Web design 1: Introduction to creating a website using Dreamweaver MX

Practical workbook

Aims and Learning Objectives

The aim of this course is to enable you to create a simple but well designed website to XHTML standards using Dreamweaver MX.

When you have completed these exercises, you will be able to:

- create a simple but functional website to present information about yourself, department or other interest using essential (X)HTML tags and Dreamweaver MX;

- apply fundamental good web design principles to your pages;

- transfer your (X)HTML files from your local PC to a web server using the SSH Secure File Transfer program.

Document Information

Format Conventions

The following format conventions are used in this document:

Computer input that you type is shown in a **bold Courier New** font	`<h2>Contact Information</h2>`
Hypertext links to be followed are shown in an underlined Courier New font	http://www.web-source.net/
Computer output, menu names and options, buttons, URLs are shown in a Courier New font	`Save, Go to, Refresh`
Computer keys that you must press are in a **bold Courier New** font and enclosed in angle brackets	**`<Enter>, <n>, <N>, </>`**
Menu selections are illustrated in a Courier New font using forward slash (/) to indicate a sub-menu. In the example, this would mean: select the Insert menu, then select Picture, then select From File…	`Insert/Picture/From File…`

Related documentation

Other related documents are available from the web at:

www.bristol.ac.uk/is/learning/documentation/docs-by-category.html#web

www.bristol.ac.uk/is/learning/documentation/docs-by-category.html#reg

www.bristol.ac.uk/is/learning/documentation/docs-by-category.html#net

Web design 1: Introduction to creating a website using Dreamweaver MX (April 05)
www.bristol.ac.uk/is/learning/documentation/web-t3/web-t3.doc

If you have any comments or queries about this document mail iser-docs@bristol.ac.uk.

Contents

Introduction

This course aims to equip you with the basic skills needed to create your own website. You will learn how to design and produce basic web pages using the (X)HTML language and how to integrate them into a well-organised and user-friendly website. The course will also cover useful design tips and techniques to improve your site, as well as how to put your website on-line.

Prerequisites

This document assumes that you are familiar with the use of a computer keyboard and mouse, Microsoft Windows based products and the use of a web browser such as Netscape or Internet Explorer.

What is the World Wide Web, how does it work?

The World Wide Web (WWW) is part of the **Internet**, which itself is "a network of interconnected computers", in other words the physical infrastructure used to transfer data (for example, emails, web documents etc.) between computers.

The WWW is a body of virtual information stored on **web servers**. A web server is a computer system that runs software to allow people to look at the web pages stored on it from their own PCs. The University has its own web server (even several) which is connected to the Joint Academic NETwork (JANET). From home, you have to connect (you must be registered first) to the web server of an Internet Service Provider (ISP) to access the Internet.

"Give me file x"

"Here it is"

Desktop computer - "client"

Computer on the Internet holding information - remote "web server"

Publishing information on the web

The HyperText Mark-up Language

(X)HTML (HyperText Mark-up Language) is a document layout and hyperlink specification **mark-up language** used to format text and information for the web; it is **NOT a programming language** like C++ or Java. (X)HTML consists of **mark-up elements**. The syntax of a typical element is as follows:

`<name attribute1="value" attribute2="value">`text**`</name>`**

opening tag   attribute   value   closing tag

At its most basic an (X)HTML element consists of an *opening tag* (`<name>`) and a *closing tag* (indicated by a forward slash before the tag name – `</name>`) containing text (or other elements). Tags consist of a tag *name* and sometimes one or more optional *attributes* carrying *values*, which modify the default behaviour and settings of the tag. (X)HTML elements instruct browsers (and other user agents such as screen readers) on how to render the content. The best way to understand the syntax of tags is to look at a few examples:

`<h1>``heading level 1`**`</h1>`** - a level 1 heading

`<a href="http://www.bristol.ac.uk">``University of Bristol`**`</a>`** - a link to the University of Bristol homepage

`<table width="80%" border="1" align="center"><tr><td>``Cell 1 Row 1`**`</td></tr></table>`** - a table consisting of 1 row and 1 column

A few (X)HTML elements do not contain anything, they either point to a resource (eg an image) or they insert an object (eg a line break, a line); these are called *empty elements* and they look like this:

`<name attribute1="value" />`

For example:

`<br />` - inserts a line break

`<hr size="3" width="50%" noshade="noshade" />` - inserts a horizontal rule

> **`<img src="picture.jpg" width="100" height="100" alt="picture of something" />`** - inserts an image

(X)HTML elements are the **building blocks** of the web. This means that (X)HTML is not going away. Since 1990, HTML standards as defined by the World Wide Web Consortium (W3C, see <u>http://www.w3.org</u>) have evolved considerably. Until recently, HTML 4.01 was the recommended standard, however it has been superseded by the eXtensible HyperText Mark-up Language (XHTML), which has now become the recommended standard.

Making the transition from HTML to XHTML is really easy. Except for a few differences, most of HTML is compliant with XHTML. In XHTML:

- **tags must be in lower case** whereas HTML accepts UPPER as well as lower case;

- **attribute values must be enclosed in double quotes** (for example, `<td width="120">` instead of `<td width=120>`);

- **elements must be correctly nested** (for example, properly nested tags instead of incorrectly nested tags),

- **empty elements must be closed** (for example, `<hr />` instead of `<hr>`, `<br />` instead of `<br>`),

- **all attributes must be given explicit values** (for example, `<hr noshade="noshade" />` instead of `<hr noshade>`),

(X)HTML **files** should be saved with the extension `html` rather than `htm` (eg `welcome.html`).

Basic XHTML document structure

(X)HTML syntax is not rocket science, and as such it is accessible to most people. To prove this point, let's look at a basic web page to see how it is structured:

```
1   <?xml version="1.0" encoding="iso-8859-1"?>
2   <!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

4   <head>
5    <title>Joe's website - Home</title>
6    <meta http-equiv="Content-Type" content="text/html;
     charset=iso-8859-1" />
7    <meta name="description" content="Joe Bloggs's homepage." />
8    <meta name="keywords" content="keyword 1, keyword 2" />
9   </head>

10  <body><!-- content starts here -->
11   <h1>Joe Blogg's homepage</h1>
12   <p>Welcome to my world!</p>
13   <hr size="1" />
14   <p>Maintained by <a href="mailto:jb@aol.com">Joe Blogg</a></p>
15  </body>

16  </html>
```

The rather cryptic lines 1 and 2 tell the browser what versions of XML and XHTML your pages are written in. The XML declaration (line 1) is optional, but the XHTML declaration (line 2) is compulsory for your pages to validate against the standards as set by the World Wide Web Consortium (W3C). You will see many web page that do not include a `<!DOCTYPE>` and still work, however it MUST be included at the top of each page to ensure your pages will function properly in all browsers.

The whole document is contained between the opening `<html>` tag (line 3) and the closing `</html>` tag (line 16).

An (X)HTML document has **two main parts**: the `head` and the `body`. The `head` (enclosed between the opening `<head>` and closing `</head>` tags – lines 4 to 9) contains information **about** the file, including the `title` (between the opening `<title>` and closing `</title>` tags – line 5) of the page – displayed in the bar at the top of the web browser. The information contained in the `body` (between the `<body>` and `</body>` tags – lines 10 to 15) is what is actually displayed in the browser's main window.

The line `<!-- content starts here -->` (line 10) is a **comment** and is ignored by browsers. Comments act as **signposts** to **help** you (or whoever might contribute to your site) **make sense of your code**, especially if you haven't looked at it for a while! They begin with `<!--` and end with `-->`.

The `<html> </html>`, `<head> </head>`, and `<body> </body>` tags are **the only required structural elements** of an (X)HTML document. As we have already said, the **contents** (text, images, links, etc) of your document **are enclosed between the** `<body>` **and** `</body>` **tags**. This means that other (X)HTML tags comprised between these two structural tags **will modify the way contents display** in a browser.

In our example, the `<h1> ... </h1>` tags (line 11) tell the browser to display the text `Joe Blogg's Homepage` as a level 1 heading (there are 6 levels) - that is, bold, larger font size with spacing before and after.

The `<p>` and `</p>` tags (lines 12 and 14) form paragraphs; they insert a blank line before and after the paragraph, thus setting it apart from other elements.

Finally, `<a href="mailto:joe@topman.net">Joe Blogg</a>` (line 14) creates a hyperlink to an email address.

Here is how the code above would be rendered in a browser:

Everything between the `<body>` … `</body>` tags displays in the main window
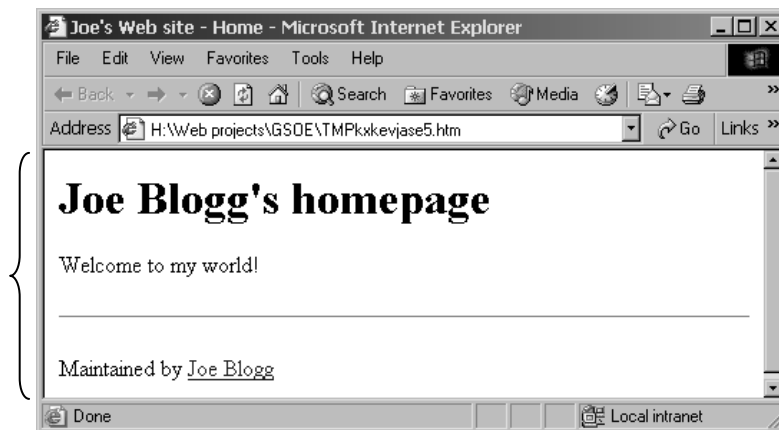
Task 1 Customising Dreamweaver

Objectives	To customise Dreamweaver MX to produce XHTML compliant documents.
Method	You will use the `Edit/Preferences` menu in Dreamweaver.
Comments	It is easy to customise Dreamweaver to reflect your own preferences. Since we recommend writing University pages in XHTML, the first thing we need to do is to ensure that it will code our pages to XHTML 1.0 standards.

