

ICTWEB429

Create a markup language document to
specification



© Copyright, 2015 by North Coast TAFE now

Date last saved: 26 August 2015 by Amanda Walker	Version: 1.0	# of Pages = 45
CHEC IT Teachers – Content writer and course adviser TAFE now Resource Development Team – Instructional and graphic design		

Copyright of this material is reserved to the Crown in the right of the State of New South Wales.

Reproduction or transmittal in whole, or in part, other than in accordance with the provisions of the Copyright Act, is prohibited without written authority of North Coast TAFE now.

Disclaimer: In compiling the information contained within, and accessed through, this document ("Information") DET has used its best endeavours to ensure that the Information is correct and current at the time of publication but takes no responsibility for any error, omission or defect therein. To the extent permitted by law, DET and its employees, agents and consultants exclude all liability for any loss or damage (including indirect, special or consequential loss or damage) arising from the use of, or reliance on, the Information whether or not caused by any negligent act or omission. If any law prohibits the exclusion of such liability, DET limits its liability to the extent permitted by law, to the re-supply of the Information.

Third party sites/links disclaimer: This document may contain website contains links to third party sites. DET is not responsible for the condition or the content of those sites as they are not under DET's control. The link(s) are provided solely for your convenience and do not indicate, expressly or impliedly, any endorsement of the site(s) or the products or services provided there. You access those sites and use their products and services solely at your own risk.

Contents

Getting Started	i
About this unit	i
Elements and performance criteria.....	i
Icon Legends.....	ii
Topic 1 – Analyse specifications and requirements	1
Web language	1
What is a mark-up language?	2
Mark-up languages for the web	3
Choosing your mark-up language.....	6
Topic 2 – Create document structure	11
HTML as a mark-up language.....	11
XHTML compliancy	14
Basic structural HTML tags	17
Heading tags.....	17
Paragraph tag.....	18
List tags.....	19
Table tags.....	22
Topic 3 – Incorporate and validate web page components.....	28
Text components	28
Images.....	31
Hyperlinks	33
Horizontal rule tag (component)	35
Presentation tags	36
Validate documents	37

Getting Started

About this unit

This unit describes the skills and knowledge required to use a text editor to design, create and save web pages to a given specification, rather than using an authoring tool.








It applies to individuals working as web designers and developers who interpret client briefs, generate a framework for internet information, and are resourceful in their application of technology in using a markup language.

Elements and performance criteria

Elements define the essential outcomes of a unit of competency. The Performance Criteria specify the level of performance required to demonstrate achievement of the Element. They are also called Essential Outcomes.

Follow this link to find the essential outcomes needed to demonstrate competency in this Unit: <https://training.gov.au>

Icon Legends

	<p>Learning Activities</p> <p>Learning activities are the tasks and exercises that assist you in gaining a clear understanding of the content in this workbook. It is important for you to undertake these activities, as they will enhance your learning.</p> <p>Activities can be used to prepare you for assessments. Refer to the assessments before you commence so that you are aware which activities will assist you in completing your assessments.</p>
	<p>Case Studies</p> <p>Case studies help you to develop advanced analytical and problem-solving skills; they allow you to explore possible options and/or solutions to complex issues and situations and to subsequently apply this knowledge and these newly acquired skills to your workplace and life.</p>
	<p>Discussions/Live chat</p> <p>Whether you discuss your learning in an online forum or in a face-to-face environment discussions allow you to create and consolidate new meaningful knowledge.</p>
	<p>Readings (Required and suggested)</p> <p>The required reading is referred to throughout this Learner Guide. You will need the required text for readings and activities.</p> <p>The suggested reading is quoted in the Learner Guide, however you do not need a copy of this text to complete the learning. The suggested reading provides supplementary information that may assist you in completing the unit.</p>
	<p>Reference</p> <p>A reference will refer you to a piece of information that will assist you with understanding the information in the Learner Guide or required text. References may be in the required text, another textbook on the internet.</p>
	<p>Self-check</p> <p>A self-check is an activity that allows you to assess your own learning progress. It is an opportunity to determine the levels of your learning and to identify areas for improvement.</p>
	<p>Work Flow</p> <p>Shows a logical series of processes for completing tasks.</p>



Topic 1 – Analyse specifications and requirements

As a web developer it is your responsibility to understand the audience and purpose of the websites you are building. You need to be familiar with the qualities of different mark-up languages so that you can choose the right one for each project. This topic will familiarise you with a range of web languages so that you can make informed decisions about the best mark-up languages for your needs.

Web language

The following are some terms you should be familiar with:

Accessibility

Accessibility for web pages is a measure of how easy it is for all web users to read and interact with your website, regardless of disability or technology used.

Mark-up language

A mark-up language is a combination of text and extra information describing the text. Typically the extra information refers to the text's structure or presentation. This information is expressed using mark-up, which is mixed with the primary text. The best-known mark-up language in modern use is the basis for the web—HTML (HyperText Mark-up Language).

