

Basic HTML

© Dewald van Niekerk
1999
dewald@nethabitat.co,.za

	<u>Page</u>
1. INTRODUCTION	5
2. FUNCTIONS OF THE ICONS	6
3. HTML FUNDAMENTALS	7
3.1 Viewing HTML code	7
3.2 Copying HTML code	9
3.3 Modifying HTML code	11
3.4 Creating an HTML document	12
3.4.1 Closing and non-closing elements	12
3.4.2 The <html> element	14
3.4.3 The <body> element	14
3.4.4 Defining the <header>, <title> and <comments>	15
3.4.4.1 Defining the <header>	15
3.4.4.2 Defining the <title>	16
3.4.4.3 The <meta> tag	17
3.4.4.4 Defining comments	17
3.4.5 The <base> tag	17
3.4.6 Defining colour for backgrounds, text, links and visited links	18
3.4.6.1 <body bgcolor="*****">	19
3.4.6.2 <text="*****">	19
3.4.6.3 <link="*****">	19
3.4.6.4 <vlink="*****">	19
3.4.6.5 	20
3.4.6.6 Defining colours	20
4. WORKING WITH TEXT	22
4.1 Allocating a font to your page	22

4.2	Formatting font properties	23
4.3	Using preformatted text	25
4.4	Defining a list	26
5.	WORKING WITH GRAPHICS, SOUND AND FILES	28
5.1	Why use graphics?	28
5.2	Graphic formats	29
5.3	Why use sound?	30
5.4	Sound file formats	31
5.5	Naming your files	32
5.6	Allocating all files to one folder	32
6.	INTERMEDIATE GRAPHICS USAGE	34
6.1	Capture/"Steal" graphics	34
6.1.1	Stealing a picture	35
6.1.2	Stealing a background	35
6.2	Defining background images	37
6.2.1	Background generators	37
6.3	Fixed background properties	38
6.4	Defining elements	38
6.4.1	Adding an image to your HTML document	39
6.4.2	Assigning alternative text	39
6.4.3	Assigning width and height values	40
6.4.4	Assigning border values	40
6.4.5	Assigning vertical and horizontal space values	40
6.4.6	Assigning two images to the same space	41
6.4.7	Aligning text in relation to images	42
6.5	Use multiple graphics	43
6.6	Creating image maps	45
6.6.1	USEMAP	45
6.6.2	ISMAP	47

7. CONCLUSION	49
----------------------	-----------

1. INTRODUCTION

After completing this module you will have a greater knowledge of the fundamental components of HTML. You will be able to design a basic web page using text and basic graphics. You will also learn the theory behind graphics, sounds and transferring files – elements that you will need in order to complete the next module.

It will periodically be required of you to complete certain **activities** that will assist you in the learning process. It is completely up to you to complete these activities. Bear in mind that experimentation is one of the key factors in understanding HTML.

2. FUNCTION OF THE ICONS



OUTCOMES: Each unit of this study guide has a certain objective. This icon indicates that the particular objective for that unit will follow.



DEFINITION: This icon indicates that a definition will follow. It is important that these definitions are studied carefully.



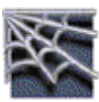
ACTIVITY: The activity icon indicates that you must perform an activity. The activity will help you to consider a certain aspect of the text in more detail.



KNOWLEDGE REVIEW: Wherever you see this icon you must complete self-evaluation questions which will test if you have mastered the preceding section. This icon will be encountered at least at the end of each unit.



TAKE NOTE: The “Take Note”-icon indicates that there is an important piece of information which you need cognisance of.



TRY THIS: The “TRY THIS”-icon indicates that an exercise will follow that you should complete on your PC.

3. HTML FUNDAMENTALS



OUTCOMES:

After the completion of this chapter you will be able to:

- view HTML code
- copy and paste HTML code from existing web pages
- modify and save source code
- create a basic HTML document
- understand the difference between closing and non-closing elements
- work with various tags or elements e.g. <html>, <body> etc.
- define colours on a web page

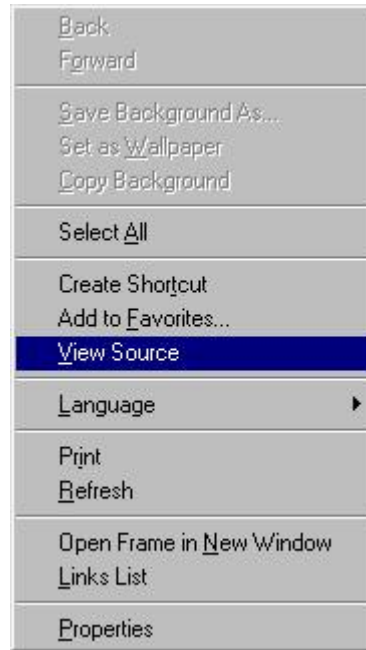
3.1 Viewing HTML code

Viewing existing HTML code can often facilitate an inexperienced programmer to identify areas that cause problems. Furthermore it enables you to learn various features on existing web pages that you may encounter while surfing the web.

There are two ways of viewing existing web pages on the web, using Microsoft's Internet Explorer. These two methods differ somewhat, and could cause some confusion if you are not sure of what you are looking for.

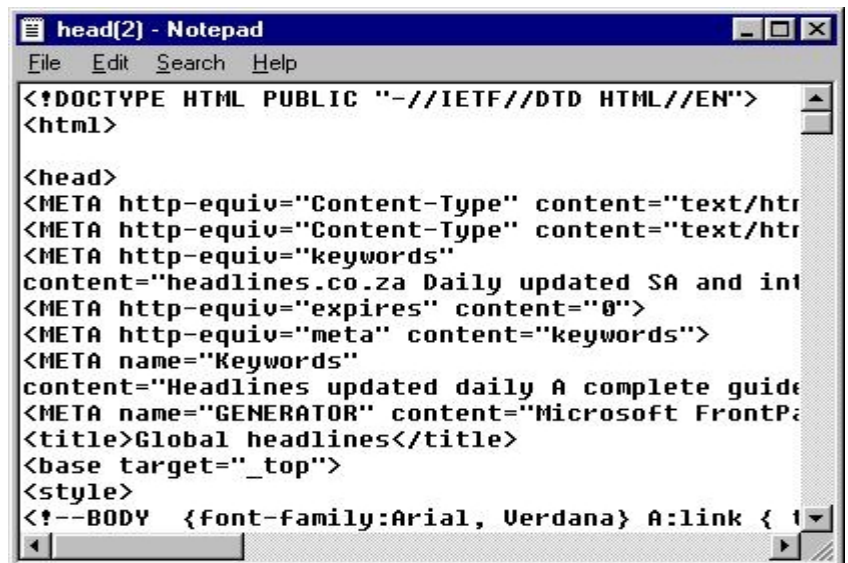
The first method is using the right-click of your mouse anywhere on the open web page, and then click **View source**. A separate window will open, with the source code of the specific page as a notepad file.

