

HTML 4.0 Reference



HTML 4.0 Reference

HTML 4.0 became a [W3C Recommendation](#) in December of 1997. The new HTML standard provides a number of significant improvements over previous versions of the language while emphasizing the concepts of accessibility and structural markup.

Contents

[What's New in HTML 4.0](#)

A summary of the new features in HTML 4.0 and a look at the key concepts behind the new standard.

[Structure of an HTML 4.0 Document](#)

An explanation of elements, tags, and attributes, and how they are used in an HTML 4.0 document.

[Organizational List of HTML 4.0 Elements](#)

All HTML 4.0 elements organized by their function.

[Alphabetical List of HTML 4.0 Elements](#)

All HTML 4.0 elements listed alphabetically.

[HTML 4.0 Entities](#)

All character entity references in HTML 4.0 along with their numeric character references.

[Offline Versions](#)

Downloadable versions of this reference suitable for offline use.

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

What's New in HTML 4.0



What's New in HTML 4.0

- [New Elements in HTML 4.0](#)
- [Separation of Structure and Presentation](#)
- [Accessibility](#)
- [Internationalization](#)
- [Style Sheets](#)
- [Client-side Scripting](#)
- [Frames](#)
- [Advanced Tables](#)

New Elements in HTML 4.0

The following elements are new in HTML 4.0:

- [ABBR](#) - Abbreviation
- [ACRONYM](#) - Acronym
- [BDO](#) - BiDi override
- [BUTTON](#) - Button
- [COL](#) - Table column
- [COLGROUP](#) - Table column group
- [DEL](#) - Deleted text
- [FIELDSET](#) - Form control group
- [FRAME](#) - Frame
- [FRAMESET](#) - Frameset
- [IFRAME](#) - Inline frame
- [INS](#) - Inserted text
- [LABEL](#) - Form field label
- [LEGEND](#) - Fieldset caption
- [NOFRAMES](#) - Frames alternate content
- [NOSCRIPT](#) - Alternate script content
- [OBJECT](#) - Object
- [OPTGROUP](#) - Option group
- [Q](#) - Short quotation
- [S](#) - Strike-through text
- [SPAN](#) - Generic inline container
- [TBODY](#) - Table body
- [TFOOT](#) - Table foot
- [THEAD](#) - Table head

Separation of Structure and Presentation

By [deprecating](#) many presentational features from [HTML 3.2](#) and adding hooks for style sheets, HTML 4.0

encourages separating a document's structure from its presentation. This concept is key to understanding HTML 4.0.

When authors use HTML to markup a document's structure and [style sheets](#) to suggest the document's presentation, they can more easily achieve the device-independence that helped bring HTML its initial popularity. A document with a rich structure can be presented in many different ways on different media, allowing the document to adjust to new technologies such as phone or in-car aural browsers. The separation of content and presentation also allows authors to change the presentation of an entire site by editing a single style sheet, providing significant advantages in site maintenance.

Accessibility

Many of HTML 4.0's improvements in the area of accessibility follow from its emphasis on the separation of structure and presentation. When HTML is used structurally, a document can adapt to different browsing environments, accommodating large fonts, special colors, speech synthesizers, and Braille tactile feedback devices. This adaptability allows blind, low-vision, colorblind, and cognitively-challenged users access to the Web, opening a door for the world's 600 million disabled people.

HTML 4.0 includes many new elements and attributes aimed at improving the Web's accessibility. The multi-dimensional nature of HTML tables has long posed problems for non-visual browsing, but new attributes on the [TABLE](#), [TH](#), and [TD](#) elements allow table summaries and a more explicit association between a cell and its header information. These attributes give non-visual browsers the ability to render a cell's header information, possibly in an abbreviated form, before giving the cell's content.

New elements in HTML 4.0 also bring accessibility improvements to forms. The new [FIELDSET](#) element allows form controls to be grouped together and the [LEGEND](#) element provides a caption for the group. By grouping related form controls, authors allow those with non-visual browsers to more easily navigate complicated forms. As well, the new [LABEL](#) element associates a text label with a form control so that users can more easily determine what information is required in a given field.

Other accessibility improvements include full image descriptions through the [LONGDESC](#) attribute on the [IMG](#) element, rich alternatives to images and videos through the [OBJECT](#) element, and richer alternatives to image maps through a new content model for the [MAP](#) element.

Internationalization

To allow representation of the world's languages, HTML 4.0 adopts the *Universal Character Set* as its character set. Previous versions of HTML were restricted to ISO-8859-1, a character set that only handled some western European languages. The Universal Character Set is character-by-character equivalent to [Unicode](#) 2.0 and contains characters for almost all of the world's languages.

The [LANG](#) and [DIR](#) attributes are new in HTML 4.0 and apply to almost all elements. These attributes allow authors to specify the language and directionality of text. The [BDO](#) element allows authors to override the bidirectional algorithm used when right-to-left text such as Hebrew is presented.

HTML 4.0 also offers new [entities](#) for easy entry of [mathematical symbols and Greek letters](#) as well as [other special characters](#).

Style Sheets

HTML 4.0 adds new hooks for [style sheets](#), which suggest how a document is presented. The new **ID**, **CLASS**, and **STYLE** attributes allow style information to be attached to specific elements. The **LINK** and **STYLE** elements have new **TYPE** and **MEDIA** attributes for specifying the style sheet language and target media, respectively.

Client-side Scripting

HTML 4.0 embraces client-side scripting through the addition of a number of new attributes. The **SCRIPT** element now includes attributes for specifying the scripting language, embedding an external script, and deferring execution of a script. As well, a number of event attributes have been added to enable execution of a script upon events such as the user clicking an element, pressing a key, moving the mouse over an element, or changing the value of a form control.

The **NOSCRIPT** element, also new in HTML 4.0, provides alternate content for browsers with client-side scripting disabled or not supported.

Frames

The inclusion of [frames](#) in HTML 4.0 gives authors the ability to present multiple documents in one window. The frames model used in HTML 4.0 is not changed from the [flawed frames model](#) originally proposed by [Netscape](#).

Advanced Tables

The simple table model of [HTML 3.2](#) is expanded in HTML 4.0 to include row and column groups, greater flexibility in defining a table's rules, and accessibility improvements. The use of row groups (**THEAD**, **TFOOT**, **TBODY**) allows visual browsers to render static header and footer rows with scrollable body rows, thus improving the readability of large tables.

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

Structure of an HTML 4.0 Document



Structure of an HTML 4.0 Document

- [Elements and Tags](#)
- [Attributes](#)
- [Special Characters](#)
- [Comments](#)
- [A Complete HTML 4.0 Document](#)
- [Validating your HTML](#)

Elements and Tags

Elements are the structures that describe parts of an HTML document. For example, the **P** element represents a *paragraph* while the **EM** element gives *emphasized* content.

An element has three parts: a start tag, content, and an end tag. A *tag* is special text--"markup"--that is delimited by "<" and ">". An end tag includes a "/" after the "<". For example, the **EM** element has a start tag, ****, and an end tag, ****. The start and end tags surround the *content* of the **EM** element:

```
<EM>This is emphasized text</EM>
```

Element names are always case-insensitive, so ****, ****, and **** are all the same.

Elements cannot overlap each other. If the start tag for an **EM** element appears within a **P**, the **EM**'s end tag must also appear within the same **P** element.

Some elements allow the start or end tag to be omitted. For example, the **LI** end tag is always optional since the element's end is implied by the next **LI** element or by the end of the list:

```
<UL>
  <LI>First list item; no end tag
  <LI>Second list item; optional end tag included</LI>
  <LI>Third list item; no end tag
</UL>
```

Some elements have no end tag because they have no content. These elements, such as the **BR** element for line breaks, are represented only by a start tag and are said to be *empty*.

Attributes

An element's *attributes* define various properties for the element. For example, the **IMG** element takes a **SRC** attribute to provide the location of the image and an **ALT** attribute to give alternate text for those not loading images:

```
<IMG SRC="wdglogo.gif" ALT="Web Design Group">
```

An attribute is included in the start tag only--never the end tag--and takes the form **Attribute-name="Attribute-value"**. The attribute value is delimited by single or double quotes. The quotes are optional if the attribute value consists solely of letters in the range A-Z and a-z, digits (0-9), hyphens ("-"), and periods (".").

Attribute names are case-insensitive, but attribute values may be case-sensitive.

Special Characters

Certain characters in HTML are reserved for use as markup and must be escaped to appear literally. The "<" character may be represented with an *entity*, **<**. Similarly, ">" is escaped as **>**, and "&" is escaped as **&**. If an attribute value contains a double quotation mark and is delimited by double quotation marks, then the quote should be escaped as **"**.

Other entities exist for special characters that cannot easily be entered with some keyboards. For example, the copyright symbol ("©") may be represented with the entity **©**. See the [Entities](#) section for a complete list of HTML 4.0 entities.

As an alternative to entities, authors may also use *numeric character references*. Any character may be represented by a numeric character reference based on its "code position" in [Unicode](#). For example, one could use **©** for the copyright symbol or **ا** for the Arabic letter ALEF.

Comments

Comments in HTML have a complicated syntax that can be simplified by following this rule: Begin a comment with "**<!--**", end it with "**-->**", and do not use "**--**" within the comment.

```
<!-- An example comment -->
```

A Complete HTML 4.0 Document

An HTML 4.0 document begins with a **DOCTYPE** declaration that declares the version of HTML to which the document conforms. The **HTML** element follows and contains the **HEAD** and **BODY**. The **HEAD** contains information about the document, such as its title and keywords, while the **BODY** contains the actual content of the document, made up of [block-level elements](#) and [inline elements](#). A basic HTML 4.0 document takes on the following form:


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>The document title</TITLE>
  </HEAD>
  <BODY>
    <H1>Main heading</H1>
    <P>A paragraph.</P>
    <P>Another paragraph.</P>
    <UL>
      <LI>A list item.</LI>
      <LI>Another list item.</LI>
    </UL>
  </BODY>
</HTML>
```

In a [Frameset](#) document, the **FRAMESET** element replaces the **BODY** element.

Validating your HTML

Each HTML document should be *validated* to check for errors such as missing quotation marks (**Oops**), misspelled element or attribute names, and invalid structures. Such errors are not always apparent when viewing a document in a browser since browsers are designed to recover from an author's errors. However, different browsers recover in different ways, sometimes resulting in invisible text on one browser but not on others.

The [W3C HTML Validation Service](#) checks the validity of HTML 4.0 documents.

Note that some programs claim to be validators but really are not. A validator checks a document against a formal *document type definition* (DTD) while other programs such as *lints* warn about valid but unsafe HTML. Both kinds of programs are useful, but validation should never be forgotten. Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

HTML 4.0 Elements



HTML 4.0 Elements

The following is an organizational list of all HTML 4.0 elements. An [alphabetical list](#) is also available.

Top-level Elements

- [HTML](#) - HTML document
- [HEAD](#) - Document head
- [BODY](#) - Document body
- [FRAMESET](#) - Frameset

Head Elements

- [BASE](#) - Document base URI
- [ISINDEX](#) - Input prompt
- [LINK](#) - Document relationship
- [META](#) - Metadata
- [SCRIPT](#) - Client-side script
- [STYLE](#) - Embedded style sheet
- [TITLE](#) - Document title

Generic Block-level Elements

- [ADDRESS](#) - Address
- [BLOCKQUOTE](#) - Block quotation
- [CENTER](#) - Centered block
- [DEL](#) - Deleted text
- [DIV](#) - Generic block-level container
- [H1](#) - Level-one heading
- [H2](#) - Level-two heading
- [H3](#) - Level-three heading
- [H4](#) - Level-four heading
- [H5](#) - Level-five heading
- [H6](#) - Level-six heading
- [HR](#) - Horizontal rule
- [INS](#) - Inserted text
- [ISINDEX](#) - Input prompt
- [NOSCRIPT](#) - Alternate script content
- [P](#) - Paragraph
- [PRE](#) - Preformatted text

Lists

- [DIR](#) - Directory list
- [DL](#) - Definition list
 - [DT](#) - Definition term
 - [DD](#) - Definition description
- [LI](#) - List item
- [MENU](#) - Menu list
- [OL](#) - Ordered list
- [UL](#) - Unordered list

Tables

- [TABLE](#) - Table
 - [CAPTION](#) - Table caption
 - [COLGROUP](#) - Table column group
 - [COL](#) - Table column
 - [THEAD](#) - Table head
 - [TFOOT](#) - Table foot
 - [TBODY](#) - Table body
 - [TR](#) - Table row
 - [TD](#) - Table data cell
 - [TH](#) - Table header cell

Forms

- [FORM](#) - Interactive form
 - [BUTTON](#) - Button
 - [FIELDSET](#) - Form control group
 - [LEGEND](#) - Fieldset caption
 - [INPUT](#) - Form input
 - [LABEL](#) - Form field label
 - [SELECT](#) - Option selector
 - [OPTGROUP](#) - Option group
 - [OPTION](#) - Menu option
 - [TEXTAREA](#) - Multi-line text input

Special Inline Elements

- [A](#) - Anchor
- [APPLET](#) - Java applet
- [BASEFONT](#) - Base font change
- [BDO](#) - BiDi override
- [BR](#) - Line break
- [FONT](#) - Font change
- [IFRAME](#) - Inline frame
- [IMG](#) - Inline image
- [MAP](#) - Image map
 - [AREA](#) - Image map region
- [OBJECT](#) - Object
- [PARAM](#) - Object parameter

- ⌞ [Q](#) - Short quotation
- ⌞ [SCRIPT](#) - Client-side script
- ⌞ [SPAN](#) - Generic inline container
- ⌞ [SUB](#) - Subscript
- ⌞ [SUP](#) - Superscript

Phrase Elements

- ⌞ [ABBR](#) - Abbreviation
- ⌞ [ACRONYM](#) - Acronym
- ⌞ [CITE](#) - Citation
- ⌞ [CODE](#) - Computer code
- ⌞ [DEL](#) - Deleted text
- ⌞ [DFN](#) - Defined term
- ⌞ [EM](#) - Emphasis
- ⌞ [INS](#) - Inserted text
- ⌞ [KBD](#) - Text to be input
- ⌞ [SAMP](#) - Sample output
- ⌞ [STRONG](#) - Strong emphasis
- ⌞ [VAR](#) - Variable

Font Style Elements

- ⌞ [B](#) - Bold text
- ⌞ [BIG](#) - Large text
- ⌞ [I](#) - Italic text
- ⌞ [S](#) - Strike-through text
- ⌞ [SMALL](#) - Small text
- ⌞ [STRIKE](#) - Strike-through text
- ⌞ [TT](#) - Teletype text
- ⌞ [U](#) - Underlined text

Frames

- ⌞ [FRAMESET](#) - Frameset
 - ⌞ [FRAME](#) - Frame
- ⌞ [NOFRAMES](#) - Frames alternate content

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:gerben@weballey.com) <[weballey@usa.net](mailto:gerben@weballey.com)>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

HTML 4.0 Elements



HTML 4.0 Elements

The following is an alphabetical list of all HTML 4.0 elements. An [organizational list](#) is also available.

- ⌞ [A](#) - Anchor
- ⌞ [ABBR](#) - Abbreviation
- ⌞ [ACRONYM](#) - Acronym
- ⌞ [ADDRESS](#) - Address
- ⌞ [APPLET](#) - Java applet
- ⌞ [AREA](#) - Image map region
- ⌞ [B](#) - Bold text
- ⌞ [BASE](#) - Document base URI
- ⌞ [BASEFONT](#) - Base font change
- ⌞ [BDO](#) - BiDi override
- ⌞ [BIG](#) - Large text
- ⌞ [BLOCKQUOTE](#) - Block quotation
- ⌞ [BODY](#) - Document body
- ⌞ [BR](#) - Line break
- ⌞ [BUTTON](#) - Button
- ⌞ [CAPTION](#) - Table caption
- ⌞ [CENTER](#) - Centered block
- ⌞ [CITE](#) - Citation
- ⌞ [CODE](#) - Computer code
- ⌞ [COL](#) - Table column
- ⌞ [COLGROUP](#) - Table column group
- ⌞ [DD](#) - Definition description
- ⌞ [DEL](#) - Deleted text
- ⌞ [DFN](#) - Defined term
- ⌞ [DIR](#) - Directory list
- ⌞ [DIV](#) - Generic block-level container
- ⌞ [DL](#) - Definition list
- ⌞ [DT](#) - Definition term
- ⌞ [EM](#) - Emphasis
- ⌞ [FIELDSET](#) - Form control group
- ⌞ [FONT](#) - Font change
- ⌞ [FORM](#) - Interactive form
- ⌞ [FRAME](#) - Frame
- ⌞ [FRAMESET](#) - Frameset
- ⌞ [H1](#) - Level-one heading
- ⌞ [H2](#) - Level-two heading
- ⌞ [H3](#) - Level-three heading
- ⌞ [H4](#) - Level-four heading
- ⌞ [H5](#) - Level-five heading
- ⌞ [H6](#) - Level-six heading
- ⌞ [HEAD](#) - Document head
- ⌞ [HR](#) - Horizontal rule
- ⌞ [HTML](#) - HTML document

- ⌞ I - Italic text
- ⌞ IFRAME - Inline frame
- ⌞ IMG - Inline image
- ⌞ INPUT - Form input
- ⌞ INS - Inserted text
- ⌞ INDEX - Input prompt
- ⌞ KBD - Text to be input
- ⌞ LABEL - Form field label
- ⌞ LEGEND - Fieldset caption
- ⌞ LI - List item
- ⌞ LINK - Document relationship
- ⌞ MAP - Image map
- ⌞ MENU - Menu list
- ⌞ META - Metadata
- ⌞ NOFRAMES - Frames alternate content
- ⌞ NOSCRIPT - Alternate script content
- ⌞ OBJECT - Object
- ⌞ OL - Ordered list
- ⌞ OPTGROUP - Option group
- ⌞ OPTION - Menu option
- ⌞ P - Paragraph
- ⌞ PARAM - Object parameter
- ⌞ PRE - Preformatted text
- ⌞ Q - Short quotation
- ⌞ S - Strike-through text
- ⌞ SAMP - Sample output
- ⌞ SCRIPT - Client-side script
- ⌞ SELECT - Option selector
- ⌞ SMALL - Small text
- ⌞ SPAN - Generic inline container
- ⌞ STRIKE - Strike-through text
- ⌞ STRONG - Strong emphasis
- ⌞ STYLE - Embedded style sheet
- ⌞ SUB - Subscript
- ⌞ SUP - Superscript
- ⌞ TABLE - Table
- ⌞ TBODY - Table body
- ⌞ TD - Table data cell
- ⌞ TEXTAREA - Multi-line text input
- ⌞ TFEET - Table foot
- ⌞ TH - Table header cell
- ⌞ THEAD - Table head
- ⌞ TITLE - Document title
- ⌞ TR - Table row
- ⌞ TT - Teletype text
- ⌞ U - Underlined text
- ⌞ UL - Unordered list
- ⌞ VAR - Variable

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra <weballey@usa.net>](mailto:weballey@usa.net).
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

HTML 4.0 Entities



Entities

Character entity references, or *entities* for short, provide a method of entering characters that cannot be expressed in the document's character encoding or that cannot easily be entered on a keyboard. Entities are case-sensitive and take the form **&name;**. Examples of entities include **©** for the copyright symbol and **Α** for the Greek capital letter alpha.

In addition to entities, authors can use *numeric character references*. While entities are limited to a subset of [Unicode characters](#), numeric character references can specify any character. Numeric character references may be given in decimal or hexadecimal, though browser support is stronger for decimal references. Decimal references are of the form **&#number;**; while hexadecimal references take the case-insensitive form **&#xnumber;**. Examples of numeric character references include **©** or **©** for the copyright symbol, **Α** or **Α** for the Greek capital letter alpha, and **ا** or **ا** for the Arabic letter ALEF.

Note that hexadecimal character references will cause errors with current validators (except the [WDG HTML Validator](#)) since these references are defined in a recent revision to the SGML standard that is not yet supported by most validators.

The following documents feature tables of the character entity references in HTML 4.0, along with the numeric character reference in decimal and hexadecimal.

- » [Latin-1 Entities](#)
- » [Symbols and Greek Letters](#)
- » [Other Special Characters](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Latin-1 Characters](#) ~ [Symbols and Greek Letters](#) ~ [Other Special Characters](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

ABBR - Abbreviation



ABBR - Abbreviation

Syntax
<ABBR>
..</ABBR
>
Attri
bute
Spe
cific
ation
s

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
[Inlin](#)
[e](#)
[elem](#)
[ents](#)
Cont
aine
d in
[Inlin](#)
[e](#)
[elem](#)
[ents](#)
[bloc](#)
[k-](#)

[level](#)
[elem](#)
[ents](#)

The **ABBR** element is used to markup *abbreviations*. The **TITLE** attribute is useful in conjunction with **ABBR** to give the long form of the abbreviation, allowing visual browsers to provide the long form as a "tooltip". If the short form is a pronounceable word, the **ACRONYM** element should be used instead of **ABBR**.

Examples:

```
⌘ <ABBR TITLE="United Nations">U.N.</ABBR>  
⌘ He weighs 180 <ABBR TITLE=pounds>lbs.</ABBR>  
⌘ <ABBR TITLE="Parti Québécois" LANG=fr-CA>PQ</ABBR>  
⌘ <ACRONYM TITLE="North Atlantic Treaty Organization">NATO</ACRONYM>
```

Some short forms, such as "SQL" and "URL," are pronounced as words by some but pronounced letter-by-letter by others. In such cases, the **ABBR** element should be favored over **ACRONYM**. A style sheet could be used to suggest the aural rendering. For example, one could use

```
<ABBR TITLE="Uniform Resource Locator">URL</ABBR>
```

with the following [CSS2](#) style sheet:

```
abbr[title="Uniform Resource Locator"] { speak: spell-out }
```

More Information

⌘ [ABBR in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

ACRONYM - Acronym



ACRONYM - Acronym

Syntax
<ACRONYM>...</ACRONYM>
Attribute Specifications

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
In
l
i
n
e
e
l
e
m
e
n
t
s
Cont
aine
d in
In
l
i
n
e
e
l
e
m
e
n
t
s
,
b
l
o
c
k

[k-](#)
[level](#)
[elem](#)
[ents](#)

The **ACRONYM** element is used to markup *acronyms*. The **TITLE** attribute is useful in conjunction with **ACRONYM** to give the long form of the acronym, allowing visual browsers to provide the long form as a "tooltip". Examples:

```
«<ACRONYM TITLE="North Atlantic Treaty Organization">NATO</ACRONYM>
«<ACRONYM TITLE="radio detecting and ranging">radar</ACRONYM>
«<ABBR TITLE="Federal Bureau of Investigation">FBI</ABBR>
```

Unlike other kinds of [abbreviations](#), acronyms are pronounceable words, though in some cases the pronunciation is strictly a presentation issue. For example, "SQL" and "URL" are pronounced as words by some people and pronounced letter-by-letter by others. In such cases, authors should use the **ABBR** element, possibly with a style sheet rule specifying the pronunciation for aural rendering.

More Information

«[ACRONYM in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

BDO - BiDi Override



BDO - BiDi Override

Syntax
<BDO>...
</BDO>
Attribute
Specifications

u
D
I
R
=
[
|
t
r
|
r
t
|
]
(
d
i
r
e
c
t
i
o
n
a
l
i
t
y

o
f
t
e
x
t
)

u

L
A
N
G

=

T

E

R

E

G

E

E

C

O

D

E

(

l

a

n

g

u

a

g

e

)

o

f

t

e

x

t

)

u

C

O

R

E

=

A

T

T

R

I

B

U

T

E

[S](#)

Cont
ents
[Inlin
e
elem
ents](#)
Cont
aine
d in
[Inlin
e
elem
ents,
bloc
k-
level
elem
ents](#)

The **BDO** element *overrides the bidirectional algorithm* for the enclosed text. Characters in [Unicode](#) are assigned a directionality, left-to-right or right-to-left, to allow the text to be rendered properly. For example, while English characters are presented left-to-right, Hebrew characters are presented right-to-left.

Unicode defines a *bidirectional algorithm* that must be applied whenever a document contains right-to-left characters. While this algorithm usually gives the proper presentation, occasionally authors must override the bidirectional algorithm and specify the directionality of the text. One such case is when Hebrew characters are stored in *visual order*, where the first character of a word is after the second character. Unicode assumes that the characters are stored in *logical order*, where the first character of a word is before the second character, so the bidirectional algorithm would result in a rendering with the first character incorrectly to the *left* of the second character.

The **BDO** element requires the **DIR** attribute to specify the directionality of the enclosed text. If a document contains Hebrew characters stored in visual order, one should use `<BDO DIR=ltr>text</BDO>` to force the proper presentation for that text.

Authors may alternatively override the bidirectional algorithm using the Unicode character **‭** to force left-to-right directionality or **‮** to force right-to-left directionality. The character **‬** ends the overriding of the algorithm. These characters should not be used in combination with the **DIR** attribute.

More Information

- [BDO in W3C HTML 4.0 Recommendation](#)
- [Introduction to the bidirectional algorithm](#)
- [Unicode](#)
- [Middle Eastern Language Issues](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

BUTTON - Button



BUTTON - Button

Syntax
<BUTTON>
<BUTTON>
Attri
bute
Spe
cific
ation
s

u
N
A
M
E
=
C
D
A
T
A

(
k
e
y

i
n

s
u
b
m
i
t
t
e
d

f
o
r
m
)

u
V
A
L
U
E
=
C
D
A
T
A

(
v
a
l
u
e

i
n

s
u
b
m
i
t
t
e
d

f
o
r
m
)

u
T
Y
P
E
=
[

s
u
b
m
i
t

|

r
e
s
e
t

I
b
u
t
t
o
n

J

(
t
y
p
e

o
f

b
u
t
t
o
n
)

u

D
I
S
A
B
L
E
D

(
d
i
s
a
b
l
e

b
u
t
t
o
n
)

u

A
C
C
E
S

S
K
E
Y
=
C
r
a
r
a
C
t
r

(
s
h
o
r
t
c
u
t

k
e
y
)

»
T
A
B
I
N
D
E
X
=
N
r
a
b
r

(
p
o
s
i
t
i
o
n

i
n

t
a

b
b
i
n
g

o
r
d
e
r
)

“ O N F O C U S
= S
i
c
i
p
i
t
i
t

(e
l
e
m
e
n
t

r
e
c
e
i
v
e
d

f
o
c
u
s
)

“ O N B L U R
=

S
C
i
p

(
e
l
e
m
e
n
t

l
o
s
t

f
o
c
u
s
)

u

c
o
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents

u

l
i
n
e
=
e
l
e

m
e
n
t
s

e
x
c
e
p
t

A

,

I
N
P
U
I

,

S
E
L
E
C
I

,

I
E
X
T
A
R
E
A

,

L
A
B
E
L

,

B
U
T
T
O
N

,

a
n
d

I
E

R
A
M
E

,

B
l
o
c
k
l
e
v
e
l
e
m
e
n
t
s

e
x
c
e
p
t

F
O
R
M

,

I
S
I
N
D
E
X

,

a
n
d

F
I
E
L
D
S
E
T

Cont

ained in
[Block-level elements](#),
[inline elements](#)
except
BUTTON

The **BUTTON** element defines a *submit button*, *reset button*, or *push button*. Authors can also use [INPUT](#) to specify these buttons, but the **BUTTON** element allows richer labels, including images and emphasis. However, **BUTTON** is new in HTML 4.0 and poorly supported among current browsers, so **INPUT** is a more reliable choice at this time.

The **TYPE** attribute of **BUTTON** specifies the kind of button and takes the value **submit** (the default), **reset**, or **button**. The **NAME** and **VALUE** attributes determine the name/value pair sent to the server when a submit button is pushed. These attributes allow authors to provide multiple submit buttons and have the form handler take a different action depending on the submit button used.

Some examples of **BUTTON** follow:

```
«<BUTTON NAME=submit VALUE=modify ACCESSKEY=M>Modify information</BUTTON>  
<BUTTON NAME=submit VALUE=continue ACCESSKEY=C>Continue with application</BUTTON>  
«<BUTTON ACCESSKEY=S>Submit <IMG SRC="checkmark.gif" ALT="&#10004;"></BUTTON>  
<BUTTON TYPE=reset ACCESSKEY=R>Reset <IMG SRC="x.gif" ALT="&#10008;"></BUTTON>  
«<BUTTON TYPE=button ID=toggler ONCLICK="toggle()" ACCESSKEY=H>Hide <strong>non-strict</strong> attributes</BUTTON>
```

The **ACCESSKEY** attribute, used throughout the preceding examples, specifies a single Unicode character as a shortcut key for pressing the button. [Entities](#) (e.g. **´**) may be used as the **ACCESSKEY** value.

The boolean **DISABLED** attribute makes the **BUTTON** element unavailable. The user is unable to push the button, the button cannot receive focus, and the button is skipped when navigating the document by tabbing.

The **TABINDEX** attribute specifies a number between 0 and 32767 to indicate the tabbing order of the button. A **BUTTON** element with **TABINDEX=0** or no **TABINDEX** attribute will be visited after any elements with a positive **TABINDEX**. Among positive **TABINDEX** values, the lower number receives focus first. In the case of a tie, the element appearing first in the HTML document takes precedence.

The **BUTTON** element also takes a number of attributes to specify client-side scripting actions for various events. In addition to the [core events](#) common to most elements, **BUTTON** accepts the following event attributes:

- «**ONFOCUS**, when the element receives focus;
- «**ONBLUR**, when the element loses focus.

More Information

[BUTTON in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

COL - Table Column



COL - Table Column

Syntax
<COL>
Attribute
Specifications

SPAN = N
u
m
b
r
(
n
u
m
b
e
r
o
f
c
o
l
u
m
n
s
)
W
I
D

`TH =
M
U
I
T
I
E
P
D
G
H`

`(width
of
each
column)`

`ALIGN =
[left
|
center
|`

r
i
g
h
t

|

j
u
s
t
i
f
y

|

c
h
a
r

|

(
h
o
r
i
z
o
n
t
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s
)

“
C
H
A
R

|||C
|||R
|||R
|||C
|||C
|||C
|||C

(a
l
i
g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

f
o
r

c
e
l
l
s

)

" C
H
A
R
O
F
F

=

|||I
|||R
|||R
|||C
|||C
|||C

(a
l

i
g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

o
f
f
s
e
t
)

"
V
A
L
L
I
G
N
=
[

t
o
p

|

m
i
d
d
l
e

|

b
o
t
t
o
m

|

b
a

s
e
l
i
n
e

]

(
v
e
r
t
i
c
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s
)

u
c
o
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Emp
ty
Cont

ained in
COL
GR
OU
P
,
TAB
LE

The **COL** element defines attributes common to a *table column*. If used, **COL** must be after the optional **CAPTION** and before the optional **THEAD** in the **TABLE**. Unlike **COLGROUP**, **COL** does not group columns structurally; it merely defines attributes common to all cells in one or more columns.

COL's **SPAN** attribute defines the number of columns that will share the **COL** element's other attributes; the default value is **1**. **COL** may be contained directly in the **TABLE** element or it may be contained within a **COLGROUP**. If **COL** is in a **COLGROUP**, the **COL**'s attributes override those of the **COLGROUP** for the columns spanned by **COL**.

The next example uses **COL** elements within **COLGROUPS** to assign a different **CLASS** to each column:

```
<TABLE SUMMARY="This table gives the character entity reference,  
                decimal character reference, and hexadecimal character  
                reference for 8-bit Latin-1 characters, as well as the  
                rendering of each in your browser.">  
  <COLGROUP CLASS="character-description">  
    <COLGROUP>  
      <COL CLASS="entity">  
        <COL SPAN=2 CLASS="numeric">  
    <COLGROUP>  
      <COL CLASS="entity-rendering">  
        <COL CLASS="decimal-rendering">  
        <COL CLASS="hex-rendering">  
  <THEAD>  
    <TR>  
      <TH SCOPE=col ROWSPAN=2>Character</TH>  
      <TH SCOPE=col ROWSPAN=2>Entity</TH>  
      <TH SCOPE=col ROWSPAN=2>Decimal</TH>  
      <TH SCOPE=col ROWSPAN=2>Hex</TH>  
      <TH SCOPE=colgroup COLSPAN=3>Rendering in Your Browser</TH>  
    </TR>  
    <TR>  
      <TH SCOPE=col>Entity</TH>  
      <TH SCOPE=col>Decimal</TH>  
      <TH SCOPE=col>Hex</TH>  
    </TR>  
  </THEAD>  
  <TBODY>  
    <TR>  
      <TD SCOPE=row>non-breaking space</TD>  
      <TD>&nbsp;</TD>  
      <TD>&#160;</TD>  
      <TD>&#xA0;</TD>  
      <TD>&nbsp;</TD>  
      <TD>&#160;</TD>  
      <TD>&#xA0;</TD>  
    </TR>  
    ...  
  </TBODY>  
</TABLE>
```

COL also takes a number of presentational attributes, many of which cannot be completely replaced by [style sheets](#). Since few browsers support **COL**, authors may wish to specify these attributes on the **TD** or **TH** elements instead.

The **WIDTH** attribute specifies a width for each column spanned by **COL**. The value must be a number in pixels, a percentage of the table width, or a relative length expressed as i^* where i is an integer. A column with **WIDTH="3"** will be allotted three times the width of a column with **WIDTH="1"**. The value **0** is equivalent to the minimum width necessary for the column's contents.

The **ALIGN** attribute specifies the horizontal alignment for each cell in the spanned columns. Possible values are **left**, **center**, **right**, **justify**, and **char**. **ALIGN=char** aligns a cell's contents on the character given in the **CHAR** attribute. The default value for the **CHAR** attribute is the decimal point of the current language--a period in English. The **CHAROFF** attribute specifies the offset to the first occurrence of the alignment character. The attribute's value is a number in pixels or a percentage of the cell's width; **CHAROFF="50%"** centers the alignment character horizontally in a cell.

The **VALIGN** attribute specifies the vertical position of a cell's contents. Possible values are:

- **top**, which positions data at the top of the cell;
- **middle**, the default value, which centers the cell data vertically;
- **bottom**, which positions data at the bottom of the cell;
- **baseline**, which specifies that the first line of each cell in the row with **ALIGN=baseline** should occur on a common baseline.

More Information

• [COL in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

COLGROUP - Table Column Group



COLGROUP - Table Column Group

Syntax
<COLGR
OUP>...<
/COLGR
OUP>
Attri
bute
Spe
cific
ation
s

u
S
P
A
N
=
N
n
m
b
e
r
(
n
u
m
b
e
r
o
f
c
o
l
u
m
n
s
i
n

group)

◦ WIDTH = M

(width

of

each

column)

◦ ALIGN = [l e f

t
l
c
e
n
t
e
r
l
r
i
g
h
t
l
j
u
s
t
i
f
y
l
c
h
a
r
l
(
h
o
r
i
z
o
n
t
a
l
a
l
i
g
n
m
e
n
t
o
f

c
e
l
l
s

i
n

g
r
o
u
p
)

“
C
H
A
R
=
C
h
r
i
s
t
i
a
n
i
t
y
”

(
a
l
i
g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

f
o
r

c
e
l
l
s

)

CHAROFF = I
e
n
g
t
p

(alignment

character

offset)

ALIGN = [

top

|

m
i
d
d
l
e

l

b
o
t
t
o
m

l

b
a
s
e
l
i
n
e

J

(
v
e
r
t
i
c
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s

i
n

g

r
o
u
p
)
u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s
Cont
ents
Zero
or
mor
e
COL
elem
ents
Cont
aine
d in
TAB
LE

The **COLGROUP** element defines a *column group* in a table. If used, **COLGROUP** must be after the optional **CAPTION** and before the optional **THEAD** in the **TABLE**. The structural divisions defined by **COLGROUP** allow authors to easily suggest a presentation for groups of columns through [style sheets](#).

COLGROUP's **SPAN** attribute defines the number of columns in the group; the default value is **1**. A number of other attributes are permitted on **COLGROUP**, and these are shared among the cells of the group. **COLGROUP** may contain **COL** elements that define attributes for the cells of individual columns, overriding attributes defined for the column group. The **SPAN** attribute should not be used if the **COLGROUP** contains any **COL** elements.

The next example features three column groups to structurally divide the table into three parts. The first part is a single column that gives the description of a character. The second part consists of three columns giving different ways of representing the character in HTML. The third part consists of three columns with renderings of the character in the user's browser.

```

<TABLE SUMMARY="This table gives the character entity reference,
                decimal character reference, and hexadecimal character
                reference for 8-bit Latin-1 characters, as well as the
                rendering of each in your browser.">
  <COLGROUP>
  <COLGROUP SPAN=3>
  <COLGROUP SPAN=3>
  <THEAD>
    <TR>
      <TH SCOPE=col ROWSPAN=2>Character</TH>
      <TH SCOPE=col ROWSPAN=2>Entity</TH>
      <TH SCOPE=col ROWSPAN=2>Decimal</TH>
      <TH SCOPE=col ROWSPAN=2>Hex</TH>
      <TH SCOPE=colgroup COLSPAN=3>Rendering in Your Browser</TH>
    </TR>
    <TR>
      <TH SCOPE=col>Entity</TH>
      <TH SCOPE=col>Decimal</TH>
      <TH SCOPE=col>Hex</TH>
    </TR>
  </THEAD>
  <TBODY>
    <TR>
      <TD SCOPE=row>non-breaking space</TD>
      <TD>&nbsp; </TD>
      <TD>&#160;</TD>
      <TD>&#xA0;</TD>
      <TD>&nbsp; </TD>
      <TD>&#160;</TD>
      <TD>&#xA0;</TD>
    </TR>
    . . .
  </TBODY>
</TABLE>

```

In place of the **<COLGROUP SPAN=3>** tag in the preceding example, a **COLGROUP** with three **COL** elements could have been used:

```

<COLGROUP>
  <COL CLASS=entity>
  <COL CLASS=decimal>
  <COL CLASS=hex>
</COLGROUP>

```

Here we have used the **CLASS** attribute to distinguish the individual columns of the group, allowing us to easily suggest different presentations for the columns through [style sheets](#).

COLGROUP also takes a number of presentational attributes, many of which cannot be completely replaced by style sheets. Since few browsers support **COLGROUP**, authors may wish to specify these attributes on the **TD** or **TH** elements instead.

The **WIDTH** attribute specifies a width for each column in the group. The value must be a number in pixels, a percentage of the table width, or a relative length expressed as **i*** where **i** is an integer. A column with **WIDTH="3"** will be allotted three times the width of a column with **WIDTH="1"**. The value **0*** is equivalent to the minimum width necessary for the column's contents.

The **ALIGN** attribute specifies the horizontal alignment for each cell in the column group. Possible values are **left**, **center**, **right**, **justify**, and **char**. **ALIGN=char** aligns a cell's contents on the character given in the **CHAR** attribute. The default value for the **CHAR** attribute is the decimal point of the current language--a period in English. The **CHAROFF** attribute specifies the offset to the first occurrence of the alignment character. The attribute's value is a number in pixels or a percentage of the cell's width; **CHAROFF="50%"** centers the alignment character horizontally in a cell.

The **VALIGN** attribute specifies the vertical position of a cell's contents. Possible values are:

- **top**, which positions data at the top of the cell;
- **middle**, the default value, which centers the cell data vertically;
- **bottom**, which positions data at the bottom of the cell;
- **baseline**, which specifies that the first line of each cell in the row with **ALIGN=baseline** should occur on a common baseline.

More Information

• [COLGROUP in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

DEL - Deleted Text



DEL - Deleted Text

Syntax
...

Attribute
Specifications

"
C
I
T
E
=
U
R
I
(
r
e
a
s
o
n

f
o
r

d
e
l
e
t
i
o
n
)
"
D
A
T

E
T
I
M
E
=
D
e
l
e
t
e
i
n
e
(
d
a
t
e
a
n
d
t
i
m
e
o
f
d
e
l
e
t
i
o
n
)
=
C
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
[Inline elements, block-level elements](#)
Contained in
[Inline elements, block-level elements](#)

The **DEL** element contains content that has been *deleted*. This element is useful in marking changes from one version of a document to the next. Through [style sheets](#), authors can suggest an appropriate rendering, such as not displaying the deleted content or rendering the text with a strike-through style.

DEL may be used as either a [block-level element](#) or an [inline element](#). If used as an inline element (e.g., within a [P](#)), then **DEL** may not contain any block-level elements.

The optional **CITE** attribute of **DEL** gives a URI with information on why the content was deleted. A brief explanation for the deletion can be given with the **TITLE** attribute, which may be rendered as a "tooltip" by some browsers.

The optional **DATETIME** attribute specifies the date and time of the deletion. The value is case-sensitive and of the form **YYYY-MM-DDThh:mm:ssTZD**. See the [values](#) section for a full explanation of this format.

An example follows:

```
<DEL CITE="http://www.w3.org/TR/REC-html40/appendix/changes.html#h-A.1.3"
DATETIME="1997-12-19T00:00:00-05:00" TITLE="XMP is obsolete"><P>The XMP element
contains preformatted text in which markup other than an end tag is treated as literal
text.</P></DEL>
```

Since **DEL** is poorly supported among browsers, authors may wish to use a [font style element](#) such as **STRIKE** ([deprecated](#) in HTML 4.0) to attempt to convey the meaning of **DEL** to non-supporting visual browsers. The previous example could also be marked up as follows:

```
<DEL CITE="http://www.w3.org/TR/REC-html40/appendix/changes.html#h-A.1.3"
DATETIME="1997-12-19T00:00:00-05:00" TITLE="XMP is obsolete"><P><STRIKE>The XMP
element is used for preformatted text in which markup other than an end tag is treated
as literal text.</STRIKE></P></DEL>
```

More Information

» [DEL in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

FIELDSET - Form Control Group



FIELDSET - Form Control Group

Syntax
<FIELDSET>...</FIELDSET>
>
Attribute Specifications

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
A
LEG
END
element
followed
by
zero
or
more

block-level elements and inline elements
Contained in APPLET, BLOCKQUOTE, BODY, CENTER, DD, DEL, DIV, FIELDSET, FORM, M, IFRAME, INS, LI, MAP, NOFRAMES, NOSCRIPT, OPT, OBJECT, TD, TH

The **FIELDSET** element defines a *form control group*. By grouping related form controls, authors can divide a form into smaller, more manageable parts, improving the usability disaster that can strike when confronting users with too many form controls. The grouping provided by **FIELDSET** also helps the accessibility of forms to those using aural browsers by allowing these users to more easily orient themselves when filling in a large form.

While **FIELDSET** is not widely supported by current browsers, it can be used safely by explicitly closing any preceding **P** element with `</P>` or by including an empty **P** prior to the **FIELDSET**. This causes non-supporting browsers to infer the start of a block-level element even though they ignore the block-level **FIELDSET** element.

The content of a **FIELDSET** element must begin with a **LEGEND** to provide a caption for the group of controls. Following the **LEGEND**, **FIELDSET** may contain any [inline](#) or [block-level](#) element, including another **FIELDSET**.

An example follows:

<FORM METHOD=post ACTION="/cgi-bin/order.cgi">

<FIELDSET>

<LEGEND ACCESSKEY=I>Contact Information</LEGEND>

<TABLE>

<TR>

<TD>

<LABEL FOR=name ACCESSKEY=N>Name:</LABEL>

</TD>

<TD>

<INPUT TYPE=text NAME=name ID=name>

</TD>

</TR>

<TR>

<TD>

<LABEL FOR=email ACCESSKEY=E>E-mail Address:</LABEL>

</TD>

<TD>

<INPUT TYPE=text NAME=email ID=email>

</TD>

</TR>

<TR>

<TD>

<LABEL FOR=addr ACCESSKEY=A>Mailing Address:</LABEL>

</TD>

<TD>

<TEXTAREA NAME=address ID=addr ROWS=4 COLS=40></TEXTAREA>

</TD>

</TR>

</TABLE>

</FIELDSET>

<FIELDSET>

<LEGEND ACCESSKEY=O>Ordering Information</LEGEND>

<P>Please select the product(s) that you wish to order:</P>

<P>

<LABEL ACCESSKEY=3>

<INPUT TYPE=checkbox NAME=products VALUE="HTML 3.2 Reference">

HTML 3.2 Reference

</LABEL>

<LABEL ACCESSKEY=4>

<INPUT TYPE=checkbox NAME=products VALUE="HTML 4.0 Reference">

HTML 4.0 Reference

</LABEL>

<LABEL ACCESSKEY=S>

<INPUT TYPE=checkbox NAME=products VALUE="CSS Guide">

Cascading Style Sheets
Guide

</LABEL>

</P>

</FIELDSET>

<FIELDSET>

<LEGEND ACCESSKEY=C>Credit Card Information</LEGEND>

<P>

<LABEL ACCESSKEY=V>