Metalinguage

A language that is used to describe another language, metalanguage can refer to any terminology or language used to discuss language itself. SGML (Standard Generalized Mark-up Language) is an example of a metalanguage.

Programming language

A programming language or computer language is a standardised communication technique for expressing instructions to a computer. It is a set of syntactic and semantic rules used to define computer programs.

What is a mark-up language?

A mark-up language is a combination of text and information describing the text. This extra information explains what the text is or how it should be displayed. The mark-up language is translated by a web browser into the web page that the user can see or interact with.

The classic mark-up language for the web is HTML, but there are many others – all with their own qualities and functionalities.

You will need to consider the advantages and disadvantages of different mark-up languages when choosing the right one for your website.

Remember that mark-up language is not the only element you will need to consider. Programming languages such as Java, PHP, etc. may be used in combination with a range of mark-up languages to deliver websites online.

Mark-up languages for the web

Here is a selection of mark-up languages. You will find that there are actually many more than are listed here.

HTML—HyperText Mark-up language

HTML is a very simple language used to describe the logical structure and layout of a web document. It describes which parts of the text the web browser should emphasise, which text should be considered body text and which text should be headings.

An HTML document can contain hyperlinks to other parts of the same file and to other files on different computers. An HTML document consists of text that is surrounded by HTML code.

The World Wide Web Consortium (or 'W3C'—see their website: www.w3.org) issues recommendations about standards for HTML.

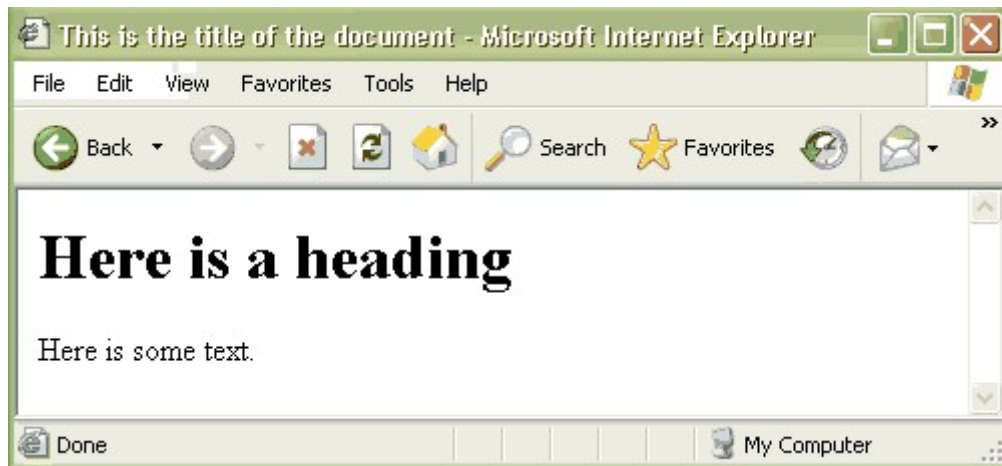
The latest version is HTML 5 based on HTML 4 there are some additional features that are intended to be included in HTML 5, these are detailed at: webdesign.about.com and stackoverflow.com

Here is a small sample of HTML code (written in a simple text editor like Notepad):

```
<html>
<head>
<title>This is the title of the document</title>
</head>
<body>
<h1>Here is a heading</h1>
<p>Here is some text.</p>
</body>
</html>
```


When viewed in a web browser the page appears as follows:

Figure 1 – Illustration of simple HTML document as shown in web browser



As you can see in the code example above, the text in the page is surrounded by mark-up—anything within brackets known as ‘tags’ (e.g. <h1> meaning ‘heading 1’) is the mark-up which tells the browser how to display the text.

HTML (and XHTML) have 3 variants, Frameset, Transitional and Strict. There are benefits of each of these variants.

	LEARNING ACTIVITIES	ACTIVITY 1
<p>Variants</p> <p>Research each of the 3 variants Frameset, Transitional and Strict and make note of when one variant might be preferred over another.</p> <ul style="list-style-type: none">a Frameset is the preference when:b Transitional is the preference when:c Strict is the preference when:		

DHTML—Dynamic HTML

Not really a language in its own right, DHTML is actually a combination of HTML (or XHTML), JavaScript and Cascading Style Sheets. It is used to create web pages with dynamic content such as animation, pop-up windows and drop down menus. A DHTML web page can produce a response to a user's action (such as a mouse click) without having to communicate with the server to have the page resent to the user's browser. For more information, go to the W3 Schools website (www.w3schools.com) and find their "Learn DHTML" tutorial - a good overview of DHTML is provided in the introduction.

XML—Extensible Mark-up language

This very versatile metalanguage (a language that describes another language) is used to describe what data is rather than just the way it looks. This means it is a semantic language rather than just a presentation language (like HTML).

XML lets the coder create their own tags to describe the data.

