Chapter 1

H t m l

B a s i c s

Chapter Objectives

This chapter aims to address the following concepts:

- Text editors
- Programs in html
- Creating an html document
- Headings
- Horizontal rule
- Paragraphs and line breaks

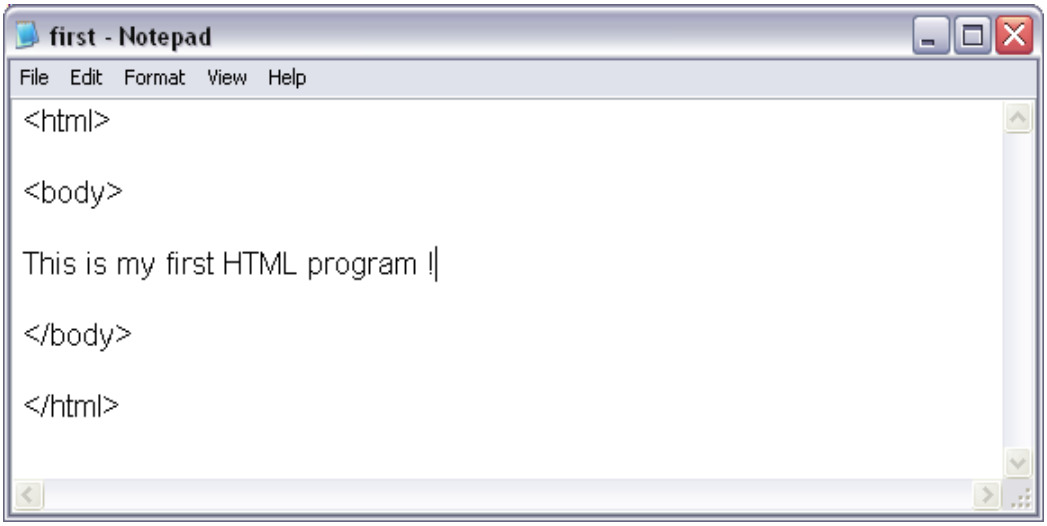
Introduction

HTML refers to Hypertext Mark-Up Language. This language enables you to create good web pages that incorporate both text and images. Web pages are more immune to viruses than other documents like Microsoft Office documents.

Text Editors

Text editors are platforms that allow you to input your commands (series of instructions) and interpret these commands accordingly. The most common text editor is the notepad. In this book, all commands that produce web pages are going to be typed in notepad.

Your first html program



There are labels that tell the text editor (Notepad in this case) what to do with the sequence of instructions supplied. The tags are identified by the < and > signs.

Saving an html document

After typing the notepad document you need to save the document in a format that is compatible across different platforms at the same time safe from viruses. This is the web format. You can save the document by following these steps:

1. Click File then Save As
2. In the Save As pop-up menu that appears, type the name of the file under File Name with an extension **.htm**.

The document can be saved as first.htm. The file extension converts the original html document into a web page displayed as an Internet Explorer or Mozilla Firefox document or as a format supported by other web browsers. If you forget this extension, the document would be saved as a notepad document. After saving the document in the specified location you then open the web document. This document will be using the default web browser on your computer. If you want to open it in a different web browser you right click the document and choose **"Open With"** then select the web browser that you want to use.

Copy the above program into notepad and save it as described above. Open the document and tell somebody the good news of programming!

If you want to alter the display of this web page, you just minimise the web page and go back to that notepad where your source code is displayed. I hope that you did not close it so far. If the notepad was still open, then do the following:

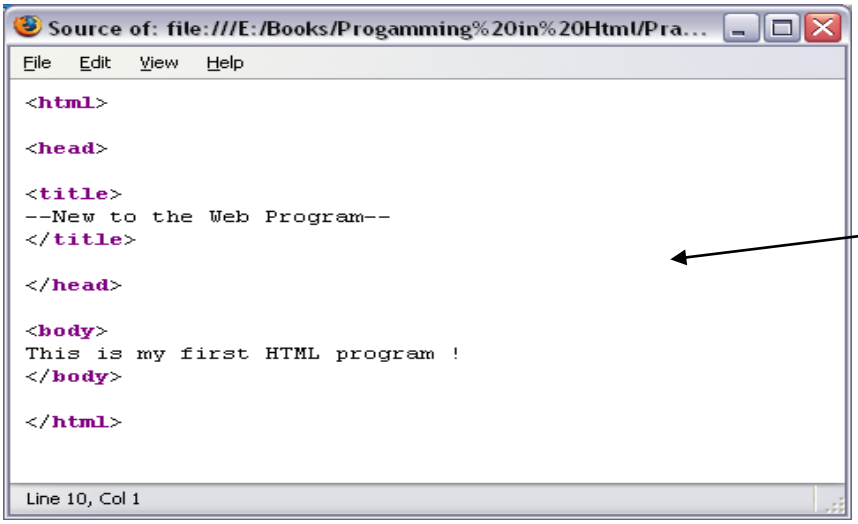
1. Type the additional commands (html) that produce the desired results.
2. Click File
3. Click Save. Don't click Save As just click Save to update the existing file
4. Maximise the previously minimised web page (first.htm).
5. Press F5 to refresh the display

The new contents of the page will then be displayed.

If you had closed the notepad and you want to update the webpage, do the following:

1. Open the existing webpage
2. Right click your mouse pointing inside the opened web document
3. If you are using Internet Explorer, click View Source; if you are using Mozilla Firefox then click View Page Source

   There are slight differences between the formats of the page source displayed. The page source is just a combination of instructions that make up the web page. Internet Explorer will take you directly to the notepad whilst Mozilla Firefox opens the page source code in web page format.



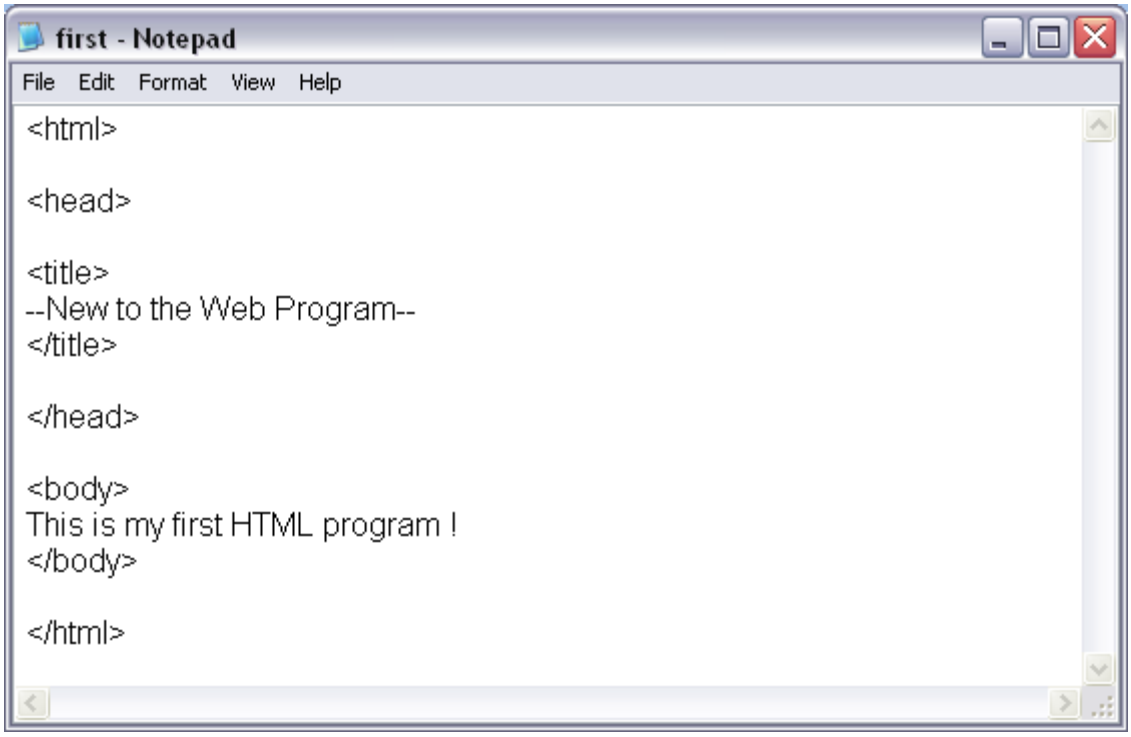
4. Copy the source code and paste it into notepad
5. Make the alterations you want to implement
6. Click **File** then **Save As**
7. Type the file name with the same name and location to that you previously saved. This should be the file you want to update. You will be asked whether you want to replace the existing file. Click **Yes**
8. Open the web page or refresh it if it is already open

Well, with these tips at hand, let's move forward more comfortably than before. The major reason behind creating websites is to display a company's details as well as its products. There is need for the site to presentable and user friendly. This forces the developer to be more artistic to produce alternative designs that satisfy internet browsers. Let us look at the html elements that make a web page.

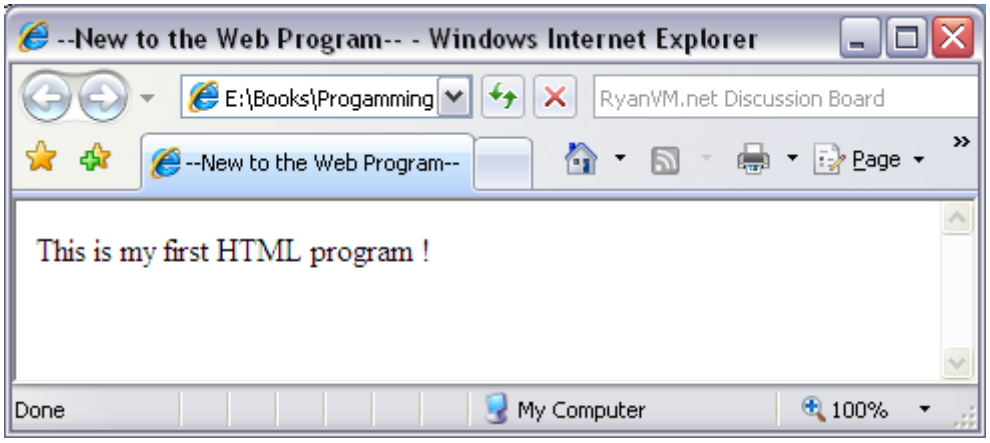
The <Head> tag

The html document is divided into two basic parts – the head and its body. The head consists of the title of the web page that is to be displayed on the upper most part of the web page. You place the <title> tag inside the <head> tag. The <title> tag then takes the contents of the title and displays them on the head of the document.

Example 1.2



Output of Example 1.2



Headings

Just like your natural language regulations where you are sometimes required to put headings on your document, there is also need to include headings in your web page. Headings improve the readability of the document since they take the eye of the reader. You need to decide the sizes of the headings as well and the number of the headings that you want to put on a web page.

The following table summarises the levels of headers as well as their font size equivalence.

Tag	Font Size (equivalence)
<h1>	24
<h2>	18
<h3>	14
<h4>	12

From the table above, you have noticed that the <h4> corresponds to size font size 12 which is bold. In most cases the headings will end at <h3> that is more than size 12 bold, but <h4> works also since the text has more emphasis that size 12 plain.

Example 1.2

```
<html>
<head>
<title>
Dealing With Headings
</title>
</head>
<body>
<h1>My first Heading
</h1>
<h2>My second Heading
</h2>
<h3>The Third Heading
</h3>
<h4>The forth heading - Corresponding to font size 12 plain!
</h4>
</body>
</html>
```

The above commands produce the following output.

Inserting Comments into HTML source code

Comments assist you in explaining the reason for a certain piece of code. The following commands will be interpreted as comments in html:

<! This is a comment >

When the web page is loaded, the comments are not displayed. They are used to assist the programmer to document the website. This is useful when other programmers want to customise the website.

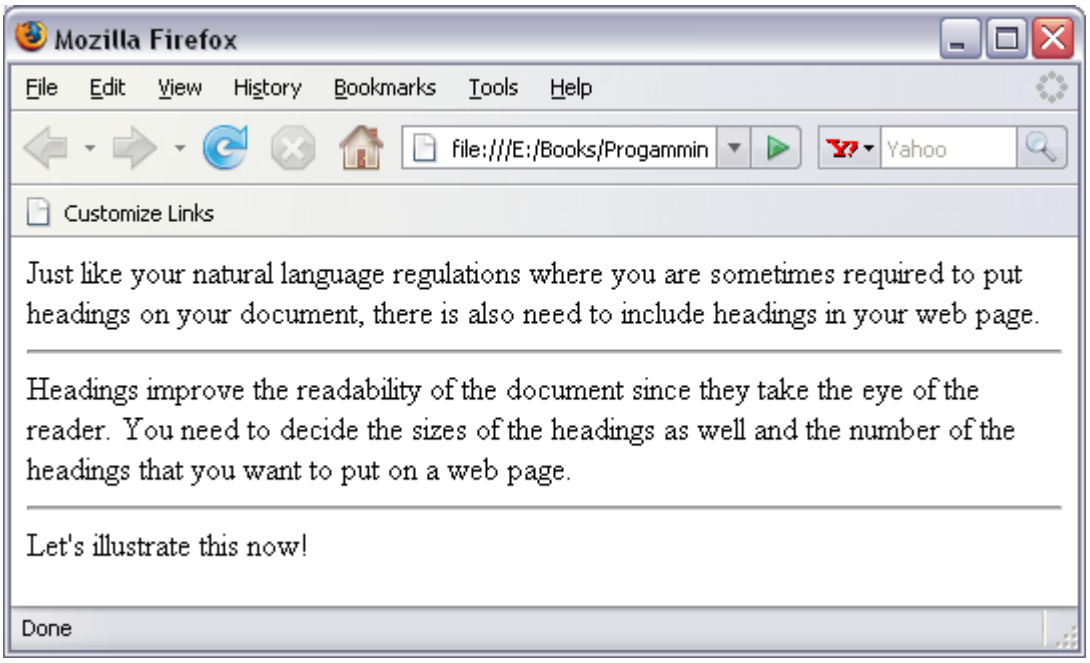
Horizontal Rule

The horizontal rule creates a line in your document. It separates the group of text as does the paragraph concept. The initial line is thin but can be altered to fit the size you want to display. The horizontal line size, width, alignment as well as the shade effect of the line can be developed to make the line presentable. The **<hr>** tag draws the line on the document.

Example 1.4

<html>
<body>
Just like your natural language regulations where you are sometimes required to put headings on your document, there is also need to include headings in your web page. **<hr>**Headings improve the readability of the document since they take the eye of the reader. You need to

decide the sizes of the headings as well and the number of the
headings that you want to put on a web page. **<hr>** Let's illustrate this now!
</body>
</html>



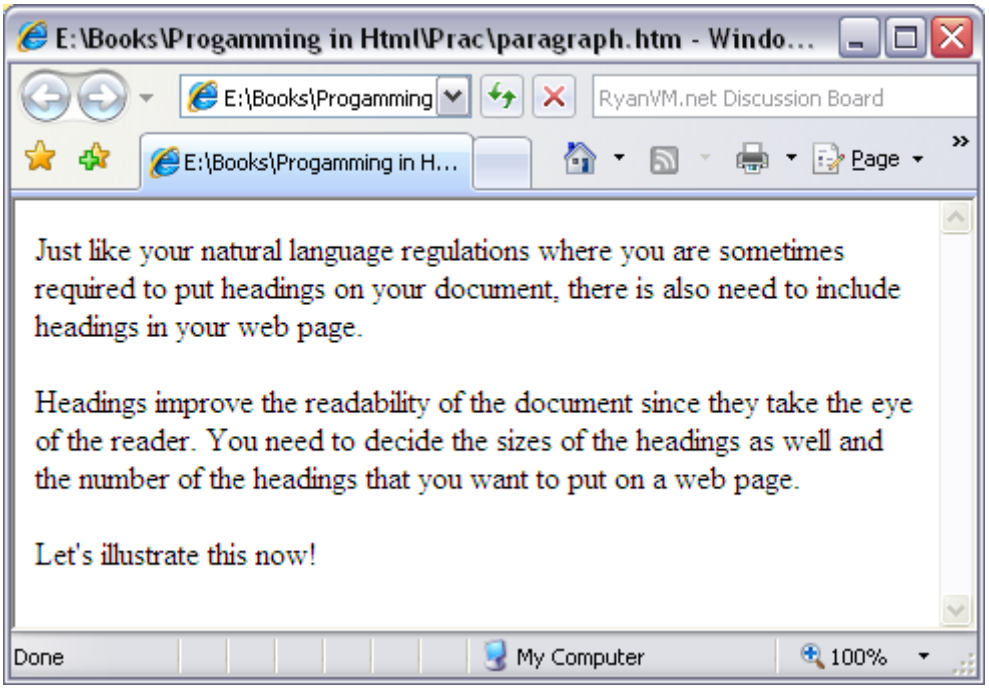
HTML paragraphs

The <p> tag starts a new paragraph. This tag creates a blank line between the previous word and the succeeding word. The following commands illustrate this.

