Contents

Considerations

(New topic text goes here.)
Talk about

- organizing directories and files
- Relative patting
- ISP Service provider Considerations
- Image editors - File Formats supported
- Taking advantage of FlexED

Creating a simple HTML Document

To create your first home page, follow the simple steps below and watch how easy you can create your own home pages with little effort and time using FlexED.

Creating a basic document involves firstly the use of document structured elements. These are the basic elements each HTML document must incorporate. These tags or elements are the following:
<HTML> ... </HTML> Identifies that the document contains HTML elements.
<HEAD> ... </HEAD> Unordered collection of data about the document
<TITLE> ... </TITLE> The Document Title
<BODY> ... </BODY> Body of the document that contains all text, images that will be rendered.

Before The <HTML> tag, a prologue document identifier should be defined. Within the identifier is the HTML DTD (Document Type Definition), that labels the HTML document adhering to. Other considerations of the above tags is that <Title></Title> tags should be defined between the <Head></Head> tags and that all HTML document text, control references etc. that will be rendered, must be defined within the <Body></Body> tags.

Therefore, when starting a new document, you should always have a template document as follows. In FlexED, by selecting new document, all this information will be loaded.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN//3.0">
<HTML>
<HEAD><TITLE> Put_Your_Title_Here </TITLE></HEAD>
<BODY>

</BODY>
</HTML>
```

Above you can see that the Document Type has been defined. The title exists between the <Head> tags, there is a body section, and this is all contained inside the <HTML> tags. This document is now ready for all your data and information.

As for a typical simple page, we will create the following using FlexED. Initially, we will format the background color. To do this, we just double click on the <BODY> tag, and a dialog box will appear asking for the background parameters. Select the BgColor icon and click on the White Color box. Then Select OK. You have now defined your background color to be white, and the HTML code inserted in the document would be: `<BODY BGCOLOR=#FFFFFF>`. Next, you can add a heading or headline. To do this, position your cursor under the <Body> tag and select the 

button on the FlexED format Toolbar. This will insert `<H1></H1>` into the editor, and between these tags you type in your heading. Double Click on the <H1> tag, and a dialog box will appear with the H1 parameters. Here you can select center from the pull down menu, and your heading will then be centered. Following this you can select the Horizontal line  button from the format toolbar to insert a <HR> tag into the editor. This will place a line underneath the heading when rendered.

Next, type in some text under the <HR>, separating it with the <P> (Paragraph tag, located as  on the format toolbar) to create paragraphs of text. Also, pick out some words in your text which you may wish to format as bold or italics. Highlight this text and select the

(Bold) or

(Italics) button from the format toolbar. All as it takes to format your text is to highlight the selected text and press the particular button on the format toolbar for which you want to format the text with. FlexED will do the rest.
Your HTML Document Code should now look like the following:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN//3.0"><HTML>
<HEAD><TITLE> Put_Your_Title_Here </TITLE></HEAD>
<BODY BGCOLOR=#FFFFFF>
<H1 ALIGN="center">Your Heading Goes Here</H1>
<HR>
<P>
This is some text in my <B>first</B> paragraph in my first new <I>HTML document.</I>
<P>
This is some more text in a new paragraph.
</BODY>
</HTML>
```

Above, all text between the <Body> and </Body> tag will be rendered.

Creating Headlines, Typeface/Style, and Paragraph Tags

Creating Headlines

HTML defines six levels of heading. A Heading element implies all the font changes, paragraph breaks before and after, and white space necessary to render the heading. The Highest level of heading is <H1>, then <H2>, down to <H6>. These Tags enlarge and bold your selected text, to stand out from the rest of your text to give the impression of a Heading or Headline. A typical example of using the Heading Tags follows. The ALIGN Tag may also be used to place your headline in the desired spot on your document. Center, Right and Left may be used to Align your headline.

<H1 ALIGH=center>This is The Main Headline</H1>
Here is some of your text.
<H2>This is a Sub Heading of your Main Headline<H2>
Here is some more of your text.

Typeface And Style.

Use these tags to format your text in your documents. These Formatting tags are located for quick reference in the FlexED formatting Tool Bar. Below are examples of how these tags are used and references to their rendered results.

<B> ... </B> **Bold**
<I> ... </I> *Italics*
<U> ... </U> Underline
<STRONG> ... </STRONG> **Strong**
<EM> ... </EM> *Emphasis*
<CITE> ... </CITE> *Citation*
<BLINK> Text Blinks
<TT> ... </TT> `Teletype Text`
<STRIKE> ... </STRIKE> ~~Strike Through Text~~
^{...} Superscript Text
_{...} Subscript Text
<BIG> ... </BIG> Big Text
<SMALL> ... </SMALL> Small Text

Also note, multiple different tags can be used to format one section of your text.
For example, if you required a bold underlined section in italics, your HTML code would appear as follows:

```
This is an example of <B><I><U>Bold, Underlined, Italic text</U></I></B> in your document.
```

Would be rendered as:

This is an example of ***Bold, Underlined, Italic*** text in your document.

Formatting Font and Color

 The font tag allows three different attributes or parameters. These are Size, Color and Face or Font set. The FlexED Font Selection Dialog Box allows all these options to be selected by the end user. By also double clicking on the <FONT> tag, FlexED will display an options dialog box for color, size and font. See also the <FONT ...> reference for examples on using the Font tag.

Paragraphs And Formatting

The following tags can be used to format whole HTML documents or different sections of your document. These include, starting new lines, creating paragraphs, Indenting various sections of your document, defining sections of text as a certain style, and pre-formatting text in fixed width text. These tags can all be easily accessed from the FlexED HTML Toolbar. Below are the available formatting tags and references to their rendered results.

<BR ...> Insert Break or start new line.
<NOBR> Do not insert breaks
<WBR> Word Break
<PRE> ... </PRE> Pre-Formatted Text
<CODE> ... </CODE> Mono-spaced Text
<DL> ... </DL> Definition List, Indented Paragraph Style
<P> ... </P> Start a new Paragraph

<STRIKE> ... </STRIKE>

The Strike Through Element specifies that your text should be rendered in Strike Through.

```
This is an <STRIKE>Example of Strike Through Text</STRIKE> in your document.
```

Would be rendered as:

This is an ~~Example of Strike Through Text~~ in your document.
.

Creating Documents with URL's

A URL (Uniform Resource Locator) is the standard used to identify files on the Internet and World Wide web using the type of server, the host name of the computer the file is on, and the complete path to the file. It is the key to locating and interpreting information on the Internet. Its a way of describing the location of a Web resource and its content.

Standard Syntax for expressing a URL:

servicename://internethost:portnumber/resource

The following is an example using this syntax:

http://www.infoflex.com.au/index.html

TO create a URL, you must first identify the service name, the most common one being the **http** service (HyperText Transfer Protocol) - The way servers and clients communicate over the network.
Next, identify the Internet Location of the resource. This is an Internet hostname, such as www.yourcompany.com.
Finally identify the name of the resource, such as the document name, file name, or other name. An example might be Index.htm.

Examples of Different Services:

Http:
Represents a document available from a World Wide web Server. The Following is an example.
http://www.infoflex.com.au/index.htm

Ftp:
Represents a file transfer protocol service. The following is an example.
ftp://infoflex.com.au/pub/win
Note: We have not specified any filename, but rather a directory. The Ftp URL can also include a filename.

Telnet:
Telnet allows you to remotely log into a computer and use your local screen an keyboard as a terminal. The following is an example:
telnet://infoflex.com.au
Note: The Telnet URL is an exception to the three part rule; it contains only the service and the hostname.

Gopher:
The Internet Gopher service is a non-graphical, menu-driven information service, For example:
gopher://infoflex.com.au
Although it looks like we have left out the resource, by leaving this blank, weve told our Web Browser to access the default, top-level Gopher menu.

News:
This represents the UseNet news service (often called the NetNews) and is a vastly distributed bulletin board service (BBS), organised into thousands of special interest subjects. For example:
news:comp.infosystems.www.html
This is an example of newsgroups dedicated to discussions about HTML.

Wais:

Internet Wide Area Information Server protocol (WAIS) servers have indexed databases against which you can run keyword searches remotely. For Example, a WAIS IRL, dealing with educational subjects is: wais://info.curtin.edu.au:210/k-12-software

Mailto:
This represents the Internet electronic mail service (e-mail). The following is an example: mailto:info@infoflex.com.au

File:
This represents a file located on your own computer. The following is an example: file://directory/myfile.htm

For Further References See Also:

Using Anchors and Links
Creating Anchors With FlexED
HRef Builder
<A...> ... </A> Anchor

Using Anchors and Links

Hyperlinks or Hotspots are colored/highlighted words (or images) you see in Web documents. When you click them, you jump to other documents or resources on the local Web server, or services somewhere out on the Internet.

Hyperlinks are based on the **anchor** HTML tag. The following is the general syntax of a anchor.

```
<A COMMAND=target>Highlighted Text </A>
```

The Command is substituted with a Href or Name. HREF signifies a hyperlink, while NAME signifies a marked place in the rendered document. The HREF element inside an anchor tag announces that the target is the target of a hyperlink. The </A> is the end of your anchor tag.
For example:

```
<A HREF=Target URL>Highlighted Text to appear to click on</A>
```

This Code Would be Rendered as:

Highlighted Text to appear to click on

By clicking on this section when rendered will execute and load the target document.

Linking To Local Documents:
The Following is an example of linking to local documents:
<HREF=about.htm>Highlighted Text To Appear</A>

Linking To URLs:
The following is an example of linking to another URL. The link we will create will link to the Infoflex FlexED page.
<HREF=http://www.infoflex.com.au/flexed.htm>Click here to go to The FlexED Home Page</A>

Linking To Specific Places In a Document:
You can make links to various spots in one document. This would be useful, if you have an Index on the top of your page, and depending on what you click on, it will take you to the specific place in your document.
For this purpose, you use the **NAME** command instead of HREF. The Following is the syntax:
<A NAME=Spot>
This anchor represents a marker in your document to the spot where you want the hyperlink in the first document to point. For example, your target document would contain the following:
<A NAME=xmarks> X Marks The Spot
And moving back to your first document, to provide a link to directly link to the marker, you would create a normal hyperlink to a local document, but with the addition of #xmarks to reference the spot you have marked in the target directory. The example follows:
<A HREF=Sample.htm#xmarks> Click Here to go to X </A>

For Further References See Also:

Creating Documents with URL's
HRef Builder
Creating Anchors With FlexED
<A...> ... </A> Anchor

Creating Anchors With FlexED

To create an Anchor (or Link) in FlexED, select the insert Anchor button  from the FlexED Main toolbar. This will prompt you with the Anchor Editor dialog box. Select whether it will be a HREF (HyperText Link) or NAME (Reference to a Marker in a document) type anchor. Type in your link and or target. Alternatively, call the Href builder from the Anchor Editor Dialog box […]. Here, select your resource, (Http, Ftp etc.) and enter in the URL for the Resource. For example, Selecting Http as the resource, www.infoflex.com.au as the host address and FlexED.htm as the file name will create the following HTML text, once you have selected OK.

```
<A HREF=http://www.infoflex.com.au/flexed.htm> </A>
```

Then insert your Hyperlink highlighted text to appear, between the <A …..> and </A> elements. The following is an example extended on the above example:

```
<A HREF=http://www.infoflex.com.au/flexed.htm>Click Here To Go To FlexED</A>
```

This would be rendered as:

Click Here To Go To FlexED

For Further references see also:

Creating Documents with URL's
Using Anchors and Links
HRef Builder
<A...> ... </A> Anchor

Creating Lists

There various type of lists which you can create in your HTML document, Ordered Lists, unordered lists, Directory Lists, Menu Lists and Directional Lists. List may be nestled within another list if required. The purpose of creating a list is to provide data in a ordered format.

The **Ordered List** element is used to present a numbered list of items, sorted by sequence or order of importance. The Ordered list must begin with a <OL> element, immediately followed by a <LI> (list item) element. An example of an Ordered List follows.

```
<BODY>
   <OL>
      <LI>Item Number 1</LI>
      <LI>Item Number 2</LI>
      <LI>Item Number 3</LI>
   </OL>
</BODY>
```

Would Be Rendered As:

1. Item Number 1
2. Item Number 2
3. Item Number 3

The **Unordered List** element is used to present a list of items which is typically separated by white space and/or marked by bullets. The Unordered list must begin with a <UL> element immediately follow by a <LI> (List Item) element. An Example of an Unordered list is:

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

Would Be Rendered As:

- Item Number 1
- Item Number 2
- Item Number 3

A **Menu List** is a list of items with typically one line per item. The menu list style is more compact than the style of an unordered list. The menu list must begin with a <MENU> element which is immediately followed by a <LI> (list item) element:

```
<MENU>
   <LI>First item in the list.</LI>
   <LI>Second item in the list.</LI>
   <LI>Third item in the list.</LI>
</MENU>
```

A **Definition list** is a list of terms and corresponding definitions. Definition lists are typically formatted with the term flush-left and the definition, formatted paragraph style, indented after the term.

A **Directory List** element is used to present a list of items containing up to 20 characters each. Items in a directory list may be arranged in columns, typically 24 characters wide. If the HTML user agent can optimise the column width as function of the widths of individual elements, so much the better. A directory list must begin with the <DIR> element which is immediately followed by a <LI> (list item) element:

For further references of these lists see also:

<UL> ... </UL> Unordered Lists
<OL> ... </OL> Ordered Lists
<DIR> ... </DIR> Directory Lists
<MENU> ... </MENU> Menu Lists
<DL> ... </DL> Definition Lists
List Wizard Using FlexED List Wizard

Creating Pre formatted Text

Different web browsers running on different computers render your HTML text in different ways, using various fonts, breaking lines in different places and generally precluding you from doing much firm document layout. Without the use of tables, the Pre-formatted Text element presents blocks of text in fixed-width font, and so is suitable for text that has been formatted on screen.

The `<PRE>` element may be used with the optional `WIDTH` attribute, which is a Level 1 feature. The `WIDTH` attribute specifies the maximum number of characters for a line and allows the HTML user agent to select a suitable font and indentation. If the `WIDTH` attribute is not present, a width of 80 characters is assumed. Where the `WIDTH` attribute is supported, widths of 40, 80 and 132 characters should be presented optimally, with other widths being rounded up.

For further references see also:

<CODE> ... </CODE>
<PRE> ... </PRE>

Creating Tables

Tables are one of the most useful elements in HTML. Tables can be used to control the layout of your page for purposes of formatting and layout of text and images, and is especially useful in creating aligned columns of text, images and data. For instance, you can use a table to insert an image on the left hand side of your page, and have your text beside that image on the right hand side. You may also use tables to Align your data without using the Pre-formatted fixed width text.

Creating a table must begin with the <TABLE> element and end with the </TABLE> element. Within the table you can define <TH> (Table Headers) on either the rows or columns. These headers will be rendered as **bold** text. Also defined are <TR> (Table Rows), which are rows of data in your table. Finally, <TD> (Table Data) are standard table data cells containing your text or images to appear in the table. Table Data Cells must reside in Table Rows. You may also use the <CAPTION> element which gives the table a caption or heading which can be aligned to the top or bottom of the table. When creating Tables, your HTML document will contain something resembling the following HTML code.

```
<BODY>
   <TABLE BORDER=1>
   <CAPTION ALIGN=Top>Sample Table</CAPTION>
   <TR>
      <TH>Table Header 1</TH>
      <TH>Table Header 2</TH>
      <TH>Table Header 3</TH>
   <TR>
      <TD>Table Row 1 Table Data 1</TD>
      <TD>Table Row 1 Table Data 2</TD>
      <TD>Table Row 1 Table Data 3</TD>
   <TR>
      <TD>Table Row 2 Table Data 1</TD>
      <TD>Table Row 2 Table Data 2</TD>
      <TD>Table Row 2 Table Data 3</TD>
   </TABLE>
</BODY>
```

Once completed, you can expand on your table format and contents. The Following tag parameters are expandable by double clicking on the following tags. Parameter dialog boxes are displayed to outlines the following settings.

<TABLE>	Whole table properties. Edit alignment, column sizes and color.
<TH>	Table Column/row headings. Edit ALIGN (Horizontal Alignment) and VALIGN (Vertical Alignment) and color.
<TR>	Table Row. Edit Alignment and color.
<TD>	Table data. Edit Alignment, Cell size and width, color and wrapping.

The following example of the HTML table element is an extension to the above example:

```
<BODY>
   <TABLE ALIGN=center WIDTH="100%" BORDER BGCOLOR=#8000FF>
   <CAPTION ALIGN=TOP><B>Sample Table</B></CAPTION>
   <TR>
      <TH ALIGN=left BGCOLOR=#00FF00>Image</TH>
      <TH ALIGN=left BGCOLOR=#00FF00>Description</TH>
      <TH ALIGN=right BGCOLOR=#00FF00>Price</TH>
   <TR>
      <TD><IMG  SRC="images/modem.gif"></TD>
      <TD>BANKSIA MODEM </TD>
      <TD ALIGN=right VALIGN=middle>$399.00 </TD>
   <TR>
      <TD><IMG  SRC="images/printer.gif"></TD>
      <TD>CANON PRINTER </TD>
      <TD ALIGN=right VALIGN=middle>$500.00 </TD>
   </TABLE>
</BODY>
```

For Further references, see also:

<TABLE> ... </TABLE> Table
<TR ...> ... </TR> Table Row
<TD ...> ... </TD> Table Data
<TH ...> ... </TH> Table Header
<CAPTION> ... </CAPTION> Table Caption
Table Wizard FlexED Table Wizard

Adding Images to your Documents

The <IMG> (Image) element handles all embedded graphical content in HTML documents. The attributes commonly supported by the IMG element have had some recent additions to allow Client side Image Maps, embedded In-line video clips and also embedded In-line VRML worlds.

