

Chapter 3. HTML: BASICS

Table of Contents

| | |
|--|----|
| Objectives | 1 |
| 3.1 Basics | 2 |
| 3.1.1 HTML Markup | 2 |
| 3.1.2 Nesting HTML Tags | 2 |
| 3.1.3 Creating HTML Text using Notepad++ | 2 |
| 3.1.4 Standard HTML Document Structure Format | 3 |
| 3.2 HTML Formatting | 4 |
| 3.2.1 The Browser As Formatter | 4 |
| 3.2.2 Paragraphs, Line Breaks and Preformatting | 5 |
| 3.2.3 Headings, Horizontal Rules and Meta Tags | 5 |
| 3.3 Lists | 7 |
| 3.4 HTML Comments | 9 |
| 3.5 Anchors | 9 |
| 3.5.1 Linking to Email Addresses & other Non-Web Links | 10 |
| 3.5.2 Linking to Sections within Documents | 10 |
| 3.5.3 Targeting Windows | 11 |
| 3.5.4 Link Appearance | 12 |
| 3.6 Multimedia | 12 |
| 3.6.1 Graphics | 12 |
| 3.6.2 Objects | 13 |
| 3.6.3 ImageMap | 14 |
| 3.7 Writing Good HTML | 14 |
| 3.8 Discussion and Answers | 15 |
| 3.8.1 Discussion Topics | 15 |
| 3.8.2 Discussion of Activity 2 | 15 |
| 3.8.3 Discussion of Activity 4 and 5 | 15 |
| 3.8.4 Discussion of Activity 6 - Lists | 15 |
| 3.8.5 Discussion of Activity 7 – Comments | 17 |
| 3.8.6 Discussion of Activity 9 – Linking to sections within a document | 17 |
| 3.8.7 Discussion of Activity 11 – Changing appearance of links | 17 |

Objectives

At the end of this chapter you will be able to:

- Create HTML files using Notepad and run them on a Tomcat server;
- Use HTML tags to write HTML files;
- Format HTML files;

- Create lists in HTML files;
- Use anchors in HTML files;
- Use multimedia in HTML files.

3.1 Basics

3.1.1 HTML Markup

HTML pages are created by tagging textual information with HTML markup. HTML markup consists of tags, which appear inside angled brackets `<` and `>`

An example of an HTML tag is ``, which causes text to appear in bold. `` only notes where text should begin to appear in bold, while the tag `` marks the end of the emboldening. Most HTML tags have a corresponding end tag, which is specified by the name of the tag preceded by the `/` character.

So, to create the text:

Internet Commerce is great!

The text is marked up as:

```
<B>Internet Commerce is great!</B>
```

Another example of an HTML tag is `<I>`, which causes text to appear in italic. In HTML 4.01, the `<I>` tag was used to render text in italics. However, this is not necessarily the case with HTML5. Style sheets can be used to format the text inside the `<I>` element. This will be demonstrated later.

Note that tags are not case-sensitive. In other words, `` or `` are the same tag, both specifying bold text.

3.1.2 Nesting HTML Tags

Text may be both bold and italicised. This is done by using both the `` and `<I>` tags. When doing so, it is important to remember not to overlap HTML tags. In other words:

```
<B><I>Internet Commerce  
is great!</I></B>
```

is **correct**, but

```
<B><I>Internet Commerce is great!</B></I>
```

is **wrong**.

Overlapping tags is a common mistake. Although Web browsers are usually smart enough to work out what is meant, it can lead to problems. Furthermore, for an HTML page to be considered valid HTML, it must contain no overlapping tags.

To Do:

Read the section on "HTML Tags" in your textbooks.

3.1.3 Creating HTML Text using Notepad++

This section covers the creation of an HTML page. You will need a Web browser and a text editor. Use

HTML: Basics

any text editor you wish to, but the following Activity descriptions will use Notepad++. Notepad++ is a free Windows editor that also supports several programming languages. For example, you will notice that HTML keywords are highlighted in different colors.

1. Open your Web browser. This sections' goal is to create a Web document that can be opened with your browser.
2. Open Notepad++. It can be found by selecting Start, then All Programs, then Notepad++.
3. Type the following text into Notepad++: your name and the module number (CSC5003). Save this file as start.txt.
4. Now load start.txt into the browser by dragging start.txt onto your browser.
5. The browser should now display the text contained in **start.txt**. (If it does not, make sure that you have saved **start.txt** and that this is the file you are opening).
6. Once you have displayed start.txt, return to Notepad. Add the text "Internet Commerce", and save the file again.
7. Return to the Web browser and reload the document (by using either by using the Refresh or Reload toolbar buttons, or by selecting File/Open once again).
8. If you are able to see the new piece of text, you have successfully used Notepad to create your first Web page.

Activity 1: Getting started with HTML

This Activity adds HTML tags to start.txt.

