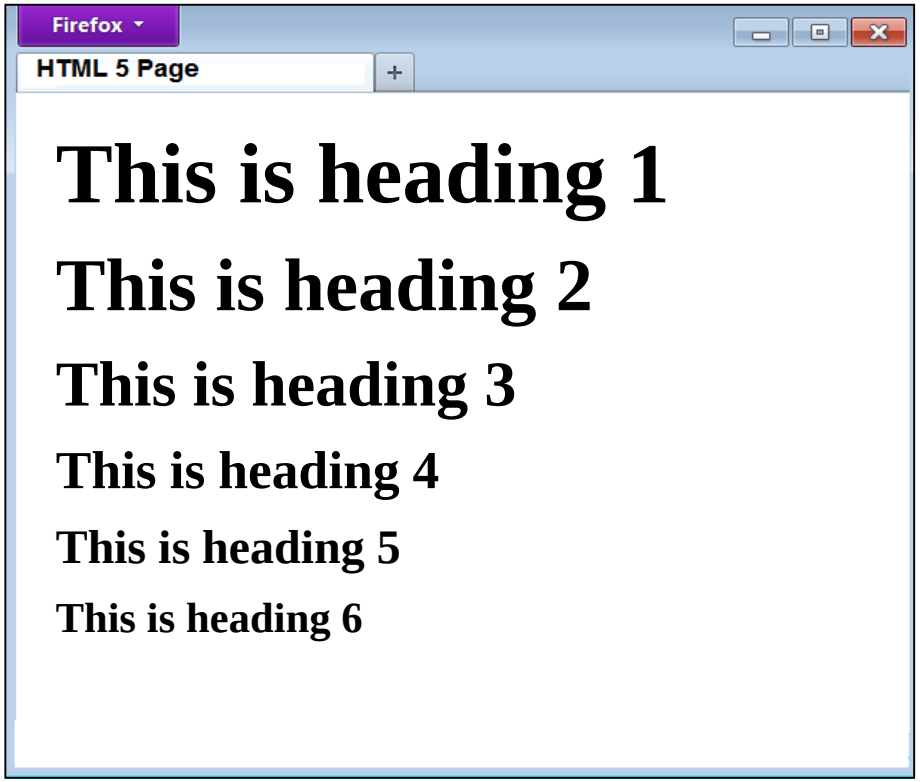
Lecture - overview

- HTML Documents
  - HTML and XML elements
  - HTML Head (meta information) and body (content)
- **HTML Body elements (page content)**
  - Headings and Paragraph
  - Phrase tags and Special Characters
  - Lists and Table
  - Image and Anchor
  - Form, Form Attributes and Form Elements
- HTML Structure

Headings (continued)

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
```

Default Firefox rendering

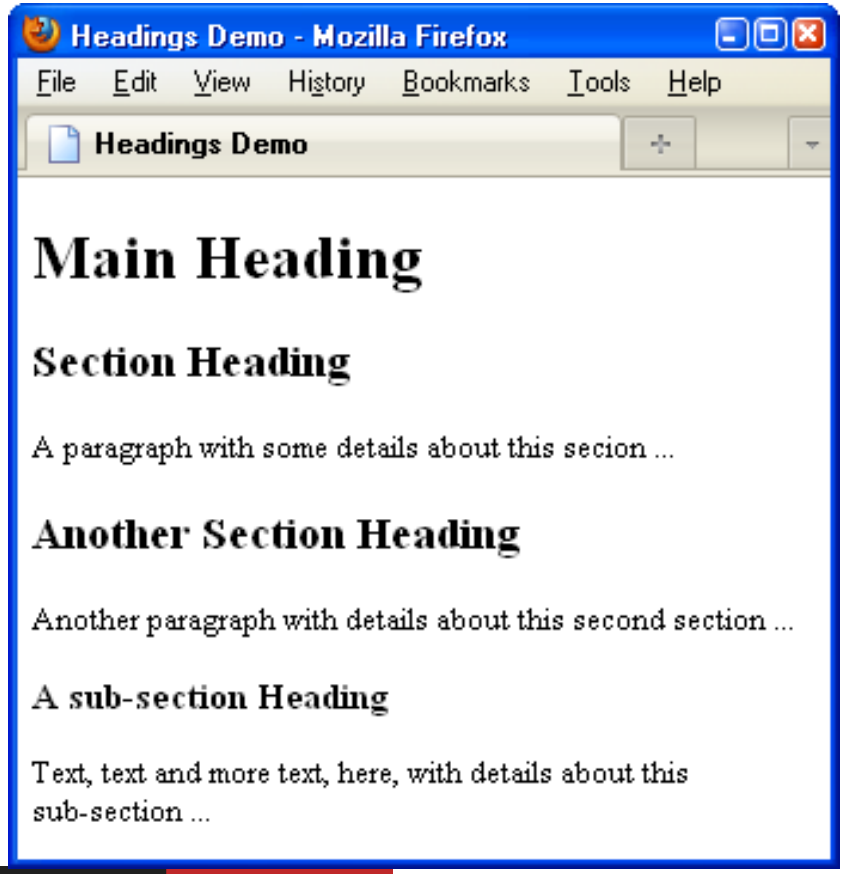


Heading Elements

- **<h#>…</h#>** is a logical block level element used to mark the significance of a heading, where # is a number from 1 to 6
- There are **six (6)** levels of **importance** from the **most** important **<h1>** to the **least** important **<h6>**.
  - ***Do not*** *skip heading levels. If the next heading is one level below the last heading, only use the next heading level.*
- Browsers display all headings larger and/or bolder than normal text,
  - ***Do not*** *use headings simply as a way to increase font size and make the text bold (use CSS instead)*

Headings (continued)

- Example:



Headings are logical markup, used to convey the order of importance of content.

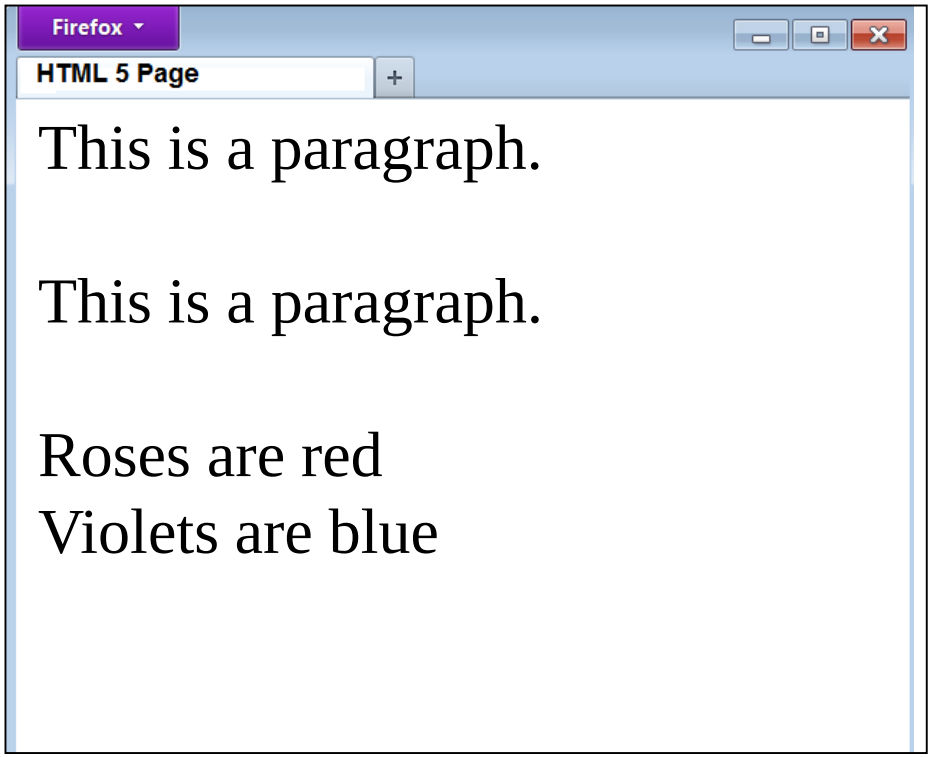
```
...
<h1>Main Heading</h1>
<h2>Section Heading</h2>
   <p>A paragraph with some details
   about this secion ...</p>
<h2>Another Section Heading</h2>
   <p>Another paragraph with details
   about this second section ...</p>
   <h3>A sub-section Heading</h3>
     <p>Text, text and more text, here,
     with details about this sub-section
     ...</p>
...
```

Paragraph Element

- ** … ** is a logical block level element used to mark **paragraphs**.
  - Note: *cannot contain other block-level elements*
  - Browsers will generally place white space before and after a paragraph – it is a block level element.
- **
** an empty / void inline element used to insert a single line break.
  - ***Do not*** *use line breaks to separate paragraphs.*

Paragraph (continued)