*Right-click with your mouse anywhere on the web page to bring up a menu. Select **View source** to open a new window with the source code of the page as a Windows Notepad file.*



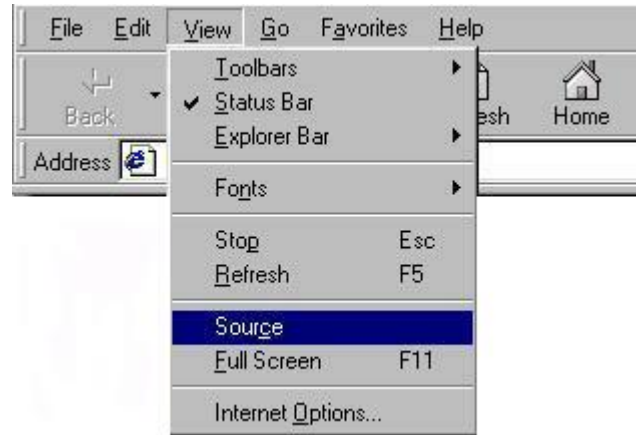
This operation will open a new window, which contains the specific page's source code. The code is displayed as a Windows Notepad file. Notepad is a small text-editing program, with very limited features, but is more than adequate to view and edit HTML source. Notepad is a standard feature of Windows 95/98, and also formed part of Windows 3.1. Use the scroll bars at the bottom and at the right of the Notepad window to scroll through the source code.

*Notepad will open in a new window when you select **View source** from the menu. Use the scroll bars to scroll through the file.*



The second method for viewing the source code of an existing web page is equally simple. Left-click with your mouse on the **V**iew button on the standard menu-bar on your browser. This will bring up a drop-menu, from which you need to select **S**ource in order to view the source code.

*Left-click with your mouse on the **V**iew button on the standard menu-bar. Click **S**ource to view the source code.*



This operation will once again open a second window, with the source code in a Notepad file.

In most cases either of these methods will have exactly the same result. There are however some exceptions when a page consists of two or more frames . As you will learn later, frames are actually two different web pages, which interact and are displayed on a single page.

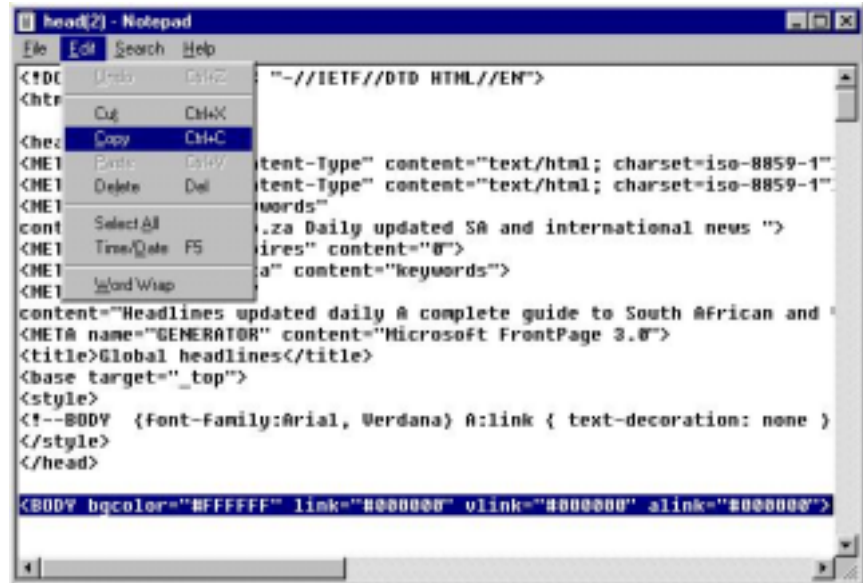
This means that when you right-click in a frame to view its source, you will see only the source for the specific frame, and not the whole display, as you see it on your browser. To view the code for the page that was actually used by your browser, use the second method. This will, however not display the source of the frames that you are looking at, but rather the frameset code.

3.2 Copying HTML code

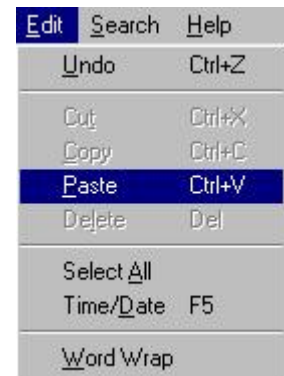
An easy way of “programming” is to copy HTML code from existing web pages, and then modifying that code to suite your own needs. This is done by viewing the specific page’s source code using either of the methods described above, and then highlighting the source that you want. Once you have

highlighted the relevant source code, use the **E**dit button in the notepad window, and then select **C**opy. This will copy the selected text to Windows' Clipboard.

Select **C**opy from the Menu-bar to copy the selected text to a new document.



Once you have copied the source code you want, you can paste it into a new or existing HTML document by opening the document in Notepad or Word, left-clicking on the **E**dit button above the toolbar and selecting paste. Make sure that the cursor is at the position where you want to place the new text before pasting.



Selecting **P**aste from the drop-menu will place the text that you selected and the copied at the current position of the cursor in your new or existing document.



Note the shortcut keys next to each option on the drop-menu. The shortcut key for copy is **Ctrl+C**, and **Ctrl+V** is the shortcut for paste. You can use these shortcuts in stead of using the drop-menus. Simply select the text you want to copy as usual, then press **Ctrl+C**. Open the document into which

you want to place the selected text, move the cursor to the desired position, and press **Ctrl+V**.

3.3 Modifying HTML code

You can modify existing HTML code to suite your needs, by firstly “capturing” it in the manner described above, and then editing it in a text-editing program, such as Microsoft Word or even Notepad.

