

03 HTML Document Structure

1 How HTML is structured

HTML consists of standard ASCII (American Standard Character Interchange) text surrounded by html tags. Tags are instructions which tell a web browser how to display a document. HTML tags are enclosed within < and > symbols so your editor knows that anything within those symbols needs to be interpreted as an instruction rather than being displayed as text. HTML files are saved with an .html or .htm extension.

1.1 Paired tags

Most HTML tags are paired, meaning that they have a tag to indicate the beginning of a command and another tag, with a forward slash (/) inside it, to indicate where that command should end.

E.g. HTML documents begin with

```
<HTML>
```

to indicate the beginning of the document and finish with

```
</HTML>
```

to indicate the end of the document.

You should avoid overlapping tags as this may cause problems in some browsers.

I.e.

```
<TAG1><TAG2> .... </TAG1></TAG2>
```

is incorrect and some browsers may ignore both tags

The correct HTML would be

```
<TAG1><TAG2> .... </TAG2></TAG1>
```

Some tags are not paired. E.g. if you indicate a carriage return with a break tag
, there is no need to indicate where the carriage return finished. It all happens in the one spot.

1.2 Tag attributes

Many HTML tags contain one or more attributes that can give extra information to tell the browser how to interpret the tag. In most tags, the attributes are optional.

E.g. `<P ALIGN="center">`

When a tag is using attributes, the name of the tag is written first, followed by a space. Then the name of the attribute and an equal sign. Next is the attribute itself enclosed in quotation marks. Single quotes can be used but double quotes are more common. Most browsers will accept attributes without quotation marks but it is best to include them as in the example below.

```
<TAG ATTRIBUTE1="value1" ATTRIBUTE2="value2"> </TAG>
```

When a tag uses attributes, they only need to be written in the opening tag. A closing tag will end a tag as well as all of its attributes.

It doesn't matter if you use upper case tags `<HTML>` or lower case tags `<html>`. It is, however, a good idea to be consistent. The WWW Consortium recommends writing your tags and tag attributes in uppercase letters. That will make it easier for you to recognise and edit your tags later since uppercase text stands out from normal text. Some editors allow you to specify whether the tags they insert are in uppercase or lowercase characters.

Note

In these exercises we'll use uppercase tags so it's easier to spot them among the rest of the text in the pages.

You can use your editor to insert many of the common tags for you when needed. Many editors will also colour code the tags in your document so you can tell them apart from the regular text. If you are looking at a website you can use your web browser's view source option to view that page's html.

Exercise 1**Look at the source of an html document**

- 1 Open your web browser
- 2 The designated home page will load
- 3 Select the view source option from the View menu or by clicking the right mouse button on the page itself (if you're using a Macintosh computer, hold your mouse button for about a second)
- 4 Take a moment to look at the html (it's not as complex as it may look)

Tip

If you see something in another website that you like, view the source to see how they did it. It's a great way to learn new html tricks.

2 Sections of an HTML Document

An html document is divided into two main sections – the head section and the body section. The head section contains information about the document, which is covered in the page attributes section below. This is usually placed at the top of the html document (so it can load into the browser first) and won't be displayed inside your browser window. The body section contains the content of your document that will be displayed, such as text and images.

2.1 Page attributes

There are several different elements that may be included within the <HEAD> section of your document. The most important is the title of your document, which is specified by the <TITLE> tag. The title tag can be used to identify your site and will also be used to provide the text that will display in your browser's title bar.

E.g.

```
<TITLE>Steve's Home Page</TITLE>
```

Would show in the title bar as



It's a good idea to make the title descriptive as the title bar is often used by search engines to describe a website. Your title should make it clear what the site is about.

Other information is often contained in the header section of the document using meta tags.

