HyperT ext Markup Langage W orking Group
Request for Comments: XXX

T. Berners-Lee, CERN
D. Connolly , HaL
K. Muldrow , HaL

HyperText Markup Language (HTML)

Version 2.0

Status of this Memo

This RFC specifies an IAB standards track protocol for the Internet community , and requests discussion and suggestions for improvements. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Contents

Abstract

HyperText Markup Language (HTML) is a simple markup system used to create hypertext documents which are portable from one platform to another. Essentially, HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. HTML markup can represent

- Hypertext news, mail, documentation, and hypermedia
- Menus of options
- Database query results
- Simple structured documents with inlined graphics
- Hypertext views of existing bodies of information

The World Wide Web initiative (W3) links information throughout the world. To accomplish this, W3 uses the Internet Hypertext Transfer Protocol (HTTP), which allows transfer representations to be negotiated between client and server. Results are returned in a MIME body part.

HTML is one of the representations used by W3, and is proposed as a MIME content type. Additionally, HTML is an application conforming to International Standard ISO 8879 -- Standard Generalized Markup Language (SGML), which is a system for defining structured document types.

The HTML Overview provides a quick reference to the HTML specification. For details, see the complete HTML Specification.

HTML Specification Overview

In HTML documents, tags define the start and end of headings, paragraphs, lists, character highlighting and links. Most HTML elements are identified in a document as a start tag, which gives the element name and attributes, followed by the content, followed by the end tag. Start tags are delimited by < and >, and end tags are delimited by </ and >. For example:

```
<h1>This is a Heading</h1>
<p>This is a paragraph.
```

Every HTML document should contain a head and body text, and use other HTML components as necessary. This overview describes briefly the syntax of HTML elements, and provides an example of HTML code.

- Anchor Element (Level 0)
- Block Elements (Level 0)
- List Elements (Level 0)
- Highlighting Elements (Level 1)
- Image Element (Level 1)
- Form Elements (Level 2)
- Special Characters (Level 0)

Note: Obsolete and proposed elements are not included in this overview.

HTML Anchor

The Anchor element is a feature supported by HTML Level 1. It is shown in the table HTML Anchor Element.

Table 1–1 HTML Anchor Element

Element	Tag	Explanation or Example
Anchor	<a>...	An anchor specifies a link to another location (<a href>) or the value to use when linking to this location from another location (<a name>): See HaL’s information for more details. Indecent dress is defined as dress which the community finds offensive. The restaurant may refuse service to anyone who is indecently dressed.

HTML Blocks

Block formatting elements supported by HTML Level 1. They are shown in the table HTML Block Elements.

Table 1–2 HTML Block Elements

Element	Tag	Explanation or Example
Address	<address>... </address>	<address> Newsletter editor J.R. Brown JimquickPost News, Jumquick, CT 01234 Tel (123) 456 7890 </address>
Body	<body>... </body>	Place the <body> and </body> tags on either side of the body of the text (not including the head) of your html document.
Blockquote	<blockquote>... </blockquote>	I think it ends <blockquote>Soft you now, the fair Ophelia. Nymph, in thy orisons, be all my sins remembered. </blockquote> but I am not sure.

Element	Tag	Explanation or Example
Head	<head>... </head>	Every HTML document must have a head, which provides a title. Example: <html> <head><title>Introduction to HTML</title> </head>
Headings	...	HTML includes heading levels from <h1> through <h6>: <h1>This is a first level heading</h1> <p>This text appears under the first level heading. <h2>Second level heading</h2> <p>This text appears under the second level heading
Horizontal Rule (Separator)	<hr>	<h1>Name of Applicant</h1> <hr> <h2>Professionnnal Experience <p>Text describing professional experience.
HTML Label	<html>... </html>	The <html> and </html> tags should appear at the beginning and end of html file contents.
Line Break	 	Pease porridge hot Pease porridge cold Pease porridge in the pot Nine days old.
Paragraph	...[ ]	<h1>This Heading Precedes the Paragraph</h1> <p>This is the text of the first paragraph. <p>This is the text of the second paragraph. Although you do not need to start paragraphs on new lines, maintaining this convention facilitates document maintenance. <p>This is the text of a third paragraph.
Preformatted Text	<pre>... </pre>	<pre width="80"> This is an example of preformatted text. </pre>
Quoted Material	<blockquote>...</blockquote>	As someone once said: <blockquote>If I could think of a good quote, I would make sure it was formatted appropriately. </blockquote> If you think of a good quote let me know.
Title	<title>...</title>	<title>A Brief History of SGML</title>

HTML Lists

List elements are a part of HTML Level 0. They are shown in the table HTML List Elements.

Table 1–3 HTML Lists

Element	Tag	Explanation or Example
Definition List	<dl>... </dl>	<dl> <dt>Term<dd>This is the first definition. <dt>Term<dd>This is the second definition. </dl> With the compact attribute: <dl compact> <dt>Term<dd>This is a compact definition. <dt>Term<dd>This is another compact definition. </dl>
Directory List	<dir>...</dir>	<dir> A-H I-M M-R S-Z </dir>
Menu List	<menu>... </menu>	<menu> First item in the list. Second item in the list. Third item in the list. </menu>
Ordered List	...	 Click the Web button of the OLIAS Launcher.OLIAS opens the Open URL window. Enter the URL number in the text field of the Open URL window.OLIAS displays the Web document you specified. Click highlighted text to move from one link to another.
Unordered List	...	 This is the first item in the list. This is the second item in the list. This is the third item in the list.

HTML Highlighting

Highlighting elements are a part of HTML Level 1. They are shown in the table HTML Highlighting Elements.

Table 1–4 HTML Highlighting Elements

Element	Tag	Explanation or Description
Bold	... 	Specifies that the text be displayed in boldface, where available, otherwise alternative mapping is allowed.
Citation	<cite>...</cite>	Specifies a citation.
Code Example	<code>... </code>	Indicates an example of code, typically monospaced font. (Do not confuse with <pre>).
Emphasis	...	Provides typographic emphasis, typically italics.
Italics	<i>... </i>	Specifies that the text be displayed in italic font (or slanted if italic unavailable).
Keyboard	<kbd>... </kbd>	Indicates text typed by a user.
Sample	<samp>... </samp>	Indicates a sequence of literal characters.
Strong Emphasis	... 	Provides strong typographic emphasis, typically bold.
Typewriter	<tt>... </tt>	Specifies that the text be displayed in fixed-width typewriter font.
Underlined	<u>... </u>	Specifies that the text be displayed as underlined. Not supported by Mosaic.
Variable	<var>... </var>	Indicates a variable name.

HTML Images

The Image element is a feature included in HTML Level 1. It is shown in the table HTML Image Element.

Table 1–5 HTML Image Element

Element	Tag	Explanation or Example
Image Import	<img>	Use the <img> tag to import graphics. <img> takes these attributes: • align - (top, middle or bottom) • alt - (alternative text for display-text-only environments) • ismap • src - (the URL of the object to import). Example: `<img src ="triangle.gif" alt="Warning:"> Be sure to read these instructions.`

HTML Forms

Forms are included in HTML Level 2. They are shown in the table HTML Form Elements.

Table 1–6 HTML Form Elements

Element	Tag	Description
form	<form>...</form>	Forms are composed by placing input fields within paragraphs, preformatted/literal text, lists and tables. Forms use these nested elements: input, option, select, textarea.

input	<input>...</input>	Takes these attributes: align, maxlength, name, size, src, type, value. The type attribute can define these field types: checkbox, hidden, image, radio, reset, submit, text.
		For example:
		`<form method="post" action="http://www.hal.com/sample">`
		`<p>Your name: <input name="name" size="48">`
		`<p>Male <input name="male" type=radio>`
		`<p>Female <input name="female" type=radio>`
		`</form>`

option	<option>...</option>	The option element can only occur within a select element. It represents one choice, and can take selected or value attributes.

Element	Tag	Description

select	<select>...</select>	`<select name="flavor">`
		`<option>Vanilla`
		`<option>Strawberry`
		`<option>Rum and Raisin`
		`<option>Peach and Orange`
		`</select>`

textarea	<textarea>... </textar-
	ea>
		`<textarea name="address" ROWS=64 COLS=6>`
		`HaL Computer Systems`
		`1314 Dell Avenue`
		`Campbell California 95008`
		`</textarea>`

Special Characters in HTML

Because of the way special characters are used in marking up HTML text, you use character strings to represent the less than (<) and greater than (>) symbols and the ampersand (&) as shown in the table HTML Graphic Character Representations.

Table 1–7 HTML Graphic Character Representations

Character	Representation
Less than sign (<)	<
Greater than" sign (>)	>
Ampersand sign itself (&)	&
Double quote sign (")	"
Non-breaking space	
Soft-hyphen	

HTML also allows references to any of the ISO Latin-1 alphabet, using the names in the table ISO Latin-1 Character Representations, which is derived from "ISO 8879:1986//ENTITIES Added Latin 1//EN".

Table 1–8 ISO Latin-1 Character Representations

Letter or Character	Representation
capital AE diphthong (ligature)	Æ
capital A, acute accent	Á

Letter or Character	Representation
capital A, circumflex accent	Â
capital A, grave accent	À
capital A, ring	Å
capital A, tilde	Ã
capital A, dieresis or umlaut mark	Ä
capital C, cedilla	Ç
capital Eth, Icelandic	Ð
capital E, acute accent	É
capital E, circumflex accent	Ê
capital E, grave accent	È
capital E, dieresis or umlaut mark	Ë
capital I, acute accent	Í
capital I, circumflex accent	Î
capital I, grave accent	Ì
capital I, dieresis or umlaut mark	Ï
capital N, tilde	Ñ
capital O, acute accent	Ó
capital O, circumflex accent	Ô
capital O, grave accent	Ò
capital O, slash	Ø
capital O, tilde	Õ
capital O, dieresis or umlaut mark	Ö
capital THORN, Icelandic	Þ
capital U, acute accent	Ú
capital U, circumflex accent	Û
capital U, grave accent	Ù
capital U, dieresis or umlaut mark	Ü
capital Y, acute accent	Ý
small a, acute accent	á
small a, circumflex accent	â

Letter or Character	Representation
small ae diphthong (ligature)	æ
small a, grave accent	à
small a, ring	å
small a, tilde	ã
small a, dieresis or umlaut mark	ä
small c, cedilla	ç
small e, acute accent	é
small e, circumflex accent	ê
small e, grave accent	è
small eth, Icelandic	ð
small e, dieresis or umlaut mark	ë
small i, acute accent	í
small i, circumflex accent	î
small i, grave accent	ì
small i, dieresis or umlaut mark	ï
small n, tilde	ñ
small o, acute accent	ó
small o, circumflex accent	ô
small o, grave accent	ò
small o, slash	ø
small o, tilde	õ
small o, dieresis or umlaut mark	ö
small sharp s, German (sz ligature)	ß
small thorn, Icelandic	þ
small u, acute accent	ú
small u, circumflex accent	û
small u, grave accent	ù
small u, dieresis or umlaut mark	ü
small y, acute accent	ý
small y, dieresis or umlaut mark	ÿ

Example HTML File

An example HTML file is:

```
<html><head>
<title>Structural Example</title>
</head>
<body>
<h1>First Header</h1>
<p>This is a paragraph in the example HTML file.
Keep in mind that the title does not appear in the
document text, but that the header (defined by h1) does.
</ul>
<li>First item in an unordered list.
<li>Second item in an unordered list.
</ul>
<p>This is an additional paragraph. Technically, end tags
are not required for paragraphs, although they are allowed.
You can include character highlighting in a paragraph.
<i>This sentence of the paragraph is in italics.</i>
<img src =”triangle.gif” alt=”Warning:”> Be sure to read
these instructions.
</body>
</html>
```

HTML Specification

An SGML Application Conforming to International Standard
ISO 8879 -- Standard Generalized Markup Language

This specification describes the current standards of HyperText Markup Language (HTML). It is a draft of a standard for interchange of information on the network which is proposed to be registered as a MIME (RFC1521) content type.