```

    <INPUT TYPE=radio NAME=card VALUE=visa> Visa
  </LABEL>
  <LABEL ACCESSKEY=M>
    <INPUT TYPE=radio NAME=card VALUE=mc> MasterCard
  </LABEL>
  <BR>
  <LABEL ACCESSKEY=u>
    Number: <INPUT TYPE=text NAME=number>
  </LABEL>
  <BR>
  <LABEL ACCESSKEY=E>
    Expiry: <INPUT TYPE=text NAME=expiry>
  </LABEL>
</P>
</FIELDSET>

<P>
  <INPUT TYPE=submit VALUE="Submit order">
  <INPUT TYPE=reset VALUE="Clear order form">
</P>

</FORM>
```

More Information

u [FIELDSET in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

FRAME - Frame



FRAME - Frame

Syntax
<FRAME
>
Attribute
Specifications

NAME = CDATA
(name of frame)
SRC = DIRECTORIE

(c o n t e n t o f f r a m e)
“ L O N G D E S C = D A I
(l o n g d e s c r i p t i o n o f f r a m e)
“

F
R
A
M
E
B
O
R
D
E
R
=
[
1
|
0
]
(
f
r
a
m
e
b
o
r
d
e
r
)
M
A
R
G
I
N
W
I
D
T
H
=
1
1
1
1
1
(
m
a
r
g

i
n
w
i
d
t
h
)

“
M
A
R
G
I
N
H
E
I
G
H
T
=
P
i
X
E
S

(
m
a
r
g
i
n
h
e
i
g
h
t
)

“
N
O
R
E
S
I
Z
E

(
d
i
s
a

l
l
o
w

f
r
a
m
e

r
e
s
i
z
i
n
g
)
"
S
C
R
O
L
L
I
N
G
=
[
y
e
s

|
n
o

|
a
u
t
o

]
(
a
b
i
l
i
t
y

t

o
s
c
r
o
l
l
)
u
c
o
r
e
=
a
t
t
r
i
b
u
t
e
s
Cont
ents
Emp
ty
Cont
aine
d in
FRA
MES
ET

The **FRAME** element defines a *frame*--a rectangular subspace within a [Frameset](#) document. Each **FRAME** must be contained within a [FRAMESET](#) that defines the dimensions of the frame.

The **SRC** attribute provides the URI of the frame's content, which is typically an HTML document. If the frame's content is an image, video, or similar object, and if the object cannot be described adequately using the [TITLE](#) attribute of **FRAME**, then authors should use the **LONGDESC** attribute to provide the URI of a full HTML description of the object.

For better accessibility to disabled users and better indexing with search engines, authors should not use an image or similar object as the content of a frame. Rather, the object should be embedded within an HTML document to allow the indexing of keywords and easier provision of alternate content.

The **NAME** attribute gives a name to the frame for use with the **TARGET** attribute of the [A](#), [AREA](#), [BASE](#), [FORM](#), and [LINK](#) elements. The **NAME** attribute value must begin with a character in the range A-Z or a-z.

The **NAME** should be human-readable and based on the content of the frame since non-windows browsers may use the **NAME** as a title for presenting a list of frames to the user. For example, **NAME=left** would be inappropriate since it says nothing about the content while **NAME=nav** would be inappropriate since it is not very human-readable. More

suitable would be **NAME=Content** and **NAME=Navigation**. The **TITLE** attribute can also be used to provide a slightly longer title for the frame, though this is not widely supported by current browsers.

An example follows:

```
<FRAMESET ROWS="*,100">
  <FRAMESET COLS="40%,*">
    <FRAME NAME="Menu" SRC="nav.html" TITLE="Menu">
    <FRAME NAME="Content" SRC="main.html" TITLE="Content">
  </FRAMESET>
  <FRAME NAME="Ad" SRC="ad.html" TITLE="Advertisement">
</NOFRAMES>
<BODY>
  <H1>Table of Contents</H1>
  <UL>
    <LI>
      <A HREF="reference/html40/">HTML 4.0 Reference</A>
    </LI>
    <LI>
      <A HREF="reference/wilbur/">HTML 3.2 Reference</A>
    </LI>
    <LI>
      <A HREF="reference/css/">CSS Guide</A>
    </LI>
  </UL>
  <P>
    <IMG SRC="ad.gif" ALT="Ad: Does your bank charge too much?">
  </P>
</BODY>
</NOFRAMES>
</FRAMESET>
```

The **FRAME** element also accepts a number of attributes to specify the presentation on visual browsers. [Style sheets](#) provide a more flexible method of defining the presentation of frames, but the element's presentational attributes are more widely supported.

The **FRAMEBORDER** attribute specifies whether or not the frame has a visible border. The default value, **1**, tells the browser to draw a border between the frame and all adjoining frames. The value **0** indicates that no border should be drawn, though borders from other frames will override this.

To fully remove the border, some browsers also require the use of other, non-standard attributes. See [How do I remove the border around frames?](#) for more details.

Note that removing the border of a frame takes away the user's ability to resize the frame on most browsers.

The **MARGINWIDTH** and **MARGINHEIGHT** attributes define the number of pixels to use as the left/right margins and top/bottom margins, respectively, within the frame. The value must be greater than one pixel.

The boolean **NORESIZE** attribute prevents the user from resizing the frame. This attribute should never be used in a user-friendly Web site.

The **SCROLLING** attribute specifies whether scrollbars are provided for the frame. The default value, **auto**, generates scrollbars only when necessary. The value **yes** gives scrollbars at all times, and the value **no** suppresses scrollbars--even when they are needed to see all the content. The value **no** should never be used.

More Information

[FRAME in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

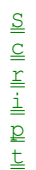
C O L S =



(c o l u m n

l e n g t h s)

= O N L O A D



(a l l f r a m e s

h

a
v
e

b
e
n

l
o
a
d
e
d

)

“
O
N
L
I
N
E

L
O
A
D
=



(
a
l
l
f
r
a
m
e
s

h
a
v
e

b
e
n

r
e
m
o
v
e
d

)
u
c
o
r
e
=
a
t
t
r
i
b
u
t
e
s
Cont
ents
One
or
mor
e
FRA
MES
ET
and
FRA
ME
elem
ents,
as
well
as
an
optio
nal
NOF
RAM
ES
Cont
aine
d in
HTM
L

The **FRAMESET** element is a *frame container* for dividing a window into rectangular subspaces called *frames*. In a [Frameset](#) document, the outermost **FRAMESET** element takes the place of **BODY** and immediately follows the **HEAD**.

The **FRAMESET** element contains one or more **FRAMESET** or **FRAME** elements, along with an optional **NOFRAMES** element to provide alternate content for browsers that do not support frames or have frames disabled. A meaningful **NOFRAMES** element should always be provided and should at the very least contain links to the main frame or frames.

The **ROWS** and **COLS** attributes define the dimensions of each frame in the set. Each attribute takes a comma-

separated list of lengths, specified in pixels, as a percentage, or as a relative length. A relative length is expressed as i^* where i is an integer. For example, a frameset defined with **ROWS="3*,*"** (* is equivalent to 1*) will have its first row allotted three times the height of the second row.

The values specified for the **ROWS** attribute give the height of each row, from top to bottom. The **COLS** attribute gives the width of each column from left to right. If **ROWS** or **COLS** is omitted, the implied value for the attribute is **100%**. If both attributes are specified, a grid is defined and filled left-to-right then top-to-bottom.

The following example sets up a grid with two rows and three columns:

```
<FRAMESET ROWS="70%,30%" COLS="33%,33%,34%">
  <FRAME NAME="Photo1" SRC="Row1_Column1.html">
  <FRAME NAME="Photo2" SRC="Row1_Column2.html">
  <FRAME NAME="Photo3" SRC="Row1_Column3.html">
  <FRAME NAME="Caption1" SRC="Row2_Column1.html">
  <FRAME NAME="Caption2" SRC="Row2_Column2.html">
  <FRAME NAME="Caption3" SRC="Row2_Column3.html">
  <NOFRAMES>
  <BODY>
    <H1>Table of Contents</H1>
    <UL>
      <LI>
        <A HREF="Row1_Column1.html">Photo 1</A>
        (<A HREF="Row2_Column1.html">Caption</A>)
      </LI>
      <LI>
        <A HREF="Row1_Column2.html">Photo 2</A>
        (<A HREF="Row2_Column2.html">Caption</A>)
      </LI>
      <LI>
        <A HREF="Row1_Column3.html">Photo 3</A>
        (<A HREF="Row2_Column3.html">Caption</A>)
      </LI>
    </UL>
  </BODY>
</NOFRAMES>
</FRAMESET>
```

The next example features nested **FRAMESET** elements to define two frames in the first row and one frame in the second row:

```

<FRAMESET ROWS="*,100">
  <FRAMESET COLS="40%,*">
    <FRAME NAME="Menu" SRC="nav.html" TITLE="Menu">
    <FRAME NAME="Content" SRC="main.html" TITLE="Content">
  </FRAMESET>
  <FRAME NAME="Ad" SRC="ad.html" TITLE="Advertisement">
</NOFRAMES>
<BODY>
  <H1>Table of Contents</H1>
  <UL>
    <LI>
      <A HREF="reference/html40/">HTML 4.0 Reference</A>
    </LI>
    <LI>
      <A HREF="reference/wilbur/">HTML 3.2 Reference</A>
    </LI>
    <LI>
      <A HREF="reference/css/">CSS Guide</A>
    </LI>
  </UL>
  <P>
    <IMG SRC="ad.gif" ALT="Ad: Does your bank charge too much?">
  </P>
</BODY>
</NOFRAMES>
</FRAMESET>

```

When pixel lengths are used, they should always be combined with a relative length to handle various window sizes. Pixel lengths should only be used when the frame consists primarily of images or other objects with a fixed size in pixels. Due to their ability to adapt to different window sizes, percentages and relative lengths are generally preferred.

The **FRAMESET** element also accepts **ONLOAD** and **ONUNLOAD** attributes to specify client-side scripting actions to perform when the frames have all been loaded or removed.

More Information

• [FRAMESET in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

IFRAME - Inline Frame



IFRAME - Inline Frame

Syntax
<IFRAME
>...</IFR
AME>
Attri
bute
Spe
cific
ation
s

u
S
R
C
=
U
R
I
(
U
R
L
o
f
f
r
a
m
e
c
o
n
t
e
n
t
)
u
N
A

M
E
=
C
D
A
T
A

(
n
a
m
e

o
f

f
r
a
m
e
)

"
L
O
N
G
D
E
S
C
=
U
R
I

(
l
i
n
k

t
o

l
o
n
g

d
e
s
c
r
i
p
t
i

o
n
)

“
W
I
D
T
H
=
I
e
n
g
t
p

(
f
r
a
m
e

w
i
d
t
h
)

“
H
E
I
G
H
T
=
I
e
n
g
t
p

(
f
r
a
m
e

h
e
i
g
h
t
)

“

A
L
I
G
N
=
[
 t
 o
 p
 |
 m
 i
 d
 d
 l
 e
 |
 b
 o
 t
 t
 o
 m
 |
 l
 e
 f
 t
 |
 r
 i
 g
 h
 t
 |
 (
 f
 r
 a
 m
 e
 a
 l
 i
 g
 n
 m
 e
 n
]

t)
" F R A M E B O R D E R = [1 | 0]
(f r a m e b o r d e r)
" M A R G I N W I D T H = p
i
x
e
i
s
(

m
a
r
g
i
n
w
i
d
t
h
)

«
M
A
R
G
I
N
H
E
I
G
H
T
=
P
i
X
I
S

(
m
a
r
g
i
n
h
e
i
g
h
t
)

«
S
C
R
O
L
L
I
N
G
=
»

[
y
e
s

|
n
o

|
a
u
t
o

]
(
a
b
i
l
i
t
y

t
o

s
c
r
o
l
l
)
u
C
o
r
e
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem

[ents](#),
[bloc](#)
[k-](#)
[level](#)
[elem](#)
[ents](#)
Cont
aine
d in
[Inlin](#)
[e](#)
[elem](#)
[ents](#),
[bloc](#)
[k-](#)
[level](#)
[elem](#)
[ents](#)

The **IFRAME** element defines an *inline frame* for the inclusion of external objects including other HTML documents. **IFRAME** provides a subset of the functionality of **OBJECT**; the only advantage to **IFRAME** is that an inline frame can act as a [target](#) for other links. **OBJECT** is more widely supported than **IFRAME**, and, unlike **IFRAME**, **OBJECT** is included in [HTML 4.0 Strict](#).

IFRAME's **SRC** attribute provides the location of the frame content--typically an HTML document. The optional **NAME** attribute specifies the name of the inline frame, allowing links to [target](#) the frame.

The content of the **IFRAME** element is used as an alternative for browsers that are not configured to show or do not support inline frames. The content may consist of [inline](#) or [block-level](#) elements, though any block-level elements must be allowed inside the containing element of **IFRAME**. For example, an **IFRAME** within an [H1](#) cannot contain an [H2](#), but an **IFRAME** within a [DIV](#) can contain any block-level elements.

The **LONGDESC** attribute gives the URI of a long description of the frame's contents. This is particularly useful for full descriptions of embedded objects. Note that **LONGDESC** *describes* the frame content while the content of the **IFRAME** element acts as a *replacement* when the external resource cannot be inlined.

An example follows:

```
<IFRAME SRC="recipe.html" TITLE="The Famous Recipe">  
<!-- Alternate content for non-supporting browsers -->  
<H2>The Famous Recipe</H2>  
<H3>Ingredients</H3>  
...  
</IFRAME>
```

The **WIDTH** and **HEIGHT** attributes specify the dimensions of the inline frame in pixels or as a percentage of the available space. The **FRAMEBORDER** attribute specifies whether or not a border should be drawn. The default value of **1** results in a border while a value of **0** suppresses the border. [Style sheets](#) allow greater flexibility in suggesting the border presentation.

The **ALIGN** attribute specifies the alignment of the inline frame. The values **top**, **middle**, and **bottom** specify the frame's position with respect to surrounding content on its left and right. **ALIGN=middle** aligns the center of the frame with the current baseline. To center the frame horizontally on the page, place the frame in a centered block, e.g.,

```
<P ALIGN=center><IFRAME SRC="foo.html" WIDTH=300 HEIGHT=100></IFRAME></P>
```

The other **ALIGN** values, **left** and **right**, specify a *floating* frame; the frame is placed at the left or right margin and content flows around it. To place content below the frame, use **<BR CLEAR=left|right|all>** as appropriate.

The [vertical-align](#) and [float](#) properties of [Cascading Style Sheets](#) provide more flexible methods of aligning inline frames.

The **MARGINWIDTH** and **MARGINHEIGHT** attributes define the number of pixels to use as the left/right margins and top/bottom margins, respectively, within the inline frame. The value must be greater than one pixel.

The **SCROLLING** attribute specifies whether scrollbars are provided for the inline frame. The default value, **auto**, generates scrollbars only when necessary. The value **yes** gives scrollbars at all times, and the value **no** suppresses scrollbars—even when they are needed to see all the content. The value **no** should never be used.

More Information

• [IFRAME in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

INS - Inserted Text



INS - Inserted Text

Syntax
<INS>...<
/INS>
Attribute
Specifications

u

C

I

T

E

=

U

R

I

(

r

e

a

s

o

n

f

o

r

i

n

s

e

r

t

i

o

n

)

u

D

A

T
E
T
I
M
E
=
D
a
t
e
t
i
m
e
(
d
a
t
e
a
n
d
t
i
m
e
o
f
i
n
s
e
r
t
i
o
n
)
=
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e

S

Cont
ents
Inlin
e
elem
ents,
bloc
k-
level
elem
ents
Cont
aine
d in
Inlin
e
elem
ents,
bloc
k-
level
elem
ents

The **INS** element contains content that has been *inserted*. This element is useful in marking changes from one version of a document to the next. Through [style sheets](#), authors can suggest an appropriate rendering, such as rendering the inserted content in italics, a different color, or a different voice.

INS may be used as either a [block-level element](#) or an [inline element](#). If used as an inline element (e.g., within a **P**), then **INS** may not contain any block-level elements.

The optional **CITE** attribute of **INS** gives a URI with information on why the content was inserted. A brief explanation for the insertion can be given with the **TITLE** attribute, which may be rendered as a "tooltip" by some browsers.

The optional **DATETIME** attribute specifies the date and time of the insertion. The value is case-sensitive and of the form **YYYY-MM-DDThh:mm:ssTZD**. See the [values](#) section for a full explanation of this format.

An example follows:

```
<P>The CENTER element defines a block whose contents are centered horizontally on  
visual browsers. <INS CITE="http://www.w3.org/TR/REC-html40/appendix/changes.html#h-  
A.1.2" DATETIME="1997-12-19T00:00:00-05:00">Note that CENTER is deprecated in  
HTML 4.0.</INS></P>
```

More Information

• [INS in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

LABEL - Form Field Label



LABEL - Form Field Label

Syntax
<LABEL
>...</LAB
EL>
Attri
bute
Spe
cific
ation
s

u
F
O
R
=
I
D
R
E
F

(
a
s
s
o
c
i
a
t
e
d

f
o
r
m

f
i
e
l
d
)

u

A
C
C
E
S
S
K
E
Y

=

[C](#)
[r](#)
[i](#)
[c](#)
[l](#)
[e](#)
[r](#)

(
s
h
o
r
t
c
u
t

k
e
y
)

u

O
N
F
O
C
C
U
S

=

[l](#)
[e](#)
[l](#)
[e](#)
[m](#)
[e](#)
[n](#)
[t](#)

(
e
l
e
m
e
n
t

r

e
c
e
i
v
e
d

f
o
c
u
s
)

“
O
N
B
L
U
R
=

s
s
r
i
t
t
t

(
e
l
e
m
e
n
t

l
o
s
t

f
o
c
u
s
)

“
c
o
m
m
o
n
=
a
t
t

r i b u t e s

Cont
ents
Inlin
e
elem
ents
exce
pt
LAB
EL
Cont
aine
d in
Bloc
k-
level
elem
ents,
inlin
e
elem
ents
exce
pt
BUT
TON

The **LABEL** element *associates a label with a form control*. By associating labels with form controls, authors give important hints to users of speech browsers while also allowing visual browsers to duplicate common GUI features (e.g., the ability to click on a text label to select a radio button or checkbox).

Each **LABEL** element is associated with exactly one form control. The element's content is the label of the form control and may include inline elements such as **IMG** and **STRONG**.

The **FOR** attribute explicitly specifies the control associated with the **LABEL**. The value of the **FOR** attribute must match the value of the associated form control's **ID** attribute. In the absence of the **FOR** attribute, the **LABEL** must contain the associated form control. This method of implicit association is convenient in many cases, but not an option when the form control and its label are in different table cells, paragraphs, or divisions. The following example illustrates both explicit and implicit label associations:

```

<TABLE>
  <TR>
    <TD>
      <LABEL FOR=user ACCESSKEY=U>User</LABEL>
    </TD>
    <TD>
      <SELECT NAME=user ID=user>
        <OPTION>Jean</OPTION>
        <OPTION>Kim</OPTION>
        <OPTION>Brian</OPTION>
      </SELECT>
    </TD>
  </TR>
  <TR>
    <TD><LABEL FOR=passwd ACCESSKEY=P>Password</LABEL></TD>
    <TD><INPUT TYPE=password NAME=password ID=passwd></TD>
  </TR>
</TABLE>
<P>
  <LABEL ACCESSKEY=S>
    <INPUT TYPE=checkbox NAME=save VALUE=yes>
    Save user name and password in a cookie
  </LABEL>
</P>
<P>
  <LABEL ACCESSKEY=C>
    Comments to post:
    <TEXTAREA NAME=comments ROWS=8 COLS=50></TEXTAREA>
  </LABEL>
</P>

```

The **ACCESSKEY** attribute, used throughout the preceding example, specifies a single Unicode character as a shortcut key for giving focus to the **LABEL**, which passes the focus on to the associated form control. [Entities](#) (e.g. **´**;) may be used as the **ACCESSKEY** value.

The **LABEL** element also takes a number of attributes to specify client-side scripting actions for various events. In addition to the [core events](#) common to most elements, **LABEL** accepts the following event attributes:

- **ONFOCUS**, when the element receives focus;
- **ONBLUR**, when the element loses focus.

More Information

- [LABEL in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)



This help file is created by [Gerben Hoekstra <weballey@usa.net>](mailto:weballey@usa.net).
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

LEGEND - Fieldset Caption



LEGEND - Fieldset Caption

Syntax
<LEGEN
D>...</LE
GEND>
Attri
bute
Spe
cific
ation
s

u

A
C
C
E
S
S
K
E
Y

=

[C](#)
[h](#)
[i](#)
[l](#)
[l](#)
[l](#)
[l](#)
[l](#)
[l](#)
[l](#)
[l](#)
[l](#)

(
s
h
o
r
t
c
u
t

k
e
y
)

u
A
L
I
G
N
=
[
t
o
p
|
b
o
t
t
o
m
|
l
e
f
t
|
r
i
g
h
t
]
(
a
l
i
g
n
m
e
n
t
r
e
l
a
t
i
v
e
t
o

f
i
e
l
d
s
e
t
)

u

C
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
FIEL
DSE
T

The **LEGEND** element defines a *caption* for form controls grouped by the **FIELDSET** element. The **LEGEND** element must be at the start of a **FIELDSET**, before any other elements.

While the **LEGEND** element is not widely supported by current browsers, it can still be used safely if a block-level element immediately follows the **LEGEND**. Combined with careful use of **FIELDSET**, this will cause non-supporting browsers to render the caption as its own paragraph. Elements such as **STRONG**, **B**, and **BIG** could also be used to help express the meaning of **LEGEND** to non-supporting browsers.

An example follows:

```
<FIELDSET>
<LEGEND ACCESSKEY=C>Credit Card Information</LEGEND>
<P>
  <LABEL ACCESSKEY=V>
    <INPUT TYPE=radio NAME=card VALUE=visa> Visa
  </LABEL>
  <LABEL ACCESSKEY=M>
    <INPUT TYPE=radio NAME=card VALUE=mc> MasterCard
  </LABEL>
  <BR>
  <LABEL ACCESSKEY=N>
    Number: <INPUT TYPE=text NAME=number>
  </LABEL>
  <BR>
  <LABEL ACCESSKEY=E>
    Expiry: <INPUT TYPE=text NAME=expiry>
  </LABEL>
</P>
</FIELDSET>
```

LEGEND's ACCESSKEY attribute, used throughout the preceding example, specifies a single Unicode character as a shortcut key for giving focus to the **LEGEND**, allowing the user to quickly jump to a group of form controls. [Entities](#) (e.g. **´**;) may be used as the **ACCESSKEY** value.

The [deprecated ALIGN](#) attribute of **LEGEND** suggests where the caption should be positioned relative to the **FIELDSET** on visual browsers. Possible values are **top**, **bottom**, **left**, and **right**. While **ALIGN** is deprecated, no alternative currently exists in [Cascading Style Sheets](#).

More Information

• [LEGEND in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

NOFRAMES - Frames Alternate Content



NOFRAMES - Frames Alternate Content

Syntax
<NOFRAMES>...</NOFRAMES>
Attribute Specifications

[c](#)
[o](#)
[m](#)
[m](#)
[o](#)
[n](#)
=
[a](#)
[t](#)
[t](#)
[r](#)
[i](#)
[b](#)
[u](#)
[t](#)
[e](#)
[s](#)

Contents

[l](#)
[n](#)
[H](#)
[I](#)
[M](#)
[L](#)
=
[4](#)
[.](#)

O = T r a n s i t i o n a ! : . i n ! i n e = e l e m e n t s . b l o c k = l e v e l e m e n t s . i n H T M L =

4

Q

F

r

a

m

e

s

e

t

.

o

n

e

B

O

D

Y

e

l

e

m

e

n

t

t

h

a

t

m

u

s

t

n

o

t

c

o

n

t

a

i

n

a

n

y

N

O

F

R

A

M
E
S

e
l
e
m
e
n
t
s

Cont
aine
d in
[APP](#)
[LET](#),
[BLO](#)
[CKQ](#)
[UOT](#)
[E](#),
[BOD](#)
[Y](#),
[BUT](#)
[TON](#)
,
[CEN](#)
[TER](#)
,
[DD](#),
[DEL](#),
[DIV](#),
[FIEL](#)
[DSE](#)
[I](#),
[FOR](#)
[M](#),
[FRA](#)
[MES](#)
[ET](#),
[IFR](#)
[AME](#)
,
[INS](#),
[LI](#),
[MAP](#)
,
[NOS](#)
[CRI](#)
[PT](#),
[OBJ](#)
[ECT](#)
, [TD](#),
[TH](#)

The **NOFRAMES** element contains content that should only be rendered when *frames are not displayed*. **NOFRAMES** is typically used in a [Frameset](#) document to provide alternate content for browsers that do not support frames or have frames disabled.

When used within a **FRAMESET**, **NOFRAMES** must contain a **BODY** element. There must not be any **NOFRAMES** elements contained within this **BODY** element.

A meaningful **NOFRAMES** element should always be provided in a Frameset document and should at the very least contain links to the main frame or frames. **NOFRAMES** should not contain a message telling the user to upgrade his or her browser. Some browsers support frames but allow the user to disable them.

Various methods of automatically creating non-frames content exist. See [Frames design guidelines: automatic no-frames content](#) for more details.

An example follows:

```
<FRAMESET ROWS="*,100">
  <FRAMESET COLS="40%,*">
    <FRAME NAME="Menu" SRC="nav.html" TITLE="Menu">
    <FRAME NAME="Content" SRC="main.html" TITLE="Content">
  </FRAMESET>
  <FRAME NAME="Ad" SRC="ad.html" TITLE="Advertisement">
  <NOFRAMES>
    <BODY>
      <H1>Table of Contents</H1>
      <UL>
        <LI>
          <A HREF="reference/html40/">HTML 4.0 Reference</A>
        </LI>
        <LI>
          <A HREF="reference/wilbur/">HTML 3.2 Reference</A>
        </LI>
        <LI>
          <A HREF="reference/css/">CSS Guide</A>
        </LI>
      </UL>
      <P>
        <IMG SRC="ad.gif" ALT="Ad: Does your bank charge too much?">
      </P>
    </BODY>
  </NOFRAMES>
</FRAMESET>
```

In [HTML 4.0 Transitional](#), the **NOFRAMES** element is also permitted within most [block-level elements](#). This allows authors to include content, such as navigation aids, that should only be displayed if the document is not being viewed within a frameset. Such use helps to ensure that a frame could stand on its own if bookmarked or accessed through a search engine while not burdening the frames user with duplicate content. However, most browsers do not support this use of **NOFRAMES** and will always display the content.

More Information

• [NOFRAMES in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

NOSCRIPT - Non-script Content



NOSCRIPT - Non-script Content

Syntax
<NOSCRIPT>...</NOSCRIPT>
Attribute Specifications

[Common Attributes](#)

Contents
[Block-level elements, inline elements](#)
Contained in

APP
LET,
BLO
CKQ
UOT
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DD,
DEL,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,
TH

The **NOSCRIPT** element provides *alternate content for a client-side script* that was not executed. A script will fail to execute if the browser does not support the scripting language or if the user has disabled client-side scripting.

NOSCRIPT should follow the **SCRIPT** element for which it provides alternate content.

Note that most browsers will fail to render the contents of **NOSCRIPT** if the scripting language is not supported and the user has enabled client-side scripting. Most browsers will only render **NOSCRIPT** when the user has disabled client-side scripting.

Also note that Netscape Navigator 2.x supports JavaScript 1.0 but still renders all **NOSCRIPT** content.

Since client-side scripts usually provide dynamic interactivity that cannot be replaced by static content, the **NOSCRIPT** element is generally not useful. Authors should try to use client-side scripts as optional enhancements that are not integral components of the Web page. In the case of form validation, any error checking done by the client-side script should be repeated by the CGI script or Java servlet that handles the submission at the server.

More Information

[NOSCRIPT in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

OBJECT - Embedded Object



OBJECT - Embedded Object

Syntax
<OBJECT>...</OBJECT>
Attribute Specifications

DATA = URI
(object data)
CLASSID = URI

I

(l o c a t i o n

o f

i m p l e m e n t a t i o n)

u A R C H I V E =

C

D

A

T

A

(a r c h i v e

f i l e s)

u

C
O
D
E
B
A
S
E
=
U
R
I

(
b
a
s
e

U
R
I

f
o
r

C
L
A
S
S
I
D
,

D
A
T
A
,

A
R
C
H
I
V
E
)

u

W
I
D
T
H
=
I
e
n
g

t
h

(o b j e c t

w i d t h)

u

H E I G H T =

L
e
n
g
t
h

(o b j e c t

h e i g h t)

u

N A M E =

C
D
A
T
A

(

n
a
m
e

f
o
r

f
o
r
m

s
u
b
m
i
s
s
i
o
n
)

"
U
S
E
M
A
P
=
U
P
I

(
c
l
i
e
n
t
-
s
i
d
e

i
m
a
g
e

m
a
p
)

"

T
Y
P
E
=



(
c
o
n
t
e
n
t
-
t
y
p
e

o
f

o
b
j
e
c
t
)

“
C
O
D
E
T
Y
P
E
=



C
O
D
E
T
Y
P
E
=

e

(c o n t e n t - t y p e

o f

c o d e)

"

S T A N D B Y =

T
e
x
t

(m e s s a g e

t o

s h o w

w h i l e

l
o
a
d
i
n
g
)
“
T
A
B
I
N
D
E
X
=
N
C
M
B
E
R
(
p
o
s
i
t
i
o
n

i
n

t
a
b
b
i
n
g

o
r
d
e
r
)
“
D
E
C
L
A
R
E

(
d
o

n
o
t

i
n
s
t
a
n
t
i
a
t
e

o
b
j
e
c
t
)
"
A
L
L
I
G
N
=
[
t
o
p
|
m
i
d
d
l
e
|
b
o
t
t
o
m
|

l
e
f
t

l
r
i
g
h
t

]

(
o
b
j
e
c
t

a
l
i
g
n
m
e
n
t
)

»

B
O
R
D
E
R
=

l
l
l
l
l
l
l
l
l
l
l

(
l
i
n
k

b
o
r
d
e
r

w
i
d
t
h
)

“
H
S
P
A
C
E
=
P
i
x
e
r
t
i
s

(
h
o
r
i
z
o
n
t
a
l
g
u
t
t
e
r
)

“
V
S
P
A
C
E
=
P
i
x
e
r
t
i
s

(
v
e
r
t
i

calculator)

u

C
o
m
m
o
n

=

a
t
t
r
i
b
u
t
e
s

Contents
PAR
AM
elements
followed
by
block-
level
elements
and/
or
inline
elements
Contained
in
HEAD
D,
inline
elements,
block-
level

elem
ents
exce
pt
PRE

The **OBJECT** element is used to include *objects* such as images, videos, Java applets, and VRML worlds. **OBJECT** is intended to replace the more specific **IMG** and **APPLET** elements, as well as the proprietary **EMBED** and **BGSOUND** elements, though a lack of browser support and severe bugs in supporting browsers make the other elements a better choice for the time being.

OBJECT's **DATA** attribute specifies the URI of the embedded object. Relative URIs are interpreted with respect to the **CODEBASE** attribute if it is given.

The **WIDTH** and **HEIGHT** attributes define the dimensions of the object. The value may be given in pixels or as a percentage of the parent element's width or height. *Most browsers require the **WIDTH** and **HEIGHT** attributes for all objects embedded using **OBJECT**.*

The **CLASSID** may be used to specify an implementation for the object. Java applets, Python applets, and ActiveX controls all provide implementations for the embedded object, and so are specified with the **CLASSID** attribute, as in the following example:

```
<OBJECT CLASSID="yahtzee.py" CODETYPE="application/x-python" STANDBY="Ready to play  
Yahtzee?" TITLE="My Yahtzee Game">  
<OBJECT CLASSID="java:Yahtzee.class" CODETYPE="application/java" WIDTH=400 HEIGHT=250  
STANDBY="Ready to play Yahtzee?" TITLE="My Yahtzee Game">  
<OBJECT DATA="yahtzee.gif" TYPE="image/gif" TITLE="A Yahtzee animation" WIDTH=200  
HEIGHT=100>  
Yahtzee is my <EM>favorite</EM> game!  
</OBJECT>  
</OBJECT>  
</OBJECT>
```

This example also demonstrates the use of alternate content for browsers that cannot display the embedded object. In the example, a Yahtzee game written in Python is used if the browser supports it. A Java version is provided as an alternate for browsers that do not support Python applets. An image is given for browsers that cannot show the Java or Python applets, and plain text is used as a final alternate if images are not loaded. Note that **OBJECT** is backwards compatible with pre-HTML 4.0 browsers since they will ignore the **<OBJECT>** tags and render the innermost alternate content (the text in the example).

The preceding example also makes use of the **TYPE** and **CODETYPE** attributes to allow browsers to avoid requesting a file that they cannot render. The **TYPE** attribute specifies the media type of the resource referenced by the **DATA** attribute while the **CODETYPE** attribute specifies the media type of the **CLASSID** data.

The **STANDBY** attribute is also utilized in the example. This attribute provides short text to display while the object is loading.

The **ARCHIVE** attribute can specify a space-separated list of archived files (either absolute URIs or URIs relative to the **CODEBASE**), allowing the browser to download many files with a single connection and hence decreasing the total download time. The standard archive format for Java files is **JAR**. JAR files can be created with the **jar** tool included with the [Java Development Kit](#) 1.1 and up.

The **DECLARE** attribute makes the object a declaration that is not immediately instantiated. This allows the object to be instantiated from a link, button, or object later in the same document. The **ID** attribute must be used with declared objects as an identifier for the instantiating element. For example:

```
<OBJECT DECLARE ID=yahtzee CLASSID="java:Yahtzee.class" CODETYPE="application/java"
WIDTH=400 HEIGHT=250 TITLE="My Yahtzee Game">
<IMG SRC="yahtzee.gif" ALT="You get the dice!" TITLE="Yahtzee animation">
</OBJECT>
...
<P>Ready to <A HREF="#yahtzee">play Yahtzee</A>?</P>
```

The **OBJECT** element may contain **PARAM** elements--before any other content--to provide run-time initialization data. The following example embeds a video, with an audio clip for alternate content, and includes parameters commonly understood by audio/video plug-ins. Note the placement of **PARAM** elements *before* alternate content.

```
<OBJECT DATA="mlk.mov" TYPE="video/quicktime" TITLE="Martin Luther King's &quot;I Have
a Dream&quot; speech" WIDTH=150 HEIGHT=150>
<PARAM NAME=pluginspage VALUE="http://quicktime.apple.com/">
<PARAM NAME=autoplay VALUE=true>
<OBJECT DATA="mlk.wav" TYPE="audio/x-wav" TITLE="Martin Luther King's &quot;I Have a
Dream&quot; speech">
<PARAM NAME=autostart VALUE=true>
<PARAM NAME=hidden VALUE=true>
<A HREF="mlk.html">Full text of Martin Luther King's "I Have a Dream" speech</A>
</OBJECT>
</OBJECT>
```

The **USEMAP** attribute can be used with **OBJECT** to embed a clickable image where different coordinates have different link destinations. Image maps via the **IMG** element are better supported, but **OBJECT**-based image maps allow a full textual alternative for browsers not loading images. The **USEMAP** attribute points to a **MAP** element whose contents define the links of the various coordinates. The **MAP** may be included within the **OBJECT**, in which case its contents are not rendered on image-loading browsers, or it may be given outside the **OBJECT** element so that its contents are rendered.

The following example gives two images, one an alternate if the first type of image is not supported. The images share a single image map definition, which is included within the **OBJECT** element. The **MAP** element contains a menu of links to be rendered on browsers not loading images.

