TABLE OF CONTENTS

CONTACTING INFOFLEX

If you would like to purchase FlexED, upgrade to a newer version of FlexED or simply obtain an evaluation copy of FlexED, please e-mail:

sales@infoflex.com.au

Or Fax: + 61 - 2 - 894 2067

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

to order FlexED fill in register.txt found in the distribution set of FlexED.

Introduction to 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 FlexED's 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 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. HTML tags are color coded so they can be easily distinguished from your own text.

FlexED Features

- Syntax Highlighting. - HTML tags are colour coded, this enables you to see the tags separate to you text.

- Built in HTMLviewer - With the built in HTML viewer you do not have to own a web browser to display your HTML documents. You can view your work as you create it.

- Tag Editor - most editors let you insert tags from menu and dialog boxes however when you need to make changes, you have to remember the parameters. Double clicking on a tag withing the html editor will pop up the dialog editor and allow you to enter in or change the parameters.

- Supports both Netscape Extensions to HTML, and proposed HTML 3 elements.

- Windows 95-style interface in Windows 3.1 and Windows NT.

- Dialogs let you perform complex tasks like creating forms and tables in a few seconds.

- Save time and money in creating home pages using FlexED.

HTML Tutorials

The following lessons are available:

[Creating an HTML Document](#)

[Adding HypertextLinks](#)

[Adding Pictures](#)

[Adding Tables](#)

TUTORIAL - Creating your first home page

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.

Select file from the menu and then new to start a new document. This sets up the basis of your first home page.

From here we will insert a heading for your page. Select the **H1** icon in your toolbar. This is the largest form of a heading available and decreases in size to H6. After you have done this, the cursor will be positioned in your HTML code to accept your heading. Type in your company name or similar here. Then press Ctrl-R or your icon and your viewer will display your heading. You are now well on your way.

Secondly, we will insert a smaller heading. Position the cursor in the HTML code after our heading text and after the </H1> code. Then select **H↓** from the toolbar and select H3 from the menu item and repeat the step above.

Next we will place a line after the headings. Position the cursor after the </H3> code and then select the horizontal line from the tool bar. Press re-draw and look at you results. Next we will add the body of your page and create two paragraphs in doing so. Position the cursor after the <HR> Code and select the paragraph icon **¶** from your toolbar. Start typing away until content. Press re-draw to view your work. To Create our next paragraph, position the cursor after the </P> code, select   the paragraph icon **¶** from the toolbar and off you type again. You can create as many paragraphs as you wish.

Next we can format your page with some colors, italics and underlining etc. First we can add some color. Double click on the <BODY> in your HTML code and a dialog box will appear outlining your options to format the page.You can select a backgrond image by pressing the button next to this option. You can then search for gif or jpg images to have in the background of your page. Next you can change your text color and background color (bgcolor) by pressing their buttons and a color pallet will be displayed from which you can select your colors. Press OK when finished selecting your colors and press your re-draw button to view your page.
Next we can attempt to format your text a little. First we can change some text into italics. Highlight some text in your HTML code that you have type (say a few words) with your mouse. Press the **I** (Italic) button on your toolbar and the selected text will change. Other formatting you can use for formatting your home page in the same way include underlining, changing font, bold, emphasising and so on. Once again, all as it takes to format your text is to highlight the selected text and press the particluar button on the toolbar for which you want to format the text with. FlexED will do the rest.

Further Tutorials

Adding HypertextLinks

Adding Pictures

[Adding Tables](#)