1. Open your file **start.txt** in Notepad.
2. Mark up the text "Internet Commerce" so that appears in bold. Do this by placing the tag in front of the text, and at the end of the text, as shown below:
`Internet Commerce`
3. Save the file as **start.html**, since it contains some HTML formatting. Save the file with this new name (using Save As). Note that saving it as **start.htm** is also accepted. Other than the obvious, the letter "L," there's not much of a difference between the two extensions. Most, if not all, web browsers and servers will treat a file with an HTM extension exactly as it would a file with an HTML extension, and vice versa¹.
4. Load **start.html** in the Web browser. **Internet Commerce** should now appear in bold.
5. Return to Notepad and add more text, some of it in bold and others in italics. (Remember <I> is the tag for italics) Save the document and reload it.

3.1.4 Standard HTML Document Structure Format

Although a number of HTML tags have been introduced that markup how text should be displayed in a browser, a correct HTML document must always include certain structural tags. These tags are <HTML>, <HEAD>, <BODY> and <TITLE>.

The <HTML> tag should be placed around the document's contents; this tells the browser that the whole document is written in HTML. Like a person, all HTML documents have only one head and one body. All the text of the HTML document should be inside either the head or the body. Roughly, the <HEAD> holds information about the document itself, and the <BODY> holds the information that should be displayed. The document's <TITLE> is given in the <HEAD>. The title is shown at the very top of the browser (i.e. in the title bar) — not in the browser window itself.

The standard structure of an HTML document is:

```
<HTML>
<HEAD>
<TITLE>Text to appear in the title bar of the browser</TITLE>
</HEAD>
<BODY>
```

¹ http://www.sightspecific.com/~mosh/www_faq/ext.html

HTML: Basics

```
    The text to appear in the main browser window.  
</BODY>  
</HTML>
```

This format should always be used when writing HTML documents.

Note: students are often confused about the use of the <BODY> tag, and they often include multiple body tags. This can lead to problems later on, so make sure to use only one <BODY> tag.

To Do: Read the section on HTML document structure in your textbooks.

Activity 2: Structuring your HTML document

In this Activity you will convert your file that contains a few HTML tags into a correctly structured HTML document. Open start.htm in Notepad.

1. Add the <HTML> tag on the first line of the file (before anything else).
2. Add the </HTML> end tag on the last line of the file (after everything else).
3. Add the document header by adding a <HEAD> tag on the line underneath the <HTML> tag and the </HEAD> tag on the line beneath that.
4. Between the opening and closing <HEAD> tags, add the <TITLE> and </TITLE> tags.
5. Enter the text "My first Web page" between the <TITLE> tags.
6. Underneath the </HEAD> tag, create the body of the document by entering the <BODY> tag.
7. At the bottom of the document, add the </BODY> tag just before the </HTML> tag.
8. Save the file.

If you have problems correctly formatting the file, look at the code in 3.1.4.

You are probably thinking that it looks the same as the previous document. However, if you look closely at the title bar you should see that it now displays the words "My first Web page". The main difference, however, is that the browser now has to do a lot less work to do, since the document informs it of the HTML's structure.

Activity 3: Loading your HTML file on Tomcat

The previous chapter guided you through tomcat installation. Let us launch the start.html file using the tomcat webserver. Make sure that your tomcat server has been started. Save start.html in the folder myapps that you created within the webapps folder. Load **start.html** in your browser by typing `http://localhost:8888/myapps/start.html`

3.2 HTML Formatting

3.2.1 The Browser As Formatter

As you will recall, it is the browser that actually formats the HTML document. But when it displays text, where does it put the line breaks?

The browser automatically determines the position of the line breaks. It tries to fit all of the text into the current window so that the user does not need to do any horizontal scrolling. If the browser window changes size, the browser reformats the document.

It also ignores extra spaces. If there are two spaces between words in the HTML file, the browser will display the text in exactly the same way as if there was only one. Blank lines are ignored in a similar way. The browser also tries to correct errors in incorrect HTML (such as HTML containing overlapping tags). When doing so, the browser may incorrectly interpret the HTML document, making it a wiser choice to write correct HTML. Sometimes it can be difficult to have the browser format things as you want. You will learn more tricks on how to do this as you work through the HTML units.

3.2.2 Paragraphs, Line Breaks and Preformatting

The tag `
` is used to start a new line. `
` is a standalone tag, that means there is no closing `</BR>` tag. Note that `
` does not place a line space between the two lines. To do that you need to use the `<P>` paragraph tag. Do not forget to add the end tag `</p>` although most browsers will display HTML correctly even if you forget the end tag. The tag `<pre>` defines preformatted text. The text inside a `<pre>` element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

Activity 4: Paragraphs, Line Breaks and Preformatting

In this Activity you will use the `<P>` and `
` tags to create line breaks in text. We will also demonstrate the use of `<pre>`.

1. Load Notepad and begin a new HTML document.
2. Enter the usual structural HTML tags. Set the title to "Formatting text".
3. Within the body type in the following text exactly as it appears below. Note how 'This is cool' has been typed. Do not use any HTML tags to format it at this stage.