XML is a complement to HTML. It can be used to describe the data, while HTML can be used to format and display the data. Languages for handheld devices such as mobile phones are based on XML.

For more information, go to the W3 Schools website (www.w3schools.com) and find their "Learn XML" tutorial - a good overview of XML is provided in the introduction.

XHTML—Extensible HyperText Markup language

XHTML is a combination of HTML and XML.

XHTML has begun to replace HTML. It is a stricter and cleaner version of HTML, it is typically thought to be easier to learn, simpler to maintain and it is XSL ready.

WML—Wireless Mark-up Language I

Used for handheld devices such as mobile phones and PDAs (Personal Digital Assistants).

SGML—Standard Generalized Mark-up language

SGML is a metalanguage – perhaps 'the' metalanguage, as all web mark-up languages are based on it. HTML, XML and others are actually simplified applications of SGML, which is very difficult to code with.

Other mark-up languages

There are many other specialised mark-up languages in use - each has been specifically designed for a particular use. Some examples include:

- > MathML - an application of XML for representing mathematical symbols and formulae.
- > DocBook - for technical documentation.
- > SVG (Scalable Vector Graphics) - an application of XML for representing two-dimensional vector graphics.
- > Open eBook - standard e-book format based on the XML format.
- > XBRL (eXtensible Business Reporting Language)—an emerging XML-based standard to define and exchange business and financial performance information.
- > MusicXML—an open, XML-based music notation file format.
- > RSS—a family of XML file formats for Web syndication used by (among other things) news websites and weblogs.

Choosing your mark-up language

With so many mark-up languages to choose from, how do you know which is the right one for your web document? As with any project you undertake, you should start by looking at what is needed and the options available to you.

Website purpose

The key element in choosing an appropriate mark-up language will be the purpose of the website.

If you are building a single 'brochure-style' website that displays straightforward information including graphics, some Flash animation and maybe some sound, you could use straightforward HTML or perhaps XHTML. Most small websites are constructed this way.

If however, your site contains financial or scientific information, you may consider employing MathML in your site. Writing formulae (for example fractions and math equations) will be faster in a specialised XML-friendly application such as 'MathType' (from Design Science: www.dessci.com) than trying to code in HTML.

If your site relies heavily on graphic design and animated menus and buttons you may consider using DHTML to liven up the user experience with interactive elements.

The flexibility of XML allows developers to create their own unique tags and classes of information. This is one of the reasons that XML in various forms is used in the creation of many larger dynamic websites.

Remember that your mark-up language will work in tandem with the programming language you choose (such as JavaScript, PHP, etc.) This aspect of web design is not covered in this learning pack.

Stakeholders

Stakeholders are people who will be affected by your website and can influence it but who may not be directly involved with doing the work. You must take the needs of stakeholders into account when you are planning a website and choosing a mark-up language.

If you are employed by an organisation to build a web page the stakeholders may include:

- > management
- > marketing
- > technical support staff
- > legal advisors
- > audience of website users

Some or all of these people will help determine the specifications and appropriate mark-up language by describing the kind of functions the website needs to perform. Access to technical resources may also mean access to people with additional programming skills as well - can you use these people in your web project?

Even if you are building a personal website just to show holiday photos to your distant relatives, there are still stakeholders to consider—mainly your relatives.

- > What are their needs?
- > Who are the stakeholders in your website?
- > Audience needs - Think about the audience of your web site. There are a range of basic questions you should ask when planning a new site—these may include the following:
 - > Who are your website users?
 - > What do your users want to do on your website (eg purchase goods, make bookings, find information, etc.)?
 - > How comfortable are they with using the web?
 - > What kind of computer will your user have?

- > How will accessibility for vision-impaired users be addressed?
- > What about accessibility for users with slow modem connections?
- > Are there any business advantages in using features such as animation and sound?
- > Will the site need to be accessed using a range of devices (for example handheld devices or mobile phones)?
- > If your site requires the latest plug-ins (e.g. Flash) how will the user find them?

Determining how to meet your audience's needs and expectations will help to define the specifications and type of mark-up language you will need to use on your site.

Standards

You will need to comply with certain web standards if you want your site to be accessed by the broadest possible range of users. Remember that you users will have a range of computer platforms, operating systems and web browsers. The manufacturers of these different systems will have made efforts to comply with agreed web standards.

Standards for the web are developed by the World Wide Web Consortium (W3C)—an international group including input from member organisations, full-time staff, and the public. W3C's mission is: 'To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web'. The W3C website (www.w3.org) is the hub for standards information and includes technical guidelines, educational information and code-checking tools called 'validators' that allow developers to ensure their sites meet required web standards.