E.g. some search engines use meta tags to provide a description of a site. If you wanted to take advantage of this feature you might put a meta tag such as the one below into your head section.

```
<META NAME="Description" CONTENT="Type a description of your site here.">
```

2.2 The body section

The content of your page needs to be between the <BODY> and </BODY> tags. It is important to make sure the content of your page is in the body section since anything before the <BODY> tag and after the </BODY> tag often won't appear in the web browser. The body tag is an example of a tag that can have attributes. Some of the attributes that can be set for the body tag are:

Attribute	Description	HTML
Background colour	Sets the background colour for the document	BGCOLOR
Text colour	Sets the colour for text on the page	TEXT
Hyperlink colour	Sets the colour for hyperlink text	LINK
Visited link colour	Sets the colour for links that have been followed	VLINK
Active link colour	Sets the colour for links that are active (when they're clicked)	ALINK
Background graphic	Specifies what graphic file will be used as the background	BACKGROUND

E.g.

```
<BODY BGCOLOR="white" TEXT="black" LINK="blue" VLINK="purple" ALINK="red" BACKGROUND="back.gif">
</BODY>
```

This would give you the following

White background with a graphic file called “back.gif” tiled across the page.

Black text. Blue hyperlink text. Followed links will be purple and active links will be red.

Notice that a background colour has been specified even though the background graphic will cover it. There is a good reason for that. A lot of websites are designed with light coloured text on a dark background **like this**. If the page’s background graphic doesn’t load for some reason, the text on the page can become very difficult to read, like the text in the paragraph below, unless you have also specified a dark background colour.

Can you read this very easily?

Another thing to consider is that a page’s background colour is often one of the last things to load in a browser while the page background colour is usually one of the first things to appear. If you don’t specify a background colour, then visitors to your site might not be able to read anything until the page has finished loading, even though most of the text had been on the screen for a little while.

The point to remember is to specify a background colour that is fairly similar to your background graphic. Of course it’s also common sense to make sure that any background graphics you use aren’t too colourful or it may be too hard to read text on it.

Note	In HTML, US spelling must be used for words like COLOR, CENTER and the colour GRAY otherwise they won’t be recognised by the web browser.
-------------	---

Exercise 2

Set up an html document

1 Open your HTML editor

You should have a blank document to begin with

2 Enter the following HTML.

```
<HTML>
<HEAD>
<TITLE>Cannington Cougars Football Club Website</TITLE>
</HEAD>
<BODY BGCOLOR="white" TEXT="black" LINK="blue" VLINK="purple" ALINK="red"
BACKGROUND="back.gif">

</BODY>
</HTML>
```

3 Save the file as **index.html**. Make sure the file called **back.gif** has also been saved in that same location.

4 Preview your document in your web browser. To do this, you can open your web browser, go to your file menu and then open the index file you just saved. Many HTML editors also have a built in preview option.

3 Structuring Your Document

Exercise 3

Adding text to your site

1 With the **index.html** file still open in your editor, type the following text between the two BODY tags.

Cannington Cougars Football Club Website
Welcome to the website of the Cannington Cougars Football Club

The HTML should now look like the sample below

```
<HTML>
<HEAD>
<TITLE>Cannington Cougars Football Club Website</TITLE>
</HEAD>
<BODY BGCOLOR="white" TEXT="black" LINK="blue" VLINK="purple" ALINK="red"
BACKGROUND="back.gif">
```

Cannington Cougars Football Club Website
Welcome to the website of the Cannington Cougars Football Club

```
</BODY>
</HTML>
```

2 Save and preview the document

You'll notice that the text is all on the same line. We'll sort that out next.

3.1 Paragraphs & line breaks

In the previous exercise we discovered something about html. A web browser won't recognise line breaks unless we specify them. It also won't recognise more than one blank space so that if you typed;

Big space

It would appear in your web browser as

Big space.

We specify line breaks with the paragraph and line break tags.

The Paragraph tag `<P>` `</P>` will create a new paragraph and will usually create a little blank space before and after the paragraph. It usually works without the closing tag but it is best to get in the habit of using a closing tag.

A division tag `<DIV>` `</DIV>` can also be used to create a new paragraph. The difference is that it doesn't create a blank space before the paragraph the way a `<P>` tag will. A division tag may also be used to group sections of a page for use with styles (covered in a later exercise). The division tag does require the closing tag or it won't work

The `<P>` and `<DIV>` tags are both referred to as Block Tags, since they can both create a block of text on the page.

The break tag `
` inserts a line break in a paragraph without beginning a new paragraph and doesn't require a closing tag.