Users of HTML are sometimes surprised to find that HTML gives them little control over the way that a page is displayed. It should be remembered that HTML was developed as a means of marking up the structure of a document not as a way of determining its presentation. Formatting text to appear on a Web page is therefore different from formatting text to appear in a printed document.

This

is

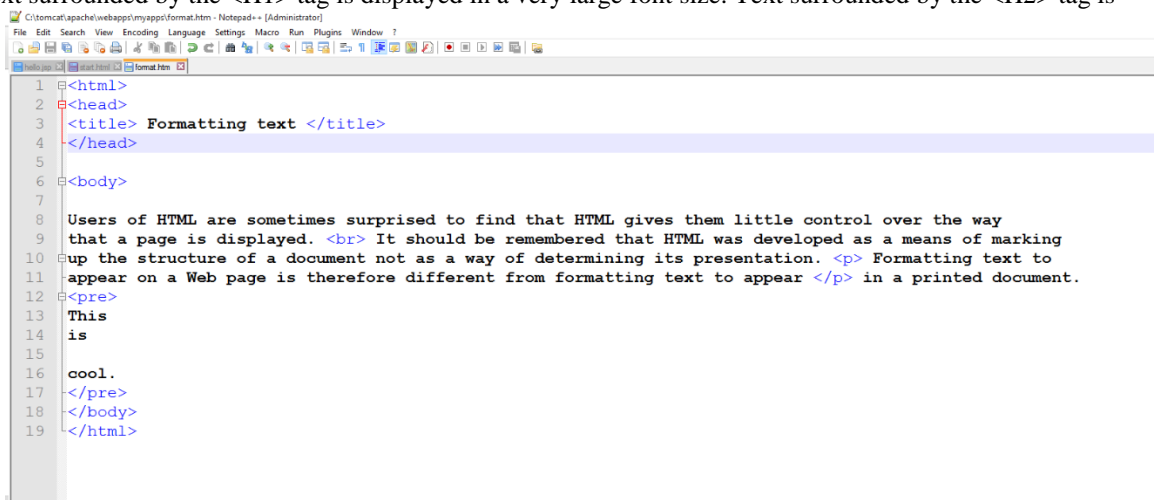
Cool.

4. Save the document as **format.html** in your myapps folder and load it in your browser to view it. Note that 'This is cool' is displayed without the line breaks.
5. Resize your browser and watch how the text is reformatted to fit in the resized browser window.
6. Return to Notepad and make the changes as shown in Figure 3.1.
7. Save the file again and load it in your browser to check your HTML. Resize the browser and watch how the document is reformatted for the resized window.

3.2.3 Headings, Horizontal Rules and Meta Tags

The DOCTYPE declaration defines the document type to be HTML. In HTML5 this is written as `<!DOCTYPE html>`. The `<!DOCTYPE>` declaration helps the browser to display a web page correctly. There are different document types on the web. To display a document correctly, the browser must know both type and version. The doctype declaration is not case sensitive. All cases are acceptable:

Another set of HTML tags are the headings tags. These are `<H1>`, `<H2>`, `<H3>`, `<H4>`, `<H5>` and `<H6>`. The text surrounded by the `<H1>` tag is displayed in a very large font size. Text surrounded by the `<H2>` tag is



```

1 <html>
2 <head>
3 <title> Formatting text </title>
4 </head>
5
6 <body>
7
8 Users of HTML are sometimes surprised to find that HTML gives them little control over the way
9 that a page is displayed. <br> It should be remembered that HTML was developed as a means of marking
10 up the structure of a document not as a way of determining its presentation. <p> Formatting text to
11 appear on a Web page is therefore different from formatting text to appear </p> in a printed document.
12 <pre>
13 This
14 is
15
16 cool.
17 </pre>
18 </body>
19 </html>

```

Figure 3.1: Tags for paragraph, line breaks and preformatting

HTML: Basics

displayed in a slightly smaller font size, and so on down to the <H6> heading tag. You can use these tags to provide your page with a standard outline format. For example, the page heading might be displayed using the <H1> tag, a section heading using <H2> and a sub-section heading using <H3> and so on. Use HTML headings for headings only. Don't use headings to make text BIG or **bold**. Search engines use your headings to index the structure and content of your web pages. It is important to use headings to show the document structure. Browsers automatically add some empty space (a margin) before and after each heading.

Earlier we noted that Web browsers are HTML readers. Each browser is free to interpret HTML any way it likes. Consequently, a document read in one browser might look a little different to one read in another browser. Although the HTML standard states that <H1> tags should be as big as or bigger than <H2> tags, and <H2> tags should be as big as or bigger than <H3> tags and so on, one browser might display the <H3> tag with the same font size as the <H2> tag, while another browser might display it in a smaller font size. Hence the difference in displaying the same text. In practice, these implementation questions will become an issue when you are using more complex tags. For now you can ignore this problem while writing HTML.

