Wilbur - HTML 3.2

Table of Contents:

WDG's Copyright and Trademark Information

Except as otherwise indicated any person is hereby authorized to view, copy, print, and distribute this document subject to the following conditions:

1. The document may be used for informational, non-commercial purposes ONLY.
2. Any copy of the document or portion thereof must include this copyright notice.
3. The Web Design Group (hereafter also known as WDG) reserves the right to revoke such authorization at any time, and any such use shall be discontinued immediately upon written notice from the Web Design Group or any of its members.

TRADEMARKS

WDG and all WDG logos and graphics contained within this site are trademarks of the Web Design Group or its members. All other brand and product names are trademarks, registered trademarks or service marks of their respective holders.

Guide for Third Parties Who Use WDG Trademarks

WDG authorizes you or any other reader of this document to include the Web Design Group's logo on any World Wide Web site, so long as the image is also a link to the WDG's main page located at http://www.htmlhelp.com.

No non-members of the WDG are authorized to display the WDG Member logo. The current list of members can be found at http://www.htmlhelp.com/about/

Warranties and Disclaimers

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. WDG ASSUMES NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION.

REFERENCES TO CORPORATIONS OR INDIVIDUALS, THEIR SERVICES AND PRODUCTS, ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. IN NO EVENT SHALL WDG BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS INFORMATION.

THIS PUBLICATION COULD INCLUDE TECHNICAL OR OTHER INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. WEB DESIGN GROUP MAY MAKE IMPROVEMENTS AND/OR CHANGES AT ANY TIME.

Should you or any viewer of this publication respond with information, feedback, data, questions, comments, suggestions or the like regarding the content of any WDG publication, any such response shall be deemed not to be confidential and WDG shall be free to reproduce, use, disclose and distribute the response to others without limitation. You agree that WDG shall be free to use any ideas, concepts or techniques contained in your response for any purpose whatsoever.

Restricted Rights Legend

Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

About the Web Design Group

In the interest of addressing your questions about who makes up the WDG, we have decided to present a short introduction in the form of a series of Questions and Answers. If you have any other questions, just ask.

What is the WDG?
> The Web Design Group is made up of experienced HTML authors that have banded together in the hopes of providing guidance and instruction to Web authors at all stages of development.

Who is in the Web Design Group?
> The members of the web design group are: (in no particular order)

> - Arnoud "Galactus" Engelfriet
> - Ian Butler
> - John Pozadzides
> - Liam Quinn
> - Tina Marie Holmboe

Retired Members include:

> - Craig D. Horton
> - Dave Salovesh
> - Filip Gieszczykiewicz

Why does the Web Design Group exist?
> *The WDG's charter reads as follows:*

>> The Web Design Group was formed to promote the use of valid and creative HTML documents. The WDG officially has no preference for browser type, screen resolution, HTML publishing tool or any other means by which HTML may be incorrectly manipulated. The WDG's sole interest is in promoting the creation of *Non-browser Specific* , *Non-resolution Specific* , *Creative* and *Informative* sites that are Accessible to *ALL* users worldwide.

When was the WDG formed?
> The WDG had it's humble beginning on May 25, 1996 with the issuance of invitations into the group by John Pozadzides. Of the original 10 invited members, four were unable to accept due to extenuating circumstances. Unfortunately, the two women who were invited both were unable to join, leaving the group with nothing but men. Despite this shortcoming, the members were able to function enough to provide all of the information available on this site until Tina Marie accepted the WDG's invitation to join on December 22, 1996.

Where is the WDG located?
> The members of the WDG reside on a virtual planet known as CyberSpace. Their physical bodies are however trapped in different countries on Earth.

How do they do that?
> It ain't easy...

What are "they" saying about HTMLHelp.com?
> Just look and see at http://www.htmlhelp.com/about/awards.html!

Introduction to Wilbur

Until recently, the latest "official" HTML version was HTML 2.0, as specified in RFC 1866. It served its purpose very well, but many HTML authors wanted more control over their document and more ways to mark up their text and enhance the appearance of their sites.

HTML 3.0

Netscape, being the leading browser at that time, introduced new tags and attributes with every new version. Other browsers tried to duplicate them, but as Netscape never fully specified their new tags, this didn't always work as expected. It led to great confusion and problems when authors used these elements and then saw they didn't work as expected in another browser.

At about the same time, the IETF's HTML working group lead by Dave Raggett introduced the HTML 3.0 draft, which included many new and very useful enhancements to HTML. Most browsers only implemented a small subset of the elements from this draft. The phrase "HTML 3.0 enhanced" quickly became popular on the Web, even though it more often than not referred to documents containing browser-specific tags, rather than documents adhering to the HTML 3.0 draft. This was one of the reasons why the draft was abandoned.

As more and more browser-specific tags were introduced, it became obvious a new standard was needed. For this reason, the W3C drafted the *Wilbur* standard, which later became known as HTML 3.2. As they put it themselves: (in the document type definition, the formal specification)

> HTML 3.2 aims to capture recommended practice as of early '96 and as such to be used as a replacement for HTML 2.0 (RFC 1866). Widely deployed rendering attributes are included where they have been shown to be interoperable. SCRIPT and STYLE are included to smooth the introduction of client-side scripts and style sheets. Browsers must avoid showing the contents of these element. Otherwise support for them is not required.

Most of the *extensions* to HTML, as introduced by the various browser developers, were not specified as thoroughly as the HTML 2.0 specs do for the standard elements. This meant that the W3C had to "reverse engineer" the correct functionality for the extensions which were chosen for HTML 3.2. Since HTML 3.2 is defined in terms of SGML, some elements had to be defined slightly differently to make them legal.

The future of HTML: Cougar

HTML 3.2 is an attempt to write down what current browsers support or should support. This will hopefully ensure that a document which is written for Wilbur will be rendered in an acceptable way by all current browsers.

The next version of HTML, which is code-named *Cougar*, will introduce new functionality, most of which comes from the now-expired HTML 3.0 draft. Some of the elements from Wilbur already hint at what can be expected. For example, the SCRIPT and STYLE elements will be used in the future to allow inclusion of inline scripts and style sheets, although currently a browser does not have to support them. It only has to hide the contents of the tags.

As it's still very early, not many details about Cougar are available yet. You can get a preview of what's to be expected from the Cougar DTD, which can be found at at http://www.w3.org/pub/WWW/MarkUp/Cougar/HTML.dtd. Cougar will introduce full style sheet support. This will allow authors to assign a *style* to a document easily, while keeping the HTML for its intended purpose: marking up the content of the document. It will also have better support for international documents.

Note

One of the reasons that HTML 3.0 didn't make it, was that it was so *big* . Because of this, future versions of HTML will be introduced in a *modular* way, so browsers can easily implement them bit by bit. An example of this approach is RFC 1942, which describes a very extensive implementation of HTML TABLEs.

HTML Basics: Terminology

Tags, content and presentation

In its most basic form, an HTML document consists of text, enclosed in tags. These tags (more accurately, these *elements*) describe the meaning of the text they contain, rather than how the enclosed text should be displayed. This concept is called **content-based** markup, as opposed to **presentational** markup.

Content-based markup allows device independence; knowing the meaning of a piece of text allows a browser to render it as good as possible on the platform it is running on. With presentational markup this is impossible. Without knowing *why* a string of text must be displayed in red 20 points Helvetica, you can't pick a good alternative way to display it on a screen where this font isn't available.

Using tags

An *element*, when used in a document, consists of an opening and a closing tag. The closing tag is not always used. It might be optional, or even forbidden. The group of elemens which have opening and closing tags are referred to as *container elements* , and the group of elements without closing tag as *empty elements* . Container tags may not overlap each other. Always close the innermost container first, if you are nesting them.

An opening tag can have certain *attributes* . These provide extra information about the tag and the text they enclose, if any. For example, the A tag has an HREF attribute which defines where the anchored text is a link to.

The attribute may have a value, although this is not necessary in all cases. If it has a value, it is specified in the "`name=value`" form. The value must be enclosed in quotes if it contains anything more than letters, digits, hyphens and/or periods. In all other cases, quoting is optional. The maximum length for an attribute value is 1024 characters, including the quotation marks (if used).

The generic structure

The document can be divided in two parts, the *head* and the *body* . The document head provides information about the document, for example its title, the author and a short description. The document body holds the actual contents of the document.

Building the body - blocks

The document body is built up with so-called *block elements* or block-level tags. A block element marks up a section of text and assigns it a particular meaning. For example, you can indicate that a section of text is a heading, a large quotation or an item in a list. There are also block elements which may only contain other block elements and no text. These elements include lists (which may only contain list items) and tables (which may only contain table rows full of cells). Some block elements may contain other block elements, instead of only text. These are sometimes referred to as *super-block elements* .

Block elements which may not contain text are used to hold certain block elements together, so they form a logical unity. A list is a good example of this; it groups all the list items inside together, so the browser knows the items are part of the same list. A slightly more complex example is the table. An HTML table is built up by rows of cells, and the table tag itself contains an optional caption, followed by one or more rows. The rows may only contain header or data cells, and the cells themselves may contain almost every element.

Special cases

A *super-block element* assigns a meaning to a set of block elements. The division tag, DIV, is probably the best example. It can be used to set a default alignment or style attributes for all the block elements it contains. This is easier to do than setting that property for each block element inside.

A special case is the preformatted text container. It is the only container in which linebreaks and spacing is used exactly as how it appears inside the source. This is very useful if you are inserting ASCII art, or text which requires a specific layout and spacing, for example the source for a program.

Text - adding the contents

Inside the block elements, the actual text is found. This text should be written only with characters in the ISO Latin 1 character set. In HTML, spaces and newlines are considered identical. They are referred to as *whitespace* , and if multiple whitespace elements are used in sequence, the browser should display only one whitespace element.

Depending on the block, the text inside it may also be marked up. In general, the *text-level tags* used for this can be divided into three categories:

- Appearance (*font tags*), which change the appearance of the text.
- Logical (*phrase tags*) which assign text a particular meaning.
- Special tags, which assign text a particular functionality.

Appearance/font tags

Font tags are used to change the appearance of the text. This includes font size changes, boldface, italics and super/subscript. However, if a browser can't perform the appearance change, it has no good way to determine a good alternative. As said above, without knowing *why* this font change should be performed, the browser can't pick another way to display/process the text. A search engine can't know something in italics is a book title unless you tell it.

This limitation can cause problems if your document depends on this appearance change. There is no guarantee or requirement that a browser will display a font tag in the way the name suggests.

Logical markup

It's not always necessary to use a font tag. Often the change in appearance is an attempt to assign a special meaning to the text. For example, italics is often used for citations or emphasized text. In these cases, a better approach is to use a *logical tag* to indicate this meaning. The browser can now pick the best way to display that kind of text on the screen.

For example, if the browser does not support italics, it can still display citations and emphasized text correctly, although probably in a different fashion.

Special markup

The third category, special tags, does assign meaning or appearance change to text, but *functionality* instead. The most common example is the hyperlink, which assigns a connection to another document to the enclosed text. Inline images also fall in this category.

Strangely enough, the Wilbur specification also include the FONT tag in this group, although it is clearly an appearance tag. The three building blocks for HTML forms (INPUT, TEXTAREA and SELECT) are also text-level tags, and can be grouped in the "Special" category.

A final note

In almost all cases, you can use each text-level tag inside another one, even when this doesn't make sense. There is no way to prevent this in the specification, so it's up to the author to use only meaningful constructs. If a meaningless construct is used (such as, for example, `<EM><INPUT TYPE=radio NAME=foo></EM>`), you can get unexpected results if a browser tries to render it.

The structure of an HTML 3.2 document

Writing a structured document does not mean that you are writing in a straitjacket. It only means you have to lay out the document in advance. It also means the document becomes easier to read, maintain and extend. While this may not seem too important if you just want a homepage, when you have a whole site to maintain, well-structured documents make life a lot easier!

It is also important to note that HTML uses the ISO-8859-1 character set. Apart from the entities defined in the Wilbur draft, the characters from this list are the only ones you should use. Other characters are not guaranteed to show up at all in a browser, let alone show up as the character you're hoping for.

Every HTML 3.2 compliant document should look basically as follows:
(Note: the line numbers are only here for the explanation below)

```
 1.   <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
 2.   <HTML>
 3.   <HEAD>
 4.   <TITLE>The title of the documents</TITLE>
 5.   <META NAME="description" CONTENT="This is a document">
 6.   <LINK REV="made" HREF="mailto:galactus@htmlhelp.com">
 7.   </HEAD>
 8.   <BODY>
 9.   ... document body
10.   </BODY>
11.   </HTML>
```

1. DOCTYPE

This is a so-called DOCTYPE declaration. It is used by SGML tools to detect what kind of document is being processed. If your document adheres to the Wilbur standard, the above is what it should look like.

If your document is HTML 2.0 compliant, the DOCTYPE of it is `<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">`

Some HTML editors like to include an arbitrary DOCTYPE declaration in your documents, even when it is not correct. Note that in particular, any doctype for HTML 3.0 is not an "official" declaration, since that proposal has been expired for a long time now.

