EdWin Help Contents

Choose from one of the following topics:

Document Tags

HTML
Head
Title
Body
Background Graphic
Watermark
Background Color
Text Color
Link Colors
Banner
Base
Division
Is Index
Meta
Next ID
Range
Style
Basefont

HTML Tag

<HTML>…</HTML>

The HTML tag <HTML> and its associated closing tag </HTML> should encompass the entire HTML document. Note that most browsers no longer require it, but it is still supported to display older documents.

Example:

```
<HTML>
<HEAD><TITLE>My Web Page</TITLE></HEAD>
<BODY>
…
</BODY>
</HTML>
```

Head Tag

<HEAD>…</HEAD>

The head of an HTML document is an unordered collection of information about the document.

Example:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HEAD>
<TITLE>Introduction to HTML</TITLE>
</HEAD>
...
```

Title Tag

<TITLE>…</TITLE>

Every HTML document must contain a TITLE element.

The title should identify the contents of the document in a global context. A short title, such as "Introduction" may be meaningless out of context. A title such as "Introduction to HTML Elements" is more appropriate.

A user agent may display the title of a document in a history list or as a label for the window displaying the document. This differs from headings, which are typically displayed within the body text flow.

Body Tag

<BODY>…</BODY>

The BODY element contains the text flow of the document, including headings, paragraphs, lists, etc.

Example:

```
<BODY>
<h1>Important Stuff</h1>
<p>Explanation about important stuff...
</BODY>
```

Background Graphic

The graphic file denoted by the BACKGROUND= element will be loaded by browsers that support this feature and displayed as the background repeating horizontally and vertically across the entire document.

Watermark

This element will cause the background not to scroll with the rest of the document.

Background Color

The color selected and inserted after the BGColor= element will be the background color shown on the page by browsers that support this feature.

Colors are represented by 6 digit hexidecimal numbers in the format:

RRGGBB

Where RR = red component, GG = green component and BB= blue component, again in hexidecimal format.

For example:

FFFFFF = white
00FF00 = green
FF00FF = purple

Text Color

The color selected and inserted after the Text= element will be the text color used on the page by browsers that support this feature.

Colors are represented by 6 digit hexidecimal numbers in the format:

RRGGBB

Where RR = red component, GG = green component and BB= blue component, again in hexidecimal format.

For example:

FFFFFF = white
00FF00 = green
FF00FF = purple

Link Colors

The color selected and inserted after the Link=, Alink=, and Vlink= elements will be the link colors used on the page by browsers that support this feature. The link color is the color used to display new links. Alink color will be used for active links, such as when the mouse button is down, or while the browser is searching for the link. Vlink color will be used to denote links which have already been visited.

Colors are represented by 6 digit hexidecimal numbers in the format:

RRGGBB

Where RR = red component, GG = green component and BB= blue component, again in hexidecimal format.

For example:

FFFFFF = white
00FF00 = green
FF00FF = purple

Banner

<BANNER>…</BANNER>

The BANNER element is proposed for corporate logos, navigation aids, disclaimers and other information which shouldn't be scrolled with the rest of the document.

Example:

<BANNER>This text should not scroll</BANNER>

Base

The optional BASE element provides a base address for interpreting relative URLs when the document is read out of context. The value of the HREF attribute must be an absolute URI.

Division

The division element is used to denote different parts of a document such as chapter or section.

Is Index

The ISINDEX element indicates that the user agent should allow the user to search an index by giving keywords.

Meta

The META element is an extensible container for use in identifying specialized document meta-information. Meta-information has two main functions:

- To provide a means to discover that the data set exists and how it might be obtained or accessed; and
- To document the content, quality, and features of a data set, indicating its fitness for use.

Each META element specifies a name/value pair. If multiple META elements are provided with the same name, their combined contents--concatenated as a comma-separated list--is the value associated with that name.

HTTP servers may read the content of the document HEAD to generate header fields corresponding to any elements defining a value for the attribute HTTP-EQUIV.

Attributes of the META element:

HTTP-EQUIV

binds the element to an HTTP header field. An HTTP server may use this information to process the document. In particular, it may include a header field in the responses to requests for this document: the header name is taken from the HTTP-EQUIV attribute value, and the header value is taken from the value of the CONTENT attribute. HTTP header names are not case sensitive.

NAME

specifies the name of the name/value pair. If not present, HTTP-EQUIV gives the name.

CONTENT

specifies the value of the name/value pair.

Examples

If the document contains:

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

then the server may include the following header fields:

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

as part of the HTTP response to a `GET' or `HEAD' request for that document.

An HTTP server must not use the META element to form an HTTP response header unless the HTTP-EQUIV attribute is present.

An HTTP server may disregard any META elements that specify information controlled by the HTTP server, for example `Server', `Date', and `Last-modified'.

Next ID

The NEXTID element is included for historical reasons only. HTML documents should not contain NEXTID elements.

The NEXTID element gives a hint for the name to use for a new A element when editing an HTML document. It should be distinct from all NAME attribute values on A elements.

Example:

<NEXTID N=Z27>

Range

This topic is under construction. If registered, e-mail msutton@mail.vantek.net for latest version.

Style

This topic is under construction. If registered, e-mail msutton@mail.vantek.net for latest version.

Basefont

The font attributes specified within this element dictate the normal paragraph font for the entire document.

Example:

<BASEFONT SIZE=3 COLOR=0000AA>

Physical Highlighting

Bold
Italic
Underline
Strikethrough
Big
Small
Superscript
Subscript
Blink
Teletype
Headings
Large First Character

Bold

<B>…</B>

The text between any <B> tag and its corresponding </B> closing tag will be made bold while retaining the current font size and non-conflicting styles. For example, if a block of text is italicized, and you insert a bold tag over a portion of the block, the text within the bold tags will be both bold and italic.

Example:

This text is <B>bold</B>.

Displays:

This text is **bold**.

Note:
All web browsers are different, and some may not recognize this tag.

Italic

<I>...</I>

The text between any <I> tag and its corresponding </I> closing tag will be made italic while retaining the current font size and non-conflicting styles. For example, if a block of text is bold, and you insert an italic tag over a portion of the block, the text within the italic tags will be both bold and italic.

Example:

This text is <I>italic</I>.

Displays:

This text is *italic*.

Note:
All web browsers are different, and some may not recognize this tag.

Underline

<U>…</U>

The text between any <U> tag and its corresponding </U> closing tag will be underlined while retaining the current font size and non-conflicting styles. For example, if a block of text is italicized, and you insert an underline tag over a portion of the block, the text within the bold tags will be both underlined and italic.

Example:

This text is <U>underlined</U>.

Displays:

This text is underlined.

Note:
All web browsers are different, and some may not recognize this tag.

Strikethrough

<S>...</S>

The text between any <S> tag and its corresponding </S> closing tag will be stuck-out while retaining the current font size and non-conflicting styles. For example, if a block of text is italicized, and you insert an strikethrough tag over a portion of the block, the text within the strikethrough tags will be both strikethrough and italic.

Example:

This text is <S>struck-out</S>.

Displays:

This text is ~~struck-out~~.

Note:
All web browsers are different, and some may not recognize this tag.

Big

<BIG>…</BIG>

The text between any <BIG> tag and its corresponding </BIG> closing tag will be made larger than the normal paragraph font for the page.

Example:

This text is <BIG>big</BIG>.

Displays:

This text is big.

Note:
All web browsers are different, and some may not recognize this tag.

Small

<SMALL>...</SMALL>

The text between any <SMALL> tag and its corresponding </SMALL> closing tag will be made smaller than the normal paragraph font for the page.

Example:

This text is <SMALL>small</SMALL>.

Displays:

This text is small.

Note:
All web browsers are different, and some may not recognize this tag.

Superscript

[…]

The text between any ^{tag and its corresponding} closing tag will be positioned slightly higher than the normal paragraph font for the page.

Example:

The answer is x².

Displays:

The answer is x2.

Note:
All web browsers are different, and some may not recognize this tag.

Subscript

_…

The text between any _{tag and its corresponding} closing tag will be positioned slightly lower than the normal paragraph font for the page.

Example:

Find the temperature at time, t⁰.

Displays:

Find the temperature at time, t0.

Note:
All web browsers are different, and some may not recognize this tag.

Blink

<BLINK>…</BLINK>

The text between any <BLINK> tag and its corresponding </BLINK> closing tag will alternate between visible and non-visible at a regular interval.

Example:

For the best food, <BLINK>Eat At Joes<BLINK>.

Note:
All web browsers are different, and some may not recognize this tag.

Teletype

<TT>...</TT>

The text between any <TT> tag and its corresponding </TT> closing tag will be displayed using a monospace (non-proportional) font.

Example:

This text is
proportional.<P>
<TT>This text is
non-proportional</TT>.

Displays:

This text is
proportional.

This text is
non-proportional.

Note:
All web browsers are different, and some may not recognize this tag.

Headings

<Hn**>..</H**n**>**

The six heading elements, H1 through H6, denote section headings. Although the order and occurrence of headings is not constrained by the HTML DTD, documents should not skip levels (for example, from H1 to H3), as converting such documents to other representations is often problematic.

Example:

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

Large First Character

This topic is under construction. If registered, e-mail msutton@mail.vantek.net for latest version.

Logical Highlighting

Emphasized
Citation
Variable
Strong
Source Code
Sample
Keyboard
Defining Instance
Abbreviation
Acronym
Person

Emphasized

<EM>…</EM>

The text between any <EM> tag and its corresponding </EM> closing tag will be displayed using the default fontstyle for emphasized text. This fontstyle varies according to browser and platform. This is recommended over tags such as bold and italic.

Example:

I want to go home <EM>now</EM>.

Displays:

I want to go home *now*.

Note:
All web browsers are different, and some may not recognize this tag.

Citation

<CITE>…</CITE>

The text between any <CITE> tag and its corresponding </CITE> closing tag will be displayed using the default fontstyle for cited text. This fontstyle varies according to browser and platform. This is recommended over tags such as bold and italic.

Example:

This information was reprinted from <CITE>The New York Times</CITE>.

Displays:

This information was reprinted from *The New York Times*.

Note:
All web browsers are different, and some may not recognize this tag.

Variable

<VAR>…</VAR>

The text between any <VAR> tag and its corresponding </VAR> closing tag will be displayed using the default fontstyle for variables. This fontstyle varies according to browser and platform. This is recommended over tags such as bold and italic.

Example:

Multiply <VAR>x</VAR> by <VAR>y</VAR> to get the answer.

Displays:

Multiply x by y to get the answer.

Note:
All web browsers are different, and some may not recognize this tag.

Strong

<STRONG>…</STRONG>

The text between any <CITE> tag and its corresponding </CITE> closing tag will be displayed using the default fontstyle for stronger emphasized text. This fontstyle varies according to browser and platform. This is recommended over tags such as bold and italic.

Example:

I <STRONG>really</STRONG> want to go home now.

Displays:

I **really** want to go home now.

Note:
All web browsers are different, and some may not recognize this tag.

Source Code

<CODE>...</CODE>

The text between any <CODE> tag and its corresponding </CODE> closing tag will be displayed using the default fontstyle for source code. This fontstyle varies according to browser and platform. This is recommended over tags such as bold and italic.

Example:

The function looks like this:<P>
<CODE>int max(int x, int y)
{. . .}<CODE>

Displays:

The function looks like this:

int max(int x, int y)
{. . .}

Note:
All web browsers are different, and some may not recognize this tag.

Sample

<SAMP>…</SAMP>

These tags are used to denote a sample. The text is usually displayed as a monotype (non-proportional) font.

Example:

This text is a <SAMP>sample</SAMP>

Displays:

This text is a sample.

Keyboard

<KBD>…</KBD>

The text between any <KBD> tag and its corresponding </KBD> closing tag will be displayed using the default fontstyle for keyboard type, usually a monospace (non-proportional) font. This fontstyle varies according to browser and platform. This is recommended over tags such as bold and italic.

Example:

This looks like it was written on a <KBD>typewriter</KBD>.

Displays:

This looks like it was written on a typewriter.

Note:
All web browsers are different, and some may not recognize this tag.

Defining Instance

<DFN>…</DFN>

These tags are used to denote defining instances of an term.

Example:

<DFN>Geology</DFN> is the study of the earth.

Abbreviation

<ABBREV>…</ABBREV>

These tags are used to denote abbreviations.

Example

The abbreviation for abbreviation is <ABBREV>abbr</ABBREV>.

Acronym

<ACRONYM>…</ACRONYM>

These tags are used to denote acronyms.

Example:

The New York Giants are a franchise of the </ACRONYM>NFL</ACRONYM>.

Person

<PERSON>…</PERSON>

These tags are used to denote the name of a person.

Example:

<PERSON>Abraham Lincoln</PERSON> was the 16th president of the United States.

Block Tags

Paragraph
Break
No Break
Word Break
Center
Blockquote
Preformatted
Literal
Font
Comment
Address
Byline
Horizontal Rule

Paragraph

<P>…</P>

The P element indicates a paragraph. The exact indentation, leading space, etc. of a paragraph is not specified and may be a function of other tags, style sheets, etc.

Typically, paragraphs are surrounded by a vertical space of one line or half a line. The first line in a paragraph is indented in some cases.

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>
<P>This is the text of a third paragraph.</P>

Break

**
**

The
 tag inserts a line break into your document.

Example:

I will write text until I get to this point.
  Then, I would like to start a new line.

Displays:

I will write text unti I get to this point.
Then, I would like to start a new line.

No Break

<NOBR>

All text between the start and end of the NOBR elements cannot have line breaks inserted between them.

Word Break

<WBR>

The WBR element exists to force the possibility of a word break in a no-break section.

Center

<CENTER>…</CENTER>

The text inserted between a <CENTER> tag and associated closing </CENTER> tag will be displayed in the center of the browsers viewing area.

Example

This is left.
<CENTER>This is center</CENTER>

Displays

This is left.

This is center

Blockquote

<BLOCKQUOTE>…</BLOCKQUOTE>

The BLOCKQUOTE element contains text quoted from another source.

A typical rendering might be a slight extra left and right indent, and/or italic font. The BLOCKQUOTE typically provides space above and below the quote.

Example:

I think the play ends
<BLOCKQUOTE>
<P>Soft you now, the fair Ophelia. Nymph, in thy orisons, be all
my sins remembered.
</BLOCKQUOTE>
but I am not sure.

Preformatted

<PRE>…</PRE>

The PRE element represents a character cell block of text and is suitable for text that has been formatted for a monospaced font.

The PRE tag may be used with the optional WIDTH attribute. 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.

Within preformatted text:

Line breaks within the text are rendered as a move to the beginning of the next line. (15) Anchor elements and phrase markup may be used. (16) Elements that define paragraph formatting (headings, address, etc.) must not be used. (17) The horizontal tab character (code position 9 in the HTML document character set) 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. Documents should not contain tab characters, as they are not supported consistently.

Example:

```
<PRE>
Line 1.
        Line 2.
        Line 3 aligns with line 2.
</PRE>
```

Literal

<LIT>…</LIT>

The literal tags work exactly like the preformatted tags, except they use a proportional font.

Font

<FONT>…</FONT>

The text between the <FONT> tag and its associated closing </FONT> tag will take on the attributes given within the opening <FONT> tag.

Example:

This is <FONT SIZE=+2>bigger</FONT> than the rest.

Displays:

This is bigger than the rest.

Comment

<!--- … ->

Text between the <! And > elements are not displayed. They are used only to document your HTML code.

Address

<ADDRESS>…</ADDRESS>

The ADDRESS element contains such information as address, signature and authorship, often at the beginning or end of the body of a document.

Typically, the ADDRESS element is rendered in an italic typeface and may be indented.

Example:

<ADDRESS>
Newsletter editor
J.R. Brown
JimquickPost News, Jimquick, CT 01234
Tel (123) 456 7890
</ADDRESS>

Byline

<BYLINE>…</BYLINE>

This element is used to denote authorship. The text between these tags is usually displayed with an italic font.

Example:

This page was created and is maintained by <BYLINE>Mike Sutton</BYLINE>.

Displays:

This page was created and is maintained by *Mike Sutton*.

Horizontal Rule

<HR>

The horizontal rule element <HR> is used to place a separator across a page. Additional elements can be used within the tag to affect the appearance of the separator.

Example:

<HR SIZE=6 NOSHADE>

Links

One of the most impressive feature of HTML is the ability to create hyperlinks. Hyperlinks are references to other parts of your document, other local documents, or other remote documents.

Examples:

<A HREF=#Section2>Section 2</A>

This reference will cause the browser to jump to the anchor named Section2. It would be denoted like this: <A NAME=Section2>

<A HREF=http://www.microsoft.com>Microsoft Home Page</A>

This reference will create a link to the Microsoft home page.

<A HREF=http://www.sample.com#ordering>Order Info</A>

This reference will create a link to the section named ordering on the index.html document on the sample.com server.

Lists

HTML includes a number of list elements. They may be used in combination; for example, a OL may be nested in an LI element of a UL.

Unordered List: UL, LI

The UL represents a list of items -- typically rendered as a bulleted list. The content of a UL element is a sequence of LI elements.

Example:

```
<UL>
<LI>First list item
<LI>Second list item
 <p>second paragraph of second item
<LI>Third list item
</UL>
```

Ordered List: OL

The OL element represents an ordered list of items, sorted by sequence or order of importance. It is typically rendered as a numbered list. The content of a OL element is a sequence of LI elements.

Example:

```
<OL>
<LI>Click the Web button to open URI window.
<LI>Enter the URI number in the text field of the Open URI
window. The Web document you specified is displayed.
  <ol>
   <li>substep 1
   <li>substep 2
  </ol>
<LI>Click highlighted text to move from one link to another.
</OL>
```

Menu List: MENU

The MENU element is a list of items with typically one line per item. The menu list style is typically more compact than the style of an unordered list. The content of a MENU element is a sequence of LI elements. Nested block elements are not allowed in the content of MENU elements.

Example:

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

Definition List: DL, DT, DD

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. The content of a DL element is a sequence of DT elements and/or DD elements, usually in pairs. Multiple DT may be paired with a single DD element. Documents should not contain multiple consecutive DD elements.

Example:

<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 (typically 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.

The optional COMPACT attribute suggests that a compact rendering be used, because the list items are small and/or the entire list is large.

Unless the COMPACT attribute is present, an 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.

<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>

Images/Maps

Inline Images

The IMG element refers to an image or icon via a hyperlink. Attributes of the IMG element include:

ALIGN alignment of the image with respect to the text baseline.

· `TOP' specifies that the top of the image aligns with the tallest item on the line containing the image.
· `MIDDLE' specifies that the center of the image aligns with the baseline of the line containing the image.
· `BOTTOM' specifies that the bottom of the image aligns with the baseline of the line containing the image.

ALT text to use in place of the referenced image resource, for example due to processing constraints or user preference.

ISMAP indicates an image map.

SRC specifies the URI of the image resource.

Examples:

<IMG SRC="triangle.xbm" ALT="Warning:"> Be sure to read these instructions.

Image Maps

This topic is under construction. If registered, e-mail msutton@mail.vantek.net for latest version.

Tables

Tables are used to present data is a tabular format. You can specify that the information be presented in a series of rows and columns.

Table Row

Each new row in a table is encapsulated in a set of table row tags <TR>…</TR>

Table Header

Information which is a column or row heading should be within a set of table header tags <TH>…</TH>

Table Data

General Information included in a table should be denoted by a set of table data tags <TD>…</TD>

Caption

The text between the caption tags <CAPTION>…</CAPTION> will be displayed as a caption above or below the table. Use the ALIGN= keyword to specify top or bottom.

Example:

```
<TABLE>
<TR><TH>Header1</TH><TH>Header2</TH></TR>
<TR><TD>Data1</TD><TD>Data2</TD></TR>
</TABLE>
```

Fill-Out Forms

A form is a template for a form data set and an associated method and action URI. A form data set is a sequence of name/value pair fields. The names are specified on the NAME attributes of form input elements, and the values are given initial values by various forms of markup and edited by the user. The resulting form data set is used to access an information service as a function of the action and method.

Forms elements can be mixed in with document structuring elements. For example, a PRE element may contain a FORM element, or a FORM element may contain lists which contain INPUT elements. This gives considerable flexibility in designing the layout of forms.

The FORM element contains a sequence of input elements, along with document structuring elements. The attributes are:

ACTION

specifies the action URI for the form. The action URI of a form defaults to the base URI of the document (see section Hyperlinks).

METHOD

selects a method of accessing the action URI. The set of applicable methods is a function of the scheme of the action URI of the form. See section Query Forms: METHOD=GET and section Forms with Side-Effects: METHOD=POST.

ENCTYPE

specifies the media type used to encode the name/value pairs for transport, in case the protocol does not itself impose a format.

The INPUT element represents a field for user input. The TYPE attribute discriminates between several variations of fields.

The INPUT element has a number of attributes. The set of applicable attributes depends on the value of the TYPE attribute.

Text Field: INPUT TYPE=TEXT

The default vaule of the TYPE attribute is `TEXT', indicating a single line text entry field. (Use the TEXTAREA element for multi-line text fields.)

Required attributes are:

NAME

name for the form field corresponding to this element.

The optional attributes are:

MAXLENGTH

constrains the number of characters that can be entered into a text input field. If the value of MAXLENGTH is greater the the value of the SIZE attribute, the field should scroll appropriately. The default number of characters is unlimited.

SIZE

specifies the amount of display space allocated to this input field according to its type. The default depends on the user agent.

VALUE

The initial value of the field.

Example:

```
<p>Street Address: <input name=street><br>
Postal City code: <input name=city size=16 maxlength=16><br>
Zip Code: <input name=zip size=10 maxlength=10 value="99999-9999"><br>
```

Password Field: INPUT TYPE=PASSWORD

An INPUT element with `TYPE=PASSWORD' is a text field as above, except that the value is obscured as it is entered. (see also: section Security Considerations).

Example:

```
<p>Name: <input name=login> Password: <input type=password name=passwd>
```

Check Box: INPUT TYPE=CHECKBOX

An INPUT element with `TYPE=CHECKBOX' represents a boolean choice. A set of such elements with the same name represents an n-of-many choice field. Required attributes are:

NAME

symbolic name for the form field corresponding to this element or group of elements.

VALUE

The portion of the value of the field contributed by this element.

Optional attributes are:

CHECKED

indicates that the initial state is on.

Example:

```
<p>What flavors do you like?
<input type=checkbox name=flavor value=vanilla>Vanilla<br>
<input type=checkbox name=flavor value=strawberry>Strawberry<br>
<input type=checkbox name=flavor value=chocolate checked>Chocolate<br>
```

Radio Button: INPUT TYPE=RADIO

An INPUT element with `TYPE=RADIO' represents a boolean choice. A set of such elements with the same name represents a 1-of-many choice field. The NAME and VALUE attributes are required as for check boxes. Optional attributes are:

CHECKED

indicates that the initial state is on.

At all times, exactly one of the radio buttons in a set is checked. If none of the INPUT elements of a set of radio buttons specifies `CHECKED', then the user agent must check the first radio button of the set initially.

Example:

<p>Which is your favorite?
<input type=radio name=flavor value=vanilla>Vanilla
<input type=radio name=flavor value=strawberry>Strawberry
<input type=radio name=flavor value=chocolate>Chocolate

Hidden Field: INPUT TYPE=HIDDEN

An INPUT element with `TYPE=HIDDEN' represents a hidden field.The user does not interact with this field; instead, the VALUE attribute specifies the value of the field. The NAME and VALUE attributes are required.

Example:

<input type=hidden name=context value="l2k3j4l2k3j4l2k3j4lk23">

Submit Button: INPUT TYPE=SUBMIT

An INPUT element with `TYPE=SUBMIT' represents an input option, typically a button, that instructs the user agent to submit the form. Optional attributes are:

NAME

indicates that this element contributes a form field whose value is given by the VALUE attribute. If the NAME attribute is not present, this element does not contribute a form field.

VALUE

indicates a label for the input (button).

You may submit this request internally:
<input type=submit name=recipient value=internal>
or to the external world:
<input type=submit name=recipient value=world>

Reset Button: INPUT TYPE=RESET

An INPUT element with `TYPE=RESET' represents an input option, typically a button, that instructs the user agent to reset the form's fields to their initial states. The VALUE attribute, if present, indicates a label for the input (button).

When you are finished, you may submit this request:
<input type=submit>
You may clear the form and start over at any time: <input type=reset>

Selection: SELECT

The SELECT element constrains the form field to an enumerated list of values. The values are given in OPTION elements. Attributes are:

MULTIPLE

indicates that more than one option may be included in the value.

NAME

specifies the name of the form field.

SIZE

specifies the number of visible items. Select fields of size one are typically pop-down menus, whereas select fields with size greater than one are typically lists.

Example:

```
<SELECT NAME="flavor">
<OPTION>Vanilla
<OPTION>Strawberry
<OPTION value="RumRasin">Rum and Raisin
<OPTION selected>Peach and Orange
</SELECT>
```

The initial state has the first option selected, unless a SELECTED attribute is present on any of the OPTION elements.

Option: OPTION

The Option element can only occur within a Select element. It represents one choice, and has the following attributes:

SELECTED

Indicates that this option is initially selected.

VALUE

indicates the value to be returned if this option is chosen. The field value defaults to the content of the OPTION element.

The content 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.

Text Area: TEXTAREA

The TEXTAREA element represents a multi-line text field. Attributes are:

COLS

the number of visible columns to display for the text area, in characters.

NAME

Specifies the name of the form field.

ROWS

The number of visible rows to display for the text area, in characters.

Example:

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

The content of the TEXTAREA element is the field's initial value.

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

Form Submission

An HTML user agent begins processing a form by presenting the document with the fields in their initial state. The user is allowed to modify the fields, constrained by the field type etc. When the user indicates that the form should be submitted (using a submit button or image input), the form data set is processed according to its method, action URI and enctype.

When there is only one single-line text input field in a form, the user agent should accept Enter in that field as a request to submit the form.

The form-urlencoded Media Type

The default encoding for all forms is `application/x-www-form-urlencoded'. A form data set is represented in this media type as follows:

1.The form field names and values are escaped: space characters are replaced by `+', and then reserved characters are escaped as per [URL]; that is, non-alphanumeric characters are replaced by `%HH', a percent sign and two hexadecimal digits representing the ASCII code of the character. Line breaks, as in multi-line text field values, are represented as CR LF pairs, i.e. `%0D%0A'.

2.The fields are listed in the order they appear in the document with the name separated from the value by `=' and the pairs separated from each other by `&'. Fields with null values may be omitted. In particular, unselected radio buttons and checkboxes should not appear in the encoded data, but hidden fields with VALUE attributes present should.

Query Forms: METHOD=GET

If the processing of a form is idempotent (i.e. it has no lasting observable effect on the state of the world), then the form method should be `GET'. Many database searches have no visible side-effects and make ideal applications of query forms.

To process a form whose action URL is an HTTP URL and whose method is `GET', the user agent starts with the action URI and appends a `?' and the form data set, in `application/x-www-form-urlencoded' format as above. The user agent then traverses the link to this URI just as if it were an anchor (see section Activation of Hyperlinks).

Forms with Side-Effects: METHOD=POST

If the service associated with the processing of a form has side effects (for example, modification of a database or subscription to a service), the method should be `POST'.

To process a form whose action URL is an HTTP URL and whose method is `POST', the user agent conducts an HTTP POST transaction using the action URI, and a message body of type `application/x-www-form-urlencoded' format as above. The user agent should display the response from the HTTP POST interaction just as it would display the response from an HTTP GET above.

Netscape Frames

Frames are generated by three things: FRAMESET tags, FRAME tags, and Frame Documents.

FRAME DOCUMENT

A Frame Document has a basic structure very much like your normal HTML document, except the BODY container is replaced by a FRAMESET container which describes the sub-HTML documents, or Frames, that will make up the page.

<HTML>

<HEAD>
</HEAD>

<FRAMESET>
</FRAMESET>

</HTML>

FRAME SYNTAX

Frame syntax is similar in scope and complexity to that used by tables, and has been designed to be quickly processed by Internet client layout engines.

<FRAMESET>

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

ROWS="row_height_value_list"

The ROWS attribute takes as its value a comma separated list of values. These values can be absolute pixel values, percentage values between 1 and 100, or relative scaling values. The number of rows is implicit in the number of elements in the list. Since the total height of all the rows must equal the height of the window, row heights might be normalized 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 center 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.

<FRAME>

This tag defines a single frame in a frameset. It has 6 possible attributes: SRC, NAME, MARGINWIDTH, MARGINHEIGHT, SCROLLING, and NORESIZE. The FRAME tag is not a container so it has no matching end tag.

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.
Named frames can have their window contents targeted with the new TARGET 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 an lower magins 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

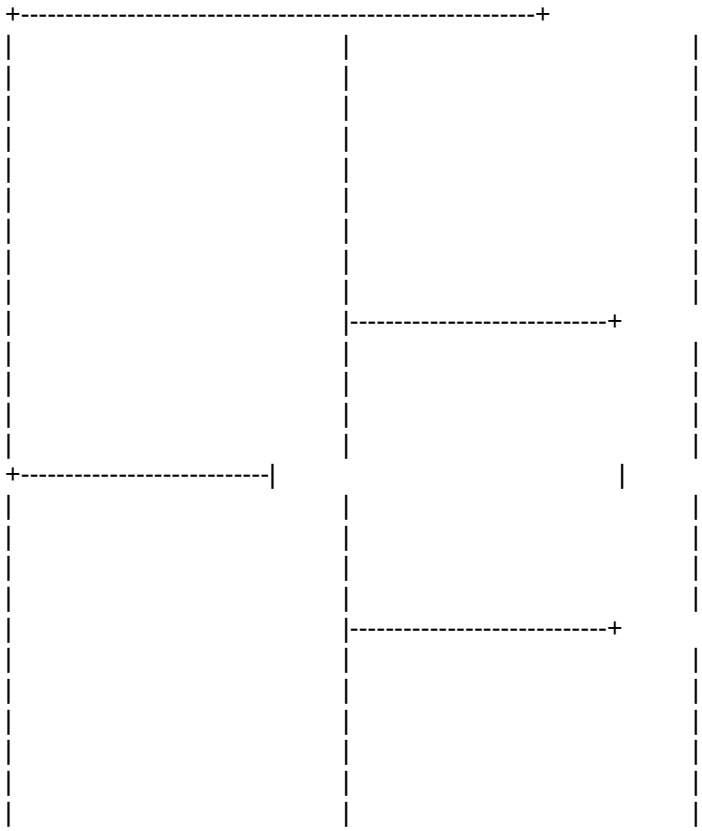
The NORESIZE attribute has no value. It is a flag that indicates that the frame is not resizable by the user. Users typically resize frames by draggin a frame edge to a new position. Note that if any frame adjacent to an edge is not resizable, that entire edge will be restricted from moving. This will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

<NOFRAMES>

This tag is for content providers who want to create alternative content that is viewable by non-Frame-capable clients. A Frame-capable Internet client ignores all tags and data between start and end NOFRAMES tags.

EXAMPLES

This example compares Frame syntax and TABLE syntax, and will show the HTML source used to display the layout below.