Opening Dreamweaver

1.1 Go to `Start/Programs/Macromedia/Macromedia Dreamweaver MX`.
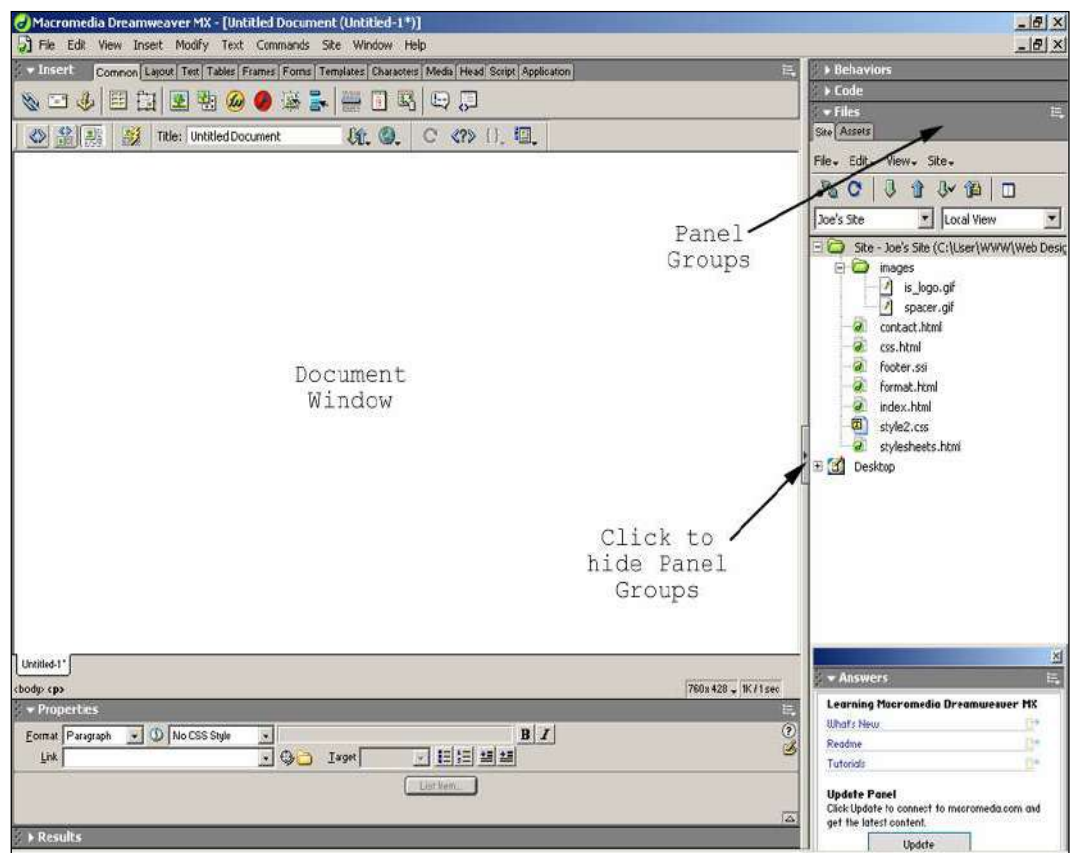
Figure 1 - Dreamweaver MX default interface on opening

1.2 Hide the panel groups on the right by clicking on the hide panel groups arrow, and maximise the `Document Window` by clicking on the Maximise icon ▢ in the top right corner.

Hide the `Property Inspector` by clicking the down arrow to the left of `Properties` in the bottom left corner.

Customising Dreamweaver

1.3 From the `Edit` menu, select `Preferences` (last option).

1.4 In the `Category` column, select `Code Hints` and, in `Options` at the top, uncheck the `Enable Auto Tag Completion` box.

Figure 2 - the code hints category in preferences

1.5 To set `Dreamweaver` to produce XHTML compliant code:

➢ In the `Category` column, select `Code Format` and make sure it is set as follows:
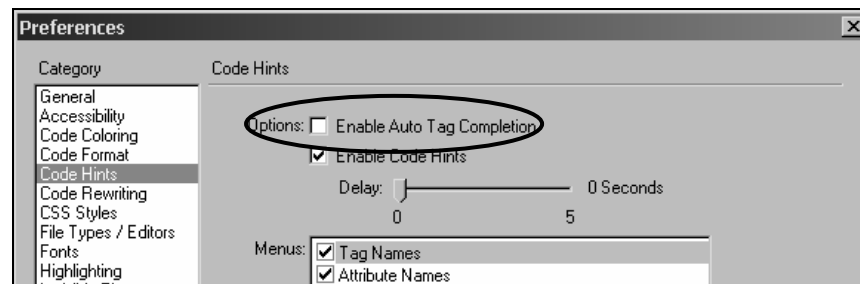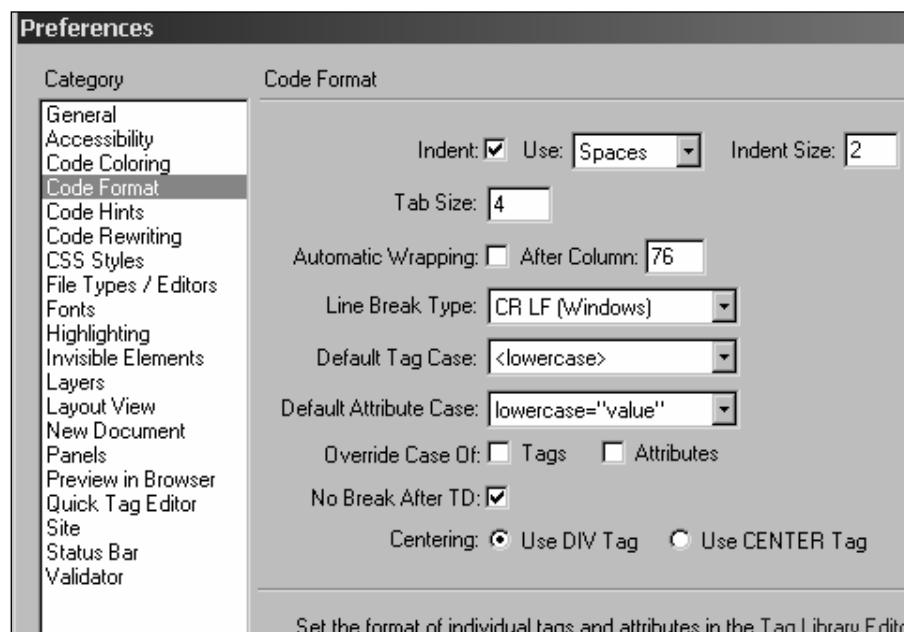

Figure 3 - code format options

➢ Now select `New Document` under `Category`. Make sure the `Default Document Type` is set to `HTML` and tick the `Make Document XHTML Compliant` box.

➢ Finally, select `Validator` under `Category` and choose `XHTML 1.0 Transitional`.

➢ Click on `OK` to close the `Preferences` dialogue box.

1.6 To activate the above changes, you need to close the current document and open a new one.

➢ Go to `File/Close` to close the current document (this will close the `Document window`) – alternatively you can click on the bottom close icon ⊠ in the top right corner of the `Document window` (be careful not to click the one right at the top, which would result in closing the programme itself).

➢ Go to `File/New` and click the `Create` button – by default the `New Document` dialogue box will be set to `Basic page: HTML`.

Also check that the `Make document XHTML compliant` option is ticked, if it's not then select it.

> ➢ Next in the `View` menu, select `Code` – alternatively, you can click on the `Show Code View` icon ⟨◇⟩ in the top left corner, just below the `File` menu.

You now see the basic 'skeleton' of a new, XHTML compliant web page as created by `Dreamweaver` (see Figure 4).