HTML Document Type Definitions on page 5–1, including the *SGML Declaration for HTML* on page 5–1, are provided at the end of this specification. This documentation also includes the *HTML DTD Reference (Level 2)* on page 6–1 and the *HTML DTD Reference (Level 0)* on page 7–1.

This specification is currently available in hypertext on the World-Wide Web as http://www.hal.com/%7Econnolly/html-spec. Please send comments to the discussion list www-html@info.cern.ch.

Version

This version of the HTML specification (Version 2.0, released July 31, 1994) introduces forms for user input of information, and adds a distinction between levels of conformance:

Level 0
: Indicates the minimum conformance level. When writing documents at Level 0, authors can have a high level of confidence that the rendering at different sites will reflect their intent.

Level 1
: Includes Level 0 features plus features such as highlighting and images.

Level 2
: Includes all Level 0 and Level 1 features, plus forms. Features of higher levels which are under discussion, (such as tables, figures, and mathematical formulae) where mentioned are described as proposed.

Understanding HTML and MIME

The World Wide Web initiative (W3) links information throughout the world. To accomplish this, W3 uses the Internet Hypertext Transfer Protocol (HTTP), which allows transfer representations to be negotiated between client and server. Results are returned in a MIME body part.

HTML is one of the representations used by W3, and is proposed as a MIME content type. The definition of the HTML Content-Type is `text/html`, and has 3 optional parameters:

Level
: The level parameter specifies the feature set which is used in the document. The level is an integer number, implying that any features of same or lower level may be present in the document. Levels are defined by this specification.

Version
: To help avoid future compatibility problems, the version parameter may be used to give the version number of the specification to which the document conforms. The version number appears at the front of this document and within the public identifier for the SGML DTD.

Character sets
: The charset parameter is reserved for future use. It will be used to override the base character set of the SGML declaration. Support of character sets other than ISO 8859/1 Latin alphabet No. 1 is not a requirement for conformance with this specification.

The SGML declaration specifies the base character set for HTML as ISO 8859/1, also known as *Latin-1* This is the set referred to by any numeric character references. The actual character set used in the representation of an HTML document may be ISO 8859/1, or its 7-bit subset which is ISO 646 (ASCII).

Since you can represent an HTML document using either 7-bit or 8-bit characters, it is the transport medium that is most likely to pose a constraint. If you are using email to send an HTML document you should convert the document to 7-bit characters. However, the HTTP access protocol used by W3 always allows 8 bit transfer.

When an HTML document is encoded using 7-bit characters, use the mechanisms of character references and entity references to encode characters in the upper half of the ISO 8859/1 Latin-1 set.

Implementations of HTML parsers and generators can be found in W3 servers and browsers and public domain W3 code. You can also build HTML parsers and generators using various public domain SGML parsers such as [sgmls].

Understanding HTML and SGML

HTML is an application conforming to International Standard ISO 8879 -- Standard Generalized Markup Language (SGML). SGML is a system for defining structured document types, and markup languages to represent instances of those document types. The SGML declaration for HTML is given in *SGML Declaration for HTML*. It is implicit among WWW implementations.

In the event of any apparent conflict between HTML and SGML standards, the SGML standard is definitive.

Every SGML document has three parts:

- An SGML declaration, which binds SGML processing quantities and syntax token names to specific values.

  Example: The SGML declaration in the HTML DTD specifies that the string that opens an end tag is `</` and the maximum length of a name is 72 characters.

- A prologue including one or more document type declarations, which specify the element types, element relationships and attributes, and references that can be represented by markup. An instance, which contains the data and markup of the document.

HTML refers to the document type as well as the markup language for representing instances of that document type.

Working with Structured Text

An HTML document is like a text file, except that some of the characters are markup. Markup (tags) define the structure of the document.

To identify information as HTML, each HTML document should start with the prologue:

```
<!doctype html public "-//W3O//DTD W3 HTML 2.0//EN">
```

Note: If an existing HTML document does not contain a prologue, this prologue must be prepended.

HTML documents should also contain an <html> tag at the beginning of the file (after the prologue), and </html> at the end. Within those tags, an HTML document is organized as a head and a body, much like memo or a mail message. Within the head, you can specify the title and other information about the document. Within the body, you can structure text into paragraphs and lists as well as highlighting phrases and creating links. You do this using HTML elements.

This section describe the syntax of HTML Elements, and provides an example HTML document.

Understanding HTML Elements

In HTML documents, tags define the start and end of headings, paragraphs, lists, character highlighting and links. Most HTML elements are identified in a document as a start tag, which gives the element name and attributes, followed by the content, followed by the end tag. Start tags are delimited by < and >, and end tags are delimited by </ and >. For example:

```
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
```

Some elements appear as just a start tag. For example, to create a line break, you use
. Additionally, the end tags of some other elements (e.g. p, li, dt, dd) may be omitted.

Note: Technically, the start and end tags for html, head, and body elements are omissible. However, this is not recommended since the head/ body structure allows an implementation to determine certain properties of a document -- the title, for example -- without parsing the entire document.

The content of an element is a sequence of characters and nested elements. Some elements, such as anchors, cannot be nested. Anchors and character highlighting may be put inside other constructs.

Note: The SGML declaration for HTML specifies SHORTTAG YES, which means that there are some other valid syntaxes for tags, such as NET tags, <em/.../; empty start tags, <>; and empty end tags, </>. Until such time as support for these idioms is widely deployed, their use is strongly discouraged.

Names

The element name immediately follows the tag open delimiter . An element name consist of a letter followed by up to 72 letters, digits, periods, or hyphens. Names are not case sensitive. For example, `H1` is equivalent to `h1`.

Attributes

In a start tag, whitespace and attributes are allowed between the element name and the closing delimiter . An attribute typically consists of an attribute name, an equal sign, and a value (although some attributes may be just a value). Whitespace is allowed around the equal sign.

The value of the attribute may be either:

- A string literal, delimited by single quotes or double quotes
- A name token (a sequence of letters, digits, periods, or hyphens)

In this example, *a* is the element name, *href* is the attribute name, and *http://host/dir/file.html* is the attribute value:

```
<a href="http://host/dir/file.html">
```

Some implementations consider any occurrence of the > character to signal the end of a tag. For compatibility with such implementations, when > appears in an attribute value, you may want to represent it with an entity or numeric character reference, such as: **`<img src="eq1.ps" alt="a > b">`**. To put quotes inside of quotes, you use the character representation *"* as in: **`<img src="image.ps" alt="First "real" example">`**

The length of an attribute value (after replacing entity and numeric character references) is limited to 1024 characters.

Note: Some implementations allow any character except space or > in a name token. Attributes values must be quoted only if they don't satisfy the syntax for a name token.

Attributes with a declared value of name (e.g. ismap, compact) may be written using a minimized syntax. The markup:

```
<ul compact="compact">
```

can be written as

```
<ul compact>
```

Note: Unless you use the minimized syntax, some implementations won't understand.

Undefined tag and attribute names

It is an accepted networking principle to be conservative in that which one produces, and liberal in that which one accepts. HTML parsers should be liberal except when verifying code. HTML generators should generate strictly conforming HTML.

The behavior of WWW applications reading HTML documents and discovering tag or attribute names which they do not understand should be to behave as though, in the case of a tag, the whole tag had not been there but its content had, or in the case of an attribute, that the attribute had not been present.

Special Characters

The characters between the tags represent text in the ISO-Latin-1 character set, which is a superset of ASCII. Because certain characters will be interpreted as markup, they should be represented by markup -- entity or numeric character references. See the Special Characters section of this specification for more information.

Comments

To include comments in an HTML document that will be ignored by the parser , surround them with <!-- and -->. After the comment delimiter , all text up to the next occurrence of -- is ignored. Hence comments cannot be nested. Whitespace is allowed between the closing -- and >. (But not between the opening <! and --.)

For example:

```
<head>
<title>HTML Guide: Recommended Usage</title>
<!-- Id: Text.html,v 1.6 1994/04/25 17:33:48 connolly Exp -->
</head>
```

Note: Some historical implementations incorrectly consider a > sign to terminate a comment.

Example HTML File

An example HTML file is:

```
<!doctype html public "-//W3O//DTD W3 HTML 2.0//EN">
<html><head>
<title>Structural Example</title>
</head>
<body>
<h1>First Header</h1>
<p>This is a paragraph in the example HTML file.
Keep in mind that the title does not appear in the
document text, but that the header (defined by h1) does.
</ul>
<li>First item in an unordered list.
<li>Second item in an unordered list.
</ul>
<p>This is an additional paragraph. Technically, end tags
are not required for paragraphs, although they are allowed.
You can include character highlighting in a paragraph.
<i>This sentence of the paragraph is in italics.</i>
<img src =”triangle.gif” alt=”Warning:”> Be sure to read
these instructions.
</body>
</html>
```

Head

Level: 0

The head of an HTML document is an unordered collection of information about the document. It requires the <title> element between <head> and </head> tags in this format:

```
<head><title>Introduction to HTML</title>
</head>
```

Within the head element, only certain elements are allowed. Information in the head element may include:

- base
- isindex
- link

- meta (proposed)
- nextid
- title

base

Level: 0

The base element allows the URL of the document itself to be recorded in situations in which the document may be read out of context. URLs within the document may be in a “partial” form relative to this base address.

Where the base address is not specified, the browser uses the URL it used to access the document to resolve any relative URLs.

The one attribute for base is href, which identifies the URL.

isindex

Level: 0

The isindex element informs the browser that the document is an index document. As well as reading it, the reader may use a keyword search.

The document can be queried with a keyword search by adding a question mark to the end of the document address, followed by a list of keywords separated by plus signs. See the network address format for more information.

Note: A server normally generates the isindex tag automatically. If added by hand to an HTML document, the browser assumes that the server can handle a search on the document. Obviously the server must have this capability for it to work: simply adding <isindex> in the document is not enough to make searches happen if the server does not have a search engine!

Example:

```
<isindex>
```

link

Level: 1

The link element occurs within the head element of an HTML document. It indicates a relationship between the document and some other object. A document may have any number of link elements.

The link element is empty, but takes the same attributes as the anchor element.

Typical uses are to indicate authorship, related indexes and glossaries, older or more recent versions, etc. Links can indicate a static tree structure in which the document was authored by pointing to a "parent" and "next" and "previous" document, for example.

Servers may also allow links to be added by those who do not have the right to alter the body of a document.

nextid

Level: 0

The nextid is a parameter used by editors to generate unique identifiers. This tag takes a single attribute which is the number of the next document-wide numeric identifier to be allocated of the form z123.

When modifying a document, old anchor ids should not be reused, as there may be references stored elsewhere which point to them. This is read and generated by hypertext editors. Human writers of HTML usually use mnemonic alphabetical identifiers. Browser software may ignore this tag.

Example:

```
<nextid n=z27>
```

Note: Support for nextid does not impact browsers in any way.

title

Level: 0

Every HTML document must contain a title element. The title should identify the contents of the document in a global context, and may be used in a history lists and as a label for the window displaying the document. In contrast with headings, titles are not normally displayed in the text of a document itself.

The title element must occur within the head of the document, and may not contain anchors, paragraph tags, or highlighting. There may only be one title in any document.

While there is no specific limit to the length of titles, long titles may be truncated in some applications. To minimize this possibility, keep titles to fewer than 64 characters. At the same time be aware that a very short title, such as *Introduction*, may be meaningless out of context.