```
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>Roses are red<br />Violets are blue</p>
```



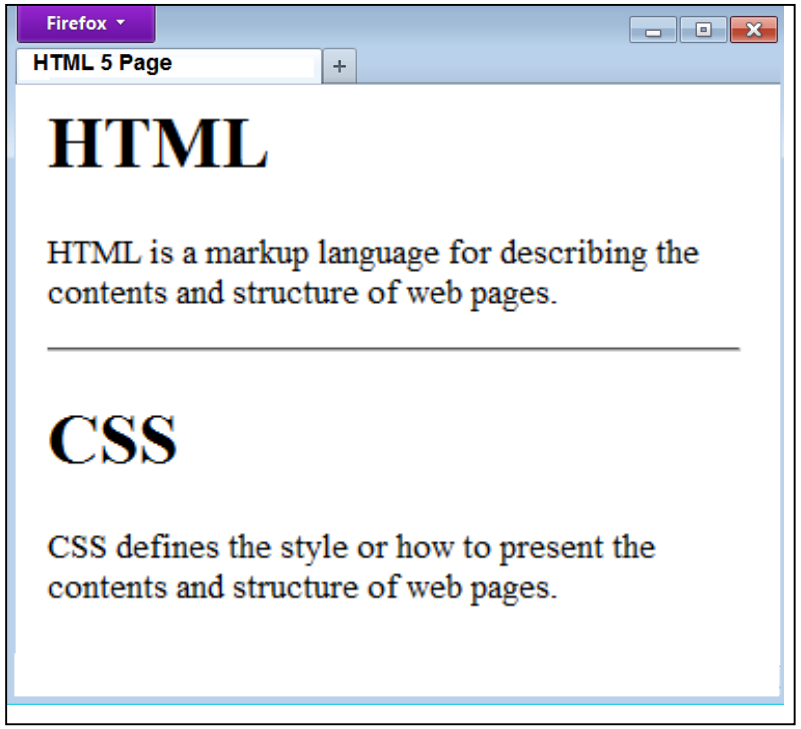
Horizontal Rule

- **<hr />** an empty / void block level element used to define a thematic break in an HTML page, or a shift of topic. It is used to separate content in an HTML page.

```
<h1>HTML</h1>
<p>HTML is a markup language for
describing the contents and
structure of web pages.</p>

<hr />

<h1>CSS</h1>
<p>CSS defines the style or how
to present the contents and
structure of web pages.</p>
```



Lecture - overview

- HTML Documents
  - HTML and XML elements
  - HTML Head (meta information) and body (content)
- **HTML Body elements (page content)**
  - Headings and Paragraph
  - *Phrase tags and Special Characters*
  - Lists and Table
  - Image and Anchor
  - Form, Form Attributes and Form Elements
- HTML Structure

Phrase Elements

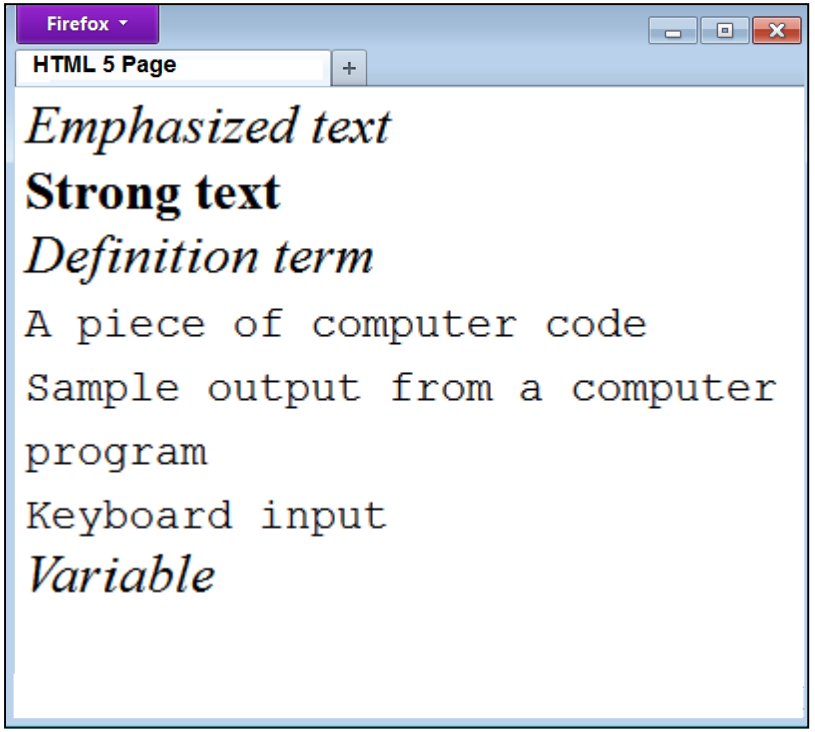
- ** <dfn> <code> <samp> <kbd> <var>** are logical ***inline*** phrase elements that define the meaning of the enclosed text

	Defines emphasized text – *renders as italics*
	Defines important text – *renders as bold*
<dfn>	Defines a definition term
<code>	Defines a piece of computer code
<samp>	Defines sample output from a computer program
<kbd>	Defines keyboard code
<var>	Defines a variable

- *Do not use just for presentation*

Phrase Elements (continued)