Exercise 4**Line breaks**

Insert paragraph and break tags in your index.html document so that the text you typed previously looks like the text below. The parts that you will add are highlighted with bold text. The rest of your HTML should remain the same.

```
<P>Cannington Cougars Football Club Website  
<BR>Welcome to the website of the Cannington Cougars Football Club</P>
```

Save and preview the document. It should look like the example below

**3.2 Headings**

HTML was originally intended to define the structure of a document, and headings are useful for this purpose. There are 6 heading levels in html with 1 being the largest so that the most important heading should use a <H1> tag; subheadings should use <H2> tags etc. The exact sizes they appear as will vary from one browser to another. Here is an example of how they may look. You should use headings to logically define sections of your document under headings and sub-headings. Since headings are a type of block tag, it is not necessary to begin a heading with a <P> tag. The heading tag takes the place of a paragraph tag when headings are being used.

<H1>Level 1 heading</H1>

<H2>Level 2 heading</H2>

<H3>Level 3 heading</H3>

<H4>Level 4 heading</H4>

<H5>Level 5 heading</H5>

<H6>Level 6 heading</H6>

3.3 Horizontal rules

A horizontal `<HR>` rule may be used to place a dividing line in your document and is a good way to separate different sections. It will usually appear as 3D style engraved line. You can see a basic Horizontal rule used below



Some of the attributes that can be included in a `<HR>` tag are:

WIDTH	Can be specified in pixels or percentage of page width <code><HR WIDTH="150"></code> or <code><HR WIDTH="80%"></code>
ALIGN	Specifies if the line will be aligned to the left, center (use US spelling) or right <code><HR ALIGN="center"></code>
SIZE	Specifies the height (thickness) of the line in pixels <code><HR SIZE="5"></code>
NOSHADE	Normally a horizontal rule will appear as an engraved line. Specifying this option will make it appear as a solid colour. <code><HR NOSHADE></code>

Note	Some browsers also recognise a colour attribute Eg. <code>COLOR="red"</code>
-------------	---

3.4 Pre-formatting

We have learnt that your web browser won't recognise line breaks and spaces unless you specify them with html. The only exception is when you've used a `PRE` tag. The `PRE` tag will display your text in a fixed width font just as you typed it so

```
<PRE>
  First Line
    S e c o n d L i n e
</PRE>
```

Would display in your web browser as

```
First Line
  S e c o n d L i n e
```

People often turn to the `<PRE>` tag as an easy way to lay out the information on a page. This is not a good idea since many browsers will not correctly show any formatting or images within a preformatted area. It is generally only used to present text where spacing and indentation need to be preserved, such as programming code.

3.5 Aligning text

There are several ways of aligning text within html. By default, text will be left aligned.

Any tag that creates a block of text is referred to as a block tag. <H1>, <P> and <DIV> are all examples of block tags. Each of these tags can use an ALIGN attribute to align the contents of the block of text. With this method you can align a block of text to the centre, left or right. Some examples are shown below.

```
<P ALIGN="center">All of this text would be aligned to the centre of the page</P>
```

```
<H2 ALIGN="right">This text would appear as a level 2 heading aligned to the right</H2>
```

```
<DIV ALIGN="center">This text would appear in the centre of the page</DIV>
```

Note that it is not necessary to include the ALIGN attribute in the closing tag. When you close a tag you also close all of its attributes.

Note <DIV ALIGN="center"> </DIV> is exactly the same as <CENTER> </CENTER>. CENTER was introduced by Netscape before they added support for the HTML 3.0 DIV element. It was retained in HTML 3.2 on account of its widespread use.

4 Special Characters

Many characters (such as < > & ") are used by your browser to represent html tags. So what if you want one of these special characters to actually display in your web page? You can use certain codes in your html to represent these characters.