Notice the changes in the code, particularly at the top of the document:
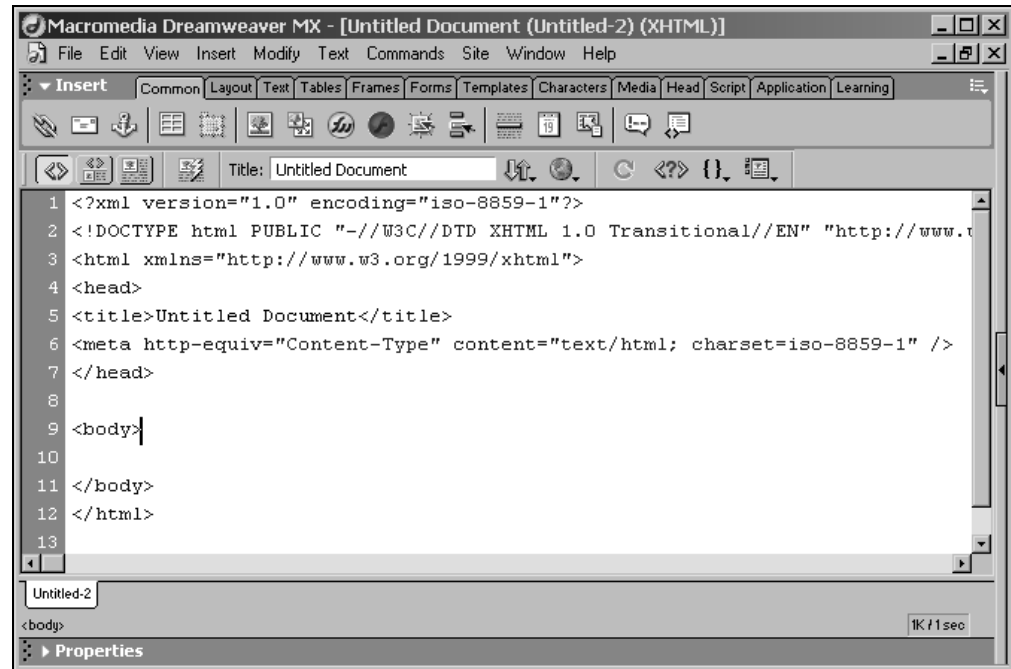
Figure 4 - code view of a new document in Dreamweaver MX

Note   If you have inherited a site created by someone else a while ago, it is likely that it has been written in HTML rather than XHTML. If this is the case, you do not need to change the default preferences as explained in task 1.5.

1.7   Finally, you are going to set Dreamweaver to wrap text and to highlight invalid code; this will help you correct your errors.

> ➢ Go to `View/Code View Options` and make sure all the options have a tick in front of them.

Task 2 Creating a basic web page

Objectives To create a basic web page document.

Method You will create a basic (X)HTML document using the Code View. You will then add some text to your new document and preview the result in a browser.

Adding content to your first page

2.1 First you are going to add some 'meat' to your new home page[1]:

Between the `<body>` and `</body>` tags, type in the following text:

> **Joe Bloggs's homepage!**
> **Welcome!**
> **This site has been hand crafted by myself to show that creating a web page is not rocket science!**
> **On it, you will find information about me, a description of my favourite hobbies and contact information.**
> **Enjoy your visit!**

Note Replace `Joe Bloggs` with your own name.

2.2 Secondly, you are going to give it a title:
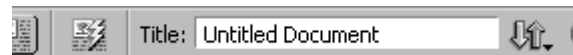
Figure 5 - the title text box in the toolbar

In the `Title` text box (Figure 5) replace `Untitled Document` with ***YourName'*s** **Homepage – Home**

Click anywhere in the `Document Window` and notice what happens. The text that you just typed appears between the `<title>` and `</title>` tags in the `<head>` element of your document.

Alternatively, you could have typed the title of your document in the `Document Window` itself, between the `<title>` and `</title>` tags.

Note Remember that this title is not part of the main content of your page, it is part of the **metadata** (information **about** the page) contained in the `<head>` element. The title will appear in the browser window title bar (right at the top of the screen). It is also the default name for a link to the document (for example when users bookmark a site, the name of that bookmark will be the title text).

Saving your first page

2.3 Save your new file on your local computer. Select `File/Save As`. In the `Save As` window do the following:

> ➢ Find the correct local directory (`C:\User\WWW\Web design 1`) in the `Save in` box.
>
> ➢ In the `File name` text box type **index.html**.

> **TIP**
> It is good practice to give the home page of your site the default name `index.html`. Thus you won't need to include it in your web address.

---

[1] See glossary

> ➢ Click the Save button.

You could save your file on your local computer in any directory that you created.

Viewing your page in a browser

2.4 Dreamweaver has a preview in browser facility that you can use to preview your pages.

> ➢ Click on the Preview/Debug in Browser icon 🌐▾ in the Toolbar and select Preview in IEXPLORE.EXE.

Your document should be displayed as shown in Figure 6.

Figure 6 - document displayed as a single paragraph

Notice how the text appears in a single paragraph despite the line breaks in the source (HTML) document.

Notice also the name given to the file by Dreamweaver in the Address box, something like TMPhjk7bhuzpv.htm. This is because Dreamweaver created a temporary version of your page to preview it in a browser.

> ➢ Close the browser.

Note If you wanted to see the actual file itself (or if you use something other than DW to edit your web pages), you would need to open it from a browser. To do this:

> ➢ Open Internet Explorer (or Mozilla if your prefer).

> ➢ Then go to File/Open (File/Open File in Mozilla).

> ➢ Click on the Browse button and navigate to your file C:\User\WWW\Web design 1\index.html in this case) and open it.

Task 3 Structuring content

Objectives To add some content-structuring elements such as paragraphs, line breaks, headings and horizontal rules.

Method You will explore the changes to your page as a result of using the text formatting features listed above.

Comments You will be using the following (X)HTML tags for this task:
Paragraph: `<p>…</p>`
Line break: `<br />`
Headings: `<h1>…</h1>` … `<h6>…</h6>`
Horizontal rule: `<hr />`

Line-break - **
**

3.1 Return to Dreamweaver and type a `<br />` tag at the end of each sentence as follows:

Joe Bloggs's Homepage**
**

Welcome!**
**

This site has been hand crafted by myself to show that HTML is not rocket science!**
**

On it, you will find information about me, a brief description of my favourite hobby and contact information.**
**

Enjoy your visit!

Note Including a space and a forward slash before closing an 'empty tag' (for example, `<br />`) will make your pages compliant both with HTML 4.01 and XHTML 1.0. As a result your pages will work well with older browsers as well as the new generation of browsers.

➢ Save (**<Ctrl + S>**) and preview in a web browser (as described in task 2.4).

➢ Close the browser.

Paragraph - **<p></p>**

3.2 Return to Dreamweaver, remove the `<br />` tags and enclose each of the sentences between the `<p>` and `</p>` tags to form separate paragraphs as follows:

<p>Joe Blogg's Homepage**</p>**

<p>Welcome!**</p>**

<p>This site has been hand crafted by myself to show that HTML is not rocket science!**</p>**

<p>On it, you will find information about me, a brief description of my favourite hobby and contact information.**</p>**

<p>Enjoy your visit!**</p>**

➢ Save and preview in your web browser.
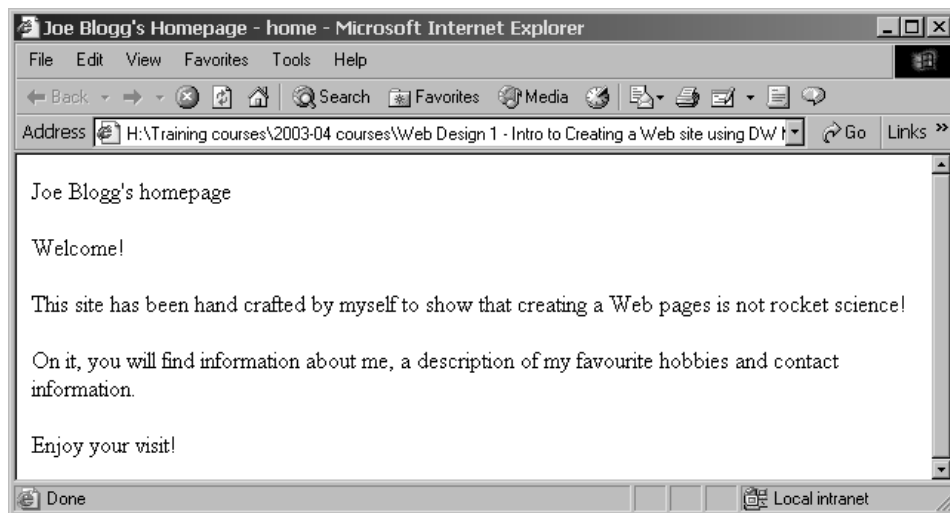
Figure 7 - five separate paragraphs

Note As you can see, text placed between the `<p>` and `</p>` tags forms a separate paragraph, with a line space added before and after. whereas a `<br />` tag simply breaks the text flow which continues on the next line.

➢ Close the browser.

Headings - <h1></h1>

3.3 Let's now add some **headings**.

➢ Make the first paragraph of your document a level 1 heading instead by replacing the `<p></p>` tags with `<h1></h1>` as follows:

➢ **<h1>**Joe Blogg's Homepage**</h1>**

➢ In the same way, make the second paragraph a level 2 heading.