You may also want to look at examples of current 'best practice' in web design. Take note of websites that you like or are similar in function to the site you intend to build—think about the technologies and mark-up they employ and their useability. Also take a look at sites showcasing recent designs such as:

Webby Awards: www.webbyawards.com

Cool Home Pages: www.coolhomepages.com

NetGuide Australian Web Awards: www.techradar.com

Project constraints

Your choice of mark-up language will inevitably be affected by the resources you have at your disposal. You will need assess and balance the amount of money, resources, time etc. that you have available for your project. Starting up an XML project from scratch will require more resources and planning than building a simple HTML site. However if you build in HTML because of initial time constraints, but the site really requires XML and database integration, you may have wasted your effort and need to rebuild your site anyway.

This is why it is important to fully assess you stakeholders' and users' needs before you start.

Available technology and skills

In a similar vein you need to assess what technology and skills you have available to you. You may personally be a competent HTML programmer but have no knowledge of MathML or RSS implementation. This does not necessarily make HTML the right choice for your project.

With a thorough analysis of the site requirements you may have identified that MathML and RSS would greatly benefit the site. Use this to argue for more resources from within your organisation or of you are working on your own—seek advice from programmers who are familiar with these technologies and employ them if you can.

If you want to learn the required skills yourself and have the time, start a small project to implement these skills. Take a course or make use of the extensive range of online tutorials and references available free on the web. Software that assists in implementing different mark-up languages (such as Dreamweaver: www.macromedia.com) often have tutorials built in to help you get started.

There are many mark-up languages to choose from when creating web documents. Different languages perform different functions and offer various advantages and disadvantages. Knowing your audience and the purpose of your website are the keys to choosing the right one for your task.

**Client needs**

In the 'Analyse specifications and requirements' stage we gather the client's requirements. It is important that you, as web designer, know exactly what your clients' needs are before you commence with the actual development. Can you provide reasons why?

Why is it important to document those requirements? Would it be a good idea just to have it all in your head? Why wouldn't this be a good idea?

If you are unsure of the answers to these questions take a look at the following sites and make sure you understand the importance of analysis and client acceptance of your documented system requirements:

- > <http://www.ayantek.com>
- > <http://www.mindtools.com>



Topic 2 – Create document structure

HTML as a mark-up language

HTML stands for Hyper-Text Mark-up Language. HTML consists of two parts:

- 1 Text
- 2 Markup (formatting) tags that tells the browser how to format the text

Markup tags are used to format the **text**.

**Tag exercise**

Open up notepad (**Start-> Accessories->Notepad**).

In notepad type:

Hello World

Save the file onto your desktop giving it the name "hello world.htm". The htm extension tells the operating system that it is a web page file.

Now double click on the file. The default browser should open. You should see the text "Hello World" printed out in the browser.

Close the browser.

Now, back in notepad, replace the existing text with:

```
<h1>Hello World</h1>
```

Once again, save the file and reopen in the browser. What do you notice?

You should notice that the `<h1></h1>` tags marks up the enclosed text, "Hello World", as a size 1 heading.

This is pretty much what HTML is all about: formatting plain text using tags! The hard part is that there are many tags.

From this point on, try all the example code using the process described above.

HTML tags can contain further HTML tags.

For example,

```
<h1><i>Hello World</i></h1>
```

marks up the enclosed text "Hello World" as size 1 heading as well as italicizes it.

`<i><h1>Hello World</h1></i>` will give the same results.

As you work through this unit guide, a valuable supporting tutorial can be found at <http://www.w3schools.com/html/default.asp>

HTML attribute/value pairs

HTML attribute/value pairs provide additional formatting information.

For example,

```
<h1 align="center">Hello World!!!</h1>
```

not only marks up the enclosed text "Hello World!!!" as a size 1 heading but aligns the text in the center of the page.

Comments

A comment is used to add comments to your HTML. Comments are not displayed in the browser.

For example,

```
<!-- Display Hello world in bold -->
<b>Hello World!!!</b>
```

The comment – "Display Hello world in bold" – does not show in the browser. Its purpose is simply to describe what your code does. Why may this be useful?

Other example uses of comments are:

```
<h1><!-- heading goes here--></h1>

<!-- image goes here -->

<b>Price Incl. GST: </b> <!-- $ value goes here -->
```

The following illustrates the use of HTML comments to supply instructions how to do something. For example:

```
<!-- start list 1-->
<ul>
    <!-- start list item 1--><li> <!-- list item 1 goes here-->
</li> <!-- end list item 1-->
```

```

        <!-- start list item 2--><i> <!-- list item 2 goes here-->
</li><!-- end list item 2-->

        <!-- to add another list item copy/paste another list item here
-->
</ul>
<!-- end list 1-->

<!-- to add another list copy/paste another list here -->

```

Comments are very useful in templates where they can be used as

- > placeholders
- > explaining some code if the meaning of the code is not evident
- > giving instructions how to do something

XHTML compliancy

We will be making sure that our HTML is XHTML compliant.