The more you learn about HTML before trying to modify it, the easier you will find it to achieve your goals with the captured copy. Don’t be afraid to experiment with captured code – it is a very easy way of learning more about HTML, and it also gives you good insight into the actual writing of the code.

Certain elements, such as frames and tables, are often very complex, which could make it difficult, and in the case of frames often impossible, to capture the source code. That doesn’t mean that you shouldn’t try working with captured code. Even experienced programmers often “grab” some code from an existing page, simply to save time.

When modifying existing HTML code, there are a number of important rules to keep in mind. Such as:

- Try and find a page that matches your goals as closely as possible
- Don’t take any more code than you need – a multitude of code can confuse, rather than help
- Test the code that you edit frequently to ensure the validity thereof

In order to understand which bits of code you need to copy in order to achieve the desired effect on your web page; you will need a basic understanding of some HTML tags.

HTML tags are code that precede specific instructions and or words that are interpreted by a browser and displayed. Tags are used to demarcate certain sections in your document, which changes the way that your browser displays the document. They define certain sections of your document, and there are basically no parts of an HTML document that work without tags.

3.4 Creating an HTML document

Creating your own, simple HTML document shouldn't take more than a few minutes. Simply open any text editing program, and start with a blank document. (For the purposes of this section we will refer to Notepad as our designated text editor.)

When starting a new document, there should be a number of questions that you ask yourself, such as:

- How do I let my computer know that I am working with an HTML document?
- How do I let my browser know how to display the different bits on my page?
- What do I want to achieve with this page?

3.4.1 Closing and non-closing elements

Even before learning more about the various elements that are used in any HTML document, it is important to understand that not all elements work in the same way. They are largely grouped into two different categories, known as *closing* and *non-closing* elements.

All elements are defined as tags. Tags are pieces of code that always start with a smaller than symbol (<), and ends with a greater than symbol (>). The format is as follows:

`<tag in here>`



Closing elements are elements that require one tag to inform the browser that an action is starting, and a second tag to show that it is complete.

The tags will not function if both of these elements are not present, and the browser will not interpret the contents of that tag.

The tags that are used to indicate the end of a closing element follows the same basic format as the start tag, but the opening symbol (<) is followed by a forward slash (/). The format for end tags is:

`</tag in here>`



Non-closing elements are elements that comprise one single tag, and require no ending-tag at the end to indicate the completion of that specific action.

It is important that you realise that certain elements can work with only one tag, and that others need two tags in order to function. This will however be specified when discussing the different elements.

Keep in mind when capturing code from existing pages that you include closing-elements where necessary. If you copy the start tag of one of the closing elements and not the end tag as well, the code is rendered useless, as it will not function properly.

3.4.2 The `<html>` element

Writing an HTML document is similar to writing an essay. A typical HTML document contains an overall encompassing tag (`<html>`), a header (`<head>`), a title (`<title>`) and a body (`<body>`). In order, however, to give structure to your page, you need to define these elements before starting a document.

The `<html>` element is used to tell the browser that the document it is displaying contains HTML code. It is usually the very first lines of code in any HTML file and serves to surround **all** of the remaining code, including all other elements, such as the `<body>` tag.

A typical example of the `<html>` tag is:

```
<html>
```

```
The rest of the document and tags should be placed here.
```

```
</html>
```

The `<html>` tag has two components, which places it in the category of **closing elements**.

3.4.3 The `<body>` element

The body of any HTML document, as its name suggests, contains all the text, images and tags that make up the page, together with all the HTML elements that provide the formatting of the page. An example of the typical use of the `<body>` element is:

```
<html>
```

```
<body>
```

The rest of the document included here

```
</body>
```

```
</html>
```

Take note that the `<body>` element is enclosed by the `<html>` elements. As with the `<html>` element, the `<body>` element is a closing element, which means that the whole body of your document must be enclosed by the starting (`<body>`) and closing (`</body>`) tags.

The body tag has much more potential than its four letters may suggest. It can dictate the background of your page, the colours that are used and even the style that is followed.

3.4.4 Defining the `<header>`, `<title>` and `<comments>`

A typical web page consists of several sections, which are defined within elements such as `<html>`. Each of these has a specific goal, and though they are not essential to the functioning of your page, it is recommended that you always include them as a reminder to yourself for future reference. It will function as a guide to other users should they look into your page, and also for search engines, which uses some information from the top of your page in order to determine what information is stored on your page.

3.4.4.1 Defining the `<header>`

The head of an HTML document is a collection of information regarding the document. Certain tags function within the `<head>` tag that are not displayed by a browser but gives useful information about your page.

The header has no influence over the actual look of your page, but is needed to give structure and identity to your page. The following tags are contained within the `<head>` tag:

```
<title>...</title>
```

```
<meta...>...</meta>
```

3.4.4.2 Defining the `<title>`

Every HTML document must have a `<title>` element. The title should identify the contents of the document and in a global context, and may be used in history lists and as a label for the windows displaying the document. Normally, the text contained within the `<title> ... </title>` elements will be displayed in the title bar of the browser window.

The title element must be used within the head of the document and may not contain anchors, paragraph elements, or highlighting. Only one title is allowed in a document. You have to put the `<title>` elements between the `<head>` and `</head>` tags.



The length of a title is not limited, however, long titles may be shortened in some applications. To minimise the possibility, titles should be kept as short as possible. Also keep in mind that a short title, such as “homepage” may be meaningless out of context. An example of a meaningful title might be “Homepage of John Smith”. This is the only element that is required within the `<head>` element.

The correct format for setting the title of a page is:

```
<HEAD>
```

```
<TITLE>Homepage of John Smith</TITLE>
```

`</HEAD>`

3.4.4.3 The `<meta>` tag

The META element is used within the `<HEAD>` element to embed any useful information not defined by other HTML elements. Such information can be extracted by servers/clients for use in identifying, indexing and cataloguing documents.

In addition, HTTP servers can read the content of the document head to generate response headers corresponding to any elements defining a value for the attribute HTTP-EQUIV. This provides document authors a mechanism (not necessarily the preferred one) for identifying information that should be included in the response headers for an HTTP request and forms the basis of the client pull mechanism.

3.4.4.4 Defining comments

The `<COMMENT>` element can be used to 'comment' on text. Alternatively you can use the `<!... ..>` element to set a comment.