```
<p>
<em>Emphasized text</em><br />
<strong>Strong text</strong><br />
<dfn>Definition term</dfn><br />
<code>A piece of computer
code</code><br />
<samp>Sample output from a
computer program</samp><br />
<kbd>Keyboard input</kbd><br />
<var>Variable</var>
</p>
```



Default Firefox rendering

Phrase Elements (continued)

- **…** *should be avoided*. Use
  Defines a part of text in an alternate voice or mood.
  The content of the tag is usually rendered in italics
  - The tag can be used to indicate a technical term, a phrase from another language (eg. scientific name), a thought, or a ship name, etc.
- **…** *should be avoided*. Use
  According to the HTML 5 specification, use only as a **LAST** resort, when no other tag is more appropriate

Deductions if used in Assignments

Phrase Elements (continued)

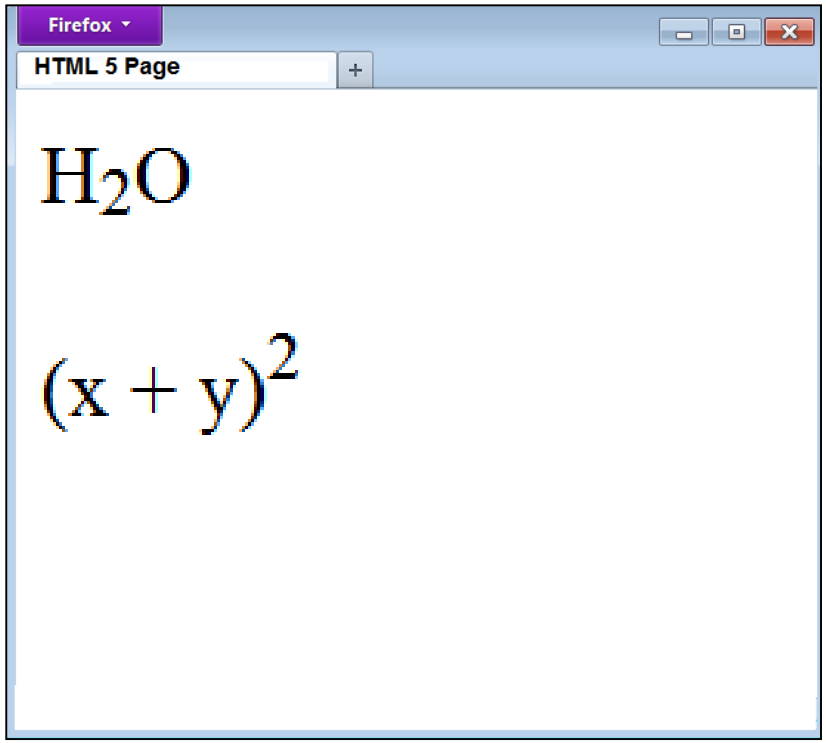
Superscript and Subscript

- **[…]** defines superscript text
  - It appears as a half character above the baseline
  - Often used to show an exponent in a mathematical equation such as $(x + y)^2$ or a footnote / citation reference.
- **_{...}** defines subscript text.
  - It appears as a half character below the baseline.
  - Often used in chemical formulas, such as H_2O.

Phrase Elements (continued)

```
<p>H<sub>2</sub>O</p>

<p>(x + y)<sup>2</sup></p>
```

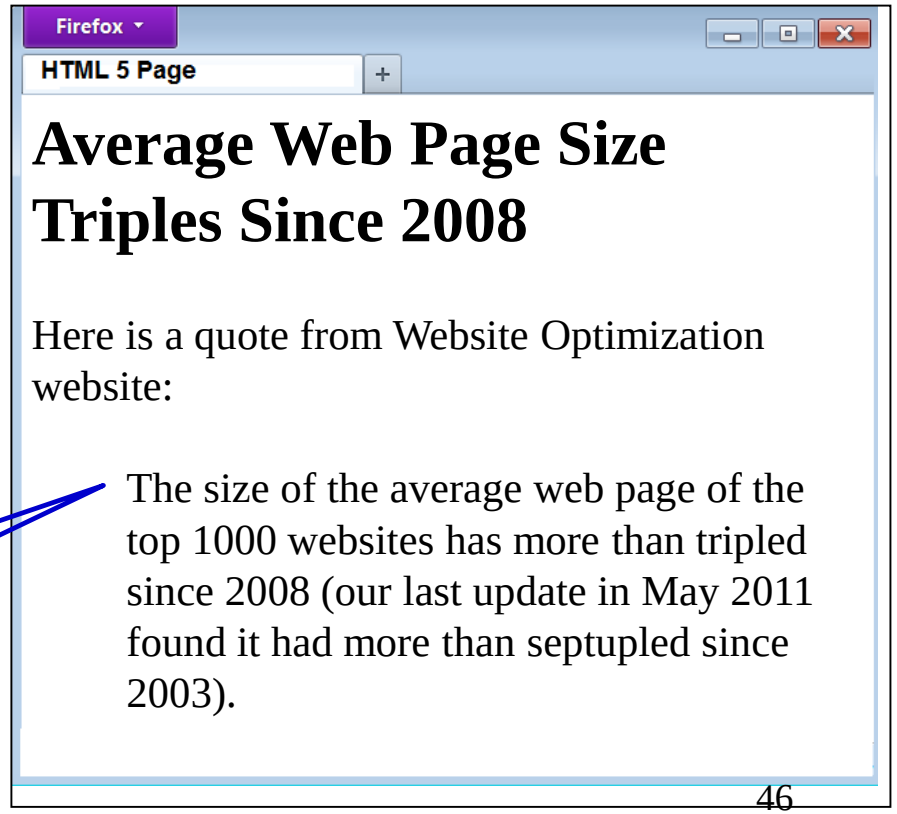


Block Quote

- **<blockquote>…</blockquote >** a logical block level element used to specify a section that is quoted from another source.
- Cite the source of quote using a 'cite' attribute

```
<body>
<h1>Average Web Page Size Triples Since
2008</h1>
<p>Here is a quote from Website
Optimization website:</p>
<blockquote
cite="http://www.websiteoptimization.co
m/speed/tweak/average-web-page/">
The size of the average web page of the
top 1000 websites has more than tripled
since 2008 (our last update in May 2011
found it had more than septupled since
2003).
</blockquote>
</body>
```

Browsers usually indent <blockquote> elements



Special Characters

- To encode reserved characters in HTML into the contents, special characters **&…;** are used
- A more descriptive term is entity encoding
- Some of the common codes are listed below:

Character	Decimal Entity Number	Named Entity	Description
"	&#34;	&quot;	quotation mark
'	&#39;	&apos;	apostrophe
&	&#38;	&amp;	ampersand
<	&#60;	&lt;	less-than
>	&#62;	&gt;	greater-than

HTML: Special Characters (continued)

Character	Decimal Entity Number	Named Entity	Description
			non-breaking space
©	©	©	copyright
			soft hyphen
®	®	®	registered trademark
¯	¯	¯	spacing macron
°	°	°	degree
±	±	±	plus-or-minus
×	×	×	multiplication
÷	÷	÷	division

These names not XML Compliant

http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references

Character Codes: http://character-code.com/

http://www.html-5.com/cheat-sheet/html-character-codes.html

Lecture - overview

- HTML Documents
  - HTML and XML elements
  - HTML Head (meta information) and body (content)
- **HTML Body elements (page content)**
  - Headings and Paragraph
  - Phrase tags and Special Characters
  - *Lists and Table*
  - Image and Anchor
  - Form, Form Attributes and Form Elements
- HTML Structure

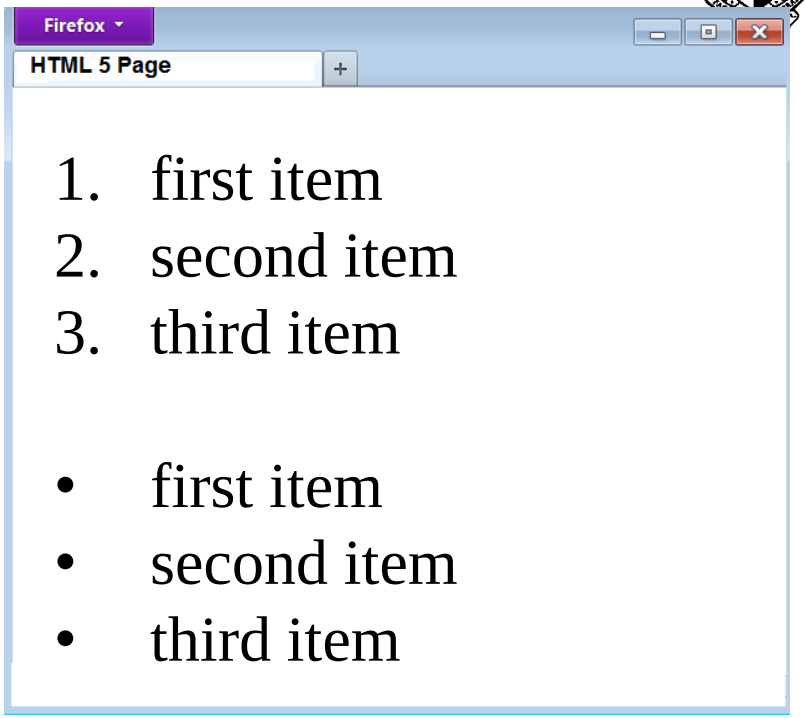
Lists

- **Ordered** list example

