HyperText Markup Language Working Group
Internet-Draft

T. Berners-Lee, CERN
D. Connolly, HAL
November 28, 1994

HYPERTEXT MARKUP LANGUAGE SPECIFICATION - 2.0

STATUS OF THIS MEMO

This document is an Internet draft. Internet drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet drafts.

Internet drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet drafts as reference material or to cite them other than as "work in progress."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet- Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited. Please send comments to the HTML working group (HTML-WG) of the Internet Engineering Task Force (IETF) at <html-wg@oclc.org>. Discussions of the group are archived at URL: http://www.acl.lanl.gov/HTML_WG/archives.html.

ABSTRACT

HyperText Markup Language (HTML) is a simple markup language used to create hypertext documents that are portable from one platform to another. 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 in-lined graphics; and hypertext views of existing bodies of information.

HTML has been in use by the World Wide Web (WWW) global information initiative since 1990. This specification corresponds to the legitimate capabilities of HTML in common use prior to June 1994. It is defined as an application of ISO Standard 8879:1986 Information Processing Text and Office Systems; Standard Generalized Markup Language (SGML). This specificiation is proposed as the Internet Media Type (RFC 1590) and MIME Content Type (RFC 1521) called "text/html", or "text/html; version=2.0".

CONTENTS

1. Overview of HTML Specification

This chapter is a summary of the HTML specification. See Section 2. for the complete specification.

HTML describes the structure and organization of a document. It only suggests appropriate presentations of the document when processed.

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 >.

Example:

```
<H1>This is a heading</H1>
```

Every HTML document starts with a HTML document identifier which contains two sections, a head and a body. The head contains HTML elements which describe the documents title, usage and relationship with other documents. The body contains other HTML elements with the entire text and graphics of the document.

This overview briefly describes the syntax of HTML elements and provides an example HTML document.

> NOTE: The term “HTML user agent” is used in this document to describe applications that are used with HTML documents.

1.1 HTML Elements

1.1.1 Document Structure Elements

HTML Identifier

<HTML> … </HTML>

The HTML identifier defines the document as containing HTML elements. It contains only the Head and Body elements.

Head

<HEAD> … </HEAD>

The Head element contains HTML elements that describe the documents title, usage and relationship with other documents.

Body

<BODY> … </BODY>

The Body element contains the text and its associated HTML elements of the document.

Example of Document Structure Elements

```
<HTML>
<HEAD>
<TITLE>The Document's Title</TITLE>
</HEAD>
<BODY>
The document's text.
</BODY>
```

1.1.2 Anchor Element

Anchor

<A> … </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 <A HREF=”http://www.hal.com/”>HaL</A>’s
information for more details.

<A NAME=”B”>Section B</A> describes…
…
See <A HREF=”#B”>Section B</A> for more information.
```

1.1.3 Block Formatting Elements

Address

<ADDRESS> … </ADDRESS>

```
<ADDRESS>
Newsletter editor<BR>
```

```
J.R. Brown<BR>
JimquickPost News, Jumquick, CT 01234<BR>
Tel (123) 456 7890
</ADDRESS>
```

Body

<BODY> … </BODY>

Place the <BODY> and </BODY> tags above and below the body of the text (not including the head) of your HTML document.

Blockquote

<BLOCKQUOTE>… </BLOCKQUOTE>

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

Head

<HEAD> … </HEAD>

Every HTML document must have a head, which provides a title. Example:

```
<HTML>
<HEAD>
<TITLE>Introduction to HTML</TITLE>
</HEAD>
```

Headings

```
<H1>This is a first level heading</H1>
<P>There are six levels of headings.
<H2>Second level heading</H2>
<P>This text appears under the second level heading
```

Horizontal Rule

<HR>

Inserts a horizontal rule that spans the width of the document. Example:

```
<HR>
<ADDRESS>November 29, 1994, CERN</ADDRESS>
</BODY>
```

HTML Identifier

<HTML> … </HTML>

An HTML document begins with an <HTML> tag and ends with the </HTML> tag.

Line Break

Forces a line break:

```
Name<BR>
Street address<BR>
City, State Zip
```

Paragraph

<P> … </P>

```
<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>
```

Title

<TITLE> … </TITLE>

```
<TITLE>Title of document</TITLE>
```

1.1.4 List Elements

Definition List

<DL> … <DT>*term*<DD>*definition*… </DL>

```
<DL>
<DT>Term<DD>This is the first definition.
```

```
<DT>Term<DD>This is the second definition.
</DL>
```

Directory List

<DIR> … <LI>*List item*… </DIR>

```
<DIR>
<LI>A-H<LI>I-M
<LI>M-R<LI>S-Z
</DIR>
```

Menu List

<MENU> … <LI>*List item*… </MENU>

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

Ordered List

<OL> … <LI>*List item*… </OL>

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

Unordered List

<UL> … <LI>*List item*… </UL>

```
<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>
```

1.1.5 Information Type and Character Formatting Elements

Bold

<B> … </B>

Suggests the rendering of the text in **boldface**. If boldface is not available, alternative mapping is allowed.

Citation

<CITE> … </CITE>

Specifies a citation; typically rendered as italic.

Code

<CODE> … </CODE>

Indicates an inline example of code; typically rendered as monospaced.. Do not confuse with the <PRE> tag.

Emphasis

<EM> … </EM>

Provides typographic emphasis; typically rendered as italics.

Italics

<I> … </I>

Suggests the rendering of text in *italic* font, or slanted if italic is not available.

Keyboard

<KBD> … </KBD>

Indicates text typed by a user; typically rendered as monospaced.

Sample

<SAMP> … </SAMP>

Indicates a sequence of literal characters; typically rendered as monospaced..

Strong

<STRONG> … </STRONG>

Provides strong typographic emphasis; typically rendered as bold.

Typetype

<TT> … </TT>

Specifies that the text be rendered in fixed-width font.

Variable

<VAR> … </VAR>

Indicates a variable name; typically rendered as italic.

1.1.6 Image Element

Image

<IMG>

Inserts the referenced graphic image into the document at the location where the element occurs.

Example:

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

1.1.7 Form Elements

Form

<FORM> … </FORM>

The Form element contains nested elements (described below) which define user input controls and allow descriptive text to be displayed when the document is processed.

Input

<INPUT>

Takes these attributes: ALIGN, MAXLENGTH, NAME, SIZE, SRC, TYPE, VALUE. The type attribute can define these field types: CHECKBOX, HIDDEN, IMAGE, PASSWORD, 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="gender" TYPE=RADIO VALUE="male">
<P>Female <INPUT NAME="gender" TYPE=RADIO VALUE="female">
```

```
</FORM>
```

Option

<OPTION>

The Option element can only occur within a Select element. It represents one choice.

Select

<SELECT NAME=”…” > … </SELECT>

Select provides a list of choices.

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

Textarea

<TEXTAREA> … </TEXTAREA>

Textarea defines a multi-line text entry input control. It contains the initial text contents of the control.

```
<TEXTAREA NAME="address" ROWS=64 COLS=6>
HaL Computer Systems
1314 Dell Avenue
Campbell California 95008
</TEXTAREA>
```

1.1.8 Character Data in HTML

Representing Graphic Characters in HTML

Because of the way special characters are used in marking up HTML text, character strings are used to represent the less than (<) and greater than (>) symbols and the ampersand (&) as shown in Section 2.17.1.

Representing ISO Latin-1 Characters in HTML

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 Standard 8879:1986//ENTITIES Added Latin 1//EN. For details, see 2.17.2.

1.2 Example HTML Document

```
<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>
```

2. HTML SPECIFICATION

HTML has been in use by the World Wide Web (WWW) global information initiative since 1990. This specification corresponds to the legitimate capabilities of HTML in common use prior to June 1994. It is defined as an application of ISO Standard 8879:1986: Standard Generalized Markup Language (SGML). This specification is proposed as the Internet Media Type (RFC 1590) and MIME Content Type (RFC 1521) called "text/html", or "text/html; version=2.0".