```
<OBJECT DATA="sitemap.png" USEMAP="#map" TYPE="image/png" TITLE="Site map" WIDTH=300
HEIGHT=200>
<OBJECT DATA="sitemap.gif" USEMAP="#map" TYPE="image/gif" TITLE="Site map" WIDTH=300
HEIGHT=200>
<MAP NAME=map>
<UL>
<LI><A href="http://www.htmlhelp.com/reference/" COORDS="5,5,95,195">HTML and CSS
Reference</A></LI>
<LI><A href="http://www.htmlhelp.com/design/" COORDS="105,5,195,195">Design
Guide</A></LI>
<LI><A href="http://www.htmlhelp.com/tools/index.html"
COORDS="205,5,295,195">Tools</A></LI>
</UL>
</MAP>
</OBJECT>
</OBJECT>
```

The **TABINDEX** attribute specifies a number between 0 and 32767 to indicate the tabbing order of the element. An

object with **TABINDEX=0** or no **TABINDEX** attribute will be visited after any elements with a positive **TABINDEX**. Among positive **TABINDEX** values, the lower number receives focus first. In the case of a tie, the element appearing first in the HTML document takes precedence.

The **ALIGN** attribute, [deprecated](#) in HTML 4.0, specifies the alignment of the object. The values **top**, **middle**, and **bottom** specify the object's position with respect to surrounding content on its left and right.

ALIGN=middle aligns the center of the object with the current baseline. To center the object horizontally on the page, place the object in a centered block, e.g.,

```
<P ALIGN=center><OBJECT DATA="foo.mov" TYPE="video/quicktime"></OBJECT></P>
```

The other **ALIGN** values, **left** and **right**, specify a *floating* object; the object is placed at the left or right margin and content flows around it. To place content below the object, use **<BR CLEAR=left|right|all>** as appropriate.

The [vertical-align](#) and [float](#) properties of [Cascading Style Sheets](#) provide more flexible methods of aligning objects.

The **BORDER** attribute, [deprecated](#) in HTML 4.0, specifies the width of the object's border. Specifying **BORDER=0** will eliminate the border around a linked object in most browsers, though some allow the user to override this. Authors should only use **BORDER=0** if the object would be clearly recognizable as a link, or as a method of de-emphasizing a link. For example:

```
<A HREF="reference/"><OBJECT DATA="icon/reference.gif" WIDTH=90 HEIGHT=90  
BORDER=0></OBJECT>Web Authoring Reference</A>
```

The [deprecated](#) **HSPACE** and **VSPACE** attributes allow an author to suggest horizontal gutters and vertical gutters, respectively, around the object. The value must be in pixels and applies to both sides of the object. [Style sheets](#) provide more flexibility in specifying the space around objects.

The **OBJECT** element is most useful as a **BODY** element and can be contained within either [inline](#) or [block-level](#) elements. The contents of **OBJECT** should be elements that can be contained within **OBJECT**'s parent element. For example, an [A](#) element containing an **OBJECT** should not have any block-level elements as the contents of the **OBJECT**.

More Information

- u [OBJECT in W3C HTML 4.0 Recommendation](#)
- u [The Java Tutorial](#)
- u [ActiveX Controls](#)
- u [Python Language Home Page](#)
- u [PNG \(Portable Network Graphics\) Home Page](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra <weballey@usa.net>](mailto:weballey@usa.net).
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

OPTGROUP - Option Group



OPTGROUP - Option Group

Syntax
<OPTGR
OUP>...<
/OPTGR
OUP>
Attri
bute
Spe
cific
ation
s

u
L
A
B
E
L
=
T
e
x
t
(
g
r
o
u
p
l
a
b
e
l
)
u
D
I
S
A
B
L
E

D

(
d
i
s
a
b
l
e

g
r
o
u
p

o
f

c
h
o
i
c
e
s
)

u

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
One
or
mor
e

OPT
ION
elem
ents
Cont
aine
d in
SEL

ECT

The **OPTGROUP** element defines a *group of choices* within a **SELECT** menu. **OPTGROUP** must contain one or more **OPTION** elements to define the actual choices.

The required **LABEL** attribute specifies the group label presented to the user. The **LABEL** should describe the group of choices available through the **OPTGROUP**'s **OPTION**s. Each **OPTION** generally uses a **LABEL** attribute as well to provide a shortened label that, together with the **OPTGROUP**'s **LABEL**, gives a complete description of the option. An example follows:

```
<P>Which Web browser do you use most often?
<SELECT NAME=browser>
  <OPTGROUP LABEL="Netscape Navigator">
    <OPTION LABEL="4.x or higher">
      Netscape Navigator 4.x or higher
    </OPTION>
    <OPTION LABEL="3.x">Netscape Navigator 3.x</OPTION>
    <OPTION LABEL="2.x">Netscape Navigator 2.x</OPTION>
    <OPTION LABEL="1.x">Netscape Navigator 1.x</OPTION>
  </OPTGROUP>
  <OPTGROUP LABEL="Microsoft Internet Explorer">
    <OPTION LABEL="4.x or higher">
      Microsoft Internet Explorer 4.x or higher
    </OPTION>
    <OPTION LABEL="3.x">Microsoft Internet Explorer 3.x</OPTION>
    <OPTION LABEL="2.x">Microsoft Internet Explorer 2.x</OPTION>
    <OPTION LABEL="1.x">Microsoft Internet Explorer 1.x</OPTION>
  </OPTGROUP>
  <OPTGROUP LABEL="Opera">
    <OPTION LABEL="3.x or higher">Opera 3.x or higher</OPTION>
    <OPTION LABEL="2.x">Opera 2.x</OPTION>
  </OPTGROUP>
  <OPTION>Other</OPTION>
</SELECT>
</P>
```

OPTGROUP is not well supported by current browsers, but its design allows authors to use it today without sacrificing compatibility with non-supporting browsers. Supporting browsers will render the preceding example using the **LABEL** attribute of **OPTION** to provide just the version number, along with the **OPTGROUP**'s **LABEL**, which gives the full name of the application. This allows a compact display with easy-to-use cascading menus.

Non-supporting browsers will ignore the **OPTGROUP** elements and **LABEL** attributes, providing the full name and version for each choice. Thus authors can fully use **OPTGROUP** despite its lack of browser support.

Note that, in HTML 4.0, **OPTGROUP** is limited to containing only **OPTION** elements, thus preventing nested **OPTGROUP**s with multi-level cascades. Future versions of HTML may add support for nested option groups.

The boolean **DISABLED** attribute makes the option group unavailable. The options of a disabled option group cannot be selected by the user and are never submitted with the form.

More Information

[OPTGROUP in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

Q - Short Quotation



Q - Short Quotation

Syntax
<Q>...</Q>
>
Attribute
Specifications

“
C
I
T
E
=
U
R
L
(
s
o
u
r
c
e
o
f
q
u
o
t
a
t
i
o
n
)
”
C
O
M

m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
Inlin
e
elem
ents,
bloc
k-
level
elem
ents

The **Q** element is used for *short, inline quotations*. For longer (block) quotations, use the **BLOCKQUOTE** element.

The **Q** element's optional **CITE** attribute provides a URI of the source of the quotation. Some examples follow:

```
<P>In the words of Albert Einstein, <Q>God does <EM>not</EM> play dice.</Q></P>
```

```
<P>According to Dave Raggett, <Q CITE="http://www.w3.org/Press/HTML4-REC">HTML 4.0  
gives Web designers the ability to create dynamic visually exciting pages that are  
accessible to all.</Q></P>
```

Note that authors should not include their own quotation marks when using the **Q** element. However, this can be a problem since almost all current browsers lack support for **Q**. Authors may prefer to avoid **Q** and insert their own quotation marks. Another alternative is to use **I** in combination with **Q** so that the quotation is distinguished from other text in most browsers. The previous example could also be given as follows:

```
<P>According to Dave Raggett, <I><Q CITE="http://www.w3.org/Press/HTML4-REC">HTML 4.0  
gives Web designers the ability to create dynamic visually exciting pages that are  
accessible to all.</Q></I></P>
```

Browsers supporting **Q** should properly handle nested quotations. They should also use quotation marks suitable to the language of the quotation, based on the **LANG** attribute of **Q** or the language of its parent.

More Information

[Q in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

S - Strike-through Text



S - Strike-through Text

Syntax
<S>...</S>
>
Attribute
Specifications

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

elem ents

The **S** element, [deprecated](#) in HTML 4.0, suggests that text be rendered with a *strike-through style*. In many cases, use of a [phrase element](#) such as **DEL** is more appropriate since such elements express the *meaning* of the text more clearly. However, since support for **DEL** among browsers is weak, **S** could be useful in combination with **DEL**, as in the following example:

```
The latest version of HTML recommended by the W3C is HTML <DEL DATETIME="1997-12-19T00:00:00-05:00"><s>3.2</s></DEL> <INS DATETIME="1997-12-19T00:00:00-05:00">4.0</INS>.
```

Note that **STRIKE** is better supported than **S** (based on Netscape 2.x and 1.22 supporting **STRIKE** but not **S**), and so **STRIKE** should be used in place of **S**. There does not appear to be any advantage to using both **STRIKE** and **S**; all browsers that support **S** also seem to support **STRIKE**.

If **DEL** is not a suitable structure, [style sheets](#) should be used to complement or replace instances of **S**. [CSS1](#) provides the [text-decoration](#) property for strike-through text.

More Information

„[S in W3C HTML 4.0 Recommendation](#)“

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

SPAN - Generic Inline Container



SPAN - Generic Inline Container

Syntax
..

Attribute
Specifications

[Common](#)
[Attributes](#)

Contents
[Inline elements](#)
Contained in
[Inline elements, block-level](#)

elem ents

The **SPAN** element is a *generic inline container*. **SPAN** carries no structural meaning itself, but it can be used to provide extra structure through its **LANG**, **DIR**, **CLASS**, and **ID** attributes. [Style sheets](#) are often used to suggest a presentation for a given class or ID.

SPAN should only be used where no other HTML [inline element](#) provides a suitable meaning. If a presentation such as bold or italic text would be suitable on visual browsers, authors may prefer to use an appropriate [font style element](#). For example:

1. `<P>La Révolution Tranquille shook Quebec in the early 1960's.`
2. `<P><I LANG=fr>La Révolution Tranquille</I> shook Quebec in the early 1960's.`

These examples are identical in meaning, but the second example uses the **I** element to suggest italic text.

DIV is a block-level equivalent of **SPAN** for containing [block-level elements](#) such as **P** and **TABLE**.

More Information

„[SPAN in W3C HTML 4.0 Recommendation](#)“

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

TBODY - Table Body



TBODY - Table Body

Syntax
<TBODY
>...</TBO
DY>
Attri
bute
Spe
cific
ation
s

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j

u
s
t
i
f
y

|

c
h
a
r

|

(
h
o
r
i
z
o
n
t
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s

i
n

g
r
o
u
p
)

“
C
H
A
R
=

||| ||| ||| ||| ||| ||| |||

(a l i g n m e n t

c h a r a c t e r

f o r

c e l l s)

= C H A R O F F =

||| ||| ||| ||| ||| ||| |||

(a l i

g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

o
f
f
s
e
t
)

"
V
A
L
L
I
G
N
=
[

t
o
p

|

m
i
d
d
l
e

|

b
o
t
t
o
m

|

b
a
s

e
l
i
n
e

J
(
v
e
r
t
i
c
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s

i
n

g
r
o
u
p
)

c
o
m
m
o
n
a
t
t
r
i
b
u
t

e
s
Cont
ents
One
or
mor
e TR
elem
ents
Cont
aine
d in
TA
BL

The **TBODY** element defines a *group of data rows* in a table. A **TABLE** must have one or more **TBODY** elements, which must follow the optional **TFOOT**. The **TBODY** end tag is always optional. The start tag is optional when the table contains only one **TBODY** and no **THEAD** or **TFOOT**. This allows the simple table structure of [HTML 3.2](#) to still be valid:

```
<TABLE>  
  <TR>  
    <TH>Abbreviation</TH>  
    <TH>Long Form</TH>  
  </TR>  
  <TR>  
    <TD>AFAIK</TD>  
    <TD>As Far As I Know</TD>  
  </TR>  
</TABLE>
```

By explicitly grouping rows with **THEAD**, **TFOOT**, and **TBODY**, authors give browsers the ability to present a long table with a scrolling body and static header and footer rows. Using **TBODY** also provides the ability to easily suggest different presentations for different row groups through [style sheets](#). While few browsers currently support **TBODY**, it can be used with no harm on non-supporting browsers.

The following example gives a table of SI units of measure. **TBODY** elements are used to group rows based on whether the unit is classed as a "base" unit, "derived" unit, or "supplementary" unit.

```

<TABLE SUMMARY="This table lists SI (International System) units of
                measure, giving the name of the unit, its symbol, and
                the quantity that it measures.">
  <CAPTION>SI Units</CAPTION>
  <THEAD>
    <TR>
      <TH SCOPE=col>Name</TH>
      <TH SCOPE=col>Symbol</TH>
      <TH SCOPE=col>Quantity</TH>
    </TR>
  </THEAD>
  <TBODY CLASS=base>
    <TR>
      <TD SCOPE=row>meter</TD>
      <TD>m</TD>
      <TD>length</TD>
    </TR>
    <TR>
      <TD SCOPE=row>kilogram</TD>
      <TD>kg</TD>
      <TD>mass</TD>
    </TR>
    ...
  </TBODY>
  <TBODY CLASS=derived>
    <TR>
      <TD SCOPE=row>hertz</TD>
      <TD>Hz</TD>
      <TD>frequency</TD>
    </TR>
    <TR>
      <TD SCOPE=row>pascal</TD>
      <TD>Pa</TD>
      <TD>pressure</TD>
    </TR>
    ...
  </TBODY>
  <TBODY CLASS=supplementary>
    <TR>
      <TD SCOPE=row>radian</TD>
      <TD>rad</TD>
      <TD>plane angle</TD>
    </TR>
    ...
  </TBODY>
</TABLE>

```

In addition to the [attributes common to most elements](#), **TBODY** takes presentational attributes for specifying the alignment of cell data. Since few browsers support **TBODY**, authors may wish to specify these attributes on the **TR** or **TD** elements instead.

The **ALIGN** attribute specifies the horizontal alignment for each cell in the row group. Possible values are **left**, **center**, **right**, **justify**, and **char**. **ALIGN=char** aligns a cell's contents on the character given in the **CHAR** attribute. The default value for the **CHAR** attribute is the decimal point of the current language--a period in English. The **CHAROFF** attribute specifies the offset to the first occurrence of the alignment character. The attribute's value is a number in pixels or a percentage of the cell's width; **CHAROFF="50%"** centers the alignment character horizontally in a cell.

The **VALIGN** attribute specifies the vertical position of a cell's contents. Possible values are:

- **top**, which positions data at the top of the cell;
- **middle**, the default value, which centers the cell data vertically;
- **bottom**, which positions data at the bottom of the cell;
- **baseline**, which specifies that the first line of each cell in the row with **ALIGN=baseline** should occur on a common baseline.

More Information

• [TBODY in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

TFOOT - Table Foot



TFOOT - Table Foot

Syntax
<TFOOT
>...</TFO
OT>
Attri
bute
Spe
cific
ation
s

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j

u
s
t
i
f
y

|

c
h
a
r

|

(
h
o
r
i
z
o
n
t
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s

i
n

g
r
o
u
p
)

“
C
H
A
R
=

||| ||| ||| ||| ||| ||| ||| ||| ||| |||

(a l i g n m e n t

c h a r a c t e r

f o r

c e l l s)

“ C H A R O F F =

||| ||| ||| ||| ||| ||| ||| ||| |||

(a l i

g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

o
f
f
s
e
t
)

"
V
A
L
L
I
G
N
=
[

t
o
p

|

m
i
d
d
l
e

|

b
o
t
t
o
m

|

b
a
s

e
l
i
n
e

J
(
v
e
r
t
i
c
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s

i
n

g
r
o
u
p
)

c
o
m
m
o
n
a
t
t
r
i
b
u
t

e
s

Cont
ents
One
or
mor
e TR
elem
ents
Cont
aine
d in
TAB
LE

The **TFOOT** element defines a *group of footer rows* in a table. A **TABLE** may have one **TFOOT**, which must follow the optional **THEAD** and precede the required **TBODY**.

By explicitly grouping footer rows with **TFOOT**, authors give browsers the ability to include the footer rows on each page of a printed, multi-page table, as well as the ability to present a long table with a scrolling body and static footer rows. However, few browsers currently support **TFOOT**, and the requirement that it be placed before the **TBODY** may make it unsuitable for non-supporting browsers. If the presentation of footer rows prior to body rows is not acceptable, authors should avoid using **TFOOT** until browser support is greater.

A table footer may provide a summary row or footnotes that apply to the entire table or portions of it. The following example uses **TFOOT** to contain footnotes for a table:

```

<TABLE SUMMARY="This table lists program available at the university
                based on the discipline and type of degree.">
  <CAPTION>Programs Available</CAPTION>
  <COLGROUP CLASS="program-discipline">
  <COLGROUP CLASS="program-type" SPAN=5>
  <THEAD>
    <TR>
      <TH SCOPE=col>Program</TH>
      <TH SCOPE=col>Honors Co-op</TH>
      <TH SCOPE=col>Honors Regular</TH>
      <TH SCOPE=col>General Regular</TH>
      <TH SCOPE=col>*Preprofessional or Professional</TH>
    </TR>
  </THEAD>
  <TFOOT CLASS=footnote>
    <TR>
      <TD COLSPAN=5>
        Many disciplines are also available as Minors and Joint
        Honors programs.
      </TD>
    </TR>
    <TR>
      <TD COLSPAN=5>
        * Preprofessional programs normally fulfill the academic
        requirements for registration in the related professions.
      </TD>
    </TR>
  </TFOOT>
  <TBODY>
    <TR>
      <TD SCOPE=row>Computer Science</TD>
      <TD>yes</TD>
      <TD>yes</TD>
      <TD>no</TD>
      <TD>no</TD>
    </TR>
    ...
  </TBODY>
</TABLE>

```

In addition to the [attributes common to most elements](#), **TFOOT** takes presentational attributes for specifying the alignment of cell data. Since few browsers support **TFOOT**, authors may wish to specify these attributes on the **TR** or **TD** elements instead.

The **ALIGN** attribute specifies the horizontal alignment for each cell in the row group. Possible values are **left**, **center**, **right**, **justify**, and **char**. **ALIGN=char** aligns a cell's contents on the character given in the **CHAR** attribute. The default value for the **CHAR** attribute is the decimal point of the current language--a period in English. The **CHAROFF** attribute specifies the offset to the first occurrence of the alignment character. The attribute's value is a number in pixels or a percentage of the cell's width; **CHAROFF="50%"** centers the alignment character horizontally in a cell.

The **VALIGN** attribute specifies the vertical position of a cell's contents. Possible values are:

- **top**, which positions data at the top of the cell;
- **middle**, the default value, which centers the cell data vertically;
- **bottom**, which positions data at the bottom of the cell;
- **baseline**, which specifies that the first line of each cell in the row with **ALIGN=baseline** should occur on a

common baseline.

More Information

u [TFOOT in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

THEAD - Table Head



THEAD - Table Head

Syntax
<THEAD
>...</THE
AD>
Attri
bute
Spe
cific
ation
s

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j

u
s
t
i
f
y

|

c
h
a
r

|

(
h
o
r
i
z
o
n
t
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s

i
n

g
r
o
u
p
)

“
C
H
A
R
=

g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

o
f
f
s
e
t
)

"
V
A
L
L
I
G
N
=
[

t
o
p

|

m
i
d
d
l
e

|

b
o
t
t
o
m

|

b
a
s

e
l
i
n
e

J
(
v
e
r
t
i
c
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s

i
n

g
r
o
u
p
)

c
o
m
m
o
n
a
t
t
r
i
b
u
t

e
s
Cont
ents
One
or
mor
e TR
elem
ents
Cont
aine
d in
TAB
LE

The **THEAD** element defines a *group of header rows* in a table. A **TABLE** may have one **THEAD**, which must follow any **CAPTION**, **COL**, or **COLGROUP** elements, and precede the optional **TFOOT** and required **TBODY** elements.

By explicitly grouping header rows with **THEAD**, authors give browsers the ability to include the header rows on each page of a printed, multi-page table, as well as the ability to present a long table with a scrolling body and static header rows. While few browsers currently support **THEAD**, it can be used with no harm on non-supporting browsers.

The following example uses **THEAD** to group the two header rows of a table:

```

<TABLE SUMMARY="This table gives the character entity reference,
                decimal character reference, and hexadecimal character
                reference for 8-bit Latin-1 characters, as well as the
                rendering of each in your browser.">
<COLGROUP>
<COLGROUP SPAN=3>
<COLGROUP SPAN=3>
<THEAD>
  <TR>
    <TH SCOPE=col ROWSPAN=2>Character</TH>
    <TH SCOPE=col ROWSPAN=2>Entity</TH>
    <TH SCOPE=col ROWSPAN=2>Decimal</TH>
    <TH SCOPE=col ROWSPAN=2>Hex</TH>
    <TH SCOPE=colgroup COLSPAN=3>Rendering in Your Browser</TH>
  </TR>
  <TR>
    <TH SCOPE=col>Entity</TH>
    <TH SCOPE=col>Decimal</TH>
    <TH SCOPE=col>Hex</TH>
  </TR>
</THEAD>
<TBODY>
  <TR>
    <TD SCOPE=row>non-breaking space</TD>
    <TD>&nbsp; </TD>
    <TD>&#160;</TD>
    <TD>&#xA0;</TD>
    <TD>&nbsp;</TD>
    <TD>&#160;</TD>
    <TD>&#xA0;</TD>
  </TR>
  . . .
</TBODY>
</TABLE>

```

In addition to the [attributes common to most elements](#), **THEAD** takes presentational attributes for specifying the alignment of cell data. Since few browsers support **THEAD**, authors may wish to specify these attributes on the **TR**, **TH**, or **TD** elements instead.

The **ALIGN** attribute specifies the horizontal alignment for each cell in the row group. Possible values are **left**, **center**, **right**, **justify**, and **char**. **ALIGN=char** aligns a cell's contents on the character given in the **CHAR** attribute. The default value for the **CHAR** attribute is the decimal point of the current language--a period in English. The **CHAROFF** attribute specifies the offset to the first occurrence of the alignment character. The attribute's value is a number in pixels or a percentage of the cell's width; **CHAROFF="50%"** centers the alignment character horizontally in a cell.

The **VALIGN** attribute specifies the vertical position of a cell's contents. Possible values are:

- **top**, which positions data at the top of the cell;
- **middle**, the default value, which centers the cell data vertically;
- **bottom**, which positions data at the bottom of the cell;
- **baseline**, which specifies that the first line of each cell in the row with **ALIGN=baseline** should occur on a common baseline.

More Information

[THEAD in W3C HTML 4.0 Recommendation](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

HTML 4.0 Deprecated Features



HTML 4.0 Deprecated Features

A number of elements and attributes are *deprecated* in HTML 4.0 as an indication that other methods of accomplishing the same task are preferred. Deprecated features may become obsolete in future versions of HTML, though browsers that support the features will likely continue to support them. Deprecated features are included in [HTML 4.0 Transitional](#) and [HTML 4.0 Frameset](#) but not [HTML 4.0 Strict](#).

Many presentational elements and attributes are deprecated in favor of [style sheets](#), which allow authors to suggest a presentation with more flexibility and without sacrificing accessibility. Presentational attributes that cannot currently be replaced with style sheets are in most cases not deprecated.

The following elements are deprecated in favor of style sheets:

- [BASEFONT](#) - Base font change
- [CENTER](#) - Centered block
- [FONT](#) - Font change
- [S](#) - Strike-through text
- [STRIKE](#) - Strike-through text
- [U](#) - Underlined text

The following elements are also deprecated:

- [APPLET](#) - Java applet (deprecated in favor of [OBJECT](#))
- [DIR](#) - Directory list (deprecated in favor of [UL](#))
- [ISINDEX](#) - Input prompt (deprecated in favor of [INPUT](#))
- [MENU](#) - Menu list (deprecated in favor of [UL](#))

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

TABLE - Table



TABLE - Table

Syntax
<TABLE>
...</TABL
E>
Attri
bute
Spe
cific
ation
s

u
S
U
M
M
A
R
Y
=
T
I
E
X
T
(
p
u
r
p
o
s
e
/
s
t
r
u
c
t
u
r
e
o
f

t
a
b
l
e
)

u
W
I
D
T
H
=
L
e
n
g
t
h

(
t
a
b
l
e

w
i
d
t
h
)

u
B
O
R
D
E
R
=
P
i
x
e
l
s

(
b
o
r
d
e
r

w
i
d

t
h
)

“
F
R
A
M
E
=
[

v
o
i
d

|
a
b
o
v
e

|
b
e
l
o
w

|
h
s
i
d
e
s

|
l
h
s

|
r
h
s

|
v
s
i
d
e

s
|
b
o
x
|
b
o
r
d
e
r
|
(
o
u
t
e
r
b
o
r
d
e
r
)
"
R
U
L
E
S
=
[
n
o
n
e
|
g
r
o
u
p
s
|
r
o
w

s

|

c

o

|

a

|

]

(

i

n

n

e

r

b

o

r

d

e

r

s

)

“

C

E

L

L

S

P

A

C

I

N

G

=

I

e

n

n

e

r

b

o

r

d

e

r

s

(

s

p

a

c

i

n

g

)

b

e
t
w
e
e
n

c
e
l
l
s
)

“
C
E
L
L
P
A
D
D
I
N
G
=
I
n
d
e
r
h

(
s
p
a
c
i
n
g

w
i
t
h
i
n

c
e
l
l
s
)

“
A
L
I
G

```
N
=
[
  l
  e
  f
  t
  |
  c
  e
  n
  t
  e
  r
  |
  r
  i
  g
  h
  t
]
(
  t
  a
  b
  l
  e
  a
  l
  i
  g
  n
  m
  e
  n
  t
)
"
B
G
C
O
L
O
R
=
C
|||
|||
|||
|||
"
(
```

t
a
b
l
e

b
a
c
k
g
r
o
u
n
d

c
o
l
o
r
)

u

c
o
m
m
o
n
= a
t
t
r
i
b
u
t
t
e
s

Cont
ents
An
optio
nal
CAP
TIO
N,
follo
wed
by
zero
or
mor
e
COL
and
COL
GR

OUP

elements,
followed
by
an
optional

THE

AD

element,
an
optional

TFO

QT

element,
and
then
one
or
more

TBO

DY

elements

Contained
in

APP

LET,

BLO

CKQ

UOT

E,

BOD

Y,

BUT

TON

,

CEN

TER

,

DD,

DEL,

DIV,

FIEL

DSE

T,

FOR

M,

IFR

AME

,

INS,

LI,

MAP

,

NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,
TH

The **TABLE** element defines a *table* for multi-dimensional data arranged in rows and columns. **TABLE** is commonly used as a layout device, but authors should avoid this practice as much as possible. Tables can cause problems for users of narrow windows, large fonts, or non-visual browsers, and these problems are often accentuated when tables are used solely for layout purposes. As well, current visual browsers will not display anything until the complete table has been downloaded, which can have very noticeable effects when an entire document is laid out within a **TABLE**. Authors should try to use [style sheets](#) in place of **TABLE** for layout, though bugs in current browser implementations of style sheets can make this difficult.

The **TABLE** may contain a number of optional elements to provide a rich structure to the table. The optional **CAPTION** element gives a caption for the table and is followed by optional **COL** and **COLGROUP** elements that specify column widths and groupings. The **THEAD**, **TFOOT**, and **TBODY** elements then follow with groups of rows. The optional **THEAD** and **TFOOT** elements contain header and footer rows, respectively, while **TBODY** elements supply the table's main row groups. A row group contains **TR** elements for individual rows, and each **TR** contains **TH** or **TD** elements for header cells or data cells, respectively.

At least one **TBODY** element is required within a **TABLE**, but **TBODY**'s start and end tags are both optional if there is only one **TBODY** and no **THEAD** or **TFOOT**. A simple table could thus be expressed as follows:

```
<TABLE>
  <TR>
    <TH>Abbreviation</TH>
    <TH>Long Form</TH>
  </TR>
  <TR>
    <TD>AFAIK</TD>
    <TD>As Far As I Know</TD>
  </TR>
  <TR>
    <TD>IMHO</TD>
    <TD>In My Humble Opinion</TD>
  </TR>
  <TR>
    <TD>OTOH</TD>
    <TD>On The Other Hand</TD>
  </TR>
</TABLE>
```

The same table could be expressed with a richer structure by grouping rows and adding a caption, as in the next example. The extra structural information allows an author to more easily suggest the presentation of the table using [style sheets](#) or **TABLE**'s [presentational attributes](#).

```

<TABLE>
  <CAPTION>Common Usenet Abbreviations</CAPTION>
  <THEAD>
    <TR>
      <TH>Abbreviation</TH>
      <TH>Long Form</TH>
    </TR>
  </THEAD>
  <TBODY>
    <TR>
      <TD>AFAIK</TD>
      <TD>As Far As I Know</TD>
    </TR>
    <TR>
      <TD>IMHO</TD>
      <TD>In My Humble Opinion</TD>
    </TR>
    <TR>
      <TD>OTOH</TD>
      <TD>On The Other Hand</TD>
    </TR>
  </TBODY>
</TABLE>

```

The **TABLE** element takes an optional **SUMMARY** attribute to describe the purpose and/or structure of the table. The overview provided by the **SUMMARY** attribute is particularly helpful to users of non-visual browsers. With simple tables, a good **CAPTION** is usually a sufficient summary, but complex tables may benefit from a more detailed overview via the **SUMMARY** attribute. The following example uses **SUMMARY** to describe a table. Note that the summary could also be included in a paragraph before the table, which is helpful since few browsers support **SUMMARY**.

```

<TABLE SUMMARY="This table gives the character entity reference,
                decimal character reference, and hexadecimal character
                reference for symbols and Greek letters.">
  <COLGROUP>
  <COLGROUP SPAN=3>
  <THEAD>
    <TR>
      <TH SCOPE=col>Character</TH>
      <TH SCOPE=col>Entity</TH>
      <TH SCOPE=col>Decimal</TH>
      <TH SCOPE=col>Hex</TH>
    </TR>
  </THEAD>
  <TBODY>
    <TR>
      <TD SCOPE=row>Latin small f with hook</TD>
      <TD>&#xf0f;</TD>
      <TD>&#402;</TD>
      <TD>&#x192;</TD>
    </TR>
    ...
  </TBODY>
</TABLE>

```

The **TABLE** element also takes a number of optional attributes to provide presentational hints in visual browsers. Equivalent of these attributes in [Cascading Style Sheets](#) are [under development](#) and not widely supported by browsers.

• The **WIDTH** attribute specifies the width of the table as a number of pixels or as a percentage of the available horizontal space. Widths in pixels should be avoided, especially widths above 500 pixels, since this causes unnecessary horizontal scrolling for some users.

• The **BORDER** attribute specifies the width in pixels of the border around a table.

• The **FRAME** attribute, poorly supported by browsers, specifies which sides of the table's outer border are visible. Possible values are **void** for no border, **above** for a top border only, **below** for a bottom border only, **hsides** for left and right borders only, **vsides** for top and bottom borders only, **lhs** for a left border only, **rhs** for a right border only, and either **box** or **border** for a border on all sides. The default value is **void** unless the **BORDER** attribute gives a positive width, in which case **FRAME=border** is the default. **<TABLE BORDER>** is a valid, well-supported shorthand for **<TABLE FRAME=border>**.

• The **RULES** attribute, poorly supported by browsers, specifies the borders between table cells. Possible values are **none** for no inner borders, **groups** for borders between row groups and column groups only, **rows** for borders between rows only, **cols** for borders between columns only, and **all** for borders between all cells. **None** is the default value if **BORDER=0** is used or if no **BORDER** attribute is given. **All** is the default value for any other use of **BORDER**.

• The **CELLSPACING** attribute defines the amount of space between table cells, and the **CELLPADDING** attribute defines the amount of space within table cells (*i.e.*, between the border and cell contents). The value may be given as a number of pixels or as a percentage, though most browsers do not support percentages, treating **CELLPADDING="20%"** as if it were **CELLPADDING="20"**. A percentage value is relative to the vertical space available for vertical padding or spacing, and the amount is split evenly between the top and bottom. Horizontal padding and spacing behave similarly. The padding or spacing is always applied to all four sides.

The **padding** properties of [Cascading Style Sheets](#) allow an author to suggest different padding for different sides, but are not as well supported as the **CELLPADDING** attribute.

• The **deprecated ALIGN** attribute suggests the horizontal alignment of the table on visual browsers. Possible values are **left**, **right**, and **center**. Browsers generally present left- or right-aligned tables as *floating* tables, with the content following the **TABLE** flowing around it. To prevent content from flowing around the table, use **<BR CLEAR=all>** after the end of the **TABLE**. Since many browsers do not support **ALIGN=center** with **TABLE**, authors may wish to place the **TABLE** within a **CENTER** element.

[Style sheets](#) provide more flexibility in suggesting table alignment but with less browser support than the **ALIGN** attribute.

• The **deprecated BGCOLOR** attribute suggests a background color for the table. The combination of this attribute with **** can leave invisible or unreadable text on Netscape Navigator 2.x, which does not support **BGCOLOR** on table elements. **BGCOLOR** is dangerous even on supporting browsers, since most fail to override it when overriding other author-specified colors. [Style sheets](#) provide a safer, more flexible method of specifying a table's background color.

More Information

• [TABLE in W3C HTML 4.0 Recommendation](#)

• [TABLE in W3C HTML 3.2 Recommendation](#)

• [TABLE in WDG HTML 3.2 Reference](#)

[TABLE in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

TH - Table Header Cell



TH - Table Header Cell

Syntax
<TH>...</
TH>
Attri
bute
Spe
cific
ation
s

u
R
O
W
S
P
A
N
=
N
u
m
p
o
r
(
r
o
w
s
s
p
a
n
n
e
d
b
y
t
h
e

cell)

COLSPAN = N

(columns

spanned

by

the

cell)

HEADERS =

I
D
E
F
S

(
l
i
s
t

o
f

h
e
a
d
e
r

c
e
l
l
s

f
o
r

c
u
r
r
e
n
t

c
e
l
l
)

u

A
B
B
R

=

F
e
x
t

(
a
b
b

r
e
v
i
a
t
i
o
n

f
o
r

h
e
a
d
e
r

c
e
l
l
)

“
S
C
O
P
E
=
[

r
o
w

|

c
o
l
|

r
o
w
g
r
o
u
p

|

c
o
l
g

r
o
u
p
]
(
c
e
l
l
s
c
o
v
e
r
e
d
b
y
h
e
a
d
e
r
c
e
l
l
)
"
A
X
I
S
=
C
D
A
T
A
(
c
a
t
e
g
o
r
y
o
f

header
cell
)
"ALIGN=
[
left
|
center
|
right
|
justify
|
char

]

(horizontal alignment)

»

C
H
A
R

=

C
||
||
||
||
||
||
||
||
||
||
||
||
||
||
||
||
||
||

(alignment

c
h
a
r
a
c
t
e
r

»

C
H
A
R
O
F
F
=
L
l
n
g
h

(
a
l
i
g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

o
f
f
s
e
t
)

u
V
A
L
I
G
N
=
[

t
o
p

|
m
i

d
d
l
e

l

b
o
t
t
o
m

l

b
a
s
e
l
i
n
e

J

(
v
e
r
t
i
c
a
l
a
l
i
g
n
m
e
n
t
)

"
W
I
D
T
H
=
P
i
X
e
i
S

(

c
e
l
l
w
i
d
t
h
)

“
H
E
I
G
H
T
=
 $\frac{p}{\rho \times g}$
+
 $\frac{v^2}{2g}$
+
 $\frac{z}{g}$
+
 $\frac{p}{\rho g}$ ”

(
c
e
l
l
h
e
i
g
h
t
)

“
N
O
W
R
A
P

(
s
u
p
p
r
e
s
s

w
o
r
d

w
r
a
p
)
= B G C O L O R =
C
o
l
o
r
(c e l l b a c k g r o u n d
c o l o r)
=
C
M
O
N
=
a
t
t
r
i
b
u
s
e

Cont
ents
[Inlin
e
elem
ents,
bloc
k-
level
elem
ents](#)
Cont
aine
d in
[TR](#)

The **TH** element defines a *header cell* in a table. **TH** elements are contained within a [TR](#) element (a table row), which may also contain [TD](#) elements for data cells. When a cell's contents act as both header information and table data, **TD** should be used.

The **ROWSPAN** and **COLSPAN** attributes of **TH** specify the number of rows and the number of columns, respectively, that are spanned by the cell. The default value is **1**. The special value **0** indicates that the cell spans all rows or columns to the end of the table. The value **0** is ignored by most browsers, so authors may wish to calculate the exact number of rows or columns spanned and use that value.

The **HEADERS** attribute specifies the header cells that apply to the **TH**. The value is a space-separated list of the header cells' [ID](#) attribute values. The **HEADERS** attribute allows non-visual browsers to render the header information for a given cell.

The **ABBR** attribute gives an abbreviated version of the cell's content. This allows visual browsers to use the short form if space is limited, and non-visual browsers can give a cell's header information in an abbreviated form before rendering each cell.

The **SCOPE** attribute specifies the cells for which the **TH** element provides header information. **SCOPE** is a simpler alternative to using **HEADERS** if the arrangement of header cells is not complex. Possible values are as follows:

- **row**, when the **TH** provides header information for the rest of the row;
- **col**, when the **TH** provides header information for the rest of the column;
- **rowgroup**, when the **TH** gives header information for the rest of the row group (*i.e.*, the remaining cells of the [THEAD](#), [TFOOT](#), or [TBODY](#));
- **colgroup**, when the **TH** gives header information for the rest of the column group (*i.e.*, the remaining cells of the [COLGROUP](#)).

The **AXIS** attribute provides a method of categorizing cells. The attribute's value is a comma-separated list of category names. See the [HTML 4.0 Recommendation](#)'s section on [categorizing cells](#) for an application of **AXIS**.

In addition to the [attributes common to most elements](#), **TH** takes a number of presentational attributes. [Style sheets](#) provide a more flexible way to suggest a presentation for table cells, but **TH**'s presentational attributes are more widely supported by current browsers.

The **ALIGN** attribute specifies the horizontal alignment for the cell. Possible values are **left**, **center**, **right**, **justify**, and **char**. **ALIGN=char**, poorly supported among browsers, aligns a cell's contents on the character given in the

CHAR attribute. The default value for the **CHAR** attribute is the decimal point of the current language--a period in English. The **CHAROFF** attribute specifies the offset to the first occurrence of the alignment character. The attribute's value is a number in pixels or a percentage of the cell's width; **CHAROFF="50%"** centers the alignment character horizontally in a cell.

The **VALIGN** attribute specifies the vertical position of a cell's contents. Possible values are:

- **top**, which positions data at the top of the cell;
- **middle**, the default value, which centers the cell data vertically;
- **bottom**, which positions data at the bottom of the cell;
- **baseline**, which specifies that the first line of each cell in the row with **ALIGN=baseline** should occur on a common baseline.

The **WIDTH**, **HEIGHT**, **NOWRAP**, and **BGCOLOR** attributes are all [deprecated](#) in favor of [style sheets](#). **WIDTH** and **HEIGHT** suggest the cell's width and height in pixels. The boolean **NOWRAP** attribute tells visual browsers to disable word wrap for the cell, which can result in unnecessary horizontal scrolling depending on the user's window width and font size.

The **BGCOLOR** attribute suggests a background color for the cell. The combination of this attribute with **** can leave invisible or unreadable text on Netscape Navigator 2.x, which does not support **BGCOLOR** on table elements. **BGCOLOR** is dangerous even on supporting browsers, since most fail to override it when overriding other author-specified colors. [Style sheets](#) provide a safer, more flexible method of specifying a table's background color.

More Information

- [TH in W3C HTML 4.0 Recommendation](#)
- [TH in W3C HTML 3.2 Recommendation](#)
- [TH in WDG HTML 3.2 Reference](#)
- [TH in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

TD - Table Data Cell



TD - Table Data Cell

Syntax
<TD>...</
TD>
Attri
bute
Spe
cific
ation
s

u
R
O
W
S
P
A
N
=
N
u
m
p
o
r
(
r
o
w
s
s
p
a
n
n
e
d
b
y
t
h
e

cell)

COLSPAN = N

(columns

spanned

by

the

cell)

HEADERS =

I
D
E
F
S

(
l
i
s
t

o
f

h
e
a
d
e
r

c
e
l
l
s

f
o
r

c
u
r
r
e
n
t

c
e
l
l
)

"

A
B
B
R

=

F
e
x
t

(
a
b
b

r
e
v
i
a
t
i
o
n

f
o
r

h
e
a
d
e
r

c
e
l
l
)

" S
C
O
P
E
=
[

r
o
w

|

c
o
l
|

r
o
w
g
r
o
u
p

|

c
o
l
g

r
o
u
p
]
(
c
e
l
l
s
c
o
v
e
r
e
d
b
y
h
e
a
d
e
r
c
e
l
l
)
"
A
X
I
S
=
C
D
A
T
A
(
c
a
t
e
g
o
r
y
o
f

header
cell
)
"ALIGN=
[
left
|
center
|
right
|
justify
|
char

l
(horizontal alignment)

» C H A R = C
||
||
||
||
||
||
||
||

(alignment

character)

»

C
H
A
R
O
F
F
=
L
l
n
g
t
h

(
a
l
i
g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

o
f
f
s
e
t
)

“
V
A
L
I
G
N
=
[

t
o
p

|
m
i

d
d
l
e

l

b
o
t
t
o
m

l

b
a
s
e
l
i
n
e

J

(
v
e
r
t
i
c
a
l
a
l
i
g
n
m
e
n
t
)

“
W
I
D
T
H
=
P
i
X
e
i
S

(

c
e
l
l
w
i
d
t
h
)

“
H
E
I
G
H
T
=
~~p~~
~~+~~
~~x~~
~~e~~
~~+~~
~~e~~

(
c
e
l
l
h
e
i
g
h
t
)

“
N
O
W
R
A
P

(
s
u
p
p
r
e
s
s

w
o
r
d

w
r
a
p
)
= B G C O L O R =
C
o
l
o
r
(c e l l b a c k g r o u n d
c o l o r)
=
C
M
O
N
=
a
t
t
r
i
b
u
s
e

Cont
ents
[Inlin
e
elem
ents,
bloc
k-
level
elem
ents](#)
Cont
aine
d in
[TR](#)

The **TD** element defines a *data cell* in a table. **TD** elements are contained within a **TR** element (a table row), which may also contain **TH** elements for header cells. When a cell's contents act as both header information and table data, **TD** should be used.

The **ROWSPAN** and **COLSPAN** attributes of **TD** specify the number of rows and the number of columns, respectively, that are spanned by the cell. The default value is **1**. The special value **0** indicates that the cell spans all rows or columns to the end of the table. The value **0** is ignored by most browsers, so authors may wish to calculate the exact number of rows or columns spanned and use that value.

The **HEADERS** attribute specifies the header cells that apply to the **TD**. The value is a space-separated list of the header cells' **ID** attribute values. The **HEADERS** attribute allows non-visual browsers to render the header information for a given cell.

The **ABBR**, **SCOPE**, and **AXIS** attributes should only be used if the cell provides header information. Like **HEADERS**, these attributes are new in HTML 4.0 and not well supported, though they should be particularly helpful to non-visual browsers in the future.

ABBR gives an abbreviated version of the cell's content. This allows visual browsers to use the short form if space is limited, and non-visual browsers can give a cell's header information in an abbreviated form before rendering each cell.

The **SCOPE** attribute specifies the cells for which the **TD** element provides header information. **SCOPE** is a simpler alternative to using **HEADERS** if the arrangement of header cells is not complex. Possible values are as follows:

- **row**, when the **TD** provides header information for the rest of the row;
- **col**, when the **TD** provides header information for the rest of the column;
- **rowgroup**, when the **TD** gives header information for the rest of the row group (*i.e.*, the remaining cells of the **THEAD**, **TFOOT**, or **TBODY**);
- **colgroup**, when the **TD** gives header information for the rest of the column group (*i.e.*, the remaining cells of the **COLGROUP**).

The **AXIS** attribute provides a method of categorizing cells. The attribute's value is a comma-separated list of category names. See the [HTML 4.0 Recommendation](#)'s section on [categorizing cells](#) for an application of **AXIS**.

In addition to the [attributes common to most elements](#), **TD** takes a number of presentational attributes. [Style sheets](#) provide a more flexible way to suggest a presentation for table cells, but **TD**'s presentational attributes are more

widely supported by current browsers.

The **ALIGN** attribute specifies the horizontal alignment for the cell. Possible values are **left**, **center**, **right**, **justify**, and **char**. **ALIGN=char**, poorly supported among browsers, aligns a cell's contents on the character given in the **CHAR** attribute. The default value for the **CHAR** attribute is the decimal point of the current language--a period in English. The **CHAROFF** attribute specifies the offset to the first occurrence of the alignment character. The attribute's value is a number in pixels or a percentage of the cell's width; **CHAROFF="50%"** centers the alignment character horizontally in a cell.

The **VALIGN** attribute specifies the vertical position of a cell's contents. Possible values are:

- **top**, which positions data at the top of the cell;
- **middle**, the default value, which centers the cell data vertically;
- **bottom**, which positions data at the bottom of the cell;
- **baseline**, which specifies that the first line of each cell in the row with **ALIGN=baseline** should occur on a common baseline.

The **WIDTH**, **HEIGHT**, **NOWRAP**, and **BGCOLOR** attributes are all [deprecated](#) in favor of [style sheets](#). **WIDTH** and **HEIGHT** suggest the cell's width and height in pixels. The boolean **NOWRAP** attribute tells visual browsers to disable word wrap for the cell, which can result in unnecessary horizontal scrolling depending on the user's window width and font size.

The **BGCOLOR** attribute suggests a background color for the cell. The combination of this attribute with **** can leave invisible or unreadable text on Netscape Navigator 2.x, which does not support **BGCOLOR** on table elements. **BGCOLOR** is dangerous even on supporting browsers, since most fail to override it when overriding other author-specified colors. [Style sheets](#) provide a safer, more flexible method of specifying a table's background color.

More Information

- [TD in W3C HTML 4.0 Recommendation](#)
- [TD in W3C HTML 3.2 Recommendation](#)
- [TD in WDG HTML 3.2 Reference](#)
- [TD in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

IMG - Inline Image



IMG - Inline Image

Syntax

Attribute
Specifications

^u
S
R
C
=
U
R
L
(
l
o
c
a
t
i
o
n
o
f
i
m
a
g
e
)
^u
A
L
T
=
T
e
x

t

(a l t e r n a t e

t e x t)

u

L O N G D E S C

=

U
R
I

(l i n k

t o

l o n g

d e s c r i p t i o n)

u

W

I
D
T
H
=
I
e
D
q
t
b

(
i
m
a
g
e

w
i
d
t
h
)

°
H
E
I
G
H
T
=
I
e
D
q
t
b

(
i
m
a
g
e

h
e
i
g
h
t
)

°
U
S
E
M
A

P
=
U
R
I
=

(c
l
i
e
n
t
-
s
i
d
e

i
m
a
g
e

m
a
p
)

u
I
S
M
A
P

(s
e
r
v
e
r
-
s
i
d
e

i
m
a
g
e

m
a
p
)

u

A
L
I
G
N
=
t
o
p
|
m
i
d
d
l
e
|
b
o
t
t
o
m
|
l
e
f
t
|
r
i
g
h
t

(
i
m
a
g
e

a
l
i
g
n
m
e
n
t
)

B
O
R
D
E
R
=
I

border

(link

border

width)

u

HSPACE =

padding

x

padding

(horizontal gutter)

u

V

S
P
A
C
E
=
P
i
X
E
I
S

(
v
e
r
t
i
c
a
l
g
u
t
t
e
r
)

u
c
o
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Emp
ty
Cont
aine
d in
Inlin
e
elem
ents,
bloc
k-

[level](#)
[elem](#)
[ents](#)
exce
pt
[PRE](#)

The **IMG** element specifies an *inline image*. The required **SRC** attribute specifies the location of the image. The image can be any format, though browsers generally only support GIF and JPEG images. Support for the [PNG](#) image format is growing slowly.

The required **ALT** attribute provides alternate text for those not loading images. Effective **ALT** text should generally give the *function* of the image rather than a *description* of the image. For example, **ALT="Welcome to XYZ Corp."** would be more appropriate than **ALT="XYZ Corp. Logo"** for a company's logo on its welcome page. Good **ALT** text is crucial to the document's accessibility for the significant portion of users who do not load images; see [Use of ALT texts in IMGs](#) for a thorough discussion.

The **LONGDESC** attribute gives the location of a long description of the image. This attribute should be used to provide a long description of an image where this would be useful. For example, a painting, graph, or corporate logo could be given a description so that blind and other text-only users can develop a mental picture of the image.

The **WIDTH** and **HEIGHT** attributes are most useful when they specify the exact dimensions of the image in pixels. This allows image-loading browsers to reserve the proper amount of space for the image and continue to render the rest of the document, thus giving the appearance of a faster-loading page.

Unfortunately, many graphical browsers will use these dimensions when not loading images, which can cause the **ALT** text to be cut off if the image is small or the **ALT** text is large. In such cases, authors may wish to leave off the **WIDTH** and **HEIGHT** attributes, depending on the importance of the **ALT** text and the placement of the image on the page (an image towards the end of the document without **WIDTH** and **HEIGHT** attributes will generally not noticeably slow the rendering of the page).

Authors can also specify different dimensions for the **WIDTH** and **HEIGHT** attributes, in which case browsers should scale the image. Percentages, relative to the horizontal or vertical space available (*not* relative to the image's natural size) can also be specified, though these are not as widely supported as pixel lengths. Since browsers typically do a poor job of scaling images, authors should avoid using **WIDTH** and **HEIGHT** for this purpose as much as possible. The **ALIGN** attribute, [deprecated](#) in HTML 4.0, specifies the alignment of the image. The values **top**, **middle**, and **bottom** specify the image's position with respect to surrounding content on its left and right. **ALIGN=middle** aligns the center of the image with the current baseline. To center the image horizontally on the page, place the image in a centered block, e.g.,

```
<H1 ALIGN=center><IMG SRC="logo.gif" ALT="Welcome to XYZ Company"></H1>
```

The other **ALIGN** values, **left** and **right**, specify a *floating* image; the image is placed at the left or right margin and content flows around it. To place content below the image, use **<BR CLEAR=left|right|all>** as appropriate.

The [vertical-align](#) and [float](#) properties of [Cascading Style Sheets](#) provide more flexible methods of aligning images.

The **BORDER** attribute, [deprecated](#) in HTML 4.0, specifies the width of the image's border. Specifying **BORDER=0** will eliminate the border around a linked image in most browsers, though some allow the user to override this. Authors should only use **BORDER=0** if the image would be clearly recognizable as a link, or as a method of de-emphasizing a link. For example:

```
<A HREF="reference/"><IMG SRC="icon/reference.gif" ALT="" WIDTH=90 HEIGHT=90  
BORDER=0>Web Authoring Reference</A>
```

The [deprecated HSPACE](#) and [VSPACE](#) attributes allow an author to suggest horizontal gutters and vertical gutters, respectively, around the image. The value must be in pixels and applies to both sides of the image. [Style sheets](#) provide more flexibility in specifying the space around images.

The [USEMAP](#) attribute is used with client-side image maps to give the location of the [map definition](#). While this value may be a full [URI](#)--allowing a single map definition to be applied to multiple pages--[Netscape Navigator](#) will only find map definitions in the same file, effectively limiting the [USEMAP](#) value to a fragment identifier such as ["#map"](#).

The [ISMAP](#) attribute is used with server-side image maps. When the [ISMAP](#) attribute is included with a linked image and the user clicks the image, the image coordinates clicked are sent to the server, from which a location can be returned. The method of handling the coordinates is server-dependent, but the [NCSA server's method](#) is most common.

Server-side image maps are better supported than client-side image maps, but almost all browsers today support both methods. Client-side image maps are generally preferred since they do not require an extra request to the server (and so are faster), and since they allow a usable menu to be provided to text-only users. Using both methods in combination is a good approach, since browsers supporting client-side image maps will use that method while older browsers will use the server-side image map. An example follows:

```
<A href="http://www.htmlhelp.com/cgi-bin/imagemap/sitemap.map"><IMG SRC="sitemap.gif"  
ALT="Site Map" ISMAP USEMAP="#map" WIDTH=100 HEIGHT=100></A>
```

A [MAP](#) element named [map](#) would have to be included in the same document.

More Information

- [IMG in W3C HTML 4.0 Recommendation](#)
- [IMG in W3C HTML 3.2 Recommendation](#)
- [IMG in HTML 2.0 Standard](#)
- [IMG in WDG HTML 3.2 Reference](#)
- [IMG in Learning HTML 3.2 by Examples](#)
- [Use of ALT texts in IMGs](#)
- [Image Use on the Web](#)
- [JPEG FAQ](#)
- [PNG \(Portable Network Graphics\) Home Page](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

MAP - Image Map



MAP - Image Map

Syntax
<MAP>...
</MAP>
Attribute
Specifications

NAME = CDATA (name of map) COMMON = at

t
r
i
b
u
t
e
s

Cont
ents
One
or
mor
e
blo
k-
level
elem
ents
or
one
or
mor
e
ARE
A
elem
ents
Cont
aine
d in
Inlin
e
elem
ents,
blo
k-
level
elem
ents

The **MAP** element defines a *client-side image map* for use with an **IMG** or **OBJECT**. **MAP**'s required **NAME** attribute is used as an anchor for the **USEMAP** attribute of the **IMG** or **OBJECT**. While a **MAP** element can define image maps embedded in other files, browsers typically only support client-side image maps with the **MAP** in the same file as the image.

MAP was originally defined to take one or more **AREA** elements that specified the coordinates of a clickable region on the image. An example follows:

```
<MAP NAME=mymap>  
<AREA href="http://www.htmlhelp.com/reference/" ALT="HTML and CSS Reference"  
COORDS="5,5,95,195">  
<AREA href="http://www.htmlhelp.com/design/" ALT="Design Guide"  
COORDS="105,5,195,195">  
<AREA href="http://www.htmlhelp.com/tools/" ALT="Tools" COORDS="205,5,295,195">  
</MAP>  
<IMG SRC="sitemap.gif" ALT="Site map" USEMAP="#mymap" WIDTH=300 HEIGHT=200>
```

HTML 4.0 extends the **MAP** element to take one or more [block-level elements](#) as an alternative to using [AREA](#) elements. Combined with the [OBJECT](#) element, this allows rich alternative content for those not loading images. However, due to poor and buggy browser support for [OBJECT](#), client-side image maps through the [IMG](#) element are more reliable.

When **MAP** is given within an **OBJECT**, its contents are not rendered on image-loading browsers. **MAP** may also be used outside the **OBJECT** element so that its contents are rendered.

The following example gives two images, one an alternate if the first type of image is not supported. The images share a single image map definition, which is included within the **OBJECT** element. The **MAP** element contains a menu of links to be rendered on browsers not loading images.