An meaningful title might be

```
<title>A Recipe for Maple Syrup Flap-Jacks</title>
```

Body

Level: 0

The body of an HTML document contains all the information which is part of the document, arranged in the order in which it is presented to the reader. Specifically, the body of a document may contain links, text, and formatting information within <body> and </body> tags.

Note: All elements described in this specification, except for Highlighting, Images, and Forms elements, are supported under HTML Level 0. HTML Level 1 adds support for Highlighting and Images elements. All elements, including those related to Forms, are HTML Level 2.

These elements may be included in the body of an HTML document:

- Address
- Anchor
- Blockquote
- Headings
- Highlighting
- Horizontal Rule
- Images
- Line Break
- Lists
- Paragraphs
- Preformatted Text

- Special Characters
- Forms and Input fields

Address

Level: 0

The address element specifies such information as address, signature and authorship, often at the top or bottom of a document.

Typically , an address element is italic and/or right justified or indented. The address element implies a paragraph break.

An example of using the address element is:

```
<address>
Newsletter editor<br>
J.R. Brown<br>
JimquickPost News, Jumquick, CT 01234<br>
Tel (123) 456 7890
</address>
```

Anchor

Level: 0

A section of text which forms the start and/or destination of a hypertext link is called an “anchor” and is defined by the <a> tag.

```
<a href=”http://www.hal.com”>HaL Computer Systems</a>
```

The most common attributes of the anchor tag are:

href
: Optional - Level 0. If the href attribute is present, the anchor is sensitive text: the start of a link. If the reader selects this text, (s)he moves to another document whose network address is defined by the value of the href attribute.

  The format of the network address is specified in the URI specification for print readers. W ith the href attribute, the form href=”#identifier” can refer to another anchor in the same document. If the anchor is in another document, the attribute may be a relative name (relative to the document’ s address or the specified base address).

name
: Optional - Level 0. If present, the attribute name allows the anchor to be the destination of a link. The value of the attribute is an identifier for the anchor .

Identifiers are arbitrary strings but must be unique within the HTML document. Another document can then make a reference explicitly to this anchor by putting the identifier after the address, separated by a hash sign.

title

Optional - Level 1. The title element is informational only. If present the title attribute should provide the title of the document whose address is given by the href attribute. This is useful for at least two reason:

- The browser software may chose to display the title of the document prior to retrieving it, for example as a margin note or on a small box while the mouse is over the anchor, or during document fetch.
- Some documents -- mainly those which are not marked up text, such as graphics, plain text and also Gopher menus, do not come with a title themselves, and so putting a title in the link is the only way to give them a title. Obviously it leads to duplication of data, and so it is dangerous to assume that the title attribute of the link is a valid and unique title for the destination document.

rel (proposed)

Proposed - Level 1. An attribute rel may give the relationship (s) described by the hypertext link.

rev (proposed)

Proposed - Level 1. The rev attribute is the same as rel, but the semantics of the link type are in the reverse direction.

urn

Optional - Level 1. If present, the urn attribute specifies a uniform resource number (URN) for the document. URNs allow a document to be recognized if duplicate copies are found. This prevents a client implementation from picking up a copy of something it already has.

The format of URNs is under discussion (1993) by various working groups of the Internet Engineering Task Force.

methods

Optional - Level 1. The methods attributes of anchors and links provide information about the functions which the user may perform on an object. These are more accurately given by the HTTP protocol when it is used, but it may, for similar reasons as for the title attribute, be useful to include the information in advance in the link. For example, the browser may chose a different rendering as a function of the methods allowed (for example something which is searchable may get a different icon).

The value of the methods attribute is a comma separated list of HTTP methods supported by the object for public use.

Although all attributes are optional, either name or href is necessary for the anchor to be useful. Example:

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

<a name=indecent>Indecent</a> dress is defined as dress which the community finds
offensive.

The restaurant may refuse service to anyone who is <a
href="#indecent">indecently</a> dressed.
```

See also: link.

Blockquote

Level: 0

The blockquote element allows text quoted from another source to be rendered in a particular way .

A typical rendering might be a slight extra left and right indent, and/or italic font. blockquote causes a paragraph break, and typically provides a line or so of white space before and after the quote.

Single-font rendition may reflect the quotation style of Internet mail by putting a vertical line of graphic characters (such as the greater than symbol, >) in left margin.

An example of using the blockquote element is:

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

Headings

Level: 0

HTML defines six heading levels. A heading element implies all the font changes, paragraph breaks before and after, and white space necessary to render the heading. Further character emphasis or paragraph marks are not required in HTML headings.

The heading elements are <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, with <h1> being the highest level of heading. For example:

```
<h1>This is a heading</h1>
Here is some text
<h2>Second level heading</h2>
Here is some more text.
```

The style of each heading level is up to the designer of the presentation software. Since the rendering software is responsible for generating vertical white space between elements, do not follow a heading element with a paragraph tag.

A typical rendering for headings is:

h1
: Bold very large font, centered. One or two lines clear space between this and anything following. If printed on paper, start new page.

h2
: Bold, large font, flush left against left margin, no indent. One or two clear lines above and below.

h3
: Italic, large font, slightly indented from the left margin. One or two clear lines above and below.

h4
: Bold, normal font, indented more than h3. One clear line above and below.

h5
: Italic, normal font, indented as h4. One clear line above.

h6
: Bold, indented same as normal text, more than h5. One clear line above.

Note: Although this is legal skip header levels (for example to go from h1 to h3), it is discouraged. Skipping header levels may produce strange results when generating other representations from HTML.

Parser Interaction with Headers

Parsers should not require any specific order to heading elements, even if the heading level increases by more than one between successive headings.

Highlighting

Level: 2

With character highlighting elements you can format sections to provide emphasis by designating an information type (such as a variable) or by using character formatting. Highlighting tags may be ignored by minimal implementations.

In-line character tagging falls left to right in the rendered flow of text. The tags do not cause a paragraph break, and may be used on sections of text within paragraphs. Whitespace between the opening and closing tags separates words.

All character highlighting elements have related closing tags and use this format:

```
This is <em>emphasized</em> text.
```

Level 1 implementations must render highlighted text distinctly from plain text. Additionally, em content must be rendered as distinct from strong content, and b content must rendered as distinct from i content.

Highlighting elements are allowed within the content of other highlighting elements, but implementations are not required to render these nested highlighting elements distinctly from non-nested elements. For example, implementations may render these two cases identically:

```
plain <B>bold <I>italic</I></B>

plain <B>bold </B><I>italic</I>
```

Some character highlighting styles are more explicit than others about how they should be physically represented. Designate the information type rather than the character format wherever possible, unless for example it is necessary to refer to the formatting in the text as in “The italic parts are mandatory”.

Information Type Elements

cite
: The cite tag specifies a citation. Sections tagged with the cite style typically display in italics.

code

The code style indicates an example of code. typically monospaced font. (Do not confuse with <pre>).

dfn (proposed)

Proposed. The dfn tag indicates the defining instance of a term.

em

The em tag provides typographic emphasis, typically italics.

While em and i often give the same effect, use em except in the case that it is necessary to refer to the formatting in the text, as in “The italic parts are mandatory”. This improves consistency between documents from various sources if, for example, a reader prefers to use color in stead of italics for emphasis.

kbd

The kbd style indicates text typed by a user . It might typically be used in an instruction manual.

samp

The samp type indicates a sequence of literal characters.

strike (proposed)

Proposed. The strike tag indicates “strike out” text, as in a legal document. This tag is not widely supported.

strong

The strong tag provides strong typographic emphasis, typically bold.

var

The var tag indicates a variable name.

Character Formatting Elements

b

The b tag specifies that the text be displayed in boldface, where available, otherwise alternative mapping is allowed.

i

The i tag specifies that the text be displayed in italic font (or slanted if italic unavailable).

tt

The tt tag specifies that the text be displayed in fixed-width typewriter font.

u (proposed)
: Proposed. The u tag specifies that the text be displayed as underlined.

Example

This example shows how you might use highlighting elements in an HTML document:

```
This text contains an <em>emphasized</em> word.
<strong>Don't assume</strong> that it will be italic!
It was made with the <CODE>EM</CODE> element. A cite is
typically italic and has no formal necessary structure:
<cite>Moby Dick</cite> is a book title.
```

Horizontal Rule

Level: 0

A horizontal rule (<hr>) is a divider between sections of text such as a full width horizontal rule or equivalent graphic. The horizontal rule is typically used for separating heading information (when more than just a heading) from content, etc.

```
<h1>Name of Applicant</h1>
<hr>
<h2>Professionnnal Experience
<p>Text describing professional experience.
```

Images

Level: 1

Note: You can use <img> for Level 0 if you include the alt attribute.

To incorporate in-line graphics into an HTML document, you use the img tag. W ith <img> you can insert another document in-line. The document is normally an icon or small graphic, etc. This element is NOT intended for embedding other HTML text.

Browsers which are not able to display in-line images ignore the img element unless it contains the alt attribute. Note that some browsers will be able to display (or print) linked graphics but not in-line graphics. If the graphic is essential, you may want to create a link to it rather than to put it in-line. If the graphic is essentially decorative, then img is appropriate.

The img element is empty: it has no closing tag. It has these attributes:

align

Take values top or middle or bottom, defining whether the tops or middles of bottoms of the graphics and text should be aligned vertically.

alt

Optional alternative text as an alternative to the graphics for display in text-only environments. This attribute is mandatory for Level 0 documents.

ismap

An image map is a graphical map by which users can navigate transparently from one information resource to another. The ismap attribute identifies an image as an image map. For example:

```
<a href="http://machine/htbin/imagemap/sample">
<img src="sample.gif" ismap>
</a>
```

For information on creating an image map, see Marc Andreessen's Graphical Information Map Tutorial at http://wintermute.ncsa.uiuc.edu:8080/map-tutorial/image-maps.html

src

The value of the src attribute is the URL of the document to be embedded. Its syntax is the same as that of the href attribute of the a tag. src is mandatory.

Note: img elements are allowed within anchors.

An example of using <img> to insert an in-line graphic is:

```
<img src ="triangle.gif" alt="Warning:">
Be sure to read these instructions.
```

Line Break

Level: 0

The line break (
) element specifies that a new line must be started at the given point. A new line indents the same as that of line-wrapped text.

An example of using the line break element is:

```
<p>Pease porridge hot<br>
Pease porridge cold<br>
Pease porridge in the pot<br>
Nine days old.</p>
```

Lists

Level: 0

HTML supports several types of lists, all of which may be nested:

- Definition List
- Directory List
- Menu List
- Ordered List
- Unordered List

Definition List (dl)

Level: 0

A definition list (<dl>)is a list of terms (<dt>) and corresponding definitions (<dd>). Definition lists are typically formatted with the term in the left wide margin and the definition on one or more lines to the right of the term.

If the <dt> term does not fit in the <dt> column (one third of the display area), it may be extended across the page with the <dd> section moved to the next line, or it may be wrapped onto successive lines of the left hand column.

The <dt> and <dd> elements should appear in pairs. Single occurrences of <dt> without a subsequent <dd> are allowed, and have the same significance as if the <dd> had been present with no text.

The opening list tag must be <dl>. It must be immediately followed by the first term (<dt>). For example:

```
<dl>
<dt>Term<dd>This is the definition of the first term.
<dt>Term<dd>This is the definition of the second term.
</dl>
```

The definition list type can take the compact attribute, which suggests that a compact rendering be used, because the enclosed elements are individually small, or the whole list is rather large, or both.

Unless you provide the compact attribute, HTML leaves white space between successive dt, dd pairs.The compact attribute may also reduce the width of the left-hand (dt) column.

The opening list tag must be dl compact. It must be immediately followed by the first term (dt). For example:

```
<dl compact>
<dt>Term<dd>This is the first definition in compact format.
<dt>Term<dd>This is the second definition in compact
format.
</dl>
```

Directory List (dir)

Level: 0

A directory list (<dir>) is a list of items containing fewer than 20 characters each. The items in a directory list may be arranged in columns (typically 24 characters wide). If the rendering software is able to optimize the column width as function of the widths of individual elements, so much the better .

The opening list tag must be <dir>. It must be immediately followed by the first list item ( ). For example:

```
<dir>
<li>A-H<li>I-M
<li>M-R<li>S-Z
</dir>
```

Menu List (menu)

Level: 0

A menu list (<menu>) is a list of items with typically one line per item. The menu list style is more compact than the style of an unordered list.

The opening list tag must be <menu>. It must be immediately followed by the first list item ( ). For example:

```
<menu>
<li> First item in the list.
<li> Second item in the list.
<li> Third item in the list.
</menu>
```

Ordered List (ol)

Level: 0

An ordered list ( )is a list of items in which the items are typically numbered in such a way as to indicate the order as significant.

The opening list tag must be <ol>. It must be immediately followed by the first list item ( ). For example:

```
<ol>
<li>Click the Web button of the OLIAS Launcher.
OLIAS opens the Open URL window.
<li>Enter the URL number in the text field of the Open URL
window.OLIAS displays the Web document you specified.
<li>Click highlighted text to move from one link to another.
</ol>
```

Unordered List (ul)

Level: 0

An *unordered list* is a list of items that is typically separated by some white space and/or marked by bullets, etc.

The opening list tag must be ul. It must be immediately followed by the first list item (li). For example:

```
<ul>
<li> This is the first item in the list.
<li> This is the second item in the list.
<li> This is the third item in the list.
</ul>
```

Paragraphs

Level: 0

The paragraph (<p>) element represents a paragraph. The exact rendering of this (indentation, leading, etc) is not defined and may be a function of other tags, style sheets etc.

Note: You do not need to use <p> to put white space around heading, list, address or blockquote elements. It is the responsibility of the rendering software to generate that white space. Do not use empty paragraphs.

Typically , paragraphs are surrounded by a small vertical space (of a line or half a line). This is not the case (typically) within address or (ever) within pre elements. With some implementations, normal paragraphs may have a small extra left indent on the first line.

Example:

```
<h1>This Heading Precedes the Paragraph</h1>
<p>This is the text of the first paragraph.
<p>This is the text of the second paragraph. Although you
do not need to start paragraphs on new lines, maintaining
this convention facilitates document maintenance.
<p>This is the text of a third paragraph.
```

Preformatted Text

Level: 0

The preformatted text element <pre> presents sections in fixed-width font, and so is suitable for text which has been formatted for a teletype by some existing formatting system.

The <pre> element may be used with the optional width attribute, which is a Level 1 feature. The width attribute specifies the maximum number of characters for a line and allows the presentation system to select a suitable font and indentation. Where the width attribute is not present, it is recommended that a width of 80 be assumed. Where width is supported, it is recommended that at least widths of 40, 80 and 132 characters be presented optimally, with other widths being rounded up.

Within preformatted text:

- Line boundaries within the text are rendered as a move to the beginning of the next line, except for one immediately following or immediately preceding a tag.
- The <p> tag should not be used. If found, it should be rendered as a move to the beginning of the next line.
- Anchor elements and character highlighting elements may be used.
- Elements which define paragraph formatting (Headings, Address, etc) must not be used.
- The ASCII Horizontal Tab (ht) character must be interpreted as the smallest positive nonzero number of spaces which will leave the number of characters so far on the line as a multiple of 8. Its use is not recommended however.

Note: References to the “beginning of a new line” do not imply that the renderer is forbidden from using a (constant) left indent for rendering preformatted text. The left indent may of course be constrained by the width required.

An example of using the <pre> element is:

```
<pre width="80">
 This is an example line.
 </pre>
```

Note: Within a preformatted element, the constraint that the rendering must be on a fixed horizontal character pitch may limit or prevent the ability of the renderer to render highlighting elements specially .

Forms and Input fields

Level: 2

Forms are composed by placing input fields within paragraphs, preformatted/literal text, lists and tables. This gives considerable flexibility in designing the layout of forms.

Elements

The form features use the following elements which are all known as HTML level 2 elements.

form
: A form within a document.

input
: One input field.

option
: One option within a select.

select
: A selection from a finite set of options.

textarea
: A multi-line input field.

Each variable field is defined by an input, textarea, or option element and must have an name attribute to identify its value in the data returned when the form is submitted.

Example

This fictitious example is a questionnaire:

```
<title>Sample Questionaire</title>
<h1>Sample Questionaire</h1>
Please fill out this questionaire:
<form method="post" action="http://www.hal.com/sample">
<p>Your name: <input name="name" size="48">
<p>Male <input name="male" type=radio>
<p>Female <input name="female" type=radio>
Number in family: <input name="family" type=int>
<p>Cities in which you maintain a residence:
<ul>
<li>Kent <input name="city" type=checkbox value="kent">
<li>Miami <input name="city" type=checkbox value="miami">
<li>Other <textarea name="other" cols=48 rows=4></textarea>
</ul>
Nickname: <input name="nickname" size ="42">
<p>Thank you for responding to this questionaire.
<p align=center><input type=submit> <input type=reset>
</form>
```

Here, the <p> and <ul> elements have been used to lay out the text (and input fields). The browser is responsible for handling the input focus, i.e. which field will currently get keyboard input.

Many platforms have existing conventions for forms, for example, using Tab and Shift keys to move the keyboard focus forwards and backwards between fields, and using the Enter key to submit the form. In the example, the Submit and Reset buttons are specified explicitly with special purpose fields. The Submit button is used to email the form or send its contents to the server as specified by the action attribute, while Reset resets the fields to their initial values. When the form consists of a single text field, it may be appropriate to leave such buttons out and rely on the Enter key.

The input element is used for a large variety of typed of input fields.

When you need to let users enter more than one line of text, you should use the textarea element.

Representing choices

The radio and checkbox types of input field can be used to specify multiple choice forms in which every alternative is visible as part of the form. An alternative is to use the select element which is generally rendered in a more compact fashion as a pull down combo list.

form

Level: 2

The form element is used to delimit a data input form. There can be several forms in a single document, but the form element can't be nested.

The action attribute is a URL specifying the location to which the contents of the form is submitted to elicit a response. If the action is missing, the URL for the document itself is assumed. The way data is submitted varies with the access protocol of the URL, and with the values of the method and enctype attributes. In general:

- the method attribute selects variations in the protocol.
- the enctype attribute specifies the format of the submitted data in case the protocol does not impose a format itself.

This standard defines and requires support for the http access protocol only .

When action is an HTTP URL, the method must be an HTTP method as defined by the HTTP method specification in the IETF draft HTTP standard. The default method is get, although for many applications, the post method may be preferred. With the post method, the enctype attribute is a MIME type specifying the format of the posted data (by default application/x-www-form-urlencoded).

Under any protocol, the submitted contents of the form logically consist of name/value pairs. The names are usually equal to the name attributes of the various interactive elements in the form.

Note: The names are not guaranteed to be unique keys, nor are the names of form elements required to be distinct. The values encode the user's input to the corresponding interactive elements. Elements capable of displaying a textual or numerical value will return a name/value pair even when they receive no explicit user input.

input

Level: 2

The input element represents a field whose contents may be edited by the user . It has these attributes.

align
: Vertical alignment of the image - for use only with type=image. The possible values are exactly the same as for the align attribute of the <img> element.

checked
: Indicates that a checkbox or radio button is selected. Unselected checkboxes and radio buttons do not return name/value pairs when the form is submitted.

maxlength
: Indicates the maximum number of characters that can be entered into a text field. This can be greater that specified by size, in which case the field will scroll appropriately . The default is unlimited.

name
: Symbolic name used when transferring the form's contents. This attribute is required for most input types and is normally used to provide a unique identifier for a field, or for a logically related group of fields.

size
: Specifies the size or precision of the field according to its type. For example, specify **`input type text size="24"`** to specify a field with a visible width of 24 characters.

src
: A URL or URN specifying an image - for use only with type=image.

type
: Defines the type of data the field accepts. Defaults to free text. Several types of fields can be defined with the type attribute:

  checkbox
  : Used for simple Boolean attributes, or for attributes which can take multiple values at the same time. The latter is represented by a number of checkbox fields each of which has the same name. Each selected checkbox generates a separate name/value pair in the submitted data, even if this results in duplicate names. The default value for checkboxes is **`on`**.

  hidden
  : No field is presented to the user , but the content of the field is sent with the submitted form. This value may be used to transmit state information about client/server interaction.

image

An image field upon which you can click with a pointing device, causing the form to be immediately submitted. The coordinates of the selected point are measured in pixel units from the upper left corner of the image, and are returned (along with the other contents of the form) in two name/value pairs. The *x*-coordinate is submitted under the name of the field with *.x* appended, and the *y*-coordinate is submitted under the name of the field with *.y* appended. Any value attribute is ignored. The image itself is specified by the src attribute, exactly as for the img element.

Note: In a future version of the HTML standard, the image functionality will be folded into an enhanced submit field.

radio

For attributes which can take a single value from a set of alternatives. Each radio button field in the group should be given the same name. Only the selected radio button in the group generates a name/value pair in the submitted data. Radio buttons require an explicit value attribute.

reset

This is a button that when pressed resets the form's fields to their initial values as specified by the value attribute.

submit

This is a button that when pressed submits the form. You can use the value attribute to provide a (non-editable) label to be displayed on the button (the default label is application-specific).

The name attribute is a name/value pair to the submitted data if it has a name and was the button that was pressed in order to submit the form.

text

Single line text entry fields. Use in conjunction with the size and maxlength attributes. Use the textarea element for text fields which can accept multiple lines.

value

The initial displayed value of the field, if it displays a textual or numerical value; or the value to be returned when the field is selected, if it displays a Boolean value. This attribute is required for radio buttons.

option

Level: 2

The option element can only occur within a select element. It represents one choice, and can take these attributes:

disabled
: Proposed.

selected
: Indicates that this option is initially selected.

value
: When present indicates the value to be returned if this option is chosen. The returned value defaults to the contents of the option element.

The contents of the option element is presented to the user to represent the option. It is used as a returned value if the value attribute is not present.

select

Level: 2

The select element allows the user to chose one of a set of alternatives described by textual labels. Every alternative is represented by the option element. Possible attributes are:

error
: Proposed.

multiple
: The multiple attribute is needed when users are allowed to make several selections, e.g. <select multiple>.

select is typically rendered as a pull down or pop-up list. For example:

```
<select name="flavor">
<option>Vanilla
<option>Strawberry
<option>Rum and Raisin
<option>Peach and Orange
</select>
```

textarea

Level: 2

When you need to let users enter more than one line of text, you should use the textarea element, e.g.

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

The text up to the end tag is used to initialize the field's value. This end tag is always required even if the field is initially blank.

In a typical rendering, the rows and cols attributes determine the visible dimension of the field in characters. The field is rendered in a fixed-width font. Browsers should allow text to grow beyond these limits by scrolling as needed.

Note: In the initial design for forms, multi-line text fields were supported by the input element with type=text. Unfortunately, this causes problems for fields with long text values as SGML limits the length of attribute literals. The HTML 2.0 DTD allows for up to 1024 characters (the SGML default is only 240 characters!).

Special Characters

Level: 0

The characters between html start and end tags represent text in the ISO-Latin-1 character set, which is a superset of ASCII. Because special characters are interpreted as markup, they should be represented by entity or numeric character references.

To indicate special characters, HTML uses the representations shown in the table HTML Character Representations. These representations are always prefixed by ampersand (&) and should be followed by a semicolon, as shown. They represent particular graphic characters which have special meanings in places in the markup, or may not be part of the character set available to the writer.

For example:

```
When a&#60;b, we can show that...
Brought to you by AT&amp;T
```

To ensure that a string of characters has no markup, it is sufficient to represent all occurrences of <, >, and & by character or entity references.

Note: There are SGML features (CDATA, RCDATA) to allow most <, >, and & characters to be entered without the use of entity or character references. Because these features tend to be used and implemented inconsistently, and because they require 8-bit characters to represent non-ASCII characters, they are not employed in this version of the HTML DTD.

An earlier HTML specification included an XMP element whose syntax is not expressible in SGML. Inside the XMP, no markup was recognized except the </xmp> end tag. While implementations are encouraged to support this idiom, its use is obsolete.

Table 2–1 HTML Character Representations

Character	Representation
Less-than sign (<)	<
Greater-than sign (>)	>
Ampersand sign itself (&)	&
Double quote sign (“)	"
Non-breaking space (proposed)	
Soft-hypen (proposed)	

HTML also allows references to any of the ISO Latin-1 alphabet, using the names in the table ISO Latin-1 Character Representations, which is derived from *ISO 8879:1986//ENTITIES Added Latin 1//EN*.

Table 2–2 ISO Latin-1 Character Representations

Letter or Character	Representation
capital AE diphthong (ligature)	Æ
capital A, acute accent	Á
capital A, circumflex accent	Â

Letter or Character	Representation
capital A, grave accent	À
capital A, ring	Å
capital A, tilde	Ã
capital A, dieresis or umlaut mark	Ä
capital C, cedilla	Ç
capital Eth, Icelandic	Ð
capital E, acute accent	É
capital E, circumflex accent	Ê
capital E, grave accent	È
capital E, dieresis or umlaut mark	Ë
capital I, acute accent	Í
capital I, circumflex accent	Î
capital I, grave accent	Ì
capital I, dieresis or umlaut mark	Ï
capital N, tilde	Ñ
capital O, acute accent	Ó
capital O, circumflex accent	Ô
capital O, grave accent	Ò
capital O, slash	Ø
capital O, tilde	Õ
capital O, dieresis or umlaut mark	Ö
capital THORN, Icelandic	Þ
capital U, acute accent	Ú
capital U, circumflex accent	Û
capital U, grave accent	Ù
capital U, dieresis or umlaut mark	Ü
capital Y, acute accent	Ý
small a, acute accent	á
small a, circumflex accent	â
small ae diphthong (ligature)	æ

Letter or Character	Representation
small a, grave accent	à
small a, ring	å
small a, tilde	ã
small a, dieresis or umlaut mark	ä
small c, cedilla	ç
small e, acute accent	é
small e, circumflex accent	ê
small e, grave accent	è
small eth, Icelandic	ð
small e, dieresis or umlaut mark	ë
small i, acute accent	í
small i, circumflex accent	î
small i, grave accent	ì
small i, dieresis or umlaut mark	ï
small n, tilde	ñ
small o, acute accent	ó
small o, circumflex accent	ô
small o, grave accent	ò
small o, slash	ø
small o, tilde	õ
small o, dieresis or umlaut mark	ö
small sharp s, German (sz ligature)	ß
small thorn, Icelandic	þ
small u, acute accent	ú
small u, circumflex accent	û
small u, grave accent	ù
small u, dieresis or umlaut mark	ü
small y, acute accent	ý
small y, dieresis or umlaut mark	ÿ

Security Considerations

Anchors, embedded images, and all other elements which contain URIs as parameters may cause the URI to be dereferenced. In this case, the security considerations of the URI specification apply .

Documents may be constructed whose visible contents mislead the reader to follow a link to unsuitable or offensive material.

Proposed HTML Features

These HTML elements and attributes are proposed. This means that they are not currently supported under HTML Levels 0, 1, or 2, but may be supported at some time in the future.

- anchor attributes
- defining instance
- meta
- special characters
- strike
- underline

anchor

The anchor tag, <a> has these proposed attributes:

rel

Proposed - Level 1. An attribute rel may give the relationship (s) described by the hypertext link. The value is a comma-separated list of relationship values. Values and their semantics will be registered by the HTML registration authority . The default relationship if none other is given is void. rel should not be present unless href is present.

rev

Proposed - Level 1. The rev attribute is the same as rel, but the semantics of the link type are in the reverse direction. A link from A to B with rel="X" expresses the same relationship as a link from B to A with rev="X". An anchor may have both rel and rev attributes.

defining instance

The defining instance, <dfn>, tag indicates the defining instance of a term. The output format for this tag is typically bold or bold italic. Although it can be found in some documents, <dfn> is not widely supported.

meta

Status: Level 1 - Proposed

The meta element is proposed to be used within the head element to embed document meta-information not defined by other HTML elements. Such information can be extracted by servers/clients for use in identifying, indexing, and cataloging specialized document meta-information.

Although it is generally preferable to use named elements which have well-defined semantics for each type of meta-information (e.g. title), this element is provided for situations where strict SGML parsing is necessary and the local DTD is not extensible.

In addition, HTTP servers can read the content of the document head to generate response headers corresponding to any elements defining a value for the attribute HTTP-EQUIV . This provides document authors a mechanism (not necessarily the preferred one) for identifying information which should be included in the response headers for an HTTP request.

The attributes of the meta element are:

http-equiv
: This attribute binds the element to an HTTP response header . It means that if you know the semantics of the HTTP response header named by this attribute, then you can process the contents based on a well-defined syntactic mapping, whether or not your DTD tells you anything about it. HTTP header names are not case sensitive. If not present, the attribute name should be used to identify this meta-information and it should not be used within an HTTP response header .

name
: Meta-information name. If not present, the name can be assumed equal to the value of http-equiv.

content

The meta-information content to be associated with the given name and/or HTTP response header .

Examples

If the document contains:

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

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

when the http-equiv attribute is not present, the server should not generate an HTTP response header for this meta-information; e.g.,

```
<meta name="IndexType" content="Service">
```

Do *not* use the meta element to define information that should be associated with an already existing HTML element. This is an inappropriate use of the meta element:

```
<meta name="Title" content="The Etymology of Dunsel">
```

Do *not* name an http-equiv equal to a response header that should normally only be generated by the HTTP server . Example names that are inappropriate include “Server”, “Date”, and “Last-modified” -- the exact list of inappropriate names is dependent on the particular server implementation. It is recommended that servers ignore any meta elements which specify http-equivalents which are equal (case-insensitively) to their own reserved response headers.

Special Characters

To indicate special characters, HTML uses entity or numeric representations. T wo proposed character presentations are shown in the table Proposed Character Representations:

Table 3–1 Proposed Character Representations

Character	Representation
Non-breaking space (proposed)	
Soft-hypen (proposed)	

strike

The <strike> tag is proposed to indicate “strike out” text, as in a legal document. Although it may be found in HTML documents, <strike> not widely supported.

underline

The underline tag, <u>, is proposed to indicate that the text should be displayed as underlined. For example:

```
The text <u>shown here</u> is displayed in
the document as underlined.
```

Obsolete HTML Features

The features described in this section are not standard HTML. Client implementors should implement these obsolete forms for back-compatibility with previous versions of the HTML specification:

- comment
- hp - highlighted phrase
- plaintext
- xmp, listing

comment

Status: Obsolete

A comment element used for bracketing off unneeded text and comments has been introduced in some browsers but will be replaced by the SGML command feature in new implementations.

hp - highlighted phrase

Status: Obsolete

The highlighted phrase (hp) elements like all others should be ignored if not implemented. Replaced will more meaningful elements -- see Highlighting.

An example of using the hp element is:

```
<hp1>first highlighted phrase</hp1>non
highlighted text<hp2>second highlighted phrase</hp2> etc.
```

plaintext

Status: Obsolete

The empty <plaintext> tag terminates the HTML entity . What follows is not SGML. In stead, an old HTTP convention specified that what followed was an ASCII (MIME “text/plain”) body .

For example:

```
<plaintext>
0001 This is line one of a ling listing
0002 file from <any@host.inc.com> which is sent
```

The <plaintext> tag allows the rest of a file to be read efficiently without parsing. Its presence is an optimization. There is no closing tag. The rest of the data is not in SGML.

xmp, listing

Status: Obsolete

The xmp and listing elements are in use in some locations and should be recognized by browsers. New servers should use <pre> instead.

These styles allow text of fixed-width characters to be embedded absolutely as is into the document. The syntax is:

```
<listing>
...
</listing>
```

or

```
<xmp>
...
</xmp>
```

The text between these tags is to be portrayed in a fixed width font, so that any formatting done by character spacing on successive lines will be maintained. Between the opening and closing tags:

- The text may contain any ISO Latin printable characters, but not the end tag opener . The xmp and listing elements have historically used specifications which do not conform to SGML. Specifically , the text may contain ISO Latin printable characters, including the tag opener , as long it they does not contain the closing tag in full.
- SGML does not support this form. Implementations may vary on how they interpret end tags apparently within these elements.

- Line boundaries are significant, except any occurring immediately after the opening tag or before the closing tag. and are rendered as a move to the start of a new line.
- The ASCII Horizontal T ab (ht) character must be interpreted as the smallest positive nonzero number of spaces which will leave the number of characters so far on the line as a multiple of 8. Its use is not recommended.

The listing element is portrayed so that at least 132 characters will fit on a line. The XMP elements portrayed in a font so that at least 80 characters will fit on a line but is otherwise identical to listing.

HTML Document Type Definitions

This section provides:

SGML Declaration for HTML

This is the SGML Declaration for HyperText Markup Language (HTML) as used by the World-Wide Web (WWW) application:

```
<!SGML  "ISO 8879:1986"
 --
        SGML Declaration for HyperText Markup Language (HTML)
        as used by the World-Wide Web (WWW) application.

 --

 CHARSET
         BASESET  "ISO 646:1983//CHARSET
                   International Reference Version (IRV)//ESC 2/5 4/0"
         DESCSET  0   9   UNUSED
                  9   2   9
                  11  2   UNUSED
                  13  1   13
                  14  18  UNUSED
                  32  95  32
                  127 1   UNUSED
     BASESET   "ISO Registration Number 100//CHARSET
                ECMA-94 Right Part of Latin Alphabet Nr. 1//ESC 2/13 4/1"
     DESCSET   128 32 UNUSED
               160 96 32

 CAPACITY         SGMLREF
                  TOTALCAP        150000
                  GRPCAP          150000

 SCOPE    DOCUMENT
```

```
SYNTAX
         SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
                           19 20 21 22 23 24 25 26 27 28 29 30 31 127
         BASESET  "ISO 646:1983//CHARSET
                   International Reference Version (IRV)//ESC 2/5 4/0"
         DESCSET  0 128 0
         FUNCTION
              --  SPACE       32
                  TAB SEPCHAR  9
                  LF  SEPCHAR 10
                  FF  SEPCHAR 12
                  CR  SEPCHAR 13 --

        -- The above is an accurate description of the usage of FUNCTION --
        -- characters in HTML implementations; that is, there is no      --
        -- Record Start or Record End character, and no occurences of    --
        -- character 10 or 13 are "ignored" by the parser.               --
        -- But because few SGML implementations support this concrete    --
        -- sytax, we include the one below.                              --

        -- Note that in order to get correct behaviour w.r.t. newline    --
        -- processing, you will have to play some tricks in construcing  --
        -- the document entity for parsing in order to keep the parser   --
        -- from ignoring newlines in surpirsing ways                     --

                  RE          13
                  RS          10
                  SPACE       32
                  TAB SEPCHAR  9


         NAMING   LCNMSTRT ""
                  UCNMSTRT ""
                  LCNMCHAR ".-"
                  UCNMCHAR ".-"
                  NAMECASE GENERAL YES
                           ENTITY  NO
         DELIM    GENERAL  SGMLREF
                  SHORTREF SGMLREF
         NAMES    SGMLREF
         QUANTITY SGMLREF
                  NAMELEN  72    -- somewhat arbitrary; taken from
                                        internet line length conventions --
                  TAGLVL   100
                  LITLEN   1024
                  GRPGTCNT 150
                  GRPCNT   64

FEATURES
```

```
    MINIMIZE
      DATATAG  NO
      OMITTAG  YES
      RANK     NO
      SHORTTAG YES
    LINK
      SIMPLE   NO
      IMPLICIT NO
      EXPLICIT NO
    OTHER
      CONCUR   NO
      SUBDOC   NO
      FORMAL   YES
    APPINFO    NONE
  >
```

HTML Level 0 DTD

This is the Document Type Definition for the HyperText Markup Language as used by minimally conforming World Wide Web applications (HTML Level 0 DTD):

```
<!--
        $Id: html-spec.ps,v 1.3 1994/08/01 15:50:32 karen Exp $

        Author: Daniel W. Connolly <connolly@hal.com>

        See also: http://www.hal.com/%7Econnolly/html-spec
                  http://info.cern.ch/hypertext/WWW/MarkUp/MarkUp.html
 -->
<!--    html-0.dtd

        Document Type Definition for the HyperText Markup Language
        as used by minimally conforming World Wide Web applications
        (HTML Level 0 DTD).