Any text placed between the `<COMMENT>` and `</COMMENT>` or `<!... ..>` elements will not be displayed on the screen, allowing comments to be placed in HTML documents.

```
<COMMENT>This text won't be displayed. I can say what I  
like here, it doesn't matter!</COMMENT>
```

3.5 The `<base>` tag

The `<base>` element allows the URL of the document itself to be set, to help browsers in situations where the document may be read out of context. It is

especially useful in allowing browsers to determine any **partial** URL's e.g.(../mypage.html), or **relative** paths that may be specified (for example, in `<a href>` elements e.g. ``, or paths pointing to `` (images) e.g. `` specifies `www.myhost.com` to be the base from which all relative URLs should be determined.

The **Netscape Navigator** (from version 2.0) and **Internet Explorer** (from version 3.0) adds one other attribute to the BASE element. With the introduction of targeted windows the TARGET attribute as used in anchors (`<a>`) can be used.

This allows the author to pick default named target window for every link in a document that does not have an explicit TARGET attribute. Its format is:

```
<base target="default_target">
```

3.4.6 Defining colours for backgrounds, text, links and visited links

Part of any web page would include defining the colours which you wish to use. This includes the **background**, **text**, **links** and **visited links** colours. Defining these colours will always appear **directly** after you have defined the `<header>`, `<title>`, `</title>`, `</header>`, and will therefore be the first line in your `<body>`. Always try to house the following command in this order as the first command in your `<body>`.

```
<body bgcolor="*****" text="*****" link="*****"  
vlink="*****">
```

The ***** in each of these instances indicates the hex triplet colour code which you need to insert in order to obtain the correct colour. But first a short explanation of each command will follow.

3.4.6.1 <BODY BGCOLOR="*****">

By using this command you insert a solid colour as your background instead of an image. The only tricky part is to determine the correct code for the colour you wish to use (more about this later). Keep in mind that if you use this background colour command you cannot use the `<body background="***.***">` command to insert an image as the background as well. Should you use both, the one will override the other.

3.4.6.2 <TEXT="*****">

This command specifies the colour of the text on the page. This is useful because certain colour backgrounds will not appear well with the standard blue colour. You can therefore change these colours to your needs and design different colour schemes for your web page.

3.4.6.3 <LINK="*****">

This command is similar to the text command. By using this command you designates a colour to the text that is used as hyperlinks to other pages. In most cases this is bright blue, but you can change this as well to suite your needs.

3.4.6.4 <VLINK="*****">

Using the `<vlink="*****">` command you assign a colour to the text that depicts hyperlinks that have already been viewed. In most cases these hyperlinks appear magenta.

The above mentioned tags are non-closing elements.

3.4.6.5

By using this command you can change the colour of text in the middle of any page. You can have different colours all over the page, anywhere you like. Remember to close your font colour command with a `` in order to return the rest of your page to the original font colour as commanded by the `<text="*****">` command.

3.4.6.6 Defining colours

Defining colours are quite simple. There are 6 figures, and each of these figures range from 0 to f. This means, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f. "0" for a particular value is the **weakest** and "f" is the **strongest**.

White is therefore a combination of all the colours, and black is a combination of no colour. To obtain white, the command would be "ffffff" (all colour slots strong) and to obtain black it would be "000000" (all colour slots weak).

The first two letters, if made strongest, would make the colour red ("ff0000"). The second two letters, if made strongest, would make green ("00ff00"). And the third two colours, if made strongest, would make blue ("0000ff"). So, by mixing and matching these combinations of red, green, and blue, you can make any colour you want. For example, yellow is "ffff00" (red and green) and grey would be "555555" (the middle of all colours, a mix of white and black). Although this might sound confusing, the best way to learn how to use these

colour schemes is to experiment with the basic knowledge of the main colours. If you find this very difficult a fairly complete list of all the combinations and the colours they are associated with can be found as an appendix to this study guide.

In conclusion, the general colour tags used in most web pages looks as follows:

```
<body bgcolor="000000" text="ffffff" link="0000ff"
vlink="ff00ff">
```

Background colour = White

Text colour = Black

Link colour = Blue

Visited link colour = Magenta

Colour pickers

HTML Colour Pickers are handy when setting background and text colours in your web pages. They provide a sort of graphical interface from which you can pick a colour or set of colours, and then the programme provides the code. Colour pickers are built into most modern HTML Editors, but for the purposes of this course, in which we will be mainly using Windows Notepad as an editor, colour pickers can come in extremely handy to make sure that you actually get the colours that you want. Various colour pickers can be downloaded from Tucows (<http://www.tucows.co.za>).

4. WORKING WITH TEXT



OUTCOMES:

After the completion of this chapter you will be able to:

- allocate a font to your page
- manipulate fonts to suite your needs
- use preformatted text
- define a list

4.1 Allocating a font to your page

Picking the font that you want to use on your page is very important. The font is one of the most important components of your web page, seeing as most of what the visitors will (hopefully) be looking at, will be written.

Keep in mind, however, that:

- You need to use a font that is common to most PC's, otherwise your chosen font will not be displayed on those PC's, and will be substituted with the default font
- The font can be used to enhance the corporate identity of your web page, so try not to change it to often
- Even though you can set a base font for your page, you can still make changes for certain elements, like headings or quotes

A popular font to use on web pages, which is clean and easy to read, is **Verdana**, but you can use any of the most popular fonts, such as:

Arial	The quick brown fox jumps over the lazy dog... 1 2 3 4 5
Courier New	The quick brown fox jumps over the lazy

	dog... 1 2 3 4 5
Verdana	The quick brown fox jumps over the lazy dog... 1 2 3 4 5
Times New Roman	The quick brown fox jumps over the lazy dog... 1 2 3 4 5
Georgia	The quick brown fox jumps over the lazy dog... 1 2 3 4 5
Bookman Old Style	The quick brown fox jumps over the lazy dog... 1 2 3 4 5

- **Note:** All the above examples are in 12 point, with the same spacing

The above table shows how various fonts can differ in size and spacing, despite having the same point size. Clearly Verdana must be a good choice, because while it may take up slightly more space than other fonts, it is very easy to read, and it works well on a variety of backgrounds.

4.2 Formatting font properties

HTML allows the designer to manipulate fonts extensively to suite every need. By changing point size, font face and even set *italics* or **bold**, you can enhance the look of your web site a great deal.

All font attributes are set within the `` tag, which is a closing element (remember: it needs a second tag, namely `` in order to function).

