

TOPIC 3: HTML CODING

T3.1) Introduction to HTML

What is HTML?

HTML is a language for describing web pages.

- HTML stands for **Hyper Text Markup Language**
- HTML is not a programming language, it is a **markup language**
- A markup language is a set of **markup tags**
- The purpose of the tags are to **describe page content**

It is the standard protocol for formatting and displaying documents on the World Wide Web.

Note

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

HTML is a formal Recommendation by the World Wide Web Consortium (W3C) and is generally adhered to by the major browsers, Microsoft's Internet Explorer and Netscape's Navigator, which also provide some additional non-standard codes.

Markup refers to the sequence of characters or other symbols that you insert at certain places in a text or word processing file to indicate how the file should look when it is printed or displayed or to describe the document's logical structure. The markup indicators are often called "tags." For example, this particular paragraph is preceded by a:

A **tag** is a generic term for a language element descriptor. The set of tags for a document or other unit of information is sometimes referred to as markup

T3.2) HTML structure

HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
<body>
<h1>This a Heading</h1>
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
</body>
</html>
```

Example Explained

What you just made is a skeleton html document. This is the minimum required information for a web document and all web documents should contain these basic components. The first tag in your html document is `<html>`. This tag tells your browser that this is the start of an html document. The last tag in your document is `</html>`. This tag tells your browser that this is the end of the html document.

The text between the `<head>` tag and the `</head>` tag is header information. Header information is not displayed in the browser window.

The text between the `<title>` tags is the title of your document. The `<title>` tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

The text between the `<body>` tags is the text that will be displayed in your browser.

The text between the `` and `` tags will be displayed in a bold font.

HTM or HTML Extension?

When you save an HTML file, you can use either the `.htm` or the `.html` extension. The `.htm` extension comes from the past when some of the commonly used software only allowed three letter extensions. It is perfectly safe to use either `.html` or `.htm`, but be consistent. **mypage.htm** and **mypage.html** are treated as different files by the browser.

How to View HTML Source

A good way to learn HTML is to look at how other people have coded their html pages. To find out, simply click on the View option in your browsers toolbar and select Source or Page Source. This will open a window that shows you the actual HTML of the page. Go ahead and view the source html for this page.

T3.3) HTML Tags

Tags, Attribute and Elements

Html Tags

A **tag** is a generic term for a language element descriptor. The set of tags for a document or other unit of information is sometimes referred to as markup.

HTML tags are the hidden *keywords* within a web page that define how your web browser must format and display the content.

HTML markup tags are usually called HTML tags

- HTML tags are keywords (tag names) surrounded by **angle brackets** like <html>
- HTML tags normally **come in pairs** like and
- The first tag in a pair is the **start tag**, the second tag is the **end tag**
- The end tag is written like the start tag, with a **forward slash** before the tag name
- Start and end tags are also called **opening tags** and **closing tags**

HTML Elements

"HTML tags" and "HTML elements" are often used to describe the same thing.

But strictly speaking, an HTML element is everything between the start tag and the end tag, including the tags:

HTML Element:

<p>This is a paragraph.</p>

HTML Attributes

Attributes provide additional information about HTML elements.

HTML Attributes

- HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in name/value pairs like: **name="value"**

Attribute Example

HTML links are defined with the <a> tag. The link address is specified in the **href attribute**:

Example

This is a link

Attributes allow you to customise a tag, and are defined within the opening tag, for example:

 or <p align="center"> ... </p>

Attributes are often assigned a value using the equals sign, such as border="0" or width="50%", but there are some that only need to be declared in the tag like this: <hr noshade>.

Most attributes are optional for most tags, and are only used when you want to change something about the default way a tag is displayed by the browser. However, some tags

such as the `` tag has required attributes such as `src` and `alt` which are needed in order for the browser to display the web page properly.

HTML tags Types

1. Types

HTML tags can be of two types. They are

1. Paired Tags
2. Unpaired Tags

Paired Tags:

A tag is said to be a paired tag if the text is placed between a tag and its companion tag. In paired tags, the first tag is referred to as *Opening Tag* and the second tag is referred to as *Closing Tag*.

Example