A note about formats:

Most Browsers support (i.e. Browser can display) progressive JPEG images, GIF, JPG, PNG (portable network graphics) images and also BMP (Bitmap) images giving the author a wide variety of image formats to choose from. Most browsers now also support the GIF89a format, which means that multi-image GIF files can be used to create animation sequences. Users are encourage to seek out the GIF construction Kit for more details and tools for preparation of multi-image GIF files.

To Add Images to your Document:

Adding images in your document is made easy by FlexED. From the main toolbar you can select the insert image button and then select img… from the pull down menu. A dialog box will appear asking you to select the source image, which is transferred to the HTML document as SRC= your image. Select your image, and the rest of the parameters, height, Width etc. will be initiated with the default settings of that image. Select OK and the image will be placed in your HTML document. The following is an example of the HTML code inserted.

```
<IMG SRC="html/images/test.gif" WIDTH=32 HEIGHT=32>
```

Where SRC is the source image and Width and height is the size of your image to be rendered.
NOTE: Please be careful of the relative pathing of your images.

For further reference see also:

<IMG...> In-line images
Creating Clickable Image Maps
Image Map Wizard

Creating Clickable Image Maps

Images can be placed in your document and act as a Hypertext link or a hotspot. In other words you can achieve a link to another source, by clicking on an image within your document when rendered.

First, add your image into your HTML document. If you are unsure on how to do this, read the help on Adding Images to your Documents. Your then will have HTML code similar to below in your document:

```
<IMG SRC=images/aboutus.gif ALT=About Us WIDTH=95 HEIGHT=75>
```

Following this, we will create a link with this image. Within the FlexED editor environment, highlight the entire reference of your image, that is highlight <IMG …… >. Then select the insert Anchor button from the FlexED Main Toolbar. This will recall the HRef Builder Dialog Box. Select your resource or link and then OK to save. This will create HTML code similar to below in your document:

```
<A Href=http://infoflex.com.au/aboutus.htm> <IMG SRC=images/aboutus.gif ALT=About Us
WIDTH=95 HEIGHT=75></A>
```

When this HTML code is rendered, the image when you click on it, will link to and render aboutus.htm. Creating a clickable image like this is similar to creating text links, with the hotspot or Hypertext link that will be rendered, lies between the <A ……> ….. </A>.

Multiple Hotspots may also be created on one image. You must create an image map to achieve this. See the section on Image Map Wizard for more information.

For Further Reference See:

<MAP ...> Client Side Image Maps
<AREA> ... </AREA>
<IMG...> In-line images

Adding Multimedia to an HTML Document

Before adding Multi Media to your HTML document, you must first be aware of how to use hyperlinks in your document pointing to other documents or to Internet services. Creating a hyperlink to an Audio or Video file works exactly the same way.

Example of Use:

```
<BODY>
   <H1>Audio And Video demonstration</H1>
   You can view a video clip of <A HREF="products.mpg">Our Company Products"</A> or
   <A HREF="music.wav"> Listen to Some Music</A>
</BODY>
```

As you can see from above, the Hyperlink points to certain Video and Audio files on your computer. The Motion Picture Experts Group (MPEG) Video clip and Audio clips were merely included as targets of the otherwise ordinary anchor links. Note, that audio and video files, even though they might be short, can be extremely large in size, causing lengthy download times.

Users who will be accessing your Web Pages, must have the appropriate helper applications and hardware installed to deal with the data (Video and audio files). Most Web Browsers only support HTML documents, plain text and only a few kinds of in-line images. Therefore, Helper applications are needed to take the data that Web Browsers can not deal with, to display unsupported images, sound and video files. Once a Helper application is installed, the web Browser must be told to use it. You must use standard file-naming conventions for your multi media files, so your Web server and your users' browser can identify and render them.

Note: The Microsoft Internet explorer now supports the <BGSOUND...> tag for background sound. the following syntax is an example of how to use the <BGSOUND> Tag. This is the only browser at present to support this tag.

```
<BGSOUND SRC="Backgroud.wav" LOOP=1>
```

Fill-in Forms and Scripts

Besides including hyperlinks and images in your HTML documents, collecting user input using fill-in forms is probably the most important feature of HTML. You can get input from users and feed it to computer programs or databases for any purpose.

Simple forms Created Automatically:

Although there are HTML tags for creating fill-in forms, the simplest of forms gets created automatically by a web browser when it encounters a single mark-up tag, <ISINDEX>. It collects user input for a program that processes it.

For Example:

```
<HTML>
<HEAD><TITLE>Address Book</TITLE><ISINDEX></HEAD>
<BODY>
<H1>Address Book</H1>
Type a name in the box above to search the Address Book database.
</BODY>
<HTML>
```

When rendered, the above HTML text will display your text as well as inserting a form that reads This is a searchable index. Enter search keywords: Please note we have not included any reference pointing to a program to search the database.

Advanced HTML Fill-in Forms:

More advanced forms must be declared in the <FORM> mark-up tag. We use the Form tag to announce that an HTML document is also a form. The general syntax is:

```
<FORM ACTION=URL>
<INPUT NAME=label1>
Contents of Form
<INPUT TYPE=submit>
</FORM>
```

There are several ways to collect information from users. These are done by simple fill in the blanks, checkboxes, radio buttons and pull-down or scrollable menus.
The action attribute is required and specifies what to do with the user input data. Within an HTML form, you include markup tags to collect user input. This is done with a tag and two attributes.
The **<INPUT>** tag collects and saves the data for the user so it can be passed off later.
The **NAME** attribute attaches a label to the information
The **TYPE** attribute signifies the type of action to be taken with the data.

The fill in the blank, Checkbox, and Radio buttons require the input, type and name attributes. The following are examples:

```
<INPUT TYPE=CHECKBOX NAME=choices VALUE=ValueToPass> Checkbox Text.
<INPUT TYPE=RADIO NAME=choice VALUE=ValueToPass> Radio Button Text.
<INPUT TYPE=TEXT  SIZE=20 NAME=Aname VALUE=ValueToPass> Type in Text.
<INPUT TYPE=reset>
<INPUT TYPE=submit VALUE=ValueToPass>
```

The **VALUE** attribute used here, is the value that will be passed to other programs or databases if selected (Checked) or inputted. These values get passed to the program that the form is a front end for.
The **SIZE** attribute specifies the length of the input size of the text form input.

Note, the reset type will create a button, that when clicked will return the form to its default settings. The Submit type will create a button, that when clicked will submit all the Values to pass, to their designated programs or databases.

If pull down menus and/or scroll boxes are required, the <**SELECT**> tag and its companion **<OPTION>** must be used. Below is an example of a scroll box and a menu:

```
<FORM ACTION=URL/CGI Script>
Choose from the Scroll box, Your Favorite Colors. Press CTRL-Click for multiple
Colors:
<SELECT NAME=choices MULTIPLE>
<OPTION SELECTED>Blue
<OPTION> Green
<OPTION> Red
<OPTION> Yellow </SELECT>
<P> Choose from the Menu, the Color you least like (One only).
<SELECT NAME=oneOnly>
<OPTION SELECTED>Blue
<OPTION> Green
<OPTION> Red
<OPTION> Yellow </SELECT>
```

You may also collect extended user input by using the **TextArea** tag. This tag provides a free-form, multi-line text box for user input. TextArea requires the NAME attribute and allows for sizing, both in width (COLS) and height (ROWS). For example of use:

```
<TEXTAREA NAME=comments COLS=40 ROWS-8 </TEXTAREA>
```

Using Common Gateway Interface (CGI)

The basic concept of CGI is taking the information entered on the form and passing it to the web server. The CGI script can be written in any language (Unix script, Dos) and is a program that processes the data from the form in some way. The CGI Program may also return the results after processing of the data, in the form of HTML so the user can see or confirm the results of the form data input.

For Further References, see also:

<FORM> ... </FORM>
<ISINDEX...>
<TEXTAREA> ... </TEXTAREA>
<INPUT>
<OPTION>

Announcing Your Web Pages to the World

Once you have created or completed your web Pages, you need people to find and locate your page to view what you are promoting, providing, or whatever your page inhibits. To do this, you need to add your URL to the search engines around the world. Most search engines provide a free listing in their search database, but for real exposure, you need to part with some hefty amounts of money. Some off the search engines that you can submit your details to are Yahoo, Excite, Infoseek, Alta Vista and the list goes on. There are also some useful sites that submit your information from one point to many search engines at once which is saves you time adding your URL to each individual search engine. One such site is located at http://www.submit-it.com. They submit your details to the top 16 or so search engines for free, otherwise they will charge around US $60.00 for submitting your URL to around 300 search engines.

Other forms of announcing your Web Pages, is on the mother of all bulletin boards, UseNet (also called NetNews). If you have no access to UseNet, post a message about your resource to the newsgroup **comp.infosystems.www.announce.** You can also post an announcement of your Web resource to the **Newsgroup comp.internet.net-happenings**.

NOTE: Make sure you use some catchy phrases to attract peoples attention.

FlexED Order Form

INFOFLEX REGISTRATION FORM for FlexED for Single Users ONLY

PROGRAM:	# COPIES	AMOUNT
FlexED HTML Editor (AUS$39.95 or US$32) for Windows 3.1x	____	$_____
FlexED HTML Editor (AUS$39.95 or US$32) for Windows 95	____	$_____
Postage & Handling (AUS$10 or US$10) Include postage and handling if you require FlexED on a Floppy Disk		$_____
TOTAL		$_____

PAYMENT INFORMATION:

CHARGE:

MasterCard [ ] Visa [ ] Bankcard [ ] Cheque/Money Order [ ]

Card Number : _____|_____|_____|_____
Expiry Date : ____/_____
Card Issued To: _____
Signature : _____

MAILING ADDRESS:
Name: _____
Address: _____
City/State/Province: _____
Country/Postal Code: _____
Telephone (Voice/FAX): _____
Internet Email Address: _____

FAX Registration Forms to +61 2 9894-2067

EMAIL sales@infoflex.com.au

MAIL Infoflex Pty Ltd
Unit 27, 15 Terminus Street
Castle Hill, 2154 NSW
AUSTRALIA

NOTE: Price is current for FlexED v2.0 - Prices may increase as FlexED versions are released.

Contacting Infoflex

If you would like to purchase FlexED, upgrade to a newer version of FlexED or simply obtain an evaluation copy of FlexED,

E-Mail: **sales@infoflex.com.au**

Or Fax: + 61 2 **9894 2067**

Or Phone: + 61 2 **9894 1313**

Or Snail Mail: **Infoflex Pty Ltd**
Unit 27, 15 Terminus Street
Castle Hill 2154, NSW
Australia

Our home page can be viewed at

http://www.infoflex.com.au

or

http://nt.infoflex.com.au/flexed/flexed.htm

About FlexED

FlexED is the most powerful tool in the market in designing and creating your Internet home pages. The ease at which FlexED operates, allows even the computer illiterate user to create professional home pages quickly and cost effectively. The FlexED Web Editor designs and creates pages for the World Wide Web (WWW). Web pages created by FlexED produce HyperText Markup Language (HTML) documents, a text-based language. Internet Browsers like Netscape and Mosaic read the HTML files, and display them on the screen. The difference between an HTML document and a word processing document is that the same HTML document can be read on many different computer platforms, for example PC, Macintosh and UNIX whilst the word processing document can not.

FlexED is a stand-alone program. This means that it doesn't need an expensive word processing program to run. It also means that you can deal with your HTML documents directly, instead of having to "pretend" that they're standard word processor documents. While you are creating your HTML documents, you can view the document at hand with FlexEDs viewer. That is, you can view your Web Pages as it would display in any Web Browser while you create your documents. This is the only available HTML editor that supports all versions of HTML and future versions.
FlexED offers many ways to creating HTML documents easier. If you are an experienced HTML author, you can type all the formatting tags directly, or select them from menus and pop-up lists, or even create your own custom tags. Many hours of productive time on your pages are saved. If you are new to HTML, FlexED has screens and wizards to allow you to create and format your documents easily, insert images and tables, and create hypertext links or hot spots (links to other documents) onto your document without having to learn the HTML language. And whilst FlexED performs its wizards, you can watch and learn about HTML at the same time.

HTML tags are color coded so they can be easily distinguished from your own text. You may also click on these tags in your HTML document, and a dialog will be presented outlining your options and parameters without needing to remember these.

FlexED is also limitless in creativity of your HTML documents. Whilst most HTML tools have wizards and the like, you are unable modify or adjust your document to meet your needs. FlexED has combined the use of wizards with a user friendly front end and also the power to modify and add to your HTML document directly.

FlexED Features

Colour Syntax Highlighting

HTML Tags appear in a different colour, HTML Tags are visible at a glance.

Tag Dialog boxes

Double click on HTML Tags to display dialog box to change parameters, you dont have to remember all the parameters available. Editor will not alter parameters it does not recognize, this enables you to use new parameters that are not supported by all Viewers.

Multiple Documents

Open up multiple documents. Work on multiple projects at once.

Built in HTML Viewer

With the built in HTML Viewer you do not need to have a HTML browser to view your HTML pages, built in viewer supports HTML 2.0 and HTML 3.0 forms. Watch your HTML page come alive as you create your document.

Frames Support

New Frames Wizard allows you to create Frames in seconds. View your frames within netscape with a click of mouse button.

Netscape DDE Support

View your documents within Netscape 2.0 +. DDE supports allows you to view your documents without opening a new copy of Netscape.

Wizards

Use the many wizards that are available. Create tables, forms, hyper text links and much more in seconds.

Large file Support

Load HTML documents up to a file size of 60MB.

ImageMap Wizard

Create image map's in minutes. Select ImageMap Wizard, Load your image, draw your hotspots then press OK. Its that simple to create image maps.

The Toolbar

Create new file - Create a blank HTML document template on which to work with.

New Frame HTML - Calls up Frame Wizard allowing the creation if framed HTML documents with multiple views.

Open File - Access previously created HTML document.

Save File - Saves all your hard work to designated location

Insert Unordered List - Use the list editor to create unordered lists. Created list will be inserted into the current active document.

Insert Ordered List - Use the list editor to create ordered lists. Created list will be inserted into the current active document.

Insert Table - Displays pull down menu of the following options

Table
Header
Data
Row
Caption

Wizard

Insert Anchor - Displays <A…> anchor editor which allows the editing of the <A> element.

Insert Image - Displays pulldown menu of the following options

Img
Map
Area

Map Wizard

Break - Inserts forced line break.

Horizontal Line - Inserts horizontal line on page.

Select font - Displays Font Selection dialog Box which allows the selection of font type, size & colour.

Forms - Displays pull down menu of options used to create fill in forms which are areas within the document that allows the reader to type in data

Form
Input
Select
Option
TextArea

Refresh HTML Viewer - renders and displays the active HTML document window.

Refresh Alternative Browser - Passes the contents of the active HTML document window to your selected external browser. The first time this button is activated you will be instructed to locate where your browser is located. You can use the FILE|PROPERTIES option to change the location of your external browser.

Paragraph - Insert <P> tag.

Pre-formatted Paragraph - Insert <PRE> tag.

Centered Text - Insert <CENTER tag.

Heading 1 - Insert <H1> tag.

Heading 2 - Insert <H2> tag.

Select heading 3..6 - display pull down menu of the Heading sizes from 3 .. 6

Bold - Insert <B> tag.

Italic - Insert <I> tag.

Underline - Insert <U> tag.

Strong - Insert <STRONG> tag.

Emphasized - Insert <EM> tag.

Citation - Insert <CITE> tag.

TT - Insert <TT> tag.

Address - Insert <ADDRESS> tag.

MENU BAR

FlexED menu items defined are:

File	**New**	Create a blank HTML document template.
	New Frame	Calls up Frame Wizard
	Open	Access previously created HTML document.
	Close	Close HTML Document
	Close All	Close All HTML documents
	Save	Save HTML Document
	Save As	Save HTML Document As another filename
	Save All	Save All HTML Documents
	Print	Print HTML Document
	Print Setup	Windows Printer Setup
	Edit HTML Toolbar	Edit Custom Toolbar
	Properties	FlexED Setup
	Refresh Int Browser	Refresh Internal Browser
	Refresh Ext Browser	Refresh External Browser
	Exit	Leave FlexED

History Of last four open HTML documents.

Edit	**Undo**	Undo the most recent change.
	Redo	Redo the most recent undo.
	Cut	Cut highlighted text from the document to the clipboard
	Copy	Copy highlighted text from the document to the clipboard
	Paste	Paste contents of clipboard to the HTML document.
	Delete	Delete highlighted section.
	Find	Search for a given phrase.
	Find Next	Find Next Phrase
	Replace	Search and replace found phrase with new phrase.
	Select All	Highlight the contents of the active HTML document.