The <hr> tag creates a horizontal line in an HTML page. The <hr> element can be used to separate content.

The HTML <meta> element is also meta data. It can be used to define the character set, and other information about the HTML document. Other meta elements that can be used are <style> and <link>.

Activity 5: Headings

In this Activity you will set up a page heading and sub-heading for the Web page begun in Activity 5 and use the HTML headings tags to implement it.

1. Load format.htm in MS-Notepad.
2. Within the <head> tags, add <meta charset="UTF-8">. It does not matter whether it is below or after the <title> tag.
3. Set up the page heading "Formatting text" and place the <H1> heading tags around it, in other words, <H1>Formatting text</H1>.
4. Reload format.html in your browser. You will notice that the effect of the <H1> tag is to display the text not only in an enlarged font size but also to include extra space above and below it. So you do not need a
 or <P> tag as well.
5. Return to Notepad and use the <H2> tag to create a sub-heading for the page, "Paragraphs and line breaks".
6. Add <hr> between 'This' and 'is'.
7. Reload the document in your browser to check the HTML and you should have an output like in Figure 3.2.

HTML Tip - How to View HTML Source

Have you ever seen a Web page and wondered "Hey! How did they do that?" To find out, right-click in the page and select "View Page Source" (in Chrome) or "View Source" (in IE), or similar in another browser. This will open a window containing the HTML code of the page.

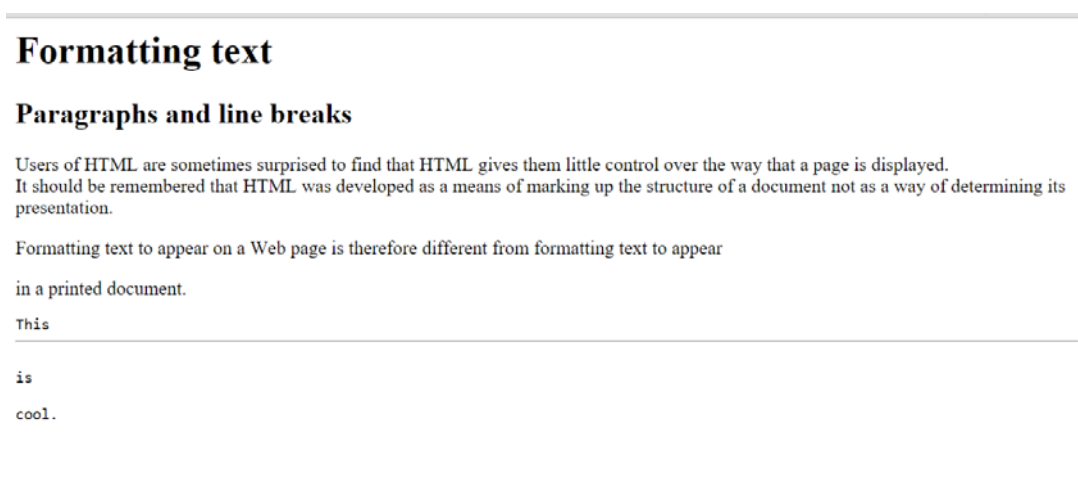


Figure 3.2: Using headings, horizontal rules and meta tags

3.3 Lists

- | | |
|-----------|------------|
| • Apple | 1. Apples |
| • Oranges | 2. Oranges |
| • Bananas | 3. Bananas |

The two examples above are lists. The list on the left uses bullets to mark the list elements, and is known as an **unordered list**. The list on the right uses numbers to mark the list elements and is known as an **ordered list**.

HTML lists consist of a list tag and list element tags.

In an **unordered list**, the list tag is `` and the list element tag is ``. Note that although the list element end tag `` was optional in previous versions of HTML, it no longer is. The list end tag `` is also not optional.

To create an **unordered list** as in the above example, use the following HTML.

```
<UL>
<LI>Apples</LI>
<LI>Oranges</LI>
<LI>Bananas</LI>
</UL>
```

Note that it is useful to indent the `` tags on the page to keep track of the level of indentation. To add more list elements, add extra list element tags `` `` containing the elements within the `` tags.

A **style** attribute can be added to an unordered list, to define the style of the marker:

| Style | Description |
|-------------------------------------|--|
| <code>list-style-type:disc</code> | The list items will be marked with bullets (default) |
| <code>list-style-type:circle</code> | The list items will be marked with circles |
| <code>list-style-type:square</code> | The list items will be marked with squares |
| <code>list-style-type:none</code> | The list items will not be marked |

For example the code below would append square bullets to the list.

```
<UL style = "list-style-type:square">
<LI>Apples</LI>
<LI>Oranges</LI>
<LI>Bananas</LI>
</UL>
```

Ordered lists are specified almost exactly the same as unordered lists, only the `` tag is used instead of the `` tag. A **type** attribute can be added to an ordered list, to define the type of the marker:

HTML: Basics

| Type | Description |
|----------|--|
| type="1" | The list items will be numbered with numbers (default) |
| type="A" | The list items will be numbered with uppercase letters |
| type="a" | The list items will be numbered with lowercase letters |
| type="I" | The list items will be numbered with uppercase roman numbers |
| type="i" | The list items will be numbered with lowercase roman numbers |

For example,

```
<OL type = "i">
<LI>Apples</LI>
<LI>Oranges</LI>
<LI>Bananas</LI>
</OL>
```

The **description** list, is different: it has neither bullets nor numbers. The description list tag is `<DL>` and the list elements consist of a term and its definition. The term is marked by `<DT>` tags and the definition by `<DD>` tags. An example use definition lists is the glossary definition that appears below.

```
<DL>
<DT>HTML </DT>
<DD> Hypertext Markup Language; the format of Web documents </DD>
</DL>
```

Lists can be nested (lists inside lists). For example,

```
<OL type = "i">
<LI>Apples</LI>
<LI>Oranges</LI>
<LI>Bananas</LI>
  <ul>
    <li> small bananas </li>
    <li> big bananas </li>
  </ul>
</OL>
```

Activity 6: Lists

In this Activity you will create a series of lists to practise your HTML list-building skills.

1. Load format.html in Notepad.
2. Underneath the text, create three lists as follows:
 - a. List one should be a circled bulleted (i.e. unordered) list, using square bullets, giving the days of the week.
 - b. List two should be a numbered list of the months of the year. Make the numbers lowercase roman numerals.

HTML: Basics

- c. List three should be a definition list of the four seasons.
3. Save the file and view it in your Web browser to ensure that it displays as desired.
4. Reload format.html in Notepad and create a new bulleted list showing the four seasons. Within each season create a numbered sub list of the appropriate months of the year.
5. Save the file and load it in your Web browser to examine the docum

3.4 HTML Comments

Notes may be left in an HTML page to, for example, explain how or why something was done; these notes are often useful for other developers who will be working on the HTML document. These notes, called **comments**, are not displayed by the Web browser, and can only be seen in the HTML source file itself. Web developers make frequent use of HTML comments, and they can found in many Web pages (using the 'View Page Source' menu option).

An HTML comment is given as: `<!-- text in the comment -->` .

Activity 7: Comments

In this Activity you will use preformatted text and HTML comments.

1. Load format.html in Notepad.
2. Place an HTML comment before the lists you created.
3. Save the file and load it in a Web browser.

3.5 Anchors

To link to another file use the `link` tag.

The term URL is the location of the file to be linked to. It could be on a hard or floppy disk — as in `a:\filename.html` or `c:\my documents\week01\filename.html` — on the same Web server, or on another Web server, as in <http://www.fortunecity.com/username/filename.html>

Activity 8: Simple hypertext links

In this Activity you will create four new Web pages: `index.html`, `filetwo.html`, `filethree.html` and `filefour.html`. You will then link them together using relative URLs.

1. Open Notepad and type in the HTML code shown below. (You may find it easier to cut and paste the code from your Web browser into Notepad rather than enter it yourself.)

```
<HTML>
<HEAD>
<TITLE>File name</TITLE>
</HEAD>
<BODY>
<h2>File name</h2>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam
erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci
tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
consequat.
<P>
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse
molestie consequat, vel illum dolore eu feugiat nulla facilisis at
vero
eros et accumsan et iusto odio dignissim qui blandit praesent
```

HTML: Basics

```
luptatum zzril delenit augue dui dolore te feugait nulla facilisi.  
<P>  
<A HREF="index.html">Homepage</A><BR>  
<A HREF="filetwo.html">Filetwo</A> <BR>  
<A HREF="filethree.html">Filethree</A> <BR>  
<A HREF="filefour">Filefour</A> <BR></BODY>  
</HTML>
```

2. Save this file as **index.htm**. Save the file a further three times using the file names from the list above. Each time also revise file name in the HTML title and body.
3. You should now have four unique files that link to each other. Test these files in your browser by first opening index.html.

4. Now add the following in index.htm to create a hyperlink to the University of Cape Town website.

```
<P>  
<A HREF="http://www.uct.ac.za">University of Cape Town</A>
```

5. Save the file and test it in your browser.

3.5.1 Linking to Email Addresses & other Non-Web Links

The previous examples have used the HTTP protocol to inform the browser to load a Web page when you click a hyperlink. Various other protocols may be used. For example, to create a link to an email address use the 'mailto' protocol. Note that this depends on the user's email programme being correctly configured, and so may not always work. However, this feature is commonly used on the Web to contact the webmaster or to get more information from sites.

The following anchor tag creates a mail link:

```
<A HREF="mailto:username@domainname">Email user</A>
```

You can test this by providing your own email address. This was tested and works for a Gmail address.