```
<OBJECT DATA="sitemap.png" USEMAP="#map" TYPE="image/png" TITLE="Site map" WIDTH=300
HEIGHT=200>
<OBJECT DATA="sitemap.gif" USEMAP="#map" TYPE="image/gif" TITLE="Site map" WIDTH=300
HEIGHT=200>
<MAP NAME=map>
<UL>
<LI><A href="http://www.htmlhelp.com/reference/" COORDS="5,5,95,195">HTML and CSS
Reference</A></LI>
<LI><A href="http://www.htmlhelp.com/design/" COORDS="105,5,195,195">Design
Guide</A></LI>
<LI><A href="http://www.htmlhelp.com/tools/index.html"
COORDS="205,5,295,195">Tools</A></LI>
</UL>
</MAP>
</OBJECT>
</OBJECT>
```

More Information

- [MAP in W3C HTML 4.0 Recommendation](#)
- [MAP in W3C HTML 3.2 Recommendation](#)
- [MAP in WDG HTML 3.2 Reference](#)
- [MAP in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

HTML 4.0 Common Attributes



HTML 4.0 Common Attributes

A number of attributes in HTML 4.0 are common to most elements. These attributes are divided into [core attributes](#), [internationalization attributes](#), and [scripting events](#).

Core Attributes

ID

The **ID** attribute uniquely identifies an element within a document. No two elements can have the same **ID** value in a single document. The attribute's value must begin with a letter in the range A-Z or a-z and may be followed by letters (A-Za-z), digits (0-9), hyphens ("-"), underscores ("_"), colons (":"), and periods (".").

The following example uses the **ID** attribute to identify each of the first two paragraphs of a document:

```
<P ID=firstp>My first paragraph.</P>  
<P ID=secondp>My second paragraph.</P>
```

The paragraphs in the example could have style rules associated with them through their **ID** attributes. The following [Cascading Style Sheet](#) defines unique colors for the two paragraphs:

```
P#firstp {  
    color: navy;  
    background: transparent  
}  
  
P#secondp {  
    color: black;  
    background: transparent  
}
```

The paragraphs in the initial example could also be used as a target anchor for links:

```
<P>See <A HREF="#firstp">the opening paragraph</A> for more information.</P>
```

Note that most browsers do not support the **ID** attribute for link anchors. For current browsers, authors should use [<A NAME> . . .](#) within the element instead of **ID**.

Since **ID** and **NAME** share the same name space, authors cannot use the same value for an **ID** attribute and a **NAME** attribute in the same document. Also note that while **NAME** may contain [entities](#), the **ID** attribute value may not.

CLASS

The **CLASS** attribute specifies the element to be a member of one or more classes. Classes allow authors to define specific *kinds* of a given element. For example, an author could use `<CODE CLASS=Java>` when giving Java code and `<CODE CLASS=Perl>` when giving Perl code.

Unlike with the **ID** attribute, any number of elements can share the same class. An element may also belong to multiple classes; the **CLASS** attribute value is a space-separated list of class names.

Note that most current browsers do not support multiple classes. Such browsers typically ignore a **CLASS** attribute that specifies multiple classes.

The **CLASS** attribute is particularly useful when combined with style sheets. For example, consider the following navigation bar:

```
<DIV CLASS=navbar>
<P><A href="http://www.htmlhelp.com/">Home</A> | <A HREF=".">Index</A> | <A
href="http://www.htmlhelp.com/search.html">Search</A></P>
<P><A href="http://www.htmlhelp.com/"><IMG SRC="logo.gif" ALT="" TITLE="WDG
Logo"></A></P>
</DIV>
```

This example's use of the **CLASS** attribute allows style rules to easily be added. The following [Cascading Style Sheet](#) suggests a presentation for the preceding example:

```
.navbar {
  margin-top: 2em;
  padding-top: 1em;
  border-top: solid thin navy
}

.navbar IMG { float: right }

@media print {
  .navbar { display: none }
}
```

STYLE

The **STYLE** attribute allows authors to specify style rules *inline* for a single occurrence of an element. An example follows:

```
<P>A popular font for on-screen reading is <SPAN STYLE="font-family:
Verdana">Verdana</SPAN>.</P>
```

When the **STYLE** attribute is used, a default style sheet language must be specified for the document by setting the **Content-Style-Type** HTTP header to the media type of the style sheet language. The previous example could use the following **META** element in the document's **HEAD**:

```
<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">
```

In most cases, use of the **CLASS** or **ID** attributes is a better choice than using **STYLE** since **ID** and **CLASS** can be selectively applied to different media and since they provide a separation of content and presentation that often simplifies maintenance.

TITLE

The **TITLE** attribute provides a title for an element and is commonly implemented as a "tooltip" on visual browsers, though many browsers lack support for **TITLE**. The attribute is most useful with **A**, **LINK**, **IMG**, and **OBJECT** elements, where it provides a title for the linked or embedded resource. Some examples follow:

```
⌘ <A HREF="mailto:liam@htmlhelp.com" TITLE="Feedback on HTML 4.0
Reference">liam@htmlhelp.com</A>
⌘ <A HREF="http://www-genome.wi.mit.edu/ftp/pub/software/WWW/cgi_docs.html"
TITLE="CGI.pm - a Perl5 CGI Library">CGI.pm</A>
⌘ <LINK REL=Alternate HREF="index.fr.html" HREFLANG=fr LANG=fr TITLE="Version
française">
⌘ <OBJECT CLASSID="java:Yahtzee.class" CODETYPE="application/java" WIDTH=400
HEIGHT=250 STANDBY="Ready to play Yahtzee?" TITLE="My Yahtzee Game">
<IMG SRC="yahtzee.gif" ALT="" TITLE="A Yahtzee animation">
Yahtzee is my <EM>favorite</EM> game!
</OBJECT>
```

TITLE is also helpful with the **ABBR** and **ACRONYM** elements to provide the long form of the abbreviation. Examples:

```
⌘ He weighs 180 <ABBR TITLE=pounds>lbs.</ABBR>
⌘ <ABBR TITLE="Parti Québécois" LANG=fr-CA>PQ</ABBR>
⌘ <ACRONYM TITLE="North Atlantic Treaty Organization">NATO</ACRONYM>
```

Internationalization Attributes

LANG

The **LANG** attribute specifies the language of an element's attribute values and its content, including all contained elements that do not specify their own **LANG** attribute. While the **LANG** attribute is not widely supported, its use may help search engines index a document by its language while allowing speech synthesizers to use language-dependent pronunciation rules. As well, visual browsers can use the language's proper quotation marks when rendering the **Q** element.

The attribute value is case-insensitive, and should be specified according to [RFC 1766](#); examples include **en** for English, **en-US** for American English, and **ja** for Japanese. Whitespace is not allowed in the language code.

Use of the **LANG** attribute also allows authors to easily change the [style](#) of text depending on the language. For example, a bilingual document may have one language in italics if rendered visually or a different voice if rendered

aurally. The HTML of such a document might be as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
  "http://www.w3.org/TR/REC-html40/strict.dtd">
<TITLE>Welcome - Bienvenue</TITLE>
<H1>
  <SPAN LANG=en>Welcome</SPAN> -
  <SPAN LANG=fr>Bienvenue</SPAN>
</H1>
<P LANG=en>This paragraph is in English.</P>
<P LANG=fr>Ce paragraphe est en français.</P>
...
```

A document's primary language may be set using the **LANG** attribute on the [HTML](#) element, or, alternatively, by using the **Content-Language** HTTP header.

DIR

The **DIR** attribute specifies the directionality of text--left-to-right (**DIR=ltr**, the default) or right-to-left (**DIR=rtl**). Characters in [Unicode](#) are assigned a directionality, left-to-right or right-to-left, to allow the text to be rendered properly. For example, while English characters are presented left-to-right, Hebrew characters are presented right-to-left.

Unicode defines a *bidirectional algorithm* that must be applied whenever a document contains right-to-left characters. While this algorithm usually gives the proper presentation, some situations leave directionally neutral text and require the **DIR** attribute to specify the base directionality.

Text is often directionally neutral when there are multiple embeddings of content with a different directionality. For example, an English sentence that contains a Hebrew phrase that contains an English quotation would require the **DIR** attribute to define the directionality of the Hebrew phrase. The Hebrew phrase, including the English quotation, should be contained within a [SPAN](#) element with **DIR=rtl**.

Common Scripting Events

A number of attributes that define client-side scripting events are common to most elements. The attribute value is a script--typically a function call or a few short statements--that is executed when the event occurs. The value may contain [entities](#) (e.g., **"**).

The following example features JavaScript code to handle two events of a submit button, giving the user a reminder in the status bar when the mouse moves over the button and clearing the status bar when the mouse moves away. Note that the attribute values are delimited by single quotes since double quotes are used within them.

```
<INPUT TYPE=submit ONMOUSEOVER='window.status="Did you fill in all required fields?";'
ONMOUSEOUT='window.status="";'>
```

When an event attribute is used, a default scripting language must be specified for the document by setting the **Content-Script-Type** HTTP header to the media type of the scripting language. The previous example could use the following [META](#) element in the document's [HEAD](#):

```
<META HTTP-EQUIV="Content-Script-Type" CONTENT="text/javascript">
```

The common event attributes are device-dependent and largely tailored for the graphical user interface. The available events are as follows:

- **ONCLICK**, when the mouse button is clicked on an element;
- **ONDBLCLICK**, when the mouse button is double-clicked on an element;
- **ONMOUSEDOWN**, when the mouse button is pressed over an element;
- **ONMOUSEUP**, when the mouse button is released over an element;
- **ONMOUSEOVER**, when the mouse is moved onto an element;
- **ONMOUSEMOVE**, when the mouse is moved while over an element;
- **ONMOUSEOUT**, when the mouse is moved away from an element;
- **ONKEYPRESS**, when a key is pressed and released over an element;
- **ONKEYDOWN**, when a key is pressed down over an element;
- **ONKEYUP**, when a key is released over an element.

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

HTML 4.0 Entities for Symbols and Greek Letters



Entities for Symbols and Greek Letters

The following table gives the character entity reference, decimal character reference, and hexadecimal character reference for symbols and Greek letters. [Glyphs](#) of the characters are available at the [Unicode Consortium](#).

Browser support for these entities is generally quite poor, but recent browsers support some of the character entity references and decimal character references.

Note that hexadecimal character references will cause errors with current validators (except the [WDG HTML Validator](#)) since these references are defined in a recent revision to the SGML standard that is not yet supported by most validators.

Character	Entity	Decimal	Hex
Latin small f with hook = function = florin	ƒ	192;	Å
Greek capital letter alpha	Α	13;	9
Greek capital letter beta	Β	14;	2
Greek capital letter gamma	Γ	15;	3

ma	
Gre	&Del	 &#x
ek	ta; 16; 394;
capit	
al	
lette	
r	
delta	
Gre	&Ep 	 &#x
ek	silon 17; 395;
capit	;
al	
lette	
r	
epsil	
on	
Gre	&Zet	 &#x
ek	a; 18; 396;
capit	
al	
lette	
r	
zeta	
Gre	&Eta	 &#x
ek	; 19; 397;
capit	
al	
lette	
r eta	
Gre	&Th 	 &#x
ek	eta; 20; 398;
capit	
al	
lette	
r	
thet	
a	
Gre	&lot 	 &#x
ek	a; 21; 399;
capit	
al	
lette	
r	
iota	
Gre	&Ka 	 &#x
ek	ppa; 22; 39A;
capit	
al	
lette	
r	
kapp	
a	
Gre	&La 	 &#x
ek	mbd 23; 39B;
capit	a;
al	
lette	
r	
lamb	
da	
Gre	&Mu 	 &#x
ek	; 24; 39C;

capit
al
lette
r mu
Gre &Nu 	 &#x
ek ; 25; 39D;
capit
al
lette
r nu
Gre Ξ 	 &#x
ek 26; 39E;
capit
al
lette
r xi
Gre &O 	 &#x
ek micr 27; 39F;
capit on;
al
lette
r
omic
ron
Gre Π 	 &#x
ek 28; 3A0;
capit
al
lette
r pi
Gre &Rh 	 &#x
ek o; 29; 3A1;
capit
al
lette
r rho
Gre &Sig 	 &#x
ek ma; 31; 3A3;
capit
al
lette
r
sigm
a
Gre &Ta 	 &#x
ek u; 32; 3A4;
capit
al
lette
r tau
Gre &Up 	 &#x
ek silon 33; 3A5;
capit ;
al
lette
r
upsil
on
Gre &Phi 	 &#x
ek ; 34; 3A6;
capit
al

lette r phi Gre ek capit al lette r chi Gre ek capit al lette r psi Gre ek capit al lette r ome ga Gre ek smal l lette r alph a Gre ek smal l lette r beta Gre ek smal l lette r gam ma Gre ek smal l lette r delta Gre ek smal l lette r epsil on Gre ek	&Chi	 &#x ; 35; 3A7;
	&Psi	 &#x ; 36; 3A8;
	&O 	 &#x meg 37; 3A9; a;
	&alp 	 &#x ha; 45; 3B1;
	&bet 	 &#x a; 46; 3B2;
	&ga 	 &#x mm 47; 3B3; a;
	&del 	 &#x ta; 48; 3B4;
	&ep 	 &#x silon 49; 3B5; ;
	&zet 	 &#x a; 50; 3B6;

small letter zeta	&eta 	 &#x ; 51; 3B7;
small letter eta	&the 	 &#x ta; 52; 3B8;
small letter theta	&iot 	 &#x a; 53; 3B9;
small letter iota	&ka 	 &#x ppa; 54; 3BA;
small letter kappa	&la 	 &#x mbd 55; 3BB; a;
small letter lambda	&mu 	 &#x ; 56; 3BC ;
small letter mu	ν 	 &#x 57; 3BD ;
small letter nu	ξ 	 &#x 58; 3BE;
small letter xi	&om 	 &#x icron59; 3BF;

small letter omicron Greek small letter pi Greek small letter rho Greek small letter sigma Greek small letter tau Greek small letter upsilon Greek small letter phi Greek small letter chi

;

π ; 60; 3C0;

ρ ; 61; 3C1;

σ ; 62; 3C2;

σ ; 63; 3C3;

τ ; 64; 3C4;

υ ; 65; 3C5;

ϕ ; 66; 3C6;

χ ; 67; 3C7;

ψ ; 68; 3C8;

ek	; 68; 3C8;
small	
letter	
psi	
Omega	&om 	 &#x
ek	ega; 69; 3C9;
small	
letter	
rome	
gamma	
Gamma	&the 	 &#x
ek	tasy 77; 3D1;
small	m;
letter	
theta	
alpha	
symbol	
Gamma	&up 	 &#x
ek	sih; 78; 3D2;
upsilon	
on	
with	
hook	
symbol	
bol	
Gamma	&piv 	 &#x
ek	; 82; 3D6;
pi	
symbol	
bullet	&bul &#x
t =	l; 226; 202
black	2;
small	
circle	
horizontal	&hel &#x
ellipsis =	lip; 230; 202
three	6;
dot	
leader	
prime =	&pri &#x
minutes =	me; 242; 203
feet	2;
double	&Pri &#x
prime =	me; 243; 203
seconds	3;

nds	
=	
inch	
es	
overl	&oli &#x
ine =	ne; 254; 203
spac	E;
ing	
over	
scor	
e	
fracti	&fra &#x
on	sl; 260; 204
slas	4;
h	
scrip	&wei &#x
t	erp; 472; 2118
capit	;
al P	
=	
pow	
er	
set =	
Wei	
erstr	
ass	
p	
blac	&im &#x
klett	age; 465; 2111
er	;
capit	
al I =	
imag	
inary	
part	
blac	&rea &#x
klett	l; 476; 211
er	C;
capit	
al R	
=	
real	
part	
sym	
bol	
trad	&tra &#x
e	de; 482; 212
mar	2;
k	
sign	
alef	&ale &#x
sym	fsym501; 213
bol =	;
5;	
first	
tran	
sfinite	
e	
cardi	
nal	
leftw	&larr &#x
ards	; 592; 219
arro	0;

w	
upward arrow	↑ 593; 2191;
rightward arrow	→ 594; 2192;
downward arrow	↓ 595; 2193;
leftward arrow	↔ 596; 2194;
downward arrow with corner leftwards	↵ 629; 21B5;
= carriage return	
double arrow	← 656; 21D0;
double arrow up	↑ 657; 21D1;
double arrow right	→ 658; 21D2;
double arrow down	↓ 659; 21D3;
double arrow left	↔ 660; 21D

double	\mathbb{R}^4 ;
arrow	
forall	\forall ;
partial	∂ ;
differential	d ;
there exists	\exists ;
empty set =	\emptyset ;
null set =	
diameter	
nabla =	∇ ;
backward	
difference	
element of	\in ;
not an element of	\notin ;
contains as member	\ni ;
n-ary product =	\prod ;
product sign	
n-ary summation	\sum ;
minus sign	$-$;
asterisk	$*$;
operator	
squareroot	$\sqrt{\quad}$;

re	ic; 730; 221
root	A;
=	
radical	
sign	
proportional	\propto ; 733; 221
to	D;
infinity	∞ ; 734; 221
	E;
angle	\angle ; 736; 222
	0;
logical	$\&$; 743; 222
and	7;
=	
wedge	
ge	
logical or	\vee ; 744; 222
=	8;
vee	
intersection =	\cap ; 745; 222
cap	9;
union =	\cup ; 746; 222
cup	A;
integral	\int ; 747; 222
	B;
therefore	\therefore ; 756; 223
e	4;
tilde	\sim ; 764; 223
operator	C;
=	
varies	
with	
=	
similar to	
approximatively	\approx ; 773; 224
equal to	5;
almost	\approx ; 776; 224
equal to	8;
=	
asymptotic	

to	
not	\neq ; 800; 226
equal to	$=$; 800; 226
identical	\equiv ; 801; 226
to	
less-than	$<$; 804; 226
or	
equal to	
greater-than	$>$; 805; 226
or	
equal to	
subset of	\subset ; 834; 228
superset of	\supset ; 835; 228
of	
not a	$\not\subset$; 836; 228
subset of	
subset of	\subseteq ; 838; 228
or	
equal to	
superset of or	\supseteq ; 839; 228
equal to	
circled plus	\oplus ; 853; 229
=	
direct	
sum	
circled	\otimes ; 855; 229
time	
s =	
vect	
or	
product	
up	\perp ; 869; 22A
tack	
=	
orthogonal	
to =	
perpendicular	

cular	
dot	&sd &#x
oper	ot; 901; 22C
ator	5;
left	&lcei &#x
ceili	l; 968; 230
ng =	8;
APL	
upsti	
le	
right	&rce &#x
ceili	il; 969; 230
ng	9;
left	&lfo &#x
floor	or; 970; 230
=	A;
APL	
dow	
nstil	
e	
right	&rflo &#x
floor	or; 971; 230
	B;
left-	&lan 	 &#x
point	g; 001; 232
ing	9;
angl	
e	
brac	
ket =	
bra	
right	&ran 	 &#x
-	g; 002; 232
point	A;
ing	
angl	
e	
brac	
ket =	
ket	
loze	&loz 	 &#x
nge	; 674; 25C
	A;
blac	&sp 	 &#x
k	ades824; 266
spad	; 0;
e	
suit	
blac	&clu 	 &#x
k	bs; 827; 266
club	3;
suit	
=	
sha	
mro	
ck	
blac	&he 	 &#x
k	arts; 829; 266
hear	5;
t suit	
=	
vale	

ntine
blac &dia 	 &#x
k ms; 830; 266
diam 6;
ond
suit

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Latin-1 Characters](#) ~

[Other Special Characters](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

HTML 4.0 Special Entities



Special Entities

The following table gives the character entity reference, decimal character reference, and hexadecimal character reference for markup-significant and internationalization characters. [Glyphs](#) of the characters are available at the [Unicode Consortium](#).

With the exception of [HTML 2.0's](#) `"`, `&`, `<`, and `>`, browser support for these entities is generally quite poor, but recent browsers support some of the character entity references and decimal character references.

Note that hexadecimal character references will cause errors with current validators (except the [WDG HTML Validator](#)) since these references are defined in a recent revision to the SGML standard that is not yet supported by most validators.

Character	Entity	Decimal	Heximal
quotation mark = APL quote	<code>&quot;</code>	<code>&#34;</code>	<code>&#x22;</code>
ampersand	<code>&amp;</code>	<code>&#38;</code>	<code>&#x26;</code>
less-than sign	<code>&lt;</code>	<code>&#60;</code>	<code>&#x3C;</code>
greater-than sign	<code>&gt;</code>	<code>&#62;</code>	<code>&#x3E;</code>
Latin capital ligature OE	<code>&OElig;</code>	<code>&#38;</code>	<code>&#x152;</code>
Latin small ligature oe	<code>&oelig;</code>	<code>&#39;</code>	<code>&#x153;</code>
Latin	<code>&Scaron;</code>	<code>&#38;</code>	<code>&#x154;</code>

capit	aron 52; 160;
al	;
lette	
r S	
with	
caro	
n	
Latin	&sc &#x
smal	aron 53; 161;
l	;
lette	
r s	
with	
caro	
n	
Latin	&Yu &#x
capit	ml; 76; 178;
al	
lette	
r Y	
with	
diae	
resis	
modi	&circ &#x
fier	; 10; 2C6;
lette	
r	
circu	
mfle	
x	
acce	
nt	
smal	&tild &#x
l	e; 32; 2DC
tilde	;
en	&en &#x
spac	sp; 194; 200
e	2;
em	&em &#x
spac	sp; 195; 200
e	3;
thin	&thi &#x
spac	nsp; 201; 200
e	9;
zero	&zw &#x
widt	nj; 204; 200
h	C;
non-	
joine	
r	
zero	&zwj &#x
widt	; 205; 200
h	D;
joine	
r	
left-	&lrm &#x
to-	; 206; 200
right	E;
mar	
k	
right	&rlm &#x
-to-	; 207; 200

left mar k	F;
en dash	– 211; 201 3;
em dash	— 212; 201 4;
left singl e	‘ 216; 201 8;
quot ation mar k	
right singl e	’ 217; 201 9;
quot ation mar k	
singl e low- 9	‚ 218; 201 A;
quot ation mar k	
left dou ble	“ 220; 201 C;
quot ation mar k	
right dou ble	” 221; 201 D;
quot ation mar k	
dou ble low- 9	„ 222; 201 E;
quot ation mar k	
dag ger	&dagger 224; 202 ;
dou ble dag ger	&Dagger 225; 202 ;
per mille	‰ 240; 203

sign	0;
singl	&lsa &#x
e	quo; 249; 203
left-	9;
point	
ing	
angl	
e	
quot	
ation	
mar	
k	
singl	&rsa &#x
e	quo; 250; 203
right	A;
-	
point	
ing	
angl	
e	
quot	
ation	
mar	
k	
euro	&eur &#x
sign	o; 364; 20A
	C;

Maintained by [Liam Quinn <liam@htmlhelp.com>](mailto:liam@htmlhelp.com)



[HTML 4.0 Reference](#) ~ [Latin-1 Characters](#) ~

[Symbols and Greek Letters](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra <weballey@usa.net>](mailto:weballey@usa.net).

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

LINK - Document Relationship



LINK - Document Relationship

Syntax
<LINK>
Attribute
Specifications

REL = LinkTypes

(relationship to link)

° R E V = [F i l e](#)

(r e l a t i o n s h i p
f r o m
l i n k)

° H R E F = [C](#)

(h y p e r t e x t

r
e
f
e
r
e
n
c
e
)

“
T
Y
P
E
=
C
o
p
p
r
t
e
p
r
t
y
p
e

(
c
o
n
t
e
n
t
-
t
y
p
e
o
f
l
i
n
k
)

“
T
A
R
G
E
T
=
F
r

[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)

(
f
r
a
m
e

t
o

r
e
n
d
e
r

l
i
n
k

i
n
)

“
M
E
D
I
A
=

[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)
[|||](#)

(
m
e
d
i
a

a
p

p
r
o
p
r
i
a
t
e

f
o
r

l
i
n
k
)
"
H
R
E
F
L
A
N
G
=
[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[6](#)
[7](#)
[8](#)
[9](#)
[0](#)
[1](#)
[2](#)
(
l
a
n
g
u
a
g
e

o
f

l
i
n
k
)
"

C H A R S E T = C r a r s e t (c h a r a c t e r e n c o d i n g o f l i n k) , c o m m o n = a t t r i b u t e

S
Contents
Empty
Contained in
HEAD
D

The **LINK** element defines *document relationships*. Any number of **LINK** elements may be contained in the HEAD of a document. Many browsers lack support for **LINK**, so authors should not depend on the browser making the links available to the user.

The **REL** and **REV** attributes define the nature of the relationship between the documents and the linked resource. **REL** defines a link relationship from the current document to the linked resource while **REV** defines a relationship in the opposite direction. For example,

```
<LINK REL=Glossary HREF="foo.html">
```

indicates that **foo.html** is a glossary for the current document while

```
<LINK REV=Subsection HREF="bar.html">
```

indicates that the current document is a subsection of **bar.html**. The value of the **REL** and **REV** attributes is a space-separated list of link types.

Commonly used relationships include the next or previous document in a sequence, the starting page in a collection of documents, a document with copyright information, and information about the author. A document could define these relationships as follows:

```
<LINK REL=Prev HREF="base.html" TITLE="BASE - Document Base URI">  
<LINK REL=Next HREF="meta.html" TITLE="META - Metadata">  
<LINK REL=Start href="../index.html" TITLE="HTML 4.0 Reference">  
<LINK REL=Copyright href="http://www.htmlhelp.com/copyright.html" TITLE="Copyright  
Notice">  
<LINK REV=Made HREF="mailto:liam@htmlhelp.com" TITLE="HTML 4.0 Reference Feedback">
```

While the value of **REL** and **REV** is case-insensitive, the Lynx browser renders the relationship exactly as given by the author. Authors should therefore be consistent in their case, and may wish to capitalize the first letter while using lowercase for the rest.

Authors can also use the **LINK** element to apply an external style sheet. **REL=StyleSheet** specifies a *persistent* or *preferred* style while **REL="Alternate StyleSheet"** defines an *alternate* style. A *persistent* style is one that is always applied when style sheets are enabled. The absence of the **TITLE** attribute indicates a persistent style.

A *preferred* style is one that is automatically applied. The combination of **REL=StyleSheet** and a **TITLE** attribute specifies a preferred style. Authors cannot specify more than one preferred style.

An *alternate* style is indicated by **REL="Alternate StyleSheet"**. The user could choose to replace the preferred style sheet with an alternate one, though current browsers generally lack the ability to choose alternate styles.

A single style may also be given through multiple style sheets:

```
<LINK REL=StyleSheet HREF="basics.css" TITLE="Contemporary" TYPE="text/css">
<LINK REL=StyleSheet HREF="tables.css" TITLE="Contemporary" TYPE="text/css">
<LINK REL=StyleSheet HREF="forms.css" TITLE="Contemporary" TYPE="text/css">
```

In this example, three style sheets are combined into one "Contemporary" style that is applied as a preferred style sheet. To combine multiple style sheets into a single style, each style sheet's **LINK** must use the same **TITLE**.

LINK's MEDIA attribute specifies the media for which the linked resource is designed. With **REL=StyleSheet**, this allows authors to restrict a style sheet to certain output devices, such as printers or aural browsers. The attribute's value is a comma-separated list of media descriptors. The following media descriptors are defined in HTML 4.0 and are case-sensitive:

- **screen** (the default), for non-paged computer screens;
- **tty**, for fixed-pitch character grid displays (such as the display used by [Lynx](#));
- **tv**, for television-type devices with low resolution and limited scrollability;
- **projection**, for projectors;
- **handheld**, for handheld devices (characterized by a small, monochrome display and limited bandwidth);
- **print**, for output to a printer;
- **braille**, for braille tactile feedback devices;
- **aural**, for speech synthesizers;
- **all**, for all devices.

[Netscape Navigator 4.x](#) incorrectly ignores any style sheet linked with a **MEDIA** value other than **screen**. For example, **MEDIA="screen, projection"** will cause the style sheet to be ignored by Navigator 4.x, even if the presentation device is a computer screen. Navigator 4.x also ignores style sheets declared with **MEDIA=all**. Most other browsers ignore the **MEDIA** attribute.

The optional **HREFLANG** and **CHARSET** attributes of **LINK** give the language and character encoding, respectively, of the link. The language should be specified according to [RFC 1766](#); examples include **en** for English, **en-US** for American English, and **ja** for Japanese. Examples of character encodings include **ISO-8859-1**, **SHIFT_JIS**, and **UTF-8**.

The **Alternate** link relationship defines an alternate version of the document. Translations of a page can be identified by using **REL=Alternate** along with the **HREFLANG** attribute. Versions of the page tailored for specific media can be provided by combining **REL=Alternate** with the **MEDIA** attribute. Some examples follow:

```
<LINK REL=Alternate HREF="index.fr.html" HREFLANG=fr LANG=fr TITLE="Version
française">
<LINK REL=Alternate HREF="index.ja.html" HREFLANG=ja CHARSET="SHIFT_JIS"
TITLE="Japanese version">
<LINK REL=Alternate href="http://www.htmlhelp.com/distribution/html40.pdf"
TYPE="application/pdf" MEDIA=print TITLE="PDF version">
```

Note that the **LANG** and **DIR** attributes apply to the text of the **TITLE** attribute, not to the content of the link. The **TARGET** attribute is used with **frames** to specify in which frame the link should be rendered. If no frame with such a name exists, the link is rendered in a new window unless overridden by the user. Special frame names begin

with an underscore:

- **_blank** renders the link in a new, unnamed window
- **_self** renders the link in the current frame (useful for overriding a **BASE TARGET**)
- **_parent** renders the link in the immediate **FRAMESET** parent
- **_top** renders the link in the full, unframed window

More Information

- [LINK in W3C HTML 4.0 Recommendation](#)
- [LINK in W3C HTML 3.2 Recommendation](#)
- [LINK in HTML 2.0 Standard](#)
- [LINK in WDG HTML 3.2 Reference](#)
- [LINK in Learning HTML 3.2 by Examples](#)
- [Metadata at W3C](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

STYLE - Embedded Style Sheet



STYLE - Embedded Style Sheet

Syntax
<STYLE>
...</STYL
E>
Attri
bute
Spe
cific
ation
s

T
Y
P
E
=
C
O
N
T
E
N
T
T
Y
P
E

(
c
o
n
t
e
n
t
-
t
y
p
e

o
f

s
t
y
l
e

l
a
n
g
u
a
g
e
)

»

M
E
D
I
A
=
M
e
d
i
a
i
s
t
r
y
e
s
t
i
c

(
m
e
d
i
a

t
o

a
p
p
l
y

s
t
y
l
e

t
o
)

»

T
I

T
L
E
=
T
e
x
t

(
t
i
t
l
e

o
f

s
t
y
l
e

s
h
e
e
t
)

=
i
n
t
e
r
n
a
t
i
o
n
a
l
i
z
a
t
i
o
n
=
a
t
t
r
i
b
u
t

e
s

(
f
o
r

t
h
e

T
I
T
L
E
)

Cont
ents
An
emb
edd
ed
style
shee
t
Cont
aine
d in
HEA
D

The **STYLE** element embeds a *style sheet* in the document. Any number of **STYLE** elements may be contained in the HEAD of a document.

The required **TYPE** attribute of **STYLE** is used to specify the Internet media type of the style language. For [Cascading Style Sheets](#), the **TYPE** attribute value should be **text/css**.

The optional **TITLE** attribute gives a title for the style sheet. Without a **TITLE** attribute, the style sheet is always applied when style sheets are enabled. With a **TITLE** attribute, the style sheet is automatically applied but the user may choose to disable the style sheet while keeping or enabling other style sheets. Style sheets with the same title are considered to be the same style sheet.

Most current browsers ignore the **TITLE** attribute on style sheets and do not allow the user to selectively enable or disable individual style sheets.

The **MEDIA** attribute specifies the media on which the style sheet should be applied. This allows authors to restrict a style sheet to certain output devices, such as printers or aural browsers. The attribute's value is a comma-separated list of media descriptors. The following media descriptors are defined in HTML 4.0 and are case-sensitive:

- **screen** (the default), for non-paged computer screens;
- **tty**, for fixed-pitch character grid displays (such as the display used by [Lynx](#));
- **tv**, for television-type devices with low resolution and limited scrollability;

- „**projection**, for projectors;
- „**handheld**, for handheld devices (characterized by a small, monochrome display and limited bandwidth);
- „**print**, for output to a printer;
- „**braille**, for braille tactile feedback devices;
- „**aural**, for speech synthesizers;
- „**all**, for all devices.

[Netscape Navigator 4.x](#) incorrectly ignores any **STYLE** element with a **MEDIA** value other than **screen**. For example, **MEDIA="screen, projection"** will cause the style sheet to be ignored by Navigator 4.x, even if the presentation device is a computer screen. Navigator 4.x also ignores style sheets declared with **MEDIA=all**. Most other browsers ignore the **MEDIA** attribute.

An example of an embedded style sheet follows:

```
<STYLE TYPE="text/css" MEDIA=screen>
<!--
  BODY { background: url(foo.gif) red; color: black }
  P EM { background: yellow; color: black }
  .note { margin-left: 5em; margin-right: 5em }
-->
</STYLE>
```

Pre-[HTML 3.2](#) browsers, unaware of the **STYLE** element, would normally show its contents as if they were part of the **BODY**, thus making the style sheet visible to the user. To prevent this, style languages like [CSS](#) allow the style sheet to be contained within an SGML comment (**<!-- comment -->**), as in the preceding example.

An embedded style sheet should be used when a single document has a unique style. If the same style sheet is used in multiple documents, then an [external style sheet](#) would be more appropriate.

More Information

- „[STYLE in W3C HTML 4.0 Recommendation](#)
- „[STYLE in W3C HTML 3.2 Recommendation](#)
- „[STYLE in WDG HTML 3.2 Reference](#)
- „[STYLE in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

SCRIPT - Client-side Script



SCRIPT - Client-side Script

Syntax
<SCRIPT
>...</SCR
IPT>
Attri
bute
Spe
cific
ation
s

u
T
Y
P
E
=
C
o
n
t
e
n
t
T
Y
P
E
(
c
o
n
t
e
n
t
-
t
y
p
e
o
f

s
c
r
i
p
t
i
n
g

l
a
n
g
u
a
g
e
)

«
L
A
N
G
U
A
G
E
=
C
D
A
T
A

(
s
c
r
i
p
t
i
n
g

l
a
n
g
u
a
g
e

n
a
m
e
)

«

S
R
C
=
DIRECTORIES
(
e
x
t
e
r
n
a
l
s
c
r
i
p
t
l
o
c
a
t
i
o
n
)
C
H
A
R
S
E
T
=
CHARSET
(
c
h
a
r
a
c
t
e
r

e
n
c
o
d
i
n
g

o
f

e
x
t
e
r
n
a
l
s
c
r
i
p
t
)

«
D
E
F
E
R

(
s
c
r
i
p
t

e
x
e
c
u
t
i
o
n

m
a
y

w
a
i
t
)

Contents
An embedded script contained in [HEAD](#), [inline elements](#), [block-level elements](#)

The **SCRIPT** element includes a *client-side script* in the document. Client-side scripts allow greater interactivity in a document by responding to user events. For example, a script could be used to check the user's form input prior to submission to provide immediate notice of any errors by the user.

Note that not all browsers support client-side scripting, and supporting browsers allow the user to disable scripting, so authors should avoid dependence on client-side scripting wherever possible. The **NOSCRIPT** element can be used to provide content for browsers that do not support client-side scripting or have it disabled. In the case of form validation, any error checking done by the client-side script should be repeated by the CGI script or Java servlet that handles the submission at the server.

Also note that different browsers support different variants of scripting languages with different bugs. Authors are encouraged to check their scripts on as many browsers as possible. Browsers that support client-side scripting include [Netscape Navigator](#) 2.0 and up, [Microsoft Internet Explorer](#) 3.0 and up, and [Opera](#) 3.0 and up.

The required **TYPE** attribute of **SCRIPT** specifies the media type of the scripting language, e.g., **text/javascript**. However, most browsers only support the [deprecated LANGUAGE](#) attribute, which specifies the language name. Examples of supported **LANGUAGE** values include **JavaScript**, **JavaScript1.1**, and **VBScript**. The values are not case sensitive.

Browsers will ignore scripts with **LANGUAGE** values that they do not support. For example, Netscape Navigator 3.0 will execute scripts with **LANGUAGE="JavaScript"** or **LANGUAGE="JavaScript1.1"** but will ignore scripts with **LANGUAGE="JavaScript1.2"** or **LANGUAGE="VBScript"**.

In the absence of the **LANGUAGE** attribute, browsers typically assume that the language is JavaScript 1.0. As there is no established convention for indicating the version of JavaScript via the **TYPE** attribute, we recommend that authors continue to use **LANGUAGE** to specify the language and version where it differs from JavaScript 1.0.