Similar codes may also be used to show characters (such as © ® ¶ ™) that can't be typed normally. Below is a list of some of the more common codes and how they will appear in your browser. Most HTML editors allow you to easily insert special character codes from a menu without the need for typing them. However for characters you need frequently, it can be just as easy to simply type them as you go.

 	non-breaking (blank) space	"	"	(quotation mark)
&	& (ampersand)	©	©	(copyright logo)
<	< (less than symbol)	>	>	(greater than symbol)
®	® (registered symbol)	™	™	(trade mark symbol)
½	½ (half fraction)	¶	¶	(paragraph symbol)

5 3.5 Logical vs Physical Formatting Tags

Originally, html was only intended to define the structure of a document, not how it actually looked. For this reason, most of the oldest html tags are logical. E.g. a document may have some text formatted as a level 1 heading. What is a level 1 heading? According to html, it's the most important heading in a document. But what does it look like? Some web browsers might interpret a level one heading as being bold, 18-point size text. Other browsers might interpret a level 1 heading to be underlined 14-point text. Logical tags simply define a type of format without specifying exactly how it looks.

Physical tags on the other hand, specify exactly how the text will look. If you wanted your main heading to be bold text and to be a certain size, you would use physical tags to specify that.

In many cases a physical tag has been created to display the same way that a particular logical tag usually displays. Since most browsers these days recognise physical tags, they are often the better choice for getting the look you want although certain browsers, such as speech based browsers for visually impaired users, will work better with logical tags that are more descriptive in nature.

6 Some Common Formatting Tags

Now we will go through some of the tags that may be used to format text. Remember that some older browsers may not support some of these tags.

6.1 3.6.1 Bold

` `

Physical tag that will display text as **bold**

` `

Logical tag typically displayed as **bold**

6.2 3.6.2 Italic

`<I> </I>`

Physical tag that will display text in *italics*

` `

Logical tag that will emphasise text, typically displayed in *italics*

6.3 3.6.3 Underline

`<U> </U>`

Physical tag that will display text underlined

6.4 3.6.4 Other logical tags

The description next to each tag shows how the tag often displays text though the way these tags will display text may vary from one browser to another.

DFN *defining instance of the enclosed term*

CODE used for extracts from program code

KBD used for text to be typed by the user

CITE *used for citations or references to other sources*

6.5 3.6.5 Other physical tags

TT typewriter style monospaced text

STRIKE ~~strike through text style~~

BIG places text in a large font

SMALL places text in a small font

SUB places text in lowered subscript style

SUP places text in raised superscript style

6.6 3.6.6 The font tag

One of the most commonly used formatting tags in standard html is the tag. With its various properties it allows you to specify several types of formatting at once. These are some of the properties that can be set in the font tag:

- **SIZE** specifies the size where 1=smallest and 7=largest. Size 2 or 3 will usually be close to normal text size. The exact size may vary from one browser to another.
- **COLOR** specifies the colour of the text (NOTE: you need to use US spelling).
- **FACE** specifies the font (such as Arial) that will be used. You can specify an alternate font in case the person viewing your site doesn't have the one you specified. Simply type the name of each font separated by commas. If the first font isn't available the next will be used. If none are available, the default system font will be used.

Here is an example of a font tag.

```
<FONT SIZE=2 COLOR="red" FACE="Comic Sans MS, Arial">Sample text</FONT>
```

Result - **Sample Text**

Note: HTML 4 introduced styles which are a much better way of formatting text than using the **FONT** tag. Styles have made the **FONT** and some other formatting tags tag pretty much redundant so don't get too attached to it.

Exercise 5

Creating a basic html document

Now we will put to use several of the things we've covered so far.

- 1 Modify your html in the BODY section of your document so it looks like the html below. Note how the use of blank spaces makes the HTML easier to read and edit. Save and preview when you're done.