Text	**Auto Indent**	Auto Indent Tags
	Word Wrap	Turn Word Wrap on and off
	Syntax Highlight	Turn Syntax Highlighting on and off
	Strip HTML Tags	Delete HTML tags from selected text
	Edit Image Map	Edit Image Map properties
	Sync Browser	Synchronize Browser with editor
	Change Edit Font	Change the Editors font
	Lowercase	Change selected text to lowercase
	Uppercase	Change selected text to uppercase

Browser

	Open	Open Browser
	Print	Print Rendered document
	Show Images	Display rendered images on and off
	Edit Document	Edit Rendered document
	Back	Go Back
	Forward	Go Forward

View	**Main Toolbar**	Display Main Toolbar on and off
	Format Toolbar	Display Format Toolbar on and off
	ColdFusion toolbar	Display Cold Fusion Toolbar on and off
	HTML Toolbar	Display HTML Toolbar on and off

	Status Bar	Display Status bar on and off
Window		
	Tile Vertical	Tile visible document windows vertically
	Tile Horizontal	Tile visible document windows horizontally
	Cascade	Cascade visible document windows
	Arrange Icons	Arrange minimized document windows
	Split vertical	Split all documents vertically
	Split Horizontal	Split all documents horizontally
Help	**Contents**	Help Contents reference
	How To Use Help	How to use the FlexED Help
	About	About Infoflex

<A...> ... </A>

An Anchor element is a marked text that is the start and/or destination of a hypertext link. Anchor elements are defined by the <A> element. The <A> element accepts several attributes, but either the NAME or HREF attribute is required.

Attributes of the <A> element :

HREF If the HREF attribute is present, the text between the opening and closing anchor elements becomes hypertext. If this hypertext is selected by readers, they are moved to another document, or to a different location in the current document, whose network address is defined by the value of the HREF attribute.

Example of use:

```
<A HREF="http://www.microsoft.com">Microsoft</A>
```

In this example, selecting "Microsoft" takes the reader to a document located at http://www.Microsoft.com. The format of the network address is specified in the URI specification for print readers.

With the HREF attribute, the form HREF="#identifier" can refer to another anchor in the same document.

Example of use:

```
The <A HREF="document.html#glossary">glossary</A>
```

In this example, selecting "glossary" takes the reader to another anchor (i.e. <A NAME="glossary">Glossary</A>) in the same document (document.html). The NAME attribute is described below. If the anchor is in another document, the HREF attribute may be relative to the document's address or the specified base address.

Several other forms of the HREF attribute are permitted by browsers. They are as follows :

<A HREF="http://..."> Makes a link to another document located on a World Wide Web server.
<A HREF="ftp://..."> Makes a link to an ftp site.
<A HREF="gopher://...> Makes a link to a gopher server.
<A HREF="mailto:..."> Activating such a link would bring up your Browser mailing dialogue box (providing it has mailing capabilities) allowing the user to send mail messages to the author of the document, or whoevers address follows the mailto attribute.
<A HREF="news:..."> Makes a link to a newsgroup. Care should be taken in using such links because the author can not know what newsgroups are carried by the local news server of the user.
<A HREF="newsrc:..."> Makes a link to a specific newsrc file.
<A HREF="nntp://..."> Can be used to specify a different news server to that which the user may normally use.
<A HREF="telnet://..."> Activating such a link would initiate a Telnet session (using an external application) to the machine specified after the Telnet:// label.
<A HREF="wais://..."> Makes a link that connects to a specified WAIS index server.

If present, the **NAME** attribute allows the anchor to be the target of a link. The value of the NAME attribute

is an identifier for the anchor. Identifiers are arbitrary strings but must be unique within the HTML document.

Example of use:

```
<A NAME=coffee>Coffee</A> is an example of...
An example of this is <A HREF=#coffee>coffee</A>.
```

Another document can then make a reference explicitly to this anchor by putting the identifier after the address, separated by a has sign :

```
<A NAME=drinks.html#coffee>
```

The **Title** attribute is informational only. If present, the Title attribute should provide the title of the document whose address is given by the `HREF` attribute.
The Title attribute is useful for at least two reasons. The HTML user agent may display the title of the document prior to retrieving it, for example, as a margin note or on a small box while the mouse is over the anchor, or while the document is being loaded. Another reason is that documents that are not marked up text, such as graphics, plain text and Gopher menus, do not have titles. The `TITLE` attribute can be used to provide a title to such documents. When using the `TITLE` attribute, the title should be valid and unique for the destination document.

The **`REL`** attribute gives the relationship(s) described by the hypertext link from the anchor to the target. The value is a comma-separated list of relationship values. Values and their semantics will be registered by the HTML registration authority. The default relationship if none other is given is void. The `REL` attribute is only used when the `HREF` attribute is present.

The **`REV`** attribute is the same as the `REL` attribute, but the semantics of the link type are in the reverse direction. A link from A to B with `REL="X"` expresses the same relationship as a link from B to A with `REV="X"`. An anchor may have both `REL` and `REV` attributes.

If present, the **`URN`** attribute specifies a uniform resource name (URN) for a target document. The format of URNs is under discussion (1994) by various working groups of the Internet Engineering Task Force.

The **`METHODS`** attributes of anchors and links provide information about the functions that the user may perform on an object. These are more accurately given by the HTTP protocol when it is used, but it may, for similar reasons as for the `TITLE` attribute, be useful to include the information in advance in the link. For example, the HTML user agent may chose a different rendering as a function of the methods allowed; for example, something that is searchable may get a different icon.
The value of the `METHODS` attribute is a comma separated list of HTTP methods supported by the object for public use.

Browser windows can now have names associated with them. Links in any window can refer to another window by name. When you click on the link, the document you asked for will appear in that named window. If the window is not already open, Netscape will open and name a new window for you.

The syntax for the targeted windows is:

```
<A HREF="url.html" TARGET="window_name">Click here and open a New Window</A>
```

<ADDRESS> ... </ADDRESS>

The Address element specifies such information as address, signature and authorship, often at the top or bottom of a document.

Typically, an Address is rendered in an italic typeface and may be indented. The Address element implies a paragraph break before and after.

Example of use:

```
<ADDRESS>
FlexED Headquarters <BR>
T. Tontu <BR>
15 Terminus Street, Castle Hill, 2154 NSW <BR>
Tel +61 (2) 9894 1313
</ADDRESS>
```

<APPLET...> ... </APPLET>

Below is a more complex example of an APPLET element:

```
<APPLET CODEBASE="http://infoflex/java/applets/JavaText" CODE="JavaText.class" width=300 height=80
  align=center >
<PARAM NAME="text" VALUE="This is the Applet Viewer.">
<BLOCKQUOTE>
<HR>
If you were using a Java(tm)-enabled browser, you would see dancing text instead of this paragraph.
<HR>
</BLOCKQUOTE>
</APPLET>
```

This tells the viewer or browser to load the applet whose compiled code is at the URL http://infoflex/java/applets/Javatext/javaText.class, to set the initial size of the applet to 300x80 pixels, and to align the applet in the centre of the line. The viewer/browser must also set the applet's "text" attribute (which customizes the text this applet displays) to be "This is the Applet Viewer." If the page is viewed by a browser that can't execute applets written in the Java Programming Language, then the browser will ignore the APPLET and PARAM elements, displaying the HTML between the <BLOCKQUOTE> and </BLOCKQUOTE> elements.

Java(tm)-enabled browsers *ignore* that HTML.

Here's the complete syntax for the APPLET element:

```
<APPLET
  [CODEBASE = codebaseURL]
  CODE = appletFile
  [ALT = alternateText]
  [NAME = appletInstanceName]
  WIDTH = pixels HEIGHT = pixels
  [ALIGN = alignment]
  [VSPACE = pixels] [HSPACE = pixels]
>
  [<PARAM NAME = appletAttribute1 VALUE = value>]
  [<PARAM NAME = appletAttribute2 VALUE = value>]
  ...
  [alternateHTML]
</APPLET>
```

CODEBASE = *codebaseURL* This optional attribute specifies the base URL of the applet -- the directory that contains the applet's code. If this attribute is not specified, then the document's URL is used.

CODE = *appletFile* This required attribute gives the name of the file that contains the applet's compiled Applet subclass. This file is relative to the base URL of the applet. It cannot be absolute.

ALT = *alternateText* This optional attribute specifies any text that should be displayed if the browser understands the APPLET element but can't run applets written in the Java(tm) Programming Language.

NAME = *appletInstanceName* This optional attribute specifies a name for the applet instance, which makes it possible for applets on the same page to find (and communicate with) each other.

WIDTH = *pixels* **HEIGHT** = *pixels* These required attributes give the initial width and height (in pixels) of the applet display area, not counting any windows or dialogs that the applet brings up.

ALIGN = *alignment* This required attribute specifies the alignment of the applet. The possible values of this attribute are the same as those for the `IMG` element: left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom.

VSPACE = *pixels* **HSPACE** = *pixels* These option attributes specify the number of pixels above and below the applet (`VSPACE`) and on each side of the applet (`HSPACE`). They're treated the same way as the IMG element's `VSPACE` and `HSPACE` attributes.

<**PARAM NAME** = *appletAttribute1* **VALUE** = *value*> This element is the only way to specify an applet-specific attribute. Applets access their attributes with the getParameter() method.

<AREA> ... </AREA>

The AREA element specifies a single area of an image which, if selected, will link to the Hypertext Link identified by HREF. If multiple AREA elements in the same MAPhtml_map define overlapping areas, the first encountered takes precedence

<B> ... </B>

The Bold Element specifies that your text should be rendered in Boldface.

```
This is an <B>Example of Boldface</B> in your document.
```

Would be rendered as:

This is an **Example of Boldface** in your document.

<BASE...>

The Base element allows the URL of the document itself to be recorded in situations in which the document may be read out of context. URLs within the document may be in a ˝partial form relative to this base address. The <BASE> Element should appear within the bounds of the <HEAD> element.

Where the base address is not specified, the HTML user agent uses the URL it used to access the document to resolve any relative URLs.

The Base element has one attribute, **HREF**, which identifies the URL.

For Example:

```
<BASE HREF="http://www.myhost.com/">
```

The Netscape Navigator (from version 2.0) adds one other attribute to the BASE element. With the introduction of targeted windows, Netscape have added the **TARGET** attribute to the BASE element.

This allows you to pick a default named target window for every link in a document that does not have an explicit TARGET attribute. It's format is:

```
<BASE TARGET="default_target">
```

NOTE : The use of the TARGET attribute is **Netscape** specific.

<BASEFONT ...>

This changes the size of the BASEFONT, such that all relative Font Size changes are based on. It defaults to 3, and has a valid range of 1-7.

```
<BASEFONT SIZE=5>
```

FACE This attribute allows changing of the face of the HTML document BASEFONT, exactly as it works for <FONT Face= …>

COLOR This allows the BASEFONT colour for the HTML document to be set. Colours can either be set by using one of the reserved colour names, or as a hex rrggbb triplet value.

<BGSOUND...>

The BGSOUND element will cause an audio file to be presented as background to the document. (MS Explorer Only)

The Following is an example:

```
<BGSOUND SRC=Audio.wav LOOP=2>
```

SRC is the source of the Audio file.

LOOP is the amount of times to play this audio file. For example, if your Audio file lasts for 2 Minutes and LOOP=5, your Audio file will in fact play over 5 times, giving 10 minutes of Audio.

<BIG> ... </BIG>

The <BIG> element specifies that the enclosed text should be displayed, if practical, using a big font (compared with the current font). This is an HTML 3.0 element and may not be widely supported.

```
This is an example of <BIG>Big Text</BIG> in your document.
```

Would be rendered as:

This is an example of Big Text in your document.

<BLINK>

Surrounding any text with this element will cause the selected text to *blink* on the viewing page. This can serve to add extra emphasis to selected text.

```
This an <BLINK>Example of Blinking Text</BLINK>
```

The Example of Blinking Text would be rendered as Blinking on your HTML Document.
NOTE : The `<BLINK> ... </BLINK>` element is currently only supported by the Netscape Navigator.

<BLOCKQUOTE> ...</BLOCKQUOTE>

The BLOCKQUOTE element defines a separated multi-line set of text to be rendered as quoted text.

<BODY> ... </BODY>

The body of a HTML document contains all the text and images that make up the page, together with all the HTML elements that provide the control and formatting of the page. The format is:

```
<BODY>
all document entities included here
</BODY>
```

The <BODY> and </BODY> elements do not directly affect the look of the document when rendered, although they are ***required*** in order for the document to conform to the specification standard.

First to be implemented by Netscape, the ability to specify background images and colours for HTML documents has recently been implemented by many other browsers. It should be noted that the following elements are not supported by every HTML user agent available.

There now is a **BACKGROUND** attribute available to the BODY element. The purpose of this attribute is to specify a URL pointing to an image that is to be used as a background for the document. In many recent browsers, this background image is used to tile the full background of the document viewing area. For Example:

```
<BODY BACKGROUND="filename.gif">
Rest of Document here
</BODY>
```

would cause whatever text, images, etc. appeared in that document to be placed on a background consisting of the (filename.gif) graphics file being tiled to cover the viewing area, much like bitmaps are used for Windows wallpaper.

The **BGCOLOR** attribute changes the colour of the background without having to specify a separate image that requires another network access to load. An example of the format is:

```
<BODY BGCOLOR="#xxxxx">
Rest of Document here
</BODY>
```

Where "#xxxxx" is a hexadecimal red-green-blue triplet used to specify the background colour.

Clearly, once the background colours or patterns have been changed, it will be necessary to also be able to control the foreground to establish the proper contrasts.

The **TEXT** attribute is used to control the colour of all the normal text in the document. This basically consists of all text that is not specially coloured to indicate a link. The format of TEXT is the same as that of BGCOLOR.

```
<BODY TEXT="#xxxxx">
Rest of Document here
</BODY>
```

LINK, **VLINK**, and **ALINK** attributes let you control the colouring of HTML link text. VLINK stands for visited link, and ALINK stands for active link. The default colouring of these is: LINK=blue, VLINK=purple, and ALINK=red. Again, the format for these attributes is the same as that for BGCOLOR and TEXT.

```
<BODY LINK="#rrggbb" VLINK="#rrggbb" ALINK="#rrggbb">
rest of Document here
```

```
</BODY>
```

Colouring Considerations.

Since these colour controls are all attributes of the `BODY` element, they can only be set once for the entire document. Document colour cannot be changed partially through a document.

Setting a background image requires the fetching of an image file from a second HTTP connection, it may slow down the perceived speed of document loading. Needless to say, background images should be kept small.

If the Auto Load Images option is turned off, background images will not be loaded. If the background image is not loaded for any reason, and a `BGCOLOR` was not also specified, then any of the foreground controlling attributes (`LINK`, `VLINK`, and `ALINK`) will be ignored. The idea behind this is that if the requested background image is unavailable, or not loaded, setting requested text colours on top of the default grey background may make the document unreadable.

<BR ...>

The Line Break element specifies that a new line must be started at the given point. A new line indents the same as that of line-wrapped text.

Example of use:

```
<BODY>
       Create a new line<BR>Create a new line<BR>Create a new line
</BODY>
```

Would be rendered as:

Create a new line
Create a new line
Create a new line

<CAPTION> ... </CAPTION>

This represents the caption for a table. `<CAPTION>` elements should appear inside the <TABLE> but not inside table rows or cells. The caption accepts an alignment attribute that defaults to `ALIGN=top` but can be explicitly set to `ALIGN=bottom`. Like table cells, any document body HTML can appear in a caption. Captions are always horizontally centred with respect to the table, and the may have their lines broken to fit within the width of the table.

The `<CAPTION>` element can accept the following attributes.

`ALIGN` The `ALIGN` attribute controls whether the caption appears above or below the table, and can have the values **right, left or center**, defaulting to center. This forces the caption text to be aligned flush-left, flush-right, or centred within the boundaries of the table.

`VALIGN` It specifies whether the caption text should be displayed at the **top** or **bottom** of the table or horizontal alignment.

<CENTER> ... </CENTER>

All lines of text between the begin and end of the <CENTER> element are centred between the current left and right margins. A new element has been introduced rather than using the proposed <P Align=Center> because using <P ALIGN=CENTER> breaks many existing browsers when the <P> element is used as a container. The <P ALIGN=CENTER> element is also less general and does not support all cases where centering may be desired.

```
<CENTER>All this text would be centred in the page</CENTER>
```

NOTE : Most browsers will internally work-round this element to produce the desired format, but it is an element introduced by Netscape authors.

<CITE> ... </CITE>

The Citation element specifies a citation; typically rendered as italics.

```
This is an <CITE>Example of a Citation</CITE> in your document.
```

Would be rendered as:

This is an *Example of a Citation* in your document

<CODE> ... </CODE>

The Code element indicates an example of code, typically rendered as mono-spaced text. Do not confuse with the <PRE> tag element.