```
<ol>
  <li>first item</li>
  <li>second item</li>
  <li>third item</li>
</ol>
```

- **Unordered** list example

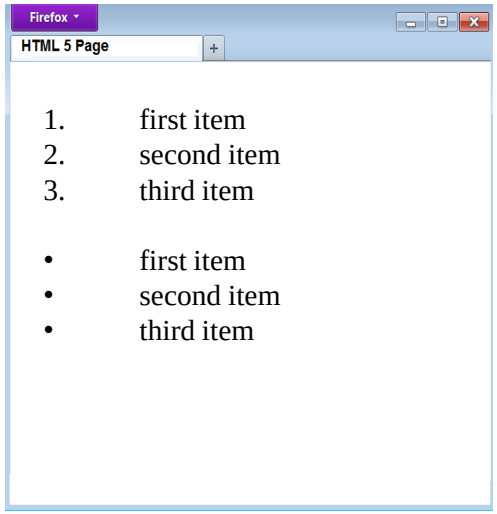
```
<ul>
  <li>first item</li>
  <li>second item</li>
  <li>third item</li>
</ul>
```



List elements

- **
…
** is used to mark each list item.
  <li> *is the only element that is allow directly in an ordered* **<ol>** *or unordered* **<ul>** *list.*

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

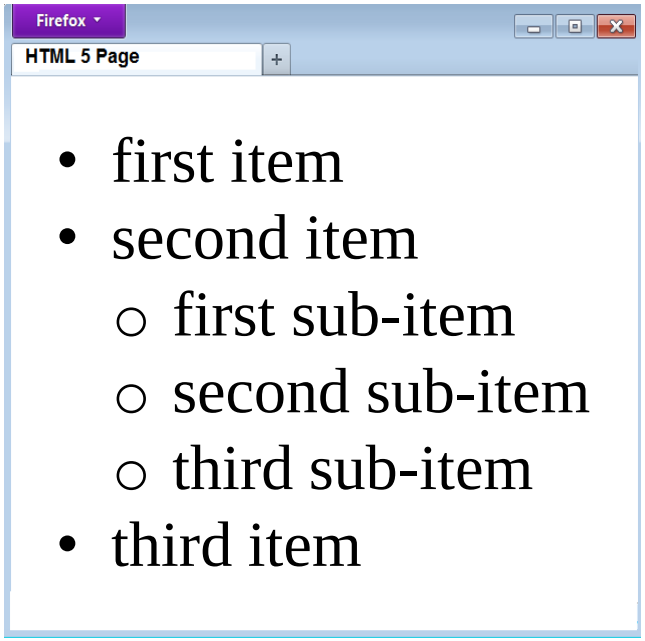



```
<ul>
  <li>item 1</li>
  <p>Paragraph</p>
  <li>item 2</li>
</ul>
```

Nested Lists

- **Nested** list example:

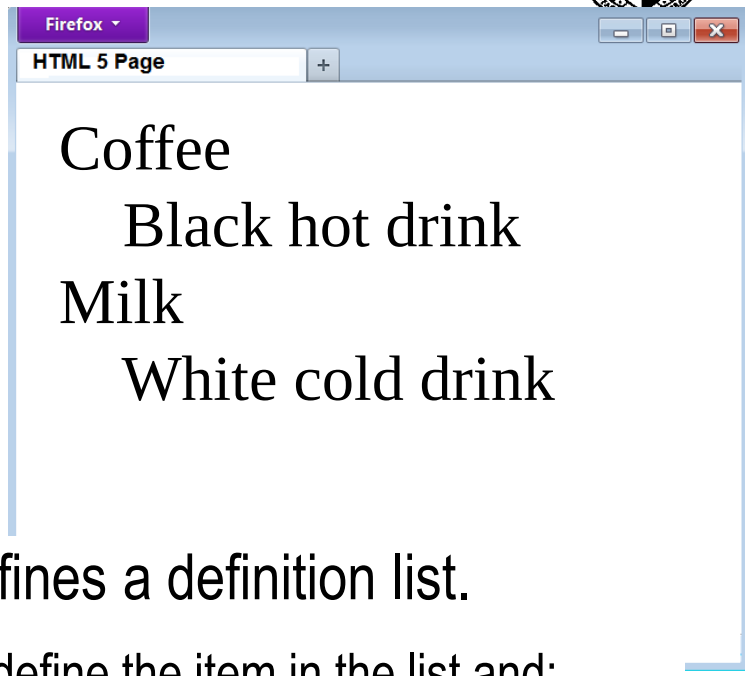
```
<ul>
   <li>first item</li>
   <li>second item
      <ul>
        <li>first sub-item</li>
        <li>second sub-item</li>
        <li>third sub-item</li>
      </ul>
   </li>
   <li>third item</li>
</ul>
```



Nested list must be inside a list item

Definition List

```
<dl>
   <dt>Coffee</dt>
   <dd>Black hot drink</dd>
   <dt>Milk</dt>
   <dd>White cold drink</dd>
</dl>
```



- **<dl>…</dl>** element defines a definition list.
  - **<dt>…</dt>** is used to define the item in the list and;
  - **<dd>…</dd>** is used to describe the item in the list
- The browser will render the item and the definition on separate lines, and the definition will be indented
- **Do not** use definition list to create second level indentation

Table (continued)

border can only be 1 (show a border) or 0 (do not show a border) – better in CSS

