



HTML 3.2 Reference

CoffeeCup HTML Editor Edition

Contents:

[Introduction to Wilbur](#)
[Structure of an HTML document](#)
[Reference Syntax Rules](#)
[Overview of All HTML Elements](#)
[Alphabetical Listing of All Tags](#)
[Glossary of Terms](#)

The WDG invites all to visit <http://www.htmlhelp.com> for updates to this and other HTML authoring resources. If you find any errors or omissions in the technical content of this Help file, please contact Arnoud "[Galactus](#)" Engelfriet, or [John Pozadzides](#).

This Help file was created by the Web Design Group for the greater good of the HTML authoring community. It may be distributed freely via the Internet so long as the file is not modified in any way. For information regarding distributing this Help file with any program, or for any commercial uses, please see <http://www.htmlhelp.com/distribution/guidelines/>.

Copyright 1996. Web Design Group. All Rights Reserved.

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. To inquire about commercial use of this document, please see <http://www.htmlhelp.com/distribution/guidelines/> or contact Arnoud "Galactus" Engelfriet at galactus@htmlhelp.com.
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 file 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/>.

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. UPDATES MAY BE OBTAINED FROM: <http://www.htmlhelp.com/distribution/>.

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.

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 documents 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](#). 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](#).

Overview of all 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 section.

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
- [IMG](#) - Image
- [APPLET](#) - Java applet
- [PARAM](#) - Parameters for Java applet
- [TEXTFLOW](#) - Alternative text for 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

Misc.

- ! - Comments
- Plain text

Alphabetical list of all tags

- [ADDRESS](#) - Address information
- [APPLET](#) - Java applet
- [AREA](#) - Hotzone in imagemap
- [A](#) - Anchor
- [BASE](#) - Document location
- [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
- [HR](#) - Horizontal rule
- [HEAD](#) - Header
- [HTML comments](#) - Comment tag
- [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
- [Plain](#) 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
- ID - Table data
- TEXTAREA - Input area
- TEXTFLOW - Alternative text for applet
- TH - Table header
- TITLE - Document title
- TR - Table row
- TT - Teletype
- UL - Unordered list
- U - Underline
- VAR - Variable

Syntax rules

In this overview of tags, some simple rules have been used to clarify the syntax. This also allows the information to be provided in a more compact format. To illustrate the format, here is the section on the IMG tag, with descriptions:

Appearance:

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, INPUT, SELECT, TEXTAREA and plain text.

The first section, **Appearance**, gives an example of common usage of the tag. As is illustrated in this case, 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 omitted.

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 any) in lower case. Note also 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 within the confines of this tag. For IMG, there are none.

Finally, **May occur in** explains which tags allow IMG inside them.

The attributes and their values are noted in a 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 contain more possible values than can be represented 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 upon 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 always be enclosed in quotes.

#RRGGBB

A color code, in hexadecimal notation. The color is constructed in the red-green-blue format. Each part defined by 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.

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.

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 content for HTML documents is an art in itself. This document will only touch lightly on the subject. 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 element, it must be contained within a block-level element.

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 Glossary of Terms

Absolute URL

An absolute URL is an URL which completely and unambiguously 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 set of connections 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 bandwidth 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 separated 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.

OC3

A type of network connection which allows for data transfer rates up to 155 mega-bits per second.

OC12

A type of network connection which allows for data transfer rates up to 622 mega-bits per second.

Relative URL

An URL which is not complete, and must be expanded to an absolute URL before it can be used. The mechanism behind this expansion is discussed in more detail in my article on using relative URLs.

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 is .

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.

ADDRESS - Address information

Appearance:

<ADDRESS> </ADDRESS>

Attributes:

None.

Contents:

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

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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

Notes:

- If an address is included, be sure to use BR for explicit linebreaks after each line. Otherwise the address will not display properly.

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

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, INPUT, SELECT,
TEXTAREA and plain text.

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 it is not specified, the current URL will be used for the location. The NAME attribute assigns the name of the applet.

Just as with the IMG tag, WIDTH and HEIGHT are used to specify the dimensions 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; however, a better practice would be to use the TEXTFLOW tag instead. This tag, which goes inside APPLET, may contain markup (unlike ALT, which may contain only text) making it a more powerful substitute.

Notes:

- Not all browsers support Java applets, and those that do often allow the user to disable it.
- Alternate text should always be provided so 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 NOHREF is specified 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 is chosen.