2. HTML

This tag goes around the entire document. Basically, it states that the rest is all HTML, as opposed to some other language which may use tags within < and > brackets. In theory, it can also be used by servers to see that the document they want to send is actually HTML and not plain text. However, this is almost never done (for performance reasons, usually).

3. HEAD

The head of your document contains information about the document itself. Nothing within the HEAD section should be displayed in the document window. The head section must include the TITLE of the document. It can optionally contain things like a description, a list of keywords for search engines, and the name of the program used to create the HTML document.

The HEAD tag is optional. If you arrange all the information about the document at the top of the document, and all body tags below, it is obvious for a parser where the header ends and where the body begins.

4. TITLE

The TITLE tag is the only required tag for the head section. It is typically displayed in the browser's window title bar, and used in bookmark files and search engine result listings. For the last two situations, you should make sure the title is descriptive for the document - "Homepage" or "Index" doesn't say much in a bookmark file.

5. META

META tags provide "meta information" about the document. For example, it can give a description of the document, indicate when the document will have expired or what program was used to generate it. There are many possible META constructs, so please read the section on meta tags in the list of HTML tags.

This particular META tag provides a description of the document, which is used by search engines such as **Alta Vista** and **Infoseek**.

6. LINK

A LINK tag provides information about the document relative to the rest of the site. For example, you can have a LINK tag stating where the table of contents is, what the next document is or where the style sheet can be found.

This particular LINK tag gives the address of the document's author. Some browsers (most notably Lynx) allow you to send a comment to this person with one keystroke if this tag is defined.

9. BODY

The BODY of the document contains the actual information. There may be only one BODY statement in the document. Some editors incorrectly insert another BODY statement for each new attribute you want to add to the body, but this can have unexpected side-effects (such as some of the attributes getting ignored completely).

Designing a structured contents for your HTML document is an art in itself. I won't go into it too deeply here. Initially, use only the six headers to set up the structure of the document, adding lists, tables and other block elements until the general layout of the document is finished. Then begin filling in the blocks, marking up the text with the appropriate text-level elements. Images are very important, but as the IMG tag is a *text*-level tag, it must be contained in a block-level tag.

Often a document will be part of a set, so it will use a common style. This style should specify a standard structure for documents, including navigation aids and standard images. Writing a template is then a very handy thing. The WDG's Style guide for online hypertext discusses this in more detail.

For more information about using the various HTML elements, see the HTML Basics series.

Grouped overview of HTML elements

As explained in the section on structure of Wilbur documents, an HTML document consists of two major sections: **HEAD** and **BODY**. Each has its own permitted elements and requirements.

The elements themselves can also have requirements about where they may occur, and which elements may occur inside them. This is only important in the BODY section of a document. In here, elements can be grouped in two distinct groups: *block level* and *text level* elements. The former make up the document's structure, and the latter "dress up" the contents of a block. This terminology is explained in more detail in the HTML Basics series.

The HTML comment is a special case.

Elements for the HEAD section

The HEAD section of a document may only contain the following elements. If any other elements, or plain text, occurs inside the HEAD section, the browser should assume the HEAD ends here, and start rendering the BODY. See the syntax rules for an explanation of the syntax used in the overview.

- TITLE - Document title
- ISINDEX - Primitive search
- META - Meta-information
- LINK - Site structure
- BASE - Document location
- SCRIPT - Inline script
- STYLE - Style information

Elements for the BODY section

Block-level elements

The BODY of a document consists of multiple block elements. If plain text is found inside the body, it is assumed to be inside a paragraph P. See the syntax rules for an explanation of the syntax used in the overview.

Headings

- H1 - Level 1 header
- H2 - Level 2 header
- H3 - Level 3 header
- H4 - Level 4 header
- H5 - Level 5 header
- H6 - Level 6 header

Lists

- UL - Unordered list
- OL - Ordered list
- DIR - Directory list
- MENU - Menu item list
- LI - List item
- DL - Definition list
  - DT - Definition term
  - DD- Definition

Text containers

- P - Paragraph
- PRE - Preformatted text
- BLOCKQUOTE - Large quotation
- ADDRESS - Address information

Others

- DIV - Logical division
- CENTER - Centered division
- FORM - Input form
- HR - Horizontal rule
- TABLE - Tables

Text-level elements

These elements are used to mark up text inside block level elements. Some block level elements exclude certain text level elements, and some text level elements may only appear inside specific block level elements. This is documented in the section on that block level element.

See the syntax rules for an explanation of the syntax used in the overview.

Logical markup

- EM - Emphasized text
- STRONG - Strongly emphasized
- DFN - Definition of a term
- CODE - Code fragment
- SAMP - Sample text
- KBD - Keyboard input
- VAR - Variable
- CITE - Short citation

Physical markup

- TT - Teletype
- I - Italics
- B - Bold
- U - Underline
- STRIKE - Strikeout
- BIG - Larger text
- SMALL - Smaller text
- SUB - Subscript
- SUP - Superscript

Special markup

- A - Anchor
- BASEFONT - Default font size
- IMG - Image
- APPLET - Java applet
    - PARAM - Parameters for Java applet
- FONT - Font modification
- BR - Line break
- MAP - Client-side imagemap
    - AREA - Hotzone in imagemap

Forms

- INPUT - Input field, button, etc.
- SELECT - Selection list
    - OPTION - Selection list option
- TEXTAREA - Input area

Tables

- CAPTION - Table caption
- TR - Table row
- TH - Header cell
- TD - Table cell

Alphabetical overview of HTML elements

Note: Please see the syntax rules if you have problems understanding the format used.

- ADDRESS - Address information
- APPLET - Java applet
- AREA - Hotzone in imagemap
- A - Anchor
- BASE - Document location
- BASEFONT - Default font size
- BIG - Larger text
- BLOCKQUOTE - Large quotation
- BODY - Document body
- BR - Line break
- B - Bold
- CAPTION - Table caption
- CENTER - Centered division
- CITE - Short citation
- CODE - Code fragment
- DD - Definition
- DFN - Definition of a term
- DIR - Directory list
- DIV - Logical division
- DL - Definition list
- DT - Definition term
- EM - Emphasized text
- FONT - Font modification
- FORM - Input form
- H1 - Level 1 header
- H2 - Level 2 header
- H3 - Level 3 header
- H4 - Level 4 header
- H5 - Level 5 header
- H6 - Level 6 header
- HEAD - Document head
- HR - Horizontal rule
- HTML - HTML Document
- IMG - Images
- INPUT - Input field, button, etc.
- ISINDEX - Primitive search
- I - Italics
- KBD - Keyboard input
- LINK - Site structure
- LI - List item
- MAP - Client-side imagemap
- MENU - Menu item list
- META - Meta-information
- OL - Ordered list
- OPTION - Selection list option
- PARAM - Parameter for Java applet
- PRE - Preformatted text
- P - Paragraph
- SAMP - Sample text
- SCRIPT - Inline script
- SELECT - Selection list
- SMALL - Smaller text
- STRIKE - Strikeout
- STRONG - Strongly emphasized
- STYLE - Style information
- SUB - Subscript
- SUP - Superscript
- TABLE - Tables
- TD - Table cell
- TEXTAREA - Input area
- TH - Header cell
- TITLE - Document title
- TR - Table row
- TT - Teletype
- UL - Unordered list
- U - Underline
- VAR - Variable

Reference syntax conventions

I have used some simple conventions to make the syntax clear. It also allows me to provide the information in a short format.

To illustrate the syntax conventions, here's the section on IMG:

Appearance:	<IMG SRC=**URL**>
Attributes:	**SRC=URL**, ALT=**string**, ALIGN=left|right|top|middle|bottom, HEIGHT=**n**, WIDTH=**n**, BORDER=**n**, HSPACE=**n**, VSPACE=**n**, USEMAP=**URL**, ISMAP
Contents:	None (Empty).
May occur in:	BODY, H1, H2, H3, H4, H5, H6, P, LI, DT, DD, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, PRE, ADDRESS as well as TT, I, B, U, STRIKE, BIG, SMALL, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, BASEFONT, INPUT, SELECT, TEXTAREA and plain text.

The first section, *Appearance* , gives a common way to use this tag. As you can see here, the IMG tag does not have an ending tag. If the beginning or ending tag appears inside square brackets, it is optional and may be left off.

The next section describes the attributes for the IMG tag. If an attribute appears in **bold**, it is required, otherwise it may be omitted. In the above case, SRC is required, but the other attributes are not. Note that the attributes themselves are listed in all caps, and the possible values (if possible) in lower case. Note that an attribute value *must* be quoted if it contains more than just letters, digits, hyphens and periods.

The *contents* section describes which tags are permitted inside this tag. For IMG, there are none. And last, you can see which tags allow IMG inside them.

The attributes and their values are noted in a very compact format as well. The "|" character is used to separate mutually exclusive attributes or values. For example, **A=foo|bar** indicates that attribute "A" may get foo or bar as value, but not both, or anything else. **A=string|B=string** indicates that you may use either A *or* B, but not both.

If an attribute can take more possible values than can be given in a list, the following special symbols are used:

n
> A number. It must be an integer, and not have a "**-**" or "**+**" sign prepended. Numbers do not have to be enclosed in quotes.

p%
> A percentage. The percentage must also be an integer. Exactly what the percentage applies to depends on the tag. Percentages must be enclosed in quotes.

URL
> An URL. This can be an absolute or a relative URL, depending on the situation. In most cases, both are permitted. It is recommended that URLs always be enclosed in quotes.

string
> A string of characters. Any character is permitted, including entities. It is recommended that strings are always enclosed in quotes.

#RRGGBB
> A color code, in hexadecimal notation. The color is constructed in the red-green-blue format. Each part gets a hexadecimal number between 00 and FF, and it should be given in two digits at all times. Note that a color code must have a # as the first character, and it must be enclosed in quotes.

Basic document elements

Elements covered in this section:

- HTML - HTML document
- HEAD - Document head
- BODY - Document body
- Plain text
- Comments

HTML - HTML Document

Appearance: [<HTML>] [</HTML>]

Attributes: VERSION=`string`

Contents: HEAD followed by BODY.

May occur in: (Not appliciable).

The HTML tag is the outermost tag. It is not required and may safely be omitted. It indicates that the text is HTML (the version can be indicated with the optional VERSION attribute), but this information is almost never used by servers or browsers.

Notes:

- If used, the HTML tags should go around the *entire* document, but directly after the DOCTYPE declaration.

HEAD - Document head

Appearance: [<HEAD>] [</HEAD>]

Attributes: None.

Contents: TITLE, ISINDEX, BASE, SCRIPT, STYLE, META, LINK.

May occur in: HTML.

The HEAD part of the document provides information about the document. It should not contain text or normal markup. If a browser encounters such markup, it will assume it has arrived in the BODY section of the document already.

Notes:

- You may only omit this tag if you group all the tags which may go in it at the top.

BODY - Document body

Appearance: [<BODY>] [</BODY>]

Attributes: BACKGROUND=**URL**, BGCOLOR=#**RRGGBB**, TEXT=#**RRGGBB**, LINK=#**RRGGBB**, VLINK=#**RRGGBB**, ALINK=#**RRGGBB**

Contents: H1, H2, H3, H4, H5, H6, P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, ADDRESS, as well as TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: HTML.

The BODY tag contains the actual contents of the document. That contents should consist of block elements only. You may put in plain text in the body, this is then assumed to be inside a P container.

The attributes contain the appearance of the document. The BACKGROUND attribute should point to the location of an image, which is used as the (tiled) background of the document. The other attributes set the colors for the background, text, links, visited links and active (currently being selected) links, using the order above.

The color is composed by specifying the red, green and blue components of the color in hexadecimal notation, with a # in front. For example, to specify white, the red, green and blue components are 255, 255, 255, so you would use "#FFFFFF". You can also use the following color names, although they are not as widely supported as the codes:

Black:	#000000	**Green:**	#008000	**Silver:**	#C0C0C0	**Lime:**	#00FF00
Gray:	#808080	**Olive:**	#808000	**White:**	#FFFFFF	**Yellow:**	#FFFF00
Maroon:	#800000	**Navy:**	#000080	**Red:**	#FF0000	**Blue:**	#0000FF
Purple:	#800080	**Teal:**	#008080	**Fuchsia:**	#FF00FF	**Aqua:**	#00FFFF

The BODY tag is optional; if you put all the HEAD elements first, the browser can immediately see where the actual document body begins.

Notes:

- If the background image cannot be displayed, the color specified in BGCOLOR will be used.
- If you set one of the attributes, set them all. Otherwise your specified color may conflict with a user's default. This could result in unreadable text. For example, imagine that you set your TEXT color to light gray, but forget to set the background. Then someone with a light gray background will not see anything at all.
- Do not set unvisited and visited links to the same color, it will confuse your readers.
- The names that you can use instead of the hexadecimal values are not as widely supported as the color codes.
- Netscape 1.1 produced a "fade" effect when more than one BODY tag was used in a document. It would render the BGCOLOR colors in sequence. This bug has been fixed in later versions. Do not expect that using multiple BODY tags will give the intended results.