```
<table border="1">
  <caption>Table of Monthly Savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
  <tr>
    <td>Total</td>
    <td>$180</td>
  </tr>
</table>
```

Note: by default the <th> cells are presented bold and centred !

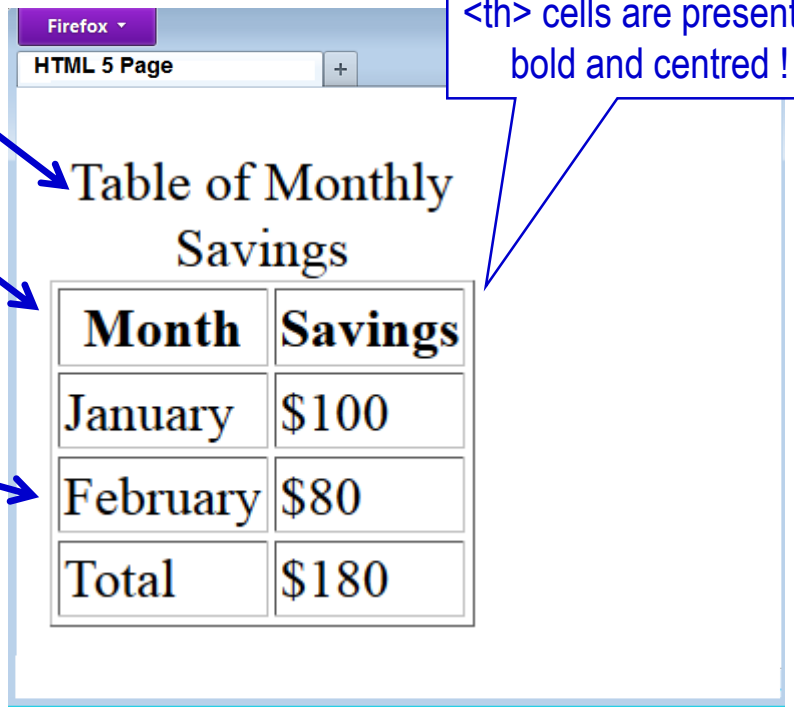
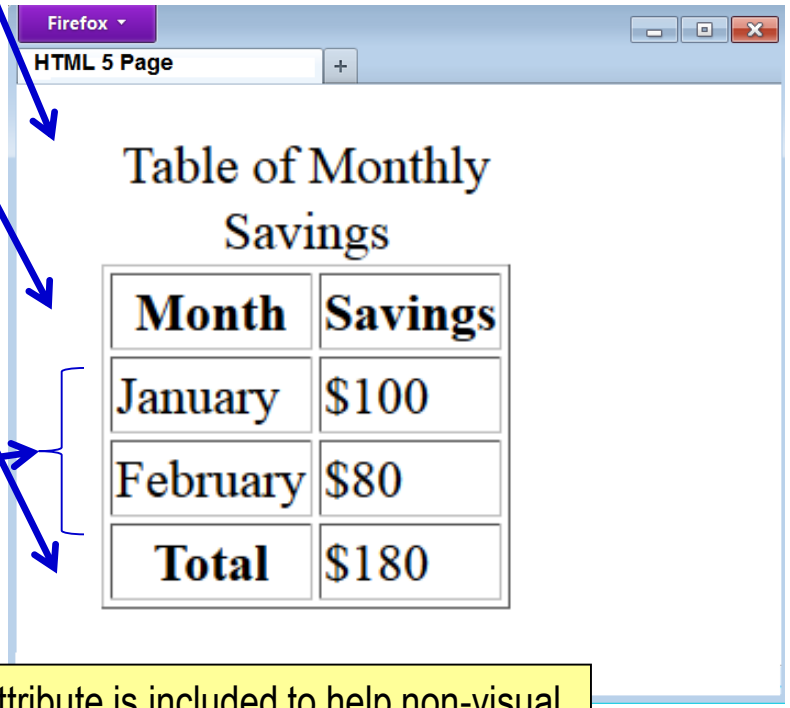


Table element

- **<table> …</table>** block level element offers a powerful way to organise *data* in a *tabular format*.
  - ***Do not*** *used table for page layout presentation.*
  - **border** is the only specific attribute supported in HTML5
- Table elements:
  - <table> … </table> declares a table
  - <caption> … </caption> captions a table's contents
  - <tr> … </tr> defines a table ***row***
  - <th> … </th> defines a table header ***cell***
  - <td> … </td> defines a table data ***cell***
  - <thead>, <tfoot>, <tbody> defines table ***sections***

Table (continued)

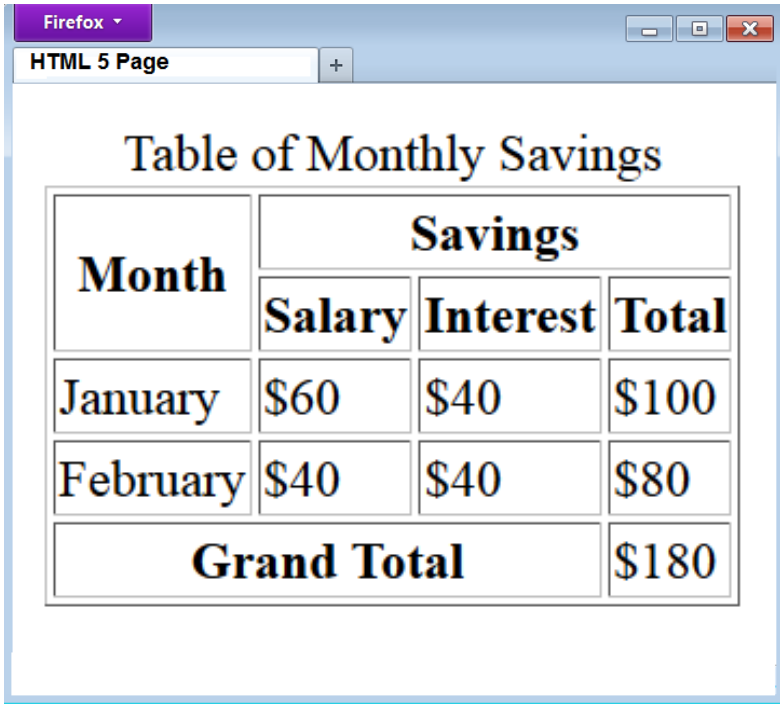
```
<table>
  <caption>Table of Monthly Savings</caption>
  <thead>
    <tr>
      <th scope="col">Month</th>
      <th scope="col">Savings</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th scope="row">Total</th>
      <td>$180</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
</table>
```



Note: the **scope** attribute is included to help non-visual devices 'understand' the tabular data.
For more complex tables, the **header** attribute could be used in each table data cell, referencing the **id** of the table header cell(s).

Table (continued)

```
<table border="1">
  <caption>Table of Monthly Savings</caption>
  <thead>
    <tr>
      <th rowspan="2">Month</th>
      <th colspan="3">Savings</th>
    </tr>
    <tr>
      <th>Salary</th>
      <th>Interest</th>
      <th>Total</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th colspan="3"> Grand Total</th>
      <td>$180</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>January</td>
      <td>$60</td>
      <td>$40</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$40</td>
      <td>$40</td>
      <td>$80</td>
    </tr>
  </tbody>
</table>
```



rowspan and **colspan** attributes for <td> and <th> allows a more complex table to be built.

Other attributes should be added to **th** and **td** to aid ***accessibility***, such as '**scope**' and '**header**'

Lecture - overview

- HTML Documents
  - HTML and XML elements
  - HTML Head (meta information) and body (content)
- **HTML Body elements (page content)**
  - Headings and Paragraph
  - Phrase tags and Special Characters
  - Lists and Table
  - *Image and Anchor*
  - Form, Form Attributes and Form Elements
- HTML Structure

Image element

- **** is an 'empty' inline element that defines an image in an HTML page.
  - Must have the two required attributes: **src** and **alt**

```
<img src="smiley.png" alt="Smiley face" />
```

- The three well supported image formats for the web are:
  - PNG (Portable Networks Graphics) *.png
  - JPEG (Joint Photographic Experts Group) *.jpg or *.jpeg
  - GIF (Graphics Interchange Format) *.gif
- Images can be edited / created using software such as:
  - GIMP www.gimp.org
  - Inkscape www.inkscape.org
  - Adobe Photoshop www.photoshop.com
  - IrfanView www.irfanview.com

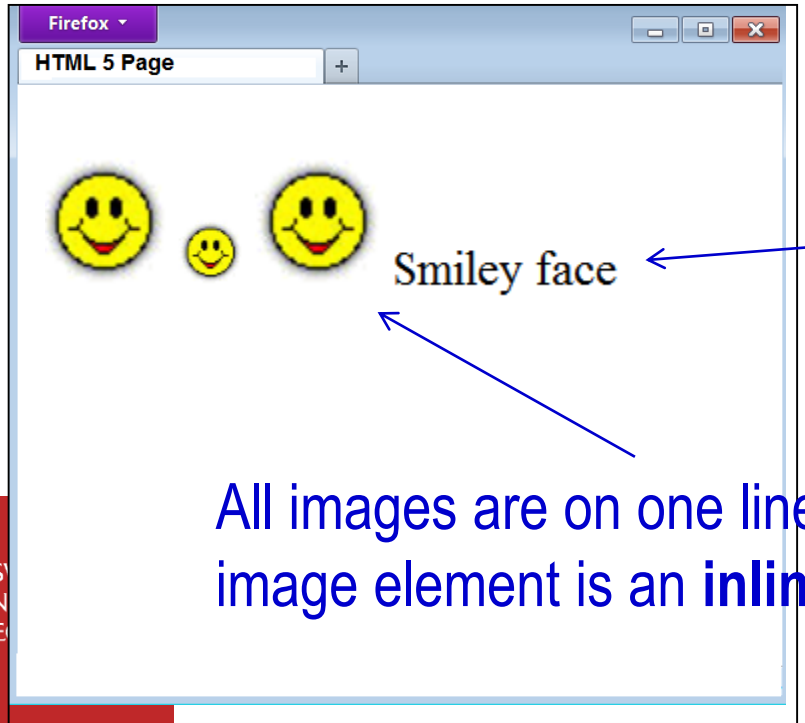
Image (continued)

- **src** attribute - indicates the location and filename of the image
  - A **relative** or **absolute (URL)** path can be used
- **alt** attribute - for clients that do not support image display (or if the user has "turned off" images).
- **height** and **width** attributes – used to set and reserve the dimensions of the image on the web page.
  (This speeds page loading)

***Do not** use height and width attributes to force a large image to "scale down" in appearance.*
Resize the image first in an image editing program to the final on-page size and put the correct height and width size in the attribute values.

Image (continued)