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) within the image.

poly - polygon

A polygon is built up by a list of coordinates. They are all connected in the order presented, with the last coordinate pair connected to the first. This allows the building of abstract 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 they are left off, the browser can only display the "bare" URLs.

Notes:

- If there is an area which should do nothing at all, the NOHREF tag should be used.
- Just as with [IMG](#), there can be no markup within the ALT attribute.

A - Hyperlinks

Appearance:

Attributes:

HREF=URL, NAME=string, REL=string, REV=string, TITLE=string

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text, but no nested anchors.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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, a definition within a document of: "Table of Contents" would be selectable via the URL "#toc".

REL and REV are not widely used although these attributes were present since the HTML 2.0 specifications. They are used to mark up relationships between the current document and the resource in the link.

Document A containing a link: , 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:

- The quotes must be closed 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. As a result, a hyperlink with an unclosed quote may not work correctly.
- 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 the link is bookmarked. Since anchors cannot be nested, a named anchor can't be hyperlinked. However the NAME and HREF anchors may be combined into one anchor: text.

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. This demonstrates an excellent use of the BASE tag. The required HREF attribute must contain a full URL which lists the real location of the document.

Notes:

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

BIG - Larger text

Appearance:

<BIG> </BIG>

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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, however 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, it may be preferable to use `` instead.

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, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, ID, TH, LI, and DD.

If more than a few lines from a document are being quoted, 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:

- When quoted text is included, it should always include a credit and/or copyright notice.
- Do not use BLOCKQUOTE simply to create indented text. This is not the required rendering, so it will not achieve the effect on all browsers. It will also confuse page indexers and summarizers.

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, SUP, SUB, EM,
STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP,
INPUT, SELECT, TEXTAREA and plain text.

May occur in:

HTML.

The **BODY** tag contains the actual contents of the document. Those contents should consist of block elements only. Plain text may be included in the body, but it is then assumed to be inside a P container.

The **Attributes** contain the appearance of the document. The optional 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".

The BODY tag is optional; if all the HEAD elements are included first, the browser can immediately see where the actual document body begins. However, it is considered proper style to include the BODY tag.

Notes:

- If the background image cannot be displayed, the color specified in BGCOLOR will be used. For this reason, the Background color should be carefully chosen to match the intended background image.
- If one attribute is set, set them all. Otherwise the specified color may conflict with a user's default. This could result in unreadable text. For example, imagine that the TEXT color is set to light gray, but the background is unassigned. Then a visitor with their preferences set to display 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 visitors.
- The specification suggests a list of names that can be used instead of the hexadecimal values, however these names are not widely supported.
- The specification also does not assign the color names to specific hex values, although it says they are picked from the "standard MS Windows VGA palette." The names are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.
- 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.

BR - Forced line break

Appearance:

Attributes:

CLEAR=left|all|right|none

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, INPUT, SELECT, TEXTAREA and plain text.

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 an IMG is present in the text. If that image uses ALIGN=LEFT or ALIGN=RIGHT, the text will flow around it.

If text should be displayed below the image, this can be accomplished 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 margins are clear. CLEAR=NONE is the default, and does nothing.

Notes:

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

B - Bold

Appearance:

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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 strong emphasis is intended, use the [STRONG](#) element instead. **B** should only be used to bold a 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 reason why the enclosed text is in boldface. This means an indexer or text-only browser might not pick a suitable alternative. With [STRONG](#) it is more likely.

Notes:

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

CAPTION - Table caption

Appearance:

<CAPTION> </CAPTION>

Attributes:

ALIGN=top|bottom

Contents:

TT, I, B, U, STRIKE, BIG, SMALL, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, 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. The relative position is indicated with the ALIGN attribute. It is usually centered with respect to the table itself, and usually appears in bold or otherwise emphasized.

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

Notes:

- Although all text-level markup may be used inside a CAPTION, it should be brief. Don't include images or large amounts of text within it.

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, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, ID, TH, LI, and DD.

The **CENTER** tag was 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.

CITE - Short citations

Appearance:

<CITE> </CITE>

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

The **CITE** element indicates that the text enclosed is cited from some reference. It could be rendered with smart quotes, but usually browsers use italics.

Notes:

- Do not use CITE for anything but citations. It will confuse indexers and automatic programs to extract information from documents.
- Use [EM](#) for emphasis or [I](#) for text which must appear in italics.