➢ Save and view in your web browser.
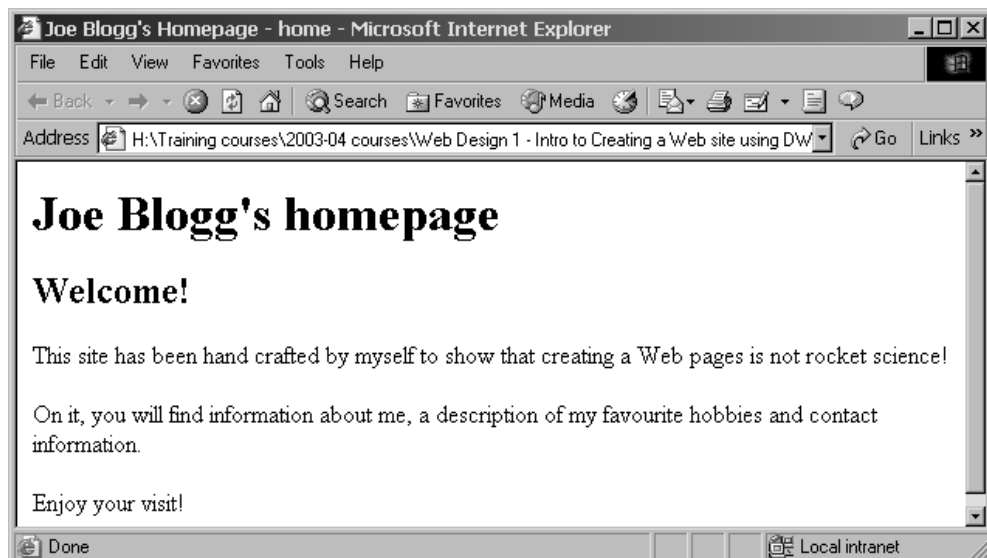
Figure 8 - level 1 and 2 headings

Note There are six levels of headings, `<h1>` through to `<h6>`, `<h1>` rendering the larger text size and `<h6>` the smallest. As with paragraph tags, they include a line space before and after. They are used to delineate new sections and subsections of a page.

3.4   Try out other sizes to see for yourself by replacing the `<h1>` and `<h2>` tags with other sizes (ie. `<h3>` down to `<h6>`). Restore the `<h1>` and `<h2>` tags as in task 3.7 once you've finished experimenting.

Horizontal rule - <hr />

3.5   Finally, to make your page look even better we'll add a couple of horizontal rules. Horizontal rules are useful to visually separate sections of your document.

➢   Add a horizontal rule below the first line of your document, as follows:

```
<h1>Joe Blogg's Homepage</h1>
<hr />
```

➢   Add another `<hr />` tag to display a horizontal rule at the bottom of your page when viewed in a browser (as in Figure 9).
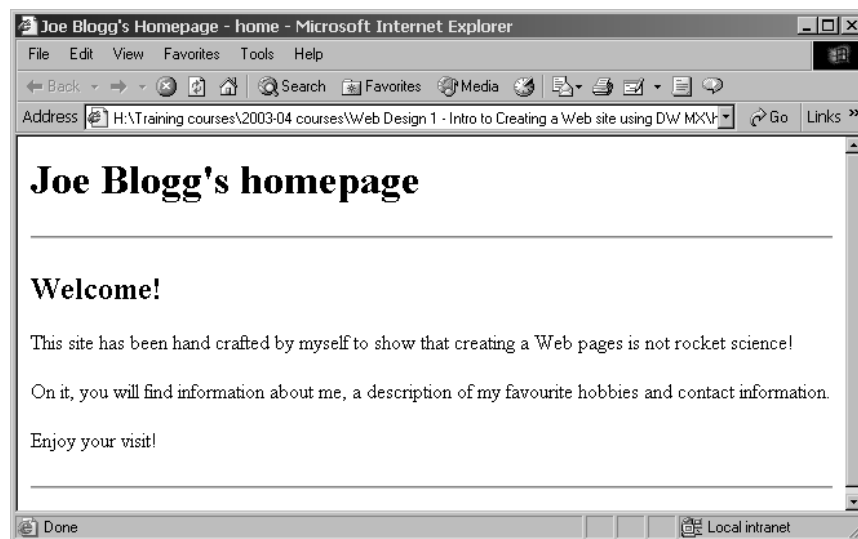
Figure 9 - horizontal rules

➢   Save and view in your browser.

3.6   Horizontal rules come with default settings - 100% width, 2 pixels high and engraved (3D) effect.

These default settings can be changed by using attributes. Attributes for the `<hr />` tag are:

- `Align` - tells the browser where to place the horizontal rule, it takes the values `left`, `center`, (default) or `right`.

- `Size` - specifies the thickness of the rule. It takes absolute numerical values corresponding to the number of pixels[2]. The default value is `2`.

- `Width` - specifies the length of the rule. It takes exact (expressed in pixels) or relative (expressed as a percentage) numerical values. The default value is `100%`.

- `noshade` - by default, a horizontal rule is shown with an engraved effect. You can add the `noshade` attribute if you prefer a flat, 2D rule. Its unique value is `noshade`.

For example:

```
<hr width="50%" align="center" noshade="noshade" />
```

---

[2] See glossary

Try this for yourself by adding and modifying attributes in the two horizontal rules in your page. Save and preview in a browser to see the changes.

Note You can have as many or as few attributes as you want in a tag. Also, the order in which you insert them is irrelevant.

Creating another page

3.7 You are going to create a second page similar to the first one:

- ➢ Go to `File/New` and in the `New Document` dialogue box click on the `Create` button.

  `Basic Page` and `HTML` are the default options and the `Make Document XHTML Compliant` option should be selected.

- ➢ Select the `Code` view (if it is not already selected (go to `View/Code`).

- ➢ In the `<head>`, make the content of the `<title>` tag **Joe Blogg's Homepage – about me**,

- ➢ In the `<body>` type **Joe Bloggs's homepage** and make it a level 1 heading.

- ➢ Insert a horizontal rule below.

- ➢ On the next line, type **About me** and make it a level 2 heading.

- ➢ Finally insert another horizontal rule below.

- ➢ Save it as **aboutme.html** in `C:\User\www\Web design 1`.

Note Notice the tabs in the bottom left of the Document Window. Use them to navigate between the documents that are currently open.

Task 4   Adding formatted lists

Objectives To insert some unordered and ordered lists in your page.

Method You will explore the changes to your page as a result of formatting text using the lists elements.

Comments You will be using the following (X)HTML tags for this task:
Unordered (bullet) list: `<ul><li>…</li></ul>`
Ordered (numbered) list: `<ol><li>…</li></ol>`

Ordered list - …

4.1 An ordered list is simply a numbered list. It consists of the `<ol>…</ol>` tags to create a new list and `<li>…</li>` tags to insert new list items.

Example:

```
<h2>My three favourite websites:</h2>
<ol>
  <li>BBC</li>
  <li>web Pages That Suck</li>
  <li>A List Apart</li>
</ol>
```
will display like:

Figure 10 - an ordered list

> ➢ Add an ordered list to your `aboutme.html` page.

> ➢ Save and preview in a browser.

4.2 In the same way that you can change the default settings of a horizontal rule, you can change the symbol type of an ordered list by using the `type` attribute in the `<ol>` tag and giving it one of the following values:

Attribute	Values
`type`	1 (default), `A, a, I or i`

For example:

```
<ol type="A">
```

> ➢ Try these out.

Unordered lists - …

4.3   An unordered list is simply a bullet list. We still use the … tags to insert new list items but this time we enclose them between … tags.

Example:

```
<h2>My three favourite websites:</h2>
<ul>
  <li>BBC</li>
  <li>web Pages That Suck</li>
  <li>A List Apart</li>
</ul>
```

will display like:

Figure 11 - an unordered list

➤   Replace the ordered list you created in tasks 4.1 and 4.2 with an unordered list.

➤   Save and preview in a browser.

4.4   You can also use the type attribute in the <ul> tag to change the default bullet type of an unordered list by giving it one of the three following values:

Attribute	Values
type	disc (default), square or circle

For example:

```
<ul type="square">
```

➤   Try these out.

Note   Notice how both <ul> and <ol> tags indent the list items and also add a line space before and after the lists.

Task 5 Changing text appearance

Objectives   To apply some style elements to text.

Method   You will explore the changes to your page as a result of using the (X)HTML tags introduced in this section.

Comments   You will be using the following HTML tags for this task:
Bold: `<strong>…</strong>`
Italics: `<em>…</em>`
You will also use a simple style sheet to change the font type and colour of text.

Making text bold or italic - …, …

5.1   One way to make text stand out is to format it in **bold face**.

In your `index.html` file, choose a word or phrase that you want to make bold and enclose it between the `<strong>` and `</strong>` tags.

Example:

> `<strong>is not</strong>`.

➢   Make 2 words/phrases in your page bold.

➢   Save and preview.

5.2   Another way to make text stand out is to format it in *italics*.

Choose a word or phrase that you want to make italics and enclose it between the `<em>` and `</em>` tags.

Example:

> `<em>rocket science</em>`.

➢   Make 2 words/phrases in your page italic.

➢   Save and preview.

Note   The `<strong>` and `<em>` elements are *content-based text styles* (ie. they attach meaning to text) for marking text as "strong" or "emphasised". They can be interpreted by non-visual browsers (for example screen readers as used by blind people), whereas the more common *physical styles* … (for bold) and <i>…</i> (for italic) tags are only understood by visual browsers. As it happens, most visual browsers are programmed to display `<strong>` in bold and `<em>` in italics.

Changing text colour and font type

5.3   We are now going to apply some styles to change the font colour and type of your document text. To do this you are going to link your document to an external Cascading Style Sheet (CSS) which has already been created for you.