Plain text

In HTML, plain text is defined as normal text and entities. For the text, you can use all characters from the ISO-8859-1 character set. Not all characters in this set might be available on your platform, or they could have a special meaning in HTML. Also, if you expect that the document will be distributed with a method other than HTTP, some characters may get converted or eaten by the transport mechanism. For example, using characters above decimal 127 in "ASCII mode" FTP is not a good idea.

In such cases, use entities. An entity is constructed as follows: the "&" character, followed either by the entity's name or "#nnn", with nnn a decimal number indicating the ISO-8859-1 character you want, and finally a ";" character.

In most cases, you should use the reserved name if possible. There are also some reserved characters which do not exist in the character set used, but which are defined for HTML.

The most commonly escaped characters are "&", "<" and ">", since these three have a special meaning in HTML.

Notes:

- You can leave off the semicolon at the end of an entity if it is followed by a space or similar character. In these cases it is clear where the entity ends. But if it is followed by text, always use the semicolon.
- Characters which do not appear in the ISO-8859-1 character set should not be used in an HTML document. The same goes for numeric values which show up blank in this set. They are undefined (apart from character 32, which is the space character, and character 160, which is the non-breaking space).

HTML comments

Since HTML is officially an SGML application, the comment syntax used in HTML documents is actually the SGML comment syntax. Unfortunately this syntax is a bit unclear at first.

The definition of an SGML comment is basically as follows:

> A *comment declaration* starts with `<!`, followed by zero or more comments, followed by `>`. A *comment* starts and ends with "`--`", and does not contain any occurrence of "`--`".

This means that the following are all legal SGML comments:

1. `<!-- Hello -->`
2. `<!-- Hello -- -- Hello-->`
3. `<!---->`
4. `<!------ Hello -->`
5. `<!>`

Note that an "empty" comment tag, with just "`--`" characters, should always have a multiple of four "`-`" characters to be legal. (And yes, `<!>` is also a legal comment - it's the empty comment).

Not all HTML parsers get this right. For example, "`<!------> hello-->`" is a legal comment, as you can verify with the rule above. It is a comment tag with two comments; the first is empty and the second one contains "> hello". If you try it in a browser, you will find that the text is displayed on screen.

There are two possible reasons for this:

1. The browser sees the ">" character and thinks the comment ends there.
2. The browser sees the "`-->`" text and thinks the comment ends there.

There is also the problem with the "`--`" sequence. Some people have a habit of using things like "`<!--------------->`" as separators in their source. Unfortunately, in most cases, the number of "`-`" characters is not a multiple of four. This means that a browser who *tries* to get it right will actually get it *wrong* here and actually hide the rest of the document.

For this reason, use the following simple rule to compose valid and accepted comments:

> An HTML comment begins with "`<!--`", ends with "`-->`" and does not contain "`--`" or ">" anywhere in the comment.

Elements of the HEAD section

Elements covered in this section:

- BASE - Document location
- ISINDEX - Primitive search
- LINK - Site structure
- META - Meta-information
- SCRIPT - Inline script
- STYLE - Style information
- TITLE - Document title

BASE - Document location

Appearance: <BASE HREF=**URL**>

Attributes: **HREF=URL**

Contents: None (Empty).

May occur in: HEAD.

The BASE tag is used to indicate the correct location of the document. Normally, the browser already knows this location. But in cases such as a mirrored site, the URL used to get the document is not the one that should be used when resolving relative URLs. That's when you use the BASE tag. The required HREF attribute must contain a full URL which lists the real location of the document.

For example, in a document which contains **<BASE HREF="http://www.htmlhelp.com/">**, the relative URL in **<IMG SRC="icon/wdglogo.gif">** corresponds with the full URL **http://www.htmlhelp.com/icon/wdglogo.gif**.

Notes:

- It is not necessary to include this tag; use it only if relative URLs won't work otherwise.

ISINDEX - Primitive search

Appearance: <ISINDEX>

Attributes: PROMPT=`string`.

Contents: None (Empty).

May occur in: HEAD.

The ISINDEX tag was used before FORMs became more popular. When inserted in a document, it will allow the user to enter keywords which are then sent to the server. The server then executes a search and returns the results. The PROMPT attribute can be used to override the default text in the dialog box ("Enter search keywords: ").

Notes:

- This tag should be inserted by the server if the document can be searched. Merely inserting this tag will not make the document searchable!
- For more flexibility, use a FORM instead.

LINK - Site structure

Appearance: <LINK REL=string HREF=URL>

Attributes: REL=string, REV=string, **HREF=URL**, TITLE=string

Contents: None (Empty).

May occur in: HEAD.

LINK is used to indicate relationships between documents. There are two possible relationships: REL indicates a normal relationship to the document specified in the URL. REV indicates a reverse relationship. In other words, the *other* document has the indicated relationship with this one. The TITLE attribute can be used to suggest a title for the referenced URL or relation.

Some possible values for REL and REV:

REV="made"
>Indicates the creator of the document. Usually the URL is a mailto: URL with the creator's e-mail address. Advanced browsers will now let the reader comment on the page with just one button or keystroke.

REL="stylesheet"
>This indicates the location of the appropriate style sheet for the current document.

The following LINK tags allow advanced browsers to automatically generate a navigational buttonbar for the site. For each possible value, the URL can be either absolute or relative.

REL="home"
>Indicates the location of the homepage, or starting page in this site.

REL="toc"
>Indicates the location of the table of contents, or overview of this site.

REL="index"
>Indicates the location of the index for this site. This doesn't have to be the same as the table of contents. The index could be alphabetical, for example.

REL="glossary"
>Indicates the location of a glossary of terms for this site.

REL="copyright"
>Indicates the location of a page with copyright information for information and such on this site.

REL="up"
>Indicates the location of the document which is logically directly above the current document.

REL="next"
>Indicates the location of the next document in a series, relative to the current document.

REL="previous"
>Indicates the location of the previous document in a series, relative to the current document.

REL="help"
>Indicates the location of a help file for this site. This can be useful if the site is complex, or if the current document may require eplanations to be used correctly (for example, a large fill-in form).

Notes:

- Support for the LINK tag is currently very limited, although it may be useful for site maintenance.

META - Meta-information

Appearance: `<META NAME=string CONTENT=string>`

Attributes: HTTP-EQUIV=`string`|NAME=`string`, **CONTENT=`string`**

Contents: None (Empty).

May occur in: HEAD.

The META tag is used to convey meta-information about the document, but can also be used to specify headers for the document. You can use either HTTP-EQUIV or NAME to name the meta-information, but CONTENT must be used in both cases. By using HTTP-EQUIV, a server *should* use the name indicated as a header, with the specified CONTENT as its value. For example,

```
<META HTTP-EQUIV="Expires" CONTENT="Tue, 04 Dec 1993 21:29:02 GMT">
<META HTTP-EQUIV="Keywords" CONTENT="Nanotechnology, Biochemistry">
<META HTTP-EQUIV="Reply-to" CONTENT="dsr@w3.org (Dave Raggett)">
```

The server should include the following response headers when the document is requested:

```
Expires: Tue, 04 Dec 1993 21:29:02 GMT
Keywords: Nanotechnology, Biochemistry
Reply-to: dsr@w3.org (Dave Raggett)
```

Popular uses for META include:

`<META NAME="generator" CONTENT="Some program">`
> This indicates the program used to generate this document. It is often the name of the HTML editor used.

`<META NAME="author" CONTENT="Name">`
> This indicates the name of the author.

`<META NAME="keywords" CONTENT="keyword keyword keyword">`
> Provides keywords for search engines such as Infoseek or Alta Vista. These are added to the keywords found in the document itself. If you insert a keyword more than seven times here, the whole tag will be ignored!

`<META NAME="description" CONTENT="This is a site">`
> Search engines which support the above tag will now display the text you specify here, rather than the first few lines of text from the actual document when the document shows up in a search result. You have about 1,000 characters for your description, but not all these will be used.

`<META HTTP-EQUIV="refresh" CONTENT="n; URL=http://foo.bar/">`
> This is a so-called "meta refresh", which on certain browsers causes the document mentioned in the URL to be loaded after *n* seconds. This can be used for slide shows or for often-changing information, but has some drawbacks. In particular, if you use a value of zero seconds, the user can no longer go "Back" with his back button. He will be transferred to the specified URL, and when he presses "back" there, he will go back to the document with the refresh, which immediately redirects him to the document he tried to get away from.

`<META HTTP-EQUIV="expires" CONTENT="Tue, 20 Aug 1996 14:25:27 GMT">`
> This indicates that the document containing this META tag will expire at this date. If the document is requested after this date, the browser should load a new copy from the server, instead of using the copy in its cache.

Notes:

- Not all servers use the information from META tags to generate headers, although some browsers will treat what they find in here like it was a header.
- The "keywords" and "description" values are currently only used by **Alta Vista** and **Infoseek**.

SCRIPT - Inline scripts

Appearance: <SCRIPT> </SCRIPT>

Attributes: None.

Contents: Plain text, but should be a valid script.

May occur in: HEAD.

The SCRIPT tag is included only to ensure upward compatibility. Newer versions of HTML will include support for inline scripts, which should be contained in this tag. The tag should contain a valid script.

Note that current browsers are only required to hide the contents of the SCRIPT tag, it does not have to use the information contained therein.

In the meantime, if you need scripts in your documents, put them inside HTML comments.

Notes:

- Not all browsers support scripts.
- Since not all browsers will hide the tag's contents, you may want to enclose it in comments.
- Note that if your script uses the ">" or "--" symbols, some browsers may end the comment accidentally.

STYLE - Style markup

Appearance: <STYLE> </STYLE>

Attributes: TYPE=`string`

Contents: Plain text, but should be valid style markup.

May occur in: HEAD.

The STYLE tag is included only to ensure upward compatibility. Newer versions of HTML will include support for style sheets, and this tag can be used to provide "in-line" style information. The tag should contain only valid style statements, in the language indicated in the TYPE attribute.

Note that current browsers are only required to hide the contents of the STYLE tag, it does not have to use the information contained therein.

Notes:

- Since not all browsers will hide the tag's contents, you may want to enclose them in comments.

TITLE - Document title

Appearance: <TITLE> </TITLE>

Attributes: None.

Contents: Plain text.

May occur in: HEAD.

Each document must have exactly one TITLE element. This element provides the title of the document. It is usually displayed at the top of the browser's window, but also used to label a bookmark entry for the document and as a caption in search engine results.

It may only contain text and entities, but no markup.

Notes:

- Make sure the TITLE tag is also useful out of context. It should still be understandable when it is used as label in a bookmarks file.
- Netscape 1.x had a bug with respect to titles. When more than one TITLE tag was used in the HEAD, it would display them in a sequence, causing a "marquee" effect. This bug has been fixed in later versions.

Elements of the BODY section

The elements of the BODY section are organized in the following subsections:

- Block level elements

- Phrase (logical) markup

- Text (physical) markup

- Special text-level markup

- Form elements

- List elements

- Table elements

Block-level elements

Block-level elements are used to construct the entire HTML document. The
BODY of the document should only contain block-level elements. Some other
elements (such as DD or TD) also allow block-level elements inside them.

Elements covered in this section:

- ADDRESS - Address information

- BLOCKQUOTE - Large quotation

- CENTER - Centered division

- DIV - Logical division

- FORM - Input form

- H1 - Level 1 header
- H2 - Level 2 header
- H3 - Level 3 header
- H4 - Level 4 header
- H5 - Level 5 header
- H6 - Level 6 header

- HR - Horizontal rule

- P - Paragraph

- PRE - Preformatted text

ADDRESS - Address information

Appearance: <ADDRESS> </ADDRESS>

Attributes: None.

Contents: P, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The ADDRESS tag should be used to enclose contact information, addresses and the likes. It is often rendered with a slightly indented left margin and italics.

Notes:

- If you include an address in here, be sure to use BR for explicit linebreaks after every line, otherwise the address won't come out right.

BLOCKQUOTE - Large quotations

Appearance: `<BLOCKQUOTE> </BLOCKQUOTE>`

Attributes: None.

Contents: H1, H2, H3, H4, H5, H6, P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, ADDRESS, as well as TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

If you are quoting more than a few lines from a document, use a BLOCKQUOTE to indicate this. Block quotations are often rendered with indented margins, and possibly in italics, although a rendering with the standard quotation symbol for E-mail, "> ", is of course also possible.

Notes:

- If you quote from someone else's work, don't forget to include a credit and/or copyright notice.
- Do not use BLOCKQUOTE simply to create indented text. This is not the required rendering, so you will not achieve the effect you want on all browsers. It will also confuse page indexers and summarizers.

CENTER - Centered division

Appearance: <CENTER> </CENTER>

Attributes: None.

