

## **HTML for Beginners**

### **Introduction**

#### **What is HTML?**

HTML stands for HyperText Mark-up Language. It is a programming language that allows web browsers to format and display text, images and other multimedia in the form of a web page or as a web site. There are many other more advanced web languages that allow the content to be dynamically produced on the web server but for in this workshop we will concentrate on simple HTML.

#### **What is needed to write HTML?**

Any simple text editor can be used to write HTML code. On Windows this could be Notepad, on Linux it could be KWrite, on MacOS it could be TextEdit. It is recommended that a slightly more advanced text editor be used that shows syntax highlighting. On windows the recommended program is called Notepad2 and can be downloaded from:

<http://www.flos-freeware.ch/notepad2.html>

This small, self-contained program can be run directly from a 3½" floppy disk or can be installed to a hard drive or USB stick.

What is needed for this short course?

You need to create a folder on your hard drive called "Web". This is where you will save all the web pages you create.

### **Your First HTML page**

#### **The "Hello World" Page**

As is traditional when learning a new computer programming language, the first output should say hello to the world.

Enter the following code into Notepad and save it in "Web" as "index1.htm"

Now open this web page in a web browser to see the results.

Breaking down the code

All programming controls in HTML are called "tags" and must always appear in opening and closing pairs. Tags are identified by the < and > symbols. Tags must also be nested correctly: the last opened tag must be closed first. Closing tags are identical to opening tags but include the / symbol before the tag name. Closing tags cannot contain

## Parameters.

### **<HTML> </HTML>**

This tag pair must surround everything on the page. It has no parameters. It informs a web browser that the information between the tags should be rendered as a web page.

### **<HEAD> </HEAD>**

This part of the code contains information that will not generally be rendered by the Browser.

### **<TITLE> </TITLE>**

Any text you place within this tag pair will appear in the window's title bar. In this For example, the words "Hello World" appear in the title bar.

### **<BODY> </BODY>**

This part of the code contains the information to be rendered to the browser.

### **<p> </p>**

This tag stands for paragraph. Any text placed in here will be rendered to the screen in a style defined by the browser.

## **A more advanced page.**

The previous example is particularly plain. In a real page there are many different styles of text within the body. Enter the following code into Notepad and save it as "index2.htm".

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<p>Hello World</p>
</body>
</html>
```

## **The new tags**

### **<Hx [Align="Left|Right|Center"]> </Hx>**

In these tags x must be a number between 1 and 6. These 6 heading tags allow the

programmer to have more control over the way a page looks. By default <H1> is the largest and <H6> is the smallest. The optional Align= parameter allows even more control over the page style. Note the Americanised spelling of "CENTER". The Align= parameter can be applied to all the header tags, the paragraph tag and to various other tags.

### **<U> </U>**

This tag underlines the text placed in between the tags. It is essential that this tag be Closed.

### **<I> </I>**

This tag makes the text placed in between the tags italic. It is essential that this tag be Closed.

### **<B> </B>**

This tag makes the text placed in between the tags bold. It is essential that this tag be Closed.

```
<html>
<head>
<title>An example using different styles</title>
</head>
<body>
<h1>This is style h1</h1>
<h2><u>This is style h2 underlined</u></h2>
<h3><i>This is style h3 in italics</i></h3>
<h4><b>This is style h4 in bold</b></h4>
<h5><strike>This is style h5 struck through</strike></h5>
<h6><marquee>This is style h6 scrolling</marquee></h6>
<p align="center">This is style p centred</p>
</body>
</html>
```

### **<STRIKE> </STRIKE>**

This tag strikes out the text placed in between the tags. It is essential that this tag be Closed.

### **<MARQUEE [Direction="Left|Right"] [Behaviour="Scroll|Alternate"]> </MARQUEE>**

This tag makes the text placed in between the tags scroll across the screen. It is

essential that this tag be closed. The two optional parameters allow you to control the scrolling direction and behavior. The defaults are Left and Scroll. Alternate causes the text to bounce across the screen

## Linking pages together

The most useful tag in HTML is the hyper-link. This allows pages to be linked together and allows such things as e-mail to be sent when clicking on a few words in a web page. Write the following code and save it as "index3.htm".

## The new tags

In order to use links successfully it is essential that computer directory structures are understood.

The highest level of a web site file structure is called the "site root". In the examples in this short course the site root is "C:\Web" where C is your hard drive. The files we have created so far are all in the same directory. They can therefore be called by just referring