```
|                              |                              |
|                              |                              |
|                              |                              |
+--------------------------------------------------------------+
```

THE ABOVE LAYOUT USING TABLES

```
<TABLE WIDTH="100%" HEIGHT="100%" BORDER>
   <TR><TD ROWSPAN=2>CELL1</TD><TD>CELL2</TD></TR>
   <TR><TD ROWSPAN=2>CELL3</TD></TR>
   <TR><TD ROWSPAN=2>CELL4</TD></TR>
   <TR><TD>CELL5</TD></TR>
</TABLE>
```

THE ABOVE LAYOUT USING FRAMES

```
<FRAMESET COLS="50%,50%">
  <FRAMESET ROWS="50%,50%">
    <FRAME SRC="cell.html">
    <FRAME SRC="cell.html">
  </FRAMESET>
  <FRAMESET ROWS="33%,33%,33%">
    <FRAME SRC="cell.html">
    <FRAME SRC="cell.html">
    <FRAME SRC="cell.html">
  </FRAMESET>
</FRAMESET>
```

THE ABOVE LAYOUT USING NOFRAMES INFO

```
<FRAMESET COLS="50%,50%">

<NOFRAMES>
<h1 align=center><blink>Frame ALERT!</blink></h1>
<p>
This document is designed to be viewed using <b>Netscape 2.0</b>'s
Frame features. If you are seeing this message, you are using
a frame <i>challenged</i> browser.
</p>
<p>
A <b>Frame-capable</b> browser can be gotten from
<a href=/>Netscape Communications</a>.
</p>
</NOFRAMES>

<FRAMESET ROWS="50%,50%">
  <FRAME SRC="cell.html">
  <FRAME SRC="cell.html">
</FRAMESET>
<FRAMESET ROWS="33%,33%,33%">
  <FRAME SRC="cell.html">
  <FRAME SRC="cell.html">
  <FRAME SRC="cell.html">
</FRAMESET>