```
<i>This text is in italics. </i>
```

Note: Here `<i>` is called opening tag. and `</i>` is called closing tag.

Unpaired Tags:

An unpaired tag does not have a companion tag. Unpaired tags are also known as *Singular* or *Stand-Alone* Tags.

Example

```
<br> , <hr>
```

etc. These tags does not require any companion tag.

2. Tags Based on their utility

We can differentiate tags based on the purpose they used. Basically we have three types here

Formatting tags

We manage the size of the font, underline part of the text, make the text bold etc by using tags like ``, `<u>`, `` etc.

Page Structure tags

Description, title, head, body etc. are part of the page structure tags. They are part of the basic html page and does not directly affect the formatting of text or image.

Control tags

Form tags, Script tags, Radio buttons etc are part of the control tags

HTML tags classification

1. Essential HTML Tags

There are four sets of HTML tags that are needed to form the basic structure for every HTML file:

- <html></html>
- <head></head>
- <title></title>
- <body></body>

Definition - <html> </html>

This basically defines the document as web page. It also identifies the beginning and end of the HTML document. All other tags must fall between the html tags.

Header - <head> </head>

The header contains information about the document that will not appear on the actual page, such as the title of the document, the author, which stylesheet to use and also meta tags.

Title - <title> </title>

The title tag defines the title that will appear in the title bar of your web browser.

The title must appear between the head tags.

Body - <body> </body>

The body tags contain all the information and other visible content on the page. All your images, links and plain text must go between the <body> and </body> tags.

These four tags are special. There must only be one set of each and they must be in the correct order like in the example below. The fun and creative part comes when using the basic tags for adding content and headings.

Example:

Below is a basic html document. Notice that everything falls between the html tags, the title appears within the head of the document, and that the body comes after the head.

```
<html>
  <head>
    <title>My Page Title</title>
  </head>
  <body>
```

This is where all my web page content goes!

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

2. HTML Basics

Once you have the essential tags sorted out, there are a number of basic tags you will use a lot in order to add content such as:

- Headings and Subheadings
- Paragraphs of Text
- Links to Other Pages
- Images or Photographs

All of these must appear between the `<body>` and `</body>` tags, and you can learn more about them by following the links shown.

Headings

```
<h1> A Heading </h1>
```

Use headings for titles and subtitles, and make some text stand out from others. See text formatting tags.

Paragraphs

```
<p> Some text </p>
```

Most content on a simple web page will appear in paragraphs or sections. See division tags.

Links

```
<a href="home.html"> My homepage </a>
```

Links are necessary for users to jump from one page to another. See linking tags.

Images

```

```

Adding your holiday photos or other images to your web page is fairly simple. See image tags.

Example:

Below is an example of some of the basic tags explained above