To make your code XHTML compliant:

- 1 ALWAYS start off your notepad file with the following code before typing anything else.
Simple paste it in.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <title></title> <!--Document title goes here. This shows in the title
bar at the top of the browser-->
  </head>
  <body>
    <!-- Most of the HTML code goes here. This code will be rendered
in the browser document window -->
  </body>
</html>

```

- 2 Make sure that you type ALL your code inside the body section of the code.

The only text that you should write outside of the body section is in the **title** section of the code. Only text, **NO markup should go in here**. This text will display as the title of the web page in the browser. Always make sure that you supply a title.

- 3 When you write your code, always abide by the following rules...

- > Include the XHTML skeleton as described above
- > All HTML tags must have a closing tag, e.g.

`<h1>Hello World` is not correct but

`<h1>Hello World</h1>` is correct as it has a closing tag

There are some tags in the HTML specification that do not need a closing tag, e.g., the `<p>` (paragraph) tag, but XHTML requires it so a closing `</p>` is mandatory

- > XHTML attribute values must always be quoted, for example,

`<h1 align=center> Hello World</h1>` is incorrect

but

`<h1 align="center">Hello World</h1>` is correct as the value of the align attribute is correctly quoted

- > All tags, attributes and predefined values are in lower case.

for example, `<h1 ALIGN="CENTER"> Hello World</h1>` is incorrect but

`<h1 align="center"> Hello World</h1>` is correct as the name and value of the align attribute is in lower case.

- > XHTML elements must be properly nested, for example,

`<h1><i>Hello World</h1></i>` is incorrect as `<h1>` closes before `<i>` does but

`<h1><i>Hello World</i></h1>` is correct as `<i>` closes before `<h1>` does. **Remember this: If you open a tag close it first before you close any other tag.**

- 4 To ensure that your HTML is XHTML compliant use a validation tool such as the one located at <http://validator.w3.org>.
- a Go to the above site. Click on the 'Validate by Direct Input' tab.
 - b Paste your code into the text area
 - c Click the "Check" button
 - d If you abide by the above rules you should see a green congratulations page. If not, check your code against the above rules and try again.

REMEMBER

From this point on, paste in the skeleton shown in Figure 1 into your notepad first before writing any other code.

Always validate once you have finished writing your code.

HTML5

HTML5 is the latest revision of HTML 4.01 and XHTML 1.0. XHTML will still continue to be supported, but it will be subsumed into HTML 5.

HTML5 will not be covered in this unit as

- > It is not yet an official standard
- > Browsers currently only minimally support HTML5

HTML5 includes new tags for better structure as well as new tags to make it easier to add media content such as video and audio. It even comes with a canvas tag that allows you to draw graphics directly onto a web page.

Because XHTML is subsumed into HTML5 your documents should validate HTML5 without the need to make changes to your code other than the document type declaration, of course.

A HTML 5 tutorial can be found at www.w3schools.com

Advantages of XHTML

The advantages of XHTML are that:

- > Browsers are less error prone as syntax requirements are stricter than HTML.
- > Pages load faster as less time is required to interpret incorrect HTML code
- > Works in all browsers that conform to XHTML standard
- > Can be rendered in other devices that adhere to the standard, that is, increased device portability

Basic structural HTML tags

There are three different types of markup. They are:

- 1 **Structural** markup which defines the **purpose** of text.
- 2 **Web page components** which defines the **content** that goes into a web page's structure, for example, an image with text wrapped around it (as a component) that goes into a paragraph. Be aware, though, that a web component such as a list can be interpreted as both a structuring device as well as a web page component depending how you look at it.
- 3 **Presentational** markup defines the **appearance** of the text regardless of its purpose.

In this section we'll focus on structural tags. We'll look at the others in the next section.

Heading tags

We have already visited `<h1>` in our introduction above. It is a structural tag. For example, the html code

```
<h1>Cat</h1>
```

says that the text 'Cat' is to be interpreted as a level 1 heading. It is important to understand that as a structural tag it has nothing to say about how it is to be presented. We can change its presentation (for example, italicize it) but it is still to be interpreted as a heading.

We also have the `<h2>`, `<h3>`, `<h4>`, `<h5>` and `<h6>` tags. These are to be interpreted as subheadings. Naturally, they present themselves in such a way as to suggest what level of heading they are, but it's their structural interpretation that's being established not their presentational.

Take:

```
<i>Cat</i>
```

This, unlike the heading tags, defines the presentation of the text that it encloses. For example:

```
<i>Cat</i>
```

simply establishes how the text 'Cat' is to be presented, in italics, not how it is to be interpreted. Whilst there are many presentation tags, these are generally being phased out in favour of Cascading Style Sheets (CSS) but not the subject matter of this unit

Each of the heading tags have "align" as an attribute. It can take the values left, center or right, so `<h2 align="right">` will align the heading to the right of the page.

Never suggest structure through presentation. For example, never style plain text as a heading. This is a big no-no. It is important from an accessibility point of view as presentation alone does not suggest the text's function that a structural tag does.

Let's look at some other commonly used structural tags:

Paragraph tag

The `<p>` tag is another structural tag. It says that any text enclosed within it is to be interpreted as a paragraph. It also has an align attribute that takes the same values as does the heading tags.

For example, what will the following snippet of HTML code do?