This specification also includes:

- 5.1 SGML Declaration for HTML
- 5.1.1 Sample SGML Open Style Entity Catalog for HTML
- 5.2 HTML DTD

This specification is currently available on the World Wide Web at URL: http://www.hal.com/%7Econnolly/html-spec

Please send comments to the discussion list at: html-wg@oclc.org

2.1 Levels of Conformance

Version 2.0 of the HTML specification introduces forms for user input of information, and adds a distinction between levels of conformance:

Level 0

Indicates the minimum conformance level. When writing Level 0 documents, authors can be confident 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, such as tables, figures, and mathematical formulae, are under discussion and are described as proposed where mentioned.

2.2 Undefined Tag and Attribute Names

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

The behavior of HTML user agents 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.

2.3 Deprecated and Recommended Sections in DTDs

In Section 5., optional "deprecated" and "recommended" sections are used. Conformance with this specification is defined with these sections disabled. In the liberal spirit of Section 2.2, HTML user agents reading HTML documents should accept syntax corresponding to the specification with "deprecated" turned on. HTML user agents generating HTML may in the spirit of conservation, generate documents that conform to the specification with the "recommended" sections turned on.

2.4 HTML and MIME

The World Wide Web initiative (WWW) links information throughout the world. To do this, WWW 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 WWW, and is proposed as a MIME content type. The definition of the HTML Content-Type is text/html, and has three optional parameters:

Level

The level parameter specifies the feature set 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. See Section 2.16 for a discussion of character sets and encodings in HTML.

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. There is no obligation for an HTML document to contain any characters above decimal 127. It is possible that a transport medium such as electronic mail imposes constraints on the number of bits in a representation of a document, though the HTTP access protocol used by WWW always allows 8 bit transfer.

When an HTML document is encoded using 7-bit characters, then the mechanisms of numeric character references (see Section 2.16.2) and character entity references (see Section 2.16.3) may be used to encode characters in the upper half of the ISO 8859/1 Latin-1 set. In this way, documents may be prepared which are suitable for mailing through 7-bit limited systems.

> NOTE: ISO 646 is, for all intents and purposes, equivalent to the ANSI standard for ASCII (American Standard Code for Information Interchange). The only notable differences between the two standards are the names assigned to the control characters that occupy positions 00 through 31 and position 127 (decimal) in that encoding. For encoding HTML documents, only three control characters in ISO 646 or ASCII are relevant (see Section 2.16.2). These are Carriage Return (CR) at position 13, Line Feed (LF) at position 10, and Horizontal Tab (HT) at position 11.

2.5 Understanding HTML and SGML

HTML is an application of ISO Standard 8879:1986 — 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 Section 5.1. It is implicit among HTML user agents.

If the HTML specification and SGML standard conflict, the SGML standard is definitive.

Every SGML document has three parts:

SGML declaration

Binds SGML processing quantities and syntax token names to specific values. For 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.

Prologue

Includes one or more document type declarations, which specify the element types, element relationships and attributes.

Instance

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.

2.6 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 "-//IETF//DTD HTML//EN//2.0">
```

> NOTE: If the body of a text/html body part does not begin with a document type declaration, an HTML user agent should infer the above document type declaration.

HTML documents should also contain an <HTML> tag at the beginning of the file, after the prologue, and an </HTML> tag 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.

> 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, such as the title, without parsing the entire document.

2.6.1 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 >.

Example:

```
<H1>This is a Heading</H1>
```

Some elements only have a start tag without an end tag. For example, to create a line break, you use the
 tag. Additionally, the end tags of some other elements, such as Paragraph (<P>), List Item (<LI>), Definition Term (<DT>), and Definition Description (<DD>) elements, may be omitted.

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 other valid syntaxes for tags, such as NET tags, <EM/…/; empty start tags, <>; and empty end tags, </>. Until support for these idioms is widely deployed, their use is strongly discouraged.

2.6.2 Names

A name consists of a letter followed by up to 71 letters, digits, periods, or hyphens. Element names are not case sensitive, but entity names are. For example, <BLOCKQUOTE>, <BlockQuote>, and <blockquote> are equivalent, whereas & is different from &.

In a start tag, the element name must immediately follow the tag open delimiter <.

2.6.3 Attributes

In a start tag, white space 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). White space is allowed around the equal sign.

The value of the attribute may be either:

- A string literal, delimited by single quotes or double quotes and not containing any occurrences of the delimiting character.

- 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">
```