TUTORIAL - Adding Hypertext Links

Hypertext links are links to other functions or documents from your source home page. These can be links to other documents or pages, ftp, mail to, news and so on. These HyperText Links may also be known as Hot Spots. Firstly, decide on the point where you want to place the hypertext link on your document. Select the point in your HTML document code to place the Hypertext link. Then select the Insert (Anchor) button in your tool bar. This will display an editor dialog box. You then have a choice of a Href or Name Link. A Href is a hypertext link to another document or function while a name is a marker inside your HTML document. If for instance you are at the top of your HTML document and select a Name Link (say call it TopofDocument) and press OK, You have created a marker at the top of your document. Then if you create a Href link at the bottom of your HTML document and called the Href Link #TopofDocument, when you click on the hot spot (Top of Document) on your home page if would advance you to the marker placed on your document. Links to other home pages you have created can be made by creating a Href Link and calling the link as your other files filename. The reverse can be put into your other page to return you to your document. Other more advance links can be made with Href and editing the parameters as required in The Href Builder. In here you can find the various links you can make and fill in the required parameters. If for instance, you were making a reference to another home page, say at www.infoflex.com.au/index.html you would first Click the anchor button, select Href, select the link button next to the link input box, select http from the resource list,   type in www.infoflex.com.au as the host address and enter the filename as index.html. This will create a link to the Home page of Infoflex from your own Home page. The cursor will then place you in the HTML document code so you may give the link a name.

Further Tutorials

Creating an HTML Document

Adding Pictures

Adding Tables

TUTORIAL - Adding Pictures

To Add Pictures or graphics into you home page, the following steps are required to load an image. Decide at which point you want the image to appear on your home page and place

the cursor in the HTML document code at the requested point. Select the insert (image) button on the toolbar. This will prompt you with an image selector dialog box. First select the image you require in the form of a gif or jpg files. Press the search button next to the image selection input box to search for your image. Once you have selected the image, type in some alternate text to appear, for those which do not wish to view images in their World Wide Web browses. For them, instead of the image displaying, the text you typed in the alternate test will be displayed. Then select the alignment you wish to set for your image. Press OK and your image has been coded into the HTML code. The cursor is ready in the HTML code if you wish to type some text which resides with the image. Your image parameters may be modified by mouse clicking on the <IMG> tag with the HTML document code.

Further Tutorials

Creating an HTML Document

Adding HypertextLinks

Adding Tables

TUTORIAL Adding Tables

Tables can be used to display rows and columns of information in an organised formation. To insert a table into your home page, decide on the location on your home page where the table is to appear and place your cursor in the relevant position in your HTML document code. Select the (table) button in you toolbar. Then select wizard on the pop-up list. This is the quickest way to add a table. A Table editor dialog box will appear and ask you for a caption. This is your table name to appear on the home page. Type your name in here and then align or place table name using the align functions available. Then select weather your headings will be in the columns, rows, both or none. Then enter the rows and columns you require in you table. Once you have completed this, the table editor is ready for text entry. Click your mouse in the appropriate boxes and enter your desired text. Once you have completed this you may adjust the cell spacing and padding as you require. Press OK when you have completed this and your table will be built into your HTML document. If you wish to edit the table parameters, mouse click on the <Table> tag to edit or adjust the parameters.

Further Tutorials

Creating an HTML Document

Adding HypertextLinks

Adding Pictures

HTML CODES INDEX

A, ADDRESS, APPLET, AREA,
B, BASE, BASEFONT, BGSOUND, BIG, BLINK, BODY, BR,
CAPTION, CENTER, CITE, CODE,
DD, DIR, DL, DT,
EM, EMBED,
FONT, FORM, FRAME, FRAMESET,
H1, H2, H3, H4, H5, H6, HEAD, HR, HTML,
I, IMG, INPUT,ISINDEX,
KBD,
LI, LINK,
MAP, MENU, META,
NOBR, NOEMBED, NOFRAMES,
OL, OPTION,
P, PARAM, PRE,
SAMP, SELECT, SMALL, STRONG, SUB, SUP,
TABLE, TD, TEXTAREA, TH, TITLE, TR, TT,
U, UL,
VAR,
WBR

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

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 :

```
See <A HREF="http://www.hal.com/">HaL</A>'s information for more details.
```

In this example, selecting "HaL" takes the reader to a document located at http://www.hal.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 :

```
The <A HREF="document.html#glossary">glossary</A> defines terms used in the
   document.
```

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.   Within an HTML document, normally a connection to an anonymous ftp site would be made.   The final beta of version 2 of NCSA Mosaic, allows connection to private ftp sites.   The Anchor would take the form ftp://jdoe@your.com.   However, Mosaic would then prompt the user for a password, so linking from documents to your private ftp site is possibly not a good idea unless youre willing to distribute passwords for the site.