Example 1.5

<html>
<body>
Just like your natural language regulations where you are sometimes
required to put headings on your document, there is also need to
include headings in your web page. **<p>**Headings improve the readability
of the document since they take the eye of the reader. You need to
decide the sizes of the headings as well and the number of the
headings that you want to put on a web page. **<p>** Let's illustrate this now!
</body>
</html>

The above commands produce the following form:



Note that there is a blank line between the first paragraph and the second one. You can illustrate this concept by including the headings in the document so that it becomes more meaningful.

The use of line breaks

Line breaks put text on the next line. In other words, the line break tag is a new line tag similar to the new line characters in C and C++. These two work in the same way although the way they are written is different. The **
** tag creates the new line character.

Example 1.6

<html>

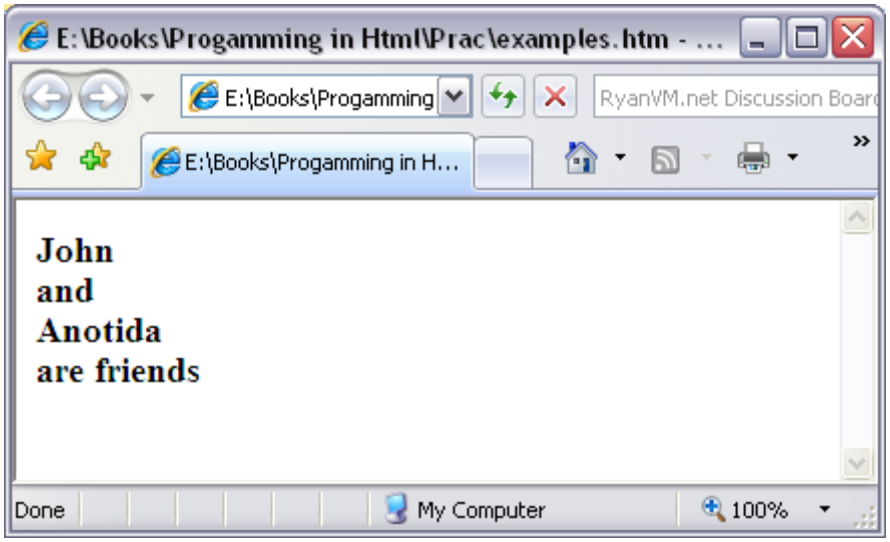
<body>

John **
** and **
** Anotida **
** are friends

</body>

</html>

The commands above split the statement into three different lines as displayed by the web page below:



Chapter 2

Formatting Text in Html

Chapter Objectives

This chapter aims to address the following concepts:

- Italic text
- Bold text
- Underlined text
- Font size
- Text alignment
- Preformatted text
- Superscript
- Subscript
- Page colours
- Text in motion
- Inserted and deleted text

Introduction

Text formatting techniques apply to the size, type as well as the strength (bold, italic or plain) of the text. As you learnt in Chapter 1, headings can be used to display text sizes. To be more artistic, you can use the font attribute tag. The following sections describe the font attribute techniques.

Underlining text

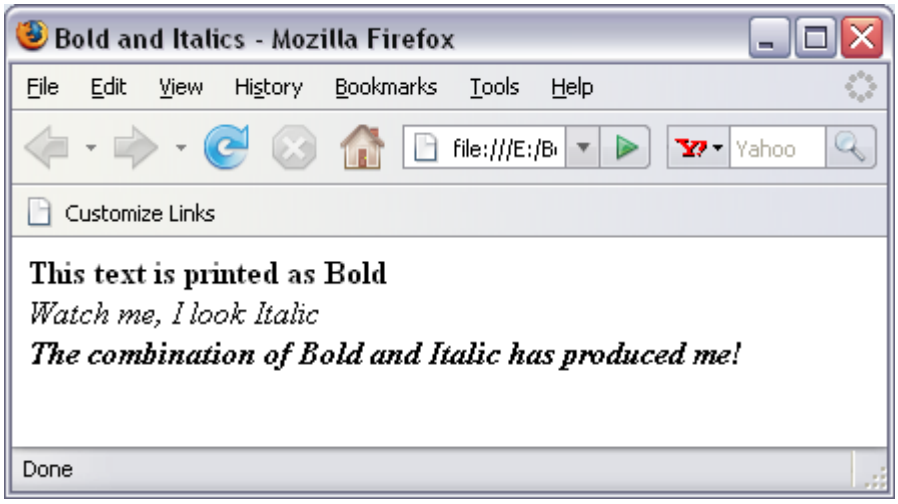
You can underline text by using the **<u>** tag.

Italic

The **<i>** tag makes the text appear italic. This increases emphasis on the document displayed. The tag can be combined with the bold, <b> as well. The italic tag is equivalent to the **** tag that is used to increase emphasis.

Bold

Bold emphasises text in the same way as headings do. The **<b>** tag formats text to be bold. I think by now you are familiar with headings. The same effect happens here; the only difference here is that the <b> tag makes the text bold without changing its size. The <b> tag produces the same effect with the tag . The following form was produced by the <i> and <b> formatting tags.



The following commands produce a web page similar to the one above.

<html>

<head>

```
<title>Bold and Italics</title>
</head>
<body>
<b>This text is printed as Bold<br>
</b>
<i>Watch me, I look Italic<br>
</i>
<b><i>The combination of Bold and Italic has produced me!
</b></i>
</body>
</html>
```

Font size

The size of the font can be specified by the **** tag. This will be accompanied by the **** closing tag to demarcate the end of the affected area. If you do not include the closing tag then the whole document will be affected by the same font size. You can use different font sizes on different paragraphs or even different fonts for one word. The following example illustrates this concept.

Example 2.5

```
<html>
<body>
<font size = 5> This is size 5 Font
</body>
</html>
```

You can produce different font sizes for each letter in the same statement. This can be illustrated by the following example:

```
<! --Displaying the word "Differences" with different font sizes>
<html>
```

<body>

<h2> Different font sizes in one word </h2>

 D

 i

 f

 f

 e

 r

 e

 n

 c

 e

 s

</body>

</html>

Note that each has a closing tag. If you do not include this closing tag, the whole document will use the font size previously specified. This can be a situation when you want to use specified fonts at some levels of the documents but retaining the default font on the rest of the document. The above program is not affected since the whole word has predefined font sizes.

The above commands can be written in the following way:

<! --Displaying the word "Differences" with different font sizes>

<html> <body> <h2> Different font sizes in one word </h2> D

 i   f   f

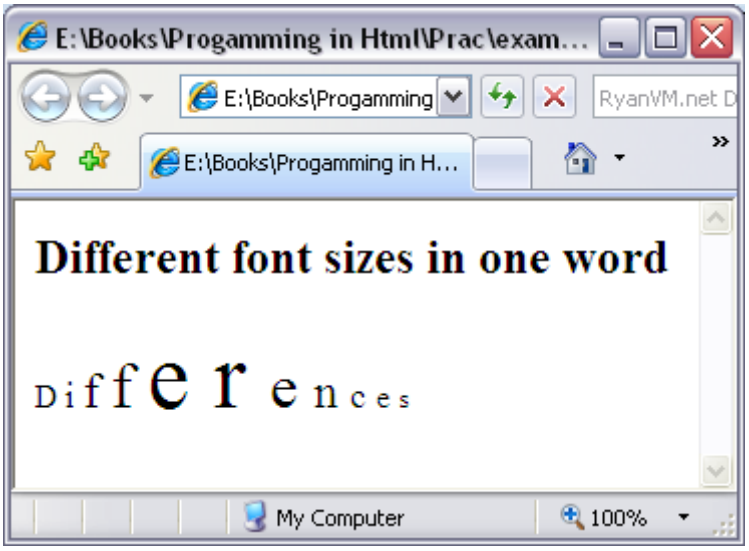
 e   r   e

 n   c   e

 s </body></html>

The biggest challenge of writing the commands in this way is the readability of the commands to the developer. Experienced developers prefer this method since it does not take up a lot of space. You do not need to scroll up and down many times since your code will be intact. Junior web developers might prefer to avoid confusing arrangements and the longer program. I will use the long programs so that I do not confuse my junior friends.

The commands above produce the following form:



You can also print the fonts (1-7) in a list using the following commands.

<! --Numbers 1 to 7 in different font sizes>

<html>

<body>

 Font sizes 1 up to 7<hr>

 1

 2

 3

 4

 5

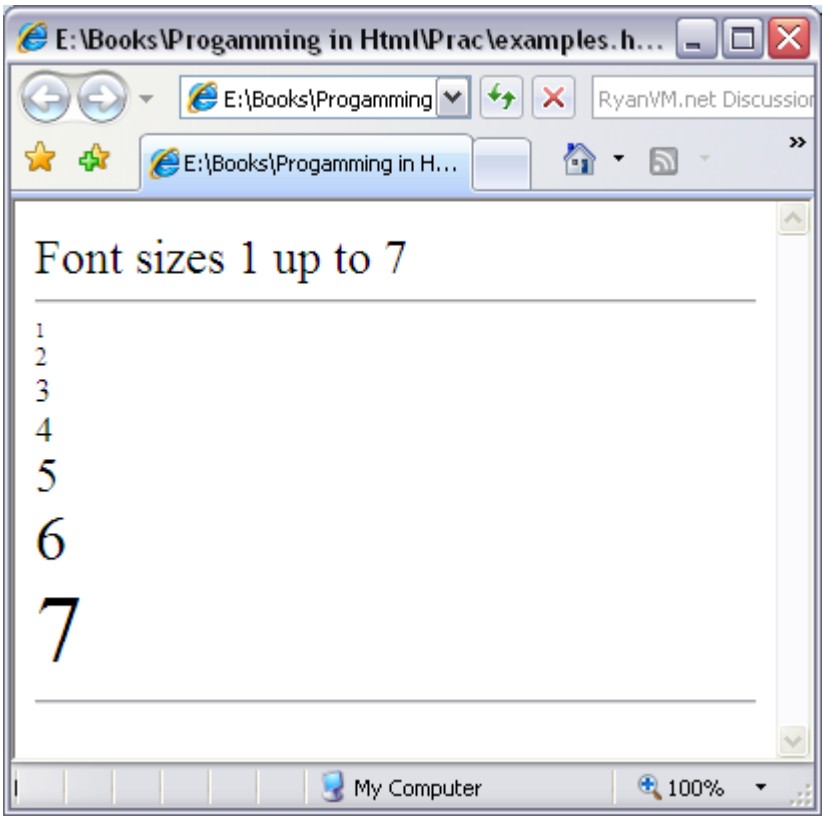
 6

 7 <hr>

</body>

</html>

These commands produce this from:



The minimum font size supported at the time of writing is 1 and the maximum size is 7. If you use any size more than 7, the web browser will display the maximum size which is 7. Choosing the font size depends on the requirements of the text that is its importance to the browser. The most important communications are displayed in fonts that do not irritate people. Choosing font size 1 is usually restricted by its size; it's too small so sometimes not used.

Text Alignment

The text displayed can be aligned to the left (the default alignment), to the right or at the centre of the document. This alignment is specified by the <align> tag which works hand in hand with a block of text not with individual text elements. You will notice that simple **<align = "right">** may not work while **<p align = "right">** works well. This is because the <p> tag identifies the block of text that you want to align. The termination of the <p align> tag is </p align>. It is a common mistake to forget the p and write </align>. The </align> tag will not terminate the </p align>.

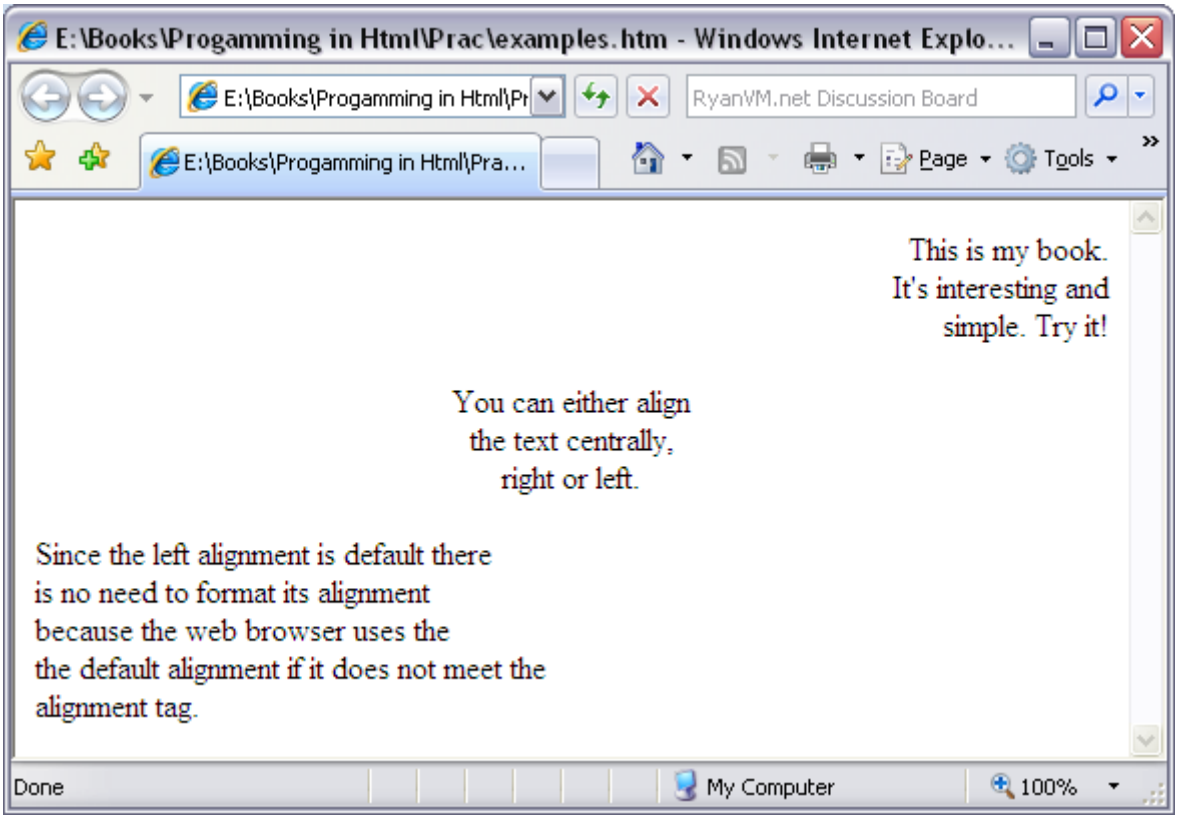
Example 2.4

```
<html>
<body>
<p align = "right">
This is my book. <br> It's interesting and <br> simple. Try it!
</p align>
<p align = "center">
You can either align <br> the text centrally, <br>right or left.
</p align>
Since the left alignment is default there <br>is no need to format
 its alignment <br> because the web browser uses the <br> the default
alignment  if it does not meet the <br> alignment tag.
</body>
</html>
```

NOTE:

The spelling for centralising the text was center, the US spelling not centre. Take note of this spelling, if you use centre, the web page will give you the default alignment since the **<centre> tag is not recognised.**

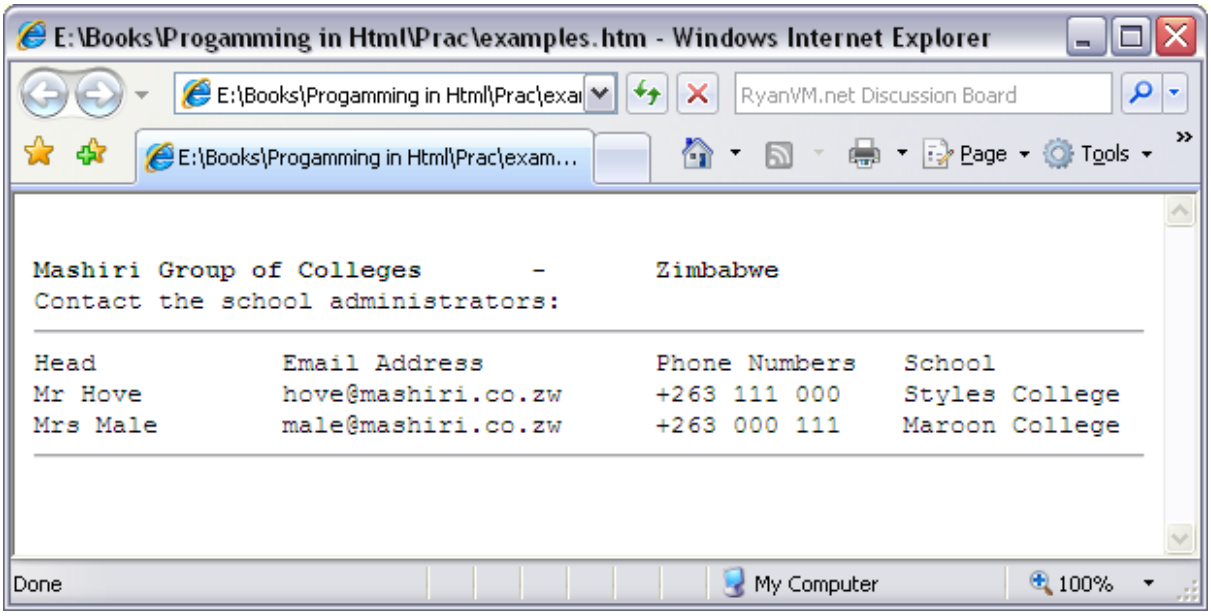
Try this practically and look at the form that you produce. If it does not look like the form below, check your commands



Pre-formatting Text

Sometimes you want to define the appearance of text to be presented in a columnar form or in a way that looks like a table. This can be done without using a table. The table methods are explained in Chapter 3. As of now, we can create a form that is similar to the one created by tables. The preformatted text is specified by the **<pre> ** tag. You develop the alignment and spacing that you want between the texts. The web browser will read the spaces as they are so that the display is maintained. This is different from situations where the **<pre> </pre>** tag is not used. In the absence of the   tag, the web browser discards any succeeding white space, it just accept one white space between words. If you use double spacing between words, the web browser will maintain this if you used the   tag.