➢Add the following line in the `<head>` of your document, just above the closing `</head>` tag:

> **`<link href="style.css" rel="stylesheet"`**
>   **`type="text/css" />`**.

This links to an external style sheet called `style.css` and stored in the same folder as `index.html`.

➢   Save and preview. See the magic?

➢ Apply the style sheet to your aboutme.html page.

Note While XHTML gives your web pages their basic structure, CSS define the way they look. Both have become recommended standards and together will go a long way to making your site accessible giving it a consistent look and feel.

5.4 We are now going to look at `style.css` and make a few changes to it.

➢ Go to `File/Open` and open the file `style.css` in `C:\User\WWW\Web design 1`.

You should see the following lines:

```
body { font-family: Arial, Helvetica, sans-serif;
       background-color: #FFFCDF; }
h1  { font-family: Haettenschweiler, Arial, Helvetica,
       sans-serif;
       color: #555339;
       background-color: #DFDCC3;
       padding: 4px; }
h2  { font-family: Verdana, Arial, Helvetica, sans-serif;
       color: #555239; }
h3  { font-family: Arial, Helvetica, sans-serif;
       color: #AAA471;
       background-color: #FFF9BF; }
p   { font-family: Arial, Helvetica, sans-serif; }
li  { font-family: Arial, Helvetica, sans-serif; }
```

You probably recognise some familiar HTML tags (`body`, `h1`, `h2`, etc) at the beginning of each line; they are known as **selectors** in CSS and are the link between the HTML document and the style. What follows, enclosed between two curly brackets (`{}`), are a list of **declarations** consisting of two parts: a **property** (eg `color`, `font-family`, etc.) and one or more **values** (eg `#FFFFCC`, `Arial`, etc).

Since it is not the purpose of this course to teach CSS, all you need to know is that **selectors** specify what HTML elements are to be affected by one or more **declaration**. However you are going to modify a few **values** to see how CSS works.

➢ Try to change the values attached to the `color and background-color` properties in one or more selectors. To help you choose a colour combination that works, open the following page in a browser:
www.wellstyled.com/tools/colorscheme2/index-en.html
(See Appendix B for more information on colours).

➢ Try to change the value attached to the `font-family` property (try `Courier New, Courier, monospace`; `"Comic Sans MS"`; `Tahoma` or `Verdana` for example). Note that you can specify several `font-family` values provided you separate them with a comma.

Note CSS specifies five generic font families that can be used instead of or in combination with individual font categories. They are: `serif`, `sans-serif`, `monospace`, `cursive` and `fantasy`.

➢ Save your CCS file after each change and preview in a browser to see the changes.

Note There is a `<font>` tag in HTML but it is deprecated, in other words it is not recommended in HTML 4, and it has been removed from XHTML. Even if a lot of people are using it, you should avoid using it and use Cascading Style Sheets (CSS) instead. CSS are covered in more detail in other courses.

Task 6 Creating hypertext links

Objectives   To create hypertext links between and within your pages, to other sites, and to an email address.

Method   You will create links to external web pages, between and within your pages. Hypertext links are what makes the web what it is: an intricate worldwide maze of interconnected documents. Their main use is to link pages within a website (as such they are an essential part of a site's navigational structure) and to link to other websites anywhere in the world. They are also used to link to an email address and to bookmarks[3] within a page.

Comments   You will use the `<a>`...`</a>` (anchor[4]) tags together with the `href` and `name` attributes.

The basic syntax of links is:

```
<a href="link_reference">link name</a>
```

where `link_reference` is the URL (address) of the file to link to and `link name` is the text of the name of the link as it will be displayed on the page (by default it is blue and underlined).

Linking to an external web page

6.1   Linking to another website is the simplest type of link. The (X)HTML syntax is as follows:

```
<a href="http://www.bbc.co.uk">BBC</a>
```

This links to an absolute URL[5] (that is, the full path `http://www.bbc.co.uk` is used).

➢   Use the list you created earlier in `aboutme.html` and make the list items links to the corresponding websites:

Name	Absolute URL
BBC	http://www.bbc.co.uk
Web Pages That Suck	http://www.webpagesthatsuck.com
A List Apart	http://www.alistapart.com

NB: Feel free to use your own favourite sites!

➢   Save, preview and try your links.

Linking to another page in your site

6.2   Linking to an internal page is very similar to linking to an external site. The only difference is that you use the relative URL[6] of the page you want to link to:

```
<a href="aboutme.html">About me</a>
```

In this example, the file we are linking to (`aboutme.html`) is in the same folder as the file we are writing to.

---

[3] See glossary

[4] See glossary

[5] See glossary

[6] See glossary

➢ Open your `index.html` page and in the second paragraph create the following links:

Text	Relative URL
about me	aboutme.html
favourite hobbies	hobbies.html
contact information	contact.html

➢ Save, preview and try your links – notice that at the moment only your link to `aboutme.html` works, as the other two files do not exist yet.

6.3 At the moment, the only way you can come back to your homepage (`index.html`) is by clicking the `Back` button on the browser.

➢ Create a link back from your `aboutme.html` page to your `index.html` page.

➢ Save, preview and try the links; you should now be able to navigate between your two pages without relying on the browser's `Back` button.

Linking to an 'anchor' within a page

6.4 Linking to a specific anchor within a page is particularly useful when you have long pages that don't display on a single screen. Forcing your users to scroll up and down the page is an option but you can make life easier for them by providing links that will enable them to jump to specific sections in your page.

There are two stages when creating this type of link:

1. creating an anchor, that is naming the specific location on your page where you want users to jump to,

2. creating a link to that anchor.

First of all you need to add some content to your `aboutme.html` page, in order to make the page long enough to see the effect properly:

➢ Open the following page in a browser: www.bristol.ac.uk/is/learning/websupport/resources/.

➢ Follow the link:

Web design 1: Introduction to creating a Website using Dreamweaver MX.

➢ Scroll down to the `Sample content for 'About me' page` section.

➢ Select and copy the sample code and text.

➢ Paste it between the `<body>` and `</body>` tags of your `aboutme.html` page after deleting the current content.

➢ Replace `Joe Bloggs` with your own name in the `<h1>` tag if you want.

➢ Save your file and preview it to see the changes.

You are going to link the items of the table of contents (numbered list) at the top of the page to the relevant section further down the page.

Note You will be able to change/edit the content of this page later on.

6.5 First you are going to name the **anchors** where you want to link to, in this case, all the level 3 headings, by using the `id` attribute in the `<h3>` tags. The `id` attribute defines a unique name (in other words, no two elements can be named with the same `id`).

➢ Modify the first `<h3>` section head as follows:

```
    <h3 id="bio_info">Biographical information:</h3>.
```

➢ In the same way, give a unique id to the other three <h3> section heads using the names in the table below:

Section heading	id
My three favourite books	fav_books
My three favourite films	fav_films
My three favourite websites	fav_websites

Note While the id="name" attribute may be added to most (X)HTML structural elements (eg h1, h2, p, etc), they are particularly useful with the <div>…</div> and … elements. This is covered in Web Design 3.

6.6 Now that you have defined your anchors, you can create links to them. The syntax for linking to an anchor is similar to other links, except that (X)HTML uses the # sign to tell browsers to look for an anchor rather than a file:

```
    <a href="#anchor_name">Link name</a>.
```

➢ Create a link from the first list item in the table of content to the corresponding section as follows:

```
<li><a href="#bio_info">Biographical
information</a></li>
```

➢ Create the links from the remaining list items in the table of content to the corresponding section – be careful to use the same anchor name that you used to name the ids in task 6.5.

➢ Save, preview and try your links (you may need to decrease the size of the browser to see the full effect).

TIP
Similarly, you can easily create links that will take your visitors back to the top of the page. All you need to do is to add the following line where you want a link back up to the top of your page to appear.

1. Create an anchor at the top of your pages by adding the following line immediately below the <body> tag:

```
<a name="top"></a>
```

2. Then create a link to this anchor by adding the following line wherever you want a "back to top" link (for example, at the end of each section):

```
<a href="#top">Top</a>
```

Linking to an e-mail address

Another useful type of link is the mailto:*email@address* which causes the default e-mail client application of users to open and makes it possible for users to send emails to the address specified.

6.7 You will add such a link at the bottom of your two pages. First, go to your index.html file and add the following just above the closing </body> tag:

```
Maintained by:
<a href="mailto:your.name@bristol.ac.uk">
your.name@Bristol.ac.uk</a>
<br />
Last updated: <em>30th September 2003</em>
```

6.8 Save and view in your browser.

6.9 Finally, **select**, **copy** and **paste** this block of code into your `aboutme.html` file.

Note	You have created a simple signature giving valuable information to users of your site: who is in charge of this site? How up-to-date is the information it contains?

Review task - creating a simple navigation bar

6.10 Now that you know how to create links you are going to create a simple navigation bar that you can re-use in all your pages.

Write the code to insert a navigation bar at the bottom of your `index.html` page as illustrated in figure 12 below:

Figure 12 - a simple navigation bar

Tips:

Make the links as follows:

Text	**Relative URL**
UoB home	http://www.bristol.ac.uk
Home	index.html
About me	aboutme.html
Hobbies	hobbies.html
Contact	contact.html

➢ To obtain the vertical bar (|) that separates the links, press **<Shift+\>** the backward slash key, to the left of the Z key on the keyboard.