`<A HREF="gopher://...>`   Makes a link to a gopher server.

`<A HREF="mailto:...">`   Activating such a link would bring up the browsers 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.   Final beta of NCSA Mosaic supports the **TITLE** sub-attribute to this attribute.   It allows the author to specify the subject of the mail message that will be sent.   **Netscape** allows specification of the subject line by using the following syntax:

e.g. `<A HREF="mailto:cmlehunt@swan.ac.uk?subject=The HTMLib is fantastic">`link text`</A>`

`<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.

NAME

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 exmple 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>
```

TITLE

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.

REL

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.

REV

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.

URN

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.

METHODS

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.

TARGET

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>
```

NOTE : The use of targeted browser windows is **Netscape** specific.   Also, if the targetted document is part of a frameset, there are various reserved names for target window available for smooth window transition.

See Also, <BASE TARGET= …>

\<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) 894 1313
</ADDRESS>
```

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

Below is a more complex example of an `APPLET` element:

```
<APPLET CODEBASE="http://java.sun.com/JDK-prebeta1/applets/NervousText" CODE="NervousText.class"
  width=400 height=75 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://java.sun.com/JDK-prebeta1/applets/NervousText/NervousText.class, to set the initial size of the
applet to 400x75 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>

see also MAP

The AREA element specifies a single area of an image which, if selected, will link to the hyperlink 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 the text should be rendered in boldface, where available. Otherwise, alternative mapping is allowed.

```
The instructions <B>must be read </B> before continuing.
```

would be rendered as :

The instructions **must be read** before continuing.

<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.

  e.g. `<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 <u>`<FONT SIZE ...>`</u> 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 <u>`<FONT FACE= ...>`</u>

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)

SRC

LOOP

\<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 normal text, with <BIG>this bit </BIG>being big text.
```

would be rendered as:

This is normal text, with **this bit** being big text.

\<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.

```
<BLINK>This text would blink on the page</BLINK>
```

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/formatting of the page.   The format is:

```
<BODY>
The document 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.

Recent versions of the proposed HTML 3.0 spec. have added a **BACKGROUND** attribute 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. Thus specifying:

```
<BODY BACKGROUND="URL or path/filename.gif">
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
This attribute to `BODY` is not currently in the proposed HTML 3.0 specification, but is supported by Netscape, the Internet Explorer, NCSA Mosaic and many other browsers and is being considered for inclusion in HTML 3.0.   This element changes the colour of the background without having to specify a separate image that requires another network access to load.   The format that is understood is:

```
<BODY BGCOLOR="#rrggbb">
Document here
</BODY>
```

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

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

TEXT
This 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="#rrggbb">
Document here
</BODY>
```

LINK, **VLINK**, and **ALINK** attributes
These attributes let you control the colouring of 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">
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:

```
<P>
Pease porridge hot <BR>
Pease porridge cold <BR>
Pease porridge in the pot <BR>
Nine days old.
```

<CAPTION> ... </CAPTION>

see also <u><table></u>

This represents the caption for a table. `<CAPTION>` elements should appear inside the <u>`<TABLE>`</u> 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 **top** or **bottom**, defaulting to top. 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.

<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 <u>`<P ALIGN=CENTER>`</u> 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 sentence, containing a <CITE>citation reference</CITE>
```

would look like:

This sentence, containing a *citation reference*

<CODE> ... </CODE>

The Code element indicates an example of code; typically rendered as monospaced . Do not confuse with the [Preformatted Text](#) element.

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

It would look like :

This sentence contains an `example of code`

<DD> ... </DD>

Description

The DD element identifies the separated multi-line definition item in a <u>DL</u> definition list. In a <u>DL</u> list a DD should always be preceded by at least one <u>DT</u> 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 <u>`<LI>`</u> (list item) element:

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

<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>
```

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:

```
<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>
```

\<DT\>

Description

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

<EM> ... </EM>

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

```
The <EM>Emphasis</EM> element typically renders as Italics.
```

would render :

The *Emphasis* element typically render as Italics.