CODE - Code fragments

Appearance:

<CODE> </CODE>

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

CODE is used for snippets of code which appear inside a paragraph of text. It is usually rendered in a monospaced font.

This tag may be used to mark up things such as: "<CODE>for (; ;) ;</CODE> is a nice way to make an endless loop in C."

For larger blocks of code, use [PRE](#). If user input instructions are being given, use [KBD](#) instead.

Notes:

- **CODE** will usually be rendered in a monospaced font, but multiple spaces are collapsed, unlike in [PRE](#). This can mess up the spacing in within the code if more than one line is being provided.

DD - Definition

Appearance:

<DD> [</DD>]

Attributes:

None.

Contents:

P, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, BLOCKQUOTE, FORM, HR, TABLE, TT, I, B, U, STRIKE, BIG, SMALL, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, 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 invalid HTML and not guaranteed to work.

DFN - Definition of a term

Appearance:

<DFN> </DFN>

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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.

DIR - Directory list

Appearance:

<DIR> </DIR>

Attributes:

COMPACT

Contents:

[LI](#).

May occur in:

[BODY](#), [DIV](#), [CENTER](#), [BLOCKQUOTE](#), [FORM](#), [TD](#), [TH](#), [LI](#), and [DD](#).

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 [DIV](#).

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, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, ID, TH, LI, and DD.

The **DIV** tag is used to mark up divisions in a document. It can enclose paragraphs, headers and other block elements.

Currently, DIV may only be used to set the default alignment for all enclosed block elements. Future standards will most likely include more options for DIV.

As with other block elements such as P or H1, the alignment may be specified as left, right and centered.

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.

DL - Definition list

Appearance:

<DL> </DL>

Attributes:

COMPACT

Contents:

DT, DD.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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 it shouldn't be expected to render 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, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in:

BODY, 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.

EM - Emphasized text

Appearance:

Attributes:

None.

Contents:

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

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, INPUT, SELECT, TEXTAREA and plain text.

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 an italic font is the only desired effect, use a more appropriate element like CITE, DFN or I.

FONT - Font appearance

Appearance:

Attributes:

SIZE=string, COLOR=#RRGGBB

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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, will make the font size one step bigger.

The COLOR attribute takes a hex value, which is the RGB-notation of the desired color. 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.
- Optionally, use [BIG](#) instead of and [SMALL](#) instead of .
- 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 FONT COLOR has been used and the visitor has overridden the 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 a document if FONT is used in this manner.
- FONT should be used only as an enhancement of the presentation.

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](#), [FORM](#), [HR](#), [TABLE](#), [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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text, but no nested form.

May occur in:

[BODY](#), [DIV](#), [CENTER](#), [BLOCKQUOTE](#), [FORM](#), [ID](#), [TH](#), [LI](#), and [DD](#).

Forms allow a person to send data to the Web server. The [INPUT](#), [TEXTAREA](#) and [SELECT](#) tags can be used to add individual elements, such as checkboxes, input fields or "drop down" lists to a form. A form may contain all markup (both text and body level tags), but it may not contain another 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 larger amounts of data can be sent, and that the URL of the form results doesn't show the encoded form.

Although encoding type may be specified with ENCTYPE, currently only "application/x-www-form-urlencoded" is supported. Once more types are developed, it will be useful to add this attribute.

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>.
- The URL specified in the ACTION attribute does not have to be a CGI script, although weird results can be achieved if data is fed to a document which isn't a CGI script.
- A popular reason to do this is to get a "button" which when pressed takes the viewer to a new page. For example: <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, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA and plain text.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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 "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.

H5 - Level 5 heading

Appearance:

<H5> </H5>

Attributes:

ALIGN=left|right|center

Contents:

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

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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 "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.

H6 - Level 6 heading

Appearance:

<H6> </H6>

Attributes:

ALIGN=left|right|center

Contents:

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

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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 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.

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:

- This tag may only be omitted if all the tags which may go in it are grouped at the top. However, this is NOT considered good practice.

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, TD, TH, LI, and DD.

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. It is possible to 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 too many rules are used 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 there is no way to know how wide the currently available window is. Use a percentage if changing the width is necessary.

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.

HTML - HTML Document

Appearance:

[<HTML>] [</HTML>]

Attributes:

VERSION=string

Contents:

HEAD followed by BODY.

May occur in:

(Not applicable).

The **HTML** tag is the outermost tag. It is not required and may safely be omitted, although it is considered good practice to include it. 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.

IMG - Inline images

Appearance:

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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 off borders. 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.

