Using HTML to Create a Home Page

HTML is short for *HyperText Mark-up Language*. *Hypertext* is the ability to link a series of documents together so that clicking on a link in one document takes you to another one. *Mark-up* is a term long-used in the printing industry for the practice of marking up a text with instructions to the printer or type-setter. *HTML* markup tells your browser how to present the text on your screen. Since there a many different browsers, the final appearance can depend on which browser you use, and how wide or tall you make the browser window.

If you use mark-up that is not supported by a particular browser, it will be ignored. Although this is better than refusing to render the page, it has its downside: If you mistype mark-up, you will not be warned, making it hard to find mistakes.

In *HTML*, mark-up ***tags*** are enclosed between angle brackets: **<** and **>**, e.g., **<Title>**. Tags are case-independent, so **<TITLE>**, **<Title>** and **<title>** are equivalent. Most tags have to be matched by a corresponding end tag. They are the same as their starting tags, but start with **</** instead of **<**. Thus, the end of a **<title>** is marked with **<\title>**.

I have prepared two home pages. Let's start with the one called **simple home page.html**. You will need to open it two different ways: Opening it in your web browser will show you how it looks when rendered. Opening it in a simple text editor will show you how the *HTML* was written.

Like every *HTML* document, it begins with the **<html>** tag and ends with the **</html>** tag. It has two main parts: the first is short, and merely provides the text that appears in the title bar at the top of the window. It consists of the **<head>** tag, the **<title>** tag, the title itself, and two matching end tags.

The explanatory comments, which look like **<!** *comment***>**, are for your benefit, and do not affect the resulting web page in any way. The browser ignores them. You are unlikely to find comments in web pages you download because they waste download time.

There are other things that can appear between **<head>** and **</head>**. For example, there is provision for writing keywords that are associated with the content of the page. The idea is that search engines, such as *Google*, should use the keywords to help find what you are looking for. This worked well when *HTML* was used exclusively by scientists, but evil marketing people have so abused this feature that search engines now completely ignore it, preferring to focus on the actual text. That is it for the heading.

The ***body***, which is everything that appears *within* the window, makes up the bulk of the document. Not surprisingly, it is enclosed between **<body>** and **</body>** tags. The body tag illustrates the use of a parameter. Parameters have the form *keyword***="***value***"**. The parameter in this example says that the background should be tiled with a graphics (**gif**) file. Since the parameter does not specify a directory path, the file is assumed to lie in the same folder as the *HTML* page itself. ***Remember this!*** When you upload a web page to your server, make sure you upload the files it refers to as well. It is no good writing **:C/homepage/sandstone.gif**. Your ISP can't read your **C** drive.

The first body item is an image (**<img>**). The **src** parameter names the file where the image is stored—in the same folder as the *HTML* file, as before. The width and height are specified in pixels. (It is wise to limit your page to fit a VGA format of 640 by 480 pixels.) Strictly speaking, 140 and 100 should be enclosed in quotes, but you can usually get away with omitting quotes around numbers. The **alt** parameter is rather out of date. It specifies

that if the user has turned off images in their browser, the associated message will be displayed instead of the image. The user can then decide if they want to download the image. Mentioning the size of the image is a courtesy to give people with slow internet connections a guide to how long it will take to download. Nowadays, browsers rarely give you the option, and images tend to display before text.

The rest of the body is divided into *paragraphs* (** **). For historical reasons, ** ** is optional. Clearly the start of a new paragraph must mark the end of the previous one.

The **<h1>** tag specifies that the enclosed text should be in the biggest heading size and style. The exact size and font depend on the browser.

The **<hr>** tag draws a horizontal rule across the window, below which is another image—it doesn't use any features that we haven't already discussed.

Below that is a paragraph, which contains an *anchor* (****). The **href** parameter specifies a *hyperlink*. It gives the *URL* of my home page at Adelaide University—which is still there, nine years after I have retired. Between the **<a>** and **** tags is the text that will actually be displayed, typically in blue and underlined. The reader of the page will therefore expect that clicking this text will case the browser to jump to the page concerned. Notice that the *URL* not only specifies the web site (//) and the file within that site (/), but also the way the file should be interpreted: **http:** means that it is a hypertext page that should be rendered by the browser.

There then follows another horizontal rule, and a smaller-sized heading (**<h4>**), followed by an unnumbered list (**
). Each list item (
**) will be preceded by a bullet. There is no **
** tag, but the list as a whole has to be terminated by **
**. If I had written **
** and **
** (*ordered* list) the items would have been numbered instead of bulleted.

Notice that the last list item specifies a *file* and does not have the usual **http:** method. Therefore the browser will *not* attempt to interpret the poem as an *HTML* document.