<EMBED …>

see also

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)

SRC

HEIGHT

WIDTH

<FONT ...>

Netscape 1.0 (and above) and Microsoft's Internet Explorer support different sized fonts within HTML documents.

The new element 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 <u>`<BASEFONT SIZE ...>`</u> element.

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

COLOR = `#rrggbb`

The colour attribute sets the colour which text will appear in on the screen. Rrggbb is a hexadecimal colour denoting a RGB colour value.

Example:

```
<FONT COLOR=#ff0000>This text is red.</FONT>
```

FACE=`name [,name] [,name]`

The `FACE` attribute 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 text will be displayed in either Arial,
  Lucida Sans, or Times Roman, depending on which fonts you have installed on your
  system.
</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-urlencoded.

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>

see also [Frame Wizard](#)

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 ...>

see also [Frame Wizard](#)

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> Headings

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.

Included in the proposed HTML level 3.0 specification is the ability to align Headings.
Basically, **ALIGN**=left|center|right attributes have been added to the `<H1>` through to `<H6>` elements. e.g :

```
<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> Introduction to HTML </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:

```
<HR>
<ADDRESS>February 8, 1995, CERN</ADDRESS>
</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>

The I element changes the physical rendering of the contents of the element to an *italics* (or slanted) font.

<IMG...> In-line images

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:

ALIGN

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:

e.g. : `<IMG SRC="triangle.gif" ALT="Warning:"> Be sure to read these instructions.`

ISMAP

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:

e.g. : `<A HREF="http://machine/htbin/imagemap/sample">`
        `<IMG SRC="sample.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 behaviour. 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

SRC

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.

Example of use:

e.g. : `<IMG SRC ="triangle.gif">Be sure to read these instructions.`

Netscape specific extensions to the <IMG> Element

    <IMG **ALIGN**= left|right|top|texttop|middle|
            absmiddle|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.

LOWSRC
Using the `LOWSRC` attribute, it is possible to use two images in the same space.   The syntax is :

 e.g. : `<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:

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. Unselected 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:

e.g. : `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   **Netscape** now supports a FILE option to 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

NOTE : The `PROMPT` attribute is only supported by **Netscape**.

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.

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

would render as :

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

<LI>

see also [Dir](#) [Menu](#) [OL](#) [UL](#)

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 Anchor 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 <u>\<IMG\></u> 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/tech/pic1.gif" USEMAP="maps.html#map1">
```

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/image map/pic2"><IMG SRC="../images/tech/pic2.gif"
  USEMAP="maps.html#map2" 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="no_csim.html"><IMG SRC="../images/tech/pic3.gif"
  USEMAP="maps.html#map3"></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 <u>\<IMG\></u> 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 co-ordinates 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 co-ordinates 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="buttonbar">
<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/tech/bar.gif" USEMAP="#buttonbar">
```

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.

Examples :

If the document contains :

```
<META HTTP-EQUIV="Expires" CONTENT="Tue, 04 Dec 1993 21:29:02 GMT">
<META HTTP-EQUIV="Keywords" CONTENT="Fred, Barney">
<META HTTP-EQUIV="Reply-to" CONTENT="fielding@ics.uci.edu <Roy Fielding">
```

Then the HTTP response header would be :

```
Expires: Tue, 04 Dec 1993 21:29:02 GMT
Keywords: Fred, Barney
Reply-to: fielding@ics.uci.edu (Roy Fielding)
```

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

```
<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 adjust 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:

```
<OL>
<LI>Click the Web button to open the Open the URL window.
<LI>Enter the URL number in the text field of the Open URL window. The Web
 document you specified is displayed.
<LI>Click highlighted text to move from one link to another.
</OL>
```

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.

NOTE : The TYPE attribute used in the <OL> Element and the <LI> Element and the START attribute in the <OL> Element are supported only by Netscape and the Internet Explorer.

<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 indicates a 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 Preformatted Text element. With some HTML user agents, the first line in a paragraph is indented.

Example of use:

```
<H1>This Heading Precedes the Paragraph</H1>
<P>This is the text of the first paragraph.
<P>This is the text of the second paragraph. Although you do not need to start
  paragraphs on new lines, maintaining this convention facilitates document
  maintenance.
<P>This is the text of a third paragraph.
```

Included in the proposed HTML level 3.0 specification is the ability to align paragraphs Basically, **ALIGN**=left|center|right attributes have been added to the `<P>` element. e.g :

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

All text with in the paragraph will be aligned to the left side of the page layout. This setting is equal to the default `<P>` element.

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

All text within the paragraph will be aligned to the centre of the page.

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

All text will be aligned to the right side of the page.

\<PRE\> ... \</PRE\>

The Preformatted 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 preformatted 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 renderer is forbidden from using a constant left indent for rendering preformatted 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 Preformatted 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.

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

would render as :

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

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

The Select element allows the user to chose 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="flavor">
<OPTION>Vanilla
<OPTION>Strawberry
<OPTION>Rum and Raisin
<OPTION>Peach and Orange
</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 normal text, with <SMALL>this bit</SMALL> being small text.
```

would be rendered as:

This is normal text, with this bit being small text.

<STRONG> ... </STRONG>

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

```
The instructions <STRONG>must be read</STRONG> before continuing.
```

would be rendered as :

The instructions **must be read** before continuing.

<SUB> ... </SUB>

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 the main text, with <SUB>this bit</SUB> being subscript.
```

<SUP> ... </SUP>

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 the main text, with <SUP>this bit</SUP> being superscript.
```

<TABLE> ... </TABLE>

This is the main wrapper for all the other table elements, and other table elements will be ignored if they aren't wrapped 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=<value>

This is a new attribute for the `<TABLE>` element.   By default Netscape uses a cell spacing of two. For those fussy about the look of their tables, this gives them a little more control.   Like it sounds, cell spacing is the amount of space inserted between individual cells in a table.

CELLPADDING=<value>

This is a new attribute for the `<TABLE>` element.   By default Netscape uses a cell padding of one. Cell padding 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.

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

gives the most compact table possible.

WIDTH=<value_or_percent>

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.

HEIGHT=<value_or_percent>

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.

ALIGN

Microsofts **Internet Explorer** (version 2.0) and **Netscape** now support the ALIGN attribute to the <TABLE> element.   Like that used for floating images, it allows a table to be aligned to the **left** or **right** of the page, 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

It specifies that the text can be **top**- or **bottom**-aligned. The default is centre-aligned.

Table - <TD ...> ... </TD>

This stands for table data, 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 this 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=<value_or_percent>

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 <u>`<TH>`</u> and <u>`<TABLE>`</u> elements.

NOWRAP

If this attribute appears in any table cell (<u>`<TH>`</u> 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 (<u>`<TH>`</u> 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 (<u>`<TH>`</u> 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 <u>`<TH>`</u> will be truncated.

\<TEXTAREA\> ... \</TEXTAREA\>

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

e.g. :
```
<TEXTAREA NAME="address" ROWS=64 COLS=6>
HaL Computer Systems
1315 Dell Avenue
Campbell, California 95008
</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.

Recent versions of the Netscape Navigator (from version 2.0) have introduced the **WRAP** attribute in the `TEXTAREA` element: 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.**

<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=<value_or_percent>
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>
```

Table - <TR ...> ... </TR>

This stands for table row.   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.

```
Text between the <TT> typetype elements</TT> should be rendered in fixed width
  typewriter font.
```

would render as :

Text between the `typetype elements` should be rendered in fixed width typewriter font.

<U> ... </U>

The `<U> ... </U>` Elements state that the enclosed text should be rendered, if practical, underlined.   This is an HTML 3.0 element and may not be widely supported.

```
The <U>main point</U> of the exercise...
```

would be rendered as :

The <u>main point</u> of the exercise...