Contents: H1, H2, H3, H4, H5, H6, P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, ADDRESS, as well as TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The CENTER tag is one of the first Netscape extensions. It is used to indicate that large blocks of text should appear centered. In the Wilbur standard, it is defined as an alias for <DIV ALIGN=CENTER>.

The tag is more widely supported than the DIV method, as it was the first widely implemented Netscape extension to HTML 2.

Notes:

- The CENTER tag is **not** text-level markup, so do not use it to center single lines of text inside a paragraph or other block element. It will introduce a new paragraph.
- Older versions of Netscape treated CENTER as if it were text-level markup, so it was rendered without a paragraph break there.
- For better portability with browsers which do not support this tag, use ALIGN=CENTER on headers and paragraphs if possible.

DIV - Logical division

Appearance: <DIV ALIGN=foo> </DIV>

Attributes: ALIGN=left|right|center

Contents: H1, H2, H3, H4, H5, H6, P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, ADDRESS, as well as TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The DIV tag is used to mark up divisions in a document. It can enclose paragraphs, headers and other block elements. Currently, you can only use it to set the default alignment for all enclosed block elements. Future standards will most likely include more options for DIV.

Just like with other block elements such as P or H1, you can specify left, right and centered alignment.

Notes:

- The align attribute on a block element inside DIV overrides the align value of the DIV element.
- Instead of <DIV ALIGN=CENTER>, use CENTER. This element is more widely supported at the moment, even though HTML 3.2 defines both as identical.

FORM - HTML forms

Appearance: <FORM ACTION=**URL**> </FORM>

Attributes: ACTION=**URL**, METHOD=get|post, ENCTYPE=**string**

Contents: H1, H2, H3, H4, H5, H6, P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, HR, TABLE, ADDRESS, as well as TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, TH, TD and DD, LI.

Forms allow a person to send data to the WWW server. You can use the INPUT, TEXTAREA and SELECT tags to add individual elements, such as checkboxes, input fields or "drop down" lists to your form. A form may contain all markup (both text and body level tags), but it may not have a nested form.

FORM has one required attribute, ACTION, specifying the URL of a CGI script which processes the form and sends back feedback. There are two methods to send form data to a server. GET, the default, will send the form input in an URL, whereas POST sends it in the body of the submission. The latter method means you can send larger amounts of data, and that the URL of the form results doesn't show the encoded form.

You can specify an encoding type with ENCTYPE, the default of "application/x-www-form-urlencoded" is most widely supported. An alternative is "text/plain", which is typically used in combination when the ACTION attribute points to a mailto: URL. If a browser supports both, the contents of the form is sent in plain text to the indicated recipient.

Notes:

- A form should always have at least one submit button. This can be done with <INPUT TYPE=submit NAME=submitit> or with an image: <INPUT TYPE=image NAME=submitit>.
- More than one submit button is legal. If each submit button has a unique NAME attribute, the name of the selected submit button is sent along with the rest of the form input. This allows the parsing script to determine which button was pressed.
- The URL specified in the ACTION attribute does not have to be a CGI script, although you can get pretty weird results if you try to feed data to a document which isn't a CGI script. A popular reason to do this is to get a "button" which when pressed takes you to a new page. This can be done with:

```
<FORM ACTION="destination_url" METHOD=GET>
<INPUT TYPE=submit NAME=foo VALUE="Go to destination">
</FORM>
```

H1 - Level 1 heading

Appearance: <H1> </H1>

Attributes: ALIGN=left|right|center

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The level 1 heading is the most important header in the document. It should be rendered more prominently than any other header. It is usually used to indicate the title of the document. Often it has the same contents as the TITLE, although this is not required and not always a good idea. The title should be useful out of context (for example, in a bookmarks file) but the level 1 heading is only used inside the document.

The optional ALIGN attribute controls the horizontal alignment of the header. It can be LEFT, CENTER or RIGHT.

Notes:

- Headers should be used in hierarchical order.
- Do not assume that this header means "very large font size and bold." While this is a popular rendering, it can be anything the browser chooses.
- Search engines may give words appearing in headers more importance in their index. The headers are also often used to build an "outline" of the document, which appears in the search results.
- Right alignment is not as widely supported as centered.

H2 - Level 2 heading

Appearance: <H2> </H2>

Attributes: ALIGN=left|right|center

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The level 2 heading is the second most important header in the document. It should be rendered more prominently than a H3, but less prominently than a H1. It is often used to mark up chapters in a document.

The optional ALIGN attribute controls the horizontal alignment of the header. It can be LEFT, CENTER or RIGHT.

Notes:

- Headers should be used in hierarchical order.
- Do not assume that this header means "very large font size and bold." While this is a popular rendering, it can be anything the browser chooses.
- Search engines may give words appearing in headers more importance in their index. The headers are also often used to build an "outline" of the document, which appears in the search results.
- Right alignment is not as widely supported as centered.

H3 - Level 3 heading

Appearance: <H3> </H3>

Attributes: ALIGN=left|right|center

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The level 3 heading is the third most important header in the document. It should be rendered more prominently than a H4, but less prominently than a H2. It is often used to mark up sections inside a chapter in a document.

The optional ALIGN attribute controls the horizontal alignment of the header. It can be LEFT, CENTER or RIGHT.

Notes:

- Headers should be used in hierarchical order.
- Do not assume that this header means "very large font size and bold." While this is a popular rendering, it can be anything the browser chooses.
- Search engines may give words appearing in headers more importance in their index. The headers are also often used to build an "outline" of the document, which appears in the search results.
- Right alignment is not as widely supported as centered.

H4 - Level 4 heading

Appearance: <H4> </H4>

Attributes: ALIGN=left|right|center

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The level 4 heading should be rendered more prominently than a H5, but less prominently than a H3. It is often used to mark up subsections in a document.

The optional ALIGN attribute controls the horizontal alignment of the header. It can be LEFT, CENTER or RIGHT.

Notes:

- Headers should be used in hierarchical order.
- Do not assume that this header means "large font size and bold." While this is a popular rendering, it can be anything the browser chooses.
- Search engines may give words appearing in headers more importance in their index. The headers are also often used to build an "outline" of the document, which appears in the search results.
- Right alignment is not as widely supported as centered.

H5 - Level 5 heading

Appearance: <H5> </H5>

Attributes: ALIGN=left|right|center

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The level 5 heading is the second least important header in the document. It should be rendered more prominently than a H6, but less prominently than a H4. Because it is often rendered in a small font, it is not used very often. It should be used to divide sections inside a H4.

The optional ALIGN attribute controls the horizontal alignment of the header. It can be LEFT, CENTER or RIGHT.

Notes:

- Headers should be used in hierarchical order.
- Do not assume that this header means "small font size and bold." While this is a popular rendering, it can be anything the browser chooses.
- Search engines may give words appearing in headers more importance in their index. The headers are also often used to build an "outline" of the document, which appears in the search results.
- Right alignment is not as widely supported as centered.

H6 - Level 6 heading

Appearance: <H6> </H6>

Attributes: ALIGN=left|right|center

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The level 6 heading is the least important header in the document. It should be rendered less prominently than a H5, but more prominently than normal text. Because it is often rendered in a small font, it is not used very often. It should be used to divide sections inside a H5.

The optional ALIGN attribute controls the horizontal alignment of the header. It can be LEFT, CENTER or RIGHT.

Notes:

- Headers should be used in hierarchical order.
- Do not assume that this header means "very small font size and bold." While this is a popular rendering, it can be anything the browser chooses.
- Search engines may give words appearing in headers more importance in their index. The headers are also often used to build an "outline" of the document, which appears in the search results.
- Right alignment is not as widely supported as centered.

HR - Horizontal rule

Appearance: <HR>

Attributes: ALIGN=left|right|center, NOSHADE, SIZE=**n**, WIDTH=**n|p%**

Contents: None (Empty).

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

HR is used to draw horizontal rules across the browser window. If the margins are currently smaller, for example because of images (IMG) which are placed against the margins, the rule will extend to these margins instead of the whole window. A horizontal rule is typically used to separate sections within a document.

In HTML 3.2, the appearance can be controlled more than in HTML 2. You can specify the thickness of the rule with the SIZE attribute, which takes an integer number of pixels. The width of the rule can be specified in number of pixels or as a percentage of the currently available window width, using the WIDTH attribute. Don't forget that percentage values must be quoted! The NOSHADE attribute is used to indicate that the rule should not get its usual shaded appearance, but instead should be drawn as a thick line.

Notes:

- None of the attributes for HR existed in HTML 2, so they may not be supported by all browsers. This can produce bizarre effects if you are using multiple HRs in a row to produce growing or shrinking "stripes".
- If you use too many rules on a document, the end result can be that the document looks like a "sandwich" because there is little text between each rule.
- Setting an absolute width is not recommended, since you have no way to know how wide the currently available window is. Use a percentage if you have to change the width.

P - Paragraph

Appearance: <P> [</P>]

Attributes: ALIGN=left|center|right

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI, ADDRESS.

The P tag is used to indicate paragraphs. The optional attribute ALIGN indicates the preferred alignment for the contents of this paragraph. Support for ALIGN=RIGHT is not as large as support for the other two. Note that ALIGN=LEFT is the default.

Notes:

- Some browsers render extra whitespace when multiple empty paragraphs are used in sequence. This is not required by the specs, so do not count on this to get vertical whitespace in your document.
- When a paragraph has the ALIGN=CENTER or ALIGN=RIGHT attribute, some browsers do not use the default alignment for the next paragraph unless this paragraph is explicitly closed.
- In the very first version of HTML, the P tag was an empty tag like BR. Some references and books still claim that this is the case. However, HTML 2.0 defines the P tag as a container, and there is *no* difference between a paragraph with and one without explicit alignment.

PRE - Preformatted text

Appearance: <PRE> </PRE>

Attributes: WIDTH=**n**

Contents: TT, I, B, U, STRIKE, , EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

PRE is used to include sections of text in which formatting is critical. Unlike in the other HTML containers, text in a PRE pair will only be wrapped at the linebreaks in the source, and spaces will not be collapsed. You can even use tabs, although it is better to use multiple spaces since those will always be the right number.

Text inside this tag will be displayed in a monospaced font to retain the formatting. This is the reason you cannot include font-changing tags inside PRE text. Images are excluded because they can introduce problems with alignment. An image can't be translated to a certain number of characters.

The optional WIDTH attribute can be used to indicate how wide the text is (for example, WIDTH=80 for a typical text file). This would allow the browser to pick a font which fits the entire text in the current window. Unfortunately this isn't very widely supported.

Notes:

- Although text-level markup is allowed inside PRE, not all tags are supported.
- A P tag is strictly not permitted inside PRE, but if a browser encounters one, it should treat it as two newlines.
- Since HTML tags are permitted inside PRE, you cannot just "insert" a text file into an HTML document by slapping <PRE> and </PRE> around them. You have to convert the &, < and > characters into entities first.

Phrase (logical) markup

Elements covered in this section:

- CITE - Short citation
- CODE - Code fragment
- DFN - Definition of a term
- EM - Emphasized text
- KBD - Keyboard input
- SAMP - Sample text
- STRONG - Strongly emphasized
- VAR - Variable

CITE - Short citations

Appearance: <CITE> </CITE>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The CITE element marks up the title of a cited work. For example, a text discussing HTML could say <CITE>RFC 1866</CITE> says ....

Text in CITE is typically rendered in italics.

Notes:

- Do not use CITE for anything but titles of cited works. It will confuse indexers and automatic programs to extract information from your documents. Use EM for emphasis or I for text which must appear in italics.
- There is no element in HTML 3.2 to mark up short cited phrases. For longer texts, you can use BLOCKQUOTE.

CODE - Code fragments

Appearance: <CODE> </CODE>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

CODE is used for snippets of code which appear inside a paragraph of text. It is usually rendered in a monospaced font. You can use this tag to mark up things like `<CODE>for ( ; ; ) ;</CODE> is a nice way to make an endless loop in C`.

For larger blocks of code, use PRE instead. If what you are marking up is what a user should type in, use KBD.

Notes:

- CODE will usually be rendered in a monospaced font, but multiple spaces are collapsed, unlike in PRE. This can screw up the spacing in your code if you want to provide more than one line.

DFN - Definition of a term

Appearance: <DFN> </DFN>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

DFN is used to mark up terms which are used for the first time. These are often rendered in italics so the user can see this is where the term is used for the first time.

Notes:

- Not all browsers render this tag in an appropriate way.

EM - Emphasized text

Appearance: <EM> </EM>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

EM is used to indicate emphasized text. While it is often rendered identical to I, italics, using EM rather than I is preferred. It allows the browser to distinguish between emphasized text and other text which can be drawn in italics (for example citations, CITE).

EM text should be rendered distinct from STRONG text.

Notes:

- Use EM only for emphasized text. If you want to use an italic font for some other reason, use a more appropriate element like CITE, DFN or I.

KBD - Keyboard input

Appearance: <KBD> </KBD>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

KBD is used to indicate text which should be entered by the user. It is often drawn in a monospaced font, although this is not required. It differs from CODE in that CODE indicates *code fragments* and KBD indicates *input*.