```
<html>
<head>
<title>An example using hyperlinks</title>
</head>
<body>
<div align="right">
<p>
<a href="http://www.lboro.ac.uk" target="_blank">Loughborough University</a>
<br />
<a href="index2.html" target="_self">Examples of heading styles</a>
<br />
<a href="mailto:youremail">Send yourself an email</a>
</p>
</div>
</body>
</html>
```

to the full file name.

Imagine there is a file in "C:\\" called "upper.htm". To access this page from our own web pages we would need to go up a directory level "../" and then call the file name. So we would use the code "../upper.htm" .

Imagine we now have another directory inside the "Web" called "Images". To access the file "logo.gif" inside "Images" from a web page in "Web" we would need to call "Images/logo.gif". This is called relative linking. Directory names and file names are case sensitive in HTML.

**<a href="URL" [target="NAME"]> </a>**

The anchor tag is the most important tag so far introduced. It also appears to be the most complex. Any text between the opening and closing tags will turn into a click-able link on the web page. The href= tag is compulsory and can take the following forms:

- **Absolute links** – the href= must contain a full URL of the file you want to reach. The link must be enclosed in inverted commas and must start “http://” or “ftp://”. The file you are trying to reach a webpage or a file to be downloaded.
- **Relative links** – the href= must contain a relative URL to the file you want to reach. The link must be enclosed in inverted commas and must be described relative to the current page. e.g. if the file you want to reach is in the same folder you can just use the code href=“newpage.htm” to access it.
- **E-mail links** – the href= takes the following form:  
href=“mailto:youremail@somewhere.com”. A target parameter cannot be used with this form of tag.

The target= parameter is optional and can take one of 4 in-built names or a name defined by a <frameset> tag (see later). The in-built tags are all lower case and start with an underscore. It is recommended that targets be specified to prevent unexpected behavior. The in-built tags are:

- **\_self** – This opens the link in the current window
- **\_blank** – This opens the link in a new window
- **\_parent** – This is related to the <frameset> tag
- **\_top** – This is again used with the <frameset> tag and will open a link in the full browser window.

**<div [parameters]> </div>**

The <div> tag (same as a section in MS Word) allows you to apply a parameter such as align= to all applicable tags in between the <div> and </div> tags. This can save a lot of time when formatting complex pages.

**<br />**

The line break tag is one of the very few tags that does not have a closing tag. The tag ends in “ />” so as to comply with current standards.

## **Images**

Most web pages have some element of graphics involved in their design. Graphics are inserted using the <img /> tag.

Create a new folder under “C:/Web” called “images”. Then copy an image under the name “logo.gif” into this folder. Write the following HTML page and save it as index4.html in “Web”.

**``**

The only required parameter in this tag is the link to the image source, src=. This should be either a relative link to an image somewhere in your websites directory structure or an absolute link as used above in the link tag.

The border tag places a border around the image. Specify the width of the border in pixels inside the inverted commas.

The alt tag defines some text to display while the image is loading or if the image can't be found.

The height and width tags specify the dimensions of the image in pixels. These tags are especially useful for large images where the browser will leave a space the correct size while the image is loaded.

## **Styling Web Pages Using Cascading Style Sheets (CSS)**

There are many different ways to style web pages. However one of the most useful method is by using CSS. This allows you to write one file that is then deployed across an entire website. If you then want to change the look of a website at a future date you need to only alter one file.

```
<html>
<head>
<title>A page with an image</title>
</head>
<body>
<p>This is an image</p>

<p>This image is also a link</p>
<a href="http://www.lboro.ac.uk" target="_blank">

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

### **Linking the to the style sheet.**

To link to the style sheet the following html tag needs to be placed somewhere in between the <head> and </head> tags. Insert the following line of code into index2.htm and re-save it.

### **Writing the style sheet**

Write the following code and save it in the site root directory as style.css

This file allows you to control the style of all elements within a website. Parameters are applied in a specific hierarchy

The BODY statements are applied to all other elements unless these are specified later.  
ss">

```
body {background-color: #0000FF; color: #FFFF00; font-family: Arial, "Times New Roman";}
```

```
h1 {font-size: 16pt}
```

```
p {color: #00FF00; font-size: 12pt}
```

```
a {text-decoration: none; font-weight: bold}
```

```
a:active {text-decoration: none; font-weight: bold}
```

```
<link rel="stylesheet" type="text/css" href="style.ca: hover {text-decoration: none; font-weight: bold}
```

```
a:visited {text-decoration: none; font-weight: bold}
```

The definitions than can be applied are:

**background-color** – sets the background color for an element

**color** – sets the foreground color for an element

**font-family** - sets the font for an element. Font names with white space must be enclosed in inverted commas

**font-size** – sets the font size for an element. The units (pt) must also be given.

**font-weight** – can take the value of "bold". If not needed then omit.

**text-decoration** – can take the value of "none" or "underline".

Any other style definition such as align may also be included in the style sheet but it is recommended that this sort of fine formatting is done in the main code itself to allow greater flexibility.