> NOTE: Some non-SGML 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 (see Section 2.17.1), such as: `<IMG SRC="eq1.ps" alt="a > b">`

To put quotes inside of quotes, you may use the character representation *"* as in:

```
<IMG SRC="image.ps" alt="First &quot;real&quot; example">
```

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

> NOTE: Some non-SGML 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, such as ISMAP and COMPACT, may be written using a minimized syntax. The markup:

```
<UL COMPACT="compact">
```

can be written using a minimized syntax:

```
<UL COMPACT>
```

> NOTE: Some non-SGML implementations only understand the minimized syntax.

2.6.4 *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. For more information, see Section 2.16.

2.6.5 Comments

To include comments in an HTML document that will be ignored by the HTML user agent, surround them with <!-- and -->. After the comment delimiter, all text up to the next occurrence of --> is ignored. Hence comments cannot be nested. White space 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 HTML user agents incorrectly consider a > sign to terminate a comment.

2.7 The Head Element and Related Elements

Only certain elements are allowed in the head of an HTML document. Elements that may be included in the head of a document are:

2.7.1 Head

<HEAD> … </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>
```

2.7.2 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 HTML user agent uses the URL it used to access the document to resolve any relative URLs.

The Base element has one attribute, HREF, which identifies the URL.

2.7.3 Isindex

Level 0

The Isindex element tells the HTML user agent 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.

> NOTE: The Isindex element is usually generated automatically by a server. If added manually to an HTML document, the HTML user agent assumes that the server can handle a search on the document. To use the Isindex element, the server must have a search engine that supports this element.

2.7.4 Link

Level 1

The Link element indicates a relationship between the document and some other object. A document may have any number of Link elements.

The Link element is empty (does not have a closing tag), 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.

2.7.5 Nextid

Level 0

The Nextid element is a parameter read by and generated by text editing software to create unique identifiers. This tag takes a single attribute which is the next document-wide alpha-numeric identifier to be allocated of the form z123:

```
<NEXTID N=Z27>
```

When modifying a document, existing anchor identifiers should not be reused, as these identifiers may be referenced by other documents. Human writers of HTML usually use mnemonic alphabetical identifiers.

HTML user agents may ignore the Nextid element. Support for the Nextid element does not impact HTML user agents in any way.

2.7.6 Title

<TITLE> … </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. Unlike headings, titles are not typically rendered 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. Only one title is allowed in a document.

> NOTE: The length of a title is not limited; however, long titles may be truncated in some applications. To minimize this possibility, titles should be fewer than 64 characters. Also keep in mind that a short title, such as *Introduction*, may be meaningless out of context. An example of a meaningful title might be *"Introduction to HTML Elements."*

2.7.7 Meta

Level 1

The Meta element is 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 that have well-defined semantics for each type of meta-information, such as a 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 that should be included in the response headers for an HTTP request.

Attributes of the Meta element:

HTTP-EQUIV

This attribute binds the element to an HTTP response header. If the semantics of the HTTP response header named by this attribute is known, then the contents can be processed based on a well-defined syntactic mapping whether or not the DTD includes anything about it. HTTP header names are *not* case sensitive. If not present, the NAME attribute should be used to identify this meta-information and it should not be used within an HTTP response header.

NAME

Meta-information name. If the NAME attribute is 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:

```
<META HTTP-EQUIV="Expires" CONTENT="Tue, 04 Dec 1993 21:29:02 GMT">
<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 existing HTML element.

Example of 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 typically only be generated by the HTTP server. Some inappropriate names are “Server”, “Date”, and “Last-modified”. Whether a name is inappropriate depends on the particular server implementation. It is recommended that servers ignore any Meta elements that specify HTTP-equivalents equal (case-insensitively) to their own reserved response headers.