Using preformatted text, develop commands that produce the following form.



Superscript

If you want to write numbers as well as their powers e.g. 2^4 you can use the superscript tag. The superscript is inserted by the **** tag. Writing the equation **$4 = 2^2$** can be done by the following commands:

```
<html>
<head>
<title>
Dealing with Superscripts
</head>
</title>
<body>
<h2>
4 = 2 <sup> 2 </sup>
</h2>
</body>
</html>
```

Subscript

The **** tags produce subscripted text. This situation can be met when you want to print numbers to their given bases e.g. 1011_2 or 13_8. This can also be the situation when you want to blog and inform readers of a site that the included text requires reference or citation. The citation has two options; it can be superscript or subscript.

```
<html>
<head>
<body>
<h2>
These are my numbers <br>
9 <sub> 10 </sub> <br>
14<sub> 8 </sub> <br>
1101<sub>2 </sub>
</h2>
</body>
</html>
```

Try these commands and comment their output:

```
9 <sub> 10
14<sub> 8
1101<sub>2
```

Note: Do not include the </sub> tag.

What do the above commands produce?

Dealing with Colours

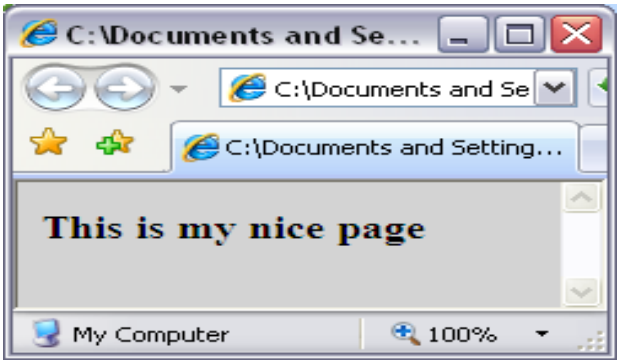
We have so far assumed that the text to be displayed is black and its background is white. We can now create different colours for web pages as well as their text. Let's look at the background as well as the text colours.

- **Page background**

The background of the page can be set by the **<body bgcolor = >** tag. This tag can be combined with text specification tag. Let's look at the page background example.

```
<html>
<head>
<body bgcolor = lightgrey>
<h3>This is my nice page</h3>
</body>
</html>
```

The above commands produce the following output



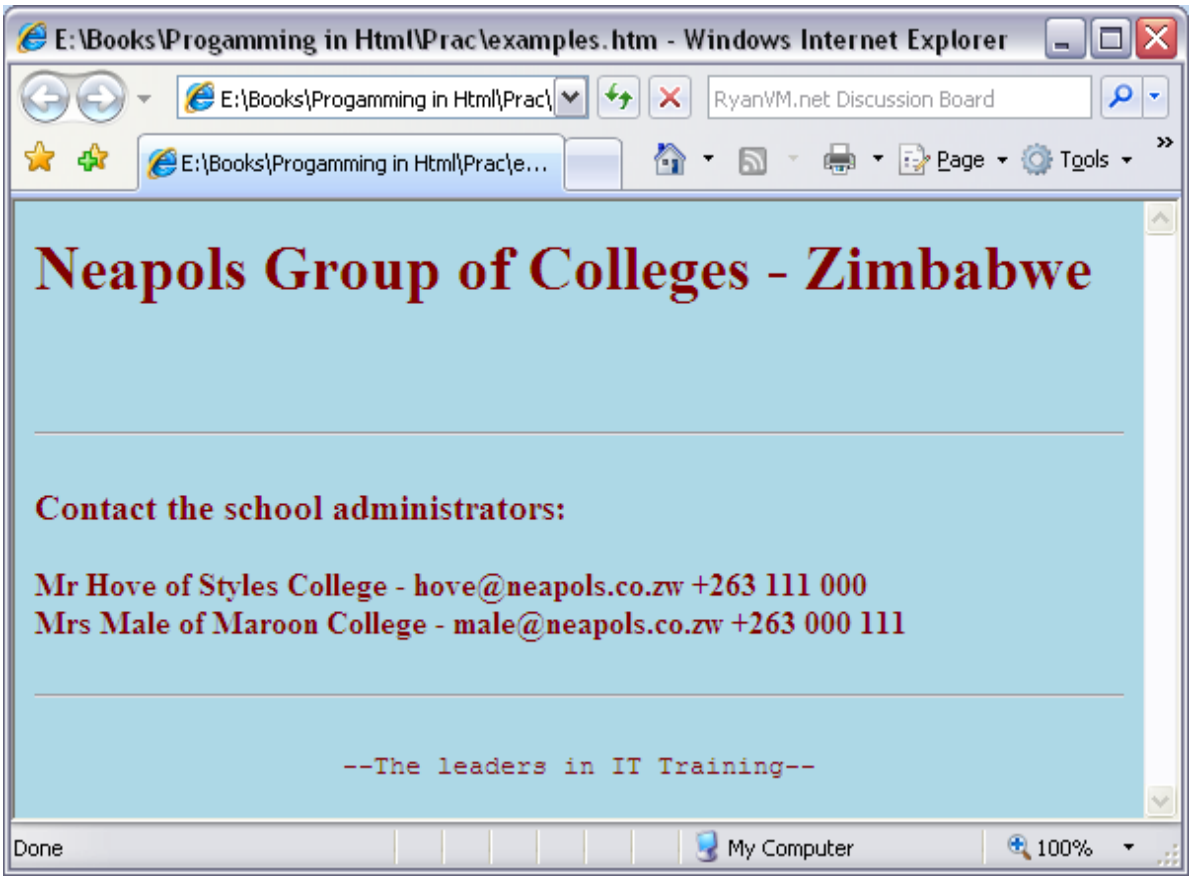
- **Text background**

The colour of text can be set by the **<body text = > tag**. The following code creates a web page with colour coded text as well as background.

```
<html>
<body bgcolor = lightblue text = maroon>
<h1>Neapols Group of Colleges - Zimbabwe</b></h1>
<br><br>
<hr><h3>Contact the school administrators:<br></h3>
<h4>Mr Hove of Styles College - hove@neapols.co.zw +263 111 000
```

```
<br>Mrs Male of Maroon College - male@neapols.co.zw +263 000 111
</h4>
<hr>
<p align = "center">
<tt>--The leaders in IT Training--
</tt>
</align>
</body>
</html>
```

The above produces the following form:



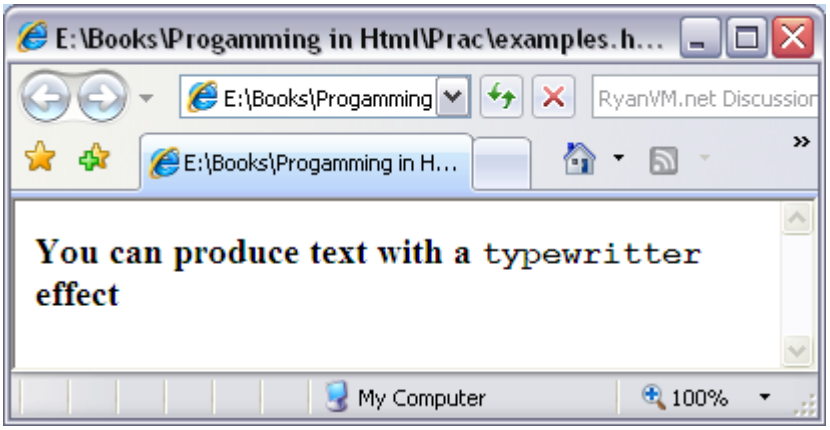
The typewriter effect

The typewriter effect is produced by the **<tt> </tt>** tags. The tag produces font that is equivalent to the text produced by typewriters. The font is

equivalent to Courier. This font has a benefit of taking the reader's eye. The following commands illustrate this concept.

```
<html>
<head>
<body>
<h3>
You can produce text with a <tt>typewritter</tt> effect
</h3>
</body>
</html>
```

The commands produce the following form:



Motion

Text and images can be moved on the web page. This is done by the <marquee> </marquee> </marquee> tags. The default movement is right to left. The following example illustrates this concept.

```
<html>
<head>
<head>
<title>
Moving Text
</title>
</head>
```

```
<body>
<marquee>
<h3>
The Big Sale !! <hr>
</h3>
</marquee>
</body>
</html>
```

Inserted text

The inserted text will be indicated by an underline. This is a common situation when you are marketing products. Let's say you are discounting your products or the products are on sale. There is need to include the original price as well as the new price. The new price will be the inserted text here. The **<ins> </ins>** tags identify the inserted text. Its effect is similar to the underlining tag **<u>**.

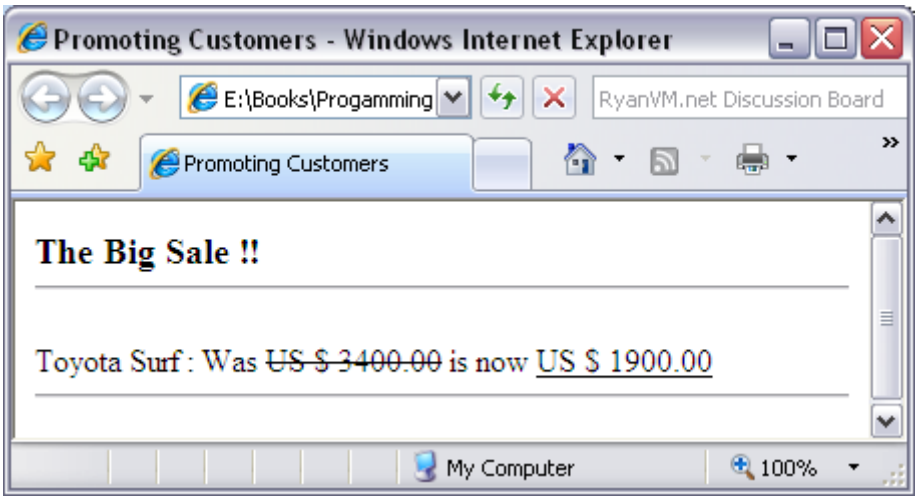
Deleted text

As explained above, we can change prices of products either to clear old stock or to promote customers. There is need to communicate the original price, cancel that price and write the new price. The old price stroked through e.g. ~~$456.99,~~ and then the new price will be underlined. The **<del> </del>** tags produce the strikethrough effect which marks the deleted text. The following example illustrates how a seller of products can market their products at reduced prices.

```
<html>
<head>
<head>
<title>
Promoting Customers
</title>
</head>
```

```
<body>
<h3>
The Big Sale !! <hr>
</h3>
Toyota Surf : Was
<del> US $ 3400.00</del> is now
<ins> US $ 1900.00 </ins> <hr>
</body>
</html>
```

The above commands produces the following form.



This concept can be further explained after covering tables in Chapter 3.

Chapter 3

Tables in Html

Chapter Objectives

This chapter aims to address the following concepts in HTML:

- Benefits of tables
- Basic table concept
- Table headers
- Table borders
- Aligning text in tables
- Setting table background colours
- Cell padding
- Cell spacing
- Creating blank cells in tables
- Spanning in tables

Benefits of tables

Tables are used to improve presentation of information on the web page. The information can be drawn in cells of the table in a way easier than using predefined text, <pre> method. Tables might be used with visible borders or the borders might not be visible. Invisible lines will produce data that is columnar but without visible margins. The default position of the table is set to be on the upper left corner of the page and columns without a visible margin. You can use images, text and other forms of graphics inside the table. All commands in html can be run inside a cell of a table. The major benefit of tables comes with its ease of use in presentation.

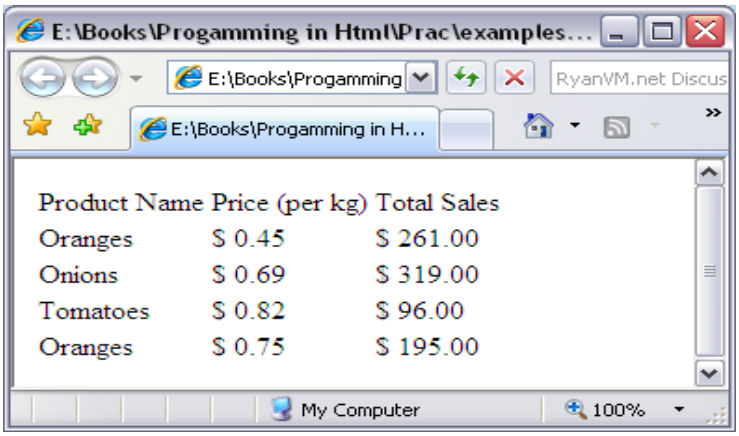
Basic table concept

The following commands produce a simple table that displays products of a company, their unit price as well as the value of their total sales.

```
<html>

<body>

<table>

<! -- the first table row >

<tr>

        <td>Product  Name </td>

        <td>Price (per kg) </td>

        <td>Total Sales </td>

</tr>

<! -- the second table row >

<tr>

        <td>Oranges </td>

        <td>$ 0.45 </td>

        <td>$ 261.00 </td>
```

```
</tr>
<! -- the third table row >
<tr>
        <td>Onions  </td>
        <td>$ 0.69 </td>
        <td>$ 319.00 </td>
</tr>
<! -- the forth table row >
<tr>
        <td>Tomatoes  </td>
        <td>$ 0.82 </td>
        <td>$ 96.00 </td>
</tr>
<! -- the fifth table row >
<tr>
        <td>Oranges </td>
        <td>$ 0.75 </td>
        <td>$ 195.00 </td>
</tr>
</table>
</body>
</html>
```

The commands produce the following web page.



Note that I have used a lot of new tags here. Let me explain these tags here. The table is created by the **<table> </table>** tags. You can now specify the contents of the cells in the <table> </table> tags. The contents of the cells are identified by rows and columns. Therefore, the **<tr> </tr>** tags identify the row in the table, for example row 1. Inside the row, you can now define the data for that cell. This is identified by the **<td> </td>**. If you do not close the tag <td>, then all the succeeding data will be treated as one cell.

Creating tables headers

The example above produced a table that has equal text size and weight. We can make one or more cells appear more of a heading so that emphasis on the text is produced. The <th> </th> tags create the table's headers.

The table might be more meaningful if it looks like this one.