```
<p align="right"> Some text here, Some text here, Some text here, Some  
text here, Some text here, Some text here</p>
```

List tags

What is a list?

Following is an example of a Word list (or bullets). It should give you an idea what a list is.

Colors

- Red
- Blue
- Green

Un-ordered lists

Un-ordered lists are lists that don't have any particular order, for example,

- the first item
- the second item
- the third item

Table 1 – Un-ordered list

HTML source	Result in browser
Items: <pre> the first list item the second list item the third list item </pre>	Items <ul style="list-style-type: none">• the first item• the second item• the third item

Note the use of indentation in the source HTML to suggest that the individual items belong to same un-ordered list. This is important for readability.

Ordered lists

Ordered lists, as it suggests, are lists that are ordered in some way, for example, to bake a cake do the following:

- 1) Add flour
- 2) Add sugar

- 3) Add egg yolk
- 4) Stir mixture
- 5) Place in oven

Table 2 - Ordered list

HTML source	Result in browser
<p>To bake a cake do the following:</p> <pre> Add Flour Add sugar Add Egg yolk Stir mixture Place in oven </pre>	<p>To bake a cake do the following:</p> <ol style="list-style-type: none"> 1) Add flour 2) Add sugar 3) Add egg yolk 4) Stir mixture 5) Place in oven

Note the use of indentation to suggest that the individual items belong to same un-ordered list

Nested lists

Nested list are not really all that complicated. A list item contains a list itself. For example:

- the first list item
- the second list item
 - first nested item
 - second nested item
- the third list item

Table 3 – Nested list

HTML source	Result in browser
<p>Example of nested list</p> <pre> the first list item the second list item first nested item second nested item the third list item </pre>	<p>Example of nested list</p> <ul style="list-style-type: none"> • the first list item • the second list item <ul style="list-style-type: none"> ○ first nested item ○ second nested item • the third list item

Choosing your marker

For un-ordered and ordered lists you can choose the marker you want to use by assigning the appropriate pre-defined value to the type attribute of the ordered / unordered list.

For un-ordered lists valid values are:

type = {disc, circle, and square}, for example,

```
<ul type="disc">
```

For ordered lists valid values are

type = {"A", "a", "I", "i", "1"}, for example,

```
<ol type="1">
```

Try each of the above attribute values to get an idea what the associated markers look like.

Definition lists

Definition list are like those you find in a dictionary. It allows you to list terms and their definitions, for example:

the first term

its definition

the second term

its definition

the third term

its definition

Notice that the definition is slightly indented.

Table 4 – Definition list

HTML source	Result in browser
<pre> <dl> <dt>the first term</dt> <dd>its definition</dd> <dt>the second term</dt> <dd>its definition</dd> <dt>the third term</dt> <dd>its definition</dd> </dl> </pre>	<p>the first term</p> <p>its definition</p> <p>the second term</p> <p>its definition</p> <p>the third term</p> <p>its definition</p>

Note the use of indentation to suggest that the individual items belong to same un-ordered list.

Table tags

What is a table?

A table consists of rows and data cells. The cells can be numbered from left to right and from top to bottom.

Figure 2 – Example of a table

	COLUMN 1	COLUMN 2	COLUMN 3
ROW 1	CELL 1	CELL 2	CELL 3
ROW2	CELL 4	CELL 5	CELL 6
ROW3	CELL 7	CELL 8	CELL 9

Creating HTML tables

Following is the code to create a simple 1 celled table. (Make sure that you try the code out!)

```
<!-- New table -->
<table border="1">
    <!-- New row -->
    <tr>
        <!-- New cell -->
        <td>Cell 1</td>
    </tr>
</table>
```

The above table consists of three parts:

A table – blue

A row – green

A data cell – orange

The following code creates an HTML table containing one row with two data cells:

```
<!-- New table -->
<table border="1">
    <!-- New row -->
    <tr>
        <!-- New cell -->
        <td>Cell 1</td>
        <!-- New cell -->
        <td>Cell 2</td>
    </tr>
</table>
```

The following code creates an HTML table containing two rows each containing two cells:

```
<!-- New table -->
<table border="1">
  <!-- New row -->
  <tr>
    <!-- New cell -->
    <td>Cell 1</td>
    <!-- New cell -->
    <td>Cell 2</td>
  </tr>
  <!-- New row -->
  <tr>
    <!-- New cell -->
    <td>Cell 3</td>
    <!-- New cell -->
    <td>Cell 4</td>
  </tr>
</table>
```



LEARNING ACTIVITIES

ACTIVITY 4

Creating tables

- a Create a table with 2 **rows** and 2 data **cells**
- b Create a table with 4 **rows** and 2 data **cells**
- c Create a table with 4 **rows** and 4 data **cells**

Table attributes

The following code illustrates the use of table attributes. Make sure that you try the code.