2.8 The Body Element and Related Elements

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

2.8.1 Body

<BODY> … </BODY>

Level 0

The Body element identifies the body component of an HTML document. Specifically, the body of a document may contain links, text, and formatting information within <BODY> and </BODY> tags.

2.8.2 Address

<ADDRESS> … </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 is rendered in an italic typeface and may be indented. The Address element implies a paragraph break before and after.

Example of use:

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

2.8.3 Anchor

<A> … </A>

Level 0

An anchor is a marked text that is the start and/or destination of a hypertext link. Anchor elements are defined by the <A> tag. The <A> tag accepts several attributes, but either the NAME or HREF attribute is required.

Attributes of the <A> tag:

HREF

Level 0

If the HREF attribute is present, the text between the opening and closing anchor tags becomes hypertext. If this hypertext is selected by readers, they are moved to another document, or to a different location in the current document, whose network address is defined by the value of the HREF attribute.

Example:

```
See <A HREF=”http://www.hal.com/”>HaL</A>’s information for more details.
```

In this example, selecting “HaL” takes the reader to a document at http://www.hal.com. The format of the network address is specified in the URI specification for print readers.

With the HREF attribute, the form HREF=”#identifier” can refer to another anchor in the same document.

Example:

```
The <A HREF=”document.html#glossary”>glossary</A> defines terms used in this
document.
```

In this example, selecting “glossary” takes the reader to another anchor (i.e., <A NAME=”glossary”>Glossary</A>) in the same document (document.html). The NAME attribute is described below. If the anchor is in another document, the HREF attribute may be relative to the document’s address or the specified base address (see 2.7.2 Base).

NAME

Level 0

If present, the NAME attribute allows the anchor to be the target of a link. The value of the NAME attribute is an identifier for the anchor. Identifiers are arbitrary strings but must be unique within the HTML document.

Example of use:

```
<A NAME=”coffee”>Coffee</A> is an example of …
…
```

```
An example of this is <A HREF="#coffee">coffee</A>.
```

Another document can then make a reference explicitly to this anchor by putting the identifier after the address, separated by a hash sign:

```
<A NAME="drinks.html#coffee">
```

TITLE

Level 1

The TITLE attribute is informational only. If present, the TITLE attribute should provide the title of the document whose address is given by the HREF attribute.

The TITLE attribute is useful for at least two reasons. The HTML user agent may 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 while the document is being loaded. Another reason is that documents that are not marked up text, such as graphics, plain text and Gopher menus, do not have titles. The TITLE attribute can be used to provide a title to such documents. When using the TITLE attribute, the title should be valid and unique for the destination document.

REL

Level 1

The REL attribute gives the relationship(s) described by the hypertext link from the anchor to the target. 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. The REL attribute is only used when the HREF attribute is present.

REV

Level 1

The REV attribute is the same as the REL attribute, 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.

URN

Level 1

If present, the URN attribute specifies a uniform resource name (URN) for a target document. The format of URNs is under discussion (1994) by various working groups of the Internet Engineering Task Force.

METHODS

The METHODS attributes of anchors and links provide information about the functions that 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 HTML user agent may chose a different rendering as a function of the methods allowed; for example, something that 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.

See also: 2.7.4 Link

2.8.4 Blockquote

<BLOCKQUOTE> … </BLOCKQUOTE>

Level 0

The Blockquote element is used to contain text quoted from another source.

A typical rendering might be a slight extra left and right indent, and/or italic font. The Blockquote element causes a paragraph break, and typically provides space above and below 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 the left margin.

Example of use:

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

2.8.5 Headings

<H1> … </H1> through <H6> … </H6>

Level 0

HTML defines six levels of heading. A Heading element implies all the font changes, paragraph breaks before and after, and white space necessary to render the heading.

The highest level of headings is H1, followed by H2 … H6.

Example of use:

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

The rendering of headings is determined by the HTML user agent, but typical renderings are:

<H1> … </H1>

Bold, very-large font, centered. One or two blank lines above and below.

<H2> … </H2>

Bold, large font, flush-left. One or two blank lines above and below.