## More HTML

### Lists

HTML allows you easily to create 2 types of list; Numbered (Ordered) Lists and Bulleted (Unordered) Lists. Both follow the same basic structure.

Create a web page containing the following code and save it as "index5.htm".

```
<html>
```

```
<head>
```

```
<title>A web page with lists</title>
```

```
</head>
```

```
<body>
```

```
<p>
```

```
<ol>
```

```
<li>This is an ordered list.</li>
```

```
<li>This is level 2.</li>
```

```
<li>This is level 3.</li>
```

```
</ol>
```

```
<hr width="50%" />
```

```
<ul>
```

```
<li>This is a bulleted list.</li>
<li>This is level 2.</li>
<ul>
<li>This is a sub list.</li>
<li>This is sub level 2.</li>
</ul>
<li>This is level 3.</li>
</ul>
</body>
</html>
```

**<UL [Align="Left|Right|Center"]> </UL>**

This tag creates and ends an unordered list. The closing tag is essential. The align parameter is optional. Lists can be nested one inside another.

**<OL [Align="Left|Right|Center"]> </OL>**

This tag creates and ends an ordered list. Again the closing tag is essential. The align parameter is optional. Lists can be nested one inside another.

**<LI> </LI>**

This defines each item in the list. The closing tag is not essential but should be used to maintain the current HTML standards.

## Horizontal Rule

Sometimes it is desirable to draw a line across the page. This is done using the following tag:

**<HR [Width="xx%|xx"] [Color="#rrggbb"] />**

The width parameter is optional and can either be set as a percentage of the full page width or as a fixed number of pixels. The color parameter is also optional and defines the color of the line. Both these parameters can be defined in the style sheet to give a consistent feel to the site.

## Bookmarks

A bookmark is a special type of link. It requires two parts; one part defines the bookmark, the other creates the link to it. This allows you to easily jump around on a single page or to jump to a specific place in another document.

### The bookmark placeholder



To create the place holder use the following tag.

**<A name="bookmark">Bookmark Name</A>**

Linking to the bookmark

To link to the bookmark use the following tag.

**<A href="page.htm#bookmark" target="\_self">Link Name</A>**

## Tables

Tables are a useful feature of HTML and can seem a little complex at first.

**<TABLE [Width=""][Border=""][Align=""][Bordercolor="#"]> </TABLE>**

All the parameters are optional. This tag must be called before any other table tag.

**Width=** this sets the table width as a relative width (%) or as a fixed width.

**Border=** this sets the border width in pixels.

**Align=** this aligns the table to the Left, Right or Center.

**Bordercolor =** this sets the color of the borders using a hex code as earlier.

**<TR> </TR>**

This tag starts and ends each table row. This must be called before writing any table Data.

**<TD [Align=""][Valign=""][Colspan=""][Rowspan=""][Width=""]> </TD>**

This tag encloses any table data. All the parameters are optional.

**Align=** this aligns the text in the table to Left, Right or Center.

**Valign=** this aligns the text in the cell to Top, Middle or Bottom.

**Width=** this sets the cell width as a percentage of table width or in pixels.

**Colspan=** This allows you to specify if a cell should be stretched over X number of cells.

**Rowspan=** This allows you to specify if a cell should be stretched over X number of rows.

Example table code

Write the following code and save it as "index6.htm"

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="style.css" />
<title>Table example</title>
</head>
<body>
<table width="75%" border="2" bordercolor="#FF0000" align="Right">
<tr>
```

```

<td>This is row 1, column 1</td>
<td>This is row 1, column 2</td>
<td>This is row 1, column 3</td>
</tr>
<tr>
<td>This is row 2, column 1</td>
<td colspan="2">This is row 2, columns 2 & 3</td>
</tr>
<tr>
<td valign="middle" rowspan="2">This is rows 3 & 4, column 1</td>
<td align="center">This is row 3, column 2</td>
<td align="right">This is row 3, column 3</td>
</tr>
<tr>
<td>This is row 4, column 2</td>
<td>This is row 4, column 3</td>
</tr>
</table>
</body>
</html>

```

## Scientific HTML

When writing HTML for science purposes it is often necessary to include subscripts, superscripts, Greek characters and mathematical symbols

.

**<sub> </sub>**

Any text placed in between these tags will be made subscript. The closing tag is Essential.

**<sup> </sup>**

Any text placed in between these tags will be made superscript. The closing tag is essential