<html>

<body>

<table>

<tr><**! The first table row** >

<th>Product Name </td>

<th>$ Unit Price </td>

<th>Total Sales </td>

</tr>

<tr><**! The second table row >**

<th>Oranges </td>

<td>$ 0.45 </td>

<td>$ 261.00 </td>

</tr>

<tr><**! The third table row >**

<th>Onions </td>

<td>$ 0.69 </td>

<td>$ 319.00 </td>

</tr>

</table>

</body>

</html>

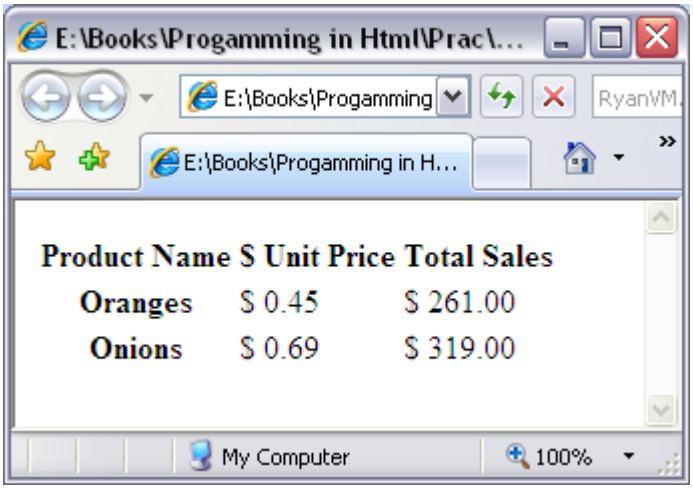
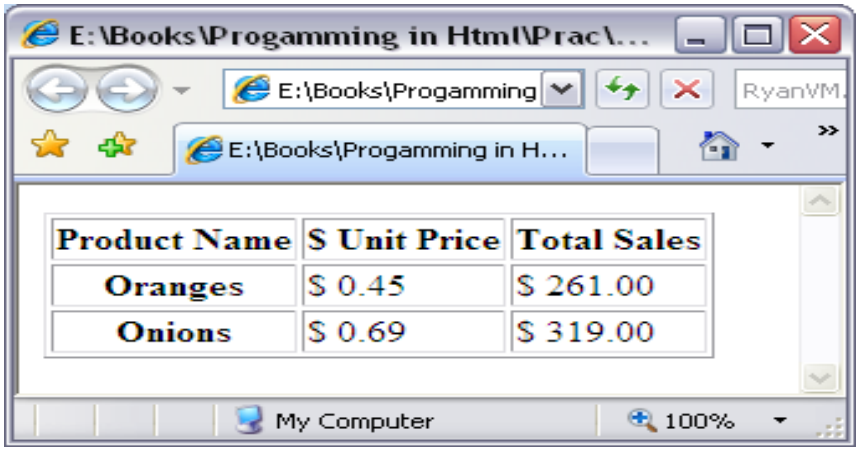


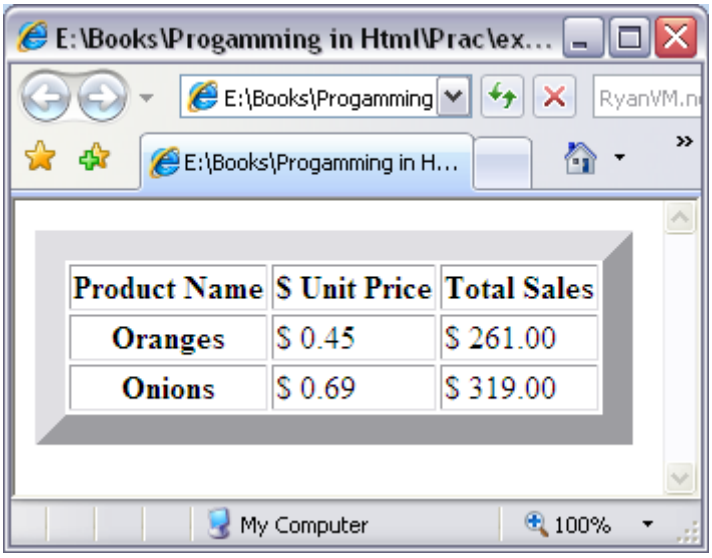
Table borders

You can specify the borders of the table so that they appear either in form of a line or as a bold margin. This can be done by the <border> tag. The following example shows the <border> tag.

```
<html>
<body>
<table border = 1>
<tr>
<th>Product Name  </td>
<th>$ Unit Price </td>
<th>Total Sales </td>
</tr>
<tr>
<th>Oranges </td>
<td>$ 0.45 </td>
<td>$ 261.00 </td>
</tr>
<tr>
<th>Onions  </td>
<td>$ 0.69 </td>
<td>$ 319.00 </td>
</tr>
</table>
</body>
</html>
```



The smallest size of a visible border is 1. You can increase the border width by choosing any number greater than 1. The following form was produced when I changed the border size to be 10.



Try other border sizes and note their benefits and challenges when it comes to presentations of your web pages.

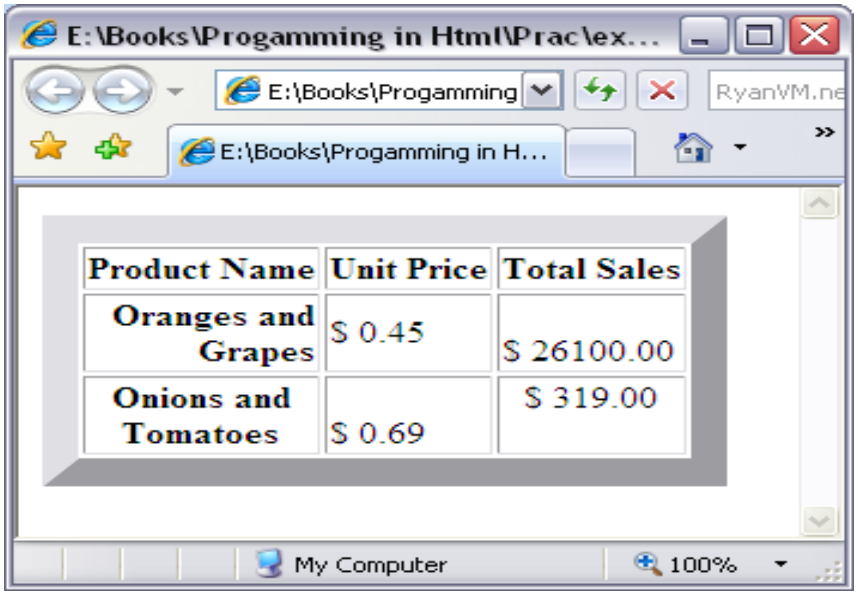
Aligning text in tables

The alignment can be done by the **<align>** tag as well as the **<valign>** tags. These two tags work inside the table data tag, <td> as well as inside the table header tag, <th> tag. The <align> tag aligns text either left, right at the centre of the cell. This is a horizontal text alignment tag. The <valign> tag aligns text with respect to its vertical position. The <valign> tag aligns text either to the top, middle or at the bottom part of the cell. This is effective when the cell has been sized to occupy more than one cell height.

The following example illustrates these two text alignment tag.

```
<html>

<body>

<table border = 15>

<tr>

<th>Product Name  </td>

<th>Unit Price </td>
```

```
<th>Total Sales </td>

</tr>

<tr>

<th>Oranges and <br>Grapes</td>

<td valign = middle >$ 0.45 </td>

<td valign = bottom>$ 26100.00 </td>

</tr>

<tr>

<th>Onions  and <br>Tomatoes</td>

<td valign = bottom >$ 0.69 </td>

<td valign = top align = center>$ 319.00 </td>

</tr>

</table>

</body>

</html>
```



From the web page above, take note of the following:

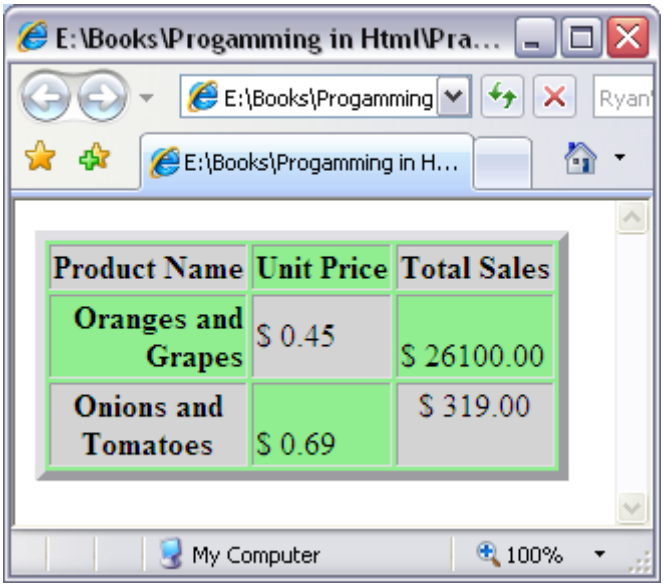
- Oranges and Grapes are right aligned
- Unit price for oranges and grapes ($0.45) is aligned in the middle of the cell
- The total sales value for oranges and grapes ($26100.00) is aligned at the bottom of the cell
- All headers, except "Oranges and Grapes", are aligned centrally. The table header tag, <th>, aligns text centrally by default. If you do not specify its alignment it remains centrally aligned in the cell. You can align it left or right.
- The total sales value for onions and tomatoes ($319.00) is aligned both centrally and on the top. This was done by the **<td valign = top align = center>** tag.
- Lastly, the unit price for Onions and Tomatoes ($0.69) is aligned to the bottom of the cell

This was a design of my choice. Try alternative layouts and evaluate their acceptability.

Setting table background colours

You can as well set the background colours of your table so that the table takes one colour or has different colour. I have chosen light green. This was implemented by the **<table bgcolor = "lightgreen">** tag. Take note of the **"lightgreen"**. It is not light green. This works well with the <border> tag, don't panic when you meet the following tag, <table bgcolor = "lightgreen" border = 5>.

You can also specify the colours of the cells inside a table. I have chosen to modify the table above so that it looks like this one!



The table was produced by the following commands:

```
<html>
<body>
<table bgcolor = "lightgreen" border = 5>
<tr>
<th bgcolor = "lightgrey">Product Name  </td>
<th>Unit Price </td>
<th bgcolor = "lightgrey">Total Sales </td>
</tr>
<tr>
<th align = right>Oranges and <br>Grapes</td>
<td bgcolor = "lightgrey" valign = middle >$ 0.45 </td>
<td valign = bottom>$ 26100.00 </td>
</tr>
<tr>
<th bgcolor = "lightgrey">Onions  and <br>Tomatoes</td>
<td valign = bottom >$ 0.69 </td>
<td bgcolor = "lightgrey" valign = top align = center>$ 319.00 </td>
</tr>
```

```
</table>
</body>
</html>
```

Cell padding

Cell padding refers to controlling the distance between the inner border of the cell and the text inside the text inside that cell. Cell padding can be set by the **<cellpadding>** tag. If cell padding is not specified, the web browser sets it by default to 5 pixels. I will combine cell padding with cell spacing in an example.

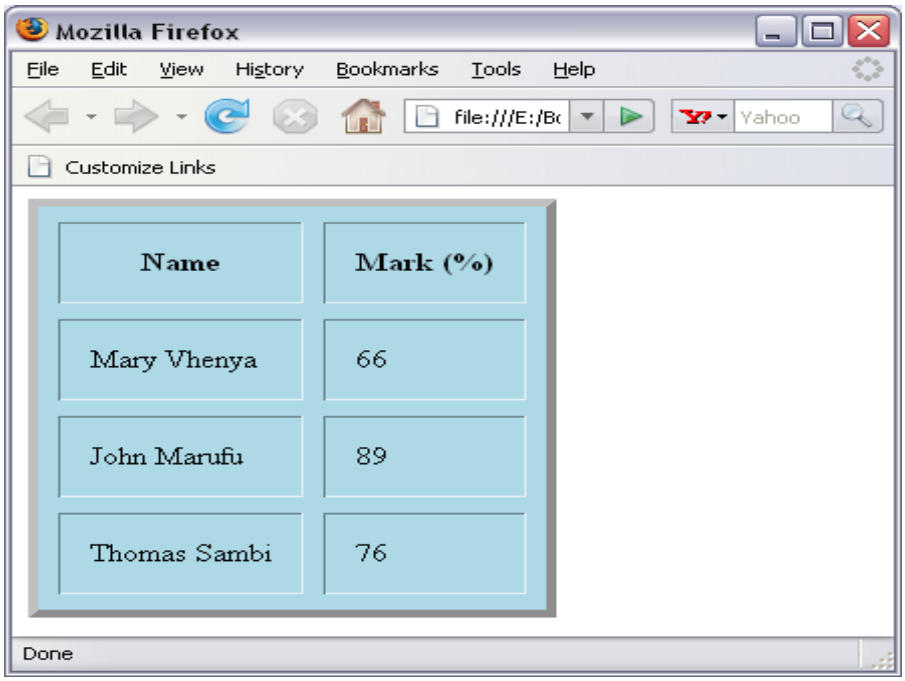
Cell spacing

Cell spacing refers to the distance between the cells in a table. The previous example used border lines that overrode each other to look as if it was one border. Imagine what happens when you draw a line with a ruler and repeat drawing the same line again without moving the ruler. You then call a friend and ask how many lines the friend can see. If your friend is fair enough to tell the truth, the answer will be “one line”. This is analogous to creating tables with borders but without cell spacing tags.

You can now specify the cell spacing by the **<cellspacing>** tag. This creates a gap between the cells of a table relative to the size of the spacing chosen. The following commands illustrate this concept.

```
<html>
<body>
<table bgcolor = "lightblue" border = 5 cellpadding = 15 cellspacing = 10>
<tr>
<th>Name</td>
<th>Mark (%)</td>
</tr>
<tr>
<td>Mary Vhenya</td>
<td>66</td>
</tr>
```

```
<tr>
<td>John Marufu</td>
<td>89</td>
</tr>
<tr>
<td>Thomas Sambi</td>
<td>76</td>
</tr>
</table>
</body>
</html>
```



The <caption> tag

This tag is used to create bold text that can be a title of a web page. You can also place it at the bottom of the page using the text alignment tags

explained above. The <caption> tag creates text with an effect similar to headings. The difference is that the text inside the <caption></caption> tags act as table text without boundaries. Inside the table developed above, I can add the <caption> tag between the

<table bgcolor = "lightblue" border = 5 cellpadding = 15 cellspacing = 10>

and <tr> commands so that the new set of commands will be:

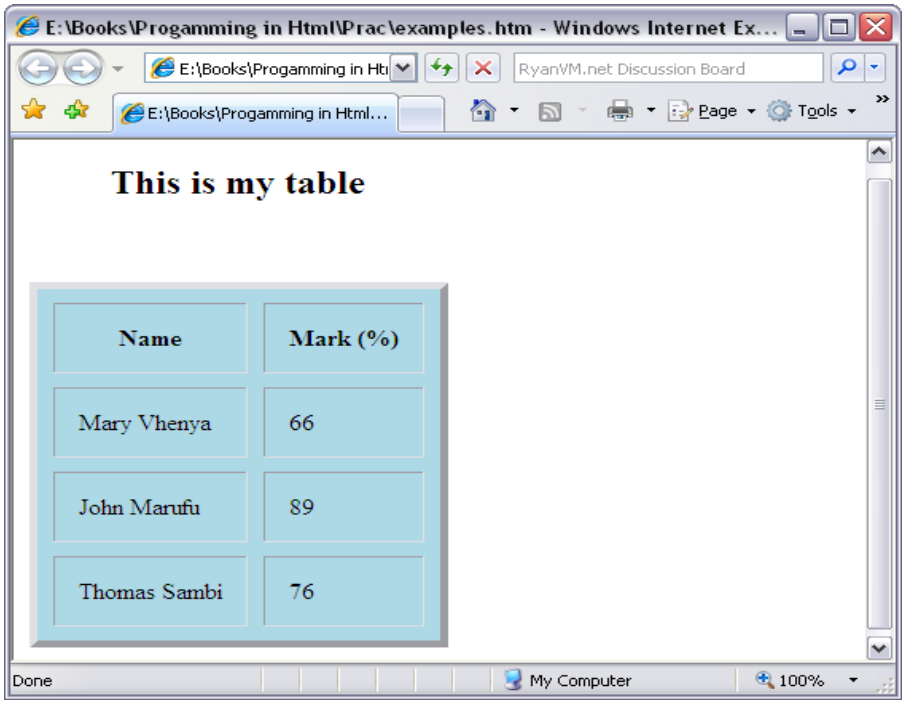
<table bgcolor = "lightblue" border = 5 cellpadding = 15 cellspacing = 10>

<caption> This is my table</caption>

<tr>

The remaining commands will be retained.

This produces the following web page.



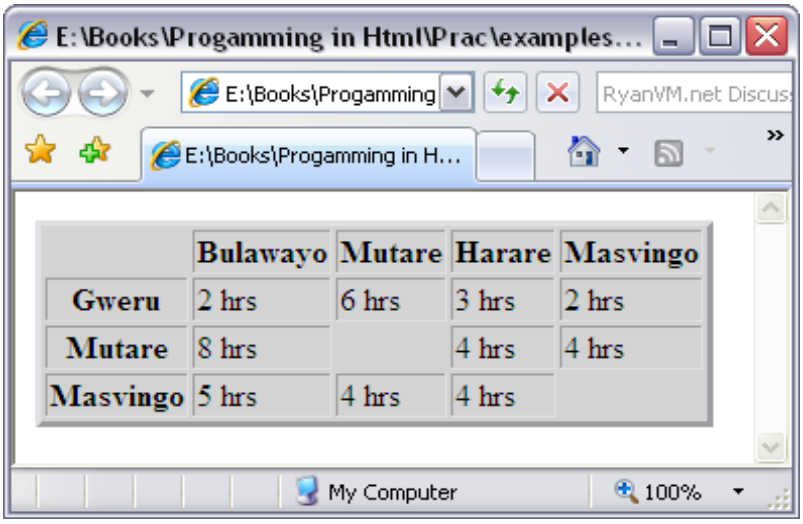
Note that the title “This is my table” is part of the table not just a heading without attachment to the table.