        $Id: html-spec.ps,v 1.3 1994/08/01 15:50:32 karen Exp $

        Author: Daniel W. Connolly <connolly@hal.com>
        See Also: http://www.hal.com/%7Econnolly/html-spec/index.html
                  http://info.cern.ch/hypertext/WWW/MarkUp2/MarkUp.html
-->

<!ENTITY % HTML.Version
        "+//ISBN 82-7640-037::WWW//DTD HTML Level 0//EN//2.0"
        -- public identifier for "minimal conformance" version            --
```

```
        -- Typical usage:

            <!DOCTYPE HTML PUBLIC
                "+//ISBN 82-7640-037::WWW//DTD HTML Level 0//EN">
            <html>
            ...
            </html>
        --
        >

<!-- Feature Test Entities -->

<!ENTITY % HTML.Prescriptive "IGNORE"
        -- marks things that may become standard in future revisions -->

<![ %HTML.Prescriptive [
        <!ENTITY % HTML.Obsolete "IGNORE">
]]>

<!ENTITY % HTML.Obsolete "INCLUDE"
        -- marks things that may disappear in future revisions -->

<!ENTITY % HTML.Proposed "IGNORE">

<!-- Link Markup -->

<!ENTITY % URI "CDATA"
        -- The term URI means a CDATA attribute
           whose value is a Uniform Resource Identifier,
           as defined by
        "Universal Resource Identifiers" by Tim Berners-Lee
        aka http://info.cern.ch/hypertext/WWW/Addressing/URL/URI_Overview.html

        Note that CDATA attributes are limited by the LITLEN
        capacity (1024 in the current version of html.decl),
        so that URIs in HTML have a bounded length.

        -->

<!ENTITY % linkattributes-1 "" -- level 1 link attributes -->

<!ENTITY % linkattributes
        "NAME CDATA #IMPLIED
        HREF %URI;  #IMPLIED
        %linkattributes-1
        ">

<!-- Text Markup -->
```

```
<!ENTITY % text "#PCDATA | IMG | BR">

<!ELEMENT BR    - O EMPTY>

<!ELEMENT IMG    - O EMPTY --  Embedded image -->
<!ENTITY % img.alt.default "#REQUIRED" -- must have ALT in level 0 images -->
<!ATTLIST IMG
        SRC %URI;  #REQUIRED     -- URI of document to embed --
        ALT CDATA %img.alt.default;
        ALIGN (top|middle|bottom) #IMPLIED
        ISMAP (ISMAP) #IMPLIED
        >

<!--   Mnemonic character entities. -->

<!ENTITY % ISOlat1 PUBLIC
  "ISO 8879:1986//ENTITIES Added Latin 1//EN">
%ISOlat1;

<!ENTITY #DEFAULT SDATA "&#38;unkown;" --display the markup-->
<!ENTITY amp CDATA "&#38;"     -- ampersand          -->
<!ENTITY gt CDATA "&#62;"      -- greater than       -->
<!ENTITY lt CDATA "&#60;"      -- less than          -->
<!ENTITY quot CDATA "&#34;"    -- double quote       -->

<![ %HTML.Proposed [
<!ENTITY nbsp CDATA " "   -- non-breaking space -->
<!ENTITY shy  CDATA "&#173;"   -- soft hyphen        -->
]]>


<!-- DTD "macros" -->

<!ENTITY % heading "H1|H2|H3|H4|H5|H6">

<!ENTITY % list " UL | OL | DIR | MENU " >

<!ENTITY % block-2 "" -- block elements added in level 2 -->

<![ %HTML.Obsolete [
        <!ENTITY % block "P | %list | DL
                | PRE | XMP | LISTING
                | BLOCKQUOTE %block-2">
]]>

<!ENTITY % block "P | %list | DL
                | PRE | BLOCKQUOTE %block-2">

<!-- HyperText -->
```

```
<!ENTITY % htext "A | %text"    -- Plus links, no structure -->

<![ %HTML.Prescriptive [
        <!ENTITY % A.content   "(%text)+"
        -- The standard content model for A allows nonsense like:
                <H2><A>xyz<H1>h1</H1></A></H2>
        -->
]]>

<!ENTITY % A.content   "(%heading|%block|%text)+">

<!ELEMENT A     - - %A.content -(A)>

<!ATTLIST A
        %linkattributes;
        >

<!-- Paragraphs -->

<!ELEMENT P     - O (%htext)+>


<!-- Headings, Titles, Sections -->

<!ELEMENT HR    - O EMPTY -- horizontal rule -->

<!ELEMENT ( %heading )  - -  (%htext;)+>

<!ELEMENT TITLE - -  (#PCDATA)
          -- The TITLE element is not considered part of the flow of text.
             It should be displayed, for example as the page header or
             window title.
          -->


<!-- Text Flows -->

<!ENTITY % flow "(%htext|%block)*">

<!-- Lists -->


<!ELEMENT DL    - -  (DT*, DD?)+>
<!ATTLIST DL
        COMPACT (COMPACT) #IMPLIED>

<!ELEMENT DT    - O (%htext)+>
<!ELEMENT DD    - O %flow>
```

```
<!ELEMENT (%list) - -  (LI)+>
<!ATTLIST (%list)
        COMPACT (COMPACT) #IMPLIED>

<!ELEMENT LI    - O %flow>

<!-- Preformatted Text -->

<!ENTITY % pre.content "#PCDATA | A">
<!ELEMENT PRE - - (%pre.content)+>

<![ %HTML.Obsolete [

<!ENTITY % literal "CDATA"
        -- special non-conforming parsing mode where
           the only markup signal is the end tag
           in full
        -->

<!ELEMENT XMP - -  %literal>
<!ELEMENT LISTING - -  %literal>
<!ELEMENT PLAINTEXT - O %literal>

]]>


<!-- Document Body -->

<![ %HTML.Prescriptive [
        <!ENTITY % body.content "(%heading|%block|HR|ADDRESS)*">
]]>

<!ENTITY % body.content "(%heading | %htext | %block | HR | ADDRESS)*">

<!ELEMENT BODY O O  %body.content>

<!-- Misc. Body Elements -->

<!ELEMENT BLOCKQUOTE - - %body.content>

<![ %HTML.Prescriptive [
        <!ENTITY % address.content "(%htext)*">
]]>
<!ENTITY % address.content "(%htext|P)*">
<!ELEMENT ADDRESS - - %address.content>


<!-- Document Head -->
```

```
<!ENTITY % head.content-1 "" -- head elements in level-1 -->

<!ENTITY % head.content "TITLE & ISINDEX? & BASE? & NEXTID? %head.content-1">

<!ELEMENT HEAD O O  (%head.content)>

<!ELEMENT ISINDEX - O EMPTY
          -- WWW clients should offer the option to perform a search on
             documents containing ISINDEX.
          -->

<!ELEMENT BASE - O EMPTY    -- Reference context for URIs -->
<!ATTLIST BASE
        HREF %URI; #REQUIRED
        >

<!ELEMENT NEXTID - O EMPTY>
<!ATTLIST NEXTID N CDATA #REQUIRED
          -- The number should be a name suitable for use
             for the ID of a new element. When used, the value
             has its numeric part incremented. EG Z67 becomes Z68
          -->


<!-- Document Structure -->

<![ %HTML.Obsolete [
        <!ENTITY % html.content "HEAD, BODY, PLAINTEXT?">
]]>
<!ENTITY % html.content "HEAD, BODY">

<!ELEMENT HTML O O  (%html.content)>
<!ENTITY % version.attr "VERSION CDATA #FIXED &#34;%HTML.Version;&#34;">

<!ATTLIST HTML
        %version.attr;-- report DTD version to application --
        >
```

HTML Level 1 DTD

This is the Document Type Definition for the HyperText Markup Language with Level 1 Extensions (HTML Level 1 DTD):

```
<!--    html-1.dtd

         Document Type Definition for the HyperText Markup Language
         with Level 1 Extensions (HTML Level 1 DTD).

         $Id: html-spec.ps,v 1.3 1994/08/01 15:50:32 karen Exp $

         Author: Daniel W. Connolly <connolly@hal.com>
         See Also: http://www.hal.com/%7Econnolly/html-spec/index.html
                   http://info.cern.ch/hypertext/WWW/MarkUp2/MarkUp.html
 -->

 <!ENTITY % HTML.Version
         "+//ISBN 82-7640-037::WWW//DTD HTML Level 1//EN//2.0"

         -- Typical usage:

             <!DOCTYPE HTML PUBLIC
                 "+//ISBN 82-7640-037::WWW//DTD HTML Level 1//EN">
             <html>
             ...
             </html>
         --
         >


 <!ENTITY % HTML.Proposed "IGNORE"
         -- marks things that have been proposed, but aren't
            yet widely supported -->

 <!ENTITY % linkType "NAME"
         -- a list of these will be specified at a later date -->

 <!ENTITY % URI "CDATA" -- see html-0.dtd -->

 <!ENTITY % linkattributes-1
         "REL %linkType #IMPLIED -- forward relationship type --
         REV %linkType #IMPLIED -- reversed relationship type
                               to referent data: --
         URN CDATA #IMPLIED -- universal resource number --

         TITLE CDATA #IMPLIED -- advisory only --
         METHODS NAMES #IMPLIED -- supported public methods of the object:
                                     TEXTSEARCH, GET, HEAD, ... --
         ">


 <![ %HTML.Proposed [
 <!ENTITY % head.content-1 "& LINK* & META*">
```

```
<!ELEMENT META - O EMPTY    -- Generic Metainformation -->
<!ATTLIST META
        HTTP-EQUIV  NAME    #IMPLIED  -- HTTP response header name  --
        NAME        NAME    #IMPLIED  -- metainformation name       --
        CONTENT     CDATA   #REQUIRED -- associated information     --
        >
]]>

<!ENTITY % head.content-1 "& LINK*">

<!ELEMENT LINK - O EMPTY>
<!ATTLIST LINK
        HREF %URI #REQUIRED
        %linkattributes-1; >

<![ %HTML.Proposed [
        <!ENTITY % font " TT | B | I | U | S ">
        <!ENTITY % phrase "EM | STRONG | CODE | SAMP | KBD | VAR | CITE
                                | STRIKE | DFN | KEY">
]]>

<!ENTITY % font " TT | B | I ">

<!ENTITY % phrase "EM | STRONG | CODE | SAMP | KBD | VAR | CITE ">



<!ENTITY % text "#PCDATA | IMG | BR | %phrase | %font">
<!ENTITY % htext "A | %text">
<!ENTITY % phrase.content "%htext">