<H3> … </H3>

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

<H4> … </H4>

Bold, normal font, indented more than H3. One blank line above and below.

<H5> … </H5>

Italic, normal font, indented as H4. One blank line above.

<H6> … </H6>

Bold, indented same as normal text, more than H5. One blank line above.

Although heading levels can be skipped (for example, from H1 to H3), this practice is discouraged as skipping heading levels may produce unpredictable results when generating other representations from HTML.

2.9 Overview of Character-Level Elements

Level 2 (all elements)

Character-level elements are used to specify either the logical meaning or the physical appearance of marked text without causing a paragraph break. Like most other elements, character-level elements include both opening and closing tags. Only the characters between the tags are affected:

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

Character-level tags may be ignored by minimal HTML applications.

Character-level tags are interpreted from left to right as they appear in the flow of text. Level 1 HTML user agents 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.

Character-level elements may be nested within the content of other character-level elements; however, HTML user agents are not required to render nested character-level elements distinctly from non-nested elements:

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

may the rendered the same as

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

Note that typical renderings for information type elements vary between applications. If a specific rendering is necessary, for example, when referring to a specific text attribute as in “The *italic* parts are mandatory”, a formating element can be used to ensure that the intended rendered is used where possible.

2.10 Information Type Elements

Note that different information type elements may be rendered in the same way.

2.10.1 Citation

<CITE>…</CITE>

The Citation element specifies a citation; typically rendered as italics.

2.10.2 Code

<CODE> … </CODE>

The Code element indicates an example of code; typically rendered as monospaced . Do not confuse with the Preformatted Text element.

2.10.3 Emphasis

<EM> … </EM>

The Emphasis element indicates typographic emphasis, typically rendered as italics.

2.10.4 Keyboard

<KBD> … </KBD>

The Keyboard element indicates text typed by a user; typically rendered as monospaced . It might commonly be used in an instruction manual.

2.10.5 Sample

<SAMP> … </SAMP>

The Sample element indicates a sequence of literal characters; typically rendered as monospaced.

2.10.6 Strong

<STRONG> … </STRONG>

The Strong element indicates strong typographic emphasis, typically rendered in bold.

2.10.7 Variable

<VAR> … </VAR>

The Variable element indicates a variable name; typically rendered as italic.

2.11 Character Format Elements

Character format elements are used to specify the format of marked text. Example of use:

2.11.1 Bold

<B> … </B>

The Bold element specifies that the text should be rendered in **boldface**, where available. Otherwise, alternative mapping is allowed.

2.11.2 Italic

<I> … </I>

The Italic element specifies that the text should be rendered in *italic* font, where available. Otherwise, alternative mapping is allowed.

2.11.3 Teletype

<TT> ... </TT>

The Teletype element specifies that the text should be rendered in fixed-width typewriter font.

2.12 Image Element

<IMG>

Level 0

The Image element is used to incorporate in-line graphics (typically icons or small graphics) into an HTML document. This element *cannot* be used for embedding other HTML text.

HTML user agents that cannot render in-line images ignore the Image element unless it contains the ALT attribute. Note that some HTML user agents can render linked graphics but not in-line graphics. If a graphic is essential, you may want to create a link to it rather than to put it in-line. If the graphic is not essential, then the Image element is appropriate.

The Image element, which is empty (no closing tag), has these attributes:

ALIGN

The ALIGN attribute accepts the values TOP or MIDDLE or BOTTOM, which specifies if the following line of text is aligned with the top, middle, or bottom of the graphic.

ALT

Optional text as an alternative to the graphic for rendering in non-graphical environments. Alternate text should be provided whenever the graphic is not rendered. Alternate text is mandatory for Level 0 documents. Example of use:

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

ISMAP

The ISMAP (is map) attribute identifies an image as an image map. Image maps are graphics in which certain regions are mapped to URLs. By clicking on different regions, different resources can be accessed from the same graphic. Example of use:

```
<A HREF="http://machine/htbin/imagemap/sample">
<IMG SRC="sample.gif" ISMAP>
</A>
```

SRC

The value of the SRC attribute is the URL of the document to be embedded; only images can be embedded, not HTML text. Its syntax is the same as that of the HREF attribute of the <A> tag. SRC is mandatory. Image elements are allowed within anchors.