For Example:

```
This sentence contains an <CODE>example of code text</CODE>.
```

Would Be Rendered as:

This sentence contains an `example of code text`

<DD> ... </DD>

Description

The DD element identifies the separated multi-line definition item in a DL definition list. In a DL list a DD should always be preceded by at least one DT element.

<DIR> ... </DIR>

A Directory List element is used to present a list of items containing up to 20 characters each. Items in a directory list may be arranged in columns, typically 24 characters wide. If the HTML user agent can optimise the column width as function of the widths of individual elements, so much the better.

A directory list must begin with the `<DIR>` element which is immediately followed by a <LI> (list item) element:

For Example:

```
<DIR>
<LI>A-H<LI>I-M
<LI>M-R<LI>S-Z
</DIR>
```

Would Be Rendered As:

- A-H
- I-M
- M-R
- S-Z

<DL> ... </DL>

A definition list is a list of terms and corresponding definitions. Definition lists are typically formatted with the term flush-left and the definition, formatted paragraph style, indented after the term.

Example of use:

```
<DL>
<DT>Term<DD>This is the definition of the first term.
<DT>Term<DD>This is the definition of the second term.
</DL>
```

Would be Rendered as:

Term
    This is the definition of the first term.
Term
    This is the definition of the second term.

If the <DT> term does not fit in the <DT> column (one third of the display area), it may be extended across the page with the <DD> section moved to the next line, or it may be wrapped onto successive lines of the left hand column.

Single occurrences of a <DT> element without a subsequent <DD> element are allowed, and have the same significance as if the <DD> element had been present with no text.

The opening list element must be <DL> and must be immediately followed by the first term (<DT>).

The definition list type can take the **COMPACT** attribute, which suggests that a compact rendering be used, because the list items are small and/or the entire list is large.

Unless you provide the COMPACT attribute, the HTML user agent may leave white space between successive <DT>, <DD> pairs. The COMPACT attribute may also reduce the width of the left-hand (<DT>) column.

If using the COMPACT attribute, the opening list element must be <DL COMPACT>, which must be immediately followed by the first <DT> element:

Example of use:

```
<DL COMPACT>
<DT>Term<DD>This is the first definition in compact format.
<DT>Term<DD>This is the second definition in compact format.
</DL>
```

Would be rendered as:

Term This is the first definition in compact format.
Term This is the second definition in compact format.

<DT>

Description

The DT element identifies the separated term item in a DL definition list. Multiple DT elements may exist prior to a single DD element. In a DL list a DD should always be preceded by at least one DT element.

<EM> ... </EM>

The Emphasis element indicates typographic emphasis, typically rendered as italics.

```
This is an <EM>Example of Emphasis Text</EM> in your document.
```

Would be rendered as:

This is an *Example of Emphasis Text* in your document

<EMBED ...>

The EMBED element defines a container that allows the insertion of arbitrary objects directly into an HTML page. Embedded objects are supported by application-specific plug-ins. EMBED is defined to allow arbitrary attributes.(Netscape 2.0)

Supported Attributes are ***SRC, HEIGHT*** and ***WIDTH.***

<FONT ...>

The **SIZE** attribute is `<FONT SIZE=value>`. Valid values range from 1-7. The default FONT size is 3. The value given to size can optionally have a '+' or '-' character in front of it to specify that it is relative to the document baseFONT. The default baseFONT is 3, and can be changed with the <BASEFONT …> element.

Example:

`<FONT SIZE=4>` changes the font size to 4 `</FONT>`
`<FONT SIZE=+2>` changes the font size to `<BASEFONT SIZE ...>` + 2 `</FONT>`

The **COLOR** attribute is `<FONT COLOR=#xxxxx>` which sets the colour with which text will appear in on the screen. Above #xxxxx is a hexadecimal colour denoting a RGB colour value.

Example:

```
<FONT COLOR=#ff0000>This is an Example of text in Red.</FONT>
```

The **FACE** attribute is `<FONT FACE=name [,name] [,name]>` which sets the typeface that will be used to display the text on the screen. The type face displayed must already be installed on the users computer. Substitute type faces can be specified in case the chosen type face is not installed on the customers computer. If no match is found, the text will be displayed in the default type.

Example:

```
<FONT FACE="Arial","Lucida Sans"> This is an example of text display in Arial,
  Lucida Sans, or Times Roman, depending on which fonts you have installed on your
  system.</FONT>
```

These three attributes may be used in conjunction with each other, for example,

```
<FONT FACE=Arial COLOR=#ff0000 SIZE=4>Displays this text as size 4, color red in
Arial</FONT>
```

<FORM> ... </FORM>

The Form element is used to delimit a data input form. There can be several forms in a single document, but the Form element can't be nested.

The **ACTION** attribute is a URL specifying the location to which the contents of the form is submitted to elicit a response. If the ACTION attribute is missing, the URL of the document itself is assumed. The way data is submitted varies with the access protocol of the URL, and with the values of the **METHOD** and **ENCTYPE** attributes.

In general:

- the METHOD attribute selects variations in the protocol.
- the ENCTYPE attribute specifies the format of the submitted data in case the protocol does not impose a format itself.

The Level 2 specification defines and requires support for the HTTP access protocol only.

When the ACTION attribute is set to an HTTP URL, the METHOD attribute must be set to an HTTP method as defined by the HTTP method specification in the IETF draft HTTP standard. The default METHOD is GET, although for many applications, the POST method may be preferred. With the post method, the ENCTYPE attribute is a MIME type specifying the format of the posted data; by default, is application/x-www-form-url encoded.

Under any protocol, the submitted contents of the form logically consist of name/value pairs. The names are usually equal to the NAME attributes of the various interactive elements in the form.

NOTE: The names are not guaranteed to be unique keys, nor are the names of form elements required to be distinct. The values encode the user's input to the corresponding interactive elements. Elements capable of displaying a textual or numerical value will return a name/value pair even when they receive no explicit user input.

<FRAME ...>

The FRAME element defines a single frame in a frameset. The SRC attribute value is the URL of the document to be displayed in this frame. A FRAME element without a SRC is displayed as blank space. The NAME element assigns a name to the frame to be used as a target of hyperlinks. (See the A element) The SCROLLING attribute is used to define whether the frame should have a scrollbar, and defaults to the value "auto". Presence of the NORESIZE attribute prevents the frame from being resized by the user. (Netscape 2.0)

SRC="*url***"** The SRC attribute takes as its value the URL of the document to be displayed in this particular frame. FRAMEs without SRC attributes are displayed as a blank space the size the frame would have been.

NAME="*window_name***"** The NAME attribute is used to assign a name to a frame so it can be targeted by links in other documents (These are usually from other frames in the same document.) The NAME attribute is optional; by default all windows are unnamed.

Names must begin with an alphanumeric character. However, several reserved names have been defined, which start with an underscore.

These are currently:

_blank	Always load this link into a new, unnamed window.
_self	Always load this link over yourself.
_parent	Always load this link over your parent. (becomes self if you have no parent).
_top	Always load this link at the top level. (becomes self if you are at the top).

NOTE : Although these are reserved names for the NAME attribute of the <FRAME> element, they should only be referred to using an Anchor target That is, used to target specific windows, allowing smoother transition between framed documents and between framed and *normal* documents. While the **Internet Explorer** supports the naming of frames for document navigation and hyperlinking, it doesn't support the use of the _blank reserved name for opening a document in a new browser window. Also, unlike **Netscape**, the Internet Explorer will not open a new window for a link whose TARGET attribute has not been set by a NAME attribute.

MARGINWIDTH="*value***"** The MARGINWIDTH attribute is used when the document author wants some control of the margins for this frame. If specified, the value for MARGINWIDTH is in pixels. Margins can not be less than one-so that frame objects will not touch frame edges-and can not be specified so that there is no space for the document contents. The MARGINWIDTH attribute is optional; by default, all frames default to letting the browser decide on an appropriate margin width.

MARGINHEIGHT="*value***"** The MARGINHEIGHT attribute is just like MARGINWIDTH above, except it controls the upper and lower margins instead of the left and right margins.

SCROLLING="*yes***|***no***|***auto***"** The SCROLLING attribute is used to describe if the frame should have a scrollbar or not. Yes results in scrollbars always being visible on that frame. No results in scrollbars never being visible. Auto instructs the browser to decide whether scrollbars are needed, and place them where necessary. The SCROLLING attribute is optional; the default value is auto.

NORESIZE The NORESIZE attribute has no value. It is a flag that indicates that the frame is not resizable by the user. Users typically resize frames by dragging a frame edge to a new position. Note that if any frame adjacent to an edge is not resizable, that entire edge will be restricted from moving. This will effect the resizability of other frames. The NORESIZE attribute is optional; by default all frames are resizable.

<FRAMESET ...>

This is the main container for a Frame. It has 2 attributes ROWS and COLS. A frame document has no BODY, and no tags that would normally be placed in the BODY can appear before the FRAMESET tag, or the FRAMESET will be ignored. The FRAMESET tag has a matching end tag, and within the FRAMESET you can only have other nested FRAMESET tags, FRAME tags, or the NOFRAMES tag.

ROWS="row_height_value_list" The ROWS attribute takes as its value a comma separated list of values. These values can be absolute pixel values, percentage values between 1 and 100, or relative scaling values. The number of rows is implicit in the number of elements in the list. Since the total height of all the rows must equal the height of the window, row heights might be normalised to achieve this. A missing ROWS attribute is interpreted as a single row arbitrarily sized to fit.

Syntax of value list.

Value A simple numeric value is assumed to be a fixed size in pixels. This is the most dangerous type of value to use since the size of the viewer's window can and does vary substantially. If fixed pixel values are used, it will almost certainly be necessary to mix them with one or more of the relative size values described below. Otherwise the client engine will likely override your specified pixel value to ensure that the total proportions of the frame are 100% of the width and height of the user's window.

Value% This is a simple percentage value between 1 and 100. If the total is greater than 100 all percentages are scaled down. If the total is less than 100, and relative-sized frames exist, extra space will be given to them. If there are no relative-sized frames, all percentages will be scaled up to match a total of 100%.

Value* The value on this field is optional. A single '*' character is a "relative-sized" frame and is interpreted as a request to give the frame all remaining space. If there exist multiple relative-sized frames, the remaining space is divided evenly among them. If there is a value in front of the '*', that frame gets that much more relative space. "2*,*" would give 2/3 of the space to the first frame, and 1/3 to the second.

Example for 3 rows, the first and the last being smaller than the centre row:

```
<FRAMESET ROWS="20%,60%,20%">
```

Example for 3 rows, the first and the last being fixed height, with the remaining space assigned to the middle row:

```
<FRAMESET ROWS="100,*,100">
```

COLS="column_width_list" The COLS attribute takes as its value a comma separated list of values that is of the exact same syntax as the list described above for the ROWS attribute.

The FRAMESET tag can be nested inside other FRAMESET tags. In this case the complete subframe is placed in the space that would be used for the corresponding frame if this had been a FRAME tag instead of a nested FRAMESET.

<H1> ... </H1>

HTML defines six levels of heading. A Heading element implies all the font changes, paragraph breaks before and after, and white space necessary to render the heading.

The highest level of headings is `<H1>`, followed by `<H2>` ... `<H6>`.

Example of use:

```
<H1>This is a heading</H1>
Here is some text
<H2>Second level heading</H2>
Here is some more text.
```

The rendering of headings is determined by the HTML user agent, but typical renderings are:

`<H1> ... </H1>`
Bold, very-large font, centred. One or two blank lines above and below.

`<H2> ... </H2>`
Bold, large font, flush-left. One or two blank lines above and below.

`<H3> ... </H3>`
Italic, large font, slightly indented from the left margin. One or two blank lines above and below.

`<H4> ... </H4>`
Bold, normal font, indented more than H3. One blank line above and below.

`<H5> ... </H5>`
Italic, normal font, indented as H4. One blank line above.

`<H6> ... </H6>`
Bold, indented same as normal text, more than H5. One blank line above.

Although heading levels can be skipped (for example, from H1 to H3), this practice is discouraged as skipping heading levels may produce unpredictable results when generating other representations from HTML.

`ALIGN``=left|center|right` attributes have been added to the `<H1>` through to `<H6>` elements.

For Example:

```
<H1 ALIGN=center>Hello, this is a heading</H1>
```

would align a heading of style 1 in the centre of the page.

<HEAD> ... </HEAD>

The head of an HTML document is an unordered collection of information about the document. It requires the Title element between <HEAD> and </HEAD> elements thus :

```
<HEAD>
<TITLE> My New HTML Document </TITLE>
</HEAD>
```

The <HEAD> and </HEAD> elements do not directly affect the look of the document when rendered. The following elements are related to the head element. While not directly affecting the look of the document when rendered, they do provide (if used) important information to the HTML user agent.

<BASE>	Allows base address of HTML document to be specified
<ISINDEX>	Allows keyword searching of the document
<LINK>	Indicate relationships between documents
<TITLE>	Specifies the title of the document
<META>	Specifies document information useable by server/clients.

NOTE : The Title element is the only element described here that is required as part of the Head of a HTML document.

<HR>

A Horizontal Rule element is a divider between sections of text such as a full width horizontal rule or equivalent graphic.

Example of use:

```
<BODY>
   <H1> Heading Of document </H1>
   <HR>
   Main body of text
   <HR>
</BODY>
```

The <HR> element specifies that a horizontal rule of some sort (The default being a shaded engraved line) be drawn across the page. To this element recent browsers have added support for 4 new attributes which allow the document author to describe how the horizontal rule should look

SIZE=*number* The SIZE attributes lets the author give an indication of how thick they wish the horizontal rule to be.

WIDTH=*number|percent* The default horizontal rule is always as wide as the page. With the WIDTH attribute, the author can specify an exact width in pixels, or a relative width measured in percent of document width.

ALIGN=*left|right|center* Now that horizontal rules do not have to be the width of the page it is necessary to allow the author to specify whether they should be pushed up against the left margin, the right margin, or centred in the page.

NOSHADE Finally, for those times when a solid bar is required, the NOSHADE attribute lets the author specify that the horizontal rule should not be shaded at all.

<HTML> ... </HTML>

This element identifies the document as containing HTML elements. That is, the document should be constructed thus :

```
<HTML>
Here is all the rest of the document, including any elements.
</HTML>
```

The HTML element is not visible upon HTML user agent rendering and can contain only the <HEAD> and <BODY> elements.

<I> ... </I>

Element specifies that your text should be rendered in Italic font.

```
This is an <I>Example of Italics</I> in your document.
```

Would be rendered as:

This is an *Example of Italics* in your document.

<IMG...>

The Image element is used to incorporate in-line graphics (typically icons or small graphics) into an HTML document. This element cannot be used for embedding other HTML text.

HTML user agents that cannot render in-line images ignore the Image element unless it contains the ALT attribute. Note that some HTML user agents can render linked graphics but not in-line graphics. If a graphic is essential, you may want to create a link to it rather than to put it in-line. If the graphic is not essential, then the Image element is appropriate.

The Image element, which is empty (no closing element), has these attributes:

The **ALIGN** attribute accepts the values `TOP` or `MIDDLE` or `BOTTOM`, which specifies if the following line of text is aligned with the top, middle, or bottom of the graphic.

ALT Optional text as an alternative to the graphic for rendering in non-graphical environments. Alternate text should be provided whenever the graphic is not rendered. Alternate text is mandatory for Level 0 documents. Example of use:

For Example:

```
<IMG SRC="flexed.gif" ALT="Flexed:"> FlexED HTML Editor
```

The **ISMAP** (is map) attribute identifies an image as an image map. Image maps are graphics in which certain regions are mapped to URLs. By clicking on different regions, different resources can be accessed from the same graphic. Example of use:

For Example:

```
<A HREF="http://infoflex/htdocs/images/flexed">
<IMG SRC="flexed.gif" ISMAP></A>
```

NOTE : To be able to employ image maps in HTML documents, the HTTP server which will be controlling document access must have the correct cgi-bin software installed to control image map behavior. i.e. the document must have access to an image map handling script which is pointed to your .map file defining the graphics hot-spots

The value of the **SRC** attribute is the URL of the document to be embedded; only images can be embedded, not HTML text. Its syntax is the same as that of the `HREF` attribute of the <A> element. `SRC` is mandatory. Image elements are allowed within anchors.

For Example:

```
<IMG SRC ="flexed.gif">FlexED HTML Editor
```

Extensions to The IMG Element:

```
<IMG ALIGN=left|right|top|texttop|middleabsmiddle|baseline|bottom|absbottom>
```

The additions to the `ALIGN` options needs a lot of explanation. First, the values "left" and "right". Images with those alignments are an entirely new *floating* image type.
`ALIGN=left` image will float the image down and over to the left margin (into the next available space there), and subsequent text will wrap around the right hand side of that image.
`ALIGN=right` will align the image aligns with the right margin, and the text wraps around the left.
`ALIGN=top` aligns itself with the top of the tallest item in the line.

