This lecture

This lecture covers basic HTML, just sufficient to write a simple webpage.

A revolution

A revolution in information exchange occurred on August 6, 1991 when the world's first website (http://info.cern.ch) went online.

Although the Internet and email had been in existence for several decades, the ability to browse documents and to follow hyperlinks between documents changed everything.

Although it is only 15 years ago, it is now almost impossible to imagine the world before the World Wide Web.

Browsing

The Web works on a client/server basis where somebody wishing to make documents available (that is, to publish them) puts them onto a web server.

A client program can locate documents by means of a Uniform Resource Locator (or URL) which contains sufficient information for the program to locate the server and request that particular document.

The server then delivers the document to the client program which can then process it in any particular way it chooses.
**HTML**

HTML (HyperText Markup Language) is a *markup language* that combines a document’s text with *markup*, which is additional information about the text.

Using HTML, the author can combine the *content* of the document along with information about its logical structure or its desired presentation or both.

Today HTML is a *standard* that authors write to, and browsers implement (or not!).

---

**HTTP**

HTTP (Hypertext Transfer Protocol) is a protocol for transferring information between a client and a server.

This is the lower-level language used by clients to “ask for” pages by sending specially formatted requests, and used by servers to respond to clients.

For our purposes we need to know little about HTTP except for one important thing — HTTP is *stateless* which means that each exchange (request/deliver) is treated as a separate exchange.

---

**URLs**

At its simplest, a URL (Uniform Resource Locator) will specify a protocol, an address and a specific file location.

For example, the URL

\[http://www.w3.org/MarkUp/Guide/Advanced.html\]

identifies the document called *Advanced.html* which is located in the directory *MarkUp/Guide* on the server with address *www.w3.org* and specifies the HTTP protocol.

If you type this into a Web-browser, then it will send an HTTP request to the server at that address and request that particular document.

---

**Dynamic Pages**

Of course, the web is used for much more than just retrieving remotely stored static documents.

In particular, nowadays we expect websites to be *dynamic*. The sites produce different output depending on choices made by the user, or based on parameters obtained *from* the user — parameters such as search boxes.

Fundamentally this is achieved by the client program sending specially formatted URLs to the web server, which recognises that they are not merely file locations. The web server then runs programs that *generate* an HTML document “on-the-fly” and sends it back to the client.

Very often these programs will connect to databases to get the information needed to generate the HTML documents.
Basic HTML

A basic HTML document consists of a sequence of HTML elements which each consist of some content together with HTML tags that provide additional information about the content.

A very simple HTML element is something like:

```html
<p>This is a well-structured paragraph.</p>
```

where the content is the sentence and the tags (the `<p>` and the `</p>`) indicate that this sentence is a paragraph.

When this is displayed, the browser can use the information in the tags to choose how to display it.

(GF Royle 2006-8, N Spadaccini 2008)

Nested Elements

HTML elements can be nested so that one element completely contains another.

```html
<p>This is a <em>very</em> well-structured paragraph.</p>
```

where the `<em>` tags are used to indicate that the browser should emphasize the content.

(GF Royle 2006-8, N Spadaccini 2008)

Element Structure

In general, an HTML element consists of four items:

- A start tag.
- (Optionally) some attribute/value pairs.
- The content.
- An end tag.

In this HTML element (which is used for a cell in a table)

```html
<td align="center">Item One</td>
```

the attribute `align` is specified to have the value `center`.

Any number of attribute/value pairs can be specified along with the start tag.

(GF Royle 2006-8, N Spadaccini 2008)

The global structure

The global structure of an HTML document consists of an `<html>`..`</html>` element containing a `<head>`..`</head>` element and a `<body>`..`</body>` element.

```html
<html>
  <head>
  
  </head>
  <body>
    
  </body>
</html>
```

The `head` contains information about the page and the `body` contains the actual page.

(GF Royle 2006-8, N Spadaccini 2008)
The doctype

One more item should be included at the start of every HTML document — the DOCTYPE says which version of HTML the document is using.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

This DOCTYPE says that the document will be written in HTML 4.01 Strict.

Strict and loose

There are a number of variants of HTML 4.01 which are intended for the use of existing websites using older or deprecated HTML elements. These older versions of HTML are less well-structured in that they permit mixing structural attributes with presentation attributes such as

```
<td align="center" bgcolor="red">Item One</td>
```

whereas the modern way is to separate the presentation attributes into style sheets.