```
<html>
  <head>
    <title>this is the title</title>
  </head>
```

```
<body>
  <h1>My Heading</h1>
  <p>This is the first paragraph of text.</p>
  <p>This is the second paragraph of text.</p>
  <p>An image:  </p>
  <p>A link: <a href="http://www.simplehtmlguide.com"> html guide
  </a></p>
</body>
</html>
```

3. Sections, Divisions & Lines

These are the tags used to divide your page up into sections. Effective use of these tags will mean that the page has a good structure and layout, making it more user friendly and easier to read.

Division - <div> </div>

The `div` tag defines a section or division within a HTML file. It typically contains headings, paragraphs, tables or other elements that need to be grouped together. Commonly used with css by setting the `<div class="?">` attribute to set the look and feel of a section of your web page.

Paragraph - <p> </p>

Used to define paragraphs of text, much like you would see in a book. A lot of text can appear within the `<p>` and `</p>` tags, and browsers will automatically wrap the text onto the next line once it reaches the edge of the screen. When another `<p>` tag is used to start the next paragraph, the browser will add some blank space between the paragraphs. It has the following attributes:

`align="?"` - Alignment of text in the paragraph: `left`, `center` or `right` (*)

`width="?"` - Paragraph will occupy a fixed width or percentage of the page, default 100%

Span -

Used to group inline elements together, such as a few words within a sentence, in order to apply a css style to those words only. The `span` tag can be used within `div` and `p` tags as it does not create a new horizontal block boundary.

Line Break -

Equivalent to one carriage return, it is used to start text on a new line. Multiple `
` tags in a row will create a large vertical space on a web page.

Horizontal Line - <hr>

The horizontal rule, often referred to as the HTML line separator tag, creates a horizontal line commonly used to visually separate sections on a page. It has the following attributes:

`width="??%"` - The line will occupy a fixed width or percentage of the default 100% width.

`color="#???????"` - Colour of the line (*)

`noshade` - Prevent the 3D 'etched' look and create a flat, solid line separator.

No Breaks - <nobr> </nobr>

If for some reason you want text to continue in one straight line, and not to wrap at the edge of the screen screen, you can use the `nobr`. Note: this will force a user to scroll to the right to see the rest of the line, which is considered bad design.

Example:

Below is an example of some of the tags explained above

```
<html>
  <body>
    <p align="right">The first paragraph of text, aligned to the
    right.</p>
    <p>This text is now in the second paragraph.
    <br>A new line, but still part of the second paragraph.</p>
    <p>A third paragraph, and then a horizontal line</p>
    <hr>
    <p>Some modified horizontal lines:</p>
    <hr width="50%" size="8" align="center">
    <hr noshade>
  </body>
</html>
```

4. Text Formatting Tags

The following HTML tags are used to format the appearance of the text on your web page. This can jazz up the look of the web page, *however*, too much variety in the text formatting can also look displeasing.

Header - <h?> </h?>

There are 6 levels of headings available, from `h1` for the largest and most important heading, down to `h6` for the smallest heading.

**Bold - **

The text in between the tags will be bold, and stand out against text around it, the same as in a word processor.

Italic - <i> </i>

Also working the same way as a word processor, italics displays the text at a slight angle.

Underline - <u> </u>

Again, the same as underline in a word processor. Note that html links are already underlined and don't need the extra tag.

Strike-out - <strike> </strike>

Puts a line right through the centre of the text, crossing it out. Often used to show that text is old and no longer relevant. Also works by using <s> </s> instead.

Preformatted Text - <pre> </pre>

Any text between the `pre` tags, including spaces, carriage returns and punctuation, will appear in the browser as it would in a text editor (normally browsers ignore multiple spaces)

Source Code - <code> </code>

Text is displayed in a fixed-width font, commonly used when showing source code. I have used it on this site, along with stylesheets, to show all tags.

Typewriter Text - <tt> </tt>

The text appears to have been typed by a typewriter, in a fixed-width font. (*)

Block Quote - <blockquote> </blockquote>

Defines a long quotation, and the quote is displayed with an extra wide margin on the left hand side of the block quote.

Small - <small> </small>

Instead of having to set a font size, you can use the `small` tag to render text slightly smaller than the text around it. Useful for displaying the 'fine-print'.

Font Colour - (*)

Change the colour of a few words or a section of text. The 6 question marks represent the hex color code, see this list of colours and codes for some samples. (*)

Font Size -

Replace the ? with a number from 1 to 7 to change the size of the font. One being the smallest and seven the largest. (*)

Font Size Change -

For an immediate change of font size with respect to the font size preceding it, this tag increase or decreases the size of the font by the number you specify. Eg: Some Text (*)

Change Font Face -

To show text in a particular font, use the font name such "Helvetica" or "Arial" or "Courier". Be aware that using some fancy font from your computer means that the person viewing that page must also have that font installed on their computer too, otherwise it will look totally different to them. (*)

Centre - <center> </center>

A useful tag, as it says, it makes everything in between the tags centred (in the middle of the page). (*)

Emphasis -

Used to emphasize text, which usually appears in italics, but can vary according to your browser.

Strong Emphasis -

Used to emphasize text more, which usually appears in bold, but can vary according to your browser.

5. Images

Images are used in HTML documents to one: make the page visually effective and two: display information. Images can also be used as links, but this is discussed in the topic on linking.

Although images are good for a number of things, a page with too many images often looks too cluttered and can take too long to load, which can be frustrating, and as a business aspect it could lose clients.

An image -

To display an image you need to specify the URL of the image using the `src` attribute, replacing `url` with the filename of your image. There are several ways this can be done:

`src="picture.jpg"` - the filename if the image is in the same directory as the html file.

`src="images/picture.jpg"` - a relative path when the image is in another directory.

`src="http://www.simplehtmlguide.com/images/photo.jpg"` - a full URL can also be used.

Alternate Text - ``

The `alt` attribute defines the text shown in place of an image when the image cannot load. This is actually a required attribute for valid html, and should briefly describe what the image normally would.

Image Size - ``

An image will normally be shown actual size, but by using the `width` and `height` attributes you can change the displayed size. You can specify the size in pixels or as a percentage. Tip: specify the size using the actual size of the image in pixels to force browsers to allocate space for the image before it is even loaded, ensuring your page layout remains the same with or without images displayed.

Border - ``

Add a border by specifying the thickness in pixels. You can also set `border="0"` to remove the border added when images are used as links. (*)

Image Alignment - ``

By default an image appears at the place specified in the html code(as with any other tag). However, you can align an image with the surrounding text or paragraph by setting any of `align="left | right | top | bottom | middle"`. (*)

Spacing - ``

Adjust the whitespace (or runaround space) around an image, in pixels. Use `vspace` to adjust the vertical spacing above and below, or `hspace` for the left and right sides. (*)

Example:

Show an image using html