<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:

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

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 thus :

```
TYPE=disc
TYPE=circle
TYPE=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.

NOTE : The `TYPE` attribute when used in the `<UL>` and `<LI>` Elements is Netscape specific.

<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 (`<BR>` does that) it simply lets the Netscape Navigator know where a line break is allowed to be inserted if needed.

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 hardwork 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.

Note - *you can paste into the list editor. For e.g. you might have a paragraph full of text you would like to convert to a list. Select the text you want to convert into a list. Select ordered/unordered list and paste into the list editor, make your adjustments and press ok.*

Insert Table - Displays pull down menu of the following options

Table   <TABLE>
Header <TH>
Data   <TD>
Row   <TR>
Caption <CAPTION>
Wizard…

Insert Anchor - Displays <A…> … </A> anchor editor wich 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. Used when wanting to separate information with a line.

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 - reneders and displays the active HTML document window.

Refresh Alternative Browser - Passes the contents of the active HTML document window to your selected external browser. Currently FlexED only supports Netscape DDE support. 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> ... </P> tag.

Preformatted Paragraph - Insert <PRE> ... </PRE> tag.

Centered Text - Insert <CENTER> ... </CENTER> tag.

Heading 1 - Insert <H1> ... </H1> tag.

Heading 2 - Insert <H2> ... </H2> tag.

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

Bold - Insert <B> ... </B> tag.

H↓

Italic - Insert <u><I> … </I></u> tag.

U

Underline - Insert <u><U> … </U></u> tag.

S

Strong - Insert <u><STRONG> … </STRONG></u> tag.

Em

Emphasized - Insert <u><EM> … </EM></u> tag.

Cit.

Citation - Insert <u><CITE> … </CITE></u> tag.

TT

TT - Insert <u><TT> … </TT></u> tag.

▣

Address - Insety <u><ADDRESS> … </ADDRESS></u> tag.

MENU BAR

FlexED menu items defined

File

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

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

    Open - Access previously created HTML document.

    Properties…

    Exit - Leave FlexED

Edit

    Undo - Undo the most recent change. You can undo to the amount of available memory

    Redo - Redo the most recent undo. You can redo to the amount of information kept in buffer.

    Cut - Copy highlighted section from the active HTML document to the clipboard

    Copy - Copy highlighted section from the active HTML document to the clipboard

    Paste - Paste contents of clipboard to the active HTML document window.

    Delete - Delete highlighted section.

    Find - Search for a given phrase.

    Replace - Search and replace found phrase with new phrase.

    Select All - highlight the contents of the active HTML document.

HTML

    See HTML

    Tags… - Display Popup tag reference. Displays, currently supported HTML tags. Allows the user to select HTML tags from dialog box.

Window

    Tile Vertical - Tile visible document windows vertically
    Tile Horizontal - Tile visible document windows horizontally
    Cascade - Cascaed visible document windows
    Arrange Icons - Arrange minimized document windows

Properties Dialog

Properties dialog allows the configuration of FlexED.

Editor Options

Click in the mini edit window to change attributes of the editor.

Tab Size

Set the tab size of edit window to nominated size.

Mapping

Select your external browser 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.

Image Map Wizard

Image map wizard allows you to create hot spot selections. Image map wizard can be activated by clicking on the ![H+] found on the toolbar and selecting the **Map Wizard** option.

Map Wizard Options

Name - name of image map. This will be used as the referece for the image map.

HRef - url to access when the user clicks on the hotspot. You can change the properties by pressing the right mouse button on the hot spot rectangle.

Using the map wizard

1. Select the image you would like to create hot spots for by selecting File|Open. Locate your file and press ok. *Note - Make sure that your images are relative to your document.*

1. Enter in the name you want to call your image map. This is the entry on the right hand side called name. Name options should not contain any blank spaces, also make sure it is a unique name within the document.