<!ENTITY % pre.content "#PCDATA | A | %font | %phrase">

<!ELEMENT (%font;) - - (%phrase.content)+>

<!ELEMENT (%phrase;) - - (%phrase.content)+>

<!ENTITY % img.alt.default "#IMPLIED"
        -- ALT attribute optional in Level 1 docs -->

<!ENTITY % html-0 PUBLIC "+//ISBN 82-7640-037::WWW//DTD HTML Level 0//EN//2.0">
%html-0;

<!ATTLIST PRE
        WIDTH NUMBER #implied
        >
```

HTML DTD

This is the Document Type Definition for the HyperText Markup Language (HTML DTD):

```
<!--    html.dtd

         Document Type Definition for the HyperText Markup Language (HTML DTD)

         $Id: html-spec.ps,v 1.3 1994/08/01 15:50:32 karen Exp $

         Author: Daniel W. Connolly <connolly@hal.com>
         See Also: html.decl, html-0.dtd, html-1.dtd
                   http://www.hal.com/%7Econnolly/html-spec/index.html
                   http://info.cern.ch/hypertext/WWW/MarkUp2/MarkUp.html
 -->

 <!ENTITY % HTML.Version
         "+//ISBN 82-7640-037::WWW//DTD HTML//EN//2.0"

         -- Typical usage:

             <!DOCTYPE HTML PUBLIC "+//ISBN 82-7640-037::WWW//DTD HTML//EN">
             <html>
             ...
             </html>
         --
         >


 <!-- Imported Names -->

 <!ENTITY % Content-Type "CDATA"
         -- meaning a MIME content type, as per RFC1521
         -->

 <!ENTITY % HTTP-Method "GET | POST">


 <!-- Include level 1 DTD + block extensions -->

 <!ENTITY % block-2 "| FORM | ISINDEX">

 <!ENTITY % html-1 PUBLIC "+//ISBN 82-7640-037::WWW//DTD HTML Level 1//EN//2.0">
 %html-1;


 <!-- Forms  -->
```

```
<!ELEMENT FORM - - %body.content -(FORM) +(INPUT|SELECT|TEXTAREA)>
<!ATTLIST FORM
        ACTION %URI #REQUIRED
        METHOD (%HTTP-Method) GET
        ENCTYPE %Content-Type; "application/x-www-form-urlencoded"
        >

<!ENTITY % InputType "(TEXT | PASSWORD | CHECKBOX |
                        RADIO | SUBMIT | RESET |
                        IMAGE | HIDDEN )">
<!ELEMENT INPUT - O EMPTY>
<!ATTLIST INPUT
        TYPE %InputType TEXT
        NAME CDATA #IMPLIED -- required for all but submit and reset --
        VALUE CDATA #IMPLIED
        SRC %URI #IMPLIED -- for image inputs --
        CHECKED (CHECKED) #IMPLIED
        SIZE CDATA #IMPLIED -- like NUMBERS,
                                but delimited with comma, not space --
        MAXLENGTH NUMBER #IMPLIED
        ALIGN (top|middle|bottom) #IMPLIED
        >

<!ELEMENT SELECT - - (OPTION+)>
<!ATTLIST SELECT
        NAME CDATA #REQUIRED
        SIZE NUMBER #IMPLIED
        MULTIPLE (MULTIPLE) #IMPLIED
        >

<!ELEMENT OPTION - O (#PCDATA)>
<!ATTLIST OPTION
        SELECTED (SELECTED) #IMPLIED
        VALUE CDATA #IMPLIED
        >

<!ELEMENT TEXTAREA - - (#PCDATA)>
<!ATTLIST TEXTAREA
        NAME CDATA #REQUIRED
        ROWS NUMBER #REQUIRED
        COLS NUMBER #REQUIRED
        >
```

HTML DTD Reference (Level 2)

This HTML DTD Reference (Level 2) was generated from +//ISBN 82-7640-037::WWW//DTD HTML Prescriptive//EN//2.0

Alphabetical Index

A

Required Parts

<A>characters... </A>

All Parts

<A NAME="..." HREF="..." REL="..." REV="..." URN="..." TITLE="..." METHODS="..." >characters... <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </A>

Allowed In Content Of...

<ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2> <H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG> <TT> <VAR>

ADDRESS

Required Parts

<ADDRESS>characters... </ADDRESS>

All Parts

<ADDRESS>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </ADDRESS>

Allowed In Content Of...
<BLOCKQUOTE> <BODY> <FORM>

B

Required Parts
`<B>characters... </B>`

All Parts
`<B>characters...` <A> <IMG>
 <EM> <STRONG> <CODE>
<SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </B>

Allowed In Content Of...
<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2>
<H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG>
<TT> <V AR>

BASE

Required Parts
`<BASE HREF="..." >`

All Parts
`<BASE HREF="..." >`

Allowed In Content Of...
<HEAD>

BLOCKQUOTE

Required Parts
`<BLOCKQUOTE></BLOCKQUOTE>`

All Parts
`<BLOCKQUOTE>`<H1> <H2> <H3> <H4> <H5> <H6> <P> <UL> <OL> <DIR>
<MENU> <DL> <PRE> <BLOCKQUOTE> <FORM> <ISINDEX> <HR>
<ADDRESS> </BLOCKQUOTE>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

BODY

Required Parts

All Parts

<BODY><H1> <H2> <H3> <H4> <H5> <H6> <P> <UL> <OL> <DIR> <MENU> <DL> <PRE> <BLOCKQUOTE> <FORM> <ISINDEX> <HR> <ADDRESS> </BODY>

Allowed In Content Of...

<HTML>

BR

Required Parts

All Parts

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2> <H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <SAMP> <STRONG> <TT> <VAR>

CITE

Required Parts

<CITE>characters... </CITE>

All Parts

<CITE>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </CITE>

Allowed In Content Of...
<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2> <H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG> <TT> <V AR>

CODE

Required Parts
`<CODE>characters... </CODE>`

All Parts
`<CODE>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </CODE>

Allowed In Content Of...
<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2> <H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG> <TT> <V AR>

DD

Required Parts
`<DD>characters...`

All Parts
`<DD>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> <P> <UL> <OL> <DIR> <MENU> <DL> <PRE> <BLOCKQUOTE> <FORM> <ISINDEX> </DD>

Allowed In Content Of...
<DL>

DIR

Required Parts
`<DIR></DIR>`

All Parts
`<DIR>`<LI> </DIR>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

DL

Required Parts

<DL></DL>

All Parts

<DL COMPACT><DT> <DD> </DL>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

DT

Required Parts

<DT>characters...

All Parts

<DT>characters... <A> <IMG>
 <EM> <STRONG> <CODE>
<SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </DT>

Allowed In Content Of...

<DL>

EM

Required Parts

<EM>characters... </EM>

All Parts

<EM>characters... <A> <IMG>
 <EM> <STRONG> <CODE>
<SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </EM>

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2>
<H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG>
<TT> <V AR>

FORM

Required Parts

`<FORM ACTION="..." ></FORM>`

All Parts

`<FORM ACTION="..."  METHOD="..."  ENCTYPE="..." >`<H1> <H2> <H3> <H4> <H5> <H6> <P> <UL> <OL> <DIR> <MENU> <DL> <PRE> <BLOCKQUOTE> <FORM> <ISINDEX> <HR> <ADDRESS> </FORM>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

H1

Required Parts

`<H1>characters... </H1>`

All Parts

`<H1>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </H1>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <FORM>

H2

Required Parts

`<H2>characters... </H2>`

All Parts

`<H2>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </H2>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <FORM>

H3

Required Parts

`<H3>characters... </H3>`

All Parts

`<H3>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </H3>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <FORM>

H4

Required Parts

`<H4>characters... </H4>`

All Parts

`<H4>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </H4>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <FORM>

H5

Required Parts

`<H5>characters... </H5>`

All Parts

`<H5>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </H5>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <FORM>

H6

Required Parts

<H6>characters... </H6>

All Parts

<H6>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </H6>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <FORM>

HEAD

Required Parts

All Parts

<HEAD><TITLE> <ISINDEX> <BASE> <NEXTID> <LINK> </HEAD>

Allowed In Content Of...

<HTML>

HR

Required Parts

<HR>

All Parts

<HR>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <FORM>

HTML

Required Parts

All Parts

<HTML VERSION="..." ><HEAD> <BODY> </HTML>

Allowed In Content Of...

I

Required Parts

<I>characters... </I>

All Parts

<I>characters... <A> <IMG>
 <EM> <STRONG> <CODE>
<SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </I>

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2>
<H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG>
<TT> <V AR>

IMG

Required Parts

<IMG SRC="..." >

All Parts

<IMG SRC="..." ALT="..." ALIGN="..." ISMAP>

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2>
<H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <SAMP> <STRONG> <TT>
<VAR>

INPUT

Required Parts

<INPUT>

All Parts

```
<INPUT TYPE="..." NAME="..." VALUE="..." SRC="..." CHECKED
SIZE="..." MAXLENGTH="..." ALIGN="..." >
```

Allowed In Content Of...

ISINDEX

Required Parts

```
<ISINDEX>
```

All Parts

```
<ISINDEX>
```

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <FORM> <HEAD> <LI>

KBD

Required Parts

```
<KBD>characters... </KBD>
```

All Parts

`<KBD>characters...` <A> <IMG>
 <EM> <STRONG> <CODE>
<SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </KBD>

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2>
<H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG>
<TT> <V AR>

LI

Required Parts