An *embedded script* is given as the content of the **SCRIPT** element. The **SRC** attribute allows authors to reuse code by specifying an *external script*. The optional **CHARSET** attribute gives the character encoding of the external script (typically **ISO-8859-1**). If the browser is unable to fetch the external script it will execute any embedded script; otherwise it will ignore the embedded script. An example follows:

```
<SCRIPT TYPE="text/javascript" SRC="foo.js" CHARSET="ISO-8859-1">
<!--
  // embedded script, only executed if foo.js is unavailable
  // -->
</SCRIPT>
```

Netscape Navigator requires that external scripts be served with a Content-Type of **application/x-javascript**.

The **DEFER** attribute indicates that the browser may wait to parse the script until the rest of the document has been rendered. Scripts that use **DEFER** must not generate any document content, and should not be required to respond to user events (e.g., form submission) that may occur while the document is loading. The **DEFER** attribute can be useful for delaying scripts that pre-load images or harass the user with scrolling messages in the status bar, though current browsers do not generally support this attribute.

The **SCRIPT** element may occur any number of times in the document **HEAD** or **BODY**. Typically the **SCRIPT** element is used in the **HEAD** unless it generates **BODY** content.

Pre-[HTML 3.2](#) browsers, unaware of the **SCRIPT** element, will treat the content of **SCRIPT** as normal HTML. To make these browsers ignore the **SCRIPT**'s content, scripting languages generally allow SGML comments to be used around an embedded script. For example:

```
<SCRIPT TYPE="text/javascript">
<!-- comment to end of line
  document.write("foo");
  // comment to end of line -->
</SCRIPT>
```

Note that "-->" is contained within a JavaScript single-line comment (started with two slashes).

Technically, the first occurrence of "</" followed by any letter is considered the end tag for the **SCRIPT** element. While browsers are forgiving in this, authors should avoid using strings such as "</P>" in their embedded scripts. JavaScript allows authors to use a backslash to avoid ending the **SCRIPT** element prematurely, e.g., `document.write("<VP>")`.

More Information

- » [SCRIPT in W3C HTML 4.0 Recommendation](#)
- » [SCRIPT in W3C HTML 3.2 Recommendation](#)
- » [SCRIPT in WDG HTML 3.2 Reference](#)
- » [SCRIPT in Learning HTML 3.2 by Examples](#)
- » [JavaScript Guide](#)
 - » [Embedding JavaScript in HTML](#)
- » [JavaScript Reference](#)
- » [JScript Web Page](#)
- » [VBScript Web Page](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

HTML 4.0 Frames



Frames

Frames allow an author to display multiple documents in a single window that is divided into rectangular subspaces called frames. Visual browsers allow these frames to be scrolled independently of each other, and links can be loaded in a frame without changing the content of other frames.

The HTML 4.0 frames model has [significant flaws](#) that make frames hated by many users. Frames should only be used with great care; see the [Guide to frames usage](#) for some guidelines on suitable use of frames.

In a [Frameset](#) document, the outermost [FRAMESET](#) element takes the place of [BODY](#) and immediately follows the [HEAD](#). Contained within the [FRAMESET](#) element are [FRAME](#) elements that define each frame, other [FRAMESET](#) elements for complex layouts, and a [NOFRAMES](#) element to provide alternate content for browsers with frames disabled or not supported.

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

P - Paragraph



P - Paragraph

Syntax
<P>...</P>
>
Attribute
Specifications

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j
u

s
t
i
f
y

J

(
h
o
r
i
z
o
n
t
a
l

a
l
l
i
g
n
m
e
n
t
)

=
c
o
m
m
o
n

=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
ADD
RES
S
APP

LET,
BLO
CKQ
UOT
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DEL,
DD,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, TD,
TH

The **P** element defines a *paragraph*. The closing tag for **P** is optional, but its use prevents common browser bugs with [style sheets](#). Note that **P** cannot contain [block-level elements](#) such as **TABLE** and **ADDRESS**.

The [deprecated](#) **ALIGN** attribute suggests the horizontal alignment for the content of the paragraph on visual browsers. Possible values are **left**, **right**, **center**, and **justify**. [Style sheets](#) provide greater flexibility in suggesting paragraph alignment.

HTML does not specify a presentation for the **P** element. Visual browsers commonly use block paragraphs with no first-line indent and separated by a blank line, but some browsers allow the user to specify a different presentation. An author can suggest paragraph indentation and spacing using style sheets. The following [CSS](#) ruleset suggests a possible presentation for paragraphs:

```
P { margin-top: 0; text-indent: 5% }
```

More Information

- [P in W3C HTML 4.0 Recommendation](#)
- [P in W3C HTML 3.2 Recommendation](#)

- » [P in HTML 2.0 Standard](#)
- » [P in WDG HTML 3.2 Reference](#)
- » [P in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

EM - Emphasis



EM - Emphasis

Syntax
...
Attribute Specifications

Common attributes

Contents
[Inline elements](#)
Contained in
[Inline elements](#),
[block-level](#)

elem ents

The **EM** element gives *emphasis* to its contents. Visual browsers typically render **EM** as italic text, but authors can suggest a rendering using [style sheets](#). Since **EM** is a structural element, it carries *meaning*, making it preferable to font style elements such as **!** when emphasis is the intended meaning.

For *strong emphasis*, use the **STRONG** element.

More Information

- [EM in W3C HTML 4.0 Recommendation](#)
- [EM in W3C HTML 3.2 Recommendation](#)
- [EM in HTML 2.0 Standard](#)
- [EM in WDG HTML 3.2 Reference](#)
- [EM in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

LI - List Item



LI - List Item

Syntax
...
>
Attribute
Specifications

"
T
Y
P
E
=
[
d
i
s
c
|
s
q
u
a
r
e
|
c
i
r
c
l
e
|
1

|
a
|
A
|
i
|
|
|
|
|
|
(
s
t
y
l
e
o
f
l
i
s
t
i
t
e
m
m
a
r
k
e
r
)
u
V
A
L
U
E
=
N
u
m
b
o
r
e
(
n
u

m
b
e
r

i
n

s
e
q
u
e
n
c
e
)

u
c
o
m
m
o
n
= a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents,
bloc
k-
level
elem
ents
(exc
ept
for
LLs
used
withi
n
DIR
and
MEN
U,
whic
h do
not

allow
block-
level
elements
)
Contained in
[OL](#),
[UL](#),
[DIR](#),
[MENU](#)
[U](#)

The **LI** element defines a *list item*. The element must be contained within [DIR](#), [MENU](#), [OL](#) or [UL](#).

Unless used with the [deprecated](#) **MENU** or **DIR** elements, **LI** may contain [block-level elements](#), including [H2](#), [TABLE](#), [UL](#), and [OL](#). This allows **OL** and **UL** to be nested, as in the following example:

```
<UL>
  <LI>
    <H2>HTML Document Type Definitions</H2>
    <UL>
      <LI><A HREF="html.dtd">HTML 2.0</A></LI>
      <LI><A HREF="HTML32.dtd">HTML 3.2</A></LI>
      <LI><A HREF="strict.dtd">HTML 4.0 Strict</A></LI>
    </UL>
  </LI>
  <LI>
    <H2>SGML Character Entity References</H2>
    <UL>
      <LI><A HREF="HTMLlat1.ent">Latin-1 Entities</A></LI>
      <LI><A HREF="HTMLsymbol.ent">Symbols and Greek Letters</A></LI>
      <LI><A HREF="HTMLspecial.ent">Other Special Characters</A></LI>
    </UL>
  </LI>
</UL>
```

When used with [MENU](#) and [DIR](#), **LI** may not contain [block-level elements](#), and lists cannot be nested.

The [deprecated](#) **TYPE** attribute of **LI** suggests the rendering of the list item marker. Possible values are as follows:

- Case-insensitive values for **LI** within a [UL](#), [DIR](#), or [MENU](#):
 - **disc** (a filled-in circle)
 - **square** (a square outline)
 - **circle** (a circle outline)
- Case-sensitive values for **LI** within an [OL](#):
 - **1** (decimal numbers: 1, 2, 3, 4, 5, ...)
 - **a** (lowercase alphabetic: a, b, c, d, e, ...)
 - **A** (uppercase alphabetic: A, B, C, D, E, ...)
 - **i** (lowercase Roman numerals: i, ii, iii, iv, v, ...)
 - **I** (uppercase Roman numerals: I, II, III, IV, V, ...)

[Style sheets](#) provide greater flexibility in suggesting list item styles. The [list-style](#) property of CSS includes the added abilities to suppress list item markers, use images as markers, and more.

LI's **VALUE** attribute specifies the number of the list item when used with an [OL](#). The number must be given as an integer, though the list item marker may be rendered in another form (for example, as a Roman numeral). Any **LI** element following in the same list will take its default sequence number based on the first preceding **VALUE** attribute. **VALUE** is [deprecated](#) in HTML 4.0, but no substitute currently exists in [CSS](#).

More Information

- [LI in W3C HTML 4.0 Recommendation](#)
- [LI in W3C HTML 3.2 Recommendation](#)
- [LI in HTML 2.0 Standard](#)
- [LI in WDG HTML 3.2 Reference](#)
- [LI in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

BR - Line Break



BR - Line Break

Syntax

Attribute
Specifications

```
<BR CLEAR = [ left | all | right | none ]>
```

(clear floating objects)

C
O
r
e
=
a
t
t
r
i
b
u
t
e
s

Contents
Empty
Container
d in
Inlin
e
elem
ents,
bloc
k-
level
elem
ents

The **BR** element forces a *break* in the current line of text. **BR** can be useful in formatting addresses within the

ADDRESS element, but it is often misused to break lines of text in a paragraph or table cell when it looks "nice" to the author. This usually results in an awkward presentation when viewed with a font size other than that used by the author.

The **CLEAR** attribute of **BR** is used to move below floating objects (typically [images](#) or [tables](#)). In the following example, the second paragraph should be rendered below the floating image:

```
<P><IMG SRC="toronto.jpg" ALIGN=left ALT="" TITLE="Toronto's CN Tower">Toronto is the
largest city in Canada and the fourth largest in North America.</P>
<BR CLEAR=left>
<P>The city is highly multicultural, with over 80 ethnic communities from Africa,
Asia, and Europe...
```

[Style sheets](#) provide more flexibility in controlling text flow around objects and eliminate the need to use **BR** for this purpose since CSS1's [clear](#) property can be applied to any element (such as the second paragraph in the preceding example).

More Information

- [BR in W3C HTML 4.0 Recommendation](#)
- [BR in W3C HTML 3.2 Recommendation](#)
- [BR in HTML 2.0 Standard](#)
- [BR in WDG HTML 3.2 Reference](#)
- [BR in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#) [HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

DOCTYPE - Document Type Declaration



DOCTYPE - Document Type Declaration

Each HTML document must begin with a *document type declaration* that declares which version of HTML the document adheres to. HTML 4.0 comes in three flavors, each with a different **DOCTYPE**:

HTML 4.0 Strict

HTML 4.0 Strict is a trimmed down version of HTML 4.0 that emphasizes structure over presentation. [Deprecated](#) elements and attributes (including most presentational attributes), [frames](#), and link targets are not allowed in HTML 4.0 Strict. By writing to HTML 4.0 Strict authors can achieve accessible, structurally rich documents that easily adapt to [style sheets](#) and different browsing situations. However, since many browsers lack full support for style sheets, HTML 4.0 Strict documents may look bland on common visual browsers such as Netscape Navigator 3.x.

The document type declaration for HTML 4.0 Strict is

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">
```

HTML 4.0 Transitional

HTML 4.0 Transitional includes all elements and attributes of [HTML 4.0 Strict](#) but adds presentational attributes, [deprecated](#) elements, and link targets. HTML 4.0 Transitional recognizes the relatively poor browser support for [style sheets](#), allowing many HTML presentation features to be used as a transition towards HTML 4.0 Strict.

The document type declaration for HTML 4.0 Transitional is

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
    "http://www.w3.org/TR/REC-html40/loose.dtd">
```

HTML 4.0 Frameset

HTML 4.0 Frameset is a variant of [HTML 4.0 Transitional](#) for documents that use [frames](#). The [FRAMESET](#) element replaces the [BODY](#) in a Frameset document.

The document type declaration for HTML 4.0 Frameset is

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
"http://www.w3.org/TR/REC-html40/frameset.dtd">
```

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

HTML - HTML Document



HTML - HTML Document

Syntax
<HTML>
..
</HTML
>
Attri
bute
Spe
cific
ation
s

u
V
E
R
S
I
O
N
=
C
D
A
T
A
(
H
T
M
L
v
e
r
s
i
o
n
)
u
i
n
t
e

r
n
a
t
i
o
n
a
l
i
z
a
t
i
o
n
= a
t
t
r
i
b
u
t
e
s

Cont
ents

u
I
n
H
T
M
L
4
.
0
S
t
r
i
c
t
a
n
d
I
r
a
n
s
i

t
i
o
n
a
l
:

H
E
A
D

f
o
l
l
o
w
e
d

b
y

B
O
D
Y

u
I
n

H
T
M
L

4
.0

F
r
a
m
e
s
e
t
:

H
E
A
D

f
o
l
l

o
w
e
d

b
y

F
R
A
M
E
S
E
T

Cont
aine
d in
Not
appli
cabl
e--
HTM
L is
a
top-
level
elem
ent

The **HTML** element contains the *HTML document*, made up of the **HEAD** followed by the **BODY**, except in Frameset documents where the FRAMESET element replaces the **BODY**. The start and end tags of the **HTML** element are both optional.

The **LANG** attribute is typically set on the **HTML** element to specify the base language of the document. The language should be specified according to RFC 1766; examples include **en** for English, **en-US** for American English, and **ja** for Japanese.

The deprecated **VERSION** attribute specifies the Document Type Definition (DTD) that describes the document. This attribute should not be used since the DOCTYPE declaration makes it redundant.

More Information

- [HTML in W3C HTML 4.0 Recommendation](#)
- [HTML in W3C HTML 3.2 Recommendation](#)
- [HTML in HTML 2.0 Standard](#)
- [HTML in WDG HTML 3.2 Reference](#)
- [HTML in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

HEAD - Document Head



HEAD - Document Head

Syntax
<HEAD>
..
</HEAD
>
Attri
bute
Spe
cific
ation
s

u
P
R
O
F
I
L
E
=
U
R
I
(
d
i
c
t
i
o
n
a
r
y
o
f
m
e
t
a
i
n

f
o
)

u
i
n
t
e
r
n
a
t
i
o
n
a
l
i
z
a
t
i
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Exa
ctly
one
TITL
E
elem
ent,
optio
nal
BAS
E
and
ISIN
DEX
elem
ents,
and
zero
or
mor
e
SCR
IPT,

[STY](#)
[LE](#),
[MET](#)
[A](#),
[LIN](#)
[K](#), or
[OBJ](#)
[ECT](#)
elem
ents
Cont
aine
d in
[HTM](#)
[L](#)

The **HEAD** element contains *header information* about the document, such as its title, keywords, description, and style sheet. **HEAD** is required in all documents, but its start and end tags are always optional. The **HEAD** element is followed by the **BODY** in HTML 4.0 [Strict](#) and [Transitional](#) documents; in HTML 4.0 [Frameset](#) documents, the **HEAD** is followed by a [FRAMESET](#) element.

Content in the **HEAD** is generally not rendered, with the exception of the required [TITLE](#) element. If the `</HEAD>` end tag is omitted, the first [BODY](#) or [FRAMESET](#) element infers the end of the **HEAD**.

The optional **PROFILE** attribute of **HEAD** gives the location of a metadata profile. A profile defines properties that may be used by [META](#) and [LINK](#) elements within the **HEAD**. There is no prescribed format for profiles.

Work is currently underway on improving the use of metadata on the Web. See the [W3C's Metadata and Resource Description](#) area for the latest information.

More Information

- [HEAD in W3C HTML 4.0 Recommendation](#)
- [HEAD in W3C HTML 3.2 Recommendation](#)
- [HEAD in HTML 2.0 Standard](#)
- [HEAD in WDG HTML 3.2 Reference](#)
- [HEAD in Learning HTML 3.2 by Examples](#)
- [Metadata at W3C](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

BODY - Document Body



BODY - Document Body

Syntax
<BODY>
..
</BODY
>
Attri
bute
Spe
cific
ation
s

u

B
A
C
K
G
R
O
U
N
D
=
U
R
I

(
b
a
c
k
g
r
o
u
n
d

i
m
a
g
e

f

o
r
d
o
c
u
m
e
n
t
)

“ B G C O L O R = C
o
o
o
r ”

(b a c k g r o u n d

c o l o r

f o r

d o c u m e n t)

“ T

E
X
T
=
C
o
l
o
r

(
t
e
x
t

c
o
l
o
r

f
o
r

d
o
c
u
m
e
n
t
)

“
L
I
N
K
=
C
o
l
o
r

(
l
i
n
k

c
o
l
o
r

f
o

r
d
o
c
u
m
e
n
t
)

»
V
L
I
N
K
=
[C
o
l
o
r](#)

(
v
i
s
i
t
e
d

l
i
n
k

c
o
l
o
r

f
o
r

d
o
c
u
m
e
n
t
)

»
A
L

I
N
K

=

C
o
l
o
r

(
a
c
t
i
v
e

l
i
n
k

c
o
l
o
r

f
o
r

d
o
c
u
m
e
n
t
)

"

O
N
L
O
A
D

=

S
o
f
t
w
a
r
e

(
d
o
c

u
m
e
n
t

h
a
s

b
e
e
n

l
o
a
d
e
d
)

u

O
N
L
O
A
D

=
S
C
I
P
T

(
d
o
c
u
m
e
n
t

h
a
s

b
e
e
n

e
x
i

t
e
d
)
u
c
o
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents

u
l
n
H
I
M
L
=
4
=
0
=
S
t
r
i
c
t
:
o
n
e
o
r
m
o
r
e
b

L O C K l i v e t e m e n t s o r S R I P T I S I N D E L I N H I M L 4 Q I R a n s i t i n O

a
|
:

i
|
n
|
i
|
n
|
e
|
e
|
e
|
m
|
e
|
n
|
t
|
s

,
b
|
l
|
o
|
c
|
k
|
=
|
e
|
v
|
e
|
e
|
e
|
m
|
e
|
n
|
t
|
s

,
I
|
N
|
S

,
D
|
E
|
L

Cont
aine
d in

u
|
n

H
T

M
L

4
.0

S
t
r
i
c
t

o
r

I
r
a
n
s
i
t
i
o
n
a
!
:

H
T
M
L

u
l
n

H
T
M
L

4
.0

F
r
a
m
e
s
e
t
:

F
O
N

R A M E S

The **BODY** element contains the *document body*. **BODY** is required in non-frames documents, but its start and end tags are always optional. In [frames documents](#), **BODY** must be contained within the **NOFRAMES** element, if **NOFRAMES** is used.

The **BODY** element contains the document's content. The content should be contained within [block-level elements](#) or [SCRIPT](#) elements, though [HTML 4.0 Transitional](#) also allows [inline elements](#) directly within **BODY**. **BODY** takes a number of attributes for specifying the background and colors of the document on visual browsers. These attributes are [deprecated](#) in HTML 4.0 in favor of [style sheets](#), which provide greater flexibility in suggesting the presentation of a document. **BGCOLOR** suggests a background color, **TEXT** suggests a text color, **LINK** suggests a link color, **VLINK** suggests a visited link color, and **ACTIVE** suggests an active link color (when the link is selected). If one of these attributes is given, then all of them should be included to ensure that the user's chosen colors do not interfere with those suggested in the **<BODY>** tag. Authors should not rely on the specified colors being used since browsers allow these colors to be overridden by the user.

The **BACKGROUND** attribute suggests a background image for tiling on the document canvas. To help ensure a readable document, the **BGCOLOR**, **TEXT**, **LINK**, **VLINK**, and **ALINK** attributes should always be included when **BACKGROUND** is given. The **BGCOLOR** will be used for those not loading images.

[Style sheets](#) allow more flexibility in suggesting a background image, including the ability to specify the position of the image, how the image is tiled, and whether the image should scroll with the document.

In addition to the [core events](#) common to most elements, **BODY** accepts the following event attributes for client-side scripting:

- **ONLOAD**, when the document has been loaded;
- **ONUNLOAD**, when the document is exited.

More Information

- [BODY in W3C HTML 4.0 Recommendation](#)
- [BODY in W3C HTML 3.2 Recommendation](#)
- [BODY in HTML 2.0 Standard](#)
- [BODY in WDG HTML 3.2 Reference](#)
- [BODY in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

HTML 4.0 Block-level Elements



HTML 4.0 Block-Level Elements

Most HTML 4.0 elements permitted within the **BODY** are classified as either *block-level elements* or *inline elements*. Block-level elements typically contain inline elements and other block-level elements. When rendered visually, block-level elements usually begin on a new line.

The following are defined as block-level elements in HTML 4.0:

- ⌋ [ADDRESS](#) - Address
- ⌋ [BLOCKQUOTE](#) - Block quotation
- ⌋ [CENTER](#) - Centered block
- ⌋ [DIR](#) - Directory list
- ⌋ [DIV](#) - Generic block-level container
- ⌋ [DL](#) - Definition list
- ⌋ [FIELDSET](#) - Form control group
- ⌋ [FORM](#) - Interactive form
- ⌋ [H1](#) - Level-one heading
- ⌋ [H2](#) - Level-two heading
- ⌋ [H3](#) - Level-three heading
- ⌋ [H4](#) - Level-four heading
- ⌋ [H5](#) - Level-five heading
- ⌋ [H6](#) - Level-six heading
- ⌋ [HR](#) - Horizontal rule
- ⌋ [ISINDEX](#) - Input prompt
- ⌋ [MENU](#) - Menu list
- ⌋ [NOFRAMES](#) - Frames alternate content
- ⌋ [NOSCRIPT](#) - Alternate script content
- ⌋ [OL](#) - Ordered list
- ⌋ [P](#) - Paragraph
- ⌋ [PRE](#) - Preformatted text
- ⌋ [TABLE](#) - Table
- ⌋ [UL](#) - Unordered list

The following elements may also be considered block-level elements since they may contain block-level elements:

- ⌋ [DD](#) - Definition description
- ⌋ [DT](#) - Definition term
- ⌋ [FRAMESET](#) - Frameset
- ⌋ [LI](#) - List item
- ⌋ [TBODY](#) - Table body
- ⌋ [TD](#) - Table data cell
- ⌋ [TFOOT](#) - Table foot
- ⌋ [TH](#) - Table header cell
- ⌋ [THEAD](#) - Table head
- ⌋ [TR](#) - Table row

The following elements may be used as either block-level elements or [inline elements](#). If used as inline elements (e.g., within another inline element or a [P](#)), these elements should not contain any block-level elements.

- [APPLET](#) - Java applet
- [BUTTON](#) - Button
- [DEL](#) - Deleted text
- [IFRAME](#) - Inline frame
- [INS](#) - Inserted text
- [MAP](#) - Image map
- [OBJECT](#) - Object
- [SCRIPT](#) - Client-side script

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

HTML 4.0 Inline Elements



HTML 4.0 Inline Elements

Most HTML 4.0 elements permitted within the **BODY** are classified as either [block-level elements](#) or *inline elements*. Inline elements typically may only contain text and other inline elements. When rendered visually, inline elements do not usually begin on a new line.

The following are defined as inline elements in HTML 4.0:

- [A](#) - Anchor
- [ABBR](#) - Abbreviation
- [ACRONYM](#) - Acronym
- [B](#) - Bold text
- [BASEFONT](#) - Base font change
- [BDO](#) - BiDi override
- [BIG](#) - Large text
- [BR](#) - Line break
- [CITE](#) - Citation
- [CODE](#) - Computer code
- [DFN](#) - Defined term
- [EM](#) - Emphasis
- [FONT](#) - Font change
- [I](#) - Italic text
- [IMG](#) - Inline image
- [INPUT](#) - Form input
- [KBD](#) - Text to be input
- [LABEL](#) - Form field label
- [Q](#) - Short quotation
- [S](#) - Strike-through text
- [SAMP](#) - Sample output
- [SELECT](#) - Option selector
- [SMALL](#) - Small text
- [SPAN](#) - Generic inline container
- [STRIKE](#) - Strike-through text
- [STRONG](#) - Strong emphasis
- [SUB](#) - Subscript
- [SUP](#) - Superscript
- [TEXTAREA](#) - Multi-line text input
- [TT](#) - Teletype text
- [U](#) - Underlined text
- [VAR](#) - Variable

The following elements may be used as either [block-level elements](#) or inline elements. If used as inline elements (e.g., within another inline element or a [P](#)), these elements should not contain any block-level elements.

- [APPLET](#) - Java Applet

- [BUTTON](#) - Button
- [DEL](#) - Deleted text
- [IFRAME](#) - Inline frame
- [INS](#) - Inserted text
- [MAP](#) - Image map
- [OBJECT](#) - Object
- [SCRIPT](#) - Client-side script

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

BASE - Document Base URI



BASE - Document Base URI

Syntax
<BASE>
Attribute
Specifications

u

H

R

E

F

=

U

R

I

(

b

a

s

e

)

U

R

I

r

e

f

e

r

e

n

c

e

)

u

T

A

R

G

E

T
=
FF
IR
IA
IB
IE
IF
IA
IR
IG
IE
IT

(
f
r
a
m
e

t
o

r
e
n
d
e
r

l
i
n
k
s

i
n
)

Cont
ents
Emp
ty
Cont
aine
d in
HEA
D

The **BASE** element defines the document's *base URI* for resolving relative URIs contained within the document. A document cannot contain more than one **BASE** element. When present, the **BASE** element must appear in the HEAD, prior to any elements that include a partial URI.

BASE's **HREF** attribute, required in HTML 4.0 Strict, specifies the absolute URI used to resolve relative URIs. See Using Relative URLs for more details on using and resolving relative URIs.

Most Web pages do not require an explicit base URI since the document's URI is a suitable base. An explicit base URI is only required when the same document may be accessed at different URIs or when the document has no URI (e.g., sending an HTML document by e-mail).

The **TARGET** attribute is used with [frames](#) to specify in which frame all links in document should be rendered by default. The target frame specified by **BASE** can be overridden by a given link using the link's **TARGET** attribute.

If no frame with the specified target name exists, the links are rendered in a new window unless overridden by the user. Special frame names begin with an underscore:

- **_blank** renders the links in a new, unnamed window
- **_self** renders the links in the current frame
- **_parent** renders the links in the immediate **FRAMESET** parent
- **_top** renders the links in the full, unframed window

More Information

- [BASE in W3C HTML 4.0 Recommendation](#)
- [BASE in W3C HTML 3.2 Recommendation](#)
- [BASE in HTML 2.0 Standard](#)
- [BASE in WDG HTML 3.2 Reference](#)
- [BASE in Learning HTML 3.2 by Examples](#)
- [RFC 1808: Relative Uniform Resource Locators](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

ISINDEX - Input Prompt



ISINDEX - Input Prompt

Syntax
<ISINDEX
X>
Attribute
Specifications

"
P
R
O
M
P
T
=
T
I
X
T

(
P
r
o
m
p
t
m
e
s
s
a
g
e
)
"
C
O
R
E
"

a
t
t
r
i
b
u
t
e
s

i
n
t
e
r
n
a
t
i
o
n
a
i
z
a
t
i
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Emp
ty
Cont
aine
d in
APP
LET,
BLO
CKQ
UOT
E,
BOD
Y,
CEN
TER

,

[DD](#),
[DEL](#),
[DIV](#),
[FIEL](#)
[DSE](#)
[T](#),
[FOR](#)
[M](#),
[HEA](#)
[D](#),
[IFR](#)
[AME](#)
,
[INS](#),
[LI](#),
[MAP](#)
,
[NOF](#)
[RAM](#)
[ES](#),
[NOS](#)
[CRI](#)
[PT](#),
[OBJ](#)
[ECT](#)
, [TD](#),
[TH](#)

The **ISINDEX** element defines a *single-line text input*. The label of the input field is specified using the element's **PROMPT** attribute. **ISINDEX** is [deprecated](#) in HTML 4.0 in favor of the [INPUT](#) element.

The **ISINDEX** element is equivalent to a **FORM** with a single **INPUT** of type **text**, a **METHOD** of **get**, and an **ACTION** pointing to the URI of the document containing the **ISINDEX** element.

More Information

- [ISINDEX in W3C HTML 4.0 Recommendation](#)
- [ISINDEX in W3C HTML 3.2 Recommendation](#)
- [ISINDEX in HTML 2.0 Standard](#)
- [ISINDEX in WDG HTML 3.2 Reference](#)
- [ISINDEX in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

META - Metadata



META - Metadata

Syntax
<META>
Attribute
Specifications

NAME = URI (property name) HTTP-EQUIV

V
=
N
B
E

(
H
T
T
P

r
e
s
p
o
n
s
e

h
e
a
d
e
r

n
a
m
e
)

"
C
O
N
T
E
N
T
=
C
D
A
T
A

(
a
s
s
o
c
i
a
t
e
d

**d
a
t
a
)**

“

**S
C
H
E
M
E
=**

**C
D
A
T
A**

**(
f
o
r
m**

**o
f**

**d
a
t
a
)**

“

**i
n
t
e
r
n
a
t
i
o
n
a
l
i
z
a
t
i
o
n**

**=
a
t
t
r
i**

b
u
t
e
s

(
f
o
r

t
h
e

C
O
N
T
E
N
T
)

Cont
ents
Emp
ty
Cont
aine
d in
HEA
D

The **META** element provides *metadata* such as a document's keywords, description, and author. Any number of **META** elements may be contained in the HEAD of a document.

META's NAME attribute provides a property name while the **CONTENT** attribute gives the corresponding value. The **CONTENT** attribute value may contain text and entities, but it may not contain HTML tags.

The optional **SCHEME** attribute gives the format of the property value. For example, a date property may require **SCHEME="Month-Day-Year"** to disambiguate the date from other formats such as **SCHEME="Day-Month-Year"**.

There is no standard list of **META** properties, so authors may define whatever metadata they like. The following example defines the author of the document:

```
<META NAME=author CONTENT="Liam Quinn">
```

Some search engines use **keywords** and **description** properties, giving extra weight to a document's keywords and providing its description with the link to the document. Example:

```
<META NAME="description" CONTENT="A description of HTML 4.0's META element for  
metadata.">  
<META NAME="keywords" CONTENT="META, meta element, metadata, metainformation, meta  
data, meta information, keywords, description, refresh, HyperText Markup Language,
```

HTML, HTML4, HTML 4.0, Web Design Group, WDG, <meta> tag, <META> tag">

To avoid being truncated by search engines, the description should be brief--no more than 200 characters. Keywords are separated by commas and may be considered case sensitive by search engines. If the same keywords are repeated too often in the **META** element, some search engines will not index the document. Search engines typically only process the first 1000 characters of the keywords list.

Some search engines also support the **robots** property for indicating whether a document should be indexed and whether its links should be followed. The associated **CONTENT** value is a comma-separated list of case-insensitive directives:

- „**index** specifies that the page should be indexed while **noindex** specifies that it should not be indexed;
- „**follow** specifies that the page's links should be followed while **nofollow** specifies that they should not be followed;
- „**all** is equivalent to **index,follow** (the default value);
- „**none** is equivalent to **noindex,nofollow**.

For example, the following **META** element tells search engines and other robots not to index the page but to follow links on it:

```
<META NAME=robots CONTENT="noindex, follow">
```

Few search engines support the **robots** property at this time. For greater compliance by robots, authors should use the [Robots Exclusion Protocol](#) if possible.

The **HTTP-EQUIV** attribute may be used in place of the **NAME** attribute to indicate that the property is an HTTP header. Some servers will send the HTTP header specified in the **META** element, and browsers often recognize the header even when it is not sent by the server. Examples:

```
<META HTTP-EQUIV=Expires CONTENT="Sun, 22 Mar 1998 16:18:35 GMT">
```

sets the expiry date of the document.

```
<META HTTP-EQUIV="Content-Script-Type" CONTENT="text/javascript">
```

sets the client-side scripting language for inline scripts to JavaScript.

```
<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">
```

sets the style language for inline styles to CSS.

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=SHIFT_JIS">
```

sets the character encoding for the document to SHIFT_JIS (a Japanese encoding). Note that using **META** for this purpose rather than a true HTTP header causes some browsers to redraw the page after initially displaying it.

```
<META HTTP-EQUIV=Refresh CONTENT="10; URL=http://www.htmlhelp.com/">
```

tells the browser to load <http://www.htmlhelp.com/> 10 seconds after the current document has finished loading.

Not all browsers support this, so authors should provide an alternate means of moving to the new page where necessary. The **Refresh** header is sometimes used for "splash screens" or when a page has moved, but the technique is not very effective since users may not even be looking at the window that is to be refreshed and since it messes up the user's history on many browsers. Some search engines penalize pages that use a **Refresh** of a few seconds or less.

More Information

„[META in W3C HTML 4.0 Recommendation](#)

„[META in W3C HTML 3.2 Recommendation](#)

„[META in HTML 2.0 Standard](#)

- » [META in WDG HTML 3.2 Reference](#)
- » [META in *Learning HTML 3.2 by Examples*](#)
- » [Metadata at W3C](#)
- » [The META tag: Controlling how your Web page is indexed by AltaVista](#)
- » [Infoseek Submitting Tips](#)
- » [HotBot FAQ: Descriptions, Titles, Keywords, and <meta> Tags](#)
- » [Lycos Help: Robots Meta Tag](#)
- » [Robots Exclusion](#)
 - » [HTML Author's Guide to the Robots META tag](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

TITLE - Document Title



TITLE - Document Title

Syntax
<TITLE>
..</TITLE
>
Attri
bute
Spe
cific
ation
s

i
n
t
e
r
n
a
t
i
o
n
a
!
i
Z
a
t
i
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont

ents
Plain
text
(including
entities)
Contained
in
HEAD

The **TITLE** element gives the document's *title*. Each document must have exactly one **TITLE** within the HEAD. **TITLE** contains plain text and entities; it may not contain other markup.

A good **TITLE** should be short and specific to the document's content so that it can be used as a title for a user's bookmark, a title for the display window on visual browsers, and a link from a search engine. A suggested limit for the number of characters in a **TITLE** is 60.

More Information

- [TITLE in W3C HTML 4.0 Recommendation](#)
- [TITLE in W3C HTML 3.2 Recommendation](#)
- [TITLE in HTML 2.0 Standard](#)
- [TITLE in WDG HTML 3.2 Reference](#)
- [TITLE in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

ADDRESS - Contact Information



ADDRESS - Contact Information

Syntax
<ADDRESS>...</ADDRESS>
>
Attribute Specifications

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents

l
n
H
I
M
L
=
4
.

O S r i c t t i n i n e l e m e n t s i n H I M L 4 O T r a n s i t i o n a i n i n e l e m e n t s

e
m
e
n
t
s
,

P

Cont
aine
d in
APP
LET
BLO
CKQ
UOT
E
BOD
Y
BUT
TON
,

CEN
TER

,

DD
DEL
DIV
FIEL
DSE
I
FOR
M
IFR
AME

,

INS
LI
MAP

,

NOF
RAM
ES
NQS
CRI
PT
OBJ
ECT
,

ID
TH

The **ADDRESS** element provides *contact information* for a document or part of a document. Information provided by **ADDRESS** may include the names of the document's maintainers, links to the maintainers' Web pages, e-mail addresses for feedback, postal addresses, phone numbers, and so on. The **ADDRESS** element is not appropriate for all postal and e-mail addresses; it should be reserved for providing such information about the contact people for the document.

The following example, most appropriate at the end of a document, gives contact information about the maintainer:

```
<ADDRESS>Maintained by <A href="http://www.htmlhelp.com/%7Eliam/">Liam Quinn</A>
<lt;<A HREF="mailto:liam@htmlhelp.com">liam@htmlhelp.com</A>&gt;</ADDRESS>
```

ADDRESS can also be used to provide contact information for a portion of a document, typically a [form](#). The next example gives users contact information to use in conjunction with an order form:

```
<FORM METHOD=post ACTION="/cgi-bin/order.cgi">

<FIELDSET>
  <LEGEND ACCESSKEY=C>Credit Card Information<BR></LEGEND>
  <P>
    <LABEL ACCESSKEY=V>
      <INPUT TYPE=radio NAME=card VALUE=visa> Visa
    </LABEL>
    <LABEL ACCESSKEY=M>
      <INPUT TYPE=radio NAME=card VALUE=mc> MasterCard
    </LABEL>
    <BR>
    <LABEL ACCESSKEY=N>
      Number: <INPUT TYPE=text NAME=number>
    </LABEL>
    <LABEL ACCESSKEY=E>
      Expiry: <INPUT TYPE=text NAME=expiry>
    </LABEL>
  </P>
</FIELDSET>

<P>
  <INPUT TYPE=submit VALUE="Submit order" ACCESSKEY=S>
</P>

<ADDRESS>
  If you have any questions about ordering, contact us at
  <A HREF="mailto:orders@htmlhelp.com">orders@htmlhelp.com</A>,
  or phone our offices at 555-5555.
</ADDRESS>

</FORM>
```

More Information

- [ADDRESS in W3C HTML 4.0 Recommendation](#)
- [ADDRESS in W3C HTML 3.2 Recommendation](#)
- [ADDRESS in HTML 2.0 Standard](#)
- [ADDRESS in WDG HTML 3.2 Reference](#)
- [ADDRESS in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

BLOCKQUOTE - Block Quotation



BLOCKQUOTE - Block Quotation

Syntax
<BLOCK
QUOTE>
..
</BLOC
KQUOTE
>
Attri
bute
Spe
cific
ation
s

u
C
I
T
E
=
C
I
T
I
N
G
(
s
o
u
r
c
e
o
f
q
u
o
t
a
t
i
o
n
)
u

c
o
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents

^u
I
n

H
I
M
L

=
4

^o
Q

=
S
t
r
i
c
t
:

o
n
e

o
r

m
o
r
e

b
l
o
c
k
=
!

level elements for
or
S
P
I
n
H
L
4
O
T
r
a
n
s
i
t
i
o
n
a
l
i
n
l
i
n
e
le
ve
l
e
m
e
n
t
s
o
r
S
P
I
n
H
L
4
O
T
r
a
n
s
i
t
i
o
n
a
l
i
n
l
i
n
e
le

n
t
s

o
r

b
l
o
c
k
q
u
e
v
e
i
e
i
e
m
e
n
t
s

Cont
aine
d in
APP
LET
BLO
CKQ
UOT
E
BOD
Y
BUT
TON
,
CEN
TER
,
DD
DEL
DIV
FIEL
DSE
I
FOR
M
IER
AME
,
INS
LI
MAP
,
NOF
RAM
ES,

[NOS](#)
[CRI](#)
[PT,](#)
[OBJ](#)
[ECT](#)
[, TD,](#)
[TH](#)

The **BLOCKQUOTE** element defines a *block quotation*. Unlike [inline quotations](#), block quotations may contain [block-level elements](#) such as [P](#) and [TABLE](#), but **BLOCKQUOTE** may not be contained within a paragraph or [inline element](#).

The optional **CITE** attribute of **BLOCKQUOTE** provides the URI of the source of the quotation. This attribute, not widely supported among browsers, allows readers to verify the authenticity of the quotation and also find related information.

The content of the **BLOCKQUOTE** element should be contained within other block-level elements, typically [P](#). The following example features a quotation that includes multiple paragraphs and its own block quotation:

```
<BLOCKQUOTE CITE="http://www.bibliomania.com/Fiction/joyce/ulysses/telemac.html">
<P>He pointed his finger in friendly jest and went over to the parapet, laughing to himself. Stephen Dedalus stepped up, followed him wearily half way and sat down on the edge of the gunrest, watching him still as he propped his mirror on the parapet, dipped the brush in the bowl and lathered cheeks and neck.</P>
<P>Buck Mulligan's gay voice went on.</P>
<BLOCKQUOTE><P>My name is absurd too: Malachi Mulligan, two dactyls. But it has a Hellenic ring, hasn't it? Tripping and sunny like the buck himself. We must go to Athens. Will you come if I can get the aunt to fork out twenty quid?</P></BLOCKQUOTE>
</BLOCKQUOTE>
```

Authors should not use **BLOCKQUOTE** for unquoted material just to achieve a block indentation in common visual browsers. With the rise of style sheets, such misuse of **BLOCKQUOTE** will become less reliable while also reducing the author's ability to fully exploit the power of style sheets. [Cascading Style Sheets](#) provide the [margin-left](#) property to indent a block.

More Information

- » [BLOCKQUOTE in W3C HTML 4.0 Recommendation](#)
- » [BLOCKQUOTE in W3C HTML 3.2 Recommendation](#)
- » [BLOCKQUOTE in HTML 2.0 Standard](#)
- » [BLOCKQUOTE in WDG HTML 3.2 Reference](#)
- » [BLOCKQUOTE in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

CENTER - Centered Block



CENTER - Centered Block

Syntax
<CENTE
R>...</CE
NTER>
Attri
bute
Spe
cific
ation
s

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
bloc
k-
level
elem
ents
Cont
aine
d in
APP

LET,
BLO
CKQ
UOT
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DD,
DEL,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, TD,
TH

The **CENTER** element defines a block whose contents are *centered* horizontally on visual browsers. **<CENTER>** is a shorthand for **<DIV ALIGN=center>**, though **CENTER** is slightly better supported among browsers. Both methods of centering are [deprecated](#) in favor of [style sheets](#).

CENTER is still useful for centering [tables](#) since many browsers lack support for **<TABLE ALIGN=center>** as well as the method of centering tables with Cascading Style Sheets (setting [margin-left](#) and [margin-right](#) to **auto**). An example follows:

```

<CENTER>
<TABLE>
  <TR ALIGN=center>
    <TH SCOPE=col>Name</TH>
    <TH SCOPE=col>Age</TH>
    <TH SCOPE=col>Country</TH>
  </TR>
  <TR ALIGN=center>
    <TD>Liam Quinn</TD>
    <TD>20</TD>
    <TD>Canada</TD>
  </TR>
</TABLE>
</CENTER>

```

Note that **CENTER** only centers the table as a whole, not the contents of each table cell. The preceding example uses the **ALIGN** attribute of **TR** to center the contents of each cell. The [text-align](#) property of [Cascading Style Sheets](#) provides greater flexibility in suggesting horizontal alignment.

More Information

- [CENTER in W3C HTML 4.0 Recommendation](#)
- [CENTER in W3C HTML 3.2 Recommendation](#)
- [CENTER in WDG HTML 3.2 Reference](#)
- [CENTER in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

DIV - Generic Block-level Container



DIV - Generic Block-level Container

Syntax
<DIV>...</DIV>
Attribute Specifications

align="left" style="text-align: center; text-align: right; text-align: justify;"

s
t
i
f
y

J

(
h
o
r
i
z
o
n
t
a
l
a
l
l
i
g
n
m
e
n
t
)

u

c
o
m
m
o
n

=

a
t
t
r
i
b
u
t
e
s

Cont
ents

Inlin

e

elem

ents,

bloc

k-

level

elem

ents

Cont

aine

d in
APP
LET,
BLO
CKQ
UOT
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DD,
DEL,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, TD,
TH

The **DIV** element defines a *generic block-level container*, allowing authors to provide style or language information to blocks of content. The element may contain any [inline](#) or [block-level](#) element, including another **DIV**.