```
<p>
<img src="smiley.png" alt="Smiley face" />
<img src="smiley.png" alt="Smiley face" height="21" width="21" />
<img src="smiley.png" alt="Smiley face" height="42" width="42" />
<img src="ismily.png" alt="Smiley face" />
</p>
```



alt displays if image is not found

All images are on one line, as the image element is an **inline** element.

Hyperlinks - Anchor element

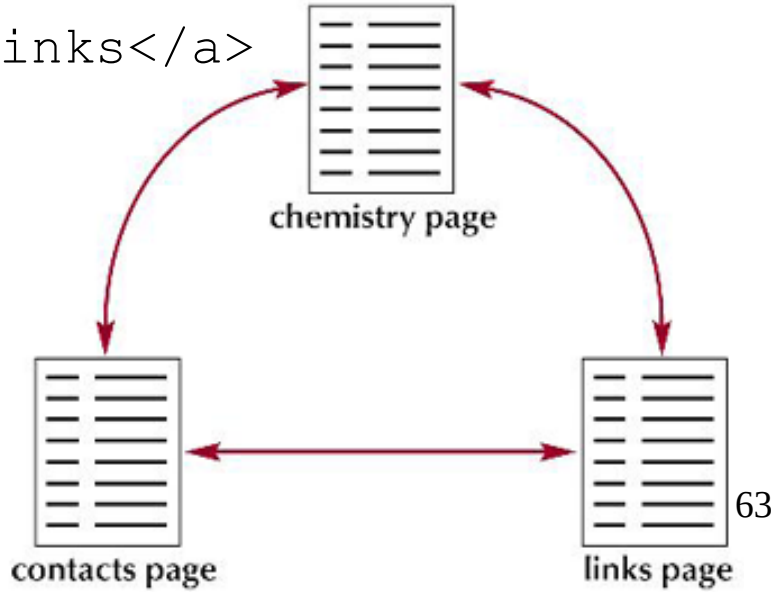
- **<a> … ** is an inline element that defines a hyperlink, used to link from one page to another.
- The **href** attribute - indicates the resource's location for the hyperlink
  - A **relative** or **absolute (URL)** path can be used
  - Resource can be a
    - A web page or a location within a web page
    - a file
    - An email address
- In HTML5, the <a> element can only be a hyperlink.

An image can be used as a hyperlink, by nesting an element as the anchor's element content

Links to Other Pages

- A web site often has a need to link between related pages.
- If all the pages are located in the ***same folder***, the links can be simple.