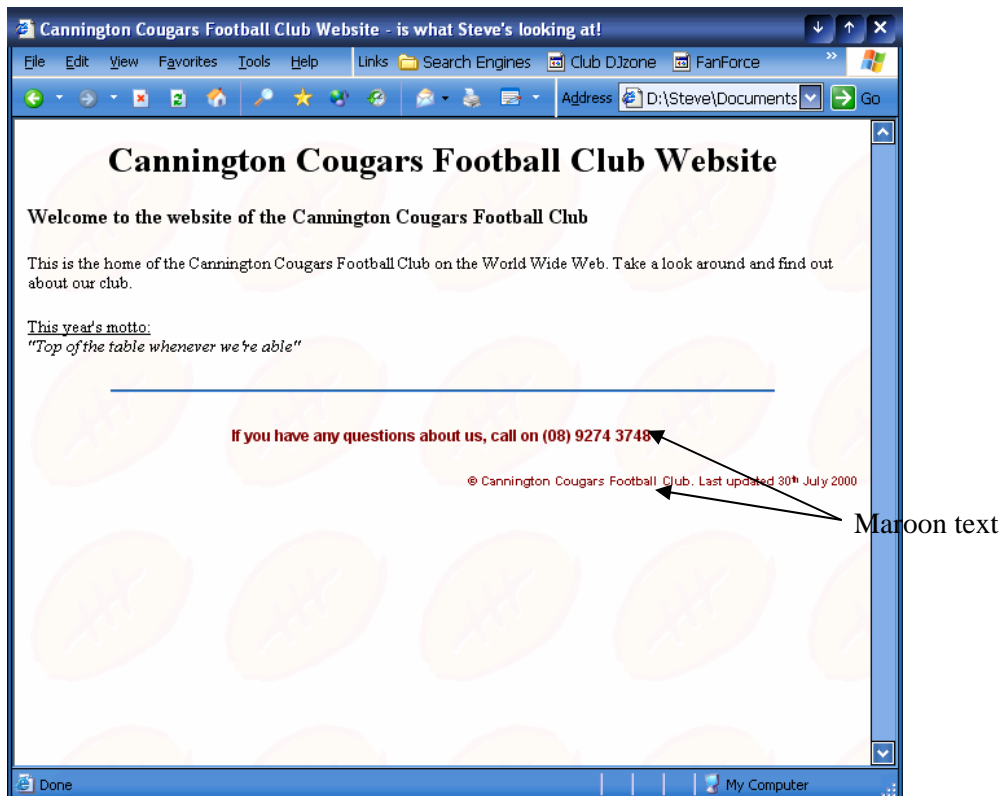
```
<H1 ALIGN="center">Cannington Cougars Football Club Website</H1>
<H3>Welcome to the website of the Cannington Cougars Football Club</H3>
<P>This is the home of the Cannington Cougars Football Club on the World Wide Web. Take a look around
and find out about our club.</P>
<P><U>This year's motto:</U>
<BR><I>"Top of the table whenever we're able"</I></P>
```

```
<HR WIDTH=80%>
```

```
<P ALIGN="center">
<FONT SIZE="2" COLOR="maroon" FACE="Arial"><B>
If you have any questions about us, call on (08) 9274 3748
</B></FONT>
</P>
```

```
<P ALIGN="right">
<FONT SIZE="1" COLOR="maroon" FACE="Arial"><B>
&copy; Cannington Cougars Football Club. Last updated 30<SUP>th</SUP> July 2003
</B></FONT>
</P>
```

The resulting document should look like the one below.



7 Lists

7.1 Bullet lists

Bulleted lists (unordered lists) are a list of items that begin and end with the UL tag. Each item in the list begins with an LI (list item) tag. A close LI tag is optional. An unordered list will typically take the form of several items with a • at the beginning of each line. If nested lists are used (lists inside lists) then different symbols may be used for the nested lists. E.g.

- Item 1
- Item 2
 - Item a
 - Item b
- Item 3

You may change the symbol used in the list by adding a TYPE attribute to either the LI or UL tag. Adding the attribute to the UL tag is usually better since that will effect the whole list rather than just one list item.

I.e.

- with <UL TYPE=disc> this is the default option
- with <UL TYPE =square>
- with <UL TYPE =circle>

E.g.

This html

```
<UL>
<LI> Item 1
<LI> Item 2
<LI> Item 3
</UL>
```

Produces this list

- Item 1
- Item 2
- Item 3

Note

 is a block tag so it can be aligned the same as a <P>, <DIV> or <H1> tag.

7.2 Number lists

Numbered lists (ordered lists) are similar to bulleted lists. They are used to display a list with items numbered. With an ordered list the OL tag is used. List items use the LI tag just as list items in a bulleted list.

Like bulleted lists, you can use the TYPE attribute to change the symbol used for each list item.

I.e.

```
1      with <OL TYPE=1> (1,2,3..)    this is the default option
a      with <OL TYPE=a> (a,b,c..)
A      with <OL TYPE=A> (A,B,C..)
i      with <OL TYPE=i> (i,ii,iii..)
I      with <OL TYPE=I> (I,II,III..)
```

E.g.