Notes:

- Do not use KBD if your text requires a monospaced font; use TT instead.

SAMP - Sample text

Appearance: <SAMP> </SAMP>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

SAMP is used to indicate a sample of text which should be used literally. For example, "The text <SAMP>General Protection Fault</SAMP> is well known to Windows users."

It differs from KBD text in that KBD text indicates text the user must *enter*, whereas SAMP text can also be output.

Notes:

- Do not use SAMP if your text requires a monospaced font; use TT instead.

STRONG - Strongly emphasized text

Appearance: <STRONG> </STRONG>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

STRONG is used to indicate strongly emphasized text. While it is often rendered identical to B, boldface, using STRONG rather than B is preferred. It allows the browser to distinguish between strongly emphasized text and other text which must be drawn in boldface in the case where boldface is not available.

STRONG text should be rendered distinct from EM text.

Notes:

- Do not use STRONG if your text requires boldface; use B instead.

VAR - Variable

Appearance:	<VAR> </VAR>
Attributes:	None.
Contents:	TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.
May occur in:	DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

VAR is used to mark up variables, for example in discussions of computer programs. Using this tag allows programs to automatically generate lists of the used variables. Example: "The variable <VAR>c</VAR> is used as a counter in this program."

Notes:

- VAR will usually be rendered in a monospaced font, but multiple spaces are collapsed, unlike in PRE. This can screw up the spacing in your VAR if you want to provide more than one line.

Text (physical) markup

Elements covered in this section:

- B - Bold

- BIG - Larger text

- I - Italics

- SMALL - Smaller text

- STRIKE - Strikeout

- SUB - Subscript

- SUP - Superscript

- TT - Teletype

- U - Underline

B - Bold

Appearance:	\<B> \</B>
Attributes:	None.
Contents:	TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.
May occur in:	DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

B is used to indicate that the enclosed text must be rendered in a bold typeface. It must be rendered distinct from I-italics text.

If you want to indicate strong emphasis, use the STRONG element instead. B should only be used when you want bold typeface for some other reason than to denote strong emphasis. While the two tags usually produce the same output, the B tag does not provide any reasons *why* the enclosed text is in boldface. This means an indexer or text-only browser cannot pick a good alternative. With STRONG this is possible.

Notes:

- Putting large blocks of text in boldface makes the text hard to read.

BIG - Larger text

Appearance:	<BIG> </BIG>
Attributes:	None.
Contents:	TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.
May occur in:	DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The BIG tag (as well as SMALL) is new. A browser should draw the enclosed text in a larger font if available, and ignore the tag otherwise. Since this tag is new, support for it is not universal. The FONT tag can do the same, with SIZE="+1".

Notes:

- Nesting BIG tags may produce text in a larger font than with just one BIG tag, but this is not required by the specs (although it is recommended).
- It is legal to nest BIG and SMALL, although the results are undefined.
- BIG is not supported by all browsers, so you might want to use <FONT SIZE="+1"> instead.

I - Italics

Appearance: <I> </I>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

I is used to indicate that the enclosed text must be rendered in a italic (slanted) typeface. It must be rendered distinct from B-bold text.

You should use EM or CITE instead of I if you can. While they usually produce the same output, the I tag does not provide any reasons *why* the enclosed text is in italics. This means an indexer or text-only browser cannot pick a good alternative. With EM and CITE this is possible. The browser can now distinguish between emphasized text and citations and choose different methods to display them.

Notes:

- The I tag should only be used if text is in italics by convention.

SMALL - Smaller text

Appearance: <SMALL> </SMALL>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The SMALL tag (as well as BIG) is new. A browser should draw the enclosed text in a smaller font if available, and ignore the tag otherwise. Since this tag is new, support for it is not universal. The FONT tag can do the same, with SIZE="-1".

Notes:

- Nesting SMALL tags may produce text in a smaller font than with just one SMALL tag, but this is not required by the specs (although it is recommended).
- It is legal to nest BIG and SMALL, although the results are undefined.
- SMALL is not supported by all browsers, so you might want to use <FONT SIZE="-1"> instead.

STRIKE - Strike-through text

Appearance: <STRIKE> </STRIKE>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The STRIKE tag specifies that the enclosed text should be rendered in a strike-through appearance. Usually this is done with a line through the middle of the text.

Notes:

- Since this tag is new, support for it is not universal. If you absolutely require strikethrough text, you will have to use an image.

SUB - Subscript

Appearance: <SUB> </SUB>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

SUB specifies that the enclosed text should be rendered in subscript, with the enclosed text slightly lower than the surrounding text. This can be useful for mathematical formulas.

Notes:

- Since this tag is new, support for it is not universal. Make sure that the text would still look readable if the SUB tag weren't used.

SUP - Superscript

Appearance: <SUP> </SUP>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

SUP specifies that the enclosed text should be rendered in superscript, with the enclosed text slightly higher than the surrounding text. This can be useful for mathematical formulas.

Notes:

- Since this tag is new, support for it is not universal. Make sure that the text would still look readable if the SUP tag weren't used.

TT - Teletype font

Appearance: <TT> </TT>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The TT tag specifies that the enclosed text should be rendered in a teletype (monospaced) font. This can be used to simulate typewriter output. If possible, use CODE, SAMP or KBD instead. These tags allow the browser to pick a suitable rendering for each specific case, instead of the generic rendering you get with TT. It also makes the job easier for convertors and search robots.

Notes:

- Text inside TT is not preformatted text like PRE; spaces are collapsed and newlines ignored.

U - Underline

Appearance: <U> </U>

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

U is used to indicate the enclosed text should be underlined. As most browsers use underlining to indicate hyperlinks, try to avoid this tag. It can confuse your users if they see "hyperlinks" that do not work.

Underlining is an alternative rendering for italic text (for example, on typewriters). Since HTML has an I tag for italics, use that if this is what you are using U for.

Notes:

- Most graphical browsers do not support underlined text, since it makes it harder to distinguish text from hyperlinks.

Special text-level markup

Elements covered in this section:

- A - Anchor

- APPLET - Java applet

- AREA - Hotzone in imagemap

- BASEFONT - Default font size

- BR - Line break

- FONT - Font modification

- IMG - Image

- MAP - Client-side imagemap

- PARAM - Parameters for Java applet

A - Hyperlinks

Appearance: <A HREF=**URL**> </A>

Attributes: HREF=**URL**, NAME=**string**, REL=**string**, REV=**string**, TITLE=**string**

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, APPLET, CAPTION.

The anchor tag is the "glue" for hypertext documents. The enclosed text and/or image(s) will be selectable by the user, and doing so will take the user to the location specified in the HREF attribute. The TITLE attribute can be used to provide a description of that location, which is displayed by some browsers when the mouse moves over the URL.

The NAME attribute is used to set up "named anchors." The enclosed text will be marked as a "target" to which a browser can jump directly. For example, if you have "<A NAME="toc">Table of Contents</A>" somewhere in the document, and the user selects the URL "#toc" he will be taken to that line.

REL and REV are not widely used, although these attributes were already present in the HTML 2.0 specs. They are used to mark up relationships between the current document and the resource in the link. REL="foo" in document A, in a link pointing to B, indicates that document A has a relationship of "foo" with document B. REV="foo" indicates B has that relationship with A. Since these attributes are not widely used, there is no standard list of values for REL and REV.

Notes:

- Be sure to close the quotes around the value in HREF. Older browsers were often quite forgiving about them, but as this caused other problems, it was fixed in newer releases. This means that a hyperlink with an unclosed quote may not work correctly.
- You may not nest anchors, not even if one uses the HREF and the other uses the NAME attribute.
- The A element used with the NAME attribute requires a closing tag and non-empty content.
- TITLE is most often used for mailto URLs, where it is used to set the subject of the message. Some browsers also use it if you bookmark the link.
- Since you cannot nest anchors, you can't hyperlink a named anchor. You can however combine the NAME and HREF anchors into one anchor; <A NAME="foo" HREF="bar">text</A> works just fine.

APPLET - Java applet

Appearance:	<APPLET CODE=**string** HEIGHT=**n** WIDTH=**n**> </APPLET>
Attributes:	CODEBASE=**URL**, **CODE=string**, NAME=**string**, ALT=**string**, ALIGN=left|right|top|middle|bottom, **HEIGHT=n, WIDTH=n**, HSPACE=**n**, VSPACE=**n**
Contents:	PARAM and TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.
May occur in:	DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The APPLET tag is used to include Java applets. The CODE attribute indicates the location of the class of the applet itself. CODEBASE can be used to specify an absolute URL for the applet, similar to the BASE element for HTML documents. Other classes for this applet will be searched at the location indicated in CODEBASE. If this is not specified, the current URL will be used for the location. The NAME attribute gives the name of the applet.

Just like with IMG, WIDTH and HEIGHT are used to specify the width and height of the applet's window, and HSPACE and VSPACE control horizontal and vertical spacing around the applet. ALIGN sets the horizontal or vertical alignment for the applet.

Arguments to the applet can be specified with the PARAM tag, which goes inside the APPLET tag.

The ALT text may contain text which should be displayed if the applet cannot be run, but you should use the contents of APPLET instead. In here you may use markup, so you can provide a better alternative than with the ALT text.

Notes:

- Not all browsers support Java applets, and those that do often allow the user to disable it.
- Always provide alternative text, so your visitors get *something* when they can't see the applet.

AREA - Client-side imagemap hotspot

Appearance: <AREA SHAPE=x HREF=**URL** COORDS=**string** ALT=**string**>

Attributes: SHAPE=rect|circle|poly|default, COORDS=**string**, NOHREF|HREF=**URL**, **ALT=string**

Contents: None (Empty).

May occur in: MAP.

Inside the MAP tag, each "hotzone" in the client-side imagemap is defined with an AREA tag. The HREF attribute specifies the URL for the destination that should be chosen if this area was selected. If you specify NOHREF instead, this area won't do anything.

SHAPE and COORDS define the actual region. SHAPE can be a rectangle, circle, or polygon, and COORDS should contain a set of coordinates describing that shape. This is done with a comma separated list of numbers, enclosed in quotes. If SHAPE is set to DEFAULT, no coordinates need to be specified. The default area is what will be chosen if no others match. The syntax for COORDS depends on what shape you choose.

rect - rectangle
>   A rectangle has four coordinates. The first specifies the top left corner, and the second the bottom right corner of the rectangle. For example, <**AREA SHAPE=rect COORDS="0, 0, 9, 9"**> would specify a rectangle of 10x10 pixels, starting in the top left corner of the image.

circle - circle
>   A circle is defined by its center and radius. The COORDS attribute first specifies the coordinates of the center, and then the radius of the circle, in pixels. For example, <**AREA SHAPE=circle COORDS="10, 10, 5"**> would specify a circle with radius 5 at location (10,10) in the image.

poly - polygon
>   A polygon is built up by a list of coordinates. They are all connected in the order you present, and the last coordinate pair is connected to the first. This way you can build arbitrary figures. For example, <**AREA SHAPE=poly COORDS="10, 50, 15, 20, 20, 50"**> would specify a triangle, with edge locations (10,50), (15,20) and (20,50).

default - default
>   The default location doesn't have any coordinates, and it should be used only once in the map. It is used to indicate what should happen if the user selects one of the coordinates which are not defined in any of the other elements.

The ALT text is used by text browsers to present the URLs in the imagemap in a more readable fashion. If you leave those off, the browser can only display the "bare" URLs. The ALT text is *required* if you want your document to be valid.

Notes:

- Coordinates are specified in X,Y order: COORDS="1,0,10,19" means from X=1, Y=0 to X=10, Y=19. The top left corner is (0,0).
- If you have an area which should not do anything, not even go to the URL specified in the default area, use NOHREF.
- Just like with IMG, there can be no markup inside the ALT attribute.

BASEFONT - Default font size

Appearance: <BASEFONT SIZE=**n**>

Attributes: SIZE=**n**

Contents: None (Empty).

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The BASEFONT tag is used to suggest a default font size. FONT tags using a relative font size will use the value set here. The SIZE attribute is an integer value ranging from 1 to 7. The base font size applies to the normal and preformatted text but not to headings, except where these are modified using the FONT element with a relative font size.

Notes:

- Do not use FONT or BASEFONT to manipulate the font appearance, use BIG and SMALL for local changes, and style sheets as a more general solution.

BR - Forced line break

Appearance:

Attributes: CLEAR=left|all|right|none

Contents: None (Empty).

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The BR tag is used to force line breaks within text. Normally, linebreaks are treated as a space by browsers (except inside the PRE tag). The optional CLEAR attribute is used when you have an IMG image in your text. If that image uses ALIGN=LEFT or ALIGN=RIGHT, the text will flow around it. If you have text you want below the image, you can do this with <BR CLEAR=LEFT> or CLEAR=RIGHT to force scrolling down to a clear left or right margin, respectively. Using CLEAR=ALL will scroll down until both marings are clear. CLEAR=NONE is the default, and does nothing.