A subject line for the email message can also be provided:

```
<A HREF="mailto:username@address.com?SUBJECT=e-mail from a friend">user</A>
```

Other protocols that can be used include: ftp://, news://, telnet:// and gopher://. These protocols often require other software besides the Web browser, and, as with emails, if the software is not correctly installed and configured the links will not work.

3.5.2 Linking to Sections within Documents

Anchor tags can be used to link to a specific location within an HTML file (even within the same HTML file).

Firstly, a location must be defined using the tag: `` with xxx being the location name. A link to the location is created using the tag `link`. If the location is in another document, its file name too must be included as well: `link`

Activity 9 - Linking to sections within a document

This Activity sets up links to sections within the documents created by Activity 8

1. Open Notepad and load **index.html**.
2. Copy the following text twice into the body of the file above the hyperlinks in order to create a long file. Rename section two to section three when copying it for the second time.

```
section two
```

```
<P>
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam  
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam
```

HTML: Basics

erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci
tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
consequat..

<P>

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse
molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero
eros et accumsan et iusto odio dignissim qui blandit praesent
luptatum zzril delenit augue dui dolore te feugait nulla facilisi.

<P>

3. Now define the location section two by amending the text in the following way.

```
<A NAME="section_two">section two</A>
```

4. Define the location section three in the same way. Save the file.
5. Save this file as filetwo.html, overwriting the previous file.
6. Re-open index.html and add the following hyperlinks to the top of the <body> section:
 - Section Two
 - Section Three
 - File Two: Section Two
 - File Two: Section Three
7. Ensure that the links work by reloading index.htm in your Web browser.

3.5.3 Targeting Windows

In modern Web browsers, anchor tags can specify target windows using the target window attribute.

```
<A HREF="URL" TARGET="New_Window"></A>
```

This specifies where the contents of a selected hyperlink should be displayed. It is commonly used with frames or multiple browser windows, allowing the link to be opened in a specified frame or a new browser window. Windows may have names defined for them, but the underscore should not be used for the first character of any target defined in your documents. Such names are reserved for four special target names:

| | |
|----------------------|--|
| <code>_blank</code> | The browser always loads a <code>target="_blank"</code> linked document in a newly opened, unnamed window. |
| <code>_self</code> | This target value is the default for all <code><A></code> tags that do not specify a target. It causes the target document to be loaded and displayed in the same frame window as the source document. |
| <code>_parent</code> | This one is useful for framed sites to create navigation links back to the parent window. |
| <code>_top</code> | <code>_top</code> forces a break out of a framed site, or to take over the browser window. That means, for example, that if a site has been linked to from within a framed site, clicking on the link only brings up the site inside the frame. The <code>_top</code> target attribute forces the link to take over the entire browser window. |

Activity 10 - Targeting windows

This Activity allows you to try the different target attributes. You will see how they behave by adding them to one or more of the files set up by Activities 9 and 10. For instance, create a link that loads the UCT site in a new browser window.

3.5.4 Link Appearance

It is possible to change the colour of a hyperlink to match the colour scheme of the Web page. The underlining of a hyperlink can also be modified or removed.

To Do

Read about affecting the appearance of links in your textbooks.

Activity 11: Changing the appearance of links

In this Activity you will set up a colour scheme for a Web page by using the **BGCOLOR**, **TEXT**, **LINK**, **VLINK** and **ALINK** attributes of the **BODY** tag.

1. Open Notepad and load **index.html**. Create the following colour schemes for the page using either hexadecimal values or colour names (see your textbooks):
 - a. white background, black text and red hyperlinks.
 - b. black background, white text and cyan hyperlinks.
 - c. your own colour scheme.
2. View the pages in the browser.

3.6 Multimedia

3.6.1 Graphics

In-line images give the Web much of its visual appeal. They can also cause people using a slow Internet connection to not visit the site; so unless large in-line images are vital, they should either be avoided or alternate pages not using them provided. Small icons, however, have almost negligible transfer cost and can greatly enhance the appearance of your pages.

Use GIF or JPG files for graphics as appropriate:

- Scanned images look better as JPG files. The JPG file format uses variable compression to make the file size smaller, but too much compression causes the picture to lose detail. As a guideline, use a JPG compression level between 60 and 75 percent. JPG is always 24 bit colour or 8 bit grey scale (such as in a black and white photograph).
- GIF is useful for special effects such as transparent artwork (background transparency) and animation. It is 8 bit.

Another format called PNG combines the features JPG and GIF.

The **** tag is used to embed an image into a Web document:

```
<IMG SRC="URL" ALT="Alternate Text">
```

The URL is the location of the image file. The ALT attribute specifies text to be displayed if the browser does not display the image.

In-line images are often placed inside anchors. The HTML code to create an image as a hyperlink is:

```
<A HREF="URL"><IMG SRC="URL" ALT="text"></A>
```

When an in-line image is used as a hyperlink, a border is placed around the image. This can be removed using **BORDER=0**. Similarly, a border, of any size, can be placed around a graphic using **BORDER=x**, where **x** is the width of the border in pixels.