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:

[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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text, 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 an 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).

TYPE=password

Same as TYPE=text, but the text will be hidden by "*" or similar characters. It is still sent without encryption to the server, however.

TYPE=checkbox

Produces a checkbox. It has two states, on and off. When the form is submitted with a value of "on", it will be sent as "name=on", otherwise it is ignored altogether. If assigned the attribute CHECKED, it will come up checked (selected) initially.

TYPE=radio

Produces a radio button. A radio button always exists within 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. Exactly one radio button should be specified as CHECKED, which then will come up selected initially.

TYPE=submit

Produces a button, which when pressed sends the contents of the form to the server. More than one submit button may be present within a form. Each should have a different NAME. The name and value of the pressed button will be sent to the server as well.

TYPE=reset

Produces a button, which will restore the form to its original state if pressed.

TYPE=file

Allows the user to upload a file. It is still very new, so it is not very widely supported.

TYPE=hidden

Allows embedded information within the form which cannot be changed. This can be useful if the document is generated by a script and state information needs to be stored. NAME and VALUE of this input field will be sent to the server without modifications.

TYPE=image

Functions as 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 identical to a normal submit button.

Notes:

- Do not expect that setting MAXLENGTH will ensure a limited number of characters. A local copy of the form could be modified to enter as many characters as desired.

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 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.

I - Italics

Appearance:

<I> </I>

Attributes:

None.

Contents:

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

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, INPUT, SELECT, TEXTAREA and plain text.

I is used to indicate that the enclosed text must be rendered in an italic (slanted) typeface. It must be rendered distinct from B-bold text. EM or CITE should be used instead of I if at all possible.

While all of these tags usually produce the same output, the I tag does not provide any rationale for the enclosed text to be in italics. For this reason, an indexer or text-only browser cannot pick a good alternative. With EM and CITE the browser can 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.

KBD - Keyboard input

Appearance:

<KBD> </KBD>

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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** simply because text requires a monospaced font; use [TT](#) 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, meaning the other document has the indicated relationship with this one. The TITLE attribute can be used to suggest a title for the referenced URL.

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 explanations 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.

LI - List item

Appearance:

 []

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, TT, I, B, U, STRIKE, BIG, SMALL, SUP, SUB, EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, A, IMG, FONT, APPLET, BR, SCRIPT, 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, the sequence can be reset 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, lowercase (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 the introduction of HTML 3.2, and as a result are 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.

MAP - Client-side imagemap

Appearance:

<MAP NAME="string"> </MAP>

Attributes:

NAME=string

Contents:

[AREA](#).

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

When a client-side imagemap is used, the information about the "hot spots" (clickable areas) in the image are defined within the **MAP** tag. 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 of the client-side imagemap should point to this.

For example, if there is a map named "foo", it could be referenced with 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 inline data.

MENU - Menu item list

Appearance:

<MENU> </MENU>

Attributes:

COMPACT

Contents:

LI.

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

The **MENU** item produces a list like UL, but it should be rendered in a more compact manner. 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, TABLE or a LI inside a MENU or DIR.

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. HTTP-EQUIV or NAME can be used 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 a keyword is inserted more than seven times here, the whole tag will be ignored in almost all cases!

<META NAME="description" CONTENT="This is a site">

Search engines which support the above tag will display the text specified here, rather than the first few lines of text from the actual document when the document shows up in a search result. The description should be limited to two lines or less.

Notes:

- Not all servers use the information from META tags to generate headers, although some browsers will treat what they find within the tag as if it was a header.
- The "keywords" and "description" values are currently only used by [Alta Vista](#) and [Infoseek](#).

OL - Ordered list

Appearance:

Attributes:

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

Contents:

[LI](#)

May occur in:

[BODY](#), [DIV](#), [CENTER](#), [BLOCKQUOTE](#), [FORM](#), [TD](#), [TH](#), [LI](#), and [DD](#).

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, uppercase (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:

- To create 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.

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 the SELECTED attribute is specified, the 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.

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.
- The source of a Java page can be read and changed by visitors. Therefore the values supplied for the parameters should be able to handle this.

Plain text

In HTML, plain text is defined as normal text and entities. For the text, all characters from the ISO-8859-1 character set may be used. Not all characters in this set may be available on every platform, and they could have a special meaning in HTML. Additionally, if the document will be distributed in 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:

- The semicolon may be left off 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).