The first attribute that you can set is the font face that is to be used. You may set two or more font faces, in which case the first will be used as the preferred font, but if it is not available the browser will automatically try the next font in line. The format for setting the font face is:

```
<font face="name, name2, name3"> </font>
```

This will set the font to “name”. Should “name” not be available, the browser will try “name2” and then “name3”. In real terms your format would look like this:

```
<font face="courier new, times new roman, verdana"> Now  
the text between the two font tags will be displayed in  
courier new, times new roman or verdana. </font>
```

As you can see, you need to place the font face names in inverted commas, after the “equals” sign that follows the “face” attribute.

The font face is not the only attribute that you can set. You can also change the size and colour of the font – and you do that within the same set of tags that you used to set the font face.

Setting the size of the font is done by inserting the element. Valid values range from 1-7, with the default FONT size being 3. The value given to size can optionally have a '+' or '-' character in front of it to specify that it is relative to the default size.

```
<FONT SIZE=4>changes the font size to 4</FONT>
```

```
<FONT SIZE=+2>changes the font size to the default size  
+2</FONT>
```

Both Internet Explorer and Netscape support the ability to change the font colour. The colour attribute sets the colour which text will appear in on the screen. Specifying a hexadecimal code that denotes the colour generally sets the font colour. As an example:

```
<FONT COLOR="#FF0000">This text is red.</FONT>
```

Please note that the “colour” element is spelled “color” for HTML purposes.

4.3 Using preformatted text

The Preformatted Text element presents blocks of text in fixed-width font, and so is suitable for text that has been formatted on screen, or formatted for a mono-spaced font.

The `<pre>` element may be used with the optional `WIDTH` attribute, which is an HTML Level 1 feature. The `WIDTH` attribute specifies the maximum number of characters for a line and allows the HTML user agent to determine which of its available fonts to use and how to indent the text (if at all). If the `WIDTH` attribute is not present, a width of 80 characters is assumed. Where the `WIDTH` attribute is supported, widths of 40, 80 and 132 characters should be presented optimally, with other widths being rounded up.

Within preformatted text, any line breaks within the text are rendered as a move to the beginning of the next line. The `<p>` element should not be used, but if it is found, it should be rendered as a move to the beginning of the next line. It is possible to use Anchor elements and character-highlighting elements are allowed. Elements that define paragraph formatting (headings, address, etc.) must not be used.



It is at the discretion of individual browsers how to render preformatted text and where "beginning of a new line" is to be implied, the browser can render that new line indented if it sees fit.

Example of use:

```
<PRE WIDTH="80">
```

```
This is an example of preformatted text.
```

This text will appear as follows:

```
</PRE>
```

This is an example of preformatted text.

This text will appear as follows:



Internet Explorer 4.0 doesn't change the font pitch when different values are given to the `WIDTH` attribute.

Within a Preformatted Text element, the constraint that the rendering must be on a fixed horizontal character pitch may limit or prevent the ability of the browser to render highlighting elements (such as ``, `<I>` etc.) specially.

4.4 Define a list

Two types of list can be used: ordered and unordered lists.

The Ordered List element is used to present a numbered list of items, sorted by sequence or order of importance and is typically rendered as a numbered list, but this is as the discretion of individual browsers.



The browser does not sort the list elements when displaying the list. This sorting should be done manually when adding the HTML elements to the desired list text.)

An ordered list must begin with the `` element that is immediately followed by a `` (list item) element:

E.g. :

```
<OL>
<LI>Click on the desired file to download.
<LI>In the presented dialog box, enter a name to save the
file with.
<LI>Click 'OK' to download the file to your local drive.
</OL>
```

Would render as:

1. Click on the desired file to download.
2. In the presented dialog box, enter a name to save the file with.
3. Click 'OK' to download the file to your local drive.

The Unordered List element is used to present a list of items which is typically separated by white space and/or marked by bullets, but this is as the discretion of individual browsers.

An unordered list must begin with the `` element that is immediately followed by a `` (list item) element. Unordered lists can be nested.

E.g.

```
<UL>
<LI>First list item
<LI>Second list item
<LI>Third list item
</UL>
```

Would render as:

- First list item
- Second list item
- Third list item

5. WORKING WITH GRAPHICS, SOUND AND FILES



OUTCOMES:

After the completion of this chapter you will be able to:

- identify the need for using graphics on your page
- choose a graphic format to suite your needs
- identify the need for using sound on your page
- choose a sound bite format and length to suite your needs
- accurately and effectively name and manage your files

5.1 Why use graphics?

The use of graphics on web pages can turn an otherwise boring and colourless page into a virtual painting. Using graphics have certain advantages, such as:



- It allows the designer to spruce up a page
- It adds atmosphere and identity to a page
- It allows you to create synergy between your pages, thereby linking them together
- It makes it easy for foreign visitors to navigate and understand your site, if you make use of semantic graphical symbols (semantics in a nutshell is an internationally understandable language which is made up by symbols that are used by people everywhere, for instance stop signs, construction hats or yellow and black stripes – each of those has a certain meaning to you, and will mean the same thing to someone else, despite a possible language difference.)

Unfortunately there's also a downside to using graphics on your web page.

- It slows down the loading of your page
- It uses more space on the web, which limits your space resource
- They can be difficult to control without being skilled in a multitude of programs
- They can contribute to confusion if the visitor's screen resolution is not set at the same size as your screen was when you originally made the page

Still, don't see these negative points as deterrents. Adding a graphic at the cost of a little more space and speed could make the difference between an exciting and a boring page. Conversely a slow loading image could cause so much frustration for the user, that he doesn't even finish loading the page – in which case the graphic is certainly not worth the space and speed that it takes up!

Keep in mind, however, that very few web pages have no graphics on them. Often graphics are not big, impressive full colour photographs, but rather tiny icons that help the user to identify the various sections of the page or site.

Plan your graphic usage carefully. A good rule to follow is that you should only use graphics that has a goal – and not waste speed and time on pictures merely for the fun of it.

5.2 Graphic Formats

The format of the graphic that you use could make or break your page. There are literally tens of different graphic file formats, each with it's own pro's and cons. The most popular graphic file formats *for web pages* are **.jpg's** and **.gif's**.

The **.jpg** files (files with the extension “jpg”), are called J-peg files, in reference to the **J**oint **P**hotographic **E**xperts **G**roup, which first conceived them. These files compress information about many colours (up to 16 million) in the image into a smaller file, which loads quickly. The image quality of a J-peg can be very high, if used with careful consideration. There are, however, file formats with a much higher quality level.