This html

```
<OL TYPE=A>
  <LI> Item 1
  <LI> Item 2
  <LI> Item 3
</OL>
```

Produces this list

```
A Item 1
B Item 2
C Item 3
```

Tip

If you want your list to start on a certain number, you can specify that number in a **START** attribute. E.g. `<OL START="2">`

7.3 Definition lists

Definition lists are another way to list information. To understand what they do, think of the way information is listed in a dictionary. Usually the term being defined (definition term) will be listed followed by the definition for the term (definition data). In HTML, the definition term normally displays like a regular paragraph while the definition data is usually indented.

The list begins and finishes with the DL (Definition List) tag. Each definition term begins with the DT (Defined Text) tag and its corresponding definition begins with the DD (Definition Data) tag. Browsers will typically display the definition text on one line with the definition data indented on the next line. You can have more than one DD tag following a DT tag if you need to have more than one paragraph's worth of definition data. DT and DD tags are both block tags.

E.g.

This html

```
<DL >
  <DT>Definition Term 1
  <DD>Definition Data Text for this item
  <DT>Definition Term 2
  <DD>Definition Data Text for this item
  <DD>A second definition data line for this item
  <DT>Definition Term 3
  <DD>Definition Data Text for this item
</DL>
```

Produces this list

```
Definition Term 1
  Definition Data Text for this item
Definition Term 2
  Definition Data Text for this item
  A second definition data line for this item
Definition Term 3
  Definition Data Text for this item
```

Exercise 6

Working with lists

- 1 Create a new blank document in your HTML editor
 - 2 Create the main document tags as shown in exercise 2 on page 4 and enter *Cannington Cougars Football Club Website – Upcoming Events* as the page title (remember that the page title is in the header section i.e. Above the <BODY> tag).
 - 3 In the body section, put in the text *Upcoming Events* and make it a centre aligned level 1 heading. The HTML should look like the following and should be placed directly below the <BODY> tag.

```
<H1 ALIGN="center">Upcoming Events</H1>
```
 - 4 Below the heading you just entered, put in the HTML below to create a numbered (ordered) list.

```
<OL TYPE="i">
  <LI>Tuesday – training at 7pm
  <LI>Friday – BBQ at 6pm
  <LI>Saturday – Games from 2pm-6pm
</OL>
```

← Note that this will give us a list with Roman Numerals
 - 5 Save the file as events.html and preview it. Your page should have a centre aligned heading followed by a list like the one below.
 - i Tuesday – training at 7pm
 - ii Friday – BBQ at 6pm
 - iii Saturday – Games from 2pm-6pm
 - 6 We will now create another page based on the one we just saved. We will do this by saving it with a different name and then modifying it. That will save us from having to repeat some steps.
 From the **File** menu select **Save As...** and call the file contacts.html. We will now change this new copy of the Upcoming Events page and make it into a Contact Details page.
 - 7 Change the text in the title and heading by replacing *Upcoming Events* with *Contacts*.
 - 8 Delete the numbered list and replace it with a bulleted list that looks like the one below. If you can't remember how to create a bulleted (unordered) list, refer to page 13 or ask your teacher for assistance.
 - Phone: (08) 9274 3748
 - Fax: (08) 9274 3747
 - Email: cougar@footy.net.au
 - 9 Save the file and preview it
- Continued on the next page

- 10 From the **F**ile menu select **S**ave **A**s... and call the file who.html. This page will contain information about members of the club.
- 11 Change the text in the title and heading by replacing *Contacts* with *Who's Who*.
- 12 On this page we'll create a definition list that looks like the one below (make sure the definition terms are bold like the ones below).
- Manager** ← Definition term (formatted in bold text)
Jeff Simpson ← Definition data
- Head Coach**
Tony Jefferson
- Assistant Coach**
Bob Miller
- Public Relations**
Sandra Moore
- 13 Save this page and preview it.

Formatting and lists Revision Questions

1. What is the difference between `<P> </P>` and `<DIV> </DIV>`?
2. What are the three types of lists used in HTML?
3. What is the difference between logical tags and physical tags?
4. Think of some things that you could use a definition list for.