Example of use:

```
<IMG SRC ="triangle.gif">Be sure to read these instructions.
```

2.13 List Elements

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

2.13.1 Definition List

<DL> … </DL>

Level 0

A definition list is a list of terms and corresponding definitions. Definition lists are typically formatted with the term flush-left and the definition, formatted paragraph style, indented after the term.

Example of use:

```
<DL>
<DT>Term<DD>This is the definition of the first term.
<DT>Term<DD>This is the definition of the second term.
</DL>
```

If the DT term does not fit in the DT column (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.

Single occurrences of a <DT> tag without a subsequent <DD> tag are allowed, and have the same significance as if the <DD> tag had been present with no text.

The opening list tag must be <DL> and must be immediately followed by the first term (<DT>).

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

Unless you provide the COMPACT attribute, the HTML user agent may leave white space between successive DT, DD pairs.The COMPACT attribute may also reduce the width of the left-hand (DT) column.

If using the COMPACT attribute, the opening list tag must be <DL COMPACT>, which must be immediately followed by the first <DT> tag:

```
<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>
```

2.13.2 Directory List

<DIR> … </DIR>

Level 0

A Directory List element is used to present a list of items containing up to 20 characters each. Items in a directory list may be arranged in columns, typically 24 characters wide. If the HTML user agent can optimize the column width as function of the widths of individual elements, so much the better.

A directory list must begin with the <DIR> tag which is immediately followed by a <LI> (list item) tag:

```
<DIR>
<LI>A-H<LI>I-M
<LI>M-R<LI>S-Z
</DIR>
```

2.13.3 Menu List

<MENU> … </MENU>

Level 0

A menu list 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.

A menu list must begin with a <MENU> tag which is immediately followed by a <LI> (list item) tag:

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

2.13.4 Ordered List

<OL> … </OL>

Level 0

The Ordered List element is used to present a numbered list of items, sorted by sequence or order of importance.

An ordered list must begin with the <OL> tag which is immediately followed by a <LI> (list item) tag:

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

The Ordered List element can take the COMPACT attribute, which suggests that a compact rendering be used.

2.13.5 Unordered List

<UL> … </UL>

Level 0

The Unordered List element is used to present a list of items which is typically separated by white space and/or marked by bullets.

An unordered list must begin with the <UL> tag which is immediately followed by a <LI> (list item) tag:

```
<UL>
<LI>First list item
<LI>Second list item
<LI>Third list item
</UL>
```

2.14 Other Elements

2.14.1 Paragraph

<P>

Level 0

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

Typically, paragraphs are surrounded by a vertical space of one line or half a line. This is typically not the case within the Address element and or is never the case within the Preformatted Text element. With some HTML user agents, the first line in a paragraph is indented.

Example of use:

```
<H1>This Heading Precedes the Paragraph</H1>
<P>This is the text of the first paragraph.
<P>This is the text of the second paragraph. Although you
do not need to start paragraphs on new lines, maintaining
this convention facilitates document maintenance.
<P>This is the text of a third paragraph.
```

2.14.2 Preformatted Text

<PRE> … </PRE>

Level 0

The Preformatted Text element presents blocks of text in fixed-width font, and so is suitable for text that has been formatted on screen.

The <PRE> tag 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 HTML user agent to select a suitable font and indentation. If the WIDTH attribute is not present, a width of 80 characters is assumed. Where the WIDTH attribute is supported, widths of 40, 80 and 132 characters should be presented optimally, with other widths being rounded up.

Within preformatted text:

- Line breaks within the text are rendered as a move to the beginning of the next line.

- 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 that define paragraph formatting (headings, address, etc.) must *not* be used.
- The ASCII horizontal tab 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 be constrained by the width required.

Example of use:

```
<PRE WIDTH=”80”>
This is an example line.
</PRE>
```

NOTE: Within a Preformatted Text element, the constraint that the rendering must be on a fixed horizontal character pitch may limit or prevent the ability of the HTML user agent to render highlighting elements specially.