The **GIF** (graphic interchange format) file format was conceived by a German company called Compuserv, who needed a stable, high-resolution file format to save pictures in. The attributes of GIF files are much more versatile than those of other file formats. You can, for instance, animate a gif, or save it with a transparent background.

In general browsers support only GIF and J-Peg files, but you are likely to run into some other formats when working with graphic files. If you want to inset a photograph, icon or general graphic, try to use the J-peg format. If, however, you need something more interactive, with animation or a transparent background, use the GIF format.



ACTIVITY:

Explain which two file formats you would use in your web pages.

5.3 Why use sound?

As with graphics, sound can make a huge difference to the entertainment value of your site. Once again there are certain advantages to using sound, such as:

- it adds more warmth and atmosphere to your site
- it is impressive for the visitor

And on the down-side:

- it is generally slow in loading
- can be corny, rather than impressive

Bear in mind, however, that while most pages use graphics, relatively few use sound. And with good reason too!

Many of the older browsers do not support the use of sound in web pages, which obviously rules it out completely. Furthermore few visitors bother to wait for a whole site to load before making their selection to a new page. So the sound seldom gets played.

Obviously if you are doing a page that refers to music, or sounds, it would be silly not to include it, but simply putting in sound for the entertainment value usually proves more trouble than it is worth.

5.4 Sound file formats

Sound files on the internet are usually confined to one single format, namely Wave files (extension **.wav**). This file format is widely used, not only on the web, but also in many other applications, including Windows itself (ever hear that sound when Windows 95 starts up? Well, there you go...)

Wave files can be downloaded directly from the Internet, or even recorded on your own machine – provided you have the hardware to do so. All you need is a sound card and either or both a CD-ROM drive and a microphone. Using the Windows application “Sound recorder” should set you off in the right direction.

Unfortunately wave files grow in size rapidly as the length of the sound clip increases. A one-minute wave file can easily take up 650 KB – a huge chunk

when you consider that an average web page (excluding graphics and sound) uses only roughly 25 KB.

5.5 Naming your files

When starting with a new web site, it is very important to first decide on a fixed structure. This sort of planning is handled in more detail later on. For now, however, you need to realise that the key to a successful web site is structure.

As for actually naming the various pages that will eventually make up your web site, there are a number of things to keep in mind:

- Always save your “start” page (the very first page of your site) as **index.html**
- Try to be logical about naming your pages
- Limit your filenames to 8 characters in length, even though longer filenames are supported by Windows 95/98, not all servers respond to them
- Standard rules of naming apply, i.e. special characters (like *, % or |) may not be used

5.6 Allocating all your files to one folder

When designing a web page, many novices tend to create new folders for each aspect of their page, i.e. graphics, sounds, and icons. While this is a very structured approach, it complicates matters when it comes to actually putting the page together.

In order for you to properly preview your page, you need to tell the browser where to find each file that is used in your page, be it a graphic, sound clip or even just an icon. If you save each of these files in a different folder, and, as

you must, specify it to your browser, the page should work just fine on your hard drive.

The problem comes, however, when you want to publish your web page. You will find it very difficult to mirror all the folders on your hard drive on the actual server, which means that visitors' browsers will not be able to locate the files that you specified in your source code.

Work with all the files connected to your page/pages in one folder on your hard drive. This method will negate the need for you to specify file locations in your code, which leaves only the actual file names to be specified. The browser will automatically assume that the files that you are specified are in the "current" folder, and will load them without a problem. Make sure that all the file names are in lower case in your source code, as few servers support upper case characters.



KNOWLEDGE REVIEW:

1. Why should you use one folder for all your files?
2. What would happen if you upload your pages to a server but you referred to a file to be used on your page using the full file location e.g. C:\My Documents\HTML\new.gif

6. INTERMEDIATE GRAPHICS USAGE



OUTCOMES:

After the completion of this section you will be able to:

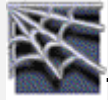
- capture graphics from the web to use on your own web site
- define background colours form your page
- use images as backgrounds
- use multiple graphics
- define the IMG elements of a graphic
- create clickable image maps
- fix background properties to your background colours or background image
- use animated graphics

6.1 Capture/"Steal" graphics

One of the easiest ways to gather graphics to use on your own pages is to capture or "steal" these graphics. It should however be kept in mind that a lot of these graphics are under copyright protection and you should make sure that you might in fact use these graphics. If no notice of copyright is present on the web site you are free to use these graphics. If you do encounter a site with copyright it is a good idea to first e-mail the creator of those pages and obtain permission to use the graphics. The resources on the internet are legion and by doing a bit of searching you can locate literally millions of sites designed to provide free graphics and clip art.

6.1.1 Stealing a picture

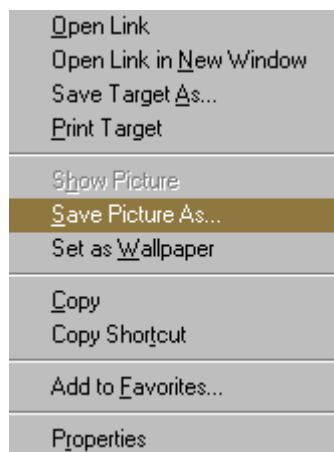
Before starting to capture graphics it is a good idea to first create a new folder or folders on your hard drive to which you can save your captured graphics. Graphics can be captured in the following way:



TRY THIS:

The following activity will teach you how to capture or “steal” web graphics:

- connect to the web
- open any web page containing graphics
- place your cursor over any graphic or picture present on the page
- right click to open a drop-down menu
- from the menu choose **Save Picture As...**
- save the graphic file to the folder you created on your hard drive



6.1.2 Stealing a background

In this way you can save **any** graphic or picture from the Internet to your hard drive. You can also choose to set any picture as the wallpaper of your desktop. Another option when capturing graphics is to capture the

background of a specific web page. When capturing a background image you have the following options:

- Save Background As...
- Set as Wallpaper
- Copy Background

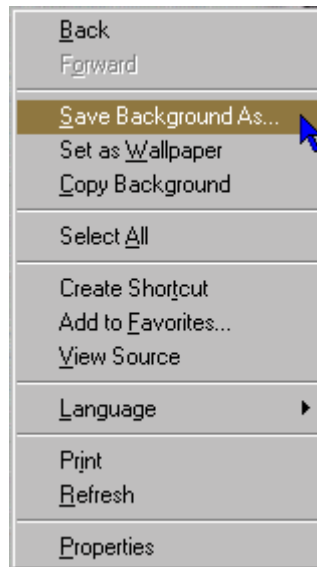
The first option will allow you to choose the folder in which you which to save the background. The second option will automatically set the background of the web page as the wallpaper of your desktop. The last option will copy the background to your clipboard, which you in tern can past in whatever programme you choose.