```
<table border="2" width="80%" align="center"
cellpadding="5" cellspacing="5">
  <tr>
    <td bgcolor="orange" align="center">This cell content is aligned to
the center of the cell</td>
    <td bgcolor="yellow" align="right">This cell content is aligned
to the right of the cell</td>
  </tr>
</table>
```

Table attributes:

- > width=x (pixels or "%") is used to set the width of the table.
- > border=n is used to set the thickness in pixels of a table.
- > cellspacing=n is used to set the number of pixels between adjacent cell borders.
- > cellpadding=n is used to set the number of pixels between the content of a cell and its cell's border. This prevents content within adjacent cells from running into each other.
- > bgcolor =color is used to color the background of the entire table.

Table data cell attributes:

- > align={top, bottom, center, left, right, absmiddle} is used to control the placement of contents within a cell.
- > bgcolor = {red, green, ...} is used to control the background color of a cell.
- > width=y (pixels or "%") is used to define the width of the cell.
- > height=y (pixels) is used to define the absolute height of a cell but is nonstandard and not well supported.



LEARNING ACTIVITIES

ACTIVITY 5

Table attributes

As an exercise, create a few simple tables and experiment by assigning table and data cell attributes with different values.

Column spanning

In the following code the first column spans 2 columns.

```
<table border="1">
  <tr>
    <td colspan="2">This is cell 1 that spans cell 2 and cell 3</td>
  </tr>
  <tr>
    <td>This is cell 2</td>
    <td>This is cell 3</td>
  </tr>
</table>
```

When you span a column if you define a colspan of 2 you will need to define 1 less cell in that row. If you define a colspan of 3 you will need to define 2 less cells in that row. And so on...

Row spanning

In the following code illustrates an example of row spanning.

```
<table border="1">
  <tr>
    <td rowspan="2">This is cell 1 that spans cells 2 and 3</td>
    <td>This is cell 2</td>
  </tr>
  <tr>
    <td>This is cell 3</td>
  </tr>
</table>
```

When you span a row if you define a rowspan of 2 you need not define the corresponding cell in the next row. If you define a rowspan of 3 you need not define the corresponding cells in the next 2 rows. And so on ...

Nesting tables

Data cells may contain entire tables. These are referred to as nested tables.

```
<table border="1" bgcolor="orange">
  <tr>
    <td>The next cell contains a nested table with two cells</td>
    <td>
      <table border="1" bgcolor="yellow">
        <tr>
          <td>Cell 1 of nested table </td>
          <td>Cell 2 of nested table </td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```



Topic 3 – Incorporate and validate web page components

Web Components are objects that you place within the structure of a web page. Note that some components can themselves be structures, for example, a list and a table can themselves be components.

Text components

Blocks of text are themselves web page components. Text must be entered in the context of some containing structure.

So,

```
<body>  
Hello World  
</body>
```

is **incorrect** as the 'Hello World' is not contained in any document structure.

```
<body>
<h1>Hello World</h1>
</body>
```

```
<body>
<p>Hello World</p>
</body>
```

are valid though as the 'Hello world' is now contained within some structural tag.

Line break tag

The line break tag is relevant when you add text to a web document. If you start on a new line in your HTML source it will not start on a new line when displayed! You will need to use the `
` tag if you wish to break to a new line.

So,

Two
Lines

Will display as

Two lines

in your browser.

But

Two

Lines

will display correctly as

Two
Lines

In your browser

Non-breaking space

Multiple white spaces are not preserved. They are collapsed as a single white space.

Try:

```
<p>Hello      World</p>
```

What's the result?

For example:

```
5 &gt; 4 &lt; 6
```

will display

```
5 > 4 < 6
```

Other useful character entities are:

- > ` ` - space
- > `©` - copyright symbol
- > `½` - ½ symbol
- > `¼` - ¼ symbol

Images

Image tag

Images are an important component of web pages. To add an image is easy.

An image can be inserted using the `` tag. The `` tag has an attribute named "src" that takes the name of the image file to load as its value, for example:

```

```

Daffodiles.jpg is an image file. In this case it's to be found in the same directory as the HTML file.

It's good practice though to locate all images in a folder with the name "Images". You can then load the image from this folder with:

```

```

or

```

```

Here the value "Daffodiles" is supplied to the "alt" attribute. When you hover your mouse cursor over the image you should see a tool tip appear with this value in it.

Also, if the browser does not support images, the image is replaced by this text. The "alt" attribute is mandatory. The "alt" attribute is important from an accessibility viewpoint.

Setting image width and height

Try:

```

```

This will set the image's width to 300 and its height to 300.

This will 'squash' the image into the supplied dimensions and may not look good.

Maintaining the aspect ratio

When setting the width and height it is best to preserve the aspect ratio of the original image.