```
<html>
<body>
<p></p>
</body>
</html>
```

6. HTML Linking Tags

Learn how to create links on your web page. Links allow you to jump from one page to another by clicking on the link text. You can also jump to places on the same page (called fragments), to different sections of your site, or to another web site alltogether.

Basic Link - link text

There are two main parts to a link tag: the text a user can click, and the web address they go to if they click it. The bit between the <a> and tags is the link text, and is generally displayed in blue and underlined by web browsers. The href="*url*" part is the web address, where *url* can be set in several ways:

href="example.html" - another page in the current directory

href="example/page.html" - a relative location

href="http://www.example.com/page.html" - a full address (URL).

Link to a Fragment - link

It is often usefull to link to other places on the same webpage, such as other sections or chapters further down the page. The technical term for this is linking to a Fragment, where browsers will automatically try and scroll to that part of the page.

Fragments first need to be defined somewhere in a webpage by giving them a name, for example , then links to this fragment are created by using the hash (#) character: Link. To link to a fragment on another page you would simply append the fragment name to the address, for example: href="example.html#fragment_name".

Target Window - link

You may not always want to link to a page and have it load up over the one you are currently viewing. Thats where the target attribute comes in handy. By setting the target="_BLANK" the page you link to will load up in a new window (or new tab in some newer browsers). Similary, "_self", "_parent", or "_top" will open the link in the current window, the parent window (used with frames) or the top level window, respectively.

Image as a Link -

By placing an image tag between the `<a>` and `` tags, you can turn an image into a link, and clicking on that image will then load the referenced page. You may notice that the image gets a blue border just as link text became underlined. This can be resolved by setting the `border="0"` attribute of the image, or using css.

Email Link - ``

A special kind of link, the `mailto` notation link instructs the browser to compose and email to the specified address using the default email program. It but does not actually send any emails automatically. You can also set a subject for the email by using `email me`. You may notice that I have used this type of link over on the contact me page.

7. HTML Lists

Learn how to create lists on a web page. Lists are the preferred way to display items one after the other, instead of using `
` tags. Lists have a tag to start and end the list itself, as well as a tag for each item in the list.

There are three types of lists, *ordered*, *unordered* and *definition* lists.

Unordered Lists

An unordered list is a bulleted list, similar to the menu on the right (although the menu has been altered using stylesheets to use images instead of the standard bullets.)

Define Unordered List - ` . . . `

Use the `` tags to define the start and end of an unordered list. A number of list items (`li` elements) will go within the `ul` tags.

Unordered List Item - ` some item `

Add the text for each item in between some `` and `` tags. Each list item must have its own `li` tags.

Bullet Type `<ul type="disc | circle | square">`

By default a browser will show a round bullet. This can be changed by using the `type` attribute of the `ul` tag, which will change the bullet type for the entire list.

Item Type `<li type="?">`

You can set the type of bullet for an item in the middle of the list by setting the `type` attribute of an `li` tag.

Example:

```
<html>
  <body>
    <ul>
      <li>An item</li>
      <li>Another item</li>
    </ul>
  </body>
</html>
```

Ordered Lists

This list is used to create an indexed list, such as a numbered or alphabetical list.

Define Ordered List - ` ... `

Use the `` tags to set the start and end of the list. A number of list items will then go between the ordered list tags.

Ordered List Item - ` an item `

Each item must use the `` tags the same as with an unordered list. But this time the browser will number each item automatically, instead of showing bullets.

List Type `<ol type="A | a | I | i | 1">`

Set the type of list index by using the `type="?"` attribute. The default style is numeric, and you can also choose from upper or lowercase, alphabetic or roman numerals.

List Starting Position `<ol start="?">`

Set the starting number (or letter) if you don't want the list to start at 1 or A.

Item Value `<li value="?">`

You can set the value of an item in the middle of the list manually, if you do not want it to follow the previous letter or number. Simply set the `value` attribute of the item you wish to change. Note: subsequent items will follow the new value.

Example:

```
<html>
  <body>
    <h3> Ordered List </h3>
    <ol>
      <li>Item one</li>
      <li>Item two</li>
    </ol>
    <h3> Modified Ordered List </h3>
```

```
<ol type="A" start="6">
  <li>List item 1</li>
  <li>List item 2</li>
  <li value="12">List item 3</li>
  <li>List item 4</li>
</ol>
</body>
</html>
```

Definition Lists

This type of list is used to define and describe terms, much like a dictionary. Typically an entry in the list consists of a term, and a definition of that term. A browser will usually bold the term, and indent the definition.

Define a Definition List - `<dl> </dl>`

Set the start and end of a definition list. All entries go within the `dl` tags. Each entry will usually consist of one `dt` and one `dd` element.

Definition Title - `<dt> </dt>`

The title of a term being defined. Note: you may have a term with no definition, or multiple terms with the same definition.

Definition Description - `<dd> </dd>`

The definition of a term. Note: you can have multiple definitions for a single term.

Example:

```
<html>
  <body>
    <dl>
      <dt>Term 1</dt>
      <dd>Definition of term 1</dd>
      <dt>Term 2</dt>
      <dd>Definition of term 2</dd>
    </dl>
  </body>
</html>
```

8. HTML Tables

Table tags are used for displaying spreadsheet-like data neatly formatted in rows and columns. They can also be used to design page layouts by placing content into invisible rows and columns of a 'table'.

Table - `<table> ... </table>`

Used to define a table, it contains all row and column tags along with their content. Think of it like the `body` tag, although there must always be at least one row in a table. It has some attributes to define the table layout.

`border="?"` - The size of the border (in pixels) surrounding the table

`cellspacing="?"` - The space (in pixels) between each cell, eg. between rows or columns

`cellpadding="?"` - The space, or margin, between the content of a cell and its border

Table Row - `<tr> </tr>`

To start a table row, the `tr` tags must appear within the `table` tags.

Table Cell - `<td> </td>`

A table cell is where the content goes. Cells must exist within rows, where the number of cells in a row determines the number of columns in the table. Cell properties can be set using the attributes:

`align="?"` - Alignment of text in the cell: `left`, `center` or `right` (*)

`valign="?"` - Vertical alignment of the cell: `top`, `middle` or `bottom`.

`width="?"` - Specify a fixed width of a cell, by default they will only take up as much space as they need.

`colspan="?"` - Column spanning allows a cell to take up more than one column, in order to match layouts of other rows. Replace ? with the number of columns to span.

`rowspan="?"` - Row spanning, similar to column spanning, forces a cell to occupy more than one row.

`nowrap` - No text in the cell will be wrapped onto the next line. Similar to the `nobr` tag for paragraphs

Header Cell - `<th> </th>`

Similar to a table cell, a header cell must appear within a table row. Normally found in the first row, header cells are usually shown in bold and centered by the browser.

Example:

A simple table with three rows and two columns.

```
<html>
<body>
<table border="1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
<td>Cell A1</td>
<td>Cell B1</td>
</tr>
<tr>
<td>Cell A2</td>
<td>Cell B2</td>
</tr>
</table>
</body>
</html>
```

9. HTML Frames

Frames allow you to have multiple sections of the browser window, called frames, each showing their own .html file within the frame. This used to be common practice when trying to show separate sections of a site in separate sections of the browser window, such as a header at the top, navigation at the side, and the rest was page content that someone could scroll down without making the header and navigation disappear.

Frame sets are rarely used these days, as the introduction of server side scripting languages such as php and asp allow you to create content pages dynamically. The introduction of HTML5 has also provided new methods of doing page layouts without having to use frames.

Frame Set - `<frameset> ... </frameset>`

Used instead of the `body` tag, the `frameset` tag defines a group of frames. Setting the `rows` and `cols` attribute allow you to create the number of frames needed for your layout.

`rows="??, ??"` - To set up multiple frames in rows, replace the question marks by the size of each row, either in pixels or as a percentage. A * can be used as a wild card, for instance: `rows="100, *"` would give you a top frame of 100 pixels high, and a bottom frame using the rest of the screen.

`cols="??, ??"` - Similar to rows, a number of frames can be set out in columns.

`border="?"` - Frame border thickness in pixels.

`bordercolor="?"` - Colour of border between frames. [\(*\)](#)

Frame - `<frame>`

Each frame within a set will need a `frame` tag to tell it which web page to load in the frame. It uses the [attribute](#):

`src="url"` - Filename or URL of page to show in the frame

`noresize="noresize"` - The frame will not be able to be resized by a visitor

`scrolling="auto"` - Each frame will have vertical and horizontal scroll bars appear automatically when needed. You can change this by setting the `scrolling` attribute to yes, no, or auto.

`frameborder="auto"` - Individual Frame Border. Set to 0, 1 to specify whether or not that frame must have a border.

Unframed Content - `<noframes> . . . </noframes>`

Very old browsers are unable to display frames, and in this case we need to specify what these browsers should display instead of the frames. Even though this is not much of a problem anymore, it is still suggested that you specify unframed content when using frames. Anything between the `noframes` tags will not be shown in modern browsers that show framed content.

Example:

A frame example with one header row, a left & right side column, and the content in the center.

```
<html>
  <head>
    <title>Example - Frames</title>
  </head>
  <frameset rows="100,*">
    <frame name="top" src="frames_top.html">
    <frameset cols="50,*,50">
      <frame name="left" src="frames_left.html">
      <frame name="mid" src="frames_middle.html">
      <frame name="right" src="frames_right.html">
    </frameset>
    <noframes>
      <i>error to display to those who cannot see frames</i>
    </noframes>
  </frameset>
</html>
```