Validation

Whichever version of HTML is being used, it is important to validate the page using an online validation service such as http://validator.w3.org/.

This is a program that compares the structure of your document with the syntax of (your version of) HTML and confirms that your document is correctly structured.

This is important because most current browsers silently accept invalid HTML, even if it is quite seriously incorrect.

This causes great problems when someone writes invalid HTML that “looks ok” on one browser, but is viewed on another browser that may treat the incorrect HTML quite differently.

Head elements

The only required element inside the head is a page title:

```
<head>
<title>DB Dudes Home Page</title>
</head>
```

Normally the browser will display the title of the page as part of the title of its own window.

Other HTML elements that might be contained in the head section include `<meta>..</meta>` and `<style>..</style>`.
Body Elements

There are quite a number of different HTML body elements, but we will just consider a few of the most important ones that suffice to make a functional web page.

We will consider just

- Headings
- Paragraphs
- Links
- Images
- Tables
- Forms

Headings

To make section headings stand out, there are 6 different levels of heading available:

- `<h1>...</h1>`, `<h2>...</h2>`, `<h3>...</h3>`
  - Most browsers will render section headings, sub-headings and sub-sub-headings by making the text larger and bolder.
- `<h4>...</h4>`, `<h5>...</h5>`, `<h6>...</h6>`
  - Although it is browser-dependent, there is usually not much point in using sub-sub-sub-sub-sub-headings because they usually all look the same.

Paragraphs

Just as in normal writing, text content should be divided into paragraphs each of which is contained within a `<p> ... </p>` pair of tags.

In addition, there are some inline elements that can be used within paragraphs to identify specific portions of the paragraph.

- `<em> ... </em>`
  - To emphasize a word or phrase.
- `<strong> ... </strong>`
  - To strongly emphasize a word or phrase.
- `<code> ... </code>`
  - To identify a fragment of computer code

Links

Hypertext wouldn’t be hypertext without links which are specially rendered parts of a web-page that lead to another location when clicked.

A link to a different web page consists of some link text enclosed within `<a> ... </a>` tags with the destination of the link being specified as the `href` attribute of the tag.

```
<p>
  To search go to
  <a href="http://www.google.com">Google</a>
</p>
```

Here the word “Google” will appear specially highlighted and clicking on it will cause the browser to move to the specified location.
Images

A browser can display *images* by using the `<img/>` tag. Notice that this tag does *not* come as a *pair* in the way that we have seen with the other tags so far.

The image to be included is specified by means of the `src` attribute of the tag. To comply with the standard an image *must* have some *alternate text* supplied using the `alt` attribute.

```
<p>
  <img src="flag.png" alt="Small AUS Flag">
</p>
```

The `alt` text will be displayed if the browser cannot locate or display that image or if it is set to never display images (for example, a speech-based browser for blind users).

Tables

*Tables* are one of the more complicated HTML elements as they consist of a number of different parts.

The entire table should be contained within a `<table> ... </table>` pair of tags. Within these tags there are *table rows* which each contain a number of *cells* containing the data.

Each row in a table is contained within a `<tr> ... </tr>` pair of tags, and each data item is contained in *either* a `<th> ... </th>` (for the header row) or a `<td> ... </td>` pair of tags for the remaining rows.

Tables are frequently (mis)used for *layout purposes* rather than simply for the display of tabular data.

```
<table>
  <caption>Temperature Conversion</caption>
  <tr> <th>Celsius</th> <th>Fahrenheit</th> </tr>
  <tr> <td>0</td> <td>32</td> </tr>
  <tr> <td>10</td> <td>50</td> </tr>
  <tr> <td>20</td> <td>68</td> </tr>
  <tr> <td>30</td> <td>86</td> </tr>
  <tr> <td>40</td> <td>104</td> </tr>
</table>
```

This code produces an entirely undecorated table, simply laying out the data items in tabular form.

Where to from here?

*HTML forms* are the mechanism for processing *user input* and are particularly relevant to database connectivity, so will be covered in a later lecture.

Otherwise, this lecture has covered only enough basic HTML to write very simple and plain web pages. However there is a vast amount of material on the Web and elsewhere about writing HTML.

As you practice, you should ensure that you only write *valid* HTML 4.01 STRICT by repeatedly validating your work — this is the best way to learn how to write properly formatted and logically structured HTML.