This simply means that the height of the image divided by the width of the image is the same.

So, if we want to halve the width then we will need to halve the height as well. Likewise if we wish to double, reduce by a third etc.

The width and height of the daffodil image is 842 and 549 respectively. So if we want to halve the width (that is, make it 421) we have to half the height as well that is, make it 275). Doing this the aspect ratio is maintained ($842/549 \approx 421/275$)

Try halving the width (make it 421) but leave the height the same (549). Load the image into the browser. Now halve the width (make it 275) and load the image into the browser. Can you see the difference?

Making images float

Go to http://www.w3schools.com/css/css_float.asp for an example that demonstrates how to let an image float to the left or right of a paragraph.

Aligning Images

Go to http://www.w3schools.com/tags/att_img_align.asp for an example that demonstrates how to align an image within a line of text.

Hyperlinks

Hyperlinks allow the user to go to another page.

- > An HTML link can be created using the "<a>" tag.
- > The "<a>" tag has an attribute named "href". You assign the web page you want to link to, to this attribute.

So, to link a page to "Page2.htm" you would use the following HTML code:

```
<a href="Page2.htm">click here</a>
```



LEARNING ACTIVITIES

ACTIVITY 6

Create a link to a new page

- > Create a new file and name it "Page2.htm".
- > In the body type the text "This is page 2".
- > Create another file and name it "Page1.htm".
- > In the body type: To read more go topage 2.
- > Load the page. You should see a link. Click on it. What happens?

Linking to someone else's page

Links can also be used to link to pages that are located on other web sites.

Try the following:

```
<a href="http://www.google.com.au">click here</a> to go to Google.
```

Note: the value for "href" **must** contain the **full** Internet address of the page.

Load the page and click on this link. What happens?

Links using images

Images can be used as links instead of text.

Try:

```
<a href="http://www.adobe.com/products/acrobat/readstep2.html"></a>
```

Load the page and click on this link. What happens?

Opening a linked page in a new window

What if we want to open the linked page in its own window?

You can achieve this by passing the value "_blank" to the "target" attribute of the "<a>" tag.

Try:

```
<a href="http://www.adobe.com/products/acrobat/readstep2.html"  
target="_blank" ></a>
```

Load the page and click on this link. What happens?

Including an email link

It is also possible to link to an email address. This will open Outlook, filling in the "to" field with the email address you supply in the link.

Try the following HTML code:

```
<a href="mailto:someperson@somedomain.com.au">Email me!</a>
```

Load the page and click on this link. What happens?

Link to downloadable files

You can also use links to files that users can download. These are files that cannot be properly displayed by the browser and have file extensions such as ".exe", ".zip" and so on.

Try:

```
<a href="ZippedFiles.zip">Click here</a> to download my zipped files.
```

Load the page and click on this link. What happens?

Anchor links

Anchors have two parts:

- 1 An anchor
- 2 A link to the anchor

When you click the link it will take you to the location on the current page where the anchor is defined.

Example:

HTML code for an anchor:

```
<a name="top"></a>
```

HTML code for the link:

```
<a href="#top"></a>
```

Now, when a user clicks on the link the user will be taken to the location of the anchor within the page.

Horizontal rule tag (component)

The `<hr />` tag is used to draw a line to separate areas on a page.

It also has some useful attributes. Some of them are

- > `width=n` - sets width in pixels
- > `width=n%` - sets width as a percentage of the width of the page
- > `size=n` - sets size
- > `noshade="noshade"` – turns shading on

For example, try:

```
<hr width="50%" size="10" noshade="noshade" />
```

Try some other attribute/value combinations.

Presentation tags

Presentation tags define the appearance of web content. Generally the uses of these are discouraged in favour of the use of Cascading Style Sheets. CSS is covered in another unit and will not be covered here.

However, for the sake of completeness, here are some HTML presentation tags:

The font tag

The `` tag is also a useful tag. Useful attributes are:

- > • `size=n` - sets the size of the font
- > • `face={Arial, Verdana, ...}` - sets the font
- > • `color={red, blue, ...}` - sets the color of the font

For example, try:

```
<font face="arial" size="4" color="red">This text is in Arial, has size 4 and is in  
red</font>
```

Note: SIZE can have a value from 1 to 7 where 3 is the "normal" size. A value less than 3 will gives a smaller size, greater than 3 a larger size.

Try some other attribute/value combinations.

Other presentation tags

Other common ones are the ``, `<i>` and `<u>` tags these bolding, italicizing and underlining text respectfully.

Other presentation tags can be found at <http://www.w3schools.com/tags/default.asp>

Validate documents

Be aware that just because a web page renders OK in a web browser does not mean that the code is XHTML compliant.

To ensure that your code is XHTML valid go to <http://validator.w3.org> to validate your document.

It is also important that you test your documents under different browsers, especially the more popular ones, for compatibility.

The outcomes of the above results should be documented.