TRY THIS:

The following activity will teach you how to capture or “steal” a background image from the web:

- connect to the web
- open any web page containing a background image
- place your cursor over any area on the background of the page
- right click to open a drop-down menu
- from the menu choose **Save Background As...**
- save the background to the folder you created on your hard drive



6.2 Defining background images

The `<body background="***.***">` command allows you to insert an image and designates it as the background you want to use. The "***.***" indicates the file name which you which to insert e.g. `<body background="curves2.jpg">`. The image `curves2.jpg` will therefore be used and tiled as the background image to your page. As mentioned earlier, this command should be used separately from the `<body bgcolor="*****">` tag.

6.2.1 Background generators

There are a number of handy programs available that can help you create your own, unique backgrounds. These backgrounds work in much the same way as backgrounds on your desktop do. They are small graphic files, in either of the two most popular formats. These graphics are then tiles to automatically fill the screen.

Pay a visit to Tucows at <http://www.tucows.co.za>, and look under graphic editors for a nice selection. Alternately type in "background generator" in any search engine, and you should find a number of programs for download.

One program that we can recommend is called Reptile. This program comes with a free license, which means that it doesn't cost you anything – and it is great for creating unique backgrounds.



Reptile is an excellent background-generating program that comes with a free license. Download it from Tucows, on the world wide web.

6.3 Fixed background properties

As many of you have already seen, while scrolling up or down a web that contains an image as a background you will find that the background scrolls with the text. The background is therefore fixed in relation to the text. One very intuitive element that one can use in the `<body background=***.***>` tag is the `<body bgproperties="fixed">` element. By applying the `bgproperties="fixed"` element to the body tag all your background images will be fixed in relation to the background and not the text. What this in effect means is that you will get the effect of your text scrolling on top of the background while the background image remains fixed. This creates an excellent 3D illusion and adds a professional look to your site. The `bgproperties="fixed"` element should be inserted before the `background=***.***` element in the body tag e.g.:

```
<body bgproperties="fixed" background="curves2.jpg">
```

6.4 Defining elements

One of the great things about the web is the ability to create and share some great looking documents across platforms. Inserting images on your web page will not only make it more interactive but will add a professional feel to

the whole document. You should however always keep in mind the more graphics you insert the longer it will take to download from the server to your browser. So in effect graphics are great to use but you need to be realistic about the download time of your web site. Even with fast network access to the Internet, some users find documents loaded down with images to be annoying and in many cases you will find that users turn off graphics viewing in their browser options. However, a dash of colourful images can be nice. Images are also often necessary to make a point that can't be made using text only.

6.4.1 Adding an image to your HTML document

To add an image to your document, you need to convert it first into .gif, .jpg or .bmp format. There are a number of tools for doing that but was covered in the previous module. As mentioned earlier always put the images that you want to show in your document into the same directory as your document.

The HTML command for inserting an image at the current position takes the following form:

```

```

The location does not have to be a full URL. It is recommended that you will stick to only inserting the image name because if you upload your pages to your server the server will be required to look for your graphics in a directory not known to that server, seeing that it refers to a location on your computer.

6.4.2 Assigning alternative text

Another attribute of the tag is the assigning of alternative text to the image. Alternative text can be any alphanumeric of any length e.g.

```

```

When your cursor hovers over the image `pic2.gif` the alternative text “**My dog**” will appear.

6.4.3 Assigning width and height values

The `width` and `height` attributes allow the browser to determine the text layout surrounding images before the entire image has been downloading, which can significantly speed up display of the document text. These attributes are measured in pixels. If the author specifies these, the viewer of their document will not have to wait for the image to be loaded over the network and its size calculated. **Internet Explorer** and **Netscape** use image placement mechanisms, so that if the display of in-line images has been turned off, the space that the images would occupy in the page is marked as if the image were there (with any `alt` text being displayed in the place holder). This allows authors to be sure that the text layout on the page will be as desired, even if the user is not displaying the images.

An example of allocating width and height to an image:

```

```

6.4.4 Assigning border values

This allows you to control the thickness of the border around an image displayed. It is useful if the image is to be a hyperlink because the `border` can be set to "0" to avoid the display of the standard blue hypertext link border e.g.:

```

```

6.4.5 Assigning vertical and horizontal space values

For the floating images (i.e. those displayed with an `align=left|right`

attribute explained a bit later) it is likely that you will not like the text wrapped around the image to be pressed up against the image. **vspace** controls the vertical space above and below the image, while **hspace** controls the horizontal space to the left and right of the image. The Value should be a pixel value e.g.

```

```

6.4.6 Assigning two images to the same space

Using the **lowsrc** attribute, it is possible to use two images in the same space. The syntax is:

```

```

Browsers that do not recognise the **lowsrc** attribute cleanly ignore it and simply load the image specified by the **src** attribute. Browsers that support this attribute however, will load the image called "lowquality.gif" on their first layout pass through the document. When the rest of the document has been completely loaded and formatted on the page, the browser will then redraw the page and load the image specified by the standard **src** attribute. This allows the author to specify a low resolution (or smaller file size version of the main image - perhaps a grey scale version) image to be displayed initially while the document is loading, which is later replaced by the higher quality version.

Any graphic file format that the browser supports can be used interchangeably within the **lowsrc** and **src** attributes. You can also specify width and/or height values in the **** element, and both the high-resolution and low-resolution versions of the image will be appropriately scaled to match. However, if no width and height values have been set, the values used for the **lowsrc** image (i.e. the dimensions of that image) will be used to re-scale the **src** image. This is to minimise page format disruption that would be caused

by the browser trying to load two different sized images into the same page space.