ALIGN=texttop aligns itself with the top of the tallest text in the line (this is usually but not always the same as ALIGN=top).
ALIGN=middle aligns the baseline of the current line with the middle of the image.
ALIGN=absmiddle aligns the middle of the current line with the middle of the image.
ALIGN=baseline aligns the bottom of the image with the baseline of the current line.
ALIGN=bottom aligns the bottom of the image with the baseline of the current line.
ALIGN=absbottom aligns the bottom of the image with the bottom of the current line.

```
<IMG WIDTH=value HEIGHT=value>
```

The WIDTH and HEIGHT attributes were added to <IMG ...> mainly to speed up display of the document. If the author specifies these, the viewer of their document will not have to wait for the image to be loaded over the network and its size calculated. **Netscape** is the only browser that will scale the whole image if either the WIDTH or HEIGHT attributes are specified, maintaining the aspect ratio. If both are specified then the image is specified accordingly.

```
<IMG BORDER=value>
```

This lets the document author control the thickness of the border around an image displayed.
Warning: setting BORDER=0 on images that are also part of anchors may confuse your users as they are used to a coloured border indicating an image is an anchor.

```
<IMG VSPACE=value HSPACE=value>
```

For the *floating* images it is likely that the author does not want them pressing up against the text wrapped around the image. VSPACE controls the vertical space above and below the image, while HSPACE controls the horizontal space to the left and right of the image.

Using the **LOWSRC** attribute, it is possible to use two images in the same space.

For Example:

```
<IMG SRC="highres.gif" LOWSRC="lowres.jpg">
```

Browsers that do not recognise the LOWSRC attribute cleanly ignore it and simply load the image called "highres.gif".

The Netscape Navigator, on the other hand, will load the image called "lowres.jpg" on its first layout pass through the document. Then, when the document and all of its images are fully loaded, the Netscape Navigator will do a second pass through and load the image called "highres.gif" in place. This means that you can have a very low-resolution version of an image loaded initially; if the user stays on the page after the initial layout phase, a higher-resolution (and presumably bigger) version of the same image can "fade in" and replace it.

Both GIF (both normal and interlaced) and JPEG images can be freely interchanged using this method. You can also specify width and/or height values in the IMG element, and both the high-res and low-res versions of the image will be appropriately scaled to match.

If the images are of different sizes and a fixed height and width are not specified in the IMG element, the second image (the image specified by the SRC attribute) will be scaled to the dimensions of the first (LOWSRC) image. This is because by the time the Netscape Navigator knows the dimensions of the second image, the first image has already been displayed in the document at its normal dimensions.

<INPUT>

The Input element represents a field whose contents may be edited by the user.

Attributes of the Input element are as follows:

ALIGN Vertical alignment of the image. For use only with TYPE=IMAGE in HTML level 2. The possible values are exactly the same as for the ALIGN attribute of the image element.

CHECKED Indicates that a checkbox or radio button is selected. Un-selected checkboxes and radio buttons do not return name/value pairs when the form is submitted.

MAXLENGTH Indicates the maximum number of characters that can be entered into a text field. This can be greater than specified by the SIZE attribute, in which case the field will scroll appropriately. The default number of characters is unlimited.

NAME Symbolic name used when transferring the form's contents. The NAME attribute is required for most input types and is normally used to provide a unique identifier for a field, or for a logically related group of fields.

SIZE Specifies the size or precision of the field according to its type. For example, to specify a field with a visible width of 24 characters:

For Example:

```
INPUT TYPE=text SIZE="24"
```

SRC A URL or URN specifying an image. For use only with TYPE=IMAGE in HTML Level 2.

TYPE Defines the type of data the field accepts. Defaults to free text. Several types of fields can be defined with the type attribute:

CHECKBOX : Used for simple Boolean attributes, or for attributes that can take multiple values at the same time. The latter is represented by a number of checkbox fields each of which has the same name. Each selected checkbox generates a separate name/value pair in the submitted data, even if this results in duplicate names. The default value for checkboxes is "on".

HIDDEN : No field is presented to the user, but the content of the field is sent with the submitted form. This value may be used to transmit state information about client/server interaction.

IMAGE : An image field upon which you can click with a pointing device, causing the form to be immediately submitted. The co-ordinates of the selected point are measured in pixel units from the upper-left corner of the image, and are returned (along with the other contents of the form) in two name/value pairs. The x-co-ordinate is submitted under the name of the field with .x appended, and the y- co-ordinate is submitted under the name of the field with .y appended. Any VALUE attribute is ignored. The image itself is specified by the SRC attribute, exactly as for the Image element.

PASSWORD is the same as the TEXT attribute, except that text is not displayed as it is entered.

RADIO is used for attributes that accept a single value from a set of alternatives. Each radio button field in the group should be given the same name. Only the selected radio button in the group generates a name/value pair in the submitted data. Radio buttons require an explicit VALUE attribute.

RESET is a button that when pressed resets the form's fields to their specified initial values. The label to be displayed on the button may be specified just as for the SUBMIT button.

SUBMIT is a button that when pressed submits the form. You can use the VALUE attribute to provide a non- editable label to be displayed on the button. The default label is application-specific. If a SUBMIT button is pressed in order to submit the form, and that button has a NAME attribute specified, then that button contributes a name/value pair to the submitted data. Otherwise, a SUBMIT button makes no contribution to the submitted data.

TEXT is used for a single line text entry fields. Use in conjunction with the SIZE and MAXLENGTH attributes. Use the Textarea element for text fields which can accept multiple lines.

TEXTAREA is used for multiple-line text-entry fields. Use in conjunction with the SIZE and MAXLENGTH attributes.

FILE The FILE option of the TYPE attribute of the INPUT element, allowing an ACCEPT attribute for the INPUT element (which is a list of media types or type patterns allowed for the input) and allowing the ENCTYPE of a from to be multipart/form-data.
This allows the inclusion of files with form information, which could prove invaluable for example, for companies providing technical support, or service providers, requesting data files.

VALUE The initial displayed value of the field, if it displays a textual or numerical value; or the value to be returned when the field is selected, if it displays a Boolean value. This attribute is required for radio buttons.

<ISINDEX...>

The Isindex element tells the HTML user agent that the document is an index document. As well as reading it, the reader may use a keyword search.

The document can be queried with a keyword search by adding a question mark to the end of the document address, followed by a list of keywords separated by plus signs.

NOTE : The Isindex element is usually generated automatically by a server. If added manually to a HTML document, the HTML user agent assumes that the server can handle a search on the document. To use the Isindex element, the server must have a search engine that supports this element.

To the `<ISINDEX>` element Netscape authors have added the `PROMPT` attribute. `<ISINDEX>` indicates that a document is a searchable index.

`PROMPT` has been added so that text, chosen by the author, can be placed before the text input field of the index. This allows any author chosen message to replace the default text of :

This is a searchable index. Enter search keywords

Another **Netscape** specific attribute is **`ACTION`**. When used in the `<ISINDEX>` element, it specifies the cgi script or program to which the text string in the input box should be passed.

For example:

```
<ISINDEX ACTION="websearch">
```

would pass the text entered into the input box on the page to the cgi script "websearch".

NOTE : "websearch" in the above example is a hypothetical cgi script. If used, the `ACTION` attribute must point to a properly configured script on the host machine. `ACTION` is **Netscape** specific.

<KBD> ... </KBD>

The Keyboard element indicates text typed by a user; typically rendered as monospaced . It might commonly be used in an instruction manual.

For Example:

```
The text inside the <KBD>KBD element, would typically</KBD> render as monospaced
  font.
```

Would be rendered as :

The text inside the `KBD element would typically` render as monospaced font.

<LI>

The LI element defines a list item. It is rendered differently depending upon the list within which it appears

<LINK...>

The Link element indicates a relationship between the document and some other object. A document may have any number of Link elements.

The Link element is empty (does not have a closing element), but takes the same attributes as the <A> element.

Typical uses are to indicate authorship, related indexes and glossaries, older or more recent versions, etc. Links can indicate a static tree structure in which the document was authored by pointing to a ″parent″ and ″next″ and ″previous″ document, for example.

Servers may also allow links to be added by those who do not have the right to alter the body of a document.

<MAP ...> Client Side Image Maps

The current technique for implementing image maps requires communication with an HTTP server to process co-ordinate information and generate a new URL. This element allows browsers to process image maps internally. These extensions allow the use of image maps in local HTML files or files accessed via alternate transport mechanisms such as FTP. They also allow image map characteristics to be specified in a format that is not specific to the server software.

Syntax

Adding a **USEMAP** attribute to an <IMG> element indicates that it is a client-side image map. The USEMAP attribute can be used with the ISMAP attribute to indicate that the image can be processed as either a client-side or server-side image map. The argument to USEMAP specifies which map to use with the image, in a format similar to the HREF attribute on anchors. If the argument to USEMAP starts with a '#', it is assumed to be in the same document as the IMG tag.

A few examples would be:

This method would only work if your browser supports client-side image maps:

```
<IMG SRC="images/Flexed.gif" USEMAP="#mapname">
```

This image map will work regardless, because it specifies a server side map file as well as a client side image map file.

```
<A HREF="/cgi-bin/imagemap/flexed"><IMG SRC="images/flexed.gif USEMAP="#mapname"
  ISMAP></A>
```

Clicking here will take you to a page with an error message if you don't have client-side image map support:

```
<A HREF="nomapsupport.html"><IMG SRC="images/flexed.gif" USEMAP="#mapname"></A>
```

The different regions of the image are described using a **MAP** element. The map describes each region in the image and indicates where it links to. The basic format for the MAP element is as follows:

```
<MAP NAME="name">
   <AREA [SHAPE="shape"] COORDS="x,y,..." [HREF="reference"] [NOHREF]>
</MAP>
```

The MAP definition can reside in the same file as the image that will use it, or in a completely separate file. This way, if you intend to use this type of image map extensively, all the map definitions can be kept separate from the main documents, allowing easier maintenance.

The name specifies the name of the map so that it can be referenced by an <IMG> element. The shape gives the shape of this area. Currently the only shape defined is "RECT", but the syntax is defined in such a way to allow other region types to be added. If the SHAPE tag is omitted, **SHAPE**="RECT" is assumed. The **COORDS** tag gives the coordinates of the shape, using image pixels as the units. For a rectangle, the coordinates are given as "left, top, right, bottom". The rectangular region defined includes the lower-right corner specified, i.e. to specify the entire area of a 100x100 image, the coordinates would be "0,0,99,99".

The NOHREF tag indicates that clicks in this region should perform no action. An HREF tag specifies where a click in that area should lead. Note that a relative anchor specification will be expanded using the URL of the map description as a base, rather than using the URL of the document from which the map description is referenced. If a BASE tag is present in the document containing the map description, that

URL will be used as the base.

An arbitrary number of **AREA** tags may be specified. If two areas intersect, the one which appears first in the map definition takes precedence in the overlapping region. For example, a button bar in a document might use a 160 pixel by 60 pixel image and appear like this:

```
<MAP NAME="mapname">
   <AREA SHAPE="RECT" COORDS="10,10,49,49" HREF="about_us.html">
   <AREA SHAPE="RECT" COORDS="60,10,99,49" HREF="products.html">
   <AREA SHAPE="RECT" COORDS="110,10,149,49" HREF="index.html">
   <AREA SHAPE="RECT" COORDS="0,0,159,59" NOHREF>
</MAP>
<IMG SRC="images/bar.gif" USEMAP="#mapname">
```

This example includes a region encompassing the entire image with a NOHREF tag, but this is actually redundant. Any region of the image that is not defined by an AREA tag is assumed to be NOHREF.

NOTE : The TARGET attribute can be used within the <AREA> element, allowing the use of Client side image maps within framed dcouments.

This syntax provides maximum flexibility to the document author for dealing with browsers which do not support this extension, since such browsers will ignore the MAP and AREA elements. If the document resides on an HTTP server, the server can still provide ISMAP-style support. Otherwise, the author can choose to have the image not appear as an anchor at all, or have a click anywhere within it lead to a another page, perhaps providing an equivalent textual list of options.

Because the map description can reside in an different file additional flexibility is provided. A common use of image maps is a button bar which appears at the bottom of every document. The map description could be specified in one file, such as the server's home page, and referenced from each document. Thus, the map could be modified by changing a single map description rather than having to modify every file on the server. There is also the possibility of advanced applications with servers dynamically generating map descriptions, similar to the way that some servers currently dynamically generate image files.

<MENU> ... </MENU>

A menu list is a list of items with typically one line per item. The menu list style is more compact than the style of an unordered list.

A menu list must begin with a `<MENU>` element which is immediately followed by a `<LI>` (list item) element:

```
<MENU>
    <LI>First item in the list.
    <LI>Second item in the list.
    <LI>Third item in the list.
</MENU>
```

<META...>

The Meta element is used within the Head element to embed document meta-information not defined by other HTML elements. Such information can be extracted by servers/clients for use in identifying, indexing and cataloguing specialised document meta-information.

Although it is generally preferable to use named elements that have well defined semantics for each type of meta-information, such as title, this element is provided for situations where strict SGML parsing is necessary and the local DTD is not extensible.

In addition, HTTP servers can read the content of the document head to generate response headers corresponding to any elements defining a value for the attribute `HTTP-EQUIV`. This provides document authors a mechanism (not necessarily the preferred one) for identifying information that should be included in the response headers for an HTTP request.

Attributes of the Meta element :

`HTTP-EQUIV` This attribute binds the element to an HTTP response header. If the semantics of the HTTP response header named by this attribute is known, then the contents can be processed based on a well-defined syntactic mapping whether or not the DTD includes anything about it. HTTP header names are not case sensitive. If not present, the `NAME` attribute should be used to identify this meta-information and it should not be used within an HTTP response header.

`NAME` Meta-information name. If the name attribute is not present, then name can be assumed equal to the value `HTTP-EQUIV`.

`CONTENT` The meta-information content to be associated with the given name and/or HTTP response header.

As For Example, if the document contained:

```
<META HTTP-EQUIV="Expires"  CONTENT="Sun, 25 Dec 1996 12:00:00 EST">
<META HTTP-EQUIV="Keywords" CONTENT="HTML, Editor">
<META HTTP-EQUIV="Reply-to" CONTENT="info@infoflex.com.au">
```

Then the HTTP response header would be :

Expires: Sun, 25 Dec 1996 12:00:00 EST
Keywords: HTML, Editor
Reply-to: info@infoflex.com.au

When the `HTTP-EQUIV` attribute is not present, the server should not generate an HTTP response header for this meta-information.

For example:

```
<META NAME="IndexType" CONTENT="Service">
```

Do *not* use the Meta element to define information that should be associated with an existing HTML element.

Example of an inappropriate use of the Meta element :

```
<META NAME="Title" CONTENT="The Etymology of Dunsel">
```

Do *not* name an `HTTP-EQUIV` equal to a responsive header that should typically only be generated by the HTTP server. Some inappropriate names are ″Server″, ″Date″ and ″Last-modified″. Whether a name is inappropriate depends on the particular server implementation. It is recommended that servers ignore any Meta elements that specify HTTP-equivalents equal (case-insensitively) to their own reserved response headers.

<NOBR>

The <NOBR> element stands for **NO BR**eak. This means all the text between the start and end of the <NOBR> elements cannot have line breaks inserted. While <NOBR> is essential for those character sequences that don't want to be broken, please be careful; long text strings inside of <NOBR> elements can look rather odd. Especially if during viewing, the user adjusts the page size by altering the window size.

<NOEMBED>

The NOEMBED element defines content within EMBED content that is to be ignored by browsers that can activiate the EMBED plug-in application. Browsers that can't/won't activate the EMBED plug-in but that understand the EMBED/NOEMBED elements or browsers that do not understand the EMBED/NOEMBED elements will display the NOEMBED content.

<NOFRAMES> ... </NOFRAMES>

The NOFRAMES element defines content within <FRAMESET> content that is to be ignored by browsers that can define Frames. Browsers that can't/won't define Frames but that understand the FRAMESET/NOFRAMES elements or browsers that do not understand the FRAMESET/NOFRAMES elements will display the NOFRAMES content

<OL> ... </OL>

The Ordered List element is used to present a numbered list of items, sorted by sequence or order of importance.

An ordered list must begin with the <OL> element which is immediately followed by a <LI> (list item) element:

As For Example:

```
<OL>
   <LI>This is Instruction 1
   <LI>This is Instruction 2
   <LI>This is The Final Instruction
</OL>
```

Would Be Rendered As:

1. This is Instruction 1
2. This is Instruction 2
3. This is The Final Instruction

The Ordered List element can take the COMPACT attribute, which suggests that a compact rendering be used.

The average ordered list counts 1, 2, 3, ... etc. The **TYPE** attribute has been added to this element to allow authors to specify whether the list items should be marked with:

(TYPE=A) - capital letters. e.g. A, B, C ...
(TYPE=a) - small letters. e.g. a, b, c ...
(TYPE=I) - large roman numerals. e.g. I, II, III ...
(TYPE=i) - small roman numerals. e.g. i, ii, iii ...
(TYPE=1) - or the default numbers. e.g. 1, 2, 3 ...

For lists that wish to start at values other than 1 the new attribute **START** is available.
START is always specified in the default numbers, and will be converted based on TYPE before display. Thus START=5 would display either an 'E', 'e', 'V', 'v', or '5' based on the TYPE attribute.

To give even more flexibility to lists, the **TYPE** attribute has been added to the <LI> element as well. It takes the same values as <OL> and it changes the list type for that item, and all subsequent items. For ordered lists we have also added the **VALUE** element so you can change the count, for that list item and all subsequent.