</FRAMESET>
```

* This help topic has been provided by Netscape.

Miscellaneous

Date
Time
Marquee
Background Sound
Less Than
Greater Than
Ampersand
Double Quote
Copyright
Registered
Trademark

Date

Clicking this toolbar button will insert the current date into your document.

Time

Clicking this toolbar button will insert the current time into your document.

Marquee

<MARQUEE>…</MARQUEE>

The text placed between these tags will scroll horizontally across the web page when viewed with a browser that supports this feature. At the time of this writing Netscape does not support it; however, Microsoft Internet Explorer does.

Example:

<MARQUEE LOOP=INFINITE>See the latest features!</MARQUEE>

Background Sound

<BGSOUND=*filename*>

The file specified by this tag will cause browsers which support this feature to play the audio file the specified number of times. At the time of this writing Netscape does not support it; however, Microsoft Internet Explorer does.

Example:

<BGSOUND=minuet.mid LOOP=INFINITE>

Less Than

<

This tag will cause the browser to display a literal < symbol in the document. Since < is a key symbol, it can not be used directly in HTML code as a literal character.

Greater Than

>

This tag will cause the browser to display a literal > symbol in the document. Since > is a key symbol, it can not be used directly in HTML code as a literal character.

Ampersand

&

This tag will cause the browser to display a literal & symbol in the document. Since & is a key symbol, it can not be used directly in HTML code as a literal character.

Double Quote

"

This tag will cause the browser to display a literal symbol in the document. Since is a key symbol, it can not be used directly in HTML code as a literal character.

Copyright

©

This tag will cause the browser to display a literal copyright symbol in the document.

Registered

®

This tag will cause the browser to display a literal registered symbol in the document.

Trademark

™

This tag will cause the browser to display a literal trademark symbol in the document.

Other Features

Working with HTML Files
Working with Projects
Working with Templates
Extended Character Support

Working with HTML Files

EdWin can read, edit, and save any ascii based HTML file or text file. It is not advised that you attempt to load any other type of document.

To create a new document, click on the New File button, or choose **File|New File** from the menu.

To save a document, click on the Save button, or choose **File|Save File** from the menu.

To save a document with a different name, choose **File|Save File As** from the menu. If the document has been created in the current session, and therefore has not been assigned a filename, EdWin will prompt for one.

To close a document, choose **File|Close File** from the menu. If the document has been modified, EdWin will ask whether to save the changes. Choose Yes to save the changes and close the file, No to close the file without saving changes, or Cancel to abort the close process and return to the document.

Working with Projects

EdWins unique project feature allows users to group files into a single entity called a project. The project is basically a list of path and filenames. The individual files are still maintained and saved in the appropriate locations.

However, by using projects, you can create a collection of files that are associated with a certain site or a frameset. Then by opening that project, you can open all associated files at one time. When EdWin is started, and each time a project is closed, a new blank project is started.

To open a project, choose **File|Open Project** from the menu. By default projects are documents with an epr extension.

To bring up the project window, click **View|Project Info** from the menu, or press F11. To add or remove files from the project, click on the appropriate button within the project dialog.

To save a project, choose **File|Save Project** from the menu. When saving a project, all files within that project are saved without prompting the user.

To save a project with a different filename, choose **File|Save Project As** from the menu.

To close a project, choose **File|Close Project** from the menu. All associated files will also be closed.

Working with Templates

Templates are simply HTML documents created with EdWin or by other means. By installing a document as a template, you have the option when creating new files to load the HTML code from these documents without changing the original. A new filename is assigned, so unless you save it with the same name as the original document, you will not overwrite it.

By default, the only choice in the templates dialog is (none) which will create a new blank document. However, a file called basic.htm is included with the edwin.zip file. You can install this document as a template by clicking on the new button in the templates dialog, then choose the file, and click Open.

EdWin will remember which templates have been installed from session to session.

To remove a template, click on it in the templates dialog, then click the remove button. This process will not delete the original file. It only removes it as a choice from the dialog box.

Extended Character Support

EdWin allows users to enter extended characters (such as foreign language and other symbols) into documents directly. However, since many browsers do not support these characters in their native format, EdWin converts these characters to their appropriate HTML entities when the file is closed. The entities are converted back to the extended character set when documents are reloaded.

Press F12 for a dialog of the extended character set.

How To Register

The cost to register EdWin is $35 (US Funds only). You will receive an update via e-mail attachment. If you wish to have a disk mailed to you via regular mail, add $10. When you register your use of EdWin you will receive the following:

The latest 16-bit version of EdWin for Windows 3.1x
The latest 32-bit version of EdWin for Window 95 and Windows NT
Free Technical Support
Free License for All New Releases for a period of one year
Substantial Discount on Releases after one year
Notification of bug fixes, new releases, and general product information and tips
If your version requires a registration password, this will also be furnished

REGISTER USING VISA or MASTERCARD

For your convenience we have contracted with NorthStar Solutions to process any orders you wish to place with your valid VISA or MasterCard. They may be contacted FOR ORDERS ONLY via any one of the following methods:

VOICE: 1-800-699-6395
(10:00 am to 10:00 pm, Eastern Standard Time)

From outside the U.S. please call:
1-803-699-6395
(10:00 am to 10:00 pm, Eastern Standard Time)

FAX: 1-803-699-5465 (Available 24 Hours.
International and business orders encouraged.)

E_MAIL: America Online: STARMAIL
Compuserve: 71561,2751
Internet: STARMAIL@AOL.COM

Have the following information ready when you call:

* Your VISA or MasterCard Number and Expiration Date
* The program and version number (EdWin v2.1)
* Where the latest version should be mailed, if mailed
* What disk size to mail, if mailed

NOTES: 1) NorthStar processes registrations only, please contact the author via e-mail at msutton@mail.vantek.net for product/technical support. 2) E-mailed and Faxed registrations are encouraged, but all registrations are very much appreciated.

Registration by Check or Money Order

If you wish to register using a check or money order, mail your payment along with your name, address, city, state, zip, and e-mail address to:

Michael Sutton
6507 Mill Creek Road
Memphis, TN 38134

Make checks payable to: Michael Sutton

Specify which disk size you would like, if you want the program mailed to you. If you do not specify, you will receive 3 1/2".

If you want a disk mailed to you, make sure you include the additional $10 for shipping and handling costs.

Thank You