Creating blank cells in tables

Blank cells can be created on a table depending on the layout required as well as the nature of the information displayed. Let's say you want to display the average time you take to travel by road between towns in Masvingo. This can be done by creating a blank table data <td> </td>. The following example illustrates this.

```
<html>
<body>
<table bgcolor = "lightgrey" border = 3>
<tr>
<th></th>
<th>Bulawayo</th>
<th>Mutare</th>
<th>Harare</th>
<th>Masvingo</th>
</tr>
<tr>
<th>Gweru</th>
<td>2 hrs</td>
<td>6 hrs</td>
<td>3 hrs</td>
<td>2 hrs</td>
</tr>
<tr>
<th>Mutare</th>
<td>8 hrs</td>
<td> </td>
<td>4 hrs</td>
<td>4 hrs</td>
</tr>
```

```
<tr>
<th>Masvingo</th>
<td>5 hrs</td>
<td>4 hrs</td>
<td>4 hrs</td>
<td> </td>
</tr>
</table>
</body>
</html>
```



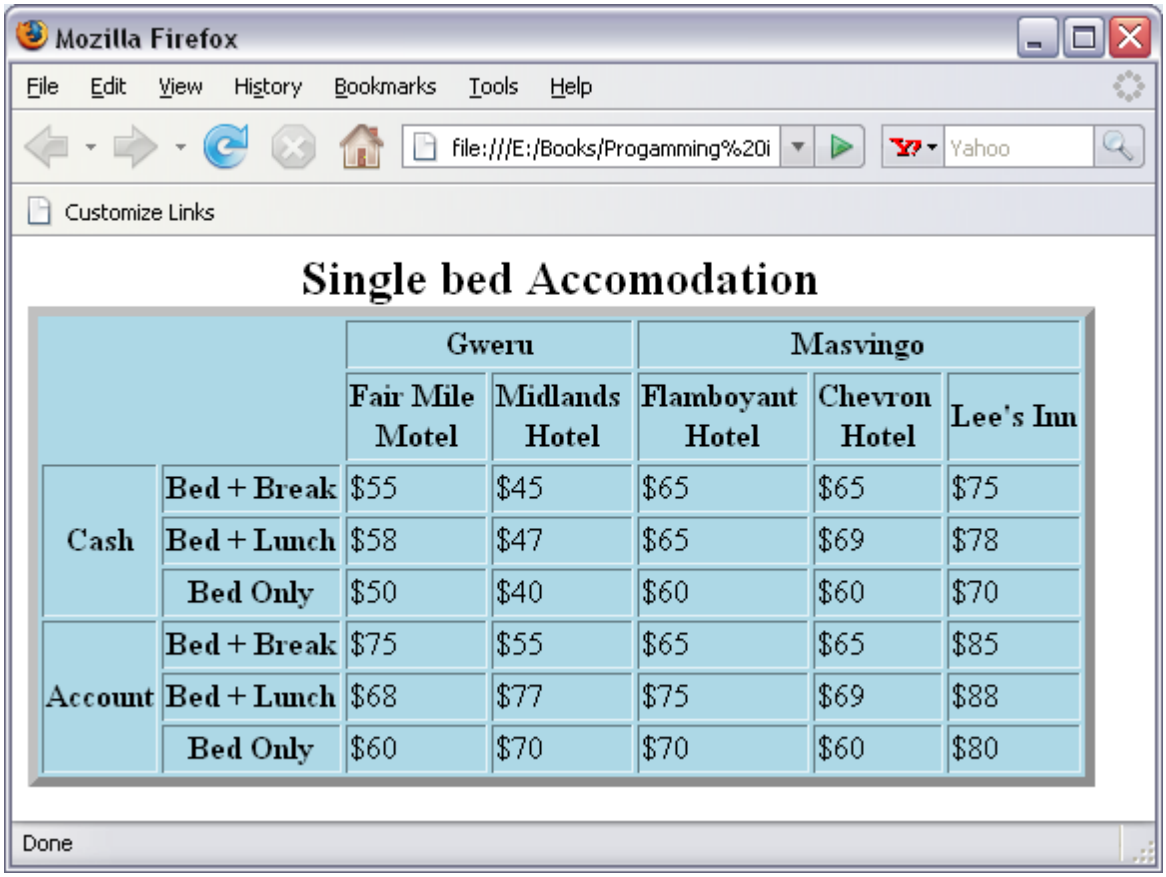
	Bulawayo	Mutare	Harare	Masvingo
Gweru	2 hrs	6 hrs	3 hrs	2 hrs
Mutare	8 hrs		4 hrs	4 hrs
Masvingo	5 hrs	4 hrs	4 hrs	

Spanning in tables

Sometimes you want to create a table with cells that occupy more than one cell. In other words, we may say one cell's length is two times the other cells. Another cell might also be 2 or three times in height as compared to its neighbouring cells. These two concepts can be developed by two tags: <rowspan> <colspan>. The following tags illustrate the concepts.

- <td rowspan = 2> </td> creates a cell whose width is 2 times the size of other cells
- <td colspan = 3 > </td> creates a cell that is 3 times wider than other cells

The following example illustrates cell spanning techniques.



Single bed Accomodation

		Gweru		Masvingo		
		Fair Mile Motel	Midlands Hotel	Flamboyant Hotel	Chevron Hotel	Lee's Inn
Cash	Bed + Break	$55	$45	$65	$65	$75
	Bed + Lunch	$58	$47	$65	$69	$78
	Bed Only	$50	$40	$60	$60	$70
Account	Bed + Break	$75	$55	$65	$65	$85
	Bed + Lunch	$68	$77	$75	$69	$88
	Bed Only	$60	$70	$70	$60	$80

You can develop this web page by typing the following commands:

```
<html>
<body>

<table bgcolor = "lightblue" border = 5>
<caption><h2>Single bed Accomodation</h2></caption>

<tr>
<th rowspan =2 colspan =2></th>
<th colspan =2 >Gweru</th>
<th colspan =3 >Masvingo</th>
</tr>
```

```
<tr>
<th>Fair Mile <br> Motel</th>
<th>Midlands <br> Hotel</th>
<th>Flamboyant <br> Hotel</th>
<th>Chevron <br> Hotel</th>
<th>Lee's Inn</th>
</tr>

<tr>
<th rowspan = 3>Cash</th>
<th>Bed + Break</th>
<td>$55</td> <td>$45</td> <td>$65</td> <td>$65</td> <td>$75</td>
</tr>

<tr>
<th>Bed + Lunch</th>
<td>$58</td> <td>$47</td> <td>$65</td> <td>$69</td> <td>$78</td>
</tr>

<tr>
<th>Bed Only</th>
<td>$50</td> <td>$40</td> <td>$60</td> <td>$60</td> <td>$70</td>
</tr>
<!-- Customers who visit on account and pay later>
<tr>
<th rowspan = 3>Account</th>
<th>Bed + Break</th>
<td>$75</td> <td>$55</td> <td>$65</td> <td>$65</td> <td>$85</td>
```

```
</tr>

<tr>
<th>Bed + Lunch</th>
<td>$68</td> <td>$77</td> <td>$75</td> <td>$69</td> <td>$88</td>
</tr>

<tr>
<th>Bed Only</th>
<td>$60</td> <td>$70</td> <td>$70</td> <td>$60</td> <td>$80</td>
</tr>

</table>
</body>
</html>
```

Chapter 4

L i s t s

Chapter Objectives

This chapter aims to address the following concepts:

- Unordered lists
- Specifying bullet type
- Ordered lists
- Nested lists
- Lists in tables

Introduction

Lists are useful when you want to emphasise chronology or when you want to display a collection of related items. Lists can be in the form of bullets, numbers or definitions. I prefer to group lists into three types: unordered lists, ordered lists and definition lists.

An unordered list

Unordered lists can be developed by the ** ** tags. The list items are identified by the **<li>** tags. The default option of an unordered list is a disk (a filled circle). Other options available are circle and square.

```
<html>
<head>
<title>Creating Lists</title>
</head>
<body>
<h2>Roman Catholic Schools in Zimbabwe</h2>

<ul><b>Midlands Province</b>
<li>Hama High School
<li>Fletcher High School
</ul>

<ul><b>Masvingo Province</b>
<li>Silveira High School
<li>Pamushana High School
<li>Gokomere High School
</ul>

</body>
</html>
```

Specifying bullet type

The type of bullets can be specified by the <UL TYPE = > tag. Options available are the square, circle and disc. The disc is a solid circle that we have used in the example above. Let's look at the other two here; the square and the circle options. Let us change the above commands so that line

Midlands Province

reads

<ul type = square>Midlands Province

and line

Masvingo Province

reads

Masvingo Province

The remainder of the commands remain unchanged. The resulting web page looks like this!

An ordered list

From the above lists, you can see that there is no need to arrange the lists items in chronological order since there is no reason either to compare the list items or to say one list item is dependent upon the other. At times you may want to display a list that requires chronology. The sequence can now be represented by an ordered list. The default list is Arabic. These are numbers you are familiar with (1, 2, 3 ...).

The following example illustrates numbered lists.

```
<html>
<head>
<head>
<title>Ordered Lists</title>
</head>
<body>
<h2>Want to be a happy landlord ?</h2>
<u><i><h2>Follow these steps</h2></i></u>
<ol>
<li>Come and register for land with us !
```

Make a monthly contribution of $100.00 for 3 months

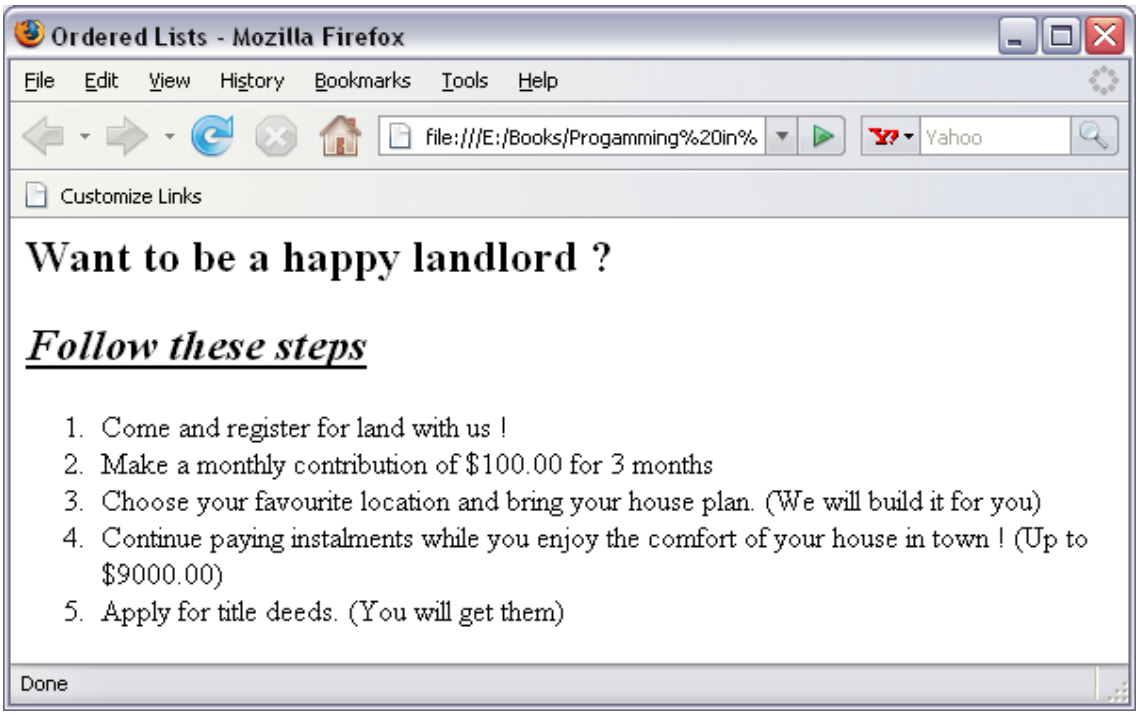
Choose your favourite location and bring your house plan. (We will build it for you)

Continue paying instalments while you enjoy the comfort of your house in town ! (Up to $9000.00)

Apply for title deeds. (You will get them)

</body>

</html>



Alternatives available include upper and lower case letters (e.g. A or a), as well as roman numerals in either upper case or lower case (e.g. i or I). The following example illustrates these alternatives.

<html>

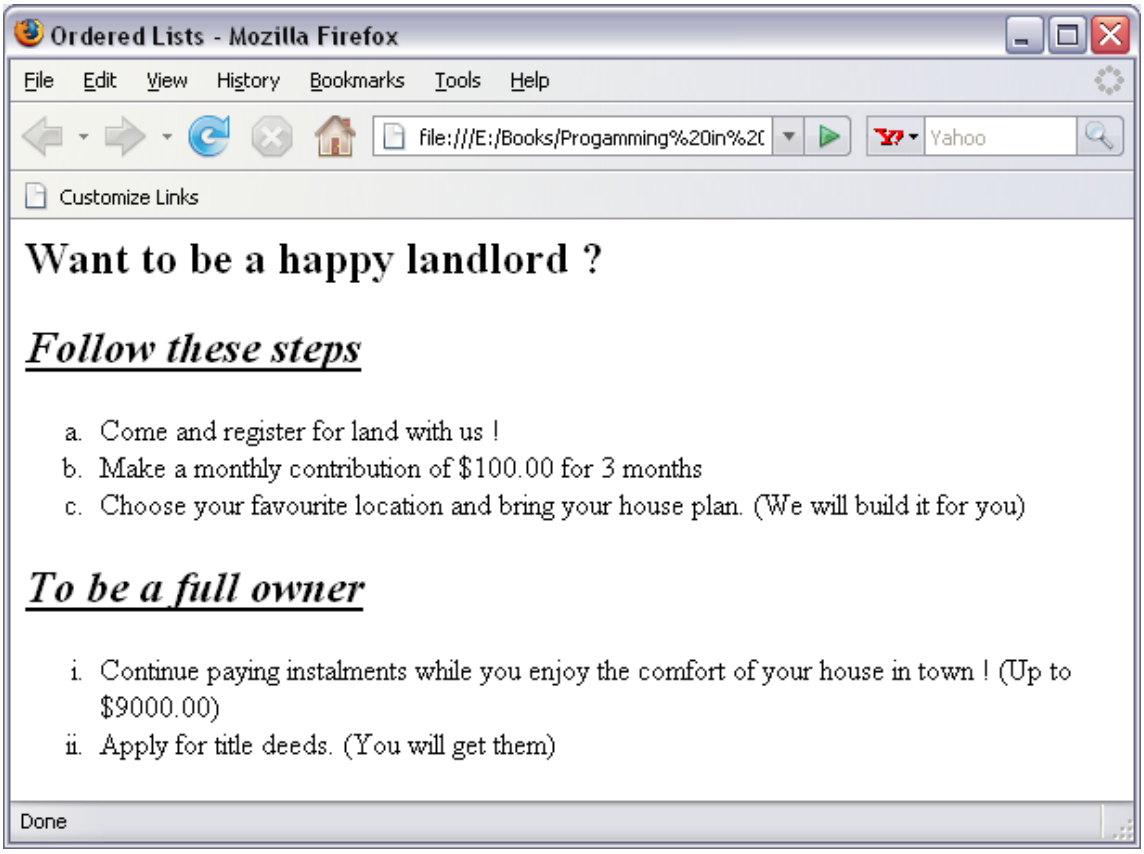
<head>

<head>

<title>Ordered Lists</title>

</head>

```
<body>
<h2>Want to be a happy landlord ?</h2>
<u><i><h2>Follow these steps</h2></i></u>
<ol type = a>
<li>Come and register for land with us !
<li>Make a monthly contribution of $100.00 for 3 months
<li>Choose your favourite location and bring your house plan. (We will build it for you)
</ol>
<u><i><h2>To be a full owner</h2></u></i>
<ol type = i>
<li>Continue paying instalments while you enjoy the comfort of your house in town ! (Up to $9000.00)
<li>Apply for title deeds. (You will get them)
</ol>
</body>
</html>
```



Nested list

Nested lists enable you to create a list inside another list. This can be useful when you have for example a website advertising courses or programs offered by a college. You can have a list of faculties e.g. commerce, sciences and education. In each faculty you then want to list all departments and programs offered by each department. This requires an understanding of nested lists. You can implement nested lists as shown in the following example.