TRY THIS:

Programme an HTML code where you:

- Insert a high and low quality graphic file
- Assign alternative text
- Assign a hyperlink
- Set the border to 3

6.4.7 Aligning text in relation to images

Another element that you can use is the alignment of text in relation to the image that you have inserted. The image element, which is empty (no closing element), has the following attributes:

```
align="left|right|top|texttop|middle|absmiddle|baseline|bottom|absbottom"
```

The following attributes must be inserted in the `` tag after your image file name, depending on your preference of alignment.

align=left will float the image to the left margin (into the next available space there), and subsequent text will wrap around the right hand side of that image.

align=right will align the image with the right margin, and the text wraps around the left.

align=top aligns itself with the top of the tallest item in the line.

align=texttop aligns itself with the top of the tallest text in the line (this is usually but not always the same as **align=top**).

align=middle aligns the baseline of the current line with the middle of the image.

align=absmiddle aligns the middle of the current line with the middle of the image.

align=baseline aligns the bottom of the image with the baseline of the current line.

align=bottom aligns the bottom of the image with the baseline of the current line. (Essentially, this is the same as **align=baseline**)

align=absbottom aligns the bottom of the image with the bottom of the current line.

alt="alternative text"

The **alt=""** allows you to allocate alternative text to the graphic you are inserting. This alternative text will be activated as soon as the users mouse cursor pauses over the image in question. Example of use:

```
 Trees from South Africa
```

6.5 Use multiple graphics

Multiple graphics can be inserted in various places in your web page by using the above mentioned procedures. One of the disadvantages when inserting an image is that at this stage it will be difficult for you to exactly insert the image in the exact spot you wish it to be. How to insert your images in the

exact spot will be covered a bit later. When inserting multiple graphics you can use the `<p align="right|left|center"></p>` closing element and `<p>`, `
` empty elements (no closing element) to space the graphics across your page. Inserting multiple graphics could look as follows:



Lizzy the Lizzard



Lizzy the Lizzard



Lizzy the Lizzard

HTML code:

```
Lizzy the  
Lizzard  
<p>  
<br>  
<p align="center">Lizzy the Lizzard</p>  
<p>  
<br>  
Lizzy the  
Lizzard
```

6.6 Creating image maps

An image map can be defined as an image that has been divided in different clickable areas known as hot-spots. By clicking on these different regions, different resources can be accessed from the same graphic. Image maps can be divided into **server** (`ismap`) and **client-side** (`usemap`) image maps. The difference between the two will be explained in the following section

6.6.1 USEMAP

Adding the `usemap` attribute to an `` element indicates that the image is a **client-side** image map. The `usemap` attribute can be used with the `ismap` attribute to indicate that the image can be processed as either a client-side or server-side image map (useful to ensure browser independence of HTML documents). The value used in the `usemap` attribute specifies the location of the map definition to use with the image. If the argument to `usemap` starts with a '#', the map description is assumed to be in the same document as the `` element. Consider the following example:

```

```

This would use the map described as "map1" in your HTML file as the overlay for the image file "image.gif". The map definition (see below) can be included either within the HTML document itself where the image is embedded, or in a completely separate file. In the last mentioned instance the HTML code will look as follows:

```

```

The `maps.html` indicates that the map definition is located in the HTML file named `maps.html`. This file needs to be in the same folder as the HTML file referring to this file.

Assigning active regions or 'Hot-Spots'

The different active regions of the image are described using **MAP** and **AREA** elements. The map describes each region in the image and indicates the location of the document to be retrieved when the defined area is activated. There is no need for the `<map> ... </map>` container to appear immediately after the `` tag. The anchor "map1" serves to tie them together so the `<map> ... </map>` container may appear **anywhere** in the page, even **before** the associated `` tag. The basic format for the **MAP** element is as follows:

```
<map name="name">
  <area shape="shape" coords="x,y,..." href="reference">
</map>
```

The **name** specifies the name of the map so that it can be referenced by an `` element

The **shape** gives the shape of the specific area. Currently the shapes "rect", "circle" and "poly" are supported.

The **coords** attribute gives the co-ordinates of the shape, using image pixels as the units.

For a rectangle (`shape="rect"`), the **coords** are expressed as "left-x, top-y, right-x, bottom-y".

For a circle, (`shape="circle"`), the **coords** are expressed as "centre-x, centre-y, radius".

For a polygon (`shape="poly"`) (an irregular shape), the **coords** are expressed in pairs of co-ordinates (i.e. "x1,y1,x2,y2,x3,y3... ") which

defines the pixel co-ordinates of the various points of the polygonal image hot-spot.

The `nohref` attribute can also be added to the `<map>` tag. This attribute indicates that clicks in this region should perform no action. An example could look as follows:

```
<map name="name">
  <area nohref shape="shape" coords="x,y,...">
</map>
```

Any click in the above-mentioned are will lead to no reference at all.

An arbitrary number of `<area>` elements may be specified. If two areas intersect, the one that appears first in the map definition takes precedence in the overlapping region.

6.6.2 ISMAP

The `ismap` attribute identifies an image as a **server side image map**. To be able to employ this type of image maps in HTML documents, the HTTP server which will be controlling document access, must have the correct **cgi-bin** software installed to control image map behaviour, i.e. the document must have access to an image map handling script and the map file defining the graphic hot-spots. The `ismap` attribute can be used in the following way:

```
<a href="http://machine/htbin/imagemap/sample">
</a>
```

In this example the `` refers to the location of the overlay for the image file on the server. The `` refers to the image file.

To accurately determine the right co-ordinates for your image map can be a time consuming task. Luckily some very innovative people has designed programmes that will determine the co-ordinates for you. Most of these programmes works in a drag-and-drop environment and even generates the HTML code for you. **Map This** and **Coffee Cup Image Mapper ++** are both two great little programmes that you can use to generate co-ordinates for you. **Tucows** has a large selection to choose from, but unfortunately all the mappers that are currently available carry only shareware licenses, which means that they eventually cost some money.



KNOWLEDGE REVIEW:

1. Explain the difference between client-side and server-side image maps.

7. CONCLUSION

By now you should have a working knowledge of HTML. You will now be able to develop a basic web page, assign text and graphics. As have been previously mentioned, experimentation is the best method to learn. HTML is quite clear-cut, it works or it doesn't. Troubleshooting can be a timely task, but at the end of the day you will have gained the knowledge you require for effective design.

The following section will give you in depths view into the more complicated HTML. Before attempting to engage in the following section, be sure to have completed the all the **TRY THIS**, **ACTIVITIES** and **KNOWLEDGE REVIEWS**.