<OPTION>

The Option element can only occur within a Select element. It represents one choice, and can take these attributes:

SELECTED Indicates that this option is initially selected.

VALUE When present indicates the value to be returned if this option is chosen. The returned value defaults to the contents of the Option element.

The contents of the Option element is presented to the user to represent the option. It is used as a returned value if the VALUE attribute is not present.

<P> ... </P>

The Paragraph element **<P>** indicates a new paragraph. The exact indentation, leading, etc. of a paragraph is not defined and may be a function of other elements, style sheets, etc.

Typically, paragraphs are surrounded by a vertical space of one line or half a line. This is typically not the case within the Address element and or is never the case within the Pre-formatted Text element. With some HTML user agents, the first line in a paragraph is indented.

Example of use:

```
<H1>This is your document headline</H1>
<P>This is the first paragraph of text.
<P>This is the second paragraph of text.
<P>This is the third paragraph of text.
```

Included is the ability to align paragraphs using the `ALIGN` attribute to align left, right or center. The Syntax, **`ALIGN`**`=left|center|right` attributes have been added to the `<P>` element.

Example of use:

```
<P ALIGN=RIGHT> ... </P>
```

All text with in the paragraph will be aligned to the right side of the document layout.

<PRE> ... </PRE>

The Pre-formatted Text element presents blocks of text in fixed-width font, and so is suitable for text that has been formatted on screen.

The `<PRE>` element may be used with the optional `WIDTH` attribute, which is a Level 1 feature. The `WIDTH` attribute specifies the maximum number of characters for a line and allows the HTML user agent to select a suitable font and indentation. If the `WIDTH` attribute is not present, a width of 80 characters is assumed. Where the `WIDTH` attribute is supported, widths of 40, 80 and 132 characters should be presented optimally, with other widths being rounded up.

Within pre-formatted text:

- Line breaks within the text are rendered as a move to the beginning of the next line.
- The `<P>` element should not be used. If found, it should be rendered as a move to the beginning of the next line.
- Anchor elements and character highlighting elements may be used.
- Elements that define paragraph formatting (headings, address, etc.) must not be used.
- The horizontal tab character (encoded in US-ASCII and ISO-8859-1 as decimal 9) must be interpreted as the smallest positive nonzero number of spaces which will leave the number of characters so far on the line as a multiple of 8. Its use is not recommended however.

NOTE: References to the "beginning of a new line" do not imply that the user is forbidden from using a constant left indent for rendering pre-formatted text. The left indent may be constrained by the width required.

Example of use:

```
<PRE WIDTH="80">
This is an example line.
</PRE>
```

NOTE: Within a Pre-formatted Text element, the constraint that the rendering must be on a fixed horizontal character pitch may limit or prevent the ability of the HTML user agent to render highlighting elements specially.

<SAMP> ... </SAMP>

The Sample element indicates a sequence of literal characters; typically rendered as monospaced.

For Example:

```
A sequence of <SAMP>literal characters</SAMP> commonly renders as monospaced font.
```

Would be rendered as :

A sequence of `literal characters` commonly renders as monospaced font.

<SELECT ...> ... </SELECT>

The Select element allows the user to choose one of a set of alternatives described by textual labels. Every alternative is represented by the Option element.

Attributes are:

MULTIPLE The `MULTIPLE` attribute is needed when users are allowed to make several selections, e.g. `<SELECT MULTIPLE>`.

NAME Specifies the name that will submitted as a name/value pair.

SIZE Specifies the number of visible items. If this is greater than one, then the resulting form control will be a list.

The `SELECT` element is typically rendered as a pull down or pop-up list. For example:

```
<SELECT NAME="Type">
  <OPTION>Type 1
  <OPTION>Type 2
  <OPTION>Type 3
</SELECT>
```

If no option is initially marked as selected, then the first item listed is selected.

<SMALL> ... </SMALL>

The <SMALL> element specifies that the enclosed text should be displayed, if practical, using a small font (compared with the current font). This is an HTML 3.0 element and may not be widely supported.

```
This is an <SMALL>Example of Small Text</SMALL> in your Document.
```

Would be rendered as:

This is an Example of Small Text in your Document.

<STRONG> ... </STRONG>

The Strong element indicates strong typographic emphasis, typically rendered in bold.

```
This is an <STRONG>Example of Strong Typeface</STRONG> in your document.
```

Would be rendered as:

This is an **Example of Strong Typeface** in your document.

_{...}

The <SUB> element specifies that the enclosed text should be displayed as a subscript, and if practical, using a smaller font (compared with normal text). This is an HTML 3.0 element and may not be widely supported.

```
This is an example of <SUB>Subscript Text</SUB> in your document.
```

Would be Rendered as:

This is an example of Subscript Text in your document.

^{...}

The <SUP> element specifies that the enclosed text should be displayed as a superscript, and if practical, using a smaller font (compared with normal text). This is an HTML 3.0 element and may not be widely supported.

```
This is an example of <SUP>Superscript Text</SUP> in your document.
```

Would be Rendered as:

This is an example of Superscript Text in your document

<TABLE> ... </TABLE>

This is the main element for all the other table elements, and other table elements will be ignored if they aren't found inside of a <TABLE> ... </TABLE> element. By default tables have no borders, borders will be added if the BORDER attribute is specified. At the time of writing, the <TABLE> element has an implied line break both before and after it. This is expected to change, allowing as much control over placement of tables as is currently available for the placement of images. Aligning them to various positions in a line of text, as well as shifting them to the left or right margins and wrapping text around them.

The <TABLE> element has the following attributes :

BORDER: This attribute appears in the <TABLE> element. If present, borders are drawn around all table cells. If absent, there are no borders, but by default space is left for borders, so the same table with and without the BORDER attribute will have the same width.

```
BORDER=<value>
```

By allowing the BORDER attribute to take a value, the document author gains two things. First they gain the ability to emphasise some tables with respect to others, a table with a border of four containing a sub-table with a border of one looks much nicer than if they both share the same default border width. Second, by explicitly setting border to zero they regain that space originally reserved for borders between cells, allowing particularly compact tables.

CELLSPACING Cell spacing is the amount of space inserted between individual cells in a table. The Cell Spacing give the HTML author a little more control on appearance of the table.

```
CELLSPACING=<Value>
```

CELLPADDING is the amount of space between the border of the cell and the contents of the cell. Setting a cell padding of zero on a table with borders might look bad because the edges of the text could touch the cell borders.

```
CELLPADDING=<Value>
```

The Most Compact table possible is:

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0>
```

WIDTH: When this attribute appears in the <TABLE> element it is used to describe the desired width of this table, either as an absolute width in pixels, or a percentage of document width. Ordinarily complex heuristics are applied to tables and their cells to attempt to present a pleasing looking table. Setting the <WIDTH> attribute overrides those heuristics and instead effort is put into fitting the table into the desired width as specified. In some cases it might be impossible to fit all the table cells at the specified width, in which case Netscape will try and get as close as possible.

When this attribute appears on either the <TH> or <TD> element it is used to describe the desired width of the cell, either as an absolute width in pixels, or a percentage of table width. Ordinarily complex heuristics are applied to table cells to attempt to present a pleasing looking table. Setting the <WIDTH> attribute overrides those heuristics for that cell and instead effort is put into fitting the cell into the desired width as specified. In some cases it might be impossible to fit all the table cells at the specified widths, in

which case Netscape will try and get as close as possible.

```
WIDTH=<value_or_percent>
```

HEIGHT: Using this attribute in the <TABLE> element, describes the height of the table, either as a particular pixel value, or as a percentage of the display window. This attribute is also supported within the <TH> and <TD> elements.

```
HEIGHT=<value_or_percent>
```

ALIGN: Like that used for floating images, it allows a table to be aligned to the **left** or **right** of the page (Horizontally), allowing text to flow around the table. Also, as with floating images, it is necessary to have knowledge of the <BR CLEAR=…> element, to be able to organise the text so as to minimise any unwanted clashing.

VALIGN: Specifies that the text can be **top**- or **bottom**-aligned. The default is centre-aligned. This is also called Vertical Alignment.

<TD ...> ... </TD>

This stands for table data <TD>, and specifies a standard table data cell. Table data cells must only appear within table rows. Each row need not have the same number of cells specified as short rows will be padded with blank cells on the right. A cell can contain any of the HTML elements normally present in the body of an HTML document. The default alignment of table data is `ALIGN=left` and `VALIGN=middle`. These alignments are overridden by any alignments specified in the containing `<TR>` element, and those alignments in turn are overridden by any `ALIGN` or `VALIGN` attributes explicitly specified on the <TD> cell. By default lines inside of table cells can be broken up to fit within the overall cell width. Specifying the `NOWRAP` attribute for a `<TD>` prevents line breaking for that cell.

`<TD ...> ... </TD>` can accept the following attributes.

`ALIGN` This attribute controls whether text inside the table cell(s) is aligned to the left side of the cell, the right side of the cell, or centred within the cell. Values are **left**, **center**, and **right**.

`VALIGN` The `VALIGN` attribute controls whether text inside the table cell(s) is aligned to the top of the cell, the bottom of the cell, or vertically centred within the cell. It can also specify that all the cells in the row should be vertically aligned to the same baseline. Values are **top**, **middle**, **bottom** and **baseline**.

`WIDTH` When this attribute is used is describes the desired width of the cell, either as an absolute width in pixels, or a percentage of table width. Ordinarily complex heuristics are applied to table cells to attempt to present a pleasing looking table. Setting the `<WIDTH>` attribute overrides those heuristics for that cell and instead effort is put into fitting the cell into the desired width as specified. In some cases it might be impossible to fit all the table cells at the specified widths, in which case browsers will try and get as close as possible.

`HEIGHT` Using this attribute in the `<TABLE>` element, describes the height of the table, either as a particular pixel value, or as a percentage of the display window. This attribute is also supported within the <TH> and <TABLE> elements.

`NOWRAP` If this attribute appears in any table cell (<TH> or `<TD>`) it means the lines within this cell cannot be broken to fit the width of the cell. Be cautious in use of this attribute as it can result in excessively wide cells.

`COLSPAN` This attribute can appear in any table cell (<TH> or `<TD>`) and it specifies how many columns of the table this cell should span. The default `COLSPAN` for any cell is 1.

`ROWSPAN` This attribute can appear in any table cell (<TH> or `<TD>`) and it specifies how many rows of the table this cell should span. The default `ROWSPAN` for any cell is 1. A span that extends into rows that were never specified with a <TH> will be truncated.

<TEXTAREA> ... </TEXTAREA>

The TEXTAREA element lets users enter more than one line of text. For example:

For Example:

```
<TEXTAREA NAME="address" ROWS=70 COLS=7>
     Infoflex Pty Ltd
     27, 15 Terminus St
     Castle hill NSW 2154
     Australia
</TEXTAREA>
```

The text up to the end element (</TEXTAREA>) is used to initialise the field's value. This end element is always required even if the field is initially blank. When submitting a form, lines in a TEXTAREA should be terminated using CR/LF.

In a typical rendering, the **ROWS** and **COLS** attributes determine the visible dimension of the field in characters. The field is rendered in a fixed-width font. HTML user agents should allow text to extend beyond these limits by scrolling as needed.

NOTE: In the initial design for forms, multi-line text fields were supported by the Input element with TYPE=TEXT. Unfortunately, this causes problems for fields with long text values. SGML's default (Reference Quantity Set) limits the length of attribute literals to only 240 characters. The HTML 2.0 SGML declaration increases the limit to 1024 characters.

The **WRAP** attribute in the TEXTAREA element is now introduced. Now it is possible to specify how to handle word-wrapping in text input areas in forms.

<TEXTAREA WRAP=OFF> -- the default setting - Wrapping doesn't happen. Lines are sent exactly as typed.

<TEXTAREA WRAP=VIRTUAL> -- The display word-wraps, but long lines are sent as one line without new-lines.

<TEXTAREA WRAP=PHYSICAL> -- The display word-wraps, and the text is transmitted at all wrap points.

NOTE : Word wrapping in a TEXTAREA text box is **Netscape specific** at present**.**

<TH ...> ... </TH>

This stands for table header. Header cells are identical to data cells in all respects, with the exception that header cells are in a **bold** FONT, and have a default `ALIGN=center`.

`<TH ...> ... </TH>` can contain the following attributes

ALIGN The `ALIGN` attribute controls whether text inside the table cell(s) is aligned to the left side of the cell, the right side of the cell, or centred within the cell. Values are **left**, **center**, and **right**.

VALIGN This attribute controls whether text inside the table cell(s) is aligned to the top of the cell, the bottom of the cell, or vertically centred within the cell. It can also specify that all the cells in the row should be vertically aligned to the same baseline. Values are **top**, **middle**, **bottom** and **baseline**.

WIDTH When this attribute is used is describes the desired width of the cell, either as an absolute width in pixels, or a percentage of table width. Ordinarily complex heuristics are applied to table cells to attempt to present a pleasing looking table. Setting the `<WIDTH>` attribute overrides those heuristics for that cell and instead effort is put into fitting the cell into the desired width as specified. In some cases it might be impossible to fit all the table cells at the specified widths, in which case browsers will try and get as close as possible.

HEIGHT Using this attribute in the `<TABLE>` element, describes the height of the table, either as a particular pixel value, or as a percentage of the display window. This attribute is also supported within the <TABLE> and <TD> elements.

NOWRAP This attribute specifies that the lines within this cell cannot be broken to fit the width of the cell. Be cautious in use of this attribute as it can result in excessively wide cells.

COLSPAN The `COLSPAN` attribute specifies how many columns of the table this cell should span. The default `COLSPAN` for any cell is 1.

ROWSPAN This attribute specifies how many rows of the table this cell should span. The default `ROWSPAN` for any cell is 1. A span that extends into rows that were never specified with a `<TR>` will be truncated.

<TITLE> ... </TITLE>

Every HTML document must have a Title element. The title should identify the contents of the document and in a global context, and may be used in history lists and as a label for the windows displaying the document. Unlike headings, titles are not typically rendered in the text of a document itself.

The Title element must occur within the head of the document and may not contain anchors, paragraph elements, or highlighting. Only one title is allowed in a document.

NOTE : The length of a title is not limited, however, long titles may be truncated in some applications. To minimise the possibility, titles should be fewer than 64 characters. Also keep in mind that a short title, such as 'Introduction' may be meaningless out of context. An example of a meaningful title might be 'Introduction to HTML elements'

This is the ***only*** element that is ***required*** within the Head element. The other elements described are optional and can be implemented when appropriate

```
<HEAD>
   <TITLE> Introduction to HTML</TITLE>
</HEAD>
```

<TR ...> ... </TR>

This stands for table row <TR>. The number of rows in a table is exactly specified by how many `<TR>` elements are contained within it, regardless of cells that may attempt to use the `ROWSPAN` attribute to span into non-specified rows. `<TR>` can have both the `ALIGN` and `VALIGN` attributes, which if specified become the default alignments for all cells in this row.
The `<TR>` element can have the following attributes.

`ALIGN` This controls whether text inside the table cell(s) is aligned to the left side of the row, the right side of the cell, or centred within the cell. Values are **left**, **center**, and **right**.

`VALIGN` This attribute controls whether text inside the table cell(s) is aligned to the top of the row, the bottom of the cell, or vertically centred within the cell. It can also specify that all the cells in the row should be vertically aligned to the same baseline. Values are **top**, **middle**, **bottom** and **baseline**.

<TT> ... </TT>

The Teletype element specifies that the text should be rendered in fixed-width typewriter font.

```
This is an <TT>Example of Teletype Text</TT> in your document.
```

Would be rendered as:

This is an `Example of Teletype Text` in your document.

<U> ... </U>

The Underline Element specifies that your text will be rendered as underlined

```
This is an <U>Example of Underlined text</U> in your document.
```

Would be rendered as:

This is an Example of Underlined text in your document.

<UL> ... </UL>

The Unordered List element is used to present a list of items which is typically separated by white space and/or marked by bullets.

An unordered list must begin with the `<UL>` element which is immediately followed by a `<LI>` (list item) element:

As For Example:

```
<UL>
   <LI>First list item
   <LI>Second list item
   <LI>Third list item
</UL>
```

Would be Rendered As:

- First List Item
- Second List Item
- Third List Item

The Unordered List element can take the `COMPACT` attribute, which suggests that a compact rendering be used.

The basic bulleted list has a default progression of bullet types that changes as you move through indented levels. From a solid disc, to a circle to a square. Netscape authors have added a **`TYPE`** attribute to the `<UL>` element so that no matter what the indent level the bullet type can be specified as Disc, circle or square.

To give even more flexibility to lists, Netscape authors have added the **`TYPE`** attribute to the `<LI>` element as well. It takes the same values as `<UL>` and it changes the list type for that item, and all subsequent items.

<VAR> ... </VAR>

The Variable element indicates a variable name; typically rendered as italic.