Notes:

- Some people use multiple BR tags to force whitespace. This is not required by the specs, so it may not work in all browsers.

FONT - Font appearance

Appearance:	<FONT> </FONT>
Attributes:	SIZE=`string`, COLOR=#`RRGGBB`
Contents:	TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.
May occur in:	DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The FONT tag can be used to change the appearance of the current block, in terms of size and color. The SIZE attribute can either take an absolute value, ranging from 1 (smallest) to 7 (largest), or a relative value. Using the latter will change the font relative to the current font size. For example, <FONT SIZE="+1"> will make the font size one step bigger.

The COLOR attribute takes a hex value, which is the RGB-notation of the desired color. You can also use a color name, although the names are less widely supported than the codes. See the section on BODY for a more detailed explanation on how to specify colors.

Notes:

- Avoid making extreme font changes if possible. They can make a document hard to read. If possible, use BIG instead of <FONT SIZE="+1"> and SMALL instead of <FONT SIZE="-1">.
- The color attribute is not very widely supported, so do not rely on it.
- None of the browsers which support FONT allow their users to disable it. It *is* possible to disable body colors, so if you use FONT COLOR to change font colors and the visitor has overriden your body colors, the text may wind up invisible.
- Do not use the FONT tag to emulate headers. Indexers rely on the six header elements to generate an overview of a document, and they will not be able to index your document if you use FONT instead. FONT should be used only as an enhancement of the presentation.

IMG - Inline images

Appearance: <IMG SRC=**URL**>

Attributes: **SRC=URL**, ALT=**string**, ALIGN=left|right|top|middle|bottom, HEIGHT=**n**, WIDTH=**n**,
        BORDER=**n**, HSPACE=**n**, VSPACE=**n**, USEMAP=**URL**, ISMAP

Contents: None (Empty).

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

The IMG tag is used to insert images within text. These are often called "inline" images. Note that the IMG tag is not a block tag by itself, so it must be used only within a block element. The location of the image file should be specified in the SRC attribute. It can be a relative or an absolute URL. When the image cannot be displayed for whatever reason, the browser should display the ALT text instead. The WIDTH and HEIGHT attributes should contain the image's dimensions. This allows a browser to lay out the page in advance, as it now knows where the text below the image should be drawn.

ALIGN controls the alignment of the image with respect to the text. Using a value of LEFT or RIGHT will make the image line up against the left or right margin, and text will flow around it. To force text below such an aligned image, use BR with the CLEAR attribute. The values TOP, MIDDLE and BOTTOM specify where any text following the image should be put. If more than one line follows after the image, it will be put below the image.

VSPACE and HSPACE get a numeric value, indicating the number of pixels that should be left free around the image. VSPACE covers vertical spacing and HSPACE the horizontal spacing.

The BORDER attribute is used when the image is a link. It indicates that the browser should draw a border of the indicated size around the image to show that it is a link. It's most often used as BORDER=0 to turn it off. This has the disadvantage that the image must make it very clear that it's a hyperlink.

ISMAP and USEMAP are used for imagemaps. The ISMAP attribute specifies that the link that this image is in goes to an imagemap program on the server, so the browser can send the coordinates of the selected location to the server. USEMAP is used for a so-called client-side imagemap. It specifies the URL of the imagemap information. Support for this is limited, especially if the URL points to a different document rather than an inline anchor. See the section on the MAP tag for more information about client-side imagemaps.

Notes:

- Most browsers only support GIF and JPEG file types for inline images.
- The ALT text may not contain markup, other than entities. Not all browsers support entities in ALT text, so be careful.
- Although the ALT attribute is not required, it is good practice to add it. It should replace the image's *meaning* , and not just provide a description *of* the image! If the image is purely decorative, use ALT="" or a decorative ALT text like "* " if possible.
- Using WIDTH and HEIGHT with incorrect values, or with percentage values is not valid. Some browsers (most notably Netscape) resize the image to the indicated size, but this often gives very poor results. Other browsers simply ignore the WIDTH and HEIGHT attributes in such a case.
- Browsers which do not draw boxes around hyperlinked images will not honor the BORDER attribute.

MAP - Client-side imagemap

Appearance: <MAP NAME=**string**> </MAP>

Attributes: NAME=**string**

Contents: AREA.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION.

When you use a client-side imagemap, the information on the "hot spots" (clickable areas) in the image is defined here. Every selectable area should be mentioned in an AREA tag inside the MAP tag. The NAME attribute on the MAP tag assigns a name to the imagemap. The URL for the client-side imagemap should point to this.

For example, if you have a map named "foo", you could reference to it with <**IMG** **SRC**="**map.gif**" **USEMAP**="**#foo**"> if the image was on the same page.

Notes:

- Client-side imagemaps are not widely supported yet, so try to offer a textual alternative or also use a server-side imagemap. This can be done by putting the IMG tag with the USEMAP attribute inside an A and by adding the ISMAP attribute.
- Having the imagemap data in a separate file is not as widely supported as inlined data.

PARAM - Parameters for Java applet

Appearance: <PARAM NAME=x VALUE=y>

Attributes: **NAME=string** VALUE=string

Contents: None (Empty).

May occur in: APPLET.

The PARAM element is used to provide "command-line" arguments to a Java applet, which is embedded in a document with the APPLET element. It has two attributes: NAME specifies the name of the argument, and VALUE specifies the value for this argument.

Notes:

- In a Java applet, the names of arguments are treated in a case-sensitive manner, so make sure you get the case right in the PARAM tag.
- Anyone can take the source of your Java page and change the values you supply for your parameters. Make sure the applet can handle this.

Form elements

Elements covered in this section:

- INPUT - Input field, button, etc.

- OPTION - Selection list option

- SELECT - Selection list

- TEXTAREA - Input area

INPUT - Input field

Appearance: <INPUT TYPE=x NAME=y>

Attributes: TYPE=text|password|checkbox|radio|submit|reset|file|hidden|image, NAME=**string**, VALUE=**string**, CHECKED, SIZE=**n**, MAXLENGTH=**n**, SRC=**URL**, ALIGN=top|middle|bottom|left|right

Contents: None (Empty).

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION, but must be inside a FORM.

The INPUT tag is probably the most useful tag inside forms. It can generate buttons, input fields and checkboxes. In all cases, the NAME attribute must be set.

TYPE=text
> This generates a input field, where the user can enter up to MAXLENGTH characters. The SIZE attribute lists the length of the input field (if the user enters more characters, the text will scroll). The VALUE attribute specifies the initial value for the input field.

TYPE=password
> Same as TYPE=text, but the text will be hidden by "*" or similar characters. It is still sent in the clear to the server, though.

TYPE=checkbox
> Produces a checkbox. It has two states, on and off. When it is on when the form is submitted, it will be sent as "name=on", otherwise it is ignored altogether. If you use CHECKED, it will come up checked (selected) initially.

TYPE=radio
> Produces a radio button. A radio button always exists in a group. All members of this group should have the same NAME attribute, and different VALUEs. The VALUE of the selected radio button will be sent to the server. You must specify CHECKED on exactly one radio button, which then will come up selected initially.

TYPE=submit
> Produces a button, which when pressed sends the contents of the form to the server. You can have more than one submit button in the form. Each should have a different NAME. The name and value of the pressed button will be sent to the server as well. The value of the VALUE attribute is typically used as text on the submit button.

TYPE=reset
> Also produces a button, which will restore the form to its original state if pressed. The value of the VALUE attribute is typically used as text on the reset button.

TYPE=file
> Allows the user to upload a file. It is still very new, so it is not very widely supported. It is typically presented as an input box with a button to start browsing the local hard disk. This way, a user can specify one or more filename(s) to upload.

TYPE=hidden
> Allows you to embed information in the form which you do not want changed. This can be useful if the document is generated by a script and you need to store state information. NAME and VALUE of this input field will be sent to the server without modifications.

TYPE=image
> Functions similar to a submit button, but uses an image instead. The ALIGN attribute controls the alignment of the image. The coordinates of the selected region will also be sent to the server, in the form of "NAME.x=**n**&NAME.y=**n**". A text browser will treat it as identical to a normal submit button.

Notes:

- Do not expect that if you set MAXLENGTH, you will get no more than so many characters. Anyone can modify his local copy of your form to enter as many characters as he wants.

OPTION - Selection list option

Appearance: <OPTION> [</OPTION>]

Attributes: VALUE=$string$, SELECTED

Contents: Plain text.

May occur in: SELECT.

The OPTION tag is used inside a SELECT selection list to indicate an option. You may not use markup in an option. All options listed will be displayed in a list or drop-down box, and the user can select one or more from the list. The VALUE of each option should be unique.

If you specify the SELECTED attribute, this option will appear selected when the form comes up initially.

Notes:

- Pre-selecting more than one item should only be done if the SELECT tag has the MULTIPLE attribute defined.

SELECT - Selection list

Appearance:	<SELECT NAME=**string**> </SELECT>
Attributes:	**NAME=string**, SIZE=n, MULTIPLE
Contents:	OPTION.
May occur in:	DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION, but must be inside a FORM.

The SELECT tag is used inside forms to generate a list of items from which the user can select one or more. Each item is listed in an OPTION tag. The value of the selected OPTION tag is assigned to the NAME of the SELECT tag, and both are sent to the server when the form is submitted.

The SIZE attribute indicates how many items are visible at once. If set to one, you will get a drop-down list. If it's more than one, you will get a scrollable list. If the MULTIPLE attribute is selected, the user can select more than one item from the list.

Notes:

- Exactly how a user can select more than one item at once is dependant on his platform, so do not include "instructions" on how to do this. They may be wrong and can cause a lot of confusion.
- If you use MULTIPLE, set the SIZE to more than one. This makes it easier to select more than one item.

TEXTAREA - Input area

Appearance: <TEXTAREA NAME=**string**, ROWS=**n**, COLS=**n**> </TEXTAREA>

Attributes: **NAME=string**, **ROWS=n**, **COLS=n**

Contents: Plain text.

May occur in: DIV, CENTER, BLOCKQUOTE, FORM, TH, TD, DT, DD, LI, P, H1, H2, H3, H4, H5, H6, PRE, ADDRESS, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, FONT, A, APPLET, CAPTION, but must be inside a FORM.

The TEXTAREA tag, used inside FORMs, sets up an area in which the user can type text. This text will be sent to the server when the form is submitted. The user can enter more than one line (as opposed to <INPUT TYPE=text> which only permits one line), although he will have to break lines himself.

The NAME attribute assigns the text area a name, used by the script which processes the form. ROWS and COLS are used to specify the height and width of the text area, in number of characters.

To supply default text for the text area, put it inside the TEXTAREA tag. You may not use markup for this default value.

Notes:

- Not all browsers send the linebreaks that users type (if they do so at all), so your script may have to rewrap the text itself.

List elements

Elements covered in this section:

- DD - Definition
- DIR - Directory list
- DL - Definition list
- DT - Definition term
- LI - List item
- MENU - Menu item list
- OL - Ordered list
- UL - Unordered list

DD - Definition

Appearance: <DD> [</DD>]

Attributes: None.

Contents: P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, ADDRESS and TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DL.

The DD tag is used inside a DL definition list to provide the definition of the text in the DT tag. It may contain block elements but also plain text and markup. The end tag is optional, as it's always clear from the context where the tag's contents ends.

A typical rendering is indented, one line below the DT, but this is not guaranteed.

Notes:

- Some people use this tag out of its proper context (DL only) to achieve an "indent" in text. Don't do this, it is not valid and not guaranteed to work.

DIR - Directory list

Appearance: <DIR> </DIR>

Attributes: COMPACT

Contents: LI.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The DIR element is similar to the UL element. It represents a list of short items, typically up to 20 characters each. Items in a directory list may be arranged in columns, typically 24 characters wide.

Notes:

- It is not permitted to use a block element, list element or TABLE in a LI inside a MENU or DIR.

DL - Definition list

Appearance: <DL> </DL>

Attributes: COMPACT

Contents: DT, DD.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

DL is used to provide a list of items with associated definitions. Every item should be put in a DT, and its definition goes in the DD directly following it. This list is typically rendered without bullets of any kind.

While it is legal to have a DL with only DD or DT tags, it doesn't make much sense (what good is a definition without a term?) and you shouldn't expect it to get rendered as a normal list.

Notes:

- DL may not contain plain text or any tag other than DT or DD.
- Do not use DL to create an indented section of text. This is not guaranteed to work and it is syntactically invalid HTML.

DT - Definition term

Appearance: <DT> [</DT>]

Attributes: None.

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: DL.

The DT tag is used inside DL. It marks up a term whose definition is provide by the next DD. The DT tag may only contain text-level markup.

Notes:

- Although it is legal, using a DT without a corresponding DD tag is quite pointless.

LI - List item

Appearance: <LI> [</LI>]

Attributes: TYPE=disc|square|circle when in UL, TYPE=1|a|A|i|I when in OL, VALUE=**n**

Contents: P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, ADDRESS and TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: UL, OL, DIR, MENU.