```
<a href="chemistry.html">Chemistry</a>
<a href="contacts.html">Contacts</a>
<a href="links.html">Links</a>
```



Note: Specifying Resource Locations

- **Relative path** is best used in specifying resources, such as files, images and web pages
  - ☐ It allows a whole folder of web pages to be moved from the "testing" environment to the "live" environment, and still retain their relative paths

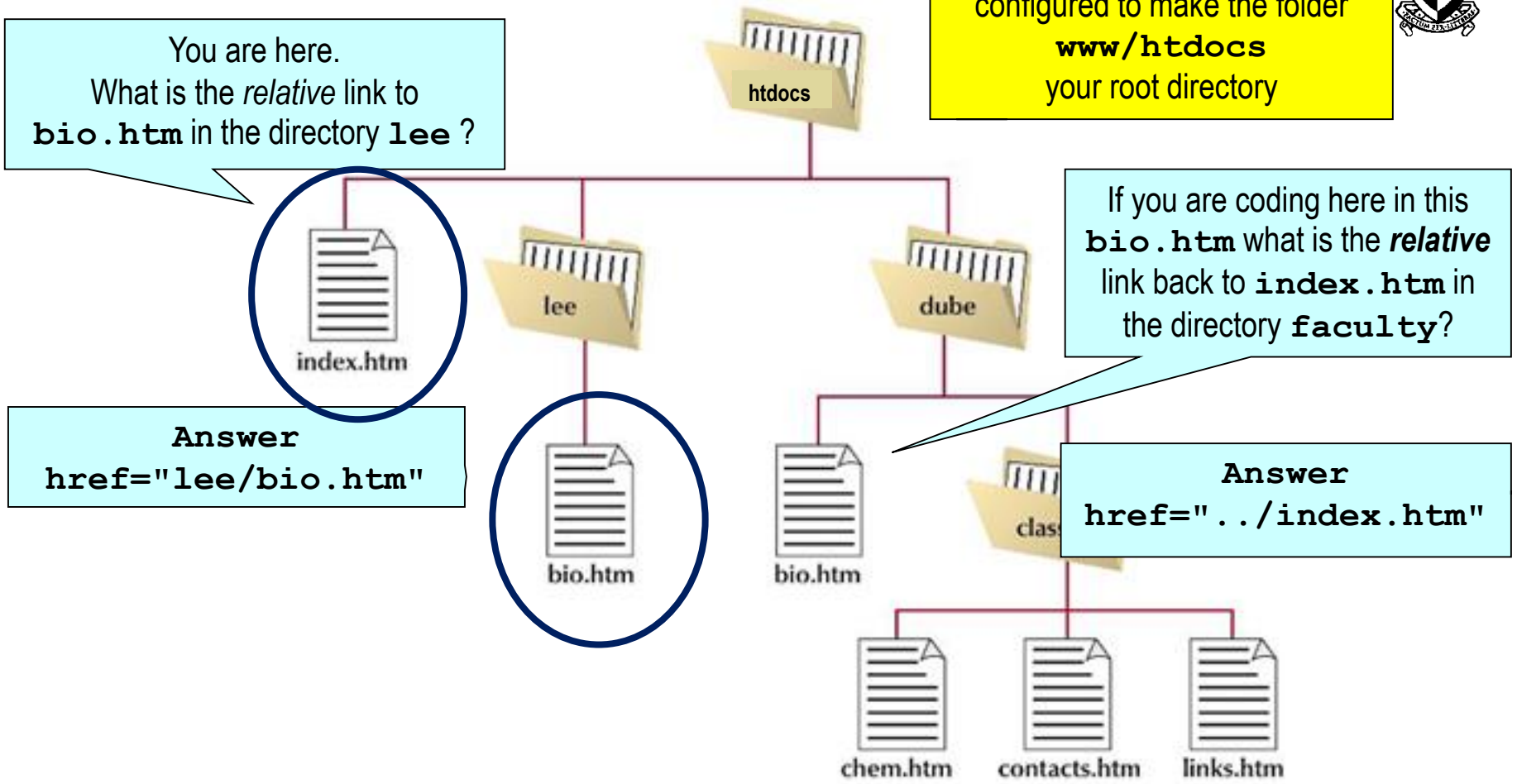
  e.g. **"index.html", "images/logo.png"**

  avoid referencing your home directory **"/images/logo.png"**
- **Absolute path** is used if the resource is not part of the website

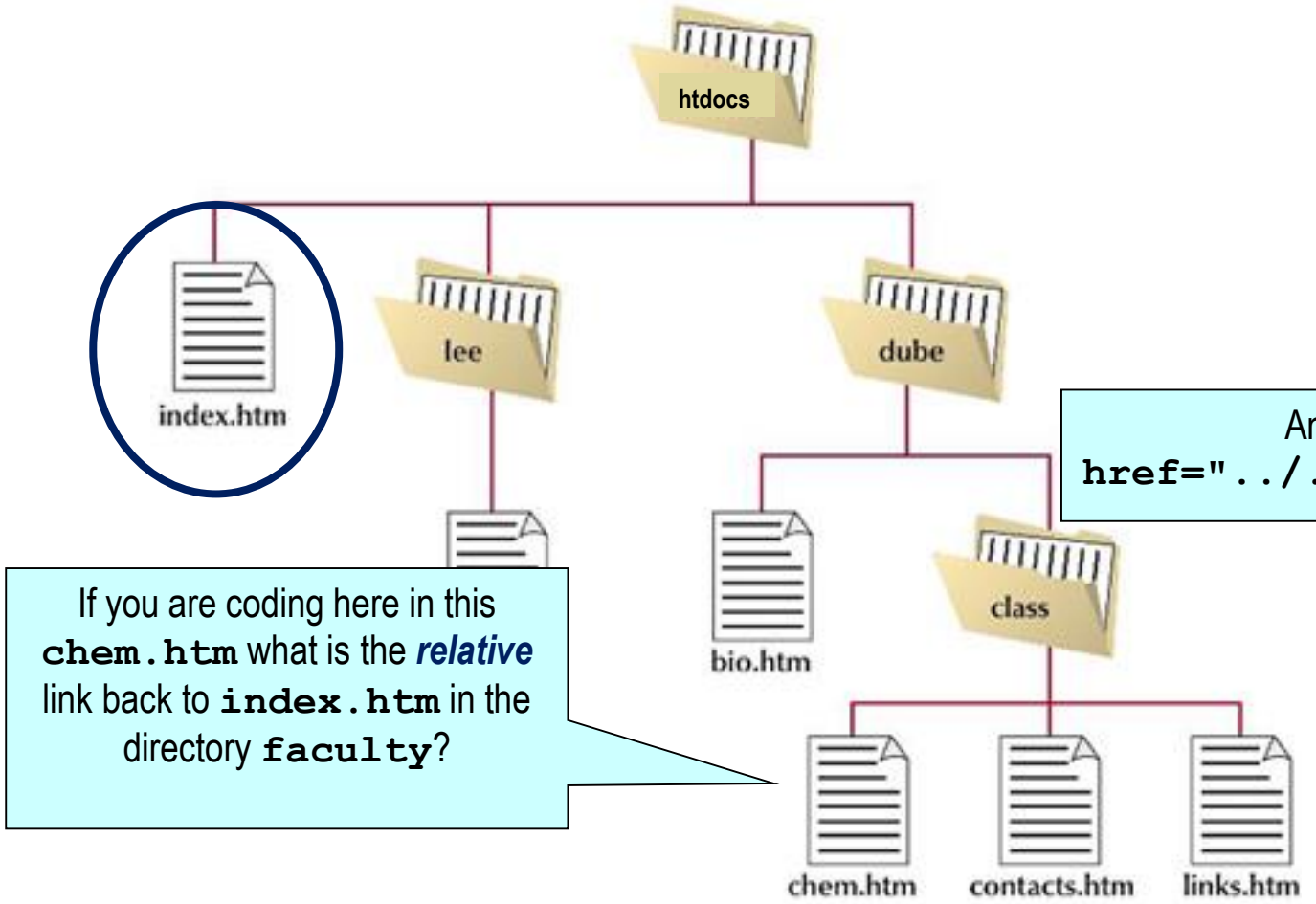
  e.g. **"http://www.swinburne.edu.au/update.htm"**

  Do not use local directory **"file://c:../update/htm"**

Paths – Relative



Paths – Relative

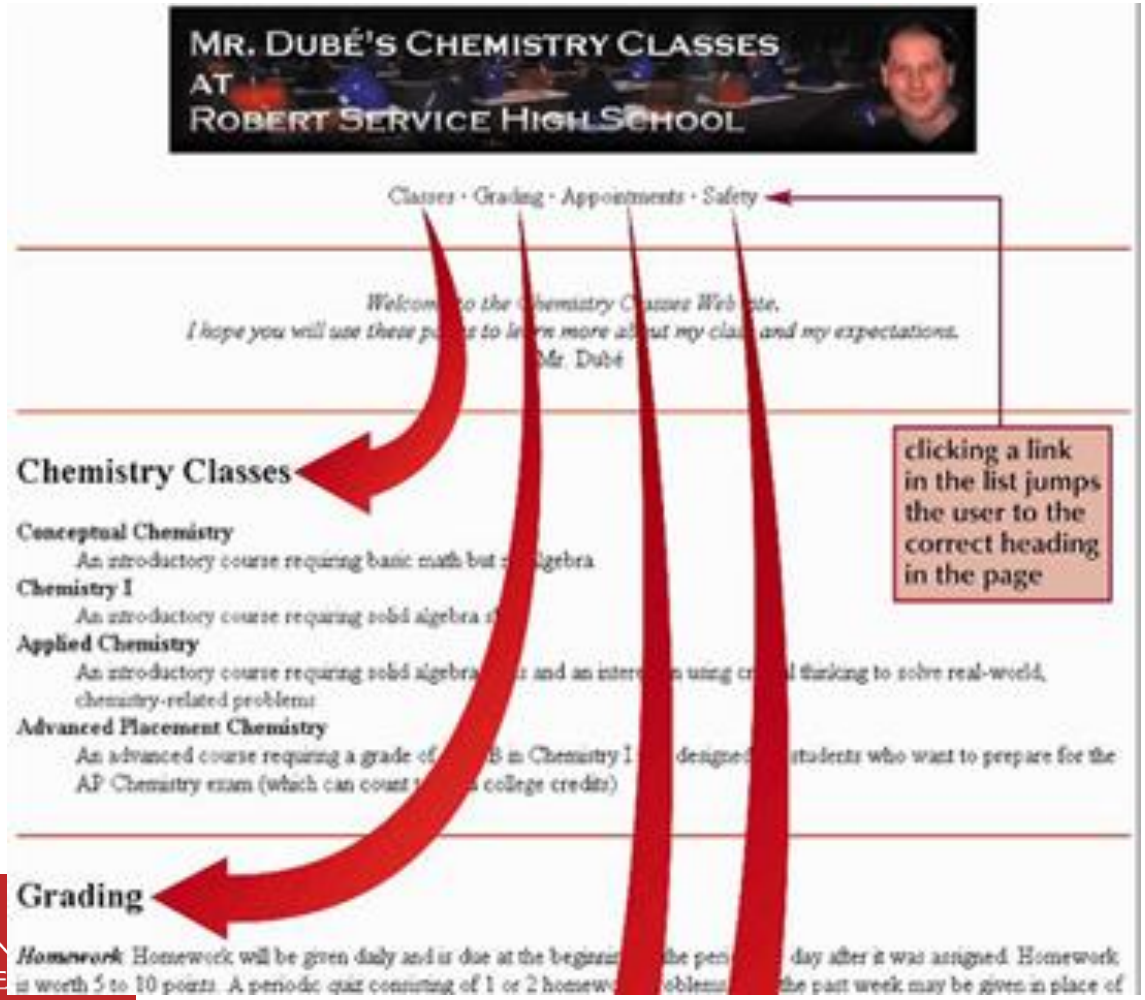


If you are coding here in this **`chem.htm`** what is the ***relative*** link back to **`index.htm`** in the directory **`faculty`**?

Answer
`href="../../index.htm"`

Linking to Sections

- How do we link to sections within a page?



Anchor element

- How to link to a resource location ***within*** a web page
- Identify a resource location with an **id** attribute in the page