```
When coding, <VAR>LeftIndent()</VAR> must be a variable
```

Would render as :

When coding *LeftIndent()* must be a variable.

<WBR>

The <WBR> element stands for **W**ord **BR**eak. This is for the very rare case when a <NOBR> section requires an exact break. Also, it can be used any time the Netscape Navigator can be helped by telling it where a word is allowed to be broken. The <WBR> element does not force a line break (
 does that) it simply lets the Netscape Navigator know where a line break is allowed to be inserted if needed.

Glossary

Anchor
Text and graphics can link to places within the same document. These links require two parts: the anchor and the link. The anchor identifies the place to jump to.

Anonymous FTP
FTP transactions that do not require a unique login name or password. Use the name *anonymous* to log in and use your e-mail address as a password.

Argument
Words or numbers you enter as part of an HTML tag to expand or modify how that tag operates.

ASCII
Acronym for American Standard Code for Information Interchange, a standard character set.

Authoring software
Computer programs that aid in creating HTML documents by inserting the code for tags.

Body
HTML tag used to enclose the body (all the text and tags) of the HTML document.

Browser
A program used to view HTML documents and navigate the World Wide Web. See *Lynx, Mosaic,* and *Netscape.*

Byte
Eight bits; the fundamental unit of persona computer data.

CERN
European Particle Physics Laboratory, the developers of the World Wide Web.

CGI
(Common Gateway Interface) The scripting language used to write gateway scripts for CERN and NCSA Web Servers.

Checkboxes
User in forms to make it possible to select one or more non-exclusive options in a list.

Coorindates
Pairs of numbers used to define the edges of the clickable areas in clickable image maps.

Dedicated connection
Access to the Internet that is always available via a special connection such as FDDI, T1, or switched 56. See *dial-up connections* and *ISDN*.

Dial-up connections
Accessing the Internet by means of a modem and communication software.

Directory
A major division on a hard drive or server used to divide and organise files.

Domain
The name for a company, organisation, or individual's Internet connection. Individual computers within this domain all end with the domain as a part of their host name.

Download
To transfer a file from another computer to your computer.

E-mail (electronic mail)
A communication system that allows you to send an electronic "letter" to one or more recipients.

Ethernet
A standard for local area network hardware, cabling, and transmission.

External files
Any files that are not directly loadable by a browser such as some image formats, sounds, videos, or even program files.

FDDI
A type of high-speed dedicated connection to the Internet with a speed of 45MB per second.

File formats
The patterns and standards used to store a program on a disk. Examples are GIF, JPEG, AIFF.

Form
HTML documents designed with fill-in text foxes, lists of options, and other elements that allow the user of the form to send information back to the Web server.

Freeware
Software that is distributed at no cost to the user (the author maintains the copyright).

FTP (File Transfer Protocol)
The primary method for accessing files via the Internet.

Gateway script
A program that is run on a Web server that processes the input from forms.

GIF
A popular type of image file format.

Gopher
A menu-based information system on the Internet popularised because of its ability to interconnect different Gopher sites on the same menu.

Graphical browser
A program used to view formatted HTML documents and Navigate the World Wide Web. Graphical browsers can display inline images and display text in various type styles. Example are Mosaic and netscape.

Head
The HTML tag used to enclose the beginning elements in the HTML document, including the title.

Home page
The first HTML document that you intend people to see at your Web site.

Host name
The name of a computer on the Internet, used to identify it in the URL naming scheme.

HTML (Hyper Text Markup Language)
The coding scheme used to format text for use on the World Wide Web.

HTTP (Hyper Text Transport Protocol)
The transmission standard used to send HTML documents across the World Wide Web.

Hyper Text
An interlinked document structure that allows you to jump freely from one topic or document to another.

Internet
The general term used to describe the worldwide network of computers and services encompassing some 20-40 million computer users and dozens of information systems including e-mail, Gopher, FTP, and the World Wide Web.

Internet access fees
Costs charged by Internet access providers to connect to the Internet.

Internet access provider
A company that sells connections to the Internet to other companies and individuals. Also called *Internet service provider*.

ISDN (Integrated Digital Services Network)
A high-speed dial-up connection to the Internet. Availability and cost are determined in part by local telephone companies.

Linear document
A type of organisation for HTML documents in which one HTML file follows the next. The HTML author determines the order in which the information is presented.

Link
The text or graphic used in an HTML document to jump from one document to another.

Load time
The amount of time it takes a user to retrieve a Web page and view it in a browser on their computer.

Lynx
A non-graphics browser for UNIX and DOS systems. See also *Mosaic* and *Netscape*.

Macintosh
A brand of personal computers manufactured by Apple Computer.

Menu
An on-screen display that lists available choices.

Method
The manner in which an HTML form is submitted to the Web server. These most common method is *post*.

Modem
A device that converts digital information from a computer to analog information that can be sent over telephone lines. This allows computers equipped with modems to communicate over telephone lines.

Monospaced font
A typeface in which the width of each character is the same.

Mosaic
One of the first graphical browsers. Developed by the NCSA, this browser fueled the growth of the Web. It is available in versions for Windows, Mac, and UNIX.

MPEG (Motion Picture Experts Group)
A standard and file format for motion video on computers.

Multimedia
Documents that combine text, graphics, sound, movies, or other media.

Named entity
Special characters whose HTML code is an ampersand (&) followed by a name.

NCSA (National Center for Supercomputing Applications)
The research group that developed Mosaic, a popular graphical browser.

Netscape
A popular commercial graphical browser. It is available in versions for Windows, Mac, and UNIX.

Network administrator
The person responsible for maintaining a network and assisting its users.

Non-graphical browser
A program used to view HTML documents and Navigate the World Wide Web. Non-graphical browsers do not display inline images and do not display formatted text. An example is Lynx.

Non-linear document
A type of organisation for HTML documents in which one HTML file contains links to more than one other HTML file. The person browsing the files determines the order in which the information is presented.

Numbered entity
Special characters whose HTML code is an ampersand (&) followed by a number.

Operating system
A master control program for a computer. Examples are DOS, UNIX, and the MacOS. (Technically speaking, Microsoft Windows 3.x is an operating environment as it still requires the separate DOS operating system to run. Microsoft's Windows 95 is a true operating system as DOS is built-in to it.)

Path name
The place where a file is stored on a computer, indicated by the drive or volume name and the sub directories needed to find the file. See *relative path.*

Platform
A computer hardware standard, such as IBM PC compatible or Macintosh.

Post
To place an HTML file on a Web server to make it available for browsing.

Preference setting
Program options in browsers that allow the user to determine such things as which fonts are used for various HTMP styles, whether or not inline images are displayed, what other applications are used to view movies, and so on.

Proportional font
A typeface in which the width of each character varies depending on the characters shape. An I takes up less space than we for example.

Public domain software
Software that is made freely available by the developer and to which the developer gives up all copy-right

ownership.

Radio button
Used in forms to make it possible to select one exclusive option in a list.

RAM (Random-Access memory)
The computer's primary working memory in which program instructions and data are stored.

Relative path
Relative paths spell out the location of a file based on the current document location.

Router
A device connecting a LAN to the Internet that routes transmissions between the two.

Searchable Index
Indexes of World Wide Web documents that you can submit a query to and the index will search to find pages matching your specification.

Server See *Web server.*

Shareware
Software that you can obtain for free (often by downloading from the Internet) on a trial basis but that may require some payment or registration for continued use.

Sound board
Hardware used to play sound files on computers.

Submit
To send a completed form to a Web server.

TI
A high-speed dedicated connection to the Internet that provides data transmission rates of 1.5MB per second.

Tag
The HTML codes used to specify text styles, links, graphics, and other HTML elements.

Text fields
Form elements that allow users to insert a single line of text.

Title
The HTML tag used to give each HTML document a title.

Transmission line
The physical connection from your computer to the Internet such as a telephone line or a T1.

Transparent graphics
Graphic images that have a clear background, which makes the graphic appear to float in the browser screen.

UNIX
An operating system used on a variety of computers from personal computers to mainframe. Many computers and servers connected to the Internet use UNIX.

URL (Uniform Resource Locator)
The standard used to identify files on the Internet and World Wide Web using the type of server, the host

name of the computer the file is on, and the complete path to the file.

Video board
Hardware used to play movie files on computers.

WAV
A popular sound file format used primarily by Windows based computers.

Web server
The hardware and software used to store and deliver HTML documents for use on the World Wide Web.

Web site
A person or company's collection of HTML documents on a Web server. A single Web server may contain one or more Web sites.

Windows
An operating environment for the IBM PC-compatible platform that allows several programs to run at once and utilises icons and menus for program control.

Word processor
A program used to create and edit text documents. When using a word processor to create HTML documents, save the documents as ASCII text rather than in the word processor's proprietary format.

World Wide Web (WWW)
An Internet service that links multimedia documents together using hypertext. Users can jump between documents using links to view text, graphics, movies, and other media.

Editor Window

The Editor Window in FlexED is a powerful Word Processing Tool. The Editor Window is the core of your HTML Text, and provides a quick and easy way of editing and creating HTML documents. The following features are available to you as an editor:

- Syntax highlighting: That is, all the HTML tags or elements are displayed in your editor as a different color (definable), which makes your HTML Text easy to read and easy to distinguish between your rendered code and the HTML tags.
- Double clicking on tags: The color syntax highlighting tags leads to one of the most powerful features in FlexED. By double Clicking on these Color coded Tags, a dialog box will appear for all the parameters and options available for formatting the tag and elements relating to that tag. This means you dont have to remember all the parameters of the Tag.
- Sync browser: You can synchronize the internal Browser with the Editor. In other words, if you split your viewer and document vertically or horizontally, and scroll down your HTML document, the viewer will synchronize with the editor and display the rendered document where you cursor is placed. This option is located in the text pull down menu in FlexED.
- Converting Case Upper/Lower: You can highlight all or particular areas of your document, and from the Text option in FlexED, convert it to uppercase or lowercase. You may also choose from the system setup, whether you require you tags to be Upper or Lower case.
- Cutting, Pasting and General Editing: The FlexED editor also allows you to Cut, Paste, Search and replace, find, undo and so on. All the helpful editing functions contained in all the most popular editors are incorporated into FlexED. There are also the standard short cut keys to access these tools, while also available is the use of clicking the right mouse button to access these options.
- Word Wrap: FlexED now has a word wrap option available from the Text options menu on the FlexED Menu. This is useful when importing data or cutting and pasting from Word Documents or the like to FlexED.
- Auto Indent: Select this option from the Text Menu, and this will format and indent your HTML code to allow the document source to be easily read and distinguished, making maintenance of the document easier.
- Fonts: You may also change Font, size, color and type of save type from the properties option within FlexED. These all help in the appearance of your HTML document when viewing.

Built In Browser

FlexED contains its own Internal Browser to view your HTML documents. The browser is accessible from the refresh HTML Viewer button  located on the FlexED main toolbar. While you are editing your HTML document, by selecting this Refresh Viewer option, it will instigate the viewer to render your HTML document and display the results in FlexEDs own Internal viewer. This means you do not require any external Web Browsers to display your Web pages and eliminates tabbing to different programs to view your rendered pages. The Internal Browser supports most HTML tags. There is also a Sync Browser option located in the Text Pull down menu from the Main Menu System. This allows you to Synchronize the Internal Viewer with your HTML document. In other words, by selecting this option, when scrolling down your HTML document, the viewer will render that particular section in your document.

You also have the option of rendering your results to two different external browsers. By accessing the properties dialog box and then by selecting Ext Browser, FlexED will allow setup of two different External Browsers. By default, Netscape should be setup as External Browser 1 and Internet Explorer as External Browser 2. Then by selecting the refresh External Browser buttons on the FlexED main Toolbar, will render your HTML documents to the designated Browser.

The Internal Browser is also sold as a separate product - **FlexView**. This product can be used in electronic distribution of HTML documents on CDROM or the like.

Custom ToolBar

FlexED contains a custom HTML Toolbar. That is, the HTML Toolbar is totally customizable, and new tags, your own defined tags and custom tags may be added to this Toolbar.

To access this option, select file from the FlexED main menu, and then select Edit HTML toolbar. The custom Toolbar dialog box will be displayed. The toolbar editor is split into 6 sections, Doc, Text, Form, list, custom and all (All tags grouped together). Within each group are the tags specific to those groups, for example, the Text group have all the text formatting tags.

The custom toolbar allows insertion of new tags, but also allows defined tags that will insert custom text into your HTML document. To add a new tag, select the tab sheet for the group where your new tag is to be located. Select the Add button to add your new tag. You will be prompted for the new Tag name. Select from the options below the custom text area, whether you require an end tag or not, for example if your new tag is <NEWTAG>, selecting the End Tag option will insert <NEWTAG></NEWTAG> into your HTML document.

To insert tags and text as custom text into your HTML document, again follow the above procedures, but select the custom option box under the custom text area. Then in the custom Text area, type in the HTML code you wish to insert into your HTML document. As for example, if your new tag is called CUSTOMHEAD, and you type <H1>This Heading Will <I>Appear</I></H1>, into the custom text area, this HTML code will be inserted into your HTML document when clicking on this element in the HTML Toolbar.

To test what will be inserted into your HTML document, press the test button to make sure your syntax is correct. Other options available are date/time. This will insert the date and time into your document. By selecting break will insert a hard return after your tag.

NOTE: The Custom toolbar does not save new tags until FlexED has been registered.

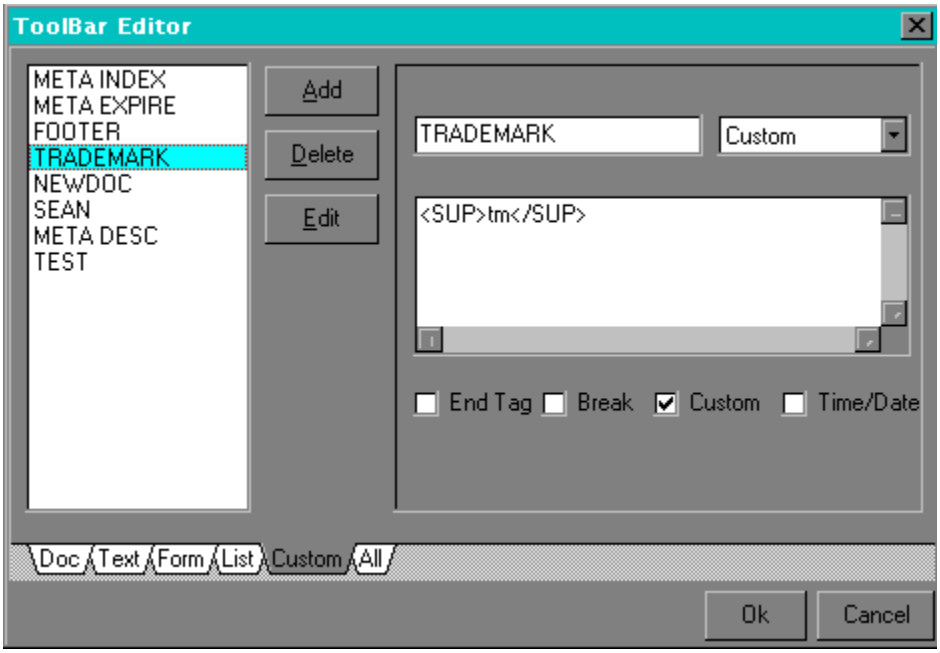


Image Map Wizard

FlexED allows simple creation of Image Maps in your HTML document. Image Maps are images in your document that have a reference or multiple references, to other documents, URLs or any other HTML reference. That is, you may have an image with multiple hotspots or Hypertext links, and depending on where you click the mouse on the image, a certain set of instructions will be performed or executed.

To create a map Image in FlexED, select the  Insert Image button on the main Toolbar and then select Map Wizard from the pull down menu. The Image Map Wizard dialog box will now display. Select File| Open to open the image with which you want to create a Map Image. Select OK when you have found the image. This image will now be displayed in the Map Wizard Dialog Box. If you move the cursor over the image a + (Cross) will appear. Plan where you want to draw the hotspot, then hold down the mouse button and drag the cursor and a bordered box will appear. This will be your hotspot. Create this box as large as you want the hotspot to be. Next, you must define the properties of the hotspot. Position the mouse inside the border and click on the right mouse button and select properties from the pull down menu. The Href Dialog box will then appear. Click on the options […] button to call the Href Builder dialog box. Select the resource and other relevant information to complete your hotspot. OK to save parameter. Repeat the above to create more than one hotspot as many times as you like. Finally, give your Image Map a name (located in the bottom left hand corner and then OK to save details.

The following HTML code is an example created by the FlexED Image Map Wizard.