<html>

<head>

<title>Nested Lists</title>

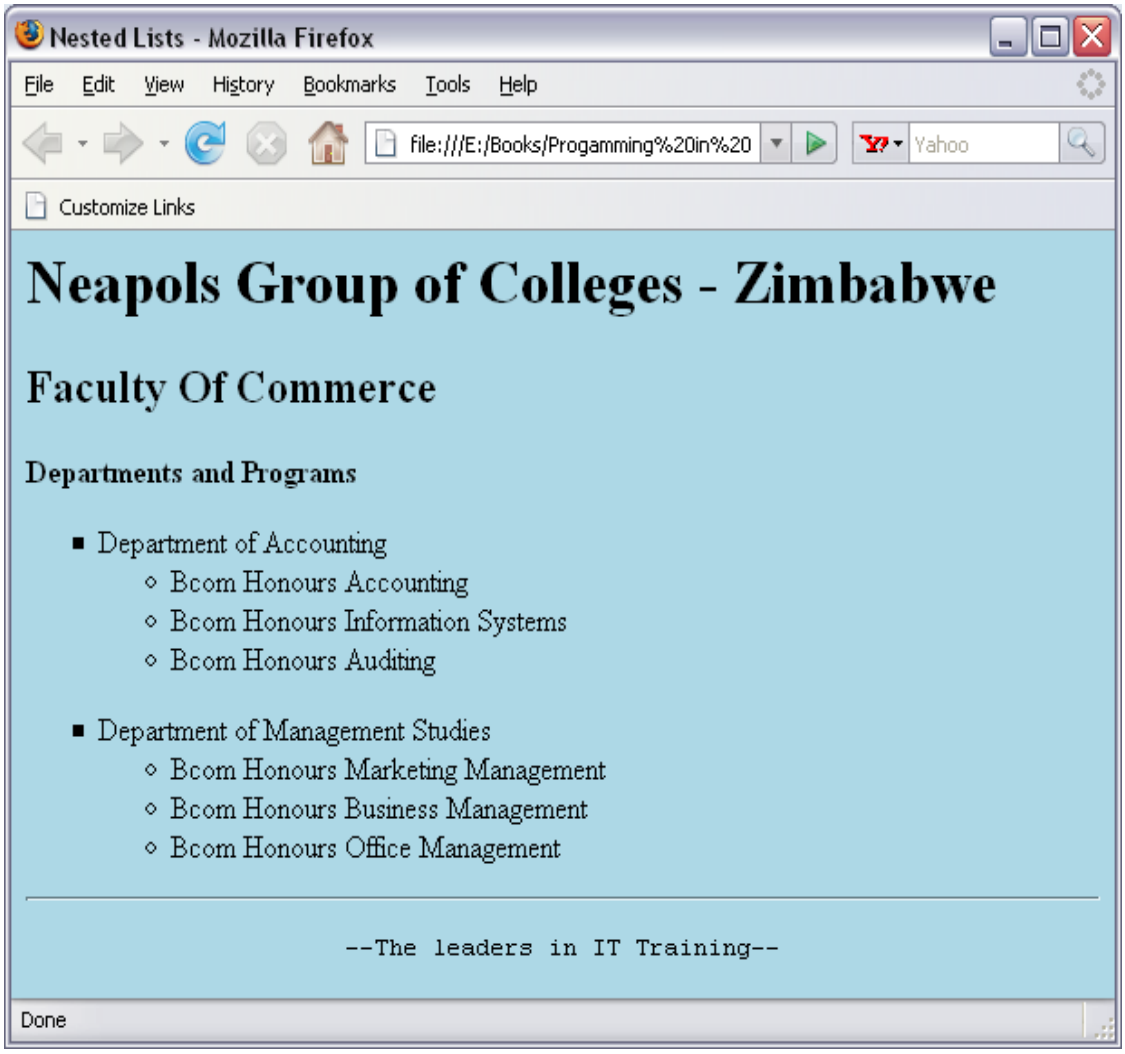
</head>

<body>

<html>

<body bgcolor = lightblue text = black>

```
<h1>Neapols Group of Colleges - Zimbabwe</h1>

<h2>Faculty Of Commerce</h2>
<b>Departments and Programs</b>
<ul type = square>
<li>Department of Accounting
<ul type = circle>
<li>Bcom Honours Accounting
<li>Bcom Honours Information Systems
<li>Bcom Honours Auditing
</ul>
</ul>

<ul type = square>
<li>Department of Management Studies
<ul type = circle>
<li>Bcom Honours Marketing Management
<li>Bcom Honours Business Management
<li>Bcom Honours Office Management
</ul>
</ul>
<hr>
<p align = "center">
<tt>--The leaders in IT Training--
</tt>
</align>
</body>
</html>
```



I have chosen unordered lists to demonstrate the concept of nested lists. You can also do the same with ordered lists. Try it!

Lists in tables

<html>

<head>

<title>Nested Lists</title>

</head>

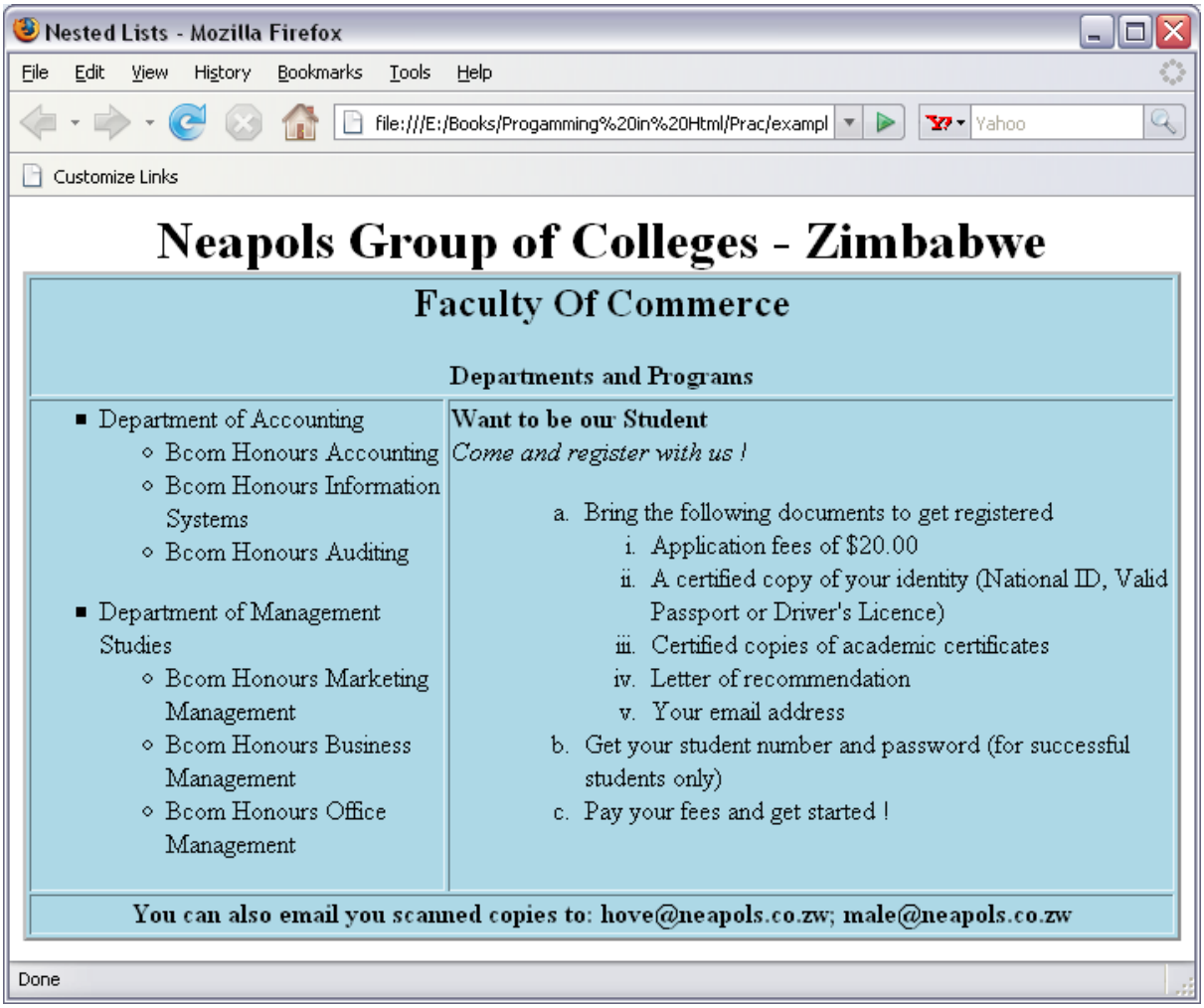
<body>

<table border =2 bgcolor = lightblue text = black>

<caption><h1>Neapols Group of Colleges - Zimbabwe</h1></caption>

<tr>

<th colspan =2><h2>Faculty Of Commerce</h2>



Departments and Programs

</th>

</tr>

<tr>

<td><ul type = square>

Department of Accounting

<ul type = circle>

Bcom Honours Accounting

Bcom Honours Information Systems

Bcom Honours Auditing


```
<ul type = square>
<li>Department of Management Studies

<ul type = circle>
<li>Bcom Honours Marketing Management
<li>Bcom Honours Business Management
<li>Bcom Honours Office Management
</ul>
</ul>
</td>

<td valign = top><b>Want to be our Student </b><br>
<i>Come and register with us !</i>

<ol>
<ol type = a>
<li>Bring the following documents to get registered
<ol type = i>
<li>Application fees of $20.00
<li>A certified copy of your identity (National ID, Valid Passport or Driver's Licence)
<li>Certified copies of academic certificates
<li>Letter of recommendation
<li>Your email address
</ol>
<li>Get your student number and password (for successful students only)
<li>Pay your fees and get started !
</ol>
</td>
```

```
</tr>
<tr>
<th colspan =2 align = center>
You can also email you scanned copies to:
hove@neapols.co.zw;
male@neapols.co.zw
</td>
</tr>
</body>
</html>
```

Chapter 5

Creating forms for input

Chapter Objectives

This chapter aims to address the following concepts:

- Creating text fields
- Passwords, reset and submit commands
- Radio buttons
- Setting the default radio button
- Setting the value in radio buttons
- Checkboxes
- Drop down lists
- Preselected drop down lists
- Buttons
- Specifying text area

Introduction

Forms are an interesting part of html. Unlike other examples so far, forms allow us to communicate in two ways. One side is sending and the other is receiving. Let's say you want to enquire on the courses offered by a college. You must have a platform to ask your question and supply your contact address. This is done by the form method. The form method allows the receiver to process your information as per the form design.

Imagine, you have browsed the internet and you come across a car dealer's website. You were all along looking for this type of car! How do you send your personal information to the dealer so that the sale is completed? Alternatively, when you visited your email service provider's website, e.g. gmail, you were asked to fill in all the relevant information (your name, email address, password etc).

You filled in your details in the textboxes, combo boxes, option buttons or checkboxes. After filling all the information you then clicked a command button that was written "submit" or "send".

How was the page developed? In this chapter, I am going to guide you through the whole process of interacting with the web server so that you get responses from the client machine. As you browse the internet, you are a client to the network server (a computer that hosts that website). Forms can be embedded on a large web page or they can be developed independently. If you develop the form independently, you then hyperlink it to the calling page. The form can contain all forms of text display and images just like any other web pages.

The form is developed by the <form> </form> tags. These identify the beginning and end of the form. Inside the form you will need two keywords: METHOD and ACTION. Well, let's look at these commands:

<form method = Post action = mailto:school@gmail.com>

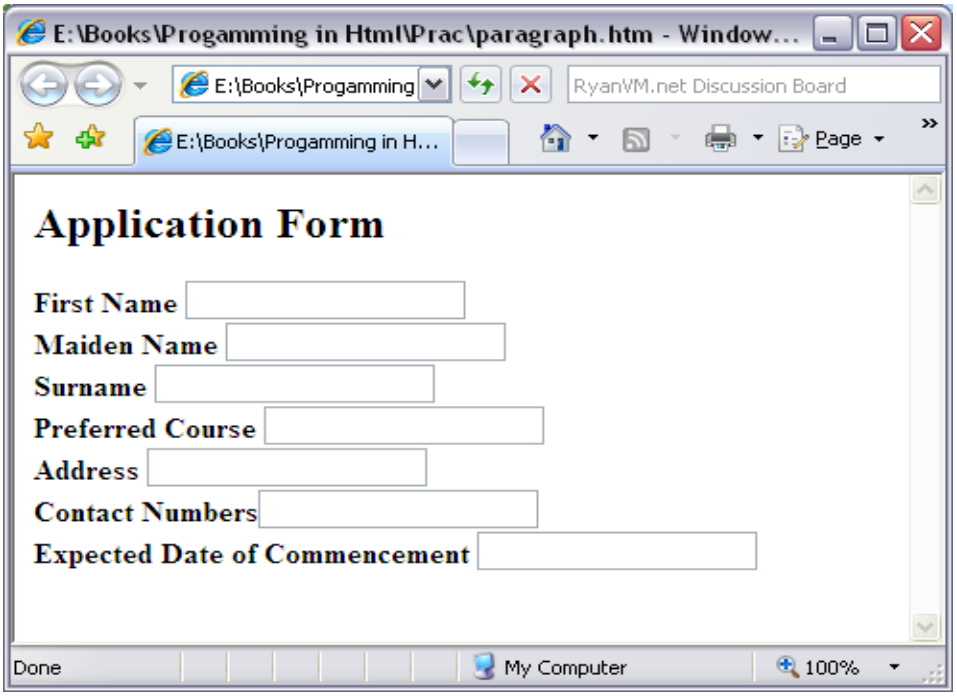
The above commands send data to an email address (school@gmail.com). This data will be send after the form is submitted. The data send will be collected from the from input boxes, option buttons etc. Let me first look at the input methods available on the form before I jump into submitting the form!

Creating text fields

You might want to type something for feedback to the website. From the previous chapters, you only concentrated on the design of the web page. Now I want you to be able to create a web page that allows you to specify your information, or comments to the company owning the site. Text fields can be created by the

<input name = > in the **<form> </form>** tags. The <input name = > creates a text box where you can type your details so that these will be submitted to the web server when you click the submit button.

Creating the following form does not take long!



The form was produced by the following commands:

<html>

<body>

<form>

<h2>Application Form </h2><p>

First Name <input name = fname>

Maiden Name <input name = mname>

Surname <input name = sname>

Preferred Course <input name = course>

Address <input name = address>

Contact Numbers<input name = phone>

Expected Date of Commencement <input name = startdate>

</form>

</body>

</html>

From the form above, you can see that the text boxes have the same size. By default, the textboxes created by the <input name = > commands 20 characters. You can further specify the size of text fields by the SIZE option e.g.

<input name = address size = 40>

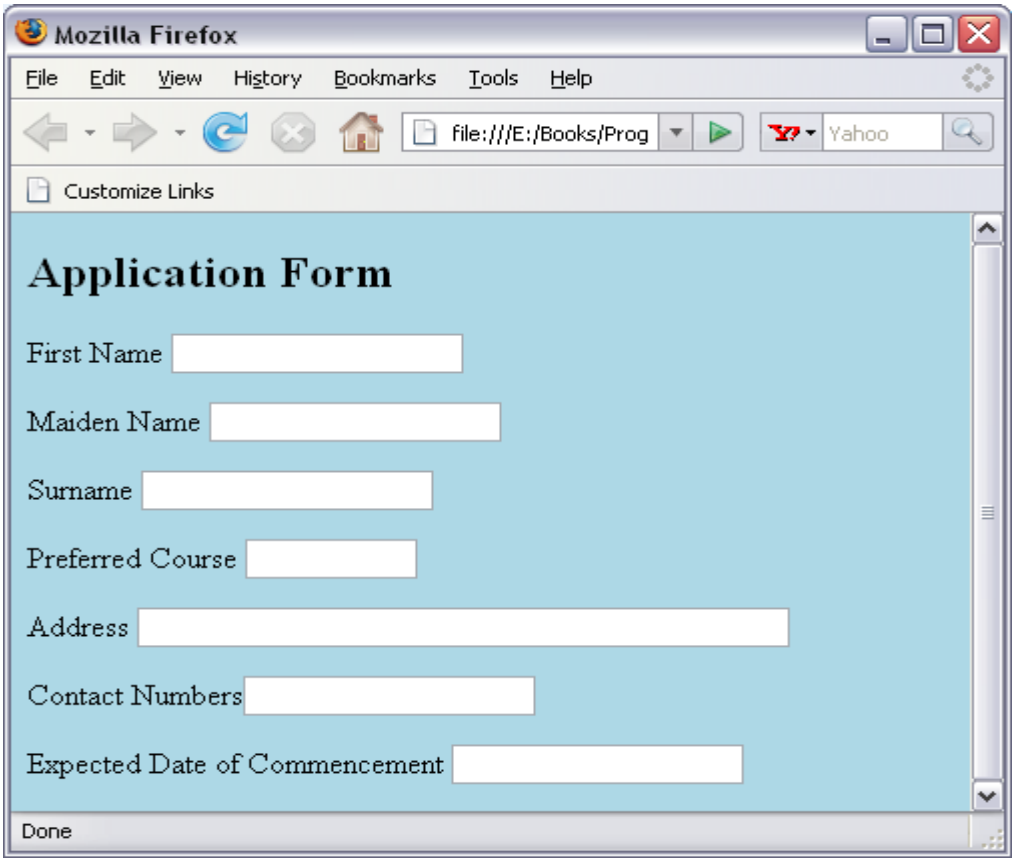
Resizing the preferred courses and address textboxes to be 10 characters and 40 characters respectively, and replacing all
 tags with <p> tags:

Preferred Course <input name = course size = 10><p>

Address <input name = address size = 50><p>

And changing the <body> tag to **<body bgcolor = "lightblue">**

produce the following form:



The form looks better with the paragraph tag, <p> than with the line break tag
.