➢ Frame your navigation bar between two horizontal rules.

➢ Copy and paste the navigation bar into your other page(s) (`aboutme.html`).

The correct code is available at:

www.bristol.ac.uk/is/learning/websupport/resources/
and follow the link to Web design 1: Introduction to creating a
website using Dreamweaver MX

Task 7   Using images

Objectives   To add images to your web pages.

Method   You will use the `<img src="image_filename" />` syntax to include an image.

Comments   The `src` attribute of the `<img>` tag specifies the source (that is, name and location) of an image. The image can either be located on a remote server or on your computer's hard disk. You will normally use `gif` (used for simple illustrations, logos, cartoons etc.) or `jpeg` (used for photographs and complex illustrations) image formats in your (X)HTML documents.

Creating an images folder

7.1   First you are going to create a new folder in your local directory called `images`:

➢   Go to `Start/Programs/Accessories/Windows Explorer`. Find your local directory (`C:\USER\WWW\Web design 1`).

➢   Then go to `File/New/Folder` and type in **images**.

Note   It is good practice to keep your images in a separate folder. This will make it easier to maintain your site as it grows.

Inserting an image on a page

7.2   There are two ways to obtain images: either you create an image yourself (based on a scanned photograph or from scratch) using a graphics package (for example, `Adobe Illustrator`, `PhotoShop` etc), or you use an existing image already optimised for the web (that is, `gif` or `jpeg` format). Since it is not within the scope of the present course to teach you how to create images, we'll go for the second option.

➢   Re-open the page `www.bristol.ac.uk/is/selfhelp/websupport/resources` in a browser (you may still have it opened on your desktop).

➢   Click on Web Design 1: Introduction to creating a Website using Dreamweaver MX

➢   Download the University logo (`uob-logo-tr.gif`) into the `images` folder you have just created. Repeat for the image of the watch (`watch.jpg`).

7.3   The basic syntax for inserting an image is:

```
<img src="image_file_name" />
```

The image element has a number of attributes, three of which you should always include: the width (`width`), the height (`height`) and alternative text (`alt`):

Attribute	Values
`width`	x − eg `width="150"`
`height`	y − eg `height="100"`
`alt`	Brief and meaningful description of the image

➢   Go back to your `index.html` file and insert the following code just below the `<body>` tag:

`<img src="images/uob-logo-tr.gif" width="141" height="41" alt="University of Bristol logo" />`

➢   In the same way, insert the image watch.jpg just below the line `<h2>Welcome!</h2>`.

> ➢ Save and preview in a browser. You should see something like this:

Figure 13 - adding an image

Note Providing alternative text for images is one of the basic requirements to ensure that your pages are accessible to people with disabilities.

Controlling the position of images

7.4 You can use the `align` attribute in the `<img />` tag to make images float left or right of your page, with text wrapping around.

> ➢ Add an `align` attribute to your two images, and experiment with the values `left` and `right`.

For example:

```
<img src="/images/uob-logo-tr.jpg" width="141"
height="41" alt="University of Bristol logo"
align="left" />.
```

Don't forget to save and preview after each change.

Once you've finished experimenting, align the University logo to the right and the image of the watch to the left.

> ➢ Save

7.5 You can also add space around your images by using the `hspace` (and `vspace` attributes. `hspace="x"` will add space on both the left and right sides of the image while `vspace="y"` will add space above and below the image (`x` and `y` are the number of pixels of space).

> ➢ Add some space around your images and experiment with different values.

For example:

```
<img src="/images/watch.jpg" width="141" height="41"
alt="close up image of pocket watch" align="right"
hspace="10" vspace="5" />
```

Don't forget to save and preview after each change.

Once you've tried a few things, remove the `hspace` and `vspace` from the University logo.

➢ Save.

Your page should now look something like this:

Figure 14 - floating images around text

Note The `align`, `hspace` and `vspace` attributes are deprecated; ideally you should use styles in a style sheet to control the alignment of and space around images. Styles give you more flexibility. For example with `hspace` and `vspace` you cannot add space just to one side, whereas in CSS you can.

Making an image a link

7.6 Finally, you can use an image as a link to another page. To illustrate this, we will link the University logo to the University home page.

All you need to do is enclose the image of the University logo between the `<a href="`*url*`">…</a>` tags as follows :

`<a href="http://www.bristol.ac.uk">``<img src="/images/ uob-logo-tr.jpg " alt="`**`University of Bristol home`**`" width="84" height="100" align="right" />`**`</a>`**

Note Since we've made the University of Bristol logo a link, it is appropriate to change the alternative text to `alt="University of Bristol home"`, which conveys the **function** of the image (link to the University of Bristol homepage). It is more meaningful in this context than `alt="University of Bristol logo"`, which merely describes the **content** of the image (University of Bristol logo).

7.7 Save and view in your browser. Notice that an ugly frame has appeared around your image now that it is a link. To get rid of it you need to add the `border="0"` attribute to the `<img />` tag:

```
<a href="http://www.bris.ac.uk"><img src="/images/uob-logo-
tr.jpg " alt="University of Bristol home" width="141"
height="41" align="right" border="0" /></a>.
```

Task 8 Using tables

Objectives To create a table for displaying tabular data.

Method You will create a standard table to present data.

Comments You will use the following tags:
table: `<table>…</table>`
table row: `<tr>…</tr>`
table data (cell): `<td>…</td>`
table head (cell): `<th>…</th>`

Displaying tabular data in a standard table

Creating a table in (X)HTML is a relatively simple task, based largely on the 'containing' and 'nesting' principles that apply to most tags or elements. A table is defined with the `<table>` element. Each table has a set of **rows**, and each row has several **cells** containing the **table data**. They are represented using the `<tr>` and `<td>` elements.

A simple, three-cell, three-column table (see Figure 11) would be coded as follows:

```
<h3>A simple table:</h3>

<table border="1"> <!-- table opens -->

  <!-- first row -->
  <tr>
    <th></th>
    <th>Col 1</th>
    <th>col 2</th>
  </tr>

  <!-- second row -->
  <tr>
    <th>Row 1</th>
    <td>cell 1 1</td>
    <td>cell 1 2</td>
  </tr>

  <!-- third row -->
  <tr>
    <th>Row 2</th>
    <td>cell 2 1</td>
    <td>cell 2 2</td>
  </tr>

</table> <!-- table ends -->
```

> **Hint:** a line such as `<!-- first row -->` is called a **comment**. Comments can be inserted anywhere in your page to document it. Although they appear in the source code, browsers ignore them. They must be contained between the special `<!--` and `-->` markup elements and there must be a space after the opening `<!--` and before the closing `-->`.

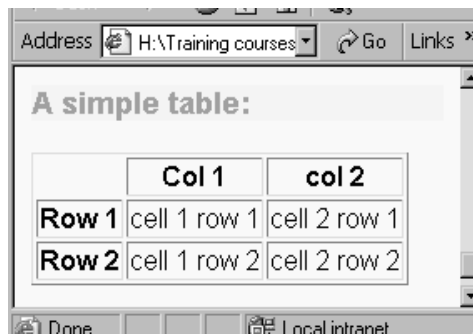… and look like Figure 15 (on the following page) in a browser.

Figure 15 - a simple table

Notice that:

- the attribute border in the opening `<table>` tag creates a border around the table (default is no border);

- the table header tags `<th>` centre-align the cell content and render the text bold.

- none of the tags - bar `<table>` - include attributes; this is the default table layout: the table is aligned to the left of the browser window, cell content is aligned to the left and cells stretch to accommodate the largest item in the cell.

➢ Open your `aboutme.html` file and replace the unordered list of the `My three favourite films` section with a table. Follow the example code above to create a four row, three column table as illustrated in Figure 16:

Figure 16 - favourite films table

Table attributes

Attributes can be applied to all table tags; in general, those applied to `<tr>` tags override those applied to the `<table>` tag. In the same way, `<td>` and `<th>` attributes supersede `<tr>` attributes.

`<table>` tag attributes - apply to the table as a whole:

Attribute	What it does	Values
width	specifies the width of the table	pixel value or percentage
align	aligns the table horizontally	left (default), center or right
border	sets a border between cells and the thickness of the external border	pixel value - default is 0 or 1
cellpadding	controls amount of space between the edge of a cell and its content	pixel value - default is 1
cellspacing	controls the amount of space between adjacent cells	pixel value - default is 2

For example:

```
<table align="center" width="80%" border="0" cellpadding="5"
cellspacing="0">
```

will create a borderless, centred table occupying 80% of the browser window width, 5 pixels space around the content of every cell and no space between cells.

Note that the align attribute in <table> is used to position the table within the browser window. In all other cases, align refers to the cell content.

`<tr>` tag attributes – apply to a row of cells

Attribute	What it does	Values
align	sets the horizontal alignment of content within the row cells	left (default), center or right
valign	sets vertical alignment of content within the row cells	top, middle (default) or bottom

For example:

```
<tr align="right" valign="bottom">
```

appearing within the table above, will render the row blue (overriding the pink table background colour), and align the content of each cell in this row to the right and at the bottom. Any other attribute is *inherited* from the <table> attributes.