The **DIV** element is most useful in combination with the **CLASS**, **ID**, or **LANG** attributes. For example, a navigation bar could be contained within a **DIV** marked as **CLASS=navbar**, allowing the author to use [style sheets](#) to easily change the background of all navigation bars on a site, or to eliminate navigation bars when printing. The [deprecated](#) **ALIGN** attribute suggests the horizontal alignment for the content of the division on visual browsers. Possible values are **left**, **right**, **center**, and **justify**. **<CENTER>** is a slightly better-supported alias for **<DIV ALIGN=center>**, though both methods of centering are deprecated in favor of [style sheets](#), which provide greater flexibility in suggesting alignment.

SPAN is a text-level equivalent of **DIV** for use within [paragraphs](#) and [inline elements](#).

More Information

- » [DIV in W3C HTML 4.0 Recommendation](#)
- » [DIV in W3C HTML 3.2 Recommendation](#)
- » [DIV in WDG HTML 3.2 Reference](#)
- » [DIV in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

H1 - Level-one Heading



H1 - Level-one Heading

Syntax
<H1>...</
H1>
Attribute
Specifications

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j
u

s
t
i
f
y

J

(
h
o
r
i
z
o
n
t
a
l
a
l
l
i
g
n
m
e
n
t
)

u

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents

Inlin
e
elem
ents

Cont
aine
d in

APP
LET,
BLO
CKQ

UO
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DEL,
DD,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,
TH

The **H1** element defines a *level-one heading*. A document generally should have exactly one **H1** element to mark the most important heading.

The [deprecated ALIGN](#) attribute suggests the horizontal alignment for the heading on visual browsers. Possible values are **left**, **right**, **center**, and **justify**. [Style sheets](#) provide greater flexibility in suggesting alignment.

Visual browsers typically render **H1** in a large, bold font. Authors can suggest a presentation for **H1** through style sheets. The following [CSS](#) ruleset suggests a presentation for all **H1** elements:

```
h1 {  
  color: #c33;  
  background: transparent;  
  font-weight: bold;  
  text-align: center  
}
```

Authors should not choose a heading level based on the font size commonly used by visual browsers. The heading level should be chosen based on the heading's importance and placement in the document. An **H1** is typically used as the main heading for a document.

More Information

- » [H1 in W3C HTML 4.0 Recommendation](#)
- » [H1 in W3C HTML 3.2 Recommendation](#)
- » [H1 in HTML 2.0 Standard](#)
- » [H1 in WDG HTML 3.2 Reference](#)
- » [H1 in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

H2 - Level-two Heading



H2 - Level-two Heading

Syntax
<H2>...</
H2>
Attribute
Specifications

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j
u

s
t
i
f
y

J

(
h
o
r
i
z
o
n
t
a
l
a
l
l
i
g
n
m
e
n
t
)

u

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
APP
LET,
BLO
CKQ

UO
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DEL,
DD,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,
TH

The **H2** element defines a *level-two heading*. This heading is more important than an **H3** but less important than an **H1**.

The [deprecated ALIGN](#) attribute suggests the horizontal alignment for the heading on visual browsers. Possible values are **left**, **right**, **center**, and **justify**. [Style sheets](#) provide greater flexibility in suggesting alignment.

Visual browsers typically render **H2** in a large, bold font. Authors can suggest a presentation for **H2** through style sheets. The following [CSS](#) ruleset suggests a presentation for all **H2** elements:

```
h2 {  
  color: #00008b;  
  background: transparent;  
  font-weight: bold;  
  margin-left: 2%;  
  margin-right: 2%;  
}
```

Authors should not choose a heading level based on the font size commonly used by visual browsers. The heading level should be chosen based on the heading's importance and placement in the document. An **H2** is typically used within a section headed by an **H1**.

More Information

- [H2 in W3C HTML 4.0 Recommendation](#)
- [H2 in W3C HTML 3.2 Recommendation](#)
- [H2 in HTML 2.0 Standard](#)
- [H2 in WDG HTML 3.2 Reference](#)
- [H2 in *Learning HTML 3.2 by Examples*](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

H3 - Level-three Heading



H3 - Level-three Heading

Syntax
<H3>...</
H3>
Attribute
Specifications

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j
u

s
t
i
f
y

J

(
h
o
r
i
z
o
n
t
a
l
a
l
l
i
g
n
m
e
n
t
)

u

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents

Inlin
e
elem
ents

Cont
aine
d in

APP
LET,
BLO
CKQ

UO
T
E
BOD
Y
,
BUT
TON
,
CEN
TER
,
DEL
DD
,
DIV
FIEL
DSE
I
,
FOR
M
IFR
AME
,
INS
LI
MAP
,
NOF
RAM
ES
NOS
CRI
PT
OBJ
ECT
,
ID
TH

The **H3** element defines a *level-three heading*. This heading is more important than an **H4** but less important than an **H2**.

The [deprecated ALIGN](#) attribute suggests the horizontal alignment for the heading on visual browsers. Possible values are **left**, **right**, **center**, and **justify**. [Style sheets](#) provide greater flexibility in suggesting alignment.

Visual browsers typically render **H3** in a bold font. Authors can suggest a presentation for **H3** through style sheets. The following [CSS](#) ruleset suggests a presentation for all **H3** elements:

```
h3 {  
  color: #006400;  
  background: transparent;  
  margin-left: 4%;  
  margin-right: 4%;  
  font-weight: bold  
}
```

Authors should not choose a heading level based on the font size commonly used by visual browsers. The heading level should be chosen based on the heading's importance and placement in the document. An **H3** is typically used within a section headed by an **H2**.

More Information

- [H3 in W3C HTML 4.0 Recommendation](#)
- [H3 in W3C HTML 3.2 Recommendation](#)
- [H3 in HTML 2.0 Standard](#)
- [H3 in WDG HTML 3.2 Reference](#)
- [H3 in *Learning HTML 3.2 by Examples*](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

H4 - Level-four Heading



H4 - Level-four Heading

Syntax
<H4>...</
H4>
Attribute
Specifications

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j
u

s
t
i
f
y

J

(
h
o
r
i
z
o
n
t
a
l
a
l
l
i
g
n
m
e
n
t
)

u

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
APP
LET
BLO
CKQ

UOT
E
BOD
Y
BUT
TON
,
CEN
TER
,
DEL
DD
DIV
FIEL
DSE
I
FOR
M
IFR
AME
,
INS
LI
MAP
,
NOF
RAM
ES
NOS
CRI
PT
OBJ
ECT
, ID
TH

The **H4** element defines a *level-four heading*. This heading is more important than an **H5** but less important than an **H3**.

The [deprecated ALIGN](#) attribute suggests the horizontal alignment for the heading on visual browsers. Possible values are **left**, **right**, **center**, and **justify**. [Style sheets](#) provide greater flexibility in suggesting alignment.

Visual browsers typically render **H4** in a bold font. Authors can suggest a presentation for **H4** through style sheets. The following [CSS](#) ruleset suggests a presentation for all **H4** elements:

```
h4 {  
  margin-left: 6%;  
  margin-right: 6%;  
  font-weight: bold  
}
```

Authors should not choose a heading level based on the font size commonly used by visual browsers. The heading level should be chosen based on the heading's importance and placement in the document. An **H4** is typically used within a section headed by an **H3**.

More Information

- » [H4 in W3C HTML 4.0 Recommendation](#)
- » [H4 in W3C HTML 3.2 Recommendation](#)
- » [H4 in HTML 2.0 Standard](#)
- » [H4 in WDG HTML 3.2 Reference](#)
- » [H4 in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

H5 - Level-five Heading



H5 - Level-five Heading

Syntax
<H5>...</
H5>
Attribute
Specifications

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j
u

s
t
i
f
y

J

(
h
o
r
i
z
o
n
t
a
l
a
l
l
i
g
n
m
e
n
t
)

u

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
APP
LET,
BLO
CKQ

UO
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DEL,
DD,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,
TH

The **H5** element defines a *level-five heading*. This heading is more important than an **H6** but less important than an **H4**.

The [deprecated ALIGN](#) attribute suggests the horizontal alignment for the heading on visual browsers. Possible values are **left**, **right**, **center**, and **justify**. [Style sheets](#) provide greater flexibility in suggesting alignment.

Visual browsers typically render **H5** in a small, bold font. Authors can suggest a presentation for **H5** through style sheets. The following [CSS](#) ruleset suggests a presentation for all **H5** elements:

```
h5 {  
  margin-left: 6%;  
  margin-right: 6%;  
  font-size: 110%;  
  font-weight: bold  
}
```

Authors should not choose a heading level based on the font size commonly used by visual browsers. The heading level should be chosen based on the heading's importance and placement in the document. An **H5** is typically used within a section headed by an **H4**.

More Information

- [H5 in W3C HTML 4.0 Recommendation](#)
- [H5 in W3C HTML 3.2 Recommendation](#)
- [H5 in HTML 2.0 Standard](#)
- [H5 in WDG HTML 3.2 Reference](#)
- [H5 in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

H6 - Level-six Heading



H6 - Level-six Heading

Syntax
<H6>...</
H6>
Attribute
Specifications

u
A
L
I
G
N
=
[
l
e
f
t
|
c
e
n
t
e
r
|
r
i
g
h
t
|
j
u

s
t
i
f
y

J

(
h
o
r
i
z
o
n
t
a
l
a
l
l
i
g
n
m
e
n
t
)

u

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
APP
LET,
BLO
CKQ

UO
T
E
BOD
Y
,
BUT
TON
,
CEN
TER
,
DEL
DD
,
DIV
FIEL
DSE
I
,
FOR
M
,
IFR
AME
,
INS
LI
,
MAP
,
NOF
RAM
ES
,
NOS
CRI
PT
,
OBJ
ECT
,
ID
,
TH

The **H6** element defines a *level-six heading*. This heading is less important than an **H5**. The **deprecated** **ALIGN** attribute suggests the horizontal alignment for the heading on visual browsers. Possible values are **left**, **right**, **center**, and **justify**. [Style sheets](#) provide greater flexibility in suggesting alignment.

Visual browsers typically render **H6** in a small, bold font. Authors can suggest a presentation for **H6** through style sheets. The following [CSS](#) ruleset suggests a presentation for all **H6** elements:

```
h6 {  
  margin-left: 6%;  
  margin-right: 6%;  
  font-size: 105%;  
  font-weight: bold  
}
```

Authors should not choose a heading level based on the font size commonly used by visual browsers. The heading level should be chosen based on the heading's importance and placement in the document. An **H6** is typically used within a section headed by an **H5**.

More Information

- » [H6 in W3C HTML 4.0 Recommendation](#)
- » [H6 in W3C HTML 3.2 Recommendation](#)
- » [H6 in HTML 2.0 Standard](#)
- » [H6 in WDG HTML 3.2 Reference](#)
- » [H6 in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

HR - Horizontal Rule



HR - Horizontal Rule

Syntax
<HR>
Attribute
Specifications

ALIGNMENT
[left | center | right]
(horizontal)

r
i
z
o
n
t
a
l
a
l
i
g
n
m
e
n
t
)

u
N
O
S
H
A
D
E

(
s
o
l
i
d
l
i
n
e
)

u
S
I
Z
E
=
P
i
X
E
I
S

(
l
i
n
e
h
e

i g h t)
= W I D T H =
line
(l i n e
w i d t h)
= c o l o r e
= a t t r i b u t e s
= c o m m o n
= e v e n

t
s

Cont
ents
Emp
ty
Cont
aine
d in
APP
LET,
BLO
CKQ
UOT
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DD,
DEL,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,
TH

The **HR** element defines a *horizontal rule* for visual browsers. While this element is inherently presentational, it can be used structurally as a section divider. However, for greater flexibility the **HR** element can be replaced with the [border-bottom](#) or [border-top](#) properties of [Cascading Style Sheets](#). For example, the following style rule would suggest a horizontal line above all [DIV](#) elements with **CLASS=navbar**:

```
div.navbar { border-top: solid medium navy }
```

HR's [deprecated ALIGN](#) attribute suggests the horizontal alignment of the line. Possible values are **left**, **right**, and **center**. The deprecated **WIDTH** attribute specifies the width of the line as a percentage or a number of pixels. If a width is specified, percentages are generally preferred since they adjust to varying window sizes. The [width](#) property

of [Cascading Style Sheets](#) provides greater flexibility in suggesting the width of horizontal rules.

The boolean **NOSHADE** attribute suggests that the rule be rendered as a solid line rather than the groove style commonly used. The **SIZE** attribute suggests the height of the line in pixels. These attributes are both [deprecated](#) in favor of [style sheets](#).

More Information

- [HR in W3C HTML 4.0 Recommendation](#)
- [HR in W3C HTML 3.2 Recommendation](#)
- [HR in HTML 2.0 Standard](#)
- [HR in WDG HTML 3.2 Reference](#)
- [HR in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#) [HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

PRE - Preformatted Text



PRE - Preformatted Text

Syntax
<PRE>...
</PRE>
Attribute
Specifications

```
W  
I  
D  
T  
H  
=  
N  
u  
m  
b  
e  
r  
  
(  
l  
i  
n  
e  
  
w  
i  
d  
t  
h  
)  
  
c  
o  
m  
m  
o  
n  
a
```

t
t
r
i
b
u
t
e
s

Cont
ents

Inlin
e
elem
ents

exce
pt

IMG,
OBJ
ECT

,
APP
LET,
BIG,
SMA
LL,
SUB

,
SUP

,
FON
I,
BAS
EFO
NT

Cont
aine
d in

APP
LET,
BLO
CKQ
UOT

E,
BOD
Y,
BUT
TON

,
CEN
TER

,
DD,
DEL,
DIV,
FIEL
DSE

I,
FOR
M,
IFR

AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, TD,
TH

The **PRE** element contains *preformatted text*. Visual browsers should render preformatted text in a fixed-pitch font, should not collapse whitespace, and should not wrap long lines.

PRE is useful for formatting computer code or poetry where whitespace is important, but since preformatted text is inherently visual, authors should avoid dependence on it wherever possible. When using **PRE**, authors should avoid altering the element's fixed-pitch font or non-collapsing whitespace properties by means of [style sheets](#).

The following example features Java code in a **PRE** element:

```
<PRE><CODE CLASS=Java>
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
</CODE></PRE>
```

The code in the preceding example would be more difficult to read if it were not preformatted, and so **PRE** is used to provide the proper spacing. Note that the **CODE** element is also included to add structural information. The **deprecated WIDTH** attribute of **PRE** tells the browser the expected line length of the preformatted block so that a suitable font size or margin can be used. Browsers ignore this attribute in practice.

The [bidirectional algorithm](#) that determines the directionality of text still applies within the **PRE** element.

More Information

- » [PRE in W3C HTML 4.0 Recommendation](#)
- » [PRE in W3C HTML 3.2 Recommendation](#)
- » [PRE in HTML 2.0 Standard](#)
- » [PRE in WDG HTML 3.2 Reference](#)
- » [PRE in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

DIR - Directory List



DIR - Directory List

Syntax
<DIR>...<
/DIR>
Attribute
Specifications

u
C
O
M
P
A
C
T

(
c
o
m
p
a
c
t

d
i
s
p
l
a
y
)

u
c
o
m
o
n
=

a
t
t
r
i
b
u
t
e
s

Cont
ents
One
or
mor
e LI
elem
ents
that
cann
ot
cont
ain
blo
k-
level
elem
ents
Cont
aine
d in
APP
LET
BLO
CKQ
UOT
E
BOD
Y
BUT
TON
,
CEN
TER
,
DD
DEL
DIV
FIEL
DSE
I
FOR
M
IFR
AME
,
INS
LI
MAP
,
NOF

[RAM](#)
[ES](#),
[NOS](#)
[CRI](#)
[PT](#),
[OBJ](#)
[ECT](#)
, [ID](#),
[TH](#)

The **DIR** element defines a *directory list*. The element contains one or more [LI](#) elements that define the actual items of the list. The **LI** elements must not contain [block-level elements](#), which prevents **DIRs** from being nested.

The **COMPACT** attribute suggests that visual browsers render the list compactly, perhaps with reduced spacing between items. This attribute is not well supported among browsers.

DIR is [deprecated](#) in HTML 4.0 in favor of [UL](#).

More Information

- » [DIR in W3C HTML 4.0 Recommendation](#)
- » [DIR in W3C HTML 3.2 Recommendation](#)
- » [DIR in HTML 2.0 Standard](#)
- » [DIR in WDG HTML 3.2 Reference](#)
- » [DIR in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

DL - Definition List



DL - Definition List

Syntax
<DL>...</DL>
Attribute Specifications

COMPACT

(compact

display)

compact

=

a
t
t
r
i
b
u
t
e
s

Cont
ents
One
or
mor
e DI
or
DD
elem
ents
Cont
aine
d in
APP
LET,
BLO
CKQ
UOT
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DD,
DEL,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,
INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,

TH

The **DL** element defines a *definition list*. An entry in the list is created using the **DT** element for the term being defined and the **DD** element for the definition of the term.

A definition list can have multiple terms for a given definition as well as multiple definitions for a given term. Authors can also give a term without a corresponding definition, and vice versa, but such a structure rarely makes sense.

An example follows:

<DL>

```
<DT>Block-level elements</DT>
<DD>
  <P>
    In HTML, block-level elements may generally contain
    inline elements and other block-level elements. They are
    usually formatted differently than inline elements,
    typically on a new line in visual browsers.
  </P>
</DD>
```

```
<DT>Inline elements</DT>
<DT>Text-level elements</DT>
<DD>
  <P>
    Inline (or text-level) elements generally only contain
    character data and other inline elements.
  </P>
</DD>
```

</DL>

The **DL** element can be adapted for use with structures that are not strict terms and definitions, a practice that is justified when other HTML elements cannot adequately describe a structure. Some examples follow:

u

```
<H1>Community Calendar</H1>
```

```
<DL CLASS=calendar>
```

```
<DT>March 8</DT>
```

```
<DD>
```

```
  The Symphony Orchestra presents <CITE>A Rising Star</CITE>  
  at the Anderson Center. Call 555-1234 for details.
```

```
</DD>
```

```
<DT>March 10</DT>
```

```
<DD>
```

```
  Bereaved Families Support Night, 7:00 to 9:00 at  
  523 Main <ABBR TITLE=Street>St.</ABBR>
```

```
</DD>
```

```
</DL>
```

```
u  
<DL CLASS=play>
```

```
<DT>Brutus</DT>
```

```
<DD CLASS="role Brutus">
```

```
<P>
```

```
  I kiss thy hand, but not in flattery, Caesar;<BR>  
  Desiring thee that Publius Cimber may<BR>  
  Have an immediate freedom of repeal.
```

```
</P>
```

```
</DD>
```

```
<DT>Caesar</DT>
```

```
<DD CLASS="role Caesar">
```

```
<P>
```

```
  What, Brutus!
```

```
</P>
```

```
</DD>
```

```
<DT>Cassius</DT>
```

```
<DD CLASS="role Cassius">
```

```
<P>
```

```
  Pardon, Caesar; Caesar, pardon:<BR>  
  As low as to thy foot doth Cassius fall,<BR>  
  To beg enfranchisement for Publius Cimber.
```

```
</P>
```

```
</DD>
```

```
</DL>
```

Note the use of the **CLASS** attribute in the preceding examples. This allows the author to easily suggest, through [style sheets](#), a distinguishing presentation for different kinds of definition lists. In addition to the [common attributes](#) shared by most elements, **DL** takes a **COMPACT** attribute. This attribute, [deprecated](#) in HTML 4.0 and poorly supported among browsers, suggests that visual browsers render the list compactly, perhaps with reduced spacing between items.

More Information

- » [DL in W3C HTML 4.0 Recommendation](#)
- » [DL in W3C HTML 3.2 Recommendation](#)
- » [DL in HTML 2.0 Standard](#)
- » [DL in WDG HTML 3.2 Reference](#)
- » [UL in *Learning HTML 3.2 by Examples*](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[Elements Alphabetically](#) [HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

DT - Definition Term



DT - Definition Term

Syntax
<DT>...</
DT>
Attribute
Specifications

C
O
M
O
N
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
DL

The **DT** element defines a *term* in a [definition list](#). The closing tag for **DT** is optional, but its use prevents common browser bugs with [style sheets](#). Note that **DT** cannot contain [block-level elements](#) such as **P** and **H2**.

A **DT** element should generally be followed by a **DD** element that provides the definition for the term given by the **DT**. A single definition term may have multiple definitions associated with it, and a single definition may have multiple terms.

More Information

- » [DT in W3C HTML 4.0 Recommendation](#)
- » [DT in W3C HTML 3.2 Recommendation](#)
- » [DT in HTML 2.0 Standard](#)
- » [DT in WDG HTML 3.2 Reference](#)
- » [DT in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

DD - Definition Description



DD - Definition Description

Syntax
<DD>...</
DD>
Attri
bute
Spe
cific
ation
s

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
bloc
k-
level
elem
ents
Cont
aine
d in
DL

The **DD** element provides the *definition* of a [term](#) in a [definition list](#). The closing tag for **DD** is optional, but its use prevents common browser bugs with [style sheets](#).

DD may contain [block-level elements](#) such as [P](#), [H2](#), [TABLE](#), and [DL](#). This allows definition lists to be nested, as in the following example:

```
<DL>

<DT><A NAME="spanning-tree">Spanning tree</A></DT>
<DD>
  <P>
    A spanning tree of a graph is a <A HREF="#tree">tree</A>
    that contains all the vertices of the graph. There are two
    main types of spanning trees:
  </P>

  <DL>
    <DT>BFS spanning tree</DT>
    <DD>
      A spanning tree formed by a breadth-first search on the graph.
    </DD>
    <DT>DFS spanning tree</DT>
    <DD>
      A spanning tree formed by a depth-first search on the graph.
    </DD>
  </DL>
</DD>

<DT><A NAME=tree>Tree</A></DT>
<DD>
  <P>
    A tree is a connected, undirected graph without cycles.
  </P>
</DD>

</DL>
```

A **DD** element should generally be preceded by a **DT** element that gives the term defined by the **DD**. A single definition term may have multiple definitions associated with it, and a single definition may have multiple terms.

More Information

- » [DD in W3C HTML 4.0 Recommendation](#)
- » [DD in W3C HTML 3.2 Recommendation](#)
- » [DD in HTML 2.0 Standard](#)
- » [DD in WDG HTML 3.2 Reference](#)
- » [DD in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

MENU - Menu List



MENU - Menu List

Syntax
<MENU>
..</MENU
>
Attri
bute
Spe
cific
ation
s

u
C
O
M
P
A
C
T

(
c
o
m
p
a
c
t

d
i
s
p
l
a
y
)

u
c
o
m
m
o
n

= a
t
t
r
i
b
u
t
e
s

Cont
ents
One
or
mor
e L
elem
ents
that
cann
ot
cont
ain
blo
k-
level
elem
ents
Cont
aine
d in
APP
LET
BLO
CKQ
UOT
E
BOD
Y
BUT
TON
,
CEN
TER
,
DD
DEL
DIV
FIEL
DSE
T
FOR
M
IFR
AME
,
INS
LI
MAP
,

[NOF](#)
[RAM](#)
[ES](#),
[NOS](#)
[CRI](#)
[PT](#),
[OBJ](#)
[ECT](#)
[, ID](#),
[TH](#)

The **MENU** element defines a *menu list*. The element contains one or more [LI](#) elements that define the actual items of the list. The **LI** elements must not contain [block-level elements](#), which prevents **MENU**s from being nested.

The **COMPACT** attribute suggests that visual browsers render the list compactly, perhaps with reduced spacing between items. This attribute is not well supported among browsers.

MENU is [deprecated](#) in HTML 4.0 in favor of [UL](#).

More Information

- [MENU in W3C HTML 4.0 Recommendation](#)
- [MENU in W3C HTML 3.2 Recommendation](#)
- [MENU in HTML 2.0 Standard](#)
- [MENU in WDG HTML 3.2 Reference](#)
- [MENU in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

OL - Ordered List



OL - Ordered List

Syntax
...</
OL>
Attribute
Specifications

"
T
Y
P
E
=
[
1
|
a
|
A
|
i
|
|
]
(
n
u
m
b
e
r
i

ng
style
)

“
START
=
N
u
m
er

(
starting
number
)

“
COMPACT

(
compact

d
i

s
p
l
a
y
)

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
One
or
mor
e L
elem
ents
Cont
aine
d in
APP
LET,
BLO
CKQ
UOT
E,
BOD
Y,
BUT
TON
,
CEN
TER
,
DD,
DEL,
DIV,
FIEL
DSE
I,
FOR
M,
IFR
AME
,

INS,
LI,
MAP
,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,
TH

The **OL** element defines an *ordered list*. The element contains one or more [LI](#) elements that define the actual items of the list.

Unlike with an unordered list ([UL](#)), the items of an ordered list have a *definite sequence*. Items in an ordered list are numbered by the browser.

The [deprecated](#) **TYPE** attribute of **OL** suggests the numbering style on visual browsers. The case-sensitive values are as follows:

- ⌞ **1** (decimal numbers: 1, 2, 3, 4, 5, ...)
- ⌞ **a** (lowercase alphabetic: a, b, c, d, e, ...)
- ⌞ **A** (uppercase alphabetic: A, B, C, D, E, ...)
- ⌞ **i** (lowercase Roman numerals: i, ii, iii, iv, v, ...)
- ⌞ **I** (uppercase Roman numerals: I, II, III, IV, V, ...)

The numbering style on an individual list item can be suggested using the **TYPE** attribute of [LI](#). The [list-style-type](#) property of [CSS](#) provides greater flexibility in suggesting numbering styles.

The [deprecated](#) **START** attribute suggests the starting number for the list and defaults to **1**. The value of **START** must be an integer, but the number may be presented in a different form (for example, as a Roman numeral). While this attribute is deprecated, there is currently no substitute for it in [Cascading Style Sheets](#).

The [deprecated](#) **COMPACT** attribute suggests that visual browsers render the list compactly, perhaps with reduced spacing between items. This attribute is not well supported among browsers.

More Information

- ⌞ [OL in W3C HTML 4.0 Recommendation](#)
- ⌞ [OL in W3C HTML 3.2 Recommendation](#)
- ⌞ [OL in HTML 2.0 Standard](#)
- ⌞ [OL in WDG HTML 3.2 Reference](#)
- ⌞ [OL in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

UL - Unordered List



UL - Unordered List

Syntax
...</
UL>
Attribute
Specifications

"
T
Y
P
E
=
[
d
i
s
c
|
s
q
u
a
r
e
|
c
i
r
c
l
e
]
(
b

u
l
l
e
t

s
t
y
l
e
)

u
C
O
M
P
A
C
T

(
c
o
m
p
a
c
t

d
i
s
p
l
a
y
)

u

c
o
m
o
n

=
a
t
t
r
i
b
u
t
e
s

Cont
ents
One

or
mor
e [LI](#)
elem
ents
Cont
aine
d in
[APP](#)
[LET](#),
[BLO](#)
[CKQ](#)
[UOT](#)
[E](#),
[BOD](#)
[Y](#),
[BUT](#)
[TON](#)
,
[CEN](#)
[TER](#)
,
[DD](#),
[DEL](#),
[DIV](#),
[FIEL](#)
[DSE](#)
[I](#),
[FOR](#)
[M](#),
[IFR](#)
[AME](#)
,
[INS](#),
[LI](#),
[MAP](#)
,
[NOF](#)
[RAM](#)
[ES](#),
[NOS](#)
[CRI](#)
[PT](#),
[OBJ](#)
[ECT](#)
, [TD](#),
[TH](#)

The **UL** element defines an *unordered list*. The element contains one or more [LI](#) elements that define the actual items of the list.

Unlike with an ordered list ([OL](#)), the items of an unordered list have *no sequence*. In theory, users should be able to change the order of items in an unordered list (e.g., alphabetizing them).

Visual browsers typically render **UL** with a bullet preceding each list item, but authors can suggest various presentations using style sheets. The [list-style](#) property of [Cascading Style Sheets](#) allows authors to suppress bullets, use images as list item markers, and more.

The [deprecated TYPE](#) attribute of **UL** suggests the bullet style on visual browsers. Possible values are as follows:

- **disc** (a filled-in circle)
- **square** (a square outline)
- **circle** (a circle outline)

The bullet style on an individual list item can be suggested using the **TYPE** attribute of [LI](#). The [list-style-type](#) property of [CSS](#) provides greater flexibility in suggesting bullet styles.

The [deprecated COMPACT](#) attribute suggests that visual browsers render the list compactly, perhaps with reduced spacing between items. This attribute is not well supported among browsers.

More Information

- [UL in W3C HTML 4.0 Recommendation](#)
- [UL in W3C HTML 3.2 Recommendation](#)
- [UL in HTML 2.0 Standard](#)
- [UL in WDG HTML 3.2 Reference](#)
- [UL in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

CAPTION - Table Caption



CAPTION - Table Caption

Syntax
<CAPTION>
<CAPTION>
<CAPTION>
Attribute
Specifications

u
A
L
I
G
N
=
[
t
o
p
|
b
o
t
t
o
m
|
l
e
f
t
|
r
i
g

h
t

]

(
c
a
p
t
i
o
n

a
l
i
g
n
m
e
n
t
)

u

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
TAB
LE

The **CAPTION** element defines a *table caption*. When used, **CAPTION** must be the first element in the **TABLE**. Only inline elements (e.g., **STRONG**) may be used within **CAPTION**.

A good caption should provide a short heading for the table. For simple tables, the caption can also act as an adequate summary, but for more complex tables, authors should supplement the **CAPTION** with a full summary, either through **TABLE**'s **SUMMARY** attribute or within a paragraph outside of the **TABLE**. The following example features a simple table where the **CAPTION** provides a heading and an adequate table summary:

```
<TABLE>
  <CAPTION>Common Usenet Abbreviations</CAPTION>
  <THEAD>
    <TR>
      <TH>Abbreviation</TH>
      <TH>Long Form</TH>
    </TR>
  </THEAD>
  <TBODY>
    <TR>
      <TD>AFAIK</TD>
      <TD>As Far As I Know</TD>
    </TR>
    <TR>
      <TD>IMHO</TD>
      <TD>In My Humble Opinion</TD>
    </TR>
    <TR>
      <TD>OTOH</TD>
      <TD>On The Other Hand</TD>
    </TR>
  </TBODY>
</TABLE>
```

The next example uses **TABLE**'s **SUMMARY** attribute to complement the **CAPTION**:

```
<TABLE SUMMARY="This table gives the character entity reference,
                decimal character reference, and hexadecimal character
                reference for symbols and Greek letters.">
  <CAPTION>Symbols and Greek Letters in HTML 4.0</CAPTION>
  <COLGROUP>
  <COLGROUP SPAN=3>
  <THEAD>
    <TR>
      <TH SCOPE=col>Character</TH>
      <TH SCOPE=col>Entity</TH>
      <TH SCOPE=col>Decimal</TH>
      <TH SCOPE=col>Hex</TH>
    </TR>
  </THEAD>
  <TBODY>
    <TR>
      <TD SCOPE=row>Latin small f with hook</TD>
      <TD>&fopf;</TD>
      <TD>&#402;</TD>
      <TD>&#x192;</TD>
    </TR>
    ...
  </TBODY>
</TABLE>
```

The **deprecated** **ALIGN** attribute of **CAPTION** specifies the alignment of the caption relative to the table. Possible values are **top** (the default), **bottom**, **left**, and **right**.

More Information

- [CAPTION in W3C HTML 4.0 Recommendation](#)
- [CAPTION in W3C HTML 3.2 Recommendation](#)
- [CAPTION in WDG HTML 3.2 Reference](#)
- [CAPTION in *Learning HTML 3.2 by Examples*](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

TR - Table Row



TR - Table Row

Syntax
<TR>...</
TR>
Attribute
Specifications

align="left|center|right|justify"

s
t
i
f
y

l
c
h
a
r
l
(
h
o
r
i
z
o
n
t
a
l
a
l
i
g
n
m
e
n
t

o
f

c
e
l
l
s

i
n

g
r
o
u
p
)

u
C
H
A
R
=
C

b
a
r
a
c
t
r

(a
l
i
g
n
m
e
n
t

c
h
a
r
a
c
t
e
r

f
o
r

c
e
l
l
s
)

“ C
H
A
R
O
F
F
=

I
e
r
g
t
p

(a
l
i
g

n
m
e
n
t

c
h
a
r
a
c
t
e
r

o
f
f
s
e
t
)
"
V
A
L
I
G
N
=
[
t
o
p
|
m
i
d
d
l
e
|
b
o
t
t
o
m
|
b
a
s
e

l
i
n
e

] (v e r t i c a l a l i g n m e n t o f c e l l s i n g r o u p)
" B G C O L O R = C
O
I
O
R
(r

o
w

b
a
c
k
g
r
o
u
n
d

c
o
l
o
r
)

u

c
o
m
m
o
n

=
a
t
t
r
i
b
u
t
e
s

Cont
ents
One
or
mor
e TH
or
ID
elem
ents
Cont
aine
d in
THE
AD,
TEO
OT,
TBO
DY

The **TR** element defines a *table row*. **TR** elements must be contained within a *row group* defined by **THEAD**, **TFOOT**, or **TBODY**. Since the start and end tags of **TBODY** are optional when the table has one **TBODY** and no **THEAD** or **TFOOT**, the simple table structure of [HTML 3.2](#) is still valid:

```
<TABLE>
  <TR>
    <TH>Abbreviation</TH>
    <TH>Long Form</TH>
  </TR>
  <TR>
    <TD>AFAIK</TD>
    <TD>As Far As I Know</TD>
  </TR>
</TABLE>
```

TR contains **TH** or **TD** elements, which in turn contain the actual data of the table. In addition to the [attributes common to most elements](#), **TR** takes presentational attributes for specifying the alignment of cells within the row and the row's background color.

The **ALIGN** attribute specifies the horizontal alignment for each cell in the row. Possible values are **left**, **center**, **right**, **justify**, and **char**. **ALIGN=char** aligns a cell's contents on the character given in the **CHAR** attribute. The default value for the **CHAR** attribute is the decimal point of the current language--a period in English. The **CHAROFF** attribute specifies the offset to the first occurrence of the alignment character. The attribute's value is a number in pixels or a percentage of the cell's width; **CHAROFF="50%"** centers the alignment character horizontally in a cell.

The **VALIGN** attribute specifies the vertical position of a cell's contents. Possible values are:

- **top**, which positions data at the top of the cell;
- **middle**, the default value, which centers the cell data vertically;
- **bottom**, which positions data at the bottom of the cell;
- **baseline**, which specifies that the first line of each cell in the row with **ALIGN=baseline** should occur on a common baseline.

The [deprecated](#) **BGCOLOR** attribute suggests a background color for the row. The combination of this attribute with **** can leave invisible or unreadable text on Netscape Navigator 2.x, which does not support **BGCOLOR** on table elements. **BGCOLOR** is dangerous even on supporting browsers, since most fail to override it when overriding other author-specified colors. [Style sheets](#) provide a safer, more flexible method of specifying a row's background color.

More Information

- [TR in W3C HTML 4.0 Recommendation](#)
- [TR in W3C HTML 3.2 Recommendation](#)
- [TR in WDG HTML 3.2 Reference](#)
- [TR in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

FORM - Interactive Form



FORM - Interactive Form

Syntax
<FORM>
..
</FORM
>
Attri
bute
Spe
cific
ation
s

u
A
C
T
I
O
N
=
U
R
I
(
f
o
r
m
h
a
n
d
l
e
r
)
u
M
E
T
H
O
D
=
=

```
[
  get
  |
  post
]
(HTTP
method
for
submitting
form)
"
ENCTYPE
=
Content-Type
```

Preprint

(content

type

to

submit

form

as)

“

ACCEPT - CHARS E T =

Preprint

|||t|||@

(s u p p o r t e d

c h a r a c t e r

e n c o d i n g s)

" T A R G E T =

|||F|||@
|||B|||@
|||F|||@
|||P|||@
|||C|||@
|||t|||@

(f r a m e

t
o

r
e
n
d
e
r

f
o
r
m

r
e
s
u
l
t

i
n
)
“
O
N
S
U
B
M
I
T
=

[S
i
c
i
p
i
t](#)

(
f
o
r
m

w
a
s

s
u
b
m
i
t
t
e

d
)

u

O
N
R
E
S
E
T
=

S
I
C
I
L
I
T
I
T
I
T

(
f
o
r
m

w
a
s

r
e
s
e
t
)

u

C
O
M
M
O
N

=
a
t
t
r
i
b
u
t
e
s

Cont
ents

u

I
n

HIML40Strict,
o n e
o r
m o r e
SRIPH
o r
bloklevelements
e
x

c
e
p
t

F
O
R
M

u
i
n

H
I
M
L

=
4

0

=
T

r

a

n

s

i

t

i

o

n

a

l

,

i

n

!

i

n

e

=

e

l

e

m

e

n

t

s

o

r

b

l

o

c

k

:

l
e
v
e
l
e
l
e
m
e
n
t
s

e
x
c
e
p
t

F
O
R
M

Cont
aine
d in
APP
LET,
BLO
CKQ
UOT
E,
BOD
Y,
CEN
TER

,
DD,
DEL,
DIV,
FIEL
DSE
I,
IFR
AME

,
INS,
LI,
MAP

,
NOF
RAM
ES,
NOS
CRI
PT,
OBJ
ECT
, ID,

TH

The **FORM** element defines an *interactive form*. The element should contain form controls--[INPUT](#), [SELECT](#), [TEXTAREA](#), and [BUTTON](#)--through which the user interacts.

When the user submits the form, through an **INPUT** or **BUTTON** element with **TYPE=submit**, the form values are submitted to the URI given in **FORM**'s required **ACTION** attribute. **ACTION** usually points to a CGI script or Java servlet that handles the form submission.

A mailto URI (e.g., <mailto:liam@htmlhelp.com>) is also allowed as an **ACTION**, but this is not supported by all browsers. Non-supporting browsers such as Microsoft Internet Explorer 3.x typically will open a blank e-mail message when the user submits a mailto form. Even on supporting browsers, mailto forms are troublesome in that they fail to provide feedback to the user after the form submission.

[Free CGI scripts](#) exist for handling forms; some are even [remotely hosted](#) for authors whose providers refuse to allow CGI scripts to be run locally.

How the form input is sent to the server depends on the **METHOD** and **ENCTYPE** attributes. When the **METHOD** is **get** (the default), the form input is submitted as an HTTP GET request with `?form_data` appended to the URI specified in the **ACTION** attribute.

Using the **get** method allows the form submission to be contained completely in a URL. This can be advantageous in that it permits bookmarking in current browsers, but it also prevents form data from containing non-ASCII characters such as "é" and "©". As well, the amount of form data that can be handled by the **get** method is limited by the maximum length of the URL that the server and browser can process. To be safe, any form whose input might contain non-ASCII characters or more than 100 characters should use **METHOD=post**.

With a **METHOD** value of **post**, the form input is submitted as an HTTP POST request with the form data sent in the body of the request. Most current browsers are unable to bookmark POST requests, but POST does not entail the character encoding and length restrictions imposed by GET.

The **ENCTYPE** attribute specifies the content type used in submitting the form, and defaults to **application/x-www-form-urlencoded**. This content type results in name/value pairs sent to the server as `name1=value1&name2=value2...` with space characters replaced by "+" and reserved characters (like "#") replaced by "%HH" where HH is the ASCII code of the character in hexadecimal. Line breaks are encoded as "%0D%0A"--a carriage return followed by a line feed.

Authors should generally only use a different **ENCTYPE** when the form includes a **TYPE=file INPUT** element, in which case the **ENCTYPE** should be **multipart/form-data** and the **METHOD** must be **post**. The format of multipart/form-data requests is given in [RFC 1867](#).

Tools such as [cg-eye](#) allow authors to easily create and view a request, simulating the submission of a form. However, authors often do not need to concern themselves with the exact format of the submission; CGI libraries including [CGI.pm](#) transparently handle **get** and **post** submissions sent as `application/x-www-form-urlencoded` or `multipart/form-data`.

The **ACCEPT-CHARSET** attribute specifies a list of character encodings that are accepted by the form handler. The value consists of a list of "charsets" separated by commas and/or spaces. The default value is **UNKNOWN** and is usually considered to be the character encoding used to transmit the document containing the **FORM**.

The **TARGET** attribute is used with [frames](#) to specify in which frame the form response should be rendered. If no frame with such a name exists, the response is rendered in a new window unless overridden by the user. Special frame names begin with an underscore:

- **_blank** renders the response in a new, unnamed window
- **_self** renders the response in the current frame (useful for overriding a **BASE TARGET**)
- **_parent** renders the response in the immediate **FRAMESET** parent
- **_top** renders the response in the full, unframed window

The **FORM** element also takes a number of attributes to specify client-side scripting actions for various events. In addition to the [core events](#) common to most elements, **INPUT** accepts the following event attributes:

- **ONSUBMIT**, when the form is submitted;
- **ONRESET**, when the form is reset.