The LI element is used to mark list items within a list. When the list used is OL, ordered list, the LI element will be rendered with a number. The appearance of that number can be controlled with the TYPE attribute. Similarly, inside an unordered list UL the type of bullet that is displayed can be controlled with TYPE. DIR and MENU can't be controlled this way, as they are not required to be bulleted or numbered. For ordered lists, you can also reset the sequence with the VALUE attribute.

The TYPEs for ordered lists should give the following appearance:

- 1 - Arabic numbers (default) (1, 2, 3, 4, ...)
- a - Alphanumeric, lowercase (a, b, c, d, ...)
- A - Alphanumeric, uppercase (A, B, C, D, ...)
- i - Roman numbers, lowercase (i, ii, iii, iv, ...)
- I - Roman numbers, uppercase (I, II, III, IV, ...)

Notes:

- The attributes on LI are new with HTML 3.2, and so not supported by all browsers yet.
- Using LI outside its proper context to get a bullet in the text is not guaranteed to work. A browser is free to ignore such an out-of-context item.
- When the LI element is used inside MENU or DIR, it is not permitted to include block elements, list elements or TABLEs in the LI's contents.

MENU - Menu item list

Appearance: <MENU> </MENU>

Attributes: COMPACT

Contents: LI.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The MENU item produces a list like UL, but it should be rendered more compact. Not all browsers make this distinction, and some render it without a bullet at all.

Notes:

- It is not permitted to use a block element, list element or TABLE in a LI inside a MENU or DIR.

OL - Ordered list

Appearance: <OL> </OL>

Attributes: TYPE=1|a|A|i|I, START=**n**, COMPACT

Contents: LI

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The OL tag marks up an ordered list of items. Each item should be marked up with a LI, and it will be displayed with a number in front of it. The appearance of the number can be controlled with the TYPE attribute:

- 1 - Arabic numbers (default) (1, 2, 3, 4, ...)
- a - Alphanumeric, lowercase (a, b, c, d, ...)
- A - Alphanumeric, lowercase (A, B, C, D, ...)
- i - Roman numbers, lowercase (i, ii, iii, iv, ...)
- I - Roman numbers, uppercase (I, II, III, IV, ...)

The START attribute indicates where the list should start. The COMPACT attribute indicates that the list contains only short list items, so it may be rendered in a more compact way. This is currently not widely supported.

Notes:

- If you want a list with bullets rather than numbers, use UL.
- Not all browsers support TYPE and START, so do not assume that they will work for your visitor.

UL - Unordered list

Appearance: <UL> </UL>

Attributes: TYPE=disc|square|circle, COMPACT

Contents: LI.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

The UL element indicates an unordered list. Every item in a list is marked with LI, and usually appears with a bullet of some sort in front of it. If you need numbering, use OL for an ordered list.

The type of bullet can be suggested with the TYPE attribute. You have three possible styles: "disc" for a closed bullet, "square" for an open square and "circle" for an open bullet. The COMPACT attribute is used to indicate that the list items are short, so the browser can render the list more compact. For example, it could put more than one item on a line.

Notes:

- Do not put anything but LI list items inside an UL.
- The COMPACT attribute is not widely implemented.
- Not all browsers support the TYPE attribute, so if you use it, make sure the list still "works" without it.

Table elements

Elements covered in this section:

- CAPTION - Table caption

- TABLE - HTML Tables

- TD - Table cell

- TH - Header cell

- TR - Table row

CAPTION - Table caption

Appearance: <CAPTION> </CAPTION>

Attributes: ALIGN=top|bottom

Contents: TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in: TABLE.

The CAPTION tag is used to provide a caption for a TABLE. This caption can either appear above or below the table. This can be indicated with the ALIGN attribute. It is usually centered with respect to the table itself, and usually appears in bold or is emphasized in some other way.

The tag should appear directly below the TABLE tag, before the first TR.

Notes:

- Although you can use all text-level markup inside a CAPTION, it should be brief, so don't include images or large amounts of text in it.

TABLE - HTML Tables

Appearance: <TABLE> </TABLE>

Attributes: ALIGN=left|center|right, WIDTH=**n**|**p%**, BORDER[=**n**], CELLSPACING=**n**, CELLPADDING=**n**

Contents: One CAPTION, TR.

May occur in: BODY, DIV, CENTER, BLOCKQUOTE, FORM, TH, TD and DD, LI.

Table start with an optional caption followed by one or more rows. Each row consists of one or more cells, which can be either header or data cells. Cells can overlap across rows and columns.

The ALIGN attribute controls the alignment of the table itself, but not of the individual cells. This can be set either in the TR element for an entire row, or in the TD and TH elements for individual cells. The WIDTH attribute can be a pixel width or a percentage. It indicates the suggested width of the table, although the browser can ignore this if it is not possible. A "100%" value means the table will span across the entire browser window.

You can get a border around the table with the BORDER attribute. If you use BORDER without a value, it defaults to a width of one. Otherwise the border is drawn as wide as you specify. You cannot widen or shrink the lines between table cells or rows.

However, you can increase the whitespace inside a table. The CELLPADDING attribute indicates how many pixels there should be between a cell's contents and the border. CELLSPACING indicates how much whitespace (in pixels) there should be between individual cells.

As an example, here is a table from the HTML 3.0 draft:

```
<TABLE BORDER>
   <CAPTION>A test table with merged cells</CAPTION>
   <TR><TH ROWSPAN=2><TH COLSPAN=2>Average
   <TH ROWSPAN=2>other<BR>category<TH>Misc
   <TR><TH>height<TH>weight
   <TR><TH ALIGN=LEFT>males<TD>1.9<TD>0.003
   <TR><TH ALIGN=LEFT ROWSPAN=2>females<TD>1.7<TD>0.002
</TABLE>
```

This could appear as follows, in a text browser:

```
          A test table with merged cells
/-----------------------------------------------------\
|           |      Average      | other  | Misc  |
|           |-------------------| category |--------|
|           | height |  weight  |        |        |
|-----------------------------------------|--------|
| males     |  1.9   |  0.003   |        |        |
|-----------------------------------------|--------|
| females   |  1.7   |  0.002   |        |        |
\-----------------------------------------------------/
```

Notes:

- Some browsers (in particular, all versions of Netscape) do not honor the ALIGN attribute on a table. For these browsers, enclose the entire table in a CENTER or <DIV ALIGN=CENTER> tag.
- Avoid using pixel widths for a table. They force that the browser window is sized to a particular width to see the entire table, which is not always desirable, let alone possible.
- Tables are often used for page layout purposes. This is not recommended, since it totally screws up the

display for browsers which do not support tables, and it also often gives awkward results on small screens.

- The HTML 3 draft did not include the values for the BORDER attribute, so browsers which use this table model might draw a border around your table when you use BORDER=0.
- An empty table cell is typically drawn differently than a non-empty cell. If you need a table cell with non content, but with the same appareance as a non-empty cell, put " " in the cell.

TD - Table cell

Appearance:	<TD> [</TD>]
Attributes:	ROWSPAN=**n**, COLSPAN=**n**, NOWRAP, ALIGN=left|right|center, VALIGN=top|middle|bottom, WIDTH=**n**, HEIGHT=**n**
Contents:	H1, H2, H3, H4, H5, H6, P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, ADDRESS, as well as TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.
May occur in:	TR.

The TD tag is used to mark up individual cells inside a table row. It may contain almost all tags, including nested tables. If the cell is a label of some sort, use TH instead of TD.

The NOWRAP attribute indicates the contents of the current cell should not be wrapped. You must use BR in the cell to force line breaks to prevent the entire cell from showing up as just one line.

The ROWSPAN and COLSPAN attributes indicate how many rows or columns this cell overlaps. If you use these attributes, make sure you count correctly or you can get some *very* weird results.

The ALIGN and VALIGN attributes control the horizontal and vertical alignment of the current cell. ALIGN can be set for left, right or centered cells. VALIGN indicates that the table cell's contents should appear at the top, the middle or the bottom of the row. Note that align and valign attributes for a cell override the values set for the row.

The WIDTH and HEIGHT attributes can be used to suggest a width and height for this cell. This should be a value in pixels. Setting different widths for multiple cells in the same column, or different heights for multiple cells in one row can cause unexpected effects.

For an example on how to construct tables, please see the section on the TABLE tag.

Notes:

- While it's not strictly required, it is good practice to close each table cell explicitly. It makes your table easier to read.
- If you include a table inside a table cell, be sure to close all cells and rows. It is not required, but some browsers get the nested tables wrong and render them incorrectly.
- An empty cell is usually rendered differently than a cell with just whitespace inside it. This especially shows if you have a border defined for your table.
- If you use images in a table cell, be sure to specify the WIDTH and HEIGHT attributes in the IMG tag. This allows the browser to determine the size of the cell in advance, so it can draw the table before the image is loaded. Otherwise, the table will not appear until the image has been loaded. This can mean a considerable delay before your page displays.

TH - Header cell

Appearance:	\<TH\> [\</TH\>]
Attributes:	ROWSPAN=**n**, COLSPAN=**n**, NOWRAP, ALIGN=left|right|center, VALIGN=top|middle|bottom, WIDTH=**n**, HEIGHT=**n**
Contents:	H1, H2, H3, H4, H5, H6, P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, ADDRESS, as well as TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, APPLET, IMG, FONT, BASEFONT, BR, MAP, INPUT, SELECT, TEXTAREA and plain text.
May occur in:	TR.

Like the TD tag, a TH is used for a table cell. However, TH should be used when the cell's contents is a heading of some sort, for example, when the text is a label for a row of column.

See the section on TD for an explanation of the various attributes.

For an example on how to construct tables, please see the section on the TABLE tag.

Notes:

- While it's not strictly required, it is good practice to close each table cell explicitly. It makes your table easier to read.
- If you include a table inside a table cell, be sure to close all cells and rows. It is not required, but some browsers get the nested tables wrong and render them incorrectly.
- An empty cell is usually rendered differently than a cell with just whitespace inside it. This especially shows if you have a border defined for your table.
- If you use images in a table cell, be sure to specify the WIDTH and HEIGHT attributes in the IMG tag. This allows the browser to determine the size of the cell in advance, so it can draw the table before the image is loaded. Otherwise, the table will not appear until the image has been loaded. This can mean a considerable delay before your page displays.

TR - Table row

Appearance: <TR> [</TR>]

Attributes: ALIGN=left|right|center, VALIGN=top|middle|bottom

Contents: TH, TD.

May occur in: TABLE.

HTML tables are constructed as a sequence of rows. Each row of table cells should be enclosed in a TR tag. The end tag is optional, since it is usually obvious to see where a row ends - where the new row begins, or where the entire table ends.

The ALIGN and VALIGN attributes control the horizontal and vertical alignment of the entire row. ALIGN can be set for left, right or centered cells. VALIGN indicates that the table cell's contents should appear at the top, the middle or the bottom of the row.

For an example on how to construct tables, please see the section on the TABLE tag.

Notes:

- While it's not strictly required, it is good practice to close each table row explicitly. It makes your table easier to read.
- If you include a table inside a table cell, be sure to close all cells and rows. If you don't, some browsers may get the nested tables wrong and render it incorrectly.

The WDG's glossary of terms

Absolute URL
An absolute URL is an URL which completely and unambigously defines the location of a document or resource. It can be used out of context, but it will still point to the same file. If the meaning of the URL would change if it were taken out of context, it is called a relative URL.

Attribute
An attribute goes with a tag to provide some extra information, or to modify the meaning slightly. An attribute can have a value, but this is not necessary. The value should be put inside double quotes if it contains anything more than letters, digits, periods and hyphens.

Backbone
A major connection within the network of computers that make up the Internet.

Bandwidth
A very diminishing natural resource... bandwidth is a measure of the amount of data passing through a network at a given time. Things like huge images on web pages and needless questions posted to Usenet newsgroups waste bandwith which could be better utilized for transferring information.

Bit
A single piece of data. A bit has a numerical value of either "1" or "0".

Block-level element
A block-level element is a tag which is used to group a large section of text together to assign it a particular meaning, or to group a number of other elements together to form a unity. Block-level elements make up the document's structure.

Browser
Software that allows you, the client, to view World Wide Web resources.

Byte
A chunk of data made up of 8 bits.

Character-level element
A character-level element is used inside a block-level element to mark up a bit of text in a particular way. It should not be used outside block-level elements.

Container element
A container element is an HTML element which has an opening and closing tag, so it can contain data.

CyberSpace
A nickname for the collective systems known as the Internet.

Document type definition (DTD)
An SGML document, which formally defines the relationship between elements, what attributes each entity has and which values these attributes may take. Used by validators and checkers to make sure your documents adheres to the standard.

Domain Name
A unique identifier which assigns a name to a specific IP address. Since IP addresses are merely strings of numbers, the Domain Name system was developed to facilitate ease of recognition and recall of site addresses.

Element
An element, often referred to as *tag* , is what HTML is constructed with. It has a name, perhaps one or more attributes and can be a container or empty.