`<td>` and `<th>` attributes - apply to individual cells

Attribute	What it does	Values
`width`	sets the width of an individual cell and therefore a column	pixel value (eg `100`) or percentage (eg `50%`)
`align`	sets the horizontal alignment of content within individual cells	`left` (default), `center` or `right`
`valign`	sets vertical alignment of content within the individual cells	`top`, `middle` (default) or `bottom`
`colspan`[7]	sets the number of **columns** a cell is to span	integer (that is, 2, 3, 4, etc.)
`rowspan`[8]	sets the number of **rows** a cell is to span	integer (that is, 2, 3, 4, etc.)

For example:

```
<td align="center" valign="top" width="20%">
```

will set/change the properties of a cell to centre the content at the top of the cell and the width to 20% of the table width.

Note By setting the width of one cell, you set the width of the corresponding column.

The `colspan` and `rowspan` attributes probably require further explanation. They are used to increase the width of a cell to span more than one corresponding cell in neighbouring rows or columns. The following example of a table shows a three-column table in which the first row consists of a cell spanning all three columns.

Figure 17 - table with cells spanning 4 columns and 3 rows

This example would be coded as on the following page:

---

[7] See glossary

[8] See glossary

```
<table width="100%" border="1" cellpadding="4">
  <tr align="center">
    <td colspan="4">cell 1 row 1 – spans four
                                    columns</td>
  </tr>
  <tr align="center">
    <td rowspan="3" width="30%">cell 1 row 2
                            – spans three rows</td>
    <td>cell 2 row 2</td>
    <td>cell 3 row 2</td>
    <td>cell 4 row 2</td>
  </tr>
  <tr align="center">
    <td>cell 2 row 3</td>
    <td>cell 3 row 3</td>
    <td>cell 4 row 3</td>
  </tr>
  <tr align="center">
    <td>cell 2 row 4</td>
    <td>cell 3 row 4</td>
    <td>cell 4 row 4</td>
  </tr>
</table>
```

➢ Returning to the table you have created in your `aboutme.html` document, apply some of these attributes in the `<table>`, `<tr>` and `<td>`/`<th>` tags, and see how the appearance of your table changes.

➢ Add one row at the top consisting of one cell spanning all 3 columns. Enter a title in the cell (for example, `My favourite films`).

Note There are a few other useful tricks you can do with tables.
For example, you can create navigational buttons. The technique is simple: create a table (containing a single row and as many cells as you need buttons for a horizontal navigation bar, or as many single cell rows as you need buttons for a vertical navigation bar) and change the background colour of each cell, that is, button.
Another useful technique is to use tables to organise your content into several newspaper style columns, or content areas.

Task 9 Review exercise - creating a personal website

Objectives   To create a personal website containing at least four pages using the course notes.

Method   Based on the pages that you have created so far, you will create a personal website on a topic of your choice, in your own time and upload it onto your personal web space.

Comments   Remember KISS (Keep It Simple Stupid)!

9.1 Use the pages you have created so far to create a personal website consisting of at least four pages on a topic of your choice (yourself, a hobby, a group that you belong to etc) using the concepts you learned during the course.

9.2 Once your site is ready, upload it to your personal file space on one of the University web servers (for example, `eis.bris.ac.uk` - for staff).

Don't forget:

Plan your site on paper before launching into coding (**site structure**, **navigation**, **visual presentation**, **content**).

Re-use your code by creating a template – don't create each page from scratch!

Check and re-check your code.

Good luck!

Appendix A. XHTML tags quick reference

Skeletal Tags

\<html\> … \</html\> *defines file as HTML document*

\<head\> … \</head\> *invisible, descriptive part of page*

\<body\> … \</body\> *visible part of page*
Attributes (values expressed within double quotes):
text = *"colour name/code"*
link = *"colour name/code"* (links)
alink = *"colour name/code"* (active links)
vlink = *"colour name/code"* (visited links)
background = *"URL of image fil"e*
bgcolor = *"colour name/code"*

Header Tags

\<meta /\> *meta tags add information to head*
Attributes (values expressed within double quotes):
name = *"type"* (for example, keywords/description)
content = *"string"*

\<title\> … \</title\> *title of page – appears in title bar*

Hyperlinks

\<a\> … \</a\> *anchor – defines link*
Attributes (values expressed within double quotes):
href = *url*
title = *"title text"*

Lists

\<ul\> … \</ul\> *unordered list*
Attributes (values expressed within double quotes):
type = *"bullet shape"* (*disc/circle/square*)
\<li\> … \</li\>
Attributes (values expressed within double quotes):
type = *"bullet shape"* (*disc/circle/square*)

\<ol\> … \</ol\> *ordered list*
Attributes (values expressed within double quotes):
type = *"symbol type"* (1/*A/a/i/I*)
\<li\> … \</li\>
Attributes (values expressed within double quotes):
type = *"symbol type"* (1/*A/a/i/I*)
value = *"starting value"*

\<dl\> … \</dl\> *definition list*
\<dd\> … \</dd\> *item to be defined*
\<dt\> … \</dt\> *definition*

Text Markup Tags (inline elements)

\<em\> … \</em\> *emphasised (usu. Italic)*
\<pre\> … \</pre\> *preformatted fixed-width text*
\<small\> … \</small\> *rel. smaller font*
\<strike\> … \</strike\> *strikethrough*
\<strong\> … \</strong\> *strong (usu. bold)*
\<sub\> … \</sub\> *subscript*
\<sup\> … \</sup\> *superscript*
\<tt\> … \</tt\> *typewriter text (for example, Courier)*
\<address\> … \</address\> *logical address format*
\<span\> … \</span\> to *select a chunk of text*
Attributes (values expressed within double quotes):
class = *"class_name"* (defined in a stylesheet)
id = *"id_name"* (defined in a stylesheet)

Layout Formatting Tags (block-level elements)

\<blockquote\> … \</blockquote\>. *indented quote*
\<br /\> *line break*
\<div\> … \</div\> *division*
Attributes (values expressed within double quotes):
align = *"left/right/center"*
class = *"class_name"* (defined in a stylesheet)
id = *"id_name"* (defined in a stylesheet)
\<h*n*\> … \</h*n*\> *heading (n=1-6)*
\<hr /\> *horizontal rule*
Attributes (values expressed within double quotes):
align = *"left/right/center"*
size = *"pixels"*
width = *"pixels or percentage of browser window"*
noshade = "noshade"
\<p\> … \</p\> *new paragraph (double line break)*

Images

\<img /\> *image*
Attributes (values expressed within double quotes):
align = *"type" (defines text position in relation to image)*
alt = *"text"*
border = *"pixel"s*
height = *"pixels"*
width = *"pixels"*
src = *"url of image file"*

Tables

<table> … </table> *table*
Attributes (values expressed within double quotes):
align = *"left/right/center"* (position of table on page)
border = *"pixels"* (width)
cellpadding = *"pixels"* (space around cell contents)
cellspacing = *"pixels"* (space between cell contents)
cols = *"number of columns"*
width = *"pixels or percentage of browser window"*
summary = *"string"* (brief summary of table content)

<caption> … </caption>
Attributes (values expressed within double quotes):
align = *"top/bottom/middle/right"*

<tr> … </tr> *table row*
Attributes (values expressed within double quotes):
align = *"left/right/center"* (cell contents – horiz.)
valign = *"top/middle/bottom"* (cell contents – vert.)

<td> … </td> *table cell*
Attributes (values expressed within double quotes):
align = *"left/right/center"* (cell contents - horiz.)
valign =*"top/middle/bottom"* (cell contents – vert.)
colspan = *"number of columns cell is to span"*
rowspan = *"number of rows cell is to span"*
width = *"pixels or percentage of table width"*
nowrap = *"nowrap"* (stops cell content from wrapping)

<th> … </th> *table head cell (bold & centred)*
Attributes (values expressed within double quotes):
*as for **<td>** element*

Additional Tags

Appendix B.   Using colours on the web

The RGB Color model

Computer monitors generate colours by mixing the primary colours of light (called *additive primaries*): Red, Green and Blue. This is known as the **RGB Color model**. Each colour channel is 8-bit (one bit being 0 or 1), that is to say it has 256 different brightness settings (from 0 to 255). This means that by mixing the various brightness values in the red, green and blue colour channels we obtain more than 16 million (24-bit or 256 x 256 x 256) colours! This is more than enough colours to realistically represent the world as we see it. When all three colours are turned on at full intensity the result is white, when they are all turned off the result is black.

The Hexadecimal code

(X)HTML uses the **hexadecimal code** (base-16 counting system) to represent colours on a web page.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Decimal to hexadecimal conversion

It is made up of a combination of six letters and/or numbers (0 to 9 and A to F) divided into three pairs, each pair representing one channel of the RGB values, and preceded by a hash (#) symbol.