HTML: Basics

Make use of the WIDTH and HEIGHT attributes in the IMG tag to specify the width and height of the image. This allows the browser to layout the page before all the graphics have finished downloading. The width and height can be specified either in pixels or as a percentage:

```
<IMG SRC="URL" WIDTH=x HEIGHT=x> (in pixels)
<IMG SRC="URL" WIDTH=x% HEIGHT=x%> (as a percentage)
```

It is also possible to specify how the graphic is to be placed on the page in relation to the rest of the text. This is done using the ALIGN attribute. By default the bottom of the image is aligned with the bottom of the text — this can be changed using ALIGN=left|right|top|middle|bottom. To include a large graphic that will take a long time to load, first load another (smaller) image while the second larger one loads using LOWSRC="URL" to specify the location of the initial image.

If not specified, the browser expects to find the image in the same folder as the web page. However, it is common to store images in a sub-folder. You must then include the folder name in the src attribute. For example,

```

```

Activity 12: Graphics

In this Activity you will use the IMG tag and its attributes to create a Web page that includes graphic elements.

1. Open Notepad and begin a new HTML document by entering the main structural tags.
2. Save this file as **image.html**.
3. Now find an image or icon to use in your page. You can save one from any Web page by right-clicking the image and selecting Save As. Save the image in the same directory as **image.html**.
4. This image is going to be embed into the document. Save the file after each instruction and view it in your browser to see the affect.
5. Embed the graphic image in the document using the tag.
6. Include some alternative text to be used
7. If you can find out the width and height dimensions of the image, include these, otherwise try resizing the image as .
8. Turn the image into a hyperlink by linking it to a page of your choice:

9. Turn the border that appears around the image off.
.
10. Enter a short paragraph of text.
11. Align the image so that appears on the right side of the Web page.

3.6.2 Objects

This section described how to incorporate other objects objects - such as multimedia and Java applets - into your Web pages:

- The <EMBED SRC="URL"> tag is used to embed media such as background sound. For example,
<EMBED SRC="music.avi" WIDTH=320 HEIGHT=200 autostart=true loop=3> inserts an AVI movie object into a page, scales it to 320 x 200, starts playing the video, and loops it three times.
- Java applets can be embedded using the <APPLET> tag. For example,
<APPLET CODE=clock.class
codebase=http://www.server.com/classes/ height=100
width=100> </ APPLET> loads a Java applet from a remote Web server.

The <OBJECT> tag is used by multimedia software publishers like Macromedia and Digital Workshop (among others) to embed their programmes into Web browsers. Originally used by Microsoft for its Active-X technology, Netscape has adapted it to some degree. You probably will not be writing code using this tag. Programmes, like Flash and Director, generate the code for you to copy and paste into your HTML files. Some programmes, like Dreamweaver, which is a Web authoring tool, produce sophisticated code snippets based upon the designer's input parameters.

The following is an example that Macromedia Director produced for a Shockwave movie. Notice how the EMBED tag is inside the OBJECT tag.

HTML: Basics

```
<object classid="clsid:166B1BCA-3F9C-11CF-8075-444553540000"  
codebase="http://download.macromedia.com/pub/shockwave/cabs/director/sw.cab  
#version=7,0,0,0" width="150" height="100">  
  
<param name="src" value="welcome.dcr">  
<embed src="welcome.dcr"  
pluginspage="http://www.macromedia.com/shockwave/download/" width="150"  
height="100">  
  
</embed>  
</object>
```

3.6.3 ImageMap

An imagemap is a graphic that has certain areas on it defined as hyperlinks. These areas can do whatever normal HTML links do — email, ftp, gopher, etc. Two very popular uses for image maps are navigation bars and thumbnail sheets.

There are two types of imagemaps: **client-side** imagemaps and **server-side** imagemaps. A **server-side** imagemap requires the imagemap information to be saved within a separate file stored on a server and accessed through a CGI script. This type of imagemap is far more complicated to set up, and is not supported by all servers. Server-side imagemap behaviour varies from system to system, even among different systems using the same server. A server-side imagemap shows mouse co-ordinates at the bottom of the screen.

A **client-side** imagemap (CSIM) requires the imagemap information to be stored within the HTML document. A client-side image map shows the actual URL in the status bar message at the bottom of the browser window. Client-side image maps store the hyperlink information in the HTML document, not in a separate map file as do server-side image maps.

When the user clicks a hotspot in the image, the associated URL is sent directly to the server. This makes client-side image maps faster than server-side image maps because the server does not need to interpret where the user clicked. Client-side image maps are supported by all modern browsers.

3.7 Writing Good HTML

This final section is concerned with the production of good HTML, as well as with the importance of testing your Web documents before going 'live'.

There are a number of online validation services that check the validity of a Web page. The World Wide Web Consortium's HTML Validator, for example, checks HTML documents for compliance with W3C HTML Recommendations and other HTML standards.