The next paragraph starts with a horizontal rule, then an image followed by some text on the same line. The fact that the text was *written* on a new line is irrelevant. *HTML* treats any amount of white space as one single space character. The sequence ** ** is used to deliberately create a *non-breaking space*. (If you want to leave three spaces, you have to write **   **. The spaces are non-breaking in the sense that all three spaces would appear on the same line.)

There are several similar **&…;** sequences used later. They are needed when the character you want would either be ignored, or misinterpreted, e.g., **<**. Similar sequences are also used for foreign accented characters, such as **é**.

Because new lines in the *HTML* source document are treated as spaces, it was necessary to write **
** to force a line break. A line break differs from a new paragraph in that it does not increase the space between lines.

Following the second **
** is an image that moves irritatingly. There is nothing special in the *HTML* to make this happen. The image file is an *animated gif*. The graphic moves of its own accord because of the way it was made. There are any number of animated gifs out there on the web. If you see one in a web page and you like it, display the *HTML* source, find out where it lives, and download it!

The anchor that follows specifies that if it is clicked it should create an email message (**mailto:**) addressed to **barry.dwyer@isp.net.au**. Although my email address does not get

displayed, thus keeping it secret from the user, any decent spambot will pick it up from the *HTML* text. Hence my comments in the following paragraph.

This final paragraph illustrates the use of the **<tt>** tag, which sets the enclosed text in `teletype font`. Characters that can appear in *HTML* tags had to be treated specially: **<** is written as **<** (less than), **>** as **>** (greater than), and **"** as **"**. In the last two lines of the paragraph, the **<b>** tag sets the following text in **boldface** type. Following the line break, the text is set in *italics* (**<i>**). Since the **<b>** tag has not yet been cancelled, the text is actually in ***boldface italics***. At the end of the text both the **<i>** and **<b>** tags are cancelled. ***In the right order.*** As you stack them up, so you must unstack them. To help get this right, one can indent the text, as this example shows.

The document ends by unstacking all the remaining tags.

The main problem with the simple home page is that its layout depends on the size and shape of the browser window. (Try changing its width to see what happens!) The file **home page.html** shows one way to overcome this problem: by using a *table*. The tutorial advantage of showing you this approach is that tables are useful for other things too.

Open the file in your browser and editor, like you did earlier. The heading is unchanged.

The body begins with a ** ** tag. It defines a *division* of the body in which certain rules apply, in this case **align=center**. Notice two things about this parameter, **center** has to be spelt the American way, and although the quotes that should surround it are missing, it is still interpreted correctly—by my browser at least. The effect is that the body will remain centred in the browser window, irrespective of its width. That is its only purpose, and has no other visible effect. It is the only division in the body.

The **<table>** tag announces the start of a table, and the **width** parameter determines that it will be 400 pixels wide. Tables are divided into *rows* (**<tr>** and **</tr>**) and within rows, they are divided into *columns*. The contents of a cell are enclosed between **<td>** (table datum) and **</td>** tags. The **<td width=50%>** tag means that the first column fills half of the table width; **<td width=200>** would have achieved the same effect.

The first table datum is a paragraph, with the **align="center"** option. This option turned out to be ineffective, because the alignment of an **<h1>** heading defaults to left. Therefore the **center** option had to be reasserted in the **<h1>** tag itself. The text contains two **
** tags: the first positions the heading downward slightly compared to the image which will appear on its right, the second breaks the heading where it will fit the table most neatly.

The second table datum is the picture of me, but this time it is aligned to the right—where I thought it looked best. As a result, the **hspace** parameter is ineffective, but I have included it for tutorial reasons. It guarantees a blank space of 10 pixels on its left.

The next row merely contains a horizontal rule, but because I wanted it to run right across the page, I specified that the datum should span both columns (**<td colspan="2">**).

The third row again has two items. The first contains an anchor, as before, and the second contains a logo. The **align** parameter is used as before.

The fourth row again has one datum spanning both columns, which is the same bulleted list as before.

The fifth and final row again spans both columns. It contains familiar material, but carefully laid out. The phone icon is indented by 10 pixels (**hspace**)—because I thought it

looked better that way. It is immediately followed by a non-breaking space. The space is to compensate for the fact that 'mobile' has one more letter than 'phone'. As a result, the two phone numbers line up prettily. After that, the story is much as before, except that there are even more tags to unstack.

Why weren't the fourth and fifth rows combined, considering they are both full width? No particular reason. I just thought of them as being separate when I was writing the *HTML*. I don't suppose it would make much difference if they were merged into a single row.