1. Draw hot spot on the locations of the image by locating the cursor at the top left corner of the section you want to highligh and press the right mouse button, then drag to the bottom   right corner of the hot spot. You will see a box drawn arround the hot spot. Press the right mouse button in the rectange and key in the url you want associated with the hot spot.

1. Repeat steps 3 for the number of hot spots required.

1. Press OK - your done. Image map wizard should have created something like the following. *Providing that you named your map name test and you selected duke.gif as your image.*

<MAP NAME="test">
   <AREA COORDS="25,18,42,33" HREF="http://www.infoflex.com.au/flexed.htm">
</MAP>
<IMG BORDER=0 USEMAP="#test" SRC="images/duke.gif">

Note - *The internal viewer does not support image maps. You can view the results of your image maps with your external browser*

Table Wizard

Table wizard allows the creation of Tables. Table wizard can be activated by clicking on the ![H↓] found on the toolbar and selecting the **wizard…** option.

Table Wizard Options

Caption - Sets the tables caption. Leaving this option blank will not include a <CAPTION> tag.

Align - Sets the alignment for the caption. Setting the align property to top will display the caption on top of the table, setting the option to bottom will display the caption at the bottom of the table.

Headings -

Column - Selecting Column will use the <TH> tag for the first column. <TH> renders the fields using the bold attribute.

Row - Selecting Row will use <TH> tag for the first row. <TH> renders the fields using the bold attribute

Table

Rows - Allows you to nominate the number of rows your table will contain.

Colls - Allows you to nominate the number of columns your table will contain.

CellSpacing - Controls the spacing between cells.

CellPadding - Controls the padding around the cells.

Border - Enable|Disable border. Border attribute uses the Cellspacing and Cellpadding to render the border.

Using the Table Wizard

1. Enter in the caption for your table. The caption is optional you can ommit this.
2. Set the allign property to the desired location.
3. Set the number of rows and columns you want for your table by addjusting the value of cols and rows properties.
4. Addjust the cellpadding and cellspacing if required.
5. Key in your data into the cells
6. You can select to highlight the first row and column by selecting **Headings|Column** and **Headings| Row**
7. Press OK.

List Wizard

List wizard allows the creation of lists. You can create two kinds of lists using the list wizard.

1. **Ordered** - list wizard is activated by clicking on the which is located on the toolbar.

1. **Unordered** - list wizard is activated by clicking on the which is located on the toolbar.

The list wizards are straigh forward to use. Activate the list wizard, key in your items and press ok.

Tips *- You can convert text to list by copying the text you want to convert to the clipboard and then paste into the list wizard.*

Frame Wizard

Frame wizard allows the creation of Frames. Frame wizard can be activated by selecting the menu option **File|New Frame...**

Frame Wizard options

Frame Type - Nominate the frame type you would like to create.   You can choose from the available styles.

Src - The SRC attribute takes as its value the URL of the document to be displayed in this particular frame.

Name - The NAME attribute is used to assign a name to a frame so it can be targeted by links in other documents

Width -   Width of the frame. You can key in percentage or pixel value. Percantage is denoted by value%. Keying in * will use the remaining amount for the frame.

Height - Height of the frame. You can key in percentage or pixel value. Percantage is denoted by value%. Keying in * will use the remaining amount for the frame.

Margin Width - Specifies the margin width for the selected frame. This parameter is optional

Margin Height - Specifies the margin height for the selected frame. This parameter is optional

Scrolling - attribute is used to describe if the frame should have a scrollbar or not.

No Resize - indicates that the frame is not resizable by the user.

Using the Frame Wizard

1. Select the frame layout suitable. Each icon represents a frame template.

1. Assign the url's to each one of the frames by selecting or keying in the HTML document name in the source field.

1. Depending on the frame layout, fill in the width and height attributes. Frame wizard will only display fields that are required to be filled.

1. Select wether you want scroll bars.

1. Select wether you want the associated frame sizeable.

1. Press OK - Congratulations you have just created a Framed document. *You should see something like the following in your editor..*

```
<FRAMESET COLS="10%,*">
    <FRAME   SRC="frame1.html" NORESIZE>
    < FRAME SRC="frame2.html">
<NOFRAMES>
This document contains frames
</NOFRAMES>
</FRAMESET>
```