```
<h1 id="chapter01">Chapter One</h1>
```

- Refer to the resource location within an anchor using an # symbol

```
<a href="home.htm#chapter01">Chapter 1</a>
```

- By default, links in all browsers are displayed as follows :
  - An unvisited link is underlined and blue
  - A visited link is underlined and purple
  - An active link is underlined and red

Anchor (continued)

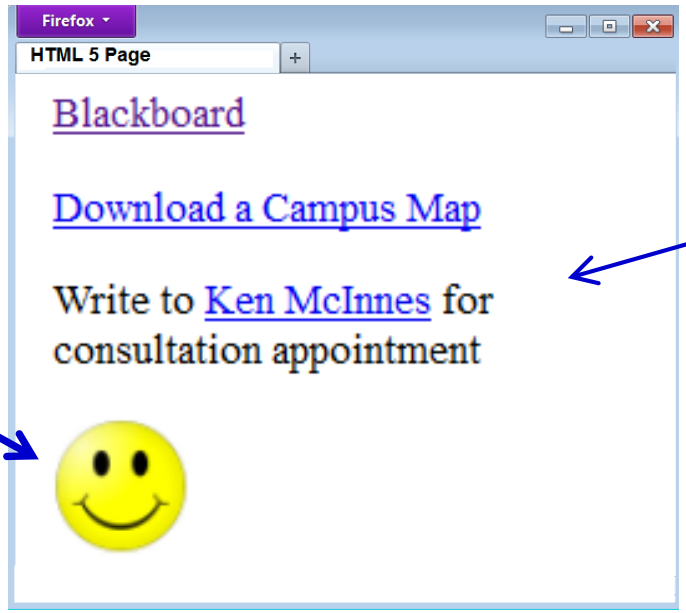
```
<p><a href="http://ilearn.swin.edu.au"> Blackboard</a></p>

<p><a href="http://www.swinburne.edu.au/campuses/hawthorn/
documents/hawthorn.pdf" >Download a Campus Map</a></p>

<p>Write to <a href="mailto:kmcinnes@swin.edu.au" >Ken McInnes</a>
for consultation appointment</p>

<p><a href="http://ilearn.swin.edu.au">
        <img src="smiley.png" alt="Smiley face" /></a></p>
```

Image as hyperlink



Paragraph tags are used in the example to display a multiple line example. List elements could also have been used.

HTML: Common Link Protocols

Protocol	Purpose
`file`	Access document stored on a local computer system
`ftp`	Access files on a server using the file transfer protocol
`http`	Web document transfer protocol
`mailto`	Use local user configuration to create new email message
`news`	Usenet newsgroup service
`wais`	Wide Area Information Server database
`tel`	User local user configuration to activate the phone and dial the number RFC3996
`sms`	User local user configuration to create / send sms RFC5724

Include full international prefix.

```
<a hef="tel:+61-3-9214-8000"> +61-3-9214-8000</a>
```

```
<a href="sms:+61-3-9214-8000?body=Remember%20the%20milk">Send reminder!</a>
```

Other <a> attributes

- **Apart from href** <a> has number of other useful attributes

```
<a href="…" title="…" target="…" … >…</a>
```

  - **title** a core attribute, shows information about the link as a "tool tip".
  - **target** specifies **where to display** the information retrieved from the URL.

Value	Description
_blank	Opens the linked document in a new window or tab
_self	Opens the linked document in the same frame as it was clicked (this is default)
_parent	Opens the linked document in the parent frame
_top	Opens the linked document in the full body of the window
framename	Opens the linked document in a named frame

Link `title` attribute

- An example of a hyperlink `title` attribute value being displayed as a tool-tip by the browser.

```
<a href="home.htm"
   title="Return to the Chemistry Home Page">
   Home Page</a>
```

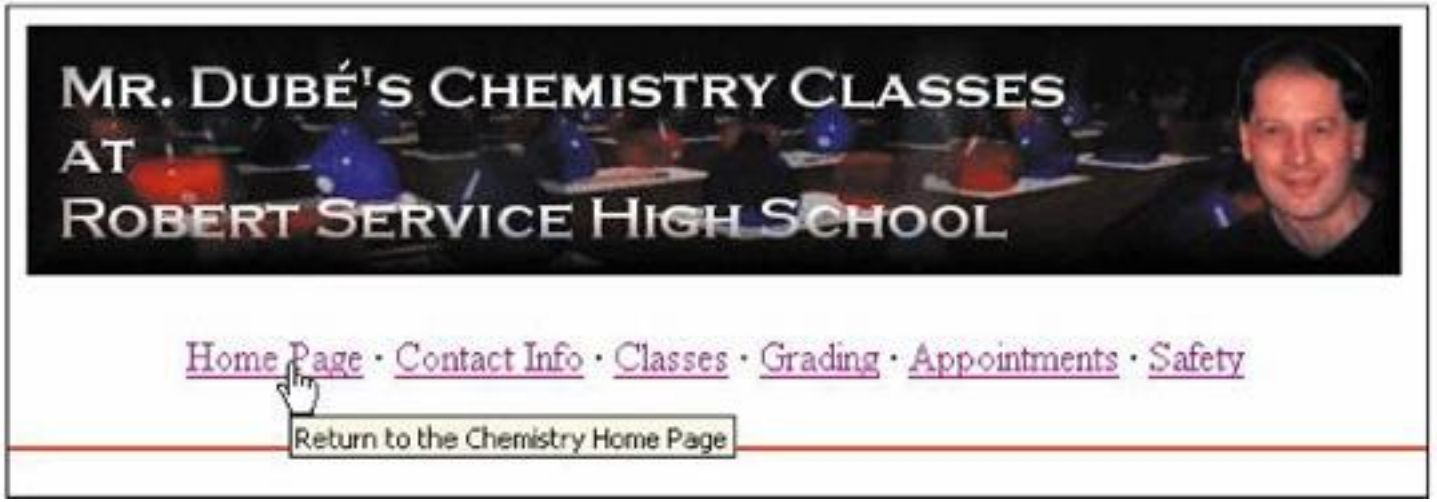


Image "Thumbnails"

- "Thumbnail" images:

  Can create a better user experience …

  - Because large image files can take time to download, it is a common practice to provide smaller "thumbnail" versions of larger images.
  - The small thumbnail images download quickly.
  - "Thumbnail" images are commonly used as hyperlinks to a larger images. The user clicks on the small image to see the larger image.

- Good technique for "gallery" or "photo album" pages.

```
…
<a href="mycat_large.jpg">
  <img src="mycat_small.jpg" height="15" width="30"
       alt="Photo of my cat" title="My cat Tommy" />
</a>
…
```

Hint: Add a `title` attribute to the anchor element to let users know how big the large.jpg file is.

HTML: Syntax References

Syntax references:

http://www.w3.org/

The W3C HTML Standards / References

http://reference.sitepoint.com/html

HTML Tutorials / References

http://www.htmlhelp.com/

HTML References

http://www.w3schools.com/

HTML Tutorials / References

See also: ***Web Links on the Blackboard***

Next week

- HTML5 Forms and Structure