```
<BODY>
   <MAP NAME="MyImageMap">
      <AREA COORDS="5,8,39,41"   HREF="http://infoflex.com.au">
      <AREA COORDS="32,50,72,85" HREF="mailto:info@infoflex.com.au">
   </MAP>
   <IMG BORDER=0 USEMAP="#MyImageMap" SRC="images//infoflex.gif">
</BODY>
```

NOTE: To easily change the parameters of the Image Map, Highlight the text from <Map> to </Map> and then click the right mouse button. Select Edit Image Map from the Pull Down Menu.

For Further Reference See:

Creating Clickable Image Maps
<MAP ...> Client Side Image Maps
<AREA> ... </AREA>
<IMG...> In-line images
HRef Builder

Frame Wizard

An alternate way to lay out your pages is to use frames. As the name suggests, you can have multiple frames of information appearing in one single screen. For instance, you may have an index in one frame, and the information relating to each index may appear in another frame, so the index becomes static for further reference.

FlexED contains a Frame wizard to create pages with frames.

- How to use
- Reference to tags created by image map wizard

Table Wizard

Tables are one of the most useful elements in HTML. Tables can be used to control the layout of your page for purposes of formatting and layout of text and images, and is especially useful in creating aligned columns of text, images and data. For instance, you can use a table to insert an image on the left hand side of your page, and have your text beside that image on the right hand side. You may also use tables to Align your data without using the Pre-formatted fixed width text.

The FlexED HTML editor has a table wizard to create your table easily without any HTML coding. This wizard is located on the Main toolbar on the insert table toolbar. Select the  Table Wizard from the pull down menu. The table editor dialog box will appear. The parameters for the table wizard are as follows:

Caption:	This is the table Name or description that is rendered. This is an Optional field.
Align:	Align The Caption to the left, right, bottom or top of your table.
Headings:	Column or Row - Will bold the column or row headings depending on selection. This is also an optional field, and if left blank will not insert headings.
Table Rows:	Number of rows required.
Columns:	Number of columns required.
Cell Spacing:	Spaces between each cell.
Cell Padding:	Determines size of cell.
Border:	Require a border or no border.

Once these parameters have been entered, click OK and FlexED will paste your table information into your HTML document. Below is an example of the HTML code that would have been created by the table wizard.

```
<BODY>
   <TABLE BORDER=1>
      <TH>Image </TH>
      <TH>Description </TH>
      <TH>Price </TH>
   <TR>
      <TD><IMG  SRC="images/modem.gif"></TD>
      <TD>Banksia Modem </TD>
      <TD>$399.00 </TD>
   <TR>
      <TD><IMG  SRC="images/printer.gif"></TD>
      <TD>Canon Printer </TD>
      <TD>$500.00 </TD>
   <CAPTION ALIGN=Top>ABC Company Products For Sale </CAPTION>
   </TABLE>
</BODY>
```

Once completed, you can expand on your table format and contents. The Following tag parameters are expandable by double clicking on the following tags. Parameter dialog boxes are displayed to outlines the following settings.

<TABLE>	Whole table properties. Edit alignment, column sizes and color.
<TH>	Table Column/row headings. Edit ALIGN (Horizontal Alignment) and VALIGN (Vertical Alignment) and color.
<TR>	Table Row. Edit Alignment and color.
<TD>	Table data. Edit Alignment, Cell size and width, color and wrapping.

The following example of the HTML table element is an extension to the above example:

```
<BODY>
   <TABLE ALIGN=center WIDTH="100%" BORDER BGCOLOR=#8000FF>
      <TH ALIGN=left BGCOLOR=#00FF00>IMAGE </TH>
```

```
      <TH ALIGN=left BGCOLOR=#00FF00>DESCRIPTION </TH>
      <TH ALIGN=right BGCOLOR=#00FF00>PRICE </TH>
   <TR>
      <TD><IMG  SRC="images/modem.gif"></TD>
      <TD>BANKSIA MODEM </TD>
      <TD ALIGN=right VALIGN=middle>$399.00 </TD>
   <TR>
      <TD><IMG  SRC="images/printer.gif"></TD>
      <TD>CANON PRINTER </TD>
      <TD ALIGN=right VALIGN=middle>$500.00 </TD>
   <CAPTION ALIGN=TOP><B>ABC COMPANY PRODUCTS FOR SALE </B></CAPTION>
   </TABLE>
</BODY>
```

For Further references, see also:

<TABLE> ... </TABLE> Table
<TR ...> ... </TR> Table Row
<TD ...> ... </TD> Table Data
<TH ...> ... </TH> Table Header
<CAPTION> ... </CAPTION> Table Caption
Creating Tables Creating Tables

List Wizard

FlexED contains wizards, accessed from the main toolbar to easily insert  Unordered or  ordered lists into your HTML documents. The dialog box presented by using the list button, allows you to input the list items in a text area. This is then placed into your HTML code. Also note, you may change the properties of your lists by double clicking on the List tags <LI> located in your document source.

An Example of an Unordered list is:

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

Would Be Rendered As:

- Item Number 1
- Item Number 2
- Item Number 3

NOTE: To Convert existing text from the Clipboard or other documents to a list, Paste the text as line delimited text into the List Wizard Dialog Editor Box.

For further examples of these lists:

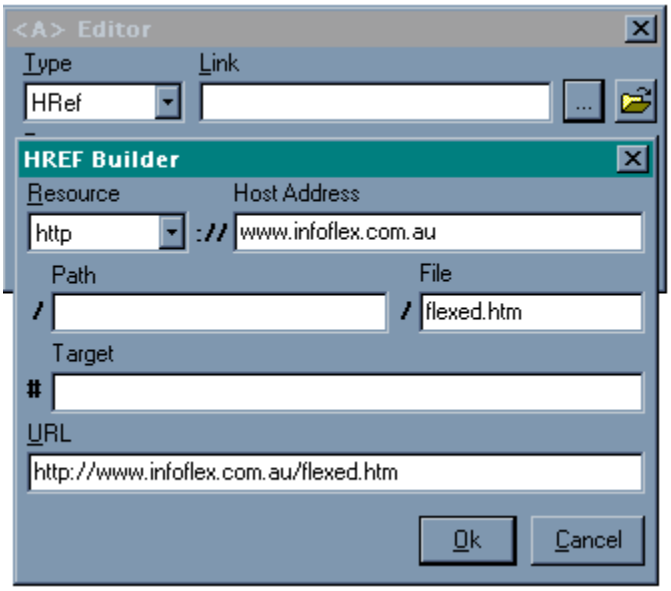
Creating Lists Creating Lists
<UL> ... </UL> Unordered Lists
<OL> ... </OL> Ordered Lists
<DIR> ... </DIR> Directory Lists
<DL> ... </DL> Definition Lists
<MENU> ... </MENU> Menu Lists

FlexED Setup

How to use the properties dialog box - Will put info in here

HRef Builder

When creating anchors (Hypertext links or Hot Spots) in HTML using FlexED, you are provided with a powerful dialog box called the Href builder that prompts you with all the required information you need to input to create your link.



As you remember from Using Anchors and Links , the standard format of an anchor is:

Servicename://internethost:portnumber/resource

As you can see from the above illustration of the Href builder, the service name or the resource is the type of protocol service, ie: http, ftp, mailto etc. The Internet host address is the web servers location. Path and filename is the location of the file, document or resource. While you are building this information, the URL is being built to form your link.

Different Service Names initiate different input screens. For example, if selecting the Mail To service name, you will be prompted for the email address to link to. Select OK when you have finished building your URL, and the HTML text will be inserted into your HTML document. When you are returned to the editor Window, Type in the Text you wish to be displayed as a hotspot when the document is rendered.

For further references, see also:

Using Anchors and Links
Creating Documents with URL's
Creating Anchors With FlexED

HTML Command Reference

TAG	MEANING
Required Elements:	
<HTML> ... </HTML>	Declares document is HTML
<HEAD> ... </HEAD>	Delineates head of document
<TITLE> ... </TITLE>	Title of document
<BODY> ... </BODY>	Delineates body of document
Headlines, Typeface Style, and Paragraph Tags	
<H1> ... </H1>	Six levels of headlines (H1 - H6)
<STRONG> ... </STRONG>	Bold
<EM> ... </EM>	Italic/Emphasis
<CODE> ... </CODE>	Fixed typeface
<KBD> ... </KBD>	Keyboard input (Fixed)
<CITE> ... </CITE>	Citation (Indented)
<P> ... </P>	Begin paragraph
<BR ...>	Line break
Uniform Resource Locators ***servicename://internethost:portnumber/resource***	
http	Hypertext Transfer Protocol
ftp	File Transfer Protocol
gopher	Gopher Protocol
telnet	Telnet Protocol
news	UseNet News Protocol
wais	Wide Area Information Server
mailto	E-Mail
Anchors and Links	
<A HREF=target>Tex	Hyperlink
<A NAME=spot>	Marker Tag
<A HREF=target#spot>…	Hyperlink to spot in document
<A HREF=URL>Tex	Hyperlink to URL
Lists	
<UL> ... </UL>	Unordered (Bulleted) List
<OL> ... </OL>	Order (Numbered) List
<DIR> ... </DIR>	Directory List
<MENU> ... </MENU>	Menu List
<LI>	Individual List Items
<DL> ... </DL>	Dictionary List
<DT>	Term in Dictionary List
<DD> ... </DD>	Dictionary List Definition

Preformatted Text and Tables

<PRE> ... </PRE>	Preformatted Text
<TABLE> ... </TABLE>	Declares Table
<TD ...> ... </TD>	Table Data (Cell)
<TH ...> ... </TH>	Table (Column or Row) Heading
<TR ...> ... </TR>	Table Row End
<TH/D ALIGN=LEFT/RIGHT>	Alignment of Heading/Data
<TH/D COLSPAN=N>	Column Span of Heading/Data
<TH/D ROWSPAN=N>	Row Span of Heading/Data
<CAPTION> ... </CAPTION>	Table Caption

Image Inclusion

<A HREF=URL> <IMG...> </A>	Image is Hyperlink
<IMG...>	Include Image file name

Fill-in Forms

<ISINDEX...>	Form Created Automatically
<FORM> ... </FORM>	Declares Form
<FORM ACTION=URL>	Back-end Program for Form Data
<INPUT NAME=label>	Name of Input Data Item
<INPUT TYPE=type>	Type (TEXT, CHECK BOX, RADIO)
<SELECT ...> ... </SELECT>	Choose One or More of Several
<OPTION>	Select One of Several
<TEXTAREA> ... </TEXTAREA>	Multi-Row/Column Text Entry

Miscellaneous HTML

<HR>	Horizontal Ruler Line
&numeric;	Numeric Special Character Code
&entity;	Entity Special Character Code
%hex-code	Special Character in URL
comment line	Comment Line

Netscape

<BLINK>	Blinks text

MSIE

<BGSOUND...>	Plays background sound
<MARQUEE...>	Allows the creation of scrolling text

<MARQUEE...>

NOTE : This element is currently only supported by the Microsoft Internet Explorer.

The new <MARQUEE> element allows the author to create a scrolling text marquee (as the name suggests, a scrolling text region much like the Windows Marquee screen saver).

Marquees can be left- or right-aligned, like images and have a variety of attributes to control them.

ALIGN This attribute can be set to either TOP, MIDDLE or BOTTOM and specifies that the text around the marquee should align with the top, middle , or bottom of the marquee.

BEHAVIOR This can be set to SCROLL, SLIDE or ALTERNATE. It specifies how the text should behave. SCROLL (the default) means start completely off one side, scroll all the way across and completely off, then start again. SLIDE means start completely off on side, scroll in and stop as soon as the text touches the other margin. ALTERNATE means bounce back and forth within the marquee.

BGCOLOR This specifies a background colour for the marquee, either as a rrggbb hex triplet, or as one of the Internet Explorer pre-named colours. (See <FONT COLOR> for more information)

DIRECTION This specifies in which direction the text should scroll. The default is LEFT, which means scrolling to the left from the right. This attribute can also be set to RIGHT, which would cause the marquee to scroll from the left to the right.

HEIGHT This specifies the height of the marquee, either in pixels (HEIGHT=n) or as a percentage of the screen height (HEIGHT=n%).

WIDTH This specifies the width of the marquee, either in pixels (WIDTH=n) or as a percentage of the screen height (WIDTH=n%).

HSPACE This specifies left and right margins for the outside of the marquee, in pixels.

LOOP specifies how many times a marquee will loop when activated. If n=-1, or LOOP=INFINITE is specifies, the marquee will loop indefinitely.

SCROLLAMOUNT Specifies the number of pixels between each successive draw of the marquee text. That is, the amount for the text to move between each draw.

SCROLLDELAY specifies the number of milliseconds between each successive draw of the marquee text. That is, it controls the speed at which text draw takes place.

VSPACE specifies the top and bottom margins for the outside of the marquee, in pixels.

Examples

```
<MARQUEE>This text will scroll from left to right slowly</MARQUEE>

<MARQUEE ALIGN=TOP>The following words, "Hi there!", will be aligned with the
top of this marquee.</MARQUEE> Hi there!

<MARQUEE BEHAVIOR=SLIDE>This marquee will scroll in and "stick."</MARQUEE>

<MARQUEE HEIGHT=50% WIDTH=80%>This marquee, is half the height of the screen
and 80% of the screen width.</MARQUEE>

<MARQUEE SCROLLDELAY=5 SCROLLAMOUNT=50>This is a very fast marquee.</MARQUEE>
```

How To Order FlexED

How Much Does FlexED Cost ?
The current cost of FlexED is **AUD $ 39.95** or **US $ 32.00**

Registrations are emailed free of postage and handling charges.
If a Floppy Disk with FlexED is required, there is a postage and handling fee of **AUD $10.00** for Australian orders or **US $10.00** for orders outside of Australia.

How Do I Order FlexED ?
Ordering FlexED can be made by the following methods:

On-line Ordering system located at **http://nt.infoflex.com.au/flexed/flexed.htm.**
or can be accessed from **http://www.infoflex.com.au**
This method allows payment via Visa or Mastercard and you order will be processed within 24 Hours.

Alternatively, you can print the FlexED Order Form located in this help, and Fax it to 612 9894 2067 for Credit Card Orders, or for money orders or cheques, snail mail it to:

Infoflex Pty Ltd
Suite 27, 15 Terminus St
Castle Hill NSW 2154
Australia

For other queries relating to FlexED, see Frequently Asked Questions located on the Infoflex Web Site located at**: http://nt.infoflex.com.au/flexed/feedback.htm**

FlexED Properties

FlexED allows detailed setup or properties of your HTML editing and rendering environments. The properties setup presents to you, the look and feel of the user interface and various options that may be turned on or off. The properties dialog box is accessed from the **file|properties** menu, located from the main menu. The following are the properties and options available:

Editor:

- Standard Text. This changes the typeface, style, color and size of all the text to appear in your HTML document in the Editing window while editing.
- Syntax Color. Sets the color of the Tags or Syntax in your HTML document. This should be a different color to the standard text, so the Syntax is easily distinguishable from the text. Note, that you can double click on the tags to display the parameters dialog box for that tag.
- Comment Color. Set your documentation or comments in your HTML document that will not be rendered as a separate color. This makes your comments easy to read while editing.
- Save Style. Select the option if you wish your document to be saved as a UNIX or MAC style document.
- Syntax Highlight. Select this option if you wish your syntax to use the color scheme selected.
- Word Wrap. Select this option if you wish to invoke word wrap in your document.
- Auto Indent. Select this option if you require your syntax and text to auto indent in the editor window. This will make your HTML code easier to read.
- Default Starting Directory. Input the directory where you would like FlexED to start when opening a document. This directory should be set to where your HTML documents are.

Internal Browser:

- Standard Text. Click on the standard text to set your typeface, style, color and size. This is the style of text that will be rendered in the internal viewer when the document is rendered.
- Link. Click on Link to change the color of your Hypertext links that will be rendered in the internal viewer.
- Pre-Format Font. Select on the Pre-format Font to change the typeface, style, color and size of pre-format text when rendered.
- **Note:** The settings of your internal browser are not relevant to any other external browser.

External Browser:

- Select your external browsers here. Selecting DDE will use DDE to communicate with Netscape browser only. When DDE is selected only one copy of Netscape is required. FlexED uses DDE to communicate with Netscape to display your HTML files.
- External Browser 1. Set the location of your first preference browser. This should be defaulted to Netscape. Make sure, that if you use netscape, that you select the use DDE option. If this is not selected, whenever you render your document to Netscape, it will keep opening another netscape window.
- External Browser 2. Set the location of your second preference browser. The default is Microsoft Internet Explorer. You should not select the DDE option here.

Options:

- Insert Raw Character. Select this if you require raw characters in your document.
- Use Temp File for Preview. Select this if you require a temp file to be created when rendering your document.
- Create Backup file. Select this if you require to make a backup of your file when saving.
- Lowercase Tags. Select this if you require lowercase syntax in your HTML document. Otherwise the syntax will default to Uppercase.

- Document - Default Extension: List the default extensions to open and save a document.
- Document - Filter: Filter the type of extensions you wish to display when opening a HTML document.
- Image selector: Default Extension: default selected extension.
- Image selector: Filter: Filter types of image formats (JPG, BMP, GIF etc).

FlexED HotKeys

Menu Hot Keys:

Alt + F	File
Alt + E	Edit
Alt + T	Text
Alt + B	Browser
Alt + W	Windows
Alt + H	Help

Editing Hot Keys:

Ctrl + Z	Undo
Shift + Ctrl + Z	Redo
Ctrl + X	Cut
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + Del	Delete
Ctrl + B	Bold
Ctrl + I	Italics
Ctrl + U	Underline
Ctrl + A	Select All

Function Hot Keys:

Ctrl + S	Save
Ctrl + N	New
Ctrl + O	Open
Ctrl + P	Print
Ctrl + F	Find
F3	Find Next
F1	Help
Ctrl + R	Refresh Internal Browser
F5	Refresh External Browser