Passwords, Reset and Submit commands

Passwords, textboxes, reset buttons and submit buttons can be created by the **type** option in the <input -- > tag. This can be done in the following form:

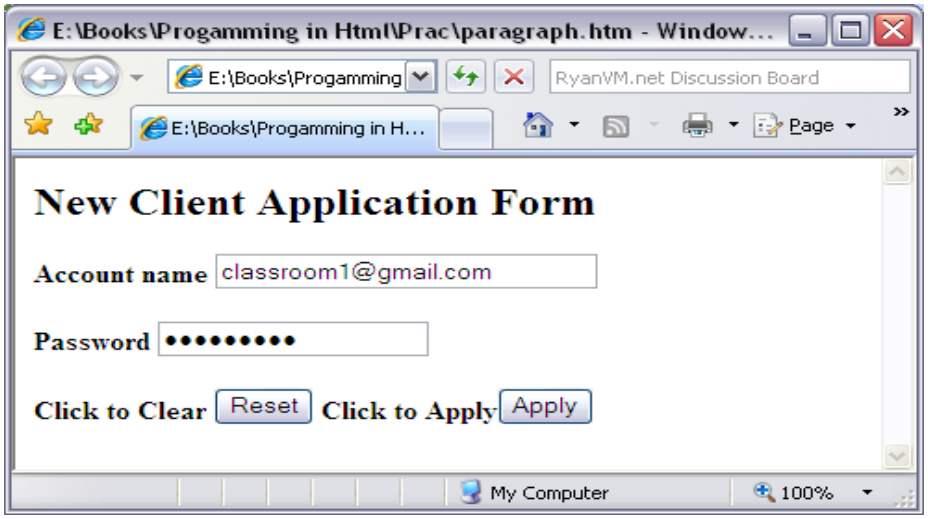
<input type = password>

<input type = reset>

<input type = submit>

The password does not need to be printed directly on the screen as you type the text. The text typed must not be displayed as it is; it is better displayed as asterisks (e.g. ****) or circles (either solid or blank circles). The following example illustrates this:

```
<html>
<body>
<form>
<h2>New Client Application Form </h2><p>
Account name <input name = usname size =30><p>
Password <input type = password><p>
Click to Clear <input type = Reset>
Click to Apply<input type = Submit value = Apply>
</form>
</body>
</html>
```

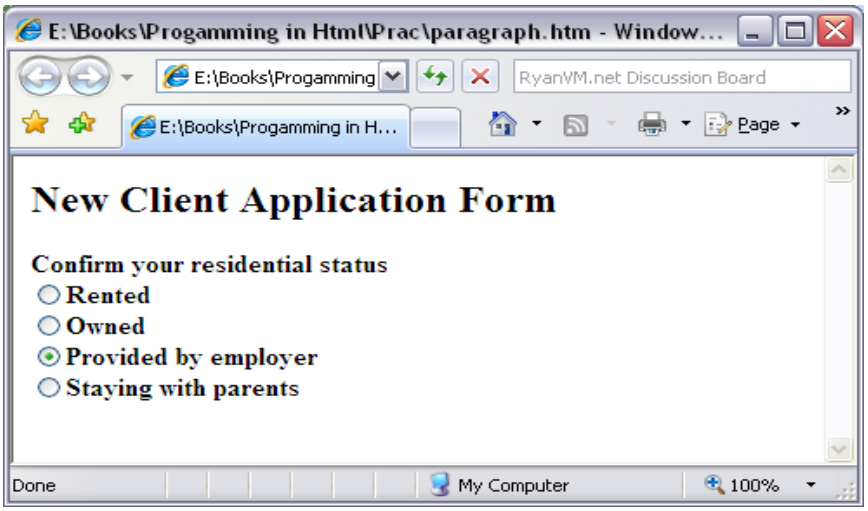


Radio buttons

Radio buttons are blank circles which you can tick to indicate selection of options. They provide the client with a platform to choose one option from a list of available options. Let's say you want a client to supply information about their residential status. The client's home might be rented, staying with parents, owned by the client or provided by employer. Well, let me demonstrate radio buttons here:

```
<html>
<body>
<form>
```

```
<h2>New Client Application Form </h2><p>

Confirm your residential status<br>

<input type = radio name = res>Rented <br>

<input type = radio name = res>Owned <br>

<input type = radio name = res>Provided by employer <br>

<input type = radio name = res>Staying with parents <br>

</form>

</body>

</html>
```



When you are developing radio buttons, take care of your definition of options. They do not have to ambiguous since the client has a restricted number of choices. From the above example, you cannot stay on a house that you both rent and own. It does not make sense. This means that there is mutual exclusion between options. Choosing "rented" excludes "owned". Since there is only one option acceptable here, you must use one name (variable) for all the options in that list. The example above used **res** as the variable. This was short for residence.

A common mistake is to use different names for radio buttons. This results in the user clicking more than one option in the radio button list. This produce wrong results when the feedback is submitted. If you write

```
<input type = radio name = res> Rental <br>
```

<input type = radio name = own> Owned

you will get confusing results since the user can click all the radio buttons. **That is wrong**! Do not use different names here.

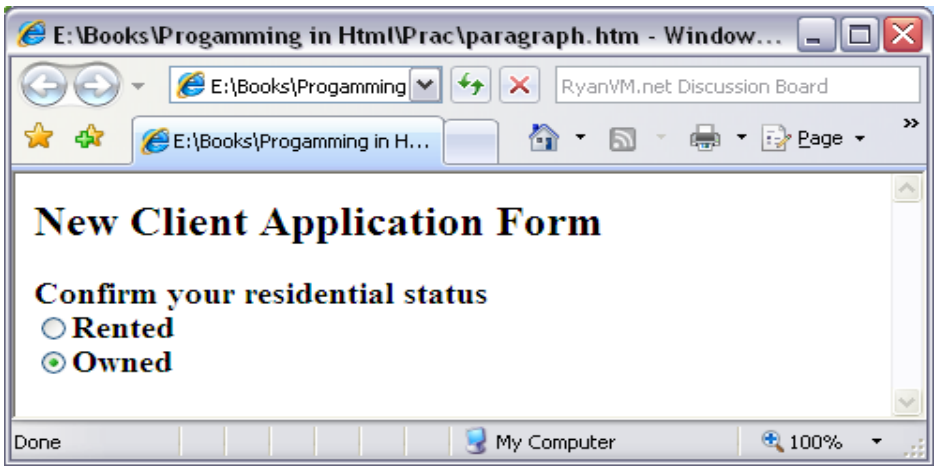
Setting the default radio button

The default radio button can be set by the **checked** keyword e.g.

<input type = radio name = sex checked>

The following example illustrates this concept.

```
<html>
<body>
<form>
<h2>New Client Application Form </h2><p>
<h3>Confirm your residential status<br>
<input type = radio name = res>Rented <br>
<input type = radio name = res checked>Owned <br>
</form>
</body>
</html>
```



Setting the value in radio buttons

The commands:

<input type = radio name = res> Owned

<input type = radio name = res> Rented

create two radio buttons identified by the same name. This is OK but, how is each unique option identified when the form is submitted. Clicking **Owned** or **Rented** cannot be differentiated using these commands.

We can now give the form a more refined differentiation of the option selected. This is done by using the **value** clause in the radio buttons. The following commands illustrate this concept.

<input type = radio name = res value = own>Owned

<input type = radio name = res value = ren>Rented

When the client clicks "owned", the value returned in place of **res** is **own**, thus:

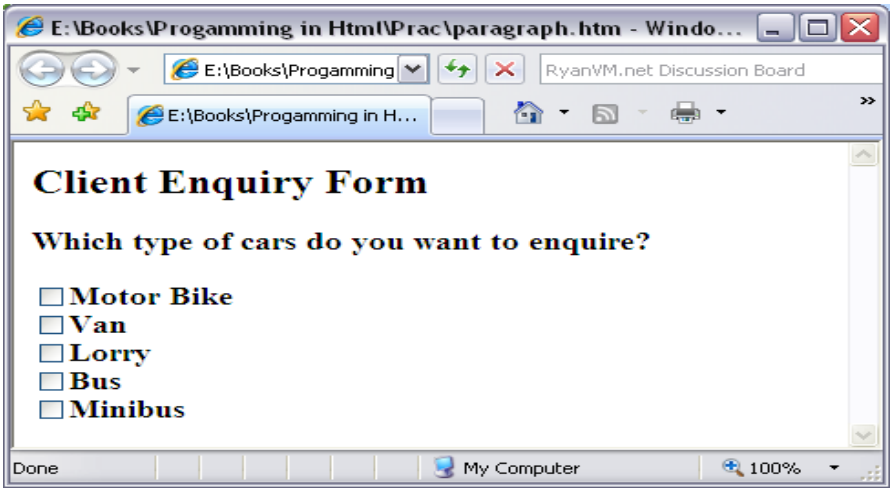
res = own

Failure by the programmer to specify the value clause, will result in the feedback reading **res = on** regardless of what option was selected. You cannot make any decision from the res = on.

Checkboxes

Checkboxes take a dimension almost similar to the radio buttons. The only difference is that checkboxes allow multiple selections of options. This is due to the fact that every option has a different name.

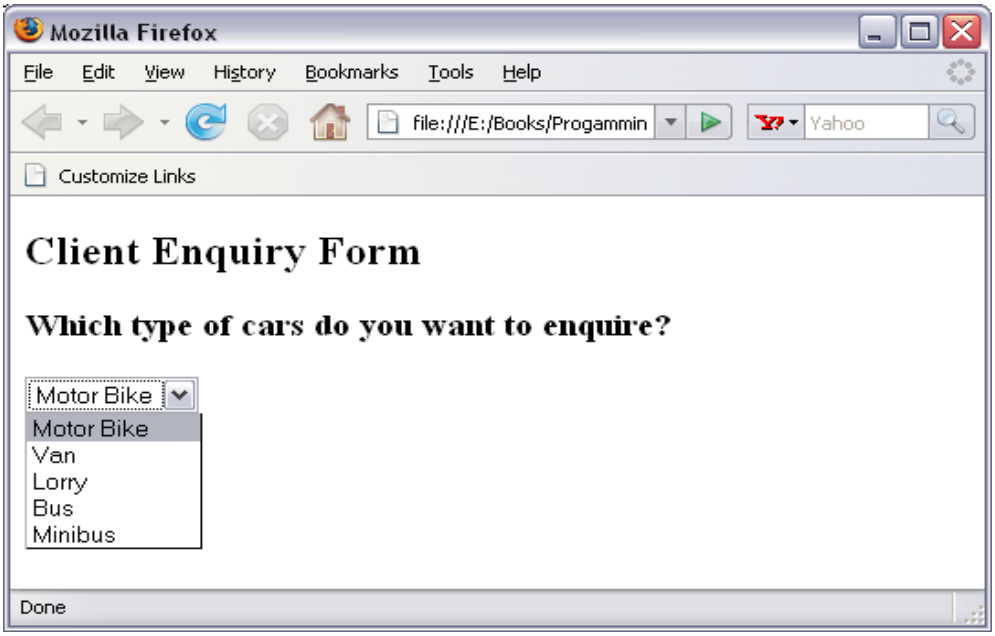
The following example illustrates this concept.




```
<html>
<body>
<form>
<h2>Client Enquiry Form </h2><p>
<h3>
Which type of cars do you want to enquire? <p>
<input type = checkbox name = bike>Motor Bike <br>
<input type = checkbox name = van>Van <br>
<input type = checkbox name = lorry>Lorry <br>
<input type = checkbox name = bus>Bus <br>
<input type = checkbox name = mini>Minibus <br>
</form>
</body>
</html>
```

Drop-down lists

Let's use the same data above to create drop create drop down lists. The drop down list works in a way almost similar to radio buttons. You can choose only one option. This time you click from the list of combo box items.




```
<html>

<body>

<form>

<h2>Client Enquiry Form </h2><p>

<h3>

Which type of cars do you want to enquire? <p>

<select name = cars>

<option = bike>Motor Bike

<option = van>Van

<option = lorry>Lorry

<option = bus>Bus

<option = mini>Minibus

</select>

</form>

</body></html>
```

The block of commands:

<select name = ... >

<option = ...>

</select>

create the drop down list that can be selected by the client. Choosing the "van" option form the list will have the following effect on your form:

cars = van.

This means we have chosen van and from programming rules the symbol, =, is not an equal sign but it is an assignment operator. The variable **cars** will then contain the value **van.**

Preselected drop down lists

From the previous example, you can see that the default item selected was Motor Bike. This is because the Motor Bike option is the first item in the list. Sometimes you may want to use the most frequently selected item as a default item on the combo box. You can achieve this by using the **selected** keyword, e.g.

<option = mini selected > Minibus

This will produce a form with minibuses set as the default option for motor vehicle enquiry.

Buttons

The commands <input type = submit> creates a command button with a default caption "Submit Query". You can also change the caption by using the following commands:

<input type = submit value = "Submit">

This option gives you very limited control over the command buttons. You can only change its caption. You can have more control over the command buttons by using the commands

<button type = "submit">Apply

This command gives the developer more control over the form. Alternative arguments available include formatting font type, size and colour. The font can be formatted by the following commands:

<button type “submit”>

 Apply Now!

</button>

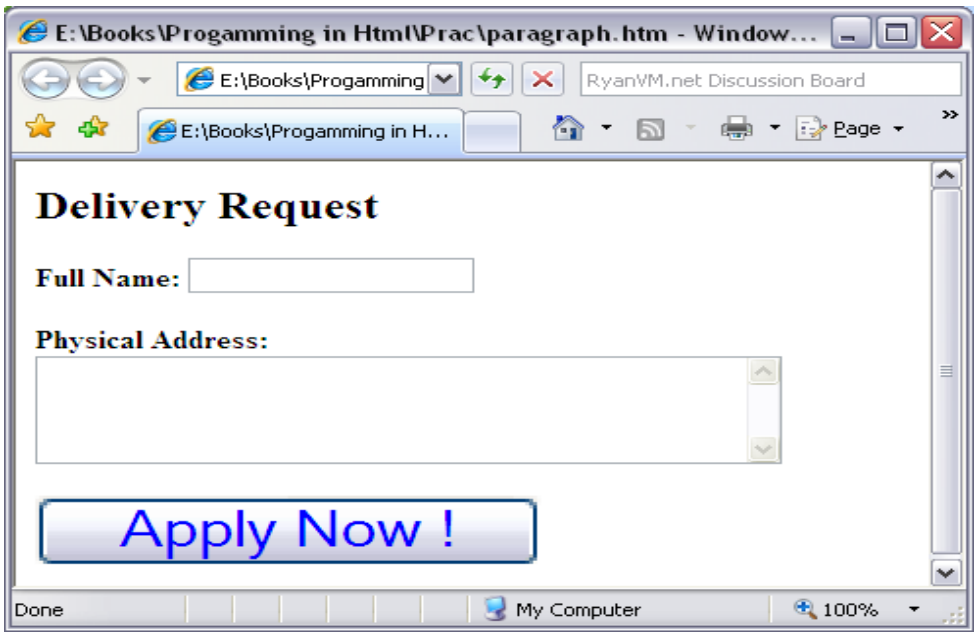
You can also use images in place of command buttons e.g.

<button type = “submit”>

<img src = “Apply.png”>

</button>

The following example illustrates the <button> tag.



<html>

<body>

<form>

<h2>Delivery Request</h2>

Full Name:

<input name = name><p>

Physical Address:

<textarea name = comm rows = 4 cols = 45>

</textarea>

<button type “submit”>

 Apply Now!

</button>

</form>

</body>

</html>

Specifying text area

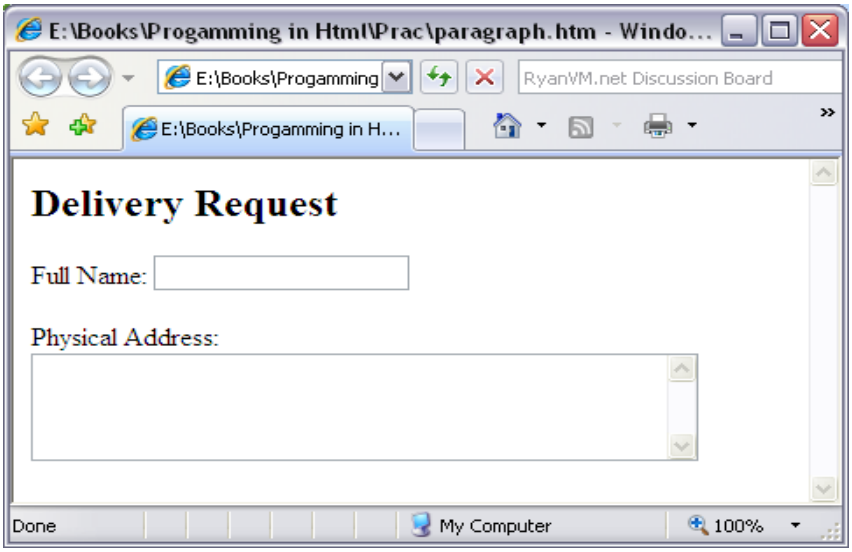
From the previous examples you can see that the textbox was created by the <input...> tag. This creates only one line of input. If you want to capture more than one line you will then have to use more textboxes. The best option is to use the

<textarea ... > tag. For example, a textarea to capture comments from a web client can be developed by the commands:

<textarea name = comm rows = 6 cols = 30>

</textarea>

The following example demonstrates this concept.




```
<html>

<body>

<form>

<h2>Delivery Request</h2>

Full Name:

<input name = name><p>

Physical Address:<br>

<textarea name = comm rows = 4 cols = 45>

</textarea>

</form>

</body>

</html>
```

Chapter 6

Images Sound and Links

Chapter Objectives

This chapter aims to address the following concepts:

- Introduction
- Inserting images
- Aligning images
- Setting images as backgrounds
- Setting images as links
- Inserting sound

Introduction

Images improve the display of web pages. They are self marketing. For example, a safari company might summarise events in the jungle by using pictures on its website.