PRE - Preformatted text

Appearance:

<PRE> </PRE>

Attributes:

WIDTH=n

Contents:

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

May occur in:

BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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. For this reason font-changing tags cannot be included inside PRE text. Images are also 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 widely supported.

Notes:

- Although text-level markup is allowed inside PRE, not all tags are supported.
- A P tag is strictly excluded from within PRE, but if a browser encounters one, it should treat it as two new lines.
- Since HTML tags are permitted inside PRE, a text file can not just be inserted into an HTML document by slapping <PRE> and </PRE> around them. The &, < and > characters must first be converted into entities.

P - Paragraph

Appearance:

<P> [</P>]

Attributes:

ALIGN=left|center|right

Contents:

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

May occur in:

ADDRESS, BODY, DIV, CENTER, BLOCKQUOTE, FORM, TD, TH, LI, and DD.

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.

Notes:

- ALIGN=LEFT is the default.
- Some browsers render extra whitespace when multiple empty paragraphs are used in sequence. This is not required by the specs, so it should not be counted on to get vertical whitespace in a 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.

SAMP - Sample text

Appearance:

<SAMP> </SAMP>

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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 merely because text requires a monospaced font; use [TT](#) instead.

SCRIPT - Inline scripts

Appearance:

<SCRIPT> </SCRIPT>

Attributes:

LANGUAGE=string.

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. The scripting language used may be listed in the LANGUAGE attribute.

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, it is a good idea to enclose it within comments.
- If a script uses the ">" or "--" symbols, some browsers may end the comment accidentally.

SELECT - Selection list

Appearance:

<SELECT NAME="string"> </SELECT>

Attributes:

NAME=string, SIZE=n, MULTIPLE

Contents:

OPTION.

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, INPUT, SELECT, TEXTAREA and plain text, 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, a drop-down list will be displayed. If it's more than one, a scrollable list will be displayed. 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 dependent on platform, so do not include "instructions" on how to do this. They may be wrong and can cause a lot of confusion.
- If MULTIPLE is used, the SIZE should be set to more than one. This makes it easier to select more than one item.

SMALL - Smaller text

Appearance:

<SMALL> </SMALL>

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

The **SMALL** tag (as well as [BIG](#)) is new. A browser should draw the enclosed text in a smaller font than current 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, it might be preferable to use `` instead.

STRIKE - Strike-through text

Appearance:

<STRIKE> </STRIKE>

Attributes:

None.

Contents:

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

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, INPUT, SELECT, TEXTAREA and plain text.

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 strike-through text is an absolute requirement, an image should be used.

STRONG - Strongly emphasized text

Appearance:

Attributes:

None.

Contents:

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

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, INPUT, SELECT, TEXTAREA and plain text.

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 merely because the text requires boldface; use B instead.

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.

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, it may be preferable to enclose them in comments.

SUB - Subscript

Appearance:

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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:

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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.

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](#), [TD](#), [TH](#), [LI](#), and [DD](#).

TABLE starts 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.

The BORDER attribute will display a border around the table with a corresponding border width="number". BORDER without a value, defaults to a width of one. The lines between table cells or rows cannot be resized.

The white space inside a table can be increased by using the CELLSPACING attribute to indicate how many pixels there should be between a cell's contents and the border. CELLPADDING indicates how much white space (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:

A test table with merged cells

	Average		other	Misc
	height	weight	category	
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 ruins 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 a table with BORDER=0.

TD - Table data

Appearance:

<TD> [</TD>]

Attributes:

ROWSPAN=n, COLSPAN=n, NOWRAP, ALIGN=left|right|center, VALIGN=top|middle|bottom|baseline, 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](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [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. [BR](#) must then be used 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 these attributes are used, make sure to count correctly else some very weird results may be displayed.

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.

The BASELINE value indicates that all cells in this row should share the same baseline for the first line of text. 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 results.

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 the table easier to read.
- If a table is included inside a table cell, be sure to close all cells and rows. It is usually not required, but some browsers render nested tables incorrectly.
- An empty cell is usually rendered differently than a cell with just whitespace inside it. This especially shows if a border is defined for the table.
- If images are used in a table cell, the WIDTH and HEIGHT attributes in the IMG tag should be specified. 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 a page displays.

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:

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, INPUT, SELECT, TEXTAREA and plain text, 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 they will have to break lines.

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. Hypertext markup may not be used for this default value.

Notes:

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

TEXTFLOW - Replacement for text in applet

Appearance:

[<TEXTFLOW>] [</TEXTFLOW>]

Attributes:

None.