```
<LI>characters...
```

All Parts

<LI>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> <P> <UL> <OL> <DIR> <MENU> <DL> <PRE> <BLOCKQUOTE> <FORM> <ISINDEX> </LI>

Allowed In Content Of...

<DIR> <MENU> <OL> <UL>

LINK

Required Parts

<LINK HREF="..." >

All Parts

<LINK HREF="..." REL="..." REV="..." URN="..." TITLE="..." METHODS="..." >

Allowed In Content Of...

<HEAD>

MENU

Required Parts

<MENU></MENU>

All Parts

<MENU><LI> </MENU>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

NEXTID

Required Parts

<NEXTID N="..." >

All Parts

<NEXTID N="..." >

Allowed In Content Of...
<HEAD>

OL

Required Parts
`<OL></OL>`

All Parts
`<OL>`<LI> </OL>

Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

OPTION

Required Parts
`<OPTION>characters...`

All Parts
`<OPTION SELECTED VALUE="..." >characters... </OPTION>`

Allowed In Content Of...
<SELECT>

P

Required Parts
`<P>characters...`

All Parts
`<P>characters...` <A> <IMG>
 <EM> <STRONG> <CODE>
<SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </P>

Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

PRE

Required Parts

<PRE>characters... </PRE>

All Parts

<PRE WIDTH="..." >characters... <A> <TT> <B> <I> <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> </PRE>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

SAMP

Required Parts

<SAMP>characters... </SAMP>

All Parts

<SAMP>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </SAMP>

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2> <H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG> <TT> <V AR>

SELECT

Required Parts

<SELECT NAME="..." ></SELECT>

All Parts

<SELECT NAME="..." SIZE="..." MULTIPLE><OPTION> </SELECT>

Allowed In Content Of...

STRONG

Required Parts

<STRONG>characters... </STRONG>

All Parts

<STRONG>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </STRONG>

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2> <H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG> <TT> <V AR>

TEXTAREA

Required Parts

<TEXTAREA NAME="..." ROWS="..." COLS="..." >characters... </TEXTAREA>

All Parts

<TEXTAREA NAME="..." ROWS="..." COLS="..." >characters... </TEXTAREA>

Allowed In Content Of...

TITLE

Required Parts

<TITLE>characters... </TITLE>

All Parts

<TITLE>characters... </TITLE>

Allowed In Content Of...

<HEAD>

TT

Required Parts

`<TT>characters... </TT>`

All Parts

`<TT>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </TT>

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2> <H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG> <TT> <V AR>

UL

Required Parts

`<UL></UL>`

All Parts

`<UL COMPACT>`<LI> </UL>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <FORM> <LI>

VAR

Required Parts

`<VAR>characters... </VAR>`

All Parts

`<VAR>characters...` <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <V AR> <CITE> <TT> <B> <I> </V AR>

Allowed In Content Of...

<A> <ADDRESS> <B> <CITE> <CODE> <DD> <DT> <EM> <H1> <H2> <H3> <H4> <H5> <H6> <I> <KBD> <LI> <P> <PRE> <SAMP> <STRONG> <TT> <V AR>

HTML DTD Reference (Level 0)

This HTML DTD Reference (Level 0) was generated from +//ISBN 82-7640-037::WWW//DTD HTML Level 0 Prescriptive//EN//2.0

Alphabetical Index

A

Required Parts
`<A>characters... </A>`

All Parts
`<A NAME="..." HREF="..." >characters...` <IMG>
 </A>

Allowed In Content Of...
<ADDRESS> <DD> <DT> <H1> <H2> <H3> <H4> <H5> <H6> <LI> <P> <PRE>

ADDRESS

Required Parts
`<ADDRESS>characters... </ADDRESS>`

All Parts
`<ADDRESS>characters...` <A> <IMG>
 </ADDRESS>

Allowed In Content Of...
<BLOCKQUOTE> <BODY>

BASE

Required Parts

`<BASE HREF="..." >`

All Parts

`<BASE HREF="..." >`

Allowed In Content Of...

<HEAD>

BLOCKQUOTE

Required Parts

`<BLOCKQUOTE></BLOCKQUOTE>`

All Parts

<BLOCKQUOTE><H1> <H2> <H3> <H4> <H5> <H6> <P> <UL> <OL> <DIR>
<MENU> <DL> <PRE> <BLOCKQUOTE> <HR> <ADDRESS>
</BLOCKQUOTE>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <LI>

BODY

Required Parts

All Parts

<BODY><H1> <H2> <H3> <H4> <H5> <H6> <P> <UL> <OL> <DIR>
<MENU> <DL> <PRE> <BLOCKQUOTE> <HR> <ADDRESS> </BODY>

Allowed In Content Of...

<HTML>

BR

Required Parts
`<BR>`

All Parts
`<BR>`

Allowed In Content Of...
<A> <ADDRESS> <DD> <DT> <H1> <H2> <H3> <H4> <H5> <H6> <LI> <P>

DD

Required Parts
`<DD>characters...`

All Parts
`<DD>characters...` <A> <IMG>
 <P> <UL> <OL> <DIR> <MENU> <DL> <PRE> <BLOCKQUOTE> </DD>

Allowed In Content Of...
<DL>

DIR

Required Parts
`<DIR></DIR>`

All Parts
`<DIR>`<LI> </DIR>

Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <LI>

DL

Required Parts
`<DL></DL>`

All Parts
`<DL COMPACT>`<DT> <DD> </DL>

Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <LI>

DT

Required Parts
`<DT>characters...`

All Parts
`<DT>characters...` <A> <IMG>
 </DT>

Allowed In Content Of...
<DL>

H1

Required Parts
`<H1>characters... </H1>`

All Parts
`<H1>characters...` <A> <IMG>
 </H1>

Allowed In Content Of...
<BLOCKQUOTE> <BODY>

H2

Required Parts
`<H2>characters... </H2>`

All Parts
`<H2>characters...` <A> <IMG>
 </H2>

Allowed In Content Of...
<BLOCKQUOTE> <BODY>

H3

Required Parts

`<H3>characters... </H3>`

All Parts

`<H3>characters...` <A> <IMG>
 </H3>

Allowed In Content Of...

<BLOCKQUOTE> <BODY>

H4

Required Parts

`<H4>characters... </H4>`

All Parts

`<H4>characters...` <A> <IMG>
 </H4>

Allowed In Content Of...

<BLOCKQUOTE> <BODY>

H5

Required Parts

`<H5>characters... </H5>`

All Parts

`<H5>characters...` <A> <IMG>
 </H5>

Allowed In Content Of...

<BLOCKQUOTE> <BODY>

H6

Required Parts

`<H6>characters... </H6>`

All Parts

`<H6>characters...` <A> <IMG>
 </H6>

Allowed In Content Of...
<BLOCKQUOTE> <BODY>

HEAD

Required Parts

All Parts
<HEAD><TITLE> <ISINDEX> <BASE> <NEXTID> </HEAD>

Allowed In Content Of...
<HTML>

HR

Required Parts
<HR>

All Parts
<HR>

Allowed In Content Of...
<BLOCKQUOTE> <BODY>

HTML

Required Parts

All Parts
<HTML VERSION="..." ><HEAD> <BODY> </HTML>

Allowed In Content Of...

IMG

Required Parts

```
<IMG SRC="..."  ALT="..." >
```

All Parts

```
<IMG SRC="..."  ALT="..."  ALIGN="..."  ISMAP>
```

Allowed In Content Of...

<A> <ADDRESS> <DD> <DT> <H1> <H2> <H3> <H4> <H5> <H6> <LI> <P>

ISINDEX

Required Parts

```
<ISINDEX>
```

All Parts

```
<ISINDEX>
```

Allowed In Content Of...

<HEAD>

LI

Required Parts

```
<LI>characters...
```

All Parts

`<LI>characters...` <A> <IMG>
 <P> <UL> <OL> <DIR> <MENU>
<DL> <PRE> <BLOCKQUOTE> </LI>

Allowed In Content Of...

<DIR> <MENU> <OL> <UL>

MENU

Required Parts

```
<MENU></MENU>
```

All Parts
<MENU><LI> </MENU>

Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <LI>

NEXTID

Required Parts
`<NEXTID N="..." >`

All Parts
`<NEXTID N="..." >`

Allowed In Content Of...
<HEAD>

OL

Required Parts
`<OL></OL>`

All Parts
`<OL>`<LI> </OL>

Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <LI>

P

Required Parts
`<P>characters...`

All Parts
`<P>characters...` <A> <IMG>
 </P>

Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <LI>

PRE

Required Parts

`<PRE>characters... </PRE>`

All Parts

`<PRE>characters...` <A> </PRE>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <LI>

TITLE

Required Parts

`<TITLE>characters... </TITLE>`

All Parts

`<TITLE>characters... </TITLE>`

Allowed In Content Of...

<HEAD>

UL

Required Parts

`<UL></UL>`

All Parts

`<UL COMPACT>`<LI> </UL>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <DD> <LI>

Terms

The HTML specification uses these words with precise meanings:

attribute
: A characteristic quality of an element, other than type or content.

browser
: A tool used to read electronic books.

document
: For the purposes of this standard, an HTML instance.

element
: A component of the hierarchical structure defined by the document type definition; it is identified in a document instance by descriptive markup, usually a start-tag and end-tag.

HTML
: HyperT ext Markup Language.

HTTP
: A generic stateless object-oriented protocol, which may be used for many similar tasks by extending the commands, or "methods", used. For example, you might use HTTP for name servers and distributed object-oriented systems, With HTTP , the negotiation of data representation allows systems to be built independent of the development of new representations.

  For more information see http://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTTP2.html.

markup
: Text that is added to the data of a document in order to convey information about it. There are four different kinds of markup: descriptive markup (tags), references, markup declarations, and processing instructions.

MIME
: Multipurpose Internet Mail Extensions as defined in *Mechanisms for Specifying and Describing the Format of Internet Message Bodies*, 09/23/1993. (Pages=81) (Format=.txt, .ps) (Obsoletes RFC1341) (Updated by RFC1590).

representation
: The encoding of information for interchange. For example, HTML is a representation of hypertext.

rendering
: The form of presentation to information to the human reader.

SGML
: Standard Generalized Markup Language as defined in ISO 8879:1986, Information Processing Text and Office Systems.

sgmls
: An SGML parser by James Clark, jjc@jclark.com, derived from the ARCSGML parser materials which were written by Charles F. Goldfarb. The source is available on the ftp.ifi.uio.no FTP server in the directory /pub/SGML/SGMLS.

tag
: Descriptive markup. There are two kinds of tags: start-tag and end-tag.

URI
: Universal Resource Identifiers. Currently available by anonymous FTP from info.cern.ch in /pub/www/doc/url*.*{ps,txt}*

W3
: The World-Wide Web, a global information initiative. For bootstrap information, telnet info.cern.ch or find documents by ftp://info.cern.ch/pub/www/doc

Imperatives

may
: The implementation is not obliged to follow this in any way.

must
: If this is not followed, the implementation does not conform to this specification.

shall
: If this is not followed, the implementation does not conform to this specification.

should
: If this is not followed, though the implementation officially conforms to the standard, undesirable results may occur in practice.

typical
: Typical rendering is described for many elements. This is not a mandatory part of the standard but is given as guidance for designers and to help explain the uses for which the elements were intended.

References

The HTML specification cites these works:

HTTP

HTTP: A Protocol for Networked Information. This document is available at http://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTTP2.html.

MIME

N. Borenstein, N. Freed, *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies*, 09/23/1993. (Pages=81) (Format=.txt, .ps) (Obsoletes RFC1341) (Updated by RFC1590).

SGML

ISO 8879:1986, Information Processing Text and Office Systems Standard Generalized Markup Language (SGML).

sgmls

An SGML parser by James Clark, jjc@jclark.com, derived from the ARCSGML parser materials which were written by Charles F. Goldfarb. The source is available on the ftp.ifi.uio.no FTP server in the directory /pub/SGML/SGMLS.

URI

Universal Resource Identifiers. RFCxxx. Currently available by anonymous FTP from info.cern.ch in /pub/www/doc/url*.*{ps, txt}*

W3

The World-Wide Web, a global information initiative. For bootstrap information, telnet info.cern.ch or find documents by ftp://info.cern.ch/pub/www/doc.

Acknowledgments

The HTML document type was designed by Tim Berners-Lee at CERN as part of the 1990 World-Wide Web project. In 1992, Dan Connolly wrote the HTML Document Type Definition (DTD) and a brief HTML specification.

Since 1993, a wide variety of Internet participants have contributed to the evolution of HTML, which has included the addition of in-line images introduced by the NCSA Mosaic software for WWW. Dave Raggett played an important role in deriving the FORMS material from the HTML+ specification.

Dan Connolly and Karen Olson Muldrow rewrote the HTML Specification in 1994.

Special thanks to the many people who have contributed to this specification:

- Terry Allen; O'Reilly & Associates; terry@ora.com
- Marc Andresson; Mosaic Communications Corp; marca@mcom.com
- Tim Berners-Lee; CERN; timbl@info.cern.ch
- Paul Burchard; The Geometry Center, University of Minnesota; burchard@geom.umn.edu
- James Clark
- Daniel W. Connolly; HaL Computer Systems; connolly@hal.com
- Jay Glicksman; Enterprise Integration Technology; jay@eit.com
- Eduardo Gutentag; Sun Microsystems; eduardo@Eng.Sun.com
- Bill Hefley; Software Engineering Institute, Carnegie Mellon University; weh@sei.cmu.edu
- Chung-Jen Ho; Xerox Corporation; cho@xsoft.xerox.com
- Murray Maloney; SCO Canada; murray@sco.com
- Karen Olson Muldrow; HaL Computer Systems; karen@hal.com
- Erik Naggum
- William Perry
- Dave Raggett

- E. Corprew Reed; Cold Spring Harbor Laboratory; corp@cshl.org
- Yuri Rubinsky; SoftQuad, Inc.; yuri@sq.com
- Eric W . Sink; Spyglass, Inc.; eric@spyglass.com
- Stuart W   eibel; OCLC Office of Research; weibel@oclc.org