Empty tag
A starting tag which does not have an ending tag. Because it doesn't end, it cannot enclose anything so it's always empty. Note that a tag which *does* have a closing tag, but does not contain anything is also empty, but such an empty tag may be ignored by a browser.

Entity

A special sequence of characters, representing another character. An entity starts with "&", ends with ";" and should contain the name or the number of the entity. If you use a number, put "#" in front of it. Check the section on the HTML character set for a list of entities and their numbers.

Extension

A tag or attribute introduced by a browser which is not part of the current HTML specification, and usually only supported by that browser. Other browsers should just ignore it.

FAQ

A list with Frequently Asked Questions and their answers. Usually accompanies a Web site or Usenet newsgroup to provide an introduction to the site or group, so you can easily locate necessary information while the regular users don't have to answer the same questions over and over again.

FTP

File Transfer Protocol is a method to transfer files from one location to another, either on local disks or via the Internet.

Finger

A protocol with which you can get information about a user on the Internet. It is generally only available for UNIX systems, although there are also clients for Macintosh and Windows.

Homepage

The central document on a World Wide Web site. This document usually directs clients to the information that can be found within the site.
Also: the personal page for a person on the Internet, which contains information about him or her.

HTML

Hyper Text Markup Language is the language of the World Wide Web. HTML is used to write documents on the World Wide Web, and to "glue" other resources together.

HTTP

Hyper Text Transfer Protocol, a protocol to transfer files similar to FTP, but with a built-in identification of the types of the file (the MIME type). Primarily used to transfer HTML documents.

Hypertext

A document that contains highlighted areas which when selected automatically transfer the client to a different location within the original document, or another document, image, movie or other resource.

IP

Internet Protocol, the most basic protocol to communicate on the Internet. An IP *number* is a numerical address consisting of four numbers seperated by periods. Each IP address uniquely identifies a certain computer on the Internet. The domain name is used to make using them easier.

ISP

Internet Service Providers provide access to the services offered by the Internet.

Newsgroup

Usenet Newsgroups, as they're formally called, started out separately from the Internet. Right now, most people read newsgroups via the Internet, although it is also popular in the BBS world. A newsgroup is devoted to a specific topic, which can usually be guessed by its name. They are generally unmoderated, and everyone is free to contribute.

Relative URL

An URL which is not complete, and must be expanded to an absolute URL before it can be used.

Resolution

A width x height measurement of computer monitor display capabilities in pixels. Some different types of displays are VGA or SVGA and different standard resolutions are 640x480, 800x600, and 1024x768.

Search Engine

A system dedicated to the search and retrieval of information for the purpose of cataloging the results. Usually based on an index of several HTML documents, so you can easily locate the document(s) you are looking for.

Server

A computer system which can be accessed from remote systems for the purpose of exchanging data.

T-1

A type of network connection which allows for data transfer rates up to 1.5 mega-bits per second.

T-3

A type of network connection which allows for data transfer rates up to 45 mega-bits per second.

Tag

This is the popular name for an HTML element, although there is a slight theoretical difference. Basically, a tag is an element in action. It starts with "<" and ends with ">". It can have attributes, but this is not necessary. The end of the element is signaled with the tag without its attributes, and with "/" as the first character. For example, the ending tag for <B> is </B>.

Text-level element

See character-level element.

URL

An Uniform Resource Locator is the standard way to express the location of a resource on the Internet. It is used most extensively on the Web. When you want to embed an URL in a document which does not support this directly, use the <URL:> convention. For example: <URL:http://www.htmlhelp.com/>.

Usenet

The collection of newsgroups.

Validator

A program or script which is used to check the validity of HTML markup, or to detect bad or deprecated elements. A validator helps to ensure that the document can be read and used by all browsers and search engines.

Virtual Server

To assign multiple domain names to one computer, it runs several virtual servers. This way, different domain names can be used without having to hook up a new computer for each of them.

Additional information

Glossary

- Explains technical terms used in this reference.

ISO-Latin1 character set

- A graphical list of all the characters which may be used in an HTML document.

The W3 consortium

- Responsible for HTML and related standards.
- http://www.w3.org

Wilbur homepage

- At the W3 Consortium.
- http://www.w3.org/pub/WWW/MarkUp/Wilbur/

HTML 3.2 Reference Specification

- W3C Recommendation 14-Jan-1997.
- http://www.w3.org/pub/WWW/TR/REC-html32.html

Document Type Definition for HTML 3.2

- Technical definition of the standard, useful for syntax checking your document.
- http://www.w3.org/pub/WWW/MarkUp/Wilbur/HTML32.dtd

ISO 8859-1 character set overview

The HTML specifications state that HTML uses the ISO 8859-1 (Latin 1) character set for the encoding of documents. If you want to send out an HTML document and ensure everyone will be able to read it as you intended, it must be in this character set. If the protocol you use is not fully 8-bit, for example e-mail, a post to Usenet or FTP in "ascii" mode, then you should not use the characters above 127 directly, but instead in escaped form.

(Of course, the above does not apply if you are writing for a specific group of users, or need another character set for your language).

The following tables give all characters which are available in the ISO Latin 1 character set. In each table, you will see four columns:

1. **Char**. This is the actual character.
2. **Code**. This is the decimal code number for the character.
3. **Name**. This is the entity name for the character.
4. **Description**. A short description on the character.

In all cases, you may use the decimal code number to represent the character, or the entity name if that's available. A number is used like this: **©** to represent the 169th character. Since this character also has a name, you can also use **©** to represent it.

- Characters 32 - 63
- Characters 64 - 95
- Characters 96 - 127
- Characters 160 - 191
- Characters 192 - 223
- Characters 224 - 255

Notes:

ISO-8859-1 explicitly does *not* define displayable characters for positions 0-31 and 127-159, and the HTML standard does not allow those to be used for displayable characters. The only characters in this range that are used are 9, 10 and 13, which are tab, newline and carriage return respectively. If you attempt to display these invalid characters on your own system, you may find some characters displayed there, but please do not assume that other users will see the same thing (or even anything at all) on their systems.

Although the specs require that all browsers support this character set, not all actually do. In particular, Macintosh browsers display the following 14 characters incorrectly: the broken vertical bar (¦), superscript 1 (¹), 2 (²) and 3 (³), quarter (¼), half (½), three quarters (¾), uppercase (Ð) and lowercase eth (ð), uppercase (Þ) and lowercase thorn (þ), uppercase (Ý) and lowercase y acute (ý) and the multiplication sign (×). Macintosh users might want to install **Profont**, a monospaced font that displays all entities correctly.

In most cases, you will not need to use the " entity for the double quote ("). It might come in handy if you need it inside a quoted attribute value, for example as in **ALT="My "new" site"**.

ISO 8859-1 character set: 32 - 63

Char	Code	Name	Description
	32	-	Normal space
!	33	-	Exclamation
"	34	quot	Double quote
#	35	-	Hash
$	36	-	Dollar
%	37	-	Percent
&	38	amp	Ampersand
'	39	-	Apostrophe
(	40	-	Open bracket
)	41	-	Close bracket
*	42	-	Asterisk
+	43	-	Plus sign
,	44	-	Comma
—	45	-	Minus sign
.	46	-	Period
/	47	-	Forward slash

Char	Code	Name	Description
0	48	-	Digit 0
1	49	-	Digit 1
2	50	-	Digit 2
3	51	-	Digit 3
4	52	-	Digit 4
5	53	-	Digit 5
6	54	-	Digit 6
7	55	-	Digit 7
8	56	-	Digit 8
9	57	-	Digit 9
:	58	-	Colon
;	59	-	Semicolon
<	60	lt	Less than
=	61	-	Equals
>	62	gt	Greater than
?	63	-	Question mark

ISO 8859-1 character set: 64 - 95

Char	Code	Name	Description
@	64	-	At sign
A	65	-	A
B	66	-	B
C	67	-	C
D	68	-	D
E	69	-	E
F	70	-	F
G	71	-	G
H	72	-	H
I	73	-	I
J	74	-	J
K	75	-	K
L	76	-	L
M	77	-	M
N	78	-	N
O	79	-	O

Char	Code	Name	Description
P	80	-	P
Q	81	-	Q
R	82	-	R
S	83	-	S
T	84	-	T
U	85	-	U
V	86	-	V
W	87	-	W
X	88	-	X
Y	89	-	Y
Z	90	-	Z
[	91	-	Open square bracket
\	92	-	Backslash
]	93	-	Close square bracket
^	94	-	Pointer
_	95	-	Underscore

ISO 8859-1 character set: 96 - 127

Char	Code	Name	Description
`	96	-	Grave accent
a	97	-	a
b	98	-	b
c	99	-	c
d	100	-	d
e	101	-	e
f	102	-	f
g	103	-	g
h	104	-	h
i	105	-	i
j	106	-	j
k	107	-	k
l	108	-	l
m	109	-	m
n	110	-	n
o	111	-	o

Char	Code	Name	Description
p	112	-	p
q	113	-	q
r	114	-	r
s	115	-	s
t	116	-	t
u	117	-	u
v	118	-	v
w	119	-	w
x	120	-	x
y	121	-	y
z	122	-	z
{	123	-	Left brace
|	124	-	Vertical bar
}	125	-	Right brace
~	126	-	Tilde
✕	127	-	(Unused)

ISO 8859-1 character set: 160 - 191

Char	Code	Name	Description
	160	nbsp	Non-breaking space
¡	161	iexcl	Inverted exclamation
¢	162	cent	Cent sign
£	163	pound	Pound sign
¤	164	curren	Currency sign
¥	165	yen	Yen sign
¦	166	brvbar	Broken bar
§	167	sect	Section sign
¨	168	uml	Umlaut or diaeresis
©	169	copy	Copyright sign
ª	170	ordf	Feminine ordinal
«	171	laquo	Left angle quotes
¬	172	not	Logical not sign
	173	shy	Soft hyphen
®	174	reg	Registered trademark
¯	175	macr	Spacing macron

Char	Code	Name	Description
°	176	deg	Degree sign
±	177	plusmn	Plus-minus sign
²	178	sup2	Superscript 2
³	179	sup3	Superscript 3
´	180	acute	Spacing acute
µ	181	micro	Micro sign
¶	182	para	Paragraph sign
·	183	middot	Middle dot
¸	184	cedil	Spacing cedilla
¹	185	sup1	Superscript 1
º	186	ordm	Masculine ordinal
»	187	raquo	Right angle quotes
¼	188	frac14	One quarter
½	189	frac12	One half
¾	190	frac34	Three quarters
¿	191	iquest	Inverted question mark

ISO 8859-1 character set: 192 - 223

Char	Code	Name	Description
À	192	Agrave	A grave
Á	193	Aacute	A acute
Â	194	Acirc	A circumflex
Ã	195	Atilde	A tilde
Ä	196	Auml	A umlaut
Å	197	Aring	A ring
Æ	198	AElig	AE ligature
Ç	199	Ccedil	C cedilla
È	200	Egrave	E grave
É	201	Eacute	E acute
Ê	202	Ecirc	E circumflex
Ë	203	Euml	E umlaut
Ì	204	Igrave	I grave
Í	205	Iacute	I acute
Î	206	Icirc	I circumflex
Ï	207	Iuml	I umlaut

Char	Code	Name	Description
Ð	208	ETH	ETH
Ñ	209	Ntilde	N tilde
Ò	210	Ograve	O grave
Ó	211	Oacute	O acute
Ô	212	Ocirc	O circumflex
Õ	213	Otilde	O tilde
Ö	214	Ouml	O umlaut
×	215	times	Multiplication sign
Ø	216	Oslash	O slash
Ù	217	Ugrave	U grave
Ú	218	Uacute	U acute
Û	219	Ucirc	U circumflex
Ü	220	Uuml	U umlaut
Ý	221	Yacute	Y acute
Þ	222	THORN	THORN
ß	223	szlig	sharp s

ISO 8859-1 character set: 224 - 255

Char	Code	Name	Description		Char	Code	Name	Description
à	224	agrave	a grave		ð	240	eth	eth
á	225	aacute	a acute		ñ	241	ntilde	n tilde
â	226	acirc	a circumflex		ò	242	ograve	o grave
ã	227	atilde	a tilde		ó	243	oacute	o acute
ä	228	auml	a umlaut		ô	244	ocirc	o circumflex
å	229	aring	a ring		õ	245	otilde	o tilde
æ	230	aelig	ae ligature		ö	246	ouml	o umlaut
ç	231	ccedil	c cedilla		÷	247	divide	division sign
è	232	egrave	e grave		ø	248	oslash	o slash
é	233	eacute	e acute		ù	249	ugrave	u grave
ê	234	ecirc	e circumflex		ú	250	uacute	u acute
ë	235	euml	e umlaut		û	251	ucirc	u circumflex
ì	236	igrave	i grave		ü	252	uuml	u umlaut
í	237	iacute	i acute		ý	253	yacute	y acute
î	238	icirc	i circumflex		þ	254	thorn	thorn
ï	239	iuml	i umlaut		ÿ	255	yuml	y umlaut