Images can be in different file formats. The most common types are GIF (Graphic Information Format) and the JPEG (Joint Photographic Experts Group). Due to compatibility reasons, I have chosen the PNG format. From my experience, this format displays well on most web browsers without complications. You rarely find a picture in PNG format. How then do you change the picture to PNG? You can do this by following these steps.

1. Open the folder containing your image that is not in PNG format
2. Right click the image
3. Click "Open With"
4. Choose Paint. If the Paint option is not visible Click "Choose Program" then click Paint.
5. Click File then Save As
6. Type the file name and Click the drop down list under Save As Type
7. Click PNG
8. Click Save

The file can have the same name with the original file but with different formats. This can be identified by file extensions. In all examples in this book, I have decided to use the PNG format.

Inserting images

The **<img src = >** produces the images. The image is produced from a specific location where the image is saved. It is advisable to save your image in the same folder as the web page. If you are familiar with path specification, you can use an image from a different folder.

The following commands produce an image on a web page.

```
<html>
<head>
<title>Images </title>
<head>
```

```
<body>
<img src = "Hilltop.png">
</body>
</html>
```

The image above was produced by the command **<img src = "Hilltop.png">**. This was possible since the image "Hilltop" was stored in the same folder as the web page. You can use the image from different folders by specifying the path of the image. This can be done by the following commands:

<img src = "E:\Books\Progamming in Html\Prac\Hilltop.png">

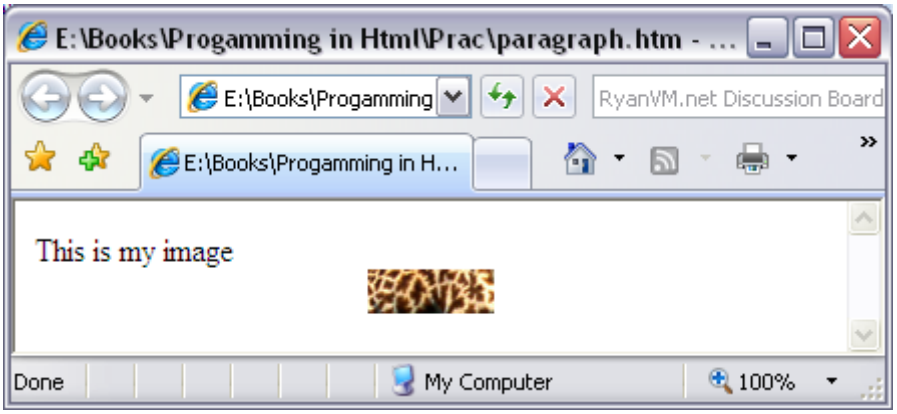
I advice junior programmers to store images in the same folder with the web page.

Aligning images

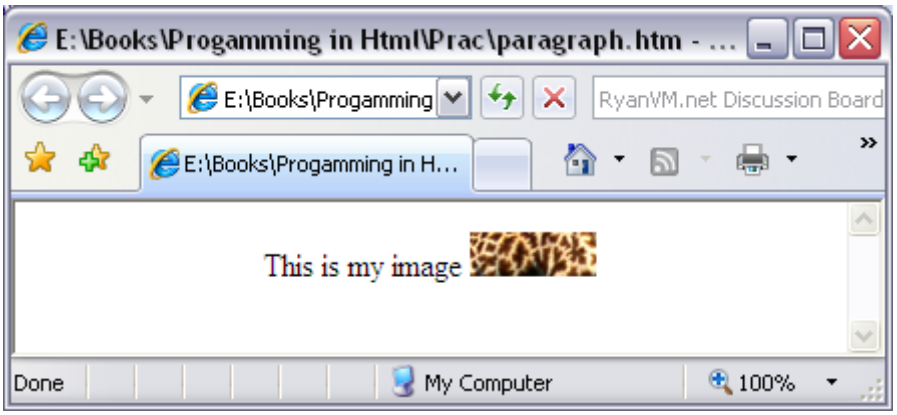
Images can be aligned just like the text alignment described in the previous chapters. Images can be aligned to the top, middle, bottom or centrally.

The ** ** tags are used to align images centrally. Take note of the American spelling of **center** not centre. The following example illustrates the tag.

```
<html>
<body>
This is my image
<center>
<img src = "gira.png" >
</center>
</body>
</html>
```



Note that the   tags are applied to the image only. Sometimes you may want to align both the text and the image. Applying the   tags to both the image and the text produces the following web page.



The webpage was produced by altering the code:

This is my image

*** ***

<img src = "gira.png" >

*** ***

So that it reads

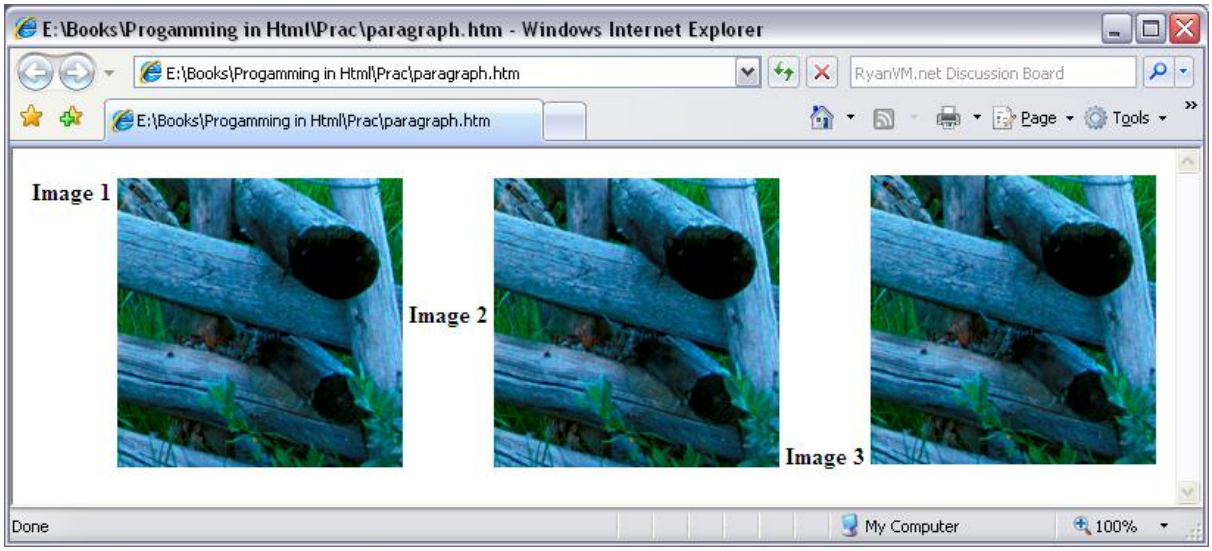
*** ***

This is my image

<img src = "gira.png" >

*** ***

You can also use the **<align = >** tag to align your images. The default alignment is bottom. The <align = > tag can be illustrated easily by incorporating tables so that you compare the positions of text and images.



<html>

<body>

<table>

<tr>

<td>

```
<b>Image 1
<img src = "logs.png" align = top >
</td>
<td>
<b>Image 2
<img src = "logs.png" align = middle>
</td>
<td>
<b>Image 3
<img src = "logs.png" align = bottom>
</td>
</tr>
</table>
</body>
</html>
```

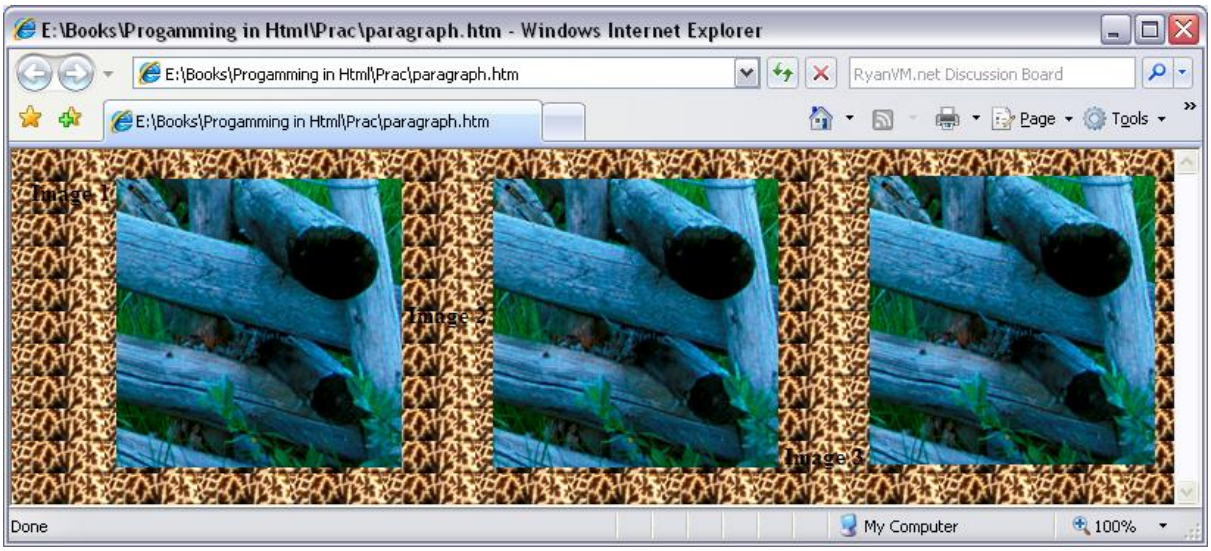
Setting images as backgrounds

Images can also be set as page backgrounds. The background image does not need to be too big to fit the whole page. The image does not need to be too dark or too bright. Too dark images colours produce unreadable pages. Some colours e.g. yellow, red and blue produce web pages that irritate eyes of readers. You can use images that are lighter e.g. “lightblue”. A smaller image produces a more interesting pattern as it is repeated continuously on the web page. Choose smaller images so that the pattern is generated. The following example illustrates this concept.

I have decided to modify the above commands so that they read:

```
<html>
<body background = "gira.png">
<table><tr>
<td><b>Image 1<img src = "logs.png" align = top ></td>
<td><b>Image 2<img src = "logs.png" align = middle></td>
```

```
<td><b>Image 3<img src = "logs.png" align = bottom></td>
</tr></table>
</body>
</html>
```

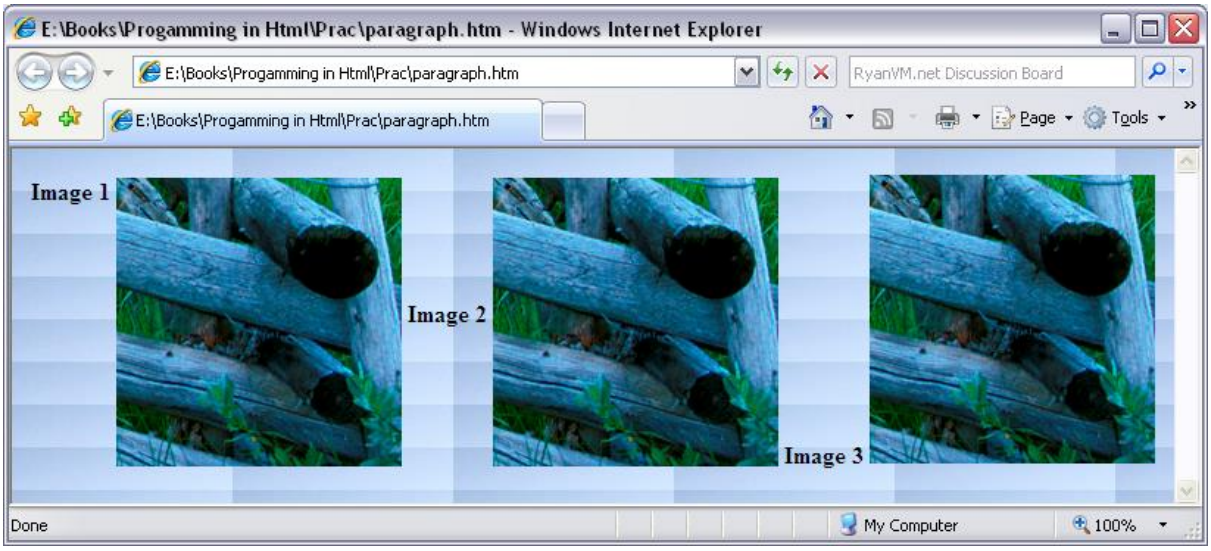


If you look at the above web page, you can see that the background pattern produced is a repeated pattern of the giraffe body. This was produced from  giraffe image whose dimensions are shown below:

Look at the original size of the image and how it appears when used as a web background.

I can also use the following image as the background of a page.

Replacing the commands **<body background = "gira.png">** with the commands <body background = "blue.png"> produces the following web page.



Hyperlinks

Links can be made between two web pages or within the page itself. This concept is so simple! If you want to link a web page to another web page, you might have the following two options.

- **Linking web pages that are in the same folder**

  If the web pages are in the same folder a link can be done by the following commands:

  My home page

  From the above commands, let’s look at the following:

  <a href > creates a hypertext reference

  =”My_page.htm” specifies the page that is supposed to be linked. This must be the exact name of the saved document. If you saved the document as Mypage.htm, then you try to hyperlink it as My_page.htm, the page will not be linked.

  My home page is the statement that displayed on the linking page. In other words, this is the prompting statement that instructs you to link a page.

- **Linking web pages from different folders**

If you are creating a link to other pages outside the folder where the linking page is, you have to specify the path where that web page is stored. It is advisable to copy the path so that errors can be minimised. Linking the web page above from a different folder can be done by the following commands:

My home page

The commands above specify the full path of the folder.

Setting images as hyperlinks

<html>

<body background = "blue.png">

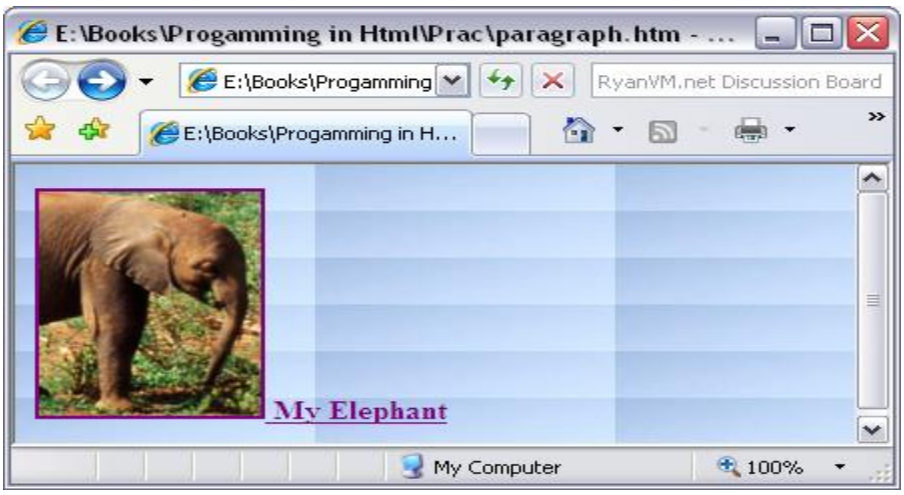
<img src = "Eleph.png"">

<b>My Elephant

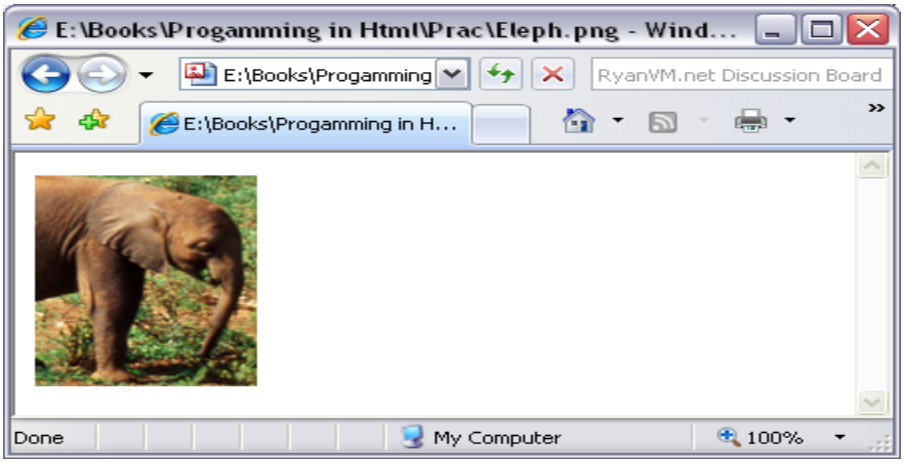
</body>

</html>

The above commands produced the following web page



Clicking the elephant image or clicking the phrase “My Elephant” produces the following page



Can you explain what happened?

Inserting sound

To insert sound on a webpage, use the sound clip as a link. Use the hyperlink concept. Clicking the hyperlink opens your sound clip.

<html>

<head>

<title>Images </title>

<head>

<body>

Play Music :

 Click here to play music

</body>

</html>