Contents:

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

May occur in:

APPLET.

The **TEXTFLOW** element was introduced mainly to make the APPLET element legal. It should contain the alternative text which should be displayed if the APPLET cannot be executed. Since it is optional, the text may also be put directly inside the APPLET tag instead.

Notes:

- If alternative text is not supplied, the TEXTFLOW tags must be inserted.

TH - Table header

Appearance:

<TH> [</TH>]

Attributes:

ROWSPAN=n, COLSPAN=n, NOWRAP, ALIGN=left|right|center,
VALIGN=top|middle|bottom|baseline, 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](#), [SUP](#), [SUB](#), [EM](#),
[STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [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 the table easier to read.
- If a table is included inside a table cell, be sure to close all cells and rows. It is usually not required, but some browsers render nested tables incorrectly.
- An empty cell is usually rendered differently than a cell with just whitespace inside it. This especially shows if a border is defined for the table.
- If images are used in a table cell, the WIDTH and HEIGHT attributes in the IMG tag should be specified. 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 a page displays.

TITLE - Document title

Appearance:

<TITLE> </TITLE>

Attributes:

None.

Contents:

Plain text.

May occur in:

HEAD.

The **TITLE** tag is the only required tag in an HTML document. It provides the title of the document. This title is usually displayed at the top of the Browser's window, but is 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 useful out of context. It should still be understandable when it is used as a 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 was fixed in later versions.

TR - Table row

Appearance:

<TR> [</TR>]

Attributes:

ALIGN=left|right|center, VALIGN=top|middle|bottom|baseline

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.

The BASELINE value indicates that all cells in this row should share the same baseline for the first line of text.

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 the table easier to read.
- If a table is included inside a table cell, be sure to close all cells and rows. It is usually not required, but some browsers render nested tables incorrectly.

TT - Teletype font

Appearance:

<TT> </TT>

Attributes:

None.

Contents:

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

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, INPUT, SELECT, TEXTAREA and plain text.

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 new lines ignored.

UL - Unordered list

Appearance:

Attributes:

TYPE=disc|square|circle, COMPACT

Contents:

[LI](#).

May occur in:

[BODY](#), [DIV](#), [CENTER](#), [BLOCKQUOTE](#), [FORM](#), [TD](#), [TH](#), [LI](#), and [DD](#) ..

The **UL** element creates 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 numbering is needed, use OL for an ordered list.

The type of bullet can be suggested with the TYPE attribute. There are 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 in a more compact manner. 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 it is used, make sure the list still "works" without it.

U - Underline

Appearance:

<U> </U>

Attributes:

None.

Contents:

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

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, INPUT, SELECT, TEXTAREA and plain text.

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 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, it should be used in those cases.

Notes:

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

VAR - Variable

Appearance:

<VAR> </VAR>

Attributes:

None.

Contents:

[TT](#), [I](#), [B](#), [U](#), [STRIKE](#), [BIG](#), [SMALL](#), [SUP](#), [SUB](#), [EM](#), [STRONG](#), [DFN](#), [CODE](#), [SAMP](#), [KBD](#), [VAR](#), [CITE](#), [A](#), [IMG](#), [FONT](#), [APPLET](#), [BR](#), [SCRIPT](#), [MAP](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) and plain text.

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 mess up the spacing in **VAR** if there is more than one line.

galactus@htmlhelp.com

Send Mail to Galactus!

john@htmlhelp.com

Send Mail to John!

Alta Vista

Alta Vista is a spider type Web indexing service located at <http://www.alta-vista.com> .

Infoseek

Infoseek is a spider type Web indexing service located at: <http://guide.infoseek.com> .

Cougar DTD

The Cougar DTD is available at:

<http://www.w3.org/pub/WWW/MarkUp/Cougar/HTML.dtd>

RFC1942

Available at: <http://ds.internic.net/rfc/rfc1942.txt>

Style Sheet Information

Available at: <http://www.w3.org/pub/WWW/Style/>

Wilbur DTD

Available at:

<http://www.w3.org/pub/WWW/MarkUp/Wilbur/HTML32.dtd>

ISO-9959 Character Set

Available at:

<http://www.htmlhelp.com/reference/charset/>

Standard Generalized Markup Language

Check out the book "The SGML Primer" by Jeffrey Spivak for excellent information on SGML. ISBN# 0-7895-0194-5.