(X)HTML Hex code	Red (#)	Green (#)	Blue (#)	Colour
#000000	00 (0)	00 (0)	00 (0)	Black
…	…	…	…	…
#F8F400	F8 (248)	F4 (244)	00 (0)	yellow
#8DCB41	8D (141)	CB (203)	41 (65)	apple green
#E3372E	E3 (227)	37 (55)	2E (46)	red
…	…	…	…	…
#FFFFFF	FF (255)	FF (255)	FF (255)	White

Examples of colours expressed as hexadecimal codes (RGB numbers)

Web-safe colours

While modern 24-bit, or "True Color" monitors can display more than 16 million colours, there are still a small number of web surfers out there using older 8-bit or 16-bit monitors limited to displaying 256 or 65,536 colours.

When a browser is called upon to display a colour that is out of the monitor's range, it has two options: it can *dither* (mix two colours to produce the missing one or it can *shift* the missing colour to one in the set available to the monitor. Sometimes the effect is hardly noticeable; sometimes it's a disaster.

This is less and less of a problem though, as the number of web surfers using an 8-bit or 16-bit monitor decreases every day.

However, there is no harm in using the web-safe colour palette; it simply limits your choices to 216 colours (40 are set aside for the operating system).

If you want to use web-safe colours, check the following site:

```
216 Web Safe Color Chart –
http://www.web-source.net/216_color_chart.htm
```

Foreground / background contrast

When choosing a colour scheme for your site, it is far more important to choose colours that combine and contrast well, so that you don't make it difficult (if not impossible) for visually impaired users (colour blind people in particular) to view your site.

The following sites will help you choose colour combinations that work:

Visibone – http://www.visibone.com/colorlab/

Color Schemes – http://www.pixy.cz/apps/barvy/index-en.html

Colour Selector – http://www.limov.com/colour/?ID=0P44BK6DL2P0010

Specifying the colour of (X)HTML elements

The best way to set the background and text colours of your web page is to use a style sheet (see tasks 5.3 and 5.4). The CSS properties you use are the **background-color** property to set the background colour of an element and the **color** property to set the colour of text.

For example, to set a light yellow background colour and a navy text colour for your pages you would redefine the <body> tag as follows:

```
body { background-color: #FFFFCC; color: #000066; }
```

You can also apply these properties to other (X)HTML structural elements. For example:

```
h1 { background-color: #FFFFCC; color: #000066; }
```
would set the background colour of all <h1> tags to light yellow and the text colour to navy.

```
td { background-color: #CCFFFF; color: #660000; }
```
would set the background colour to light green and the colour of text to dark red in all table cells.

Appendix C.   Design and planning tips

Directory names and structure and filenames

There is a good chance your website will continue to grow, so it is important to think carefully about the structure. Remember that directory and file names become part of the URL for a page.

- Keep directory and file names brief and sensible.
- Leave no blank spaces in names.
- Don't use underscore (_) as it is not always visible in underlined hypertext or the location bar on the browser.
- Use only lower case. Browsers are case sensitive, and it just makes things simpler. It is becoming the standard.
- If you are using relational links, make sure you don't move files without updating the links.

File Size

Large files take a long time to download and may cause viewers to give up on your website.

- Split large files into smaller ones and create links.
- Use images sparingly and only where they 'add value'. Use thumbnails where possible, and link to larger images if required.
- Colours, layout and formatting also add to file size.

Navigation

Analyse the information you are providing and identify logical routes through it and links between sections. Think about how people are going to want to look at your pages.

- Don't assume people will enter through your home page, so always include a link to it.
- Don't create dead end pages; always have a link back to a logical place.
- Don't rely on people using the 'Back' and 'Forward' buttons.
- Include a site map or index.
- Include a local search if you can.
- Include a navigation bar.

Consistent design

Be consistent in your design. This includes:

- page layout
- navigation
- colour schemes
- images and logos
- fonts and heading styles

This will help viewers navigate your pages, make them attractive, and save you time if you create a 'template' instead of redesigning every page.

Include these elements:

- signature line – institution, author/contact details, date of last update;

- link to home page;
- alternative text for images (use the `alt` attribute in the `<img />` tag.

Accessibility

You must make sure that your pages are accessible to all users whatever their disability (for example, visual impairment, motor impairment, dyslexia etc).
Detailed guidelines are available (http://www.techdis.ac.uk), but follow these guidelines:

- Use colour sensibly, with high contrast.
- Use ALT text for images.
- Don't use f**rames**.
- Don't use animations.
- Use meaningful hyperlink text.
- Provide alternative versions if required.
- Validate your pages the W3C Validator (validator.w3.org).
- Use metadata to provide a description of your page.

Printing web pages

Web pages often print badly. There are no page breaks, and a lot depends on the browser settings. If you want web pages to be printable, you might consider publishing your pages as word processing documents or .pdf files to be downloaded.

Maintenance

Don't leave your pages languishing once you have created them! Keep them up to date. Frequently check that:

- the content is up to date;
- the links are still working;
- the images are still linked;
- the date in the signature reflects the latest check, even if nothing on the page has changed.

Promoting your pages - search engines - metadata (keywords and description)

Now you've put in all this work – make sure people can find your pages!

- Register with search engines and indexes.
- Give your page a title - `<title></title>` in your header.
- Ask authors of related websites to include a link to your page.
- Announce it in relevant newsgroups.
- Add `<meta>` tags to the `<head>` of your home page to help indexing by major search engines:
  ```
  <meta name="description" content="a description of the page
  and/or website in about 25 words or less">
  <meta name="keywords" content="a list of keywords associated with
  your page, separated by commas" />.
  ```
- Tell colleagues, friends, etc about it.
- Include your home page URL in your e-mail signature.
- Submit it to `http://siteowner.linkexchange.com/sifree.cfm`, a FREE service that sends your URL to search engines.

Appendix D. Glossary of terms

Anchor	The anchor (`<a>`) tag is the HTML element used for defining both the source and the destination of a hyperlink.
Absolute link	A hypertext link that refers to the full path of a URL (for example, `http://www.bbc.co.uk`).
Boilerplate	A template document that you use as a skeleton for all your pages. It is the framework - page layout and content outline - into which you add the detail and character for each page. It is a way of re-using HTML code which ensures a consistent look and feel across your pages and saves time.
Bookmark	An anchor (`<a name="target_name">` tag) on a page used as a target for a hypertext link on the same or a different page within your website.
Domain name	The name given to a web server. The University of Bristol domain name is `www.bristol.ac.uk`; `ac` denotes "academic" and `uk` the country.
File access permissions	Access permissions determine who can read, write or execute programs in your files and directories. For example, you can set permissions so that others can view your files but not make changes to them.
FTP	File Transfer Protocol – a program for transferring files from your computer and a web server.
Home page	Your site's home page is the first page (also the welcome page) that will be accessed by users. The file should always be named `index.html`.
Hyperlink	A link to an object or location on the web.
Hypertext	A non-linear way of presenting information. Chunks or pages of text are linked together via paths, or hyperlinks.
Pixel	A pixel is one of the many tiny dots that make up the display on your computer. Display sizes vary but the most common one currently is 800 pixels wide by 600 pixels high (800 by 600).
Relative link	A hypertext link that refers to files within the same website (for example, `news/article.html`).
URL	Universal Resource Locator. The technical name of a web address.
WYSIWYG	'What You See Is What You Get' – HTML editing software that enables web designers to create pages in a browser-like view, and creates the HTML coding in the background.

Appendix E.   Useful resources

Books:

Castro Elizabeth (2002), *HTML for the World Wide Web with XHTML and CSS: Visual QuickStart Guide*, Fifth Edition, Visual Quickstart Guide, Peachpit Press

An excellent book for beginners, this book will also prove a great reference as you progress. It is clear, concise and reasonably priced (about £13).

On the web:

(X)HTML tutorials:

Webmonkey
HTML basics – http://hotwired.lycos.com/webmonkey/authoring/html_basics/.
Tables – http://hotwired.lycos.com/webmonkey/authoring/tables/.
These are excellent tutorials to brush up HTML basics and learn the power of tables for page layout.

W3Schools.com – www.w3schools.com/.
Lots of free web building tutorials starting with basic HTML tutorials and leading to more advanced stuff. If you only bookmark one site, make it this one!

Web Style Guide – http://info.med.yale.edu/caim/manual/.
Lots of useful information on website design. Won't teach you (X)HTML but will show you how to stretch it to design visually pleasing and well structured sites.

Usability and Accessibility:
University of Bristol Web Accessibility Policy and Guidelines –
www.bristol.ac.uk/Depts/Info-Office/policies/web_accessibility.htm.
If you are going to provide information on a University hosted web server, you **must** be aware of this policy and familiarise yourself with the guidelines.

TechDis – www.techdis.ac.uk/.
A JISC-funded service supporting the further and higher education community in all aspects of technology and disabilities and/or learning difficulties.

useit.com: Jakob Nielsen's website – www.useit.com/.
Jakob Nielsen is the usability guru. Check his 'Alertbox column' in particular; it contains useful 'How to' articles (for example, *Top ten guidelines for homepage usability*).

Validation tools:

W3C HTML Validation service – http://validator.w3.org/.
An extremely useful tool to check that your site conforms to W3C HTML standards. Enter the URL of your site and validate.

WAVE – http://wave.webaim.org/.
A tool to help web authors assess whether their pages are compliant with the W3C Web Content Accessibility Guidelines, and identify what changes are needed to make them compliant.