More Information

- [FORM in W3C HTML 4.0 Recommendation](#)
- [FORM in W3C HTML 3.2 Recommendation](#)
- [FORM in HTML 2.0 Standard](#)
- [FORM in WDG HTML 3.2 Reference](#)
- [FORM in *Learning HTML 3.2 by Examples*](#)
- [RFC 1867: Form-based File Upload in HTML](#)
- [The Common Gateway Interface](#)
- [Java Servlets](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

INPUT - Form Input



INPUT - Form Input

Syntax
<INPUT>
Attribute
Specifications

u
T
Y
P
E
=
[
t
e
x
t
|
p
a
s
s
w
o
r
d
|
c
h
e
c
k
b
o
x
|

r
a
d
i
o

|
s
u
b
m
i
t

|
r
e
s
e
t

|
f
i
l
e

|
h
i
d
d
e
n

|
i
m
a
g
e

|
b
u
t
t
o
n

] (t y

p
e

o
f

i
n
p
u
t
)

«
N
A
M
E
=
C
D
A
T
A

(
k
e
y

i
n

s
u
b
m
i
t
t
e
d

f
o
r
m
)

«
V
A
L
U
E
=
C
D
A
T
A

(
v
a
l
u
e

o
f

i
n
p
u
t
)
"
C
H
E
C
K
E
D

(
c
h
e
c
k

r
a
d
i
o

b
u
t
t
o
n

o
r

c
h
e
c
k
b
o
x
)
"
S
I

Z
E
=
C
D
A
T
A

(
s
u
g
g
e
s
t
e
d

n
u
m
b
e
r

o
f

c
h
a
r
a
c
t
e
r
s

f
o
r

t
e
x
t

i
n
p
u
t
)

u
M
A
X
L

LENGTH = ~~14~~ ~~15~~ ~~16~~ ~~17~~ ~~18~~
(maximum number of characters for text input)
SR

C
=
U
R
I

(s
o
u
r
c
e

f
o
r

i
m
a
g
e
)

“
A
L
T
=
C
D
A
T
A

(a
l
t
e
r
n
a
t
e

t
e
x
t

f
o
r

i
m
a
g
e

i
n
p
u
t
)

u
U
S
E
M
A
P
=
U
I
I

(
c
l
i
e
n
t
-
s
i
d
e

i
m
a
g
e

m
a
p
)

u
A
L
I
G
N
=
[

t
o
p

|

m
i
d
d

l
e

l
b
o
t
t
o
m

l
l
e
f
t

l
r
i
g
h
t

l
(
a
l
i
g
n
m
e
n
t

o
f

i
m
a
g
e

i
n
p
u
t
)
u
D
I
S
A
B

L
E
D

(
d
i
s
a
b
l
e

e
l
e
m
e
n
t
)

“
R
E
A
D
O
N
L
Y

(
p
r
e
v
e
n
t

c
h
a
n
g
e
s
)

“
A
C
C
E
P
T
=
C
o
n
t
r

h
o
r
t
c
u
t

k
e
y
)

“

T
A
B
I
N
D
E
X

=

N

u

m

b

e

r

(

p

o

s

i

t

i

o

n

i

n

t

a

b

b

i

n

g

o

r

d

e

r

)

“

O

N

F

O
C
C
S
=

(
e
l
e
m
e
n
t

r
e
c
e
i
v
e
d

f
o
c
u
s
)

°
O
N
B
L
U
R
=

(
e
l
e
m
e
n
t

(
e
l
e
m
e
n
t

I

o
s
t

f
o
c
u
s
)

“
O
N
S
E
L
E
C
T
=
s
c
r
i
p
t

(
e
l
e
m
e
n
t

t
e
x
t

s
e
l
e
c
t
e
d
)

“
O
N
C
H
A
N
G
E
=
=

S
C
i
p
t

(
e
l
e
m
e
n
t

v
a
l
u
e

c
h
a
n
g
e
d
)

u

c
o
m
o
n

=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Emp
ty
Cont
aine
d in
Bloc
k-
level
elem
ents,

[inline
elements
except
BUTTON](#)

The **INPUT** element defines a *form control* for the user to enter input. While **INPUT** is most useful within a **FORM**, HTML 4.0 allows **INPUT** in any [block-level](#) or [inline](#) element other than **BUTTON**. However, [Netscape Navigator](#) will not display any **INPUT** elements outside of a **FORM**.

When a form is submitted, the current value of each **INPUT** element within the **FORM** is sent to the server as name/value pairs. The **INPUT** element's **NAME** attribute provides the name used. The value sent depends on the type of form control and on the user's input.

The type of form control defined by **INPUT** is given by the **TYPE** attribute. The default **TYPE** is **text**, which provides a single-line text input field. The **VALUE** attribute specifies the initial value for the text field. The **SIZE** and **MAXLENGTH** attributes suggest the number of characters and maximum number of characters, respectively, of the text field.

While the **MAXLENGTH** attribute can be an effective guide to the user, authors should not depend on the enforcement of a maximum number of characters by the client. A user could copy the HTML document, remove the **MAXLENGTH** attribute, and submit the form. Thus authors of form handlers should ensure that any necessary input length checks are repeated on the server-side.

The **password** input type is a variation on the **text** type. The only difference is that the input characters are masked, typically by a series of asterisks, to protect sensitive information from onlookers. Note, however, that the actual value is transmitted to the server as clear text, so **password** inputs do not provide sufficient security for credit card numbers or other highly sensitive information.

The following example uses **text** and **password** fields with the [LABEL](#) element to bind text labels to the **INPUT** elements:

```
<P><LABEL ACCESSKEY=U>User name: <INPUT TYPE=text NAME=username SIZE=8  
MAXLENGTH=8></LABEL></P>  
<P><LABEL ACCESSKEY=P>Password: <INPUT TYPE=password NAME=pw SIZE=12  
MAXLENGTH=12></LABEL></P>
```

The boolean **READONLY** attribute, new in HTML 4.0 and poorly supported by current browsers, prevents the user from editing the content of the **text** or **password** input types. Read-only elements are still submitted with the form. The **DISABLED** attribute, which applies to all input types but is also poorly supported, disables the control. Disabled elements are read-only elements with the added restrictions that the values are not submitted with the form, the elements cannot receive focus, and the elements are skipped when navigating the document by tabbing.

The **radio** and **checkbox** input types provide switches that can be turned on and off by the user. The two types differ in that radio buttons are grouped (by specifying the same **NAME** attribute on each **INPUT**) so that only one radio button in a group can be selected at any time. Checkboxes can be checked without changing the state of other checkboxes with the same **NAME**. The **VALUE** attribute, required for radio buttons and checkboxes, gives the value of the control when it is checked. The boolean **CHECKED** attribute specifies that the control is initially checked.

Some browsers require one radio button in a group to be selected at all times. To ensure that an appropriate default choice is made, authors may wish to define one of the radio **INPUT** elements as **CHECKED**.

In the following example, only one payment method may be selected by the user since the radio buttons have the same **NAME**:

```
<P>Please indicate your method of payment:</P>
<P><LABEL ACCESSKEY=C><INPUT TYPE=radio NAME="payment_method" VALUE="credit card"
CHECKED> Credit card</LABEL><BR>
<LABEL ACCESSKEY=D><INPUT TYPE=radio NAME="payment_method" VALUE="debit card"> Debit
card</LABEL><BR>
<LABEL ACCESSKEY=M><INPUT TYPE=radio NAME="payment_method" VALUE="money order"> Money
order</LABEL></P>
```

```
<P><LABEL ACCESSKEY=S><INPUT TYPE=checkbox NAME="send_receipt" VALUE="yes" CHECKED>
Send receipt by e-mail</LABEL></P>
```

The **file** input type creates a field through which users can upload files from their local computer or network. The **VALUE** attribute specifies the name of the initial file, but it is typically ignored by browsers as a security precaution. The **ACCEPT** attribute gives a comma-separated list of media types accepted, allowing the browser to filter out inappropriate files. Current browsers generally ignore the **ACCEPT** attribute.

A form that includes a **file INPUT** must specify **METHOD=post** and **ENCTYPE="multipart/form-data"** in the **<FORM>** tag. CGI libraries such as [CGI.pm](#) allow simple handling of such forms.

Form-based file upload is unsupported by many currently deployed browsers. Authors should provide alternative methods of input where possible.

The following example allows the user to upload an HTML document for validation:

```
<FORM METHOD=post ACTION="/cgi-bin/validate.cgi" ENCTYPE="multipart/form-data">
<P>Select an HTML document to upload and validate. If your browser does not support
form-based file upload, use one of our <A HREF="methods.html">alternate methods of
validation</A>.</P>
<P><INPUT TYPE=file NAME="html_file" ACCEPT="text/html"></P>
<P><INPUT TYPE=submit VALUE="Validate it!"></P>
</FORM>
```

The **hidden** input type allows authors to include form data without having it rendered to the user. This is particularly useful in form applications that span several HTML documents; user input can be carried from form to form by **hidden INPUTs**. Some generalized CGI scripts use **hidden INPUTs** to define variables for the script, as in the following example, which defines a recipient and subject for the e-mailed contents of a form:

```
<INPUT TYPE=hidden NAME=recipient VALUE="liam@htmlhelp.com">
<INPUT TYPE=hidden NAME=subject VALUE="Feedback on your HTML Reference">
```

Note that the fields are "hidden" in the sense that they are not rendered by the browser. Anyone can still view the HTML document's source to find the "hidden" fields.

The **TYPE** value **reset** defines a button by which the user can reset the form to its initial values. The optional **VALUE** attribute of a reset button overrides the browser's default text for the button.

The **submit** input type defines a button for submitting the form. As with **reset**, the optional **VALUE** attribute provides the text of the button. The presence of the **NAME** attribute will cause the browser to send a name/value pair for the submit button if it is used to submit the form. This allows authors to provide multiple submit buttons and have the form handler take a different action depending on the submit button used.

The **image** input type specifies a graphical submit button. The **SRC** attribute must be included to specify the URI of the image. The **ALT** attribute should be used to give replacement text for those not loading images. **ALT** is a new addition in HTML 4.0; many browsers rely on either the **NAME** or **VALUE** attribute as alternate text, so authors should use all three attributes for the same purpose where possible. The topic of graphical submit buttons for text users is discussed in detail in the article [INPUT TYPE=IMAGE for text users?](#).

When the graphical submit button is clicked, the coordinates of the click are sent with the form submission as `name.x=x-value` and `name.y=y-value` where `name` is the value of the **NAME** attribute, `x-value` is the click's pixels from the left of the image, and `y-value` is the click's pixels from the top of the image. The **USEMAP** attribute combined with **TYPE=image** defines a client-side image map that can be used with client-side scripting, but this method is poorly supported. The **USEMAP** attribute gives the URI of the defining **MAP**. The **deprecated ALIGN** attribute specifies the alignment of the graphical submit button. The values **top**, **middle**, and **bottom** specify the button's position with respect to surrounding content on its left and right. The values **left** and **right** specify a *floating* button; the image is placed at the left or right margin and content flows around it. To place content below the button, use `<BR CLEAR=left|right|all>` as appropriate.

The [vertical-align](#) and [float](#) properties of [Cascading Style Sheets](#) provide more flexible methods of aligning buttons.

The input type **button** specifies a push button for use with client-side scripting. The **VALUE** attribute gives the text label of the button. The **ONCLICK** attribute is typically used to define the action taken when the button is activated. An example follows:

```
<INPUT TYPE=button VALUE="Hide non-strict attributes" ID=toggler ONCLICK="toggle()">
```

In this example, the **toggle()** function, which would be defined earlier in a **SCRIPT** element, will be executed when the button is clicked. Since the button is only useful with client-side scripting enabled, authors should usually output the `<INPUT TYPE=button>` tag using the scripting language to avoid providing a non-functioning button to some users. A more complete version of the previous example would thus be as follows:

```
<SCRIPT TYPE="text/javascript">
<!--
  document.write("<INPUT TYPE=button VALUE=\"Hide non-strict attributes\"
                + \"ID=toggler ONCLICK=\"toggle()\">");
// -->
</script>
```

The **BUTTON** element allows richer labels for submit, reset, and push buttons, but a lack of browser support makes **INPUT** a more reliable choice at this time.

The **ACCESSKEY** and **TABINDEX** attributes apply to all input types except **hidden**. **ACCESSKEY** specifies a single Unicode character as a shortcut key for giving focus to the form control. Authors can set the access key on the **INPUT** element or the **LABEL** element associated with it. [Entities](#) (e.g. **´**) may be used as the **ACCESSKEY** value.

The **TABINDEX** attribute specifies a number between 0 and 32767 to indicate the tabbing order of the element. A form control with **TABINDEX=0** or no **TABINDEX** attribute will be visited after any elements with a positive **TABINDEX**. Among positive **TABINDEX** values, the lower number receives focus first. In the case of a tie, the element appearing first in the HTML document takes precedence.

The **INPUT** element also takes a number of attributes to specify client-side scripting actions for various events. In addition to the [core events](#) common to most elements, **INPUT** accepts the following event attributes:

- **ONFOCUS**, when the element receives focus;
- **ONBLUR**, when the element loses focus;
- **ONSELECT**, when text in an input of type **text** or **password** is selected;
- **ONCHANGE**, when the element loses focus and its value has changed since it received focus.

More Information

- [INPUT in W3C HTML 4.0 Recommendation](#)
- [INPUT in W3C HTML 3.2 Recommendation](#)
- [INPUT in HTML 2.0 Standard](#)
- [INPUT in WDG HTML 3.2 Reference](#)
- [INPUT in Learning HTML 3.2 by Examples](#)
- [RFC 1867: Form-based File Upload in HTML](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

SELECT - Option Selector



SELECT - Option Selector

Syntax
<SELEC
T>...</SE
LECT>
Attri
bute
Spe
cific
ation
s

u
N
A
M
E
=
C
D
A
T
A
(
k
e
y

i
n

s
u
b
m
i
t
t
e
d

f
o
r
m
)

u
M
U
L
T
I
P
L
E

(
a
l
l
o
w

m
u
l
t
i
p
l
e

s
e
l
e
c
t
i
o
n
s
)

u
S
I
Z
E
=

N
u
m
b
l
e
r

(
n
u
m
b
e
r

o
f

V
I
S
I
B
L
E

O
P
T
I
O
N
S
)

“
D
I
S
A
B
L
E
D

(
d
i
s
a
b
l
e

e
l
e
m
e
n
t
)

“
T
A
B
I
N
D
E
X
=
N
U
B
R
I

(p o s i t i o n

i n

t a b b i n g

o r d e r)

"

O N F O C U S

=

|||

|||

|||

|||

|||

|||

(e l e m e n t

r e c e i v e d

f
o
c
u
s
)

°

O
N
B
L
U
R

=

S
I
C
I
R
I
T
P

(
e
l
e
m
e
n
t

l
o
s
t

f
o
c
u
s
)

°

O
N
C
H
A
N
G
E

=

S
I
C
I
R
I
T
P

(

element value changed)

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
One
or
more
OPT
GR
OUP
or
OPT
ION
elements
Contained in
Bloc
k-
level

[elem
ents,
inlin
e
elem
ents
exce
pt
BUT
TON](#)

The **SELECT** element defines a form control for the *selection of options*. While **SELECT** is most useful within a **FORM**, HTML 4.0 allows **SELECT** in any [block-level](#) or [inline](#) element other than **BUTTON**. However, [Netscape Navigator](#) will not display any **SELECT** elements outside of a **FORM**.

The **SELECT** element contains one or more **OPTGROUP** or **OPTION** elements to provide a menu of choices for the user. Each choice is contained within an **OPTION** element. Choices can be grouped logically through the **OPTGROUP** element. **SELECT**'s **NAME** attribute provides the key sent to the server with the value of the selected option.

By default, the user can only select one option. The boolean **MULTIPLE** attribute allows the user to select multiple options, which are submitted as separate name/value pairs. The following example uses the **MULTIPLE** attribute to allow the selection of one or more options:

```
<P>Select one or more sections to search:  
<SELECT NAME=sections MULTIPLE>  
<OPTION>Web Authoring Reference</OPTION>  
<OPTION>FAQ Archives</OPTION>  
<OPTION>Design Elements</OPTION>  
<OPTION>Tools</OPTION>  
<OPTION>Feature Article</OPTION>  
</SELECT>  
</P>
```

The boolean **DISABLED** attribute, new in HTML 4.0 and poorly supported by current browsers, makes the **SELECT** element unavailable. The user is unable to edit the disabled selection, no value is submitted with the form, the **SELECT** element cannot receive focus, and the element is skipped when navigating the document by tabbing.

The **TABINDEX** attribute specifies a number between 0 and 32767 to indicate the tabbing order of the element. A **SELECT** element with **TABINDEX=0** or no **TABINDEX** attribute will be visited after any elements with a positive **TABINDEX**. Among positive **TABINDEX** values, the lower number receives focus first. In the case of a tie, the element appearing first in the HTML document takes precedence.

The **SIZE** attribute of **SELECT** hints that visual browsers should display the element as a list box with the specified number of options visible at any time. A scroll bar would allow access to any non-visible options. The **SIZE** attribute is especially useful in **SELECT** elements with numerous **OPTIONS** and multiple selections allowed. In such a situation, some visual browsers will render the entire list in one large box without scrolling; a suitable **SIZE** attribute helps such browsers give a more appropriate presentation.

The **SELECT** element also takes a number of attributes to specify client-side scripting actions for various events. In addition to the [core events](#) common to most elements, **SELECT** accepts the following event attributes:

• **ONFOCUS**, when the element receives focus;

- **ONBLUR**, when the element loses focus;
- **ONCHANGE**, when the element loses focus and its value has changed since it received focus.

More Information

- [SELECT in W3C HTML 4.0 Recommendation](#)
- [SELECT in W3C HTML 3.2 Recommendation](#)
- [SELECT in HTML 2.0 Standard](#)
- [SELECT in WDG HTML 3.2 Reference](#)
- [SELECT in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

OPTION - Menu Option



OPTION - Menu Option

Syntax
<OPTION
>...</OPT
ION>
Attri
bute
Spe
cific
ation
s

u

V

A

L

U

E

=

C

D

A

T

A

(

v

a

l

u

e

o

f

o

p

t

i

o

n

)

u

S

E

L

E
C
T
E
D

(
c
h
o
i
c
e

i
n
i
t
i
a
l
l
y

s
e
l
e
c
t
e
d
)

“
D
I
S
A
B
L
E
D

(
d
i
s
a
b
l
e

c
h
o
i
c
e
)

“

L
A
B
E
L
=
T
e
x
t

(
o
p
t
i
o
n

l
a
b
e
l
)

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
e
s

Cont
ents
Plain
text
(incl
udin
g
entiti
es)
Cont
aine
d in
SEL
ECT
,
OPT

GR ROUP

The **OPTION** element defines a *menu choice* within a **SELECT** menu. The value of the option, sent with a submitted form, is specified with the **VALUE** attribute. In the absence of a **VALUE** attribute, the value is the content of the **OPTION** element.

The boolean **SELECTED** attribute defines the **OPTION** to be initially selected. A **SELECT** element can only have one **OPTION** selected at any time unless the **MULTIPLE** attribute is present on **SELECT**.

If the **SELECT** element does not use the **MULTIPLE** or **SIZE** attributes, some browsers will automatically (and incorrectly) select an option. To ensure that a suitable option is selected, authors may wish to use the **SELECTED** attribute on an **OPTION**. If no option is a suitable default, consider using a dummy option, as in the following example:

```
<SELECT NAME="marital_status">
<OPTION SELECTED VALUE="">Select...</OPTION>
<OPTION>Single</OPTION>
<OPTION>Married</OPTION>
<OPTION>Separated</OPTION>
<OPTION>Divorced</OPTION>
<OPTION>Widowed</OPTION>
</SELECT>
```

The boolean **DISABLED** attribute, new in HTML 4.0 and poorly supported by current browsers, makes the **OPTION** element unavailable. A disabled option cannot be selected by the user and is never submitted with the form.

The **LABEL** attribute specifies the option label presented to the user. This defaults to the content of the **OPTION** element, but the **LABEL** attribute allows authors to more easily use **OPTGROUP** without sacrificing compatibility with browsers that do not support option groups. The following example illustrates the technique:

```
<P>Which Web browser do you use most often?
<SELECT NAME=browser>
  <OPTGROUP LABEL="Netscape Navigator">
    <OPTION LABEL="4.x or higher">
      Netscape Navigator 4.x or higher
    </OPTION>
    <OPTION LABEL="3.x">Netscape Navigator 3.x</OPTION>
    <OPTION LABEL="2.x">Netscape Navigator 2.x</OPTION>
    <OPTION LABEL="1.x">Netscape Navigator 1.x</OPTION>
  </OPTGROUP>
  <OPTGROUP LABEL="Microsoft Internet Explorer">
    <OPTION LABEL="4.x or higher">
      Microsoft Internet Explorer 4.x or higher
    </OPTION>
    <OPTION LABEL="3.x">Microsoft Internet Explorer 3.x</OPTION>
    <OPTION LABEL="2.x">Microsoft Internet Explorer 2.x</OPTION>
    <OPTION LABEL="1.x">Microsoft Internet Explorer 1.x</OPTION>
  </OPTGROUP>
  <OPTGROUP LABEL="Opera">
    <OPTION LABEL="3.x or higher">Opera 3.x or higher</OPTION>
    <OPTION LABEL="2.x">Opera 2.x</OPTION>
  </OPTGROUP>
  <OPTION>Other</OPTION>
</SELECT>
</P>
```

OPTGROUP and **OPTION**'s **LABEL** attribute were introduced together, so browsers should support both features or neither of them. Supporting browsers will render the preceding example using the **LABEL** attribute of **OPTION** to provide just the version number, along with the **OPTGROUP**'s **LABEL**, which gives the full name of the application. This allows a compact display with cascading menus, but many browsers do not yet support **OPTGROUP**. These browsers will ignore the **OPTGROUP** elements and **LABEL** attributes, providing the full name and version for each choice. Thus authors can fully use **OPTGROUP** despite its lack of browser support.

More Information

- [OPTION in W3C HTML 4.0 Recommendation](#)
- [OPTION in W3C HTML 3.2 Recommendation](#)
- [OPTION in HTML 2.0 Standard](#)
- [OPTION in WDG HTML 3.2 Reference](#)
- [OPTION in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

TEXTAREA - Multi-line Text Input



TEXTAREA - Multi-line Text Input

Syntax
<TEXTAREA>...</TEXTAREA>
Attributes

NAME
= C
D
A
T
A
(key
in
submitted
form

)

°
R
O
W
S
=
N
U
B
E
R

(
n
u
m
b
e
r
o
f
r
o
w
s
)

°
C
O
L
S
=
N
U
B
E
R

(
n
u
m
b
e
r
o
f
c
o
l
u
m
n

s
)

u
D
I
S
A
B
L
E
D

(
d
i
s
a
b
l
e
e
l
e
m
e
n
t
)

u
R
E
A
D
O
N
L
Y

(
p
r
e
v
e
n
t

c
h
a
n
g
e
s
)

u
A
C

C
E
S
S
K
E
Y
=

C
r
r
r
r
c
t
r

(
s
h
o
r
t
c
u
t
k
e
y
)

»
T
A
B
I
N
D
E
X

=
N
u
m
b
e
r

(
p
o
s
i
t
i
o
n
i
n

U
R
=
||s
||c
||r
||-
||-
||p
||t
||t

(
e
l
e
m
e
n
t

l
o
s
t

f
o
c
u
s
)

“
O
N
S
E
L
E
C
T
=
||s
||c
||r
||-
||-
||p
||t
||t

(
e
l
e
m
e
n
t

t
e
x
t

s
e
l
e
c
t
e
d
)

“
O
N
C
H
A
N
G
E
”
=

(
e
l
e
m
e
n
t

v
a
l
u
e
c
h
a
n
g
e
d
)

“
c
o
m
m
o
n
”
=
a
t
t
r

r i b u t e s

Cont
ents
Plai
n
text
(incl
udin
g
entiti
es)
Cont
aine
d in
Bloc
k-
level
elem
ents,
inlin
e
elem
ents
exce
pt
BUT
TON

The **TEXTAREA** element defines a form control for the user to enter *multi-line text input*. While **TEXTAREA** is most useful within a **FORM**, HTML 4.0 allows **TEXTAREA** in any block-level or inline element other than BUTTON. However, Netscape Navigator will not display any **TEXTAREA** elements outside of a **FORM**.

The initial value of the **TEXTAREA** is provided as the content of the element and must not contain any HTML tags. When a form is submitted, the current value of any **TEXTAREA** element within the **FORM** is sent to the server as a name/value pair. The **TEXTAREA** element's **NAME** attribute provides the name used.

The required **ROWS** and **COLS** attributes specify the number of visible rows and columns, respectively, in a visual browser. These attributes provide a *guide* for the user rather than a *restriction*; browsers allow an unlimited amount of text input in theory, though in practice many browsers limit the contents of a **TEXTAREA** to 32 or 64 kilobytes. Author restrictions on the amount of data entered should be enforced by the CGI script or Java servlet handling the form.

The boolean **READONLY** attribute, new in HTML 4.0 and poorly supported by current browsers, prevents the user from editing the content of the **TEXTAREA**. Read-only elements are still submitted with the form. The **DISABLED** attribute, also poorly supported, disables the **TEXTAREA**. Disabled elements are read-only elements with the added restrictions that the values are not submitted with the form, the elements cannot receive focus, and the elements are skipped when navigating the document by tabbing.

The **ACCESSKEY** attribute specifies a single Unicode character as a shortcut key for giving focus to the **TEXTAREA**. Authors can set the access key on the **TEXTAREA** element or the LABEL element associated with it. Entities (e.g.

´) may be used as the **ACCESSKEY** value.

The **TABINDEX** attribute specifies a number between 0 and 32767 to indicate the tabbing order of the element. A **TEXTAREA** with **TABINDEX=0** or no **TABINDEX** attribute will be visited after any elements with a positive **TABINDEX**. Among positive **TABINDEX** values, the lower number receives focus first. In the case of a tie, the element appearing first in the HTML document takes precedence.

The **TEXTAREA** element also takes a number of attributes to specify client-side scripting actions for various events. In addition to the [core events](#) common to most elements, **TEXTAREA** accepts the following event attributes:

- **ONFOCUS**, when the element receives focus;
- **ONBLUR**, when the element loses focus;
- **ONSELECT**, when text in the element is selected;
- **ONCHANGE**, when the element loses focus and its value has changed since it received focus.

More Information

- [TEXTAREA in W3C HTML 4.0 Recommendation](#)
- [TEXTAREA in W3C HTML 3.2 Recommendation](#)
- [TEXTAREA in HTML 2.0 Standard](#)
- [TEXTAREA in WDG HTML 3.2 Reference](#)
- [TEXTAREA in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

A - Anchor



A - Anchor

Syntax
<A>...
>
Attribute
Specifications

u

H

R

E

F

=

U

R

I

(

h

y

p

e

r

t

e

x

t

r

e

f

e

r

e

n

c

e

)

u

N

A

M

E
=
C
D
A
T
A

(
n
a
m
e
d

l
i
n
k

d
e
s
t
i
n
a
t
i
o
n
)

«
R
E
L
=
I
n
t
e
r
n
e
t
i
o
n
s
h
i
p

(
r
e
l
a
t
i
o
n
s
h
i
p

t
o
l
i
n
k
)

“ R E V
= I
i
P
K
T
Y
P
E
S

(r e l a t i o n s h i p
f r o m
l i n k
)

“ T Y P
= C
O
P
T
E
P
T

[T
Y
P
E](#)

(c o n t e n t - t y p e

o f l i n k)

“ T A R G E T =

[T
Y
P
E](#)

(f r a m e t o r e n d

e
r

l
i
n
k

i
n
)

u
H
R
E
F
L
A
N
G
=
[T](#)
[a](#)
[n](#)
[g](#)
[u](#)
[e](#)
[l](#)
[i](#)
[n](#)
[g](#)
[u](#)
[e](#)

(
l
a
n
g
u
a
g
e

o
f

l
i
n
k
)

u
C
H
A
R
S
E
T

https://t.me/r

(character

encoding

of

link)

ACCESSKEY

https://t.me/r

(sh

o
r
t
c
u
t

k
e
y
)

u
T
A
B
I
N
D
E
X
=
N
u
m
b
e
r

(
p
o
s
i
t
i
o
n

i
n

t
a
b
b
i
n
g

o
r
d
e
r
)

u
S
H
A
P

```
E
=
[
  r
  e
  c
  t
  |
  c
  i
  r
  c
  l
  e
  |
  p
  o
  l
  y
  |
  d
  e
  f
  a
  u
  l
  t
]
(
  c
  l
  i
  e
  n
  t
  -
  s
  i
  d
  e
  i
  m
  a
  g
  e
  m
  a
  p
)
"
```

C
O
O
R
D
S
=

(
c
l
i
e
n
t
-
s
i
d
e
i
m
a
g
e
m
a
p
)

=
O
N
F
O
C
U
S
=

(
e
l
e
m
e
n
t

r
e
c
e
i
v
e
d

f
o
c
u
s
)

“
O
N
B
L
U
R
=

s
c
r
r
t

(
e
l
e
m
e
n
t

l
o
s
t

f
o
c
u
s
)

“
c
o
m
m
o
n
=
a

t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
exce
pt A
Cont
aine
d in
Bloc
k-
level
elem
ents
inlin
e
elem
ents
exce
pt A

The **A** element denotes an *anchor*--a hypertext link or the destination of a link. The **HREF** attribute specifies a hypertext link to another resource, such as an HTML document or a JPEG image. Examples:

```
␣<A HREF="album.html">My photo album</A>  
␣<A HREF="../images/me.jpg">Picture of me</A>  
␣<A href="http://www.htmlhelp.com/sounds/auldlang.mid" TYPE="audio/midi"  
ACCESSKEY=A>Auld Lang Syne (5 kB MIDI)</A>  
␣<A HREF="section2.html" TARGET="content" TITLE="Elements of the HEAD"  
REL=next>Section 2</A>  
␣<A HREF="mailto:liam@htmlhelp.com" TITLE="Feedback on HTML 4.0  
Reference">liam@htmlhelp.com</A>  
␣<A HREF="http://www.htmlhelp.com/" TARGET="_top">Web Design Group</A>  
␣<A HREF="http://babel.alis.com:8080/langues/iso639.zh.htm" CHARSET="big5"  
HREFLANG=zh>ISO 639</A>
```

The value of the **HREF** attribute is the URI of the link. The **TYPE** attribute can be used to specify the Internet media type of the link, allowing browsers to avoid fetching a resource that they cannot handle.

The **TITLE** attribute can be used to briefly describe the contents of the link and is rendered as a "tooltip" by some visual browsers. With mailto links, some browsers use the **TITLE** attribute value as a subject for the e-mail message.

The content of an **A** element used as a link should be as context-free as possible. In other words, a user should be able to pull all **A** elements from a document and still have an idea what lies behind each link. Link text that contains **Click here** or simply **here** is extremely bad form.

The **TARGET** attribute is used with [frames](#) to specify in which frame the link should be rendered. If no frame with such a name exists, the link is rendered in a new window unless overridden by the user. Special frame names begin with an underscore:

- **_blank** renders the link in a new, unnamed window
- **_self** renders the link in the current frame (useful for overriding a **BASE TARGET**)
- **_parent** renders the link in the immediate **FRAMESET** parent
- **_top** renders the link in the full, unframed window

The optional **HREFLANG** and **CHARSET** attributes give the language and character encoding, respectively, of the link. The language should be specified according to [RFC 1766](#); examples include **en** for English, **en-US** for American English, and **ja** for Japanese. Examples of character encodings include **ISO-8859-1**, **SHIFT_JIS**, and **UTF-8**.

The **ACCESSKEY** attribute specifies a single Unicode character as a shortcut key for following the link. [Entities](#) (e.g. **´**;) may be used as the **ACCESSKEY** value.

The **TABINDEX** attribute specifies a number between 0 and 32767 to indicate the tabbing order of the element. An anchor with **TABINDEX=0** or no **TABINDEX** attribute will be visited after any elements with a positive **TABINDEX**. Among positive **TABINDEX** values, the lower number receives focus first. In the case of a tie, the element appearing first in the HTML document takes precedence.

The **REL** and **REV** attributes define relationships between an anchor and the linked resource. **REL** defines a link relationship from the current document to the linked document while **REV** defines a relationship in the opposite direction. For example,

```
<A HREF="foo.html" REL=glossary>...</A>
```

indicates that **foo.html** is a glossary for the current document while

```
<A HREF="bar.html" REV=subsection>...</A>
```

indicates that the current document is a subsection of **bar.html**. The value of the **REL** and **REV** attributes is a space-separated list of [link types](#).

The **NAME** attribute defines a destination for a link. For example, a document containing

```
<H1><A NAME=foo>My Heading</A></H1>
```

defines a link destination named "foo" at the indicated heading. One could then use **HREF="#foo"** in an **A** element within the same document or **HREF="somedoc.html#foo"** from within another document.

An **A** element cannot contain another **A** element, so one must be careful that named anchors do not contain link anchors. Authors can use both the **NAME** and **HREF** attributes in a single **A** element to avoid this problem.

HTML 4.0's **ID** attribute is intended to eliminate the need for **A NAME**. The **ID** attribute can be used with almost any element to define a link destination, so that the following could be used in place of the previous example:

```
<H1 ID=foo>My heading</H1>
```

However, browser support for **ID** link destinations is very poor, so **A NAME** will be needed for quite awhile.

NAME and **ID** values must be unique in any document, and different values must differ by more than just the case. Values must begin with a letter in the range A-Z or a-z, and may be followed by A-Z, a-z, 0-9, hyphens, underscores, colons, or periods. When linking to a named anchor, the name is treated as case sensitive.

The **SHAPE** and **COORDS** attributes of **A** can be used to create [client-side image maps](#) via the **OBJECT** element. The default **SHAPE** value is **rect**, which defines a rectangular region using **COORDS**="left, top, right, bottom". Other **SHAPE** values are

- **default**, which specifies the entire image;
- **circle**, which specifies a circular region using **COORDS**="center-x, center-y, radius";
- **poly**, which specifies a polygonal region using **COORDS**="x1, y1, x2, y2, . . . , xN, yN".

Coordinate values are relative to the top left corner of the object and may be expressed as pixels or percentages. A percentage radius value for circular regions is calculated relative to the smaller of the object's width and height. If two or more regions overlap, the earliest specified region takes precedence.

In addition to the [core events](#) common to most elements, **A** accepts the following event attributes for client-side scripting:

- **ONFOCUS**, when the link receives focus;
- **ONBLUR**, when the link loses focus.

More Information

- [A in W3C HTML 4.0 Recommendation](#)
- [A in W3C HTML 3.2 Recommendation](#)
- [A in HTML 2.0 Standard](#)
- [A in WDG HTML 3.2 Reference](#)
- [A in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

APPLET - Java Applet



APPLET - Java Applet

Syntax
<APPLET>...</APPLET>
Attribute Specifications

u

C
O
D
E
=
C
D
A
T
A

(
c
l
a
s
s

f
i
l
e
)

u

C
O
D
E
B
A
S
E
=
=

U
R
L

(
b
a
s
e

U
R
L

f
o
r

c
l
a
s
s

f
i
l
e
s
)

u

W
I
D
T
H
=

I
e
n
g
t
h

(
a
p
p
l
e
t

w
i
d
t
h
)

u

H

EIGHT = **EEHGH**

(**a p p l e t** **h e i g h t**)

u

ARCHIVE =

CDATA

(**a r c h i v e** **f i l e s**)

u

O

B
J
E
C
T
=
C
D
A
T
A

(
s
e
r
i
a
l
i
z
e
d

a
p
p
l
e
t
)

“
N
A
M
E
=
C
D
A
T
A

(
n
a
m
e

f
o
r

i
n
t
e
r
-
a
p

p
l
e
t

c
o
m
m
u
n
i
c
a
t
i
o
n
)

“
A
L
T
=
T
e
x
t

(
a
l
t
e
r
n
a
t
e

t
e
x
t
)

“
A
L
I
G
N
=
[

t
o
p

|

X
E
I
S

(h o r i z o n t a l g u t t e r)

“ V S P A C E =

P
I
X
E
I
S

(v e r t i c a l g u t t e r)

“ C
O
M
O

n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
PAR
AM
elem
ents
follo
wed
by
bloc
k-
level
elem
ents
and/
or
inlin
e
elem
ents
Cont
aine
d in
Inlin
e
elem
ents,
bloc
k-
level
elem
ents
exce
pt
PRE

The **APPLET** element is used to embed *Java applets*. It has been [deprecated](#) in HTML 4.0 in favor of the more generalized [OBJECT](#) element. However, since the few browsers that support **OBJECT** do so with significant bugs, **APPLET** is currently a more reliable method of embedding Java applets.

APPLET's **CODE** attribute specifies the name of the class file that contains the compiled Applet subclass. The value is relative to the URI specified in the **CODEBASE** attribute, or to the HTML document's base URI if the **CODEBASE** attribute is not given.

The required **WIDTH** and **HEIGHT** attributes define the dimensions of the applet. The value may be given in pixels or

as a percentage of the parent element's width or height.

The **ALT** attribute can be used to give alternate text for browsers that recognize the **APPLET** element but do not support Java or do not have Java enabled. Authors can also give alternate content between the start and end tags of the **APPLET** element--a better method than using the **ALT** attribute since it allows authors to include HTML markup in the alternate content and also works with pre-[HTML 3.2](#) browsers that do not support **APPLET**.

An **APPLET** may contain **PARAM** elements to define applet-specific parameters. **PARAM** elements should be specified before any other content of the **APPLET** element. In the following example, a decorative Java applet takes two parameters. The **APPLET** contains an animated GIF as an alternative for non-Java browsers.

```
<APPLET CODE="Animate.class" WIDTH=100 HEIGHT=100>
<PARAM NAME=img1 VALUE="/images/1.jpg">
<PARAM NAME=img2 VALUE="/images/2.jpg">
<IMG SRC="animation.gif" ALT="" WIDTH=100 HEIGHT=100>
</APPLET>
```

The **ARCHIVE** attribute can specify a comma-separated list of archived files (either absolute URIs or URIs relative to the **CODEBASE**), allowing the browser to download many files with a single connection and hence decreasing the total download time. The standard archive format for Java files is [JAR](#). JAR files can be created with the **jar** tool included with the [Java Development Kit](#) 1.1 and up.

Note that some browsers do not support the **ARCHIVE** attribute, so all necessary files should be available unarchived as well. Other browsers only support a single URI as the **ARCHIVE** value.

The **OBJECT** attribute specifies a serialized (saved) representation of an applet. The **CODE** attribute should not be used if and only if the **OBJECT** attribute is specified. When the applet is deserialized, its **init()** method is not invoked, but its **start()** method is. Sun recommends restraint in using this poorly supported feature.

The **ALIGN** attribute specifies the alignment of the applet. The values **top**, **middle**, and **bottom** specify the applet's position with respect to surrounding content on its left and right. **ALIGN=middle** aligns the center of the applet with the current baseline. To center the applet horizontally on the page, place the applet in a centered block, *e.g.*,

```
<P ALIGN=center><APPLET CODE="Game.class" WIDTH=300 HEIGHT=100></APPLET></P>
```

The other **ALIGN** values, **left** and **right**, specify a *floating* applet; the applet is placed at the left or right margin and content flows around it. To place content below the applet, use **<BR CLEAR=left|right|all>** as appropriate.

The [vertical-align](#) and [float](#) properties of [Cascading Style Sheets](#) provide more flexible methods of aligning applets.

The **HSPACE** and **VSPACE** attributes allow an author to suggest horizontal gutters and vertical gutters, respectively, around the applet. The value must be in pixels and applies to both sides of the applet. [Style sheets](#) provide more flexibility in specifying the space around applets.

More Information

- [APPLET in W3C HTML 4.0 Recommendation](#)
- [APPLET in W3C HTML 3.2 Recommendation](#)

- » [APPLET in WDG HTML 3.2 Reference](#)
- » [APPLET in *Learning HTML 3.2 by Examples*](#)
- » [The APPLETTAG \(JDK 1.1 Documentation\)](#)
- » [The Java Tutorial](#)
 - » [Using the <APPLET> Tag](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

BASEFONT - Base Font Change



BASEFONT - Base Font Change

Syntax
<BASEFONT>
Attribute
Specifications

u

**S
I
Z
E
=
C
D
A
T
A**

(
f
o
n
t

s
i
z
e

a
d
j
u
s
t
m
e
n
t
)

u

C
O
L
O
R
=
C
O
I
O
r

(
f
o
n
t

c
o
l
o
r

a
d
j
u
s
t
m
e
n
t
)

“
F
A
C
E
=
C
D
A
T
A

(
f
o
n
t

f
a
c
e

a
d
j

u
s
t
m
e
n
t
)
u
I
D
=
I
D
(
u
n
i
q
u
e
I
D
)
Cont
ents
Emp
ty
Cont
aine
d in
Inlin
e
elem
ents,
bloc
k-
level
elem
ents
exce
pt
PRE

The **BASEFONT** element, [deprecated](#) in HTML 4.0 in favor of [style sheets](#), allows authors to suggest rudimentary *font changes*. Use of the **BASEFONT** element brings the same usability and accessibility problems as **FONT**, as discussed in the article [What's Wrong With FONT?](#)

Unlike **FONT**, **BASEFONT**'s changes affect the base font, and so apply to all content following the **BASEFONT** element except for headings. However, most browsers fail to apply changes in the base font size and color to **TABLE**s.

BASEFONT's required **SIZE** attribute specifies the font size to use on a browser-dependent scale of 1 to 7, with the default being 3.

The poorly supported **COLOR** and **FACE** attributes suggest a font color and face, respectively. [Style sheets](#) are better supported and more flexible than **BASEFONT**'s **COLOR** and **FACE** attributes.

More Information

- [BASEFONT in W3C HTML 4.0 Recommendation](#)
- [BASEFONT in W3C HTML 3.2 Recommendation](#)
- [BASEFONT in WDG HTML 3.2 Reference](#)
- [BASEFONT in *Learning HTML 3.2 by Examples*](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

FONT - Font Change



FONT - Font Change

Syntax
..

Attribute
Specifications

u

S

I

Z

E

=

C

D

A

T

A

(

f

o

n

t

s

i

z

e

a

d

j

u

s

t

m

e

n

t

)

u

C
O
L
O
R
=
C
O
I
O
R

(
f
o
n
t

c
o
l
o
r

a
d
j
u
s
t
m
e
n
t
)

“
F
A
C
E
=
C
D
A
T
A

(
f
o
n
t

f
a
c
e

a
d
j

u
s
t
m
e
n
t
)

u
c
o
r
e
=
a
t
t
r
i
b
u
t
e
s

u
i
n
t
e
r
n
a
t
i
o
n
a
l
i
n
i
t
i
a
t
i
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin

e
elem
ents
Cont
aine
d in
Inlin
e
elem
ents
bloc
k-
level
elem
ents
exce
pt
PRE

The **FONT** element, [deprecated](#) in HTML 4.0 in favor of [style sheets](#), allows authors to suggest rudimentary *font changes*. Use of the **FONT** element brings numerous usability and accessibility problems, as discussed in the article [What's Wrong With FONT?](#)

The least harmful use of the **FONT** element is in suggesting relative changes in font size with `` or ``. These tags increment or decrement the font size relative to the size specified in the **BASEFONT** element, or relative to a base size of **3** if no **BASEFONT** element is used. Sizes are based on a seven-point scale (1..7) that is browser dependent.

While authors can specify **SIZE** values such as **-2** and **+3**, as well as absolute values such as **1**, these kinds of changes are strongly discouraged due to the sensitivity some users have to different font sizes. While a value like **-2** may look right to you with your eyesight and user settings, it could easily be unreadable to a user with different eyesight and user settings.

The **COLOR** attribute suggests a text color. While most browsers allow users to override author color changes, the widely used Netscape Navigator 2.x, 3.x, and 4.x do not override colors specified with **FONT**. This makes the **COLOR** attribute very dangerous from an accessibility point of view.

Authors often use the **COLOR** attribute as a form of emphasis or to indicate a heading. In these cases, use of structural HTML (e.g., **STRONG**, **H1**) along with a style sheet provides a more flexible, accessible document.

The **FACE** attribute gives a comma-separated list of font faces in which to display text. The fonts are listed in order of preference, so that if the browser does not have the first font listed, it will try the second, then the third, and so on.

The **FONT** element is an [inline element](#), meaning that it cannot contain [block-level elements](#) such as **P** and **TABLE**. Again, [style sheets](#) provide much more flexibility in suggesting font styles.