If there are mistakes in the HTML document, the validator will list the problems — but if it is correct, it provides a message similar to:

```
Valid HTML 4.0  
No errors found!  
Congratulations, this document validates as HTML 4.0 Transitional!
```

Certain HTML editors or authoring tools have built-in HTML checkers or validators; these are sometimes useful, but are not always accurate. Sometimes they notice 'errors' you have intentionally entered, but mostly they can be of great help, particularly when producing complex Web pages. The more sophisticated tools are able to adjust to the level of the target browser, as well as fine tune options such as listing missing ALT attributes inside IMG tags.

Activity 13: Validating HTML (Optional)

This Activity examines online HTML Validation Services

HTML: Basics

1. Look at the at W3C's Validator [<http://validator.w3.org>] and submit two or three of your Web pages. What happens?
Note that you need to give the validators an URL. In other words, your document must exist somewhere on the WWW for it to be validated. You could search for a free hosting service, or place the page on the personal server space most ISPs provide for their clients. Obtaining an URL is not trivial, but it will allow you to validate your code.
2. Try at least one other online validation service and compare the results. Which is the easiest to use? What features of the services do you like or dislike?
3. Discuss your results with the other students in the course in the Discussion Forum.

3.8 Discussion and Answers

3.8.1 Discussion Topics

At an early stage of learning HTML it is often the little problems that provoke discussion. So you may want to spend some time online discussing the problems you have encountered with your fellow students or tutors.

In particular, you might want to discuss the different format of HTML tags. Some (like bold) have end markers, some (like the line break) do not have end markers, and some (like anchors and images) take arguments. Why do different tags have different formats? What other tags might there be? Is there anything that cannot be done in a tag?

If you managed to validate your code discuss the usability of the Validating Services you investigated as part of Activity 13. In particular, you should:

1. Compare the results obtained from the different online validation services.
2. Discuss their ease of use and their functionality.
3. And finally, you should take an online vote on your preferred online validation service.

3.8.2 Discussion of Activity 2

Your HTML document should look something like the one below:

```
<html>
<head>
<title> My first Web page </title>
</head>

<body>
Your name
<i> csc5003 </i>
<b>Internet Commerce</b>

</body>
</html>
```

3.8.3 Discussion of Activity 4 and 5

Your solution should look like what is shown in Figure 3.1.

3.8.4 Discussion of Activity 6 - Lists

The code to create the lists should look something like the one below. The HTML of particular interest is shown in bold.

```
<UL TYPE=square>
```

HTML: Basics

```
<LI>Monday</LI>
<LI>Tuesday</LI>
<LI>Wednesday</LI>
<LI>Thursday</LI>
<LI>Friday</LI>
<LI>Saturday</LI>
<LI>Sunday</LI>
<UL>
<OL TYPE=i>
<LI>January</LI>
<LI>February</LI>
<LI>March</LI>
<LI>April</LI>
<LI>May</LI>
<LI>June</LI>
<LI>July</LI>
<LI>August</LI>
<LI>September</LI>
<LI>October</LI>
<LI>November</LI>
<LI>December</LI>
</OL>
<DL>
<DT>Spring
<DD>First season of the year: March - May
<DT>Summer
<DD>Second season of the year: June - August
<DT>Autumn
<DD>Third season of the year: September - November
<DT>Winter
<DD>Fourth season of the year: December - February
</DL>
<UL>
<LI>Spring</LI>
<OL START=3>
<LI>March</LI>
<LI>April</LI>
<LI>May</LI>
</OL>
<LI>Summer</LI>
<OL START=6>
<LI>June</LI>
<LI>July</LI>
<LI>August</LI>
</OL>
<LI>Autumn</LI>
<OL START=9>
<LI>September</LI>
<LI>October</LI>
<LI>November</LI>
</OL>
<LI>Winter</LI>
<OL START=12>
<LI>December</LI>
<LI VALUE=1>January</LI>
<LI>February</LI>
</OL>
</UL>
```


3.8.5 Discussion of Activity 7 – Comments

You should include the following tag:

```
<!-- Creating lists -->
```

3.8.6 Discussion of Activity 9 – Linking to sections within a document

The link to the specified sections should look as below:

```
<P>  
<A HREF="#section_two">Section two</A><BR>  
<A HREF="#section_three">Section three</A> <BR>  
<A HREF="filetwo.html#section_two">Filetwo-Sec2</A> <BR>  
<A HREF="filetwo.html#section_three">Filetwo-Sec3</A> <BR>
```

3.8.7 Discussion of Activity 11 – Changing appearance of links

The code to generate the colour schemes is shown below: white background, black text and red hyperlinks

```
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#FF000000">  
or  
<BODY BGCOLOR="white" TEXT="black" LINK="red">  
<BODY BGCOLOR="#000000" TEXT="#FFFFFF" LINK="#00FFFF">  
or  
<BODY BGCOLOR="black" TEXT="white" LINK="cyan">
```