More Information

- [FONT in W3C HTML 4.0 Recommendation](#)
- [FONT in W3C HTML 3.2 Recommendation](#)
- [FONT in WDG HTML 3.2 Reference](#)
- [FONT in Learning HTML 3.2 by Examples](#)

- » [What's Wrong With FONT?](#)
- » [considered harmful](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

AREA - Image Map Region



AREA - Image Map Region

Syntax
<AREA>
Attribute
Specifications

u
S
H
A
P
E
=
[
r
e
c
t
|
c
i
r
c
l
e
|
p
o
l
y
|
d
e
f
a

u
l
t
l
(
s
h
a
p
e
o
f
r
e
g
i
o
n
)
“
C
O
O
R
D
S
=
C
o
o
o
o
o
o
o
o
o
(
c
o
o
r
d
i
n
a
t
e
s
o
f
r
e
g
i
o
n
)

»
H
R
E
F
=
[G
R
I](#)

(
l
i
n
k
e
d

r
e
s
o
u
r
c
e
)

»
T
A
R
G
E
T
=
[F
r
a
m
e
t
a
r
i
t
e
t](#)

(
f
r
a
m
e

t
o
r
e
n

d
e
r

l
i
n
k

)
»
N
O
H
R
E
F

(
i
n
a
c
t
i
v
e

r
e
g
i
o
n
)
»
A
L
T
=
T
e
x
t

(
a
l
t
e
r
n
a
t
e

t
e

x
t
)

“
T
A
B
I
N
D
E
X
=
N
U
B
E
R

(
p
o
s
i
t
i
o
n

i
n

t
a
b
b
i
n
g

o
r
d
e
r
)

“
O
N
F
O
C
U
S
=
S
C
H
E
T

||t|p

(r e g i o n

r e c e i v e d

f o c u s)

“ O N B L U R =

||s||s||r||p||t||p

(r e g i o n

l o s t

f o c u s)

“

C
O
M
O
N
=
a
t
t
r
i
b
u
t
e
s
Cont
ents
Emp
ty
Cont
aine
d in
MAP

The **AREA** element defines a *map region* in a [client-side image map](#). Each map region is a piece of the image with a different action when clicked.

The **SHAPE** and **COORDS** attributes of **AREA** specify what part of the image is included in the region. The default **SHAPE** value is **rect**, which defines a rectangular region using **COORDS="left, top, right, bottom"**. Other **SHAPE** values are

- **default**, which specifies the entire image;
- **circle**, which specifies a circular region using **COORDS="center-x, center-y, radius"**;
- **poly**, which specifies a polygonal region using **COORDS="x1, y1, x2, y2, . . . , xN, yN"**.

Coordinate values are relative to the top left corner of the object and may be expressed as pixels or percentages. A percentage radius value for circular regions is calculated relative to the smaller of the object's width and height. If two or more regions overlap, the earliest specified region takes precedence.

The **HREF** attribute specifies a link to another resource, such as an HTML document or a JPEG image. The **TITLE** attribute can be used to briefly describe the contents of the link and is rendered as a "tooltip" by some visual browsers. The boolean **NOHREF** attribute indicates that the region has no link.

The required **ALT** attribute provides alternate text for those not loading images. Effective **ALT** text should generally give the *function* of the map region rather than a *description* of the region. For example, **ALT="WDG Home"** or **ALT="Web Design Group"** would be more appropriate than **ALT="WDG Logo"** for a map region whose link went to the [WDG home page](#). Good **ALT** text is crucial to the document's accessibility for the significant portion of users who do not load images.

The **TARGET** attribute is used with [frames](#) to specify in which frame the link should be rendered. If no frame with such a name exists, the link is rendered in a new window unless overridden by the user. Special frame names begin with an underscore:

- **_blank** renders the link in a new, unnamed window
- **_self** renders the link in the current frame (useful for overriding a **BASE TARGET**)
- **_parent** renders the link in the immediate **FRAMESET** parent
- **_top** renders the link in the full, unframed window

The **ACCESSKEY** attribute specifies a single Unicode character as a shortcut key for following the link. [Entities](#) (e.g. **´**) may be used as the **ACCESSKEY** value.

The **TABINDEX** attribute specifies a number between 0 and 32767 to indicate the tabbing order of the map region. A region with **TABINDEX=0** or no **TABINDEX** attribute will be visited after any elements with a positive **TABINDEX**. Among positive **TABINDEX** values, the lower number receives focus first. In the case of a tie, the element appearing first in the HTML document takes precedence.

In addition to the [core events](#) common to most elements, **AREA** accepts the following event attributes for client-side scripting:

- **ONFOCUS**, when the region receives focus;
- **ONBLUR**, when the region loses focus.

More Information

- [AREA in W3C HTML 4.0 Recommendation](#)
- [AREA in W3C HTML 3.2 Recommendation](#)
- [AREA in WDG HTML 3.2 Reference](#)
- [AREA in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

PARAM - Object Parameter



PARAM - Object Parameter

Syntax
<PARAM
>
Attribute
Specification
s

u

**N
A
M
E**
=
C
D
A
T
A

(
p
r
o
p
e
r
t
y

n
a
m
e
)

u

V
A
L
U
E
=
C

D
A
T
A

(p
r
o
p
e
r
t
y

v
a
l
u
e
)

"
V
A
L
U
E
T
Y
P
E
=
[

d
a
t
a

|
r
e
f

|
o
b
j
e
c
t

]

(
t
y
p
e

o
f
v
a
l
u
e
)

u
T
Y
P
E
=
C
I
P
I
T
I
L
I
Z
E
D
R
E
S
O
U
R
C
E
S

(
c
o
n
t
e
n
t
-
t
y
p
e

o
f
v
a
l
u
e
r
e
s
o
u
r
c
e
)

u

I
D
=
I
D

Cont
ents
Emp
ty
Cont
aine
d in
APP
LET
OBJ
ECT

The **PARAM** element provides parameters for the **OBJECT** and **APPLET** elements. An **OBJECT** or **APPLET** may contain any number of **PARAM** elements prior to the alternate content that is also contained within the **OBJECT** or **APPLET** element.

The required **NAME** attribute of **PARAM** gives the name of the parameter while the **VALUE** attribute gives the parameter's value. The parameters recognized are specific to the kind of object being embedded or to the plug-in that renders the embedded object. For example, a clock applet may accept parameters to specify the style of the clock and the colors to use:

```
<OBJECT CLASSID="java:Clock.class" CODETYPE="application/java" WIDTH=100 HEIGHT=100  
TITLE="A real live clock!" STANDBY="Do you know what time it is?">  
<PARAM NAME=TYPE VALUE=ANALOG>  
<PARAM NAME=BGCOLOR VALUE=WHITE>  
<PARAM NAME=FGCOLOR VALUE=NAVY>  
</OBJECT>
```

Note that the **PARAM** elements used in the preceding example would not change if the **APPLET** element were used in place of **OBJECT**.

Objects such as videos, audio clips, and VRML worlds are typically handled by browser plug-ins. Each plug-in recognizes certain parameters, which can make choosing parameters difficult when the author does not know which plug-in the user has. However, unsupported parameters should be safely ignored. The following example uses parameters specific to the [QuickTime](#) movie plug-in as well as parameters specific to the [LiveAudio](#) audio plug-in:

```
<OBJECT DATA="mlk.mov" TYPE="video/quicktime" TITLE="Martin Luther King's "I Have  
a Dream"; speech" WIDTH=150 HEIGHT=150>  
<PARAM NAME=pluginspage VALUE="http://quicktime.apple.com/">  
<PARAM NAME=autoplay VALUE=true>  
<OBJECT DATA="mlk.wav" TYPE="audio/x-wav" TITLE="Martin Luther King's "I Have a  
Dream"; speech">  
<PARAM NAME=autostart VALUE=true>  
<PARAM NAME=hidden VALUE=true>  
<A HREF="mlk.html">Full text of Martin Luther King's "I Have a Dream" speech</A>  
</OBJECT>  
</OBJECT>
```

Note that only the parameters specified within the **OBJECT** and prior to the alternate content are passed onto the plug-in, so that the **pluginspage** and **autoplay** parameters are used if the QuickTime movie is played while the

autostart and **hidden** parameters are used if the audio clip is played.

The **VALUETYPE** parameter of **PARAM** designates the type of the **VALUE** attribute. The default value for **VALUETYPE** is **data**, which indicates that the **VALUE** attribute contains a string.

The **ref** value for **VALUETYPE** indicates that the **VALUE** attribute contains a URI where run-time values are stored. The Internet media type of the resource is specified by the **TYPE** attribute. The following example uses values with **VALUETYPE=ref** to specify the location of images to animate and a sound to play during the animation:

```
<OBJECT CLASSID="java:Animator.class" CODETYPE="application/java" WIDTH=200 HEIGHT=300
TITLE="Wedding Photos">
<PARAM NAME=IMAGE1 VALUE="images/wedding/bride.jpg" VALUETYPE=ref TYPE="image/jpeg">
<PARAM NAME=IMAGE2 VALUE="images/wedding/groom.jpg" VALUETYPE=ref TYPE="image/jpeg">
<PARAM NAME=IMAGE3 VALUE="images/wedding/cake.jpg" VALUETYPE=ref TYPE="image/jpeg">
<PARAM NAME=SOUND VALUE="http://www.htmlhelp.com/sounds/weddingmarch.au" VALUETYPE=ref
TYPE="audio/basic">
</OBJECT>
```

The URI specified by **VALUE** is passed to the object without being resolved to a full URL. In the preceding example, the Java class would be responsible for resolving and fetching the contents of partial URIs like "images/wedding/cake.jpg".

VALUETYPE also takes an **object** value, for use when a **VALUE** attribute specifies an identifier of a separate **OBJECT** in the document. The following example features a hypothetical application for interactively walking through the frames of an animated GIF. The GIF is loaded in a separate **OBJECT** from the application that uses it.

```
<OBJECT DECLARE ID=mygif DATA="animation.gif" TYPE="image/gif">
</OBJECT>
<OBJECT CLASSID="framepicker">
<PARAM NAME=image VALUE="#mygif" VALUETYPE=object>
<IMG SRC="animation.gif" ALT="[Example of an animated GIF]">
</OBJECT>
```

More Information

- [PARAM in W3C HTML 4.0 Recommendation](#)
- [PARAM in W3C HTML 3.2 Recommendation](#)
- [PARAM in WDG HTML 3.2 Reference](#)
- [PARAM in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~



This help file is created by [Gerben Hoekstra](mailto:Gerben.Hoekstra@weballey.com) <weballey@usa.net>.
Visit my web site [WEBalley](http://www.weballey.com) - web publishing for beginners.
This file is freeware and may be distributed freely.

SUB - Subscript



SUB - Subscript

Syntax
<SUB>...
</SUB>
Attribute
Specifications

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

elem ents

The **SUB** element is used for *subscripts*. Since **SUB** is inherently presentational, it should not be relied upon to express a given meaning. However, it can be useful for chemical formulas and mathematical indices, where the subscript presentation is helpful but not required. For example:

```
«Chemical formulas include H<SUB>2</SUB>O (water) and  
C<SUB>21</SUB>H<SUB>27</sub>NO (methadone).  
«Let <VAR>x</VAR> = <VAR>x<SUB>1</SUB></VAR> + <VAR>x<SUB>2</SUB></VAR> + ... +  
<VAR>x<SUB>n</SUB></VAR>.
```

More Information

- « [SUB in W3C HTML 4.0 Recommendation](#)
- « [SUB in W3C HTML 3.2 Recommendation](#)
- « [SUB in WDG HTML 3.2 Reference](#)
- « [SUB in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

SUP - Superscript



SUP - Superscript

Syntax
<SUP>...
</SUP>
Attribute
Specifications

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

elem ents

The **SUP** element is used for *superscripts*. Since **SUP** is inherently presentational, it should not be relied upon to express a given meaning. However, it can be useful for mathematical exponents where the context implies the meaning of the exponent, as well as other cases where superscript presentation is helpful but not required. For example:

„The rent is due on the 1st of each month.
„An example of a quadratic polynomial is 3<VAR>x</VAR><SUP
CLASS=exponent>2</SUP> + 5<VAR>x</VAR> - 7.

The following two examples are ambiguous when presented on a browser incapable of superscript text:

„2⁴ = 16
„sin²<VAR>x</VAR> + cos<SUP
CLASS=exponent>2</SUP><VAR>x</VAR> = 1

There is no simple solution for this problem. One could use notation such as **2^4** to represent "two raised to the exponent four." If it gains browser support, [MathML](#) should provide a more suitable solution in the future.

More Information

- „[SUP in W3C HTML 4.0 Recommendation](#)
- „[SUP in W3C HTML 3.2 Recommendation](#)
- „[SUP in WDG HTML 3.2 Reference](#)
- „[SUP in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

CITE - Citation



CITE - Citation

Syntax
<CITE>...
</CITE>
Attribute
Specifications

[Common attributes](#)

Contents
[Inline elements](#)
Contained in
[Inline elements, block-level](#)

elem ents

The **CITE** element is used to markup *citations*, such as titles of magazines or newspapers, ship names, references to other sources, and quotation attributions. Visual browsers typically render **CITE** as italic text, but authors can suggest a rendering using [style sheets](#). Since **CITE** is a structural element, it carries *meaning*, making it preferable to font style elements such as **I** when marking up citations.

Example:

```
<CITE>The Toronto Star</CITE> gave its review of the movie <CITE>Titanic</CITE>
yesterday.
```

More Information

- [CITE in W3C HTML 4.0 Recommendation](#)
- [CITE in W3C HTML 3.2 Recommendation](#)
- [CITE in HTML 2.0 Standard](#)
- [CITE in WDG HTML 3.2 Reference](#)
- [CITE in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

CODE - Computer Code



CODE - Computer Code

Syntax
<CODE>
..</CODE
>
Attri
bute
Spe
cific
ation
s

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
Inlin
e
elem
ents
bloc
k-

[level](#)
[elem](#)
[ents](#)

The **CODE** element denotes *computer code*. Visual browsers typically render **CODE** as monospaced text, but authors can suggest a rendering using [style sheets](#). Since **CODE** is a structural element, it carries *meaning*, making it preferable to font style elements such as **TT** when marking up computer code.

Since spacing is often important when presenting computer code, the **PRE** element can be useful as a container for **CODE** elements. When used within other containers, a **CODE** element has multiple spaces collapsed. The following example uses **CODE** within **PRE**:

```
<PRE><CODE>
class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
</CODE></PRE>
```

More Information

- [CODE in W3C HTML 4.0 Recommendation](#)
- [CODE in W3C HTML 3.2 Recommendation](#)
- [CODE in HTML 2.0 Standard](#)
- [CODE in WDG HTML 3.2 Reference](#)
- [CODE in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

DFN - Defined Term



DFN - Defined Term

Syntax
<DFN>...
</DFN>
Attribute
Specifications

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

elem ents

The **DFN** element denotes the *defining instance of a term*. Visual browsers typically render **DFN** as italic text, but authors can suggest a rendering using [style sheets](#). Since **DFN** is a structural element, it carries *meaning*, making it preferable to font style elements such as **!** when marking up the defining instance of a term.

[Netscape Navigator](#) does not support the **DFN** element. In most cases, the lack of support is not a significant problem; concerned authors could nest the **DFN** element within an **!** element or another [font style](#) element.

More Information

- » [DFN in W3C HTML 4.0 Recommendation](#)
- » [DFN in W3C HTML 3.2 Recommendation](#)
- » [DFN in WDG HTML 3.2 Reference](#)
- » [DFN in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

KBD - Text to be Input



KBD - Text to be Input

Syntax
<KBD>...
</KBD>
Attribute
Specifications

Common attributes

Contents
[Inline elements](#)
Contained in
[Inline elements, block-level](#)

[elem](#)
[ents](#)

The **KBD** element denotes *text to be entered by the user*. Visual browsers typically render **KBD** as monospaced text, but authors can suggest a rendering using [style sheets](#). Since **KBD** is a structural element, it carries *meaning*, making it preferable to font style elements such as **TT** when marking up text to be entered by the user.

Example:

```
<P>If the question does not apply to you, enter <KBD>N/A</KBD>.
```

More Information

- [KBD in W3C HTML 4.0 Recommendation](#)
- [KBD in W3C HTML 3.2 Recommendation](#)
- [KBD in HTML 2.0 Standard](#)
- [KBD in WDG HTML 3.2 Reference](#)
- [KBD in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

SAMP - Sample Output



SAMP - Sample Output

Syntax
<SAMP>
..</SAMP
>
Attri
bute
Spe
cific
ation
s

c
o
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
Inlin
e
elem
ents
bloc
k-

[level](#)
[elem](#)
[ents](#)

The **SAMP** element denotes *sample output*, such as from a program or script. Visual browsers typically render **SAMP** as monospaced text, but authors can suggest a rendering using [style sheets](#). Since **SAMP** is a structural element, it carries *meaning*, making it preferable to font style elements such as **TT** when marking up sample output.

Example:

```
<P>When an undefined element is used in an HTML document, a validator will give an error like the following:</P>  
<P><SAMP>C:\SP\BIN\NSGMLSU.EXE:test.html:4:7:E: element "FOOBAR" undefined</SAMP></P>
```

More Information

- [SAMP in W3C HTML 4.0 Recommendation](#)
- [SAMP in W3C HTML 3.2 Recommendation](#)
- [SAMP in HTML 2.0 Standard](#)
- [SAMP in WDG HTML 3.2 Reference](#)
- [SAMP in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

STRONG - Strong Emphasis



STRONG - Strong Emphasis

Syntax
...
Attributes

c
o
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Content
in
line
elements
Content
in
line
elements,
block-

[level](#)
[elem](#)
[ents](#)

The **STRONG** element gives *strong emphasis* to its contents. Visual browsers typically render **STRONG** as bold text, but authors can suggest a rendering using [style sheets](#). Since **STRONG** is a structural element, it carries *meaning*, making it preferable to font style elements such as **B** when strong emphasis is the intended meaning.

For weaker emphasis, use the **EM** element.

More Information

- [STRONG in W3C HTML 4.0 Recommendation](#)
- [STRONG in W3C HTML 3.2 Recommendation](#)
- [STRONG in HTML 2.0 Standard](#)
- [STRONG in WDG HTML 3.2 Reference](#)
- [STRONG in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

VAR - Variable



VAR - Variable

Syntax
<VAR>...
</VAR>
Attribute
Specifications

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

elem ents

The **VAR** element is used to markup *variables or program arguments*. Visual browsers typically render **VAR** as italic text, but authors can suggest a rendering using [style sheets](#). Since **VAR** is a structural element, it carries *meaning*, making it preferable to font style elements such as **!** when marking up variables.

Example:

```
<P>Versions of HTML are typically numbered in an <VAR>x</VAR>.<VAR>x</VAR> format.</P>
```

More Information

- [VAR in W3C HTML 4.0 Recommendation](#)
- [VAR in W3C HTML 3.2 Recommendation](#)
- [VAR in HTML 2.0 Standard](#)
- [VAR in WDG HTML 3.2 Reference](#)
- [VAR in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

B - Bold Text



B - Bold Text

Syntax
...
>
Attribute
Specifications

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

elem ents

The **B** element suggests that text be rendered as *bold text*. In most cases, use of a [phrase element](#) such as **STRONG** is more appropriate since such elements express the *meaning* of the text more clearly.

The **B** element is a suitable choice for marking a structure for which no [phrase element](#) exists. For example, vectors in mathematics have no structural element in HTML 4.0, but bold text is often an appropriate visual rendering:

If two vectors `<B CLASS=vector>v` and `<B CLASS=vector>w` are orthogonal, we write `<B CLASS=vector>v ⊥ <B CLASS=vector>w`.

Note the use of the [CLASS](#) attribute to add structural significance to the **B** elements. This allows greater flexibility when applying [style sheets](#) to different kinds of bold text.

More Information

- » [B in W3C HTML 4.0 Recommendation](#)
- » [B in W3C HTML 3.2 Recommendation](#)
- » [B in HTML 2.0 Standard](#)
- » [B in WDG HTML 3.2 Reference](#)
- » [B in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

BIG - Large Font



BIG - Large Font

Syntax
<BIG>...<
/BIG>
Attribute
Specifications

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

elem
ents
exce
pt
PRE

The **BIG** element suggests that text be rendered in a *larger font*. In most cases, use of a structural element such as **STRONG** or a heading (e.g., **H3**) is more appropriate since these elements express the *meaning* of the text more clearly. One can suggest that **STRONG** text be rendered in a larger font with the following [Cascading Style Sheet](#):

```
STRONG { font-size: larger }
```

Most browsers support nested BIG elements, but authors should be wary of making significant changes to the font size. Different users have different font sizes, eyesight, and window sizes. Large changes in font size may look right to the author but ridiculous to some users.

More Information

- [BIG in W3C HTML 4.0 Recommendation](#)
- [BIG in W3C HTML 3.2 Recommendation](#)
- [BIG in WDG HTML 3.2 Reference](#)
- [BIG in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

I - Italic Text



I - Italic Text

Syntax
<I>...</I>
Attribute
Specifications

Common
Attributes

Contents
[Inline
elements](#)
Contained in
[Inline
elements](#),
[block-
level
elem](#)

[ents](#)

The **I** element suggests that text be rendered as *italic text*. In most cases, use of a [phrase element](#) such as **EM**, **DFN**, **VAR**, or **CITE** is more appropriate since these elements express the *meaning* of the text more clearly.

The **I** element is a suitable choice for marking a structure for which no [phrase element](#) exists. For example, foreign phrases and taxonomic names have no structural element in HTML 4.0, but italic text is often an appropriate visual rendering:

```
<H1><I LANG=fr>Chacun son goût !</I></H1>
<P>Some people prefer dogs--<I CLASS=species>Canis familiaris</I>--while others are
eternal lovers of cats--<I CLASS=species>Felis cattus</I>.
```

Note the use of the [CLASS](#) and [LANG](#) attributes to add structural significance to the **I** elements. This allows greater flexibility when applying [style sheets](#) to different kinds of italic text.

More Information

- ⌋ [in W3C HTML 4.0 Recommendation](#)
- ⌋ [in W3C HTML 3.2 Recommendation](#)
- ⌋ [in HTML 2.0 Standard](#)
- ⌋ [in WDG HTML 3.2 Reference](#)
- ⌋ [in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#) [HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>. Visit my web site [WEBalley](#) - web publishing for beginners. This file is freeware and may be distributed freely.

SMALL - Small Font



SMALL - Small Font

Syntax
<SMALL
>...</SM
ALL>
Attri
bute
Spe
cific
ation
s

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
Inlin
e
elem
ents
bloc
k-

[level](#)
[elem](#)
[ents](#)
[exce](#)
[pt](#)
[PRE](#)

The **SMALL** element suggests that text be rendered in a *smaller font*. Since HTML 4.0 has no element to indicate de-emphasis, **SMALL** is often useful for this purpose. For example:

```
<P><SMALL>Copyright © 1998 Liam Quinn. All rights reserved.</SMALL></P>
```

Most browsers support nested **SMALL** elements, but this practice should be avoided. Since different users have different font sizes and eyesight, significant changes in font size can leave text too small to read even though it may look fine to the author.

Using [style sheets](#) in place of **SMALL** provides greater flexibility in changing the presentation. The previous example could also be marked up as follows:

```
<P CLASS=copyright>Copyright © 1998 Liam Quinn. All rights reserved.</P>
```

This could then be [linked](#) to the following style sheet:

```
.copyright {  
  font-size: smaller;  
  text-align: center  
}  
  
@media aural {  
  .copyright { volume: soft }  
}
```

More Information

- [SMALL in W3C HTML 4.0 Recommendation](#)
- [SMALL in W3C HTML 3.2 Recommendation](#)
- [SMALL in WDG HTML 3.2 Reference](#)
- [SMALL in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

STRIKE - Strike-through Text



STRIKE - Strike-through Text

Syntax
<STRIKE
>...</STR
IKE>
Attri
bute
Spe
cific
ation
s

c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Cont
ents
Inlin
e
elem
ents
Cont
aine
d in
Inlin
e
elem
ents
bloc
k-

[level](#)
[elem](#)
[ents](#)

The **STRIKE** element, [deprecated](#) in HTML 4.0, suggests that text be rendered with a *strike-through style*. In many cases, use of a [phrase element](#) such as **DEL** is more appropriate since such elements express the *meaning* of the text more clearly. However, since support for **DEL** among browsers is weak, **STRIKE** is useful in combination with **DEL**, as in the following example:

```
The latest version of HTML recommended by the W3C is HTML <DEL DATETIME="1997-12-19T00:00:00-05:00"><STRIKE>3.2</STRIKE></DEL> <INS DATETIME="1997-12-19T00:00:00-05:00">4.0</INS>.
```

If **DEL** is not a suitable structure, [style sheets](#) should be used to complement or replace instances of **STRIKE**. [CSS1](#) provides the [text-decoration](#) property for strike-through text.

More Information

- [STRIKE in W3C HTML 4.0 Recommendation](#)
- [STRIKE in W3C HTML 3.2 Recommendation](#)
- [STRIKE in WDG HTML 3.2 Reference](#)
- [STRIKE in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~ [Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

TT - Teletype Text



TT - Teletype Text

Syntax
<TT>...</
TT>
Attribute
Specifications

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

[elem](#)
[ents](#)

The **TT** element suggests that text be rendered as *teletype* or *monospaced text*. In most cases, use of a [phrase element](#) such as **CODE**, **SAMP**, or **KBD** is more appropriate since these elements express the *meaning* of the text more clearly.

[Style sheets](#) should be used to complement or replace instances of **TT**. For example, replacing

```
<H1><TT>My heading</TT></H1>
```

with

```
<H1>My heading</H1>
```

and

```
H1 { font-family: monospace }
```

in a style sheet allows the author to change the presentation of all [H1](#) elements in an entire site by changing just one line in the style sheet.

More Information

- [TT in W3C HTML 4.0 Recommendation](#)
- [TT in W3C HTML 3.2 Recommendation](#)
- [TT in HTML 2.0 Standard](#)
- [TT in WDG HTML 3.2 Reference](#)
- [TT in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra <weballey@usa.net>](mailto:weballey@usa.net).
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

U - Underlined Text



U - Underlined Text

Syntax
<U>...</U>
>
Attribute
Specifications

u
c
o
m
m
o
n
=
a
t
t
r
i
b
u
t
e
s

Contents
Inlin
e
elem
ents
Contained in
Inlin
e
elem
ents,
bloc
k-
level

elem ents

The **U** element, [deprecated](#) in HTML 4.0, suggests that text be rendered as *underlined text*. In most cases, use of a [phrase element](#) such as **CITE** or **STRONG** is more appropriate since such elements express the *meaning* of the text more clearly.

[Style sheets](#) should be used to complement or replace instances of **U**. For example, replacing

```
<H1><U>My heading</U></H1>
```

with

```
<H1>My heading</H1>
```

and

```
H1 { text-decoration: underline }
```

in a style sheet allows the author to change the presentation of all [H1](#) elements in an entire site by changing just one line in the style sheet.

More Information

- [U in W3C HTML 4.0 Recommendation](#)
- [U in W3C HTML 3.2 Recommendation](#)
- [U in WDG HTML 3.2 Reference](#)
- [U in Learning HTML 3.2 by Examples](#)

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

HTML 4.0 Latin-1 Entities



Latin-1 Entities

The following table gives the character entity reference, decimal character reference, and hexadecimal character reference for 8-bit characters in the Latin-1 (ISO-8859-1) character set. [Glyphs](#) of the characters are available at the [Unicode Consortium](#).

Browser support is generally best for the decimal character references, except for the accented characters (decimal 192-214, 216-246, 248-255), where the character entity references hold a slight edge.

Note that most Mac browsers will render fourteen Latin-1 characters incorrectly. These characters are decimal 166, 178, 179, 185, 188, 189, 190, 208, 215, 221, 222, 240, 253, and 254. See [ISO-8859-1 and the Mac platform](#) for more information.

Note that hexadecimal character references will cause errors with current validators (except the [WDG HTML Validator](#)) since these references are defined in a recent revision to the SGML standard that is not yet supported by most validators.

Character	Entity	Decimal	Hex
no-breaking space	 	160	A0
inverted exclamation mark	&icml;	161	A1
cent sign	¢	162	A2
pound sign	£	163	A3
currency sign	&cur;	164	A4

etion	
ary	
hyph	
en	
regis	® &#x
tere	; 74; AE;
d	
sign	
=	
regis	
tere	
d	
trad	
e	
mar	
k	
sign	
mac	¯ &#x
ron	cr; 75; AF;
=	
spac	
ing	
mac	
ron	
=	
overl	
ine =	
APL	
over	
bar	
degr	° &#x
ee	g; 76; B0;
sign	
plus-	± &#x
minu	smn;77; B1;
s	
sign	
=	
plus-	
or-	
minu	
s	
sign	
supe	² &#x
rscri	p2; 78; B2;
pt	
two	
=	
supe	
rscri	
pt	
digit	
two	
=	
squa	
red	
supe	³ &#x
rscri	p3; 79; B3;
pt	
thre	
e =	
supe	

rscript
digit
three =
cubed
acute ‘ ’
accent =
spacing
acute
micro – —
sign
pilcrow — ―
sign
=
paragraph
sign
middle dot – —
=
Georgian
comma
=
Greek
middle dot
cedilla = – —
spacing
cedilla
superscript
one
=
superscript
digit
one
masculine – —
ordinal

indicator
right
-
pointing
double
angle
quotation
mark =
right
pointing
guillemet
vulgar
fraction
one
quarter =
fraction
one
quarter
vulgar
fraction
one
half =
fraction
one
half
vulgar
fraction
three
quarters =
fraction
three
quarters
inverted
question
mark

&raq &#x
uo; 87; BB;

&fra &#x
c14; 88; BC;

&fra &#x
c12; 89; BD;

&fra &#x
c34; 90; BE;

&iqu &#x
est; 91; BF;

mark = turned question mark
Latin capital letter A with grave = Latin capital letter A grav e
Latin capital letter A with acute
Latin capital letter A with circumflex
Latin capital letter A with tilde
Latin capital letter A with diaeresis
Latin capital letter A with ring

&Ag &#x rave 92; C0;
;
&Aa &#x cute;93; C1;
&Aci &#x rc; 94; C2;
&Atil &#x de; 95; C3;
&Au &#x ml; 96; C4;
&Ari &#x ng; 97; C5;

capit al lette r l with grav e	ave; 04; CC;
Latin capit al lette r l with acut e	&lac &#x ute; 05; CD;
Latin capit al lette r l with circu mfle x	&lcir &#x c; 06; CE;
Latin capit al lette r l with diae resis	&lu &#x ml; 07; CF;
Latin capit al lette r ETH	&ET &#x H; 08; D0;
Latin capit al lette r N with tilde	&Ntil &#x de; 09; D1;
Latin capit al lette r O with grav e	&Og &#x rave 10; D2; ;
Latin capit al lette r O with acut e	&Oa &#x cute;11; D3;
Latin	&Oci &#x

capit al lette r O with circu mfle x	rc; 12; D4;
Latin capit al lette r O with tilde	&Otil &#x de; 13; D5;
Latin capit al lette r O with diae resis	&Ou &#x ml; 14; D6;
multi plica tion sign	&tim &#x es; 15; D7;
Latin capit al lette r O with strok e =	&Osl &#x ash; 16; D8;
Latin capit al lette r O slas h	
Latin capit al lette r U with grav e	&Ug &#x rave 17; D9; ;
Latin capit al lette r U with acut e	&Ua &#x cute;18; DA;
Latin capit al lette	&Uci &#x rc; 19; DB;

r U with circu mfl e x	&Uu &#x ml; 20; DC;
Latin capit al lette r U with diae resis	&Ya &#x cute;21; DD;
Latin capit al lette r Y with acut e	&TH &#x ORN22; DE; ;
Latin capit al lette r THO RN	&szli &#x g; 23; DF;
Latin smal l lette r shar p s = ess- zed	&agr &#x ave; 24; E0;
Latin smal l lette r a with grav e =	&aa &#x cute;25; E1;
Latin smal l lette r a grav e	&aci &#x rc; 26; E2;

l
lette
r a
with
circu
mflex
Latin
small
l
lette
r a
with
tilde
Latin
small
l
lette
r a
with
diae
resis
Latin
small
l
lette
r a
with
ring
above
=
Latin
small
l
lette
r a
ring
Latin
small
l
lette
r ae
=
Latin
small
l
ligat
ure
ae
Latin
small
l
lette
r c
with
cedil
la
Latin
small
l
lette
r e

ã ⟣

&au ˤ

å ⧥

&ael ユ

&cc ㇧

è ㋨

with grav e Latin small l lette r e with acut e	&ea &#x cute;33; E9;
Latin small l lette r e with circu mfl e x	&eci &#x rc; 34; EA;
Latin small l lette r e with diae resis	&eu &#x ml; 35; EB;
Latin small l lette r i with grav e	&igr &#x ave; 36; EC;
Latin small l lette r i with acut e	&iac &#x ute; 37; ED;
Latin small l lette r i with circu mfl e x	&icir &#x c; 38; EE;
Latin small l lette r i with diae resis	&iu &#x ml; 39; EF;
Latin	ð &#x

small letter Latin small letter with tilde	&ntil &#x de; 41; F1;
small letter with tilde Latin small letter with grave	&ogr &#x ave; 42; F2;
small letter with acute Latin small letter with acute	&oa &#x cute; 43; F3;
small letter with acute Latin small letter with acute	&oci &#x rc; 44; F4;
small letter with circumflex Latin small letter with tilde	&otil &#x de; 45; F5;
small letter with tilde Latin small letter with tilde	&ou &#x ml; 46; F6;
small letter with diaeresis Latin small letter with diaeresis	&divi &#x de; 47; F7;
small letter with diaeresis Latin small letter with diaeresis	&osl &#x ash; 48; F8;
small letter with stroke	=

Latin small letter ro slash h	
Latin small letter ru with grave	&ugr &#x ave; 49; F9;
Latin small letter ru with acute	&ua &#x cute;50; FA;
Latin small letter ru with circu mflex	&uci &#x rc; 51; FB;
Latin small letter ru with diae resis	&uu &#x ml; 52; FC;
Latin small letter ry with acute	&ya &#x cute;53; FD;
Latin small letter rn	&tho &#x rn; 54; FE;
Latin small letter ry with diae	&yu &#x ml; 55; FF;

resis

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Symbols and Greek Letters](#) ~

[Other Special Characters](#)

[Copyright ©](#) 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

HTML 4.0 Attributes Values



HTML 4.0 Attribute Values

There are various kinds of attribute values in HTML 4.0. This document describes common types of values.

CDATA

Attribute values of type CDATA are made up of a sequence of characters that may include [entities](#). Line feeds are ignored while each carriage return and tab is replaced with a space. Browsers may ignore leading and trailing whitespace within the attribute value.

CDATA attribute values are typically case-sensitive, though this is not the case with all attributes that take CDATA values.

ID and NAME

Attribute values of type ID and NAME must begin with a letter in the range A-Z or a-z and may be followed by letters (A-Za-z), digits (0-9), hyphens ("-"), underscores ("_"), colons (":"), and periods ("."). These values are case-sensitive.

IDREF and IDREFS

IDREF and IDREFS values refer to values of other elements' [ID](#) attributes. An IDREF value is a single ID while an IDREFS value is a space-separated list of IDs. IDREF and IDREFS are case-sensitive.

Number

Number attribute values are numbers made of at least one digit in the range 0-9.

Text

Text attribute values are [CDATA](#) values intended to be human readable strings.

URI

URI attribute values are Uniform Resource Identifiers (URIs), as defined in the [URI Internet-Draft](#), a work-in-progress that is expected to replace [RFC 1738](#) and [RFC 1808](#).

URI attribute values may include full URIs such as <http://www.htmlhelp.com/> as well as relative URIs such as [foo.html](#) and [../foo/](#).

While parts of a URI may be case-insensitive, in general URI values are case-sensitive.

Color

Color attribute values give a color definition. The value can be any hexadecimal number, specified according to the sRGB color space, or one of sixteen color names. Hexadecimal numbers must be prefixed by a "#" character.

The case-insensitive color names and their sRGB values are as follows:



Note that the hexadecimal form is better supported than the color names for the color attributes of [BODY](#).

Pixels

These attribute values are integers that represent a number of pixels.

Length

Length attribute values may be either an integer--interpreted as a number of pixels--or a percentage of the horizontal or vertical space. The value **50%** means half the available space while **50** means 50 pixels.

MultiLength and MultiLengths

MultiLength attribute values may be an integer in pixels, a percentage of the horizontal or vertical space, or a relative length expressed as i^* where i is an integer. In allotting space, a browser first allots pixel and percentage lengths, then divides the remaining space among all elements with a relative length. An element with a length of 3^* will be allotted three times the space of an element with length 1^* . The value $*$ is equivalent to 1^* and is often used to mean "fill the remaining space."

A MultiLengths value is a comma-separated list of MultiLength values.

ContentType and ContentTypes

These attribute values are content types (also known as media types or MIME types) of a linked or embedded resource. Values of type `ContentType` give a single content type while values of type `ContentTypes` give a comma-separated list of content types. Content types are case-insensitive.

Commonly used content types include `text/html`, `image/jpeg`, `model/vrml`, `video/quicktime`, `application/java`, `text/css`, and `text/javascript`. Many [common content types](#) are registered at the IANA.

LanguageCode

Attribute values of type `LanguageCode` specify a language code according to [RFC 1766](#). Examples of language codes include `en` for English, `en-US` for American English, and `ja` for Japanese. Whitespace is not allowed in the language code, which is case-insensitive.

Charset and Charsets

These attribute values specify character encodings of linked resources. Values of type `Charset` give a single character encoding while values of type `Charsets` give a space- and/or comma-separated list of character encodings. Character encodings are case-insensitive.

Examples of character encodings include `ISO-8859-1`, `SHIFT_JIS`, and `UTF-8`. A list of [registered character encodings](#) is available at the IANA.

Character

Character attribute values take a single Unicode character. The character may be specified with an [entity](#).

Datetime

Datetime attribute values give a date and time in the format `YYYY-MM-DDThh:mm:ssTZD` where `YYYY` is a four-digit year, `MM` is a two-digit month (`01` through `12`), `DD` is a two-digit day (`01` through `31`), `hh` is a two-digit hour (`00` through `23`), `mm` is a two-digit minute (`00` through `59`), `ss` is a two-digit second (`00` through `59`), and `TZD` is the time zone designator. Note that the `T` separating the date from the time must appear literally and is case-sensitive.

The time zone designator may be one of the following:

- `Z` (case-sensitive), which indicates Universal Coordinated Time (UTC, basically the same as GMT);
- `+hh:mm`, the time ahead of UTC in hours and minutes;
- `-hh:mm`, the time behind UTC in hours and minutes.

If any of the two-digit components of the Datetime value are unknown, `00` should be used.

LinkTypes

Attribute values of type LinkTypes give a space-separated list of link types. A link type is case-insensitive and may not contain whitespace. Each link type may be used any number of times in a given document.

While link types are case-insensitive, the [Lynx](#) browser will render **LINK** elements with the same case as is given in the **REL** or **REV** attribute. Authors should therefore be consistent in their case, and may wish to capitalize the first letter while using lowercase for the rest.

The following link types are defined in HTML 4.0, though authors may use other link types. The **Made** link type, widely used as `<LINK REV=Made HREF="mailto:liam@htmlhelp.com">` to provide a contact link for the document author, is notably missing from the list of link types defined in HTML 4.0.

- **Alternate** specifies an alternate version of the document. When used with the **LINK** element's **HREFLANG** attribute, this value implies a translation of the document. When used with **LINK**'s **MEDIA** attribute, a media-specific version (e.g., for printing) is implied.
- **StyleSheet** specifies an [external style sheet](#) for the document. This link type can be combined with the **Alternate** link type to define an alternate style sheet for the user to choose.
- **Start** specifies the first document in a collection.
- **Next** specifies the next document in a suggested sequence of reading. Browsers such as WebTV will preload documents identified as "next" to improve the perceived load time.
- **Prev** specifies the previous document in a suggested sequence of reading.
- **Contents** specifies a table of contents for the document.
- **Index** gives an index for the document.
- **Glossary** gives a glossary of terms used in the document.
- **Copyright** specifies a document with copyright information.
- **Chapter** specifies the chapter of a collection of documents.
- **Section** specifies the section of a collection of documents.
- **Subsection** specifies a subsection of a collection of documents.
- **Appendix** gives an appendix for the collection of documents.
- **Help** specifies a help document.
- **Bookmark** refers to a key related document. The **TITLE** attribute provides a label for the bookmark.

MediaDesc

Attribute values of type MediaDesc are *media descriptors*--a comma-separated list of media for which the linked resource is tailored. Media descriptors are *case-sensitive*.

The following media descriptors are defined in HTML 4.0:

- **screen** (the default), for non-paged computer screens;
- **tty**, for fixed-pitch character grid displays (such as the display used by [Lynx](#));
- **tv**, for television-type devices with low resolution and limited scrollability;
- **projection**, for projectors;
- **handheld**, for handheld devices (characterized by a small, monochrome display and limited bandwidth);
- **print**, for output to a printer;
- **braille**, for braille tactile feedback devices;
- **aural**, for speech synthesizers;
- **all**, for all devices.

Script

Script attribute values are client-side scripts, typically a function call or a few short statements. The value may contain [entities](#) (e.g., `"`).

StyleSheet

Attribute values of type StyleSheet are style sheet data. The value may contain [entities](#).

FrameTarget

Attribute values of type FrameTarget must begin with a letter in the range A-Z or a-z, with the exception of the following special values that begin with an underscore:

- **_blank** renders the link in a new, unnamed window
- **_self** renders the link in the current frame (useful for overriding a **BASE TARGET**)
- **_parent** renders the link in the immediate **FRAMESET** parent
- **_top** renders the link in the full, unframed window

Maintained by [Liam Quinn](#) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](#) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

HTML 4.0 Font Style Elements



Font Style Elements

- **B** - Bold text
- **BIG** - Large text
- *I* - Italic text
- ~~S~~ - Strike-through text
- SMALL - Small text
- ~~STRIKE~~ - Strike-through text
- TT - Teletype text
- U - Underlined text

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Elements Alphabetically](#)

Copyright © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.

Visit my web site [WEBalley](#) - web publishing for beginners.

This file is freeware and may be distributed freely.

HTML 4.0 Phrase Elements



Phrase Elements

- ⌞ [ABBR](#) - Abbreviation
- ⌞ [ACRONYM](#) - Acronym
- ⌞ [CITE](#) - Citation
- ⌞ [CODE](#) - Computer code
- ⌞ [DEL](#) - Deleted text
- ⌞ [DFN](#) - Defined term
- ⌞ [EM](#) - Emphasis
- ⌞ [INS](#) - Inserted text
- ⌞ [KBD](#) - Text to be input
- ⌞ [SAMP](#) - Sample output
- ⌞ [STRONG](#) - Strong emphasis
- ⌞ [VAR](#) - Variable

Maintained by [Liam Quinn](mailto:liam@htmlhelp.com) <liam@htmlhelp.com>



[Elements Alphabetically](#)

[HTML 4.0 Reference](#) ~ [Elements by Function](#) ~

[Copyright](#) © 1998 Liam Quinn. All rights reserved.



This help file is created by [Gerben Hoekstra](mailto:weballey@usa.net) <weballey@usa.net>.
Visit my web site [WEBalley](#) - web publishing for beginners.
This file is freeware and may be distributed freely.

If you can see this, your help file has no topics!

This URL was excluded from the pulldown

The protocol used in this URL is not supported

This URL referenced an unknown protocol

This link was broken while compiling the help file

This URL contained unknown content

Robots are excluded from this URL

This link was not retrieved

This link was skipped because it was either a query string or an invalid URL

This link was skipped at user request