2.14.3 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.

Example of use:

```
<P>
Pease porridge hot<BR>
Pease porridge cold<BR>
Pease porridge in the pot<BR>
Nine days old.
```

2.14.4 Horizontal Rule

<HR>

Level 0

A Horizontal Rule element is a divider between sections of text such as a full width horizontal rule or equivalent graphic.

Example of use:

```
<HR>
<ADDRESS>November 29, 1994, CERN</ADDRESS>
</BODY>
```

2.15 Form Elements

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

The following elements (all are HTML 2 features) are used to create forms:

FORM

A form within a document.

INPUT

One input field.

OPTION

One option within a Select element.

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 of use (a questionnaire form):

```
<H1>Sample Questionnaire</H1>
<P>Please fill out this questionnaire:
<FORM METHOD="POST" ACTION="http://www.hal.com/sample">
<P>Your name: <INPUT NAME="name" size="48">
<P>Male <INPUT NAME="gender" TYPE=RADIO VALUE="male">
<P>Female <INPUT NAME="gender" TYPE=RADIO VALUE="female">
<P>Number in family: <INPUT NAME="family" TYPE=text>
<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 questionnaire.
<P><INPUT TYPE=SUBMIT> <INPUT TYPE=RESET>
</FORM>
```

In the example above, the <P> and <UL> tags have been used to lay out the text and input fields. The HTML user agent is responsible for handling 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 e-mail 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 types of input fields.

To let users enter more than one line of text, use the Textarea element.

2.15.1 Representing Choices

The radio button 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 typically rendered in a more compact fashion as a pull down combo list.

2.15.2 Form

<FORM> … </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 attribute is missing, the URL of 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.

The Level 2 specification defines and requires support for the HTTP access protocol only.

When the ACTION attribute is set to an HTTP URL, the METHOD attribute must be set to 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, is 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.

2.15.3 Input

<INPUT>

Level 2

The Input element represents a field whose contents may be edited by the user.

Attributes of the Input element:

ALIGN

Vertical alignment of the image. For use only with TYPE=IMAGE in HTML level 2. The possible values are exactly the same as for the ALIGN attribute of the image 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 than specified by the SIZE attribute, in which case the field will scroll appropriately. The default number of characters is unlimited.

NAME

Symbolic name used when transferring the form's contents. The NAME 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, to specify a field with a visible width of 24 characters:

```
INPUT TYPE=text SIZE="24"
```

SRC

A URL or URN specifying an image. For use only with TYPE=IMAGE in HTML Level 2.

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 that 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 Image element.

NOTE: In a future version of the HTML specification, the IMAGE functionality may be folded into an enhanced SUBMIT field.

PASSWORD is the same as the TEXT attribute, except that text is not displayed as it is entered.

RADIO is used for attributes that accept 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 is a button that when pressed resets the form's fields to their specified initial values. The label to be displayed on the button may be specified just as for the SUBMIT button.

SUBMIT 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. If a SUBMIT button is pressed in order to submit the form, and that button has a NAME attribute specified, then that button contributes a name/value pair to the submitted data. Otherwise, a SUBMIT button makes no contribution to the submitted data.

TEXT is used for a 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.

2.15.4 Option

<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.

2.15.5 Select

<SELECT NAME=… > … </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.

Attributes are:

ERROR

Proposed.

MULTIPLE

The MULTIPLE attribute is needed when users are allowed to make several selections, e.g. <SELECT MULTIPLE>.

NAME

Specifies the name that will submitted as a name/value pair.

SIZE

Specifies the number of visible items. If this is greater than one, then the resulting form control will be a list.

The Select element 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>
```

If no option is initially marked as selected, then the first item listed is selected.

2.15.6 Text Area

<TEXTAREA> … </TEXTAREA>

Level 2

The Textarea element lets users enter more than one line of text. For example:

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

The text up to the end tag (</TEXTAREA>) is used to initialize the field's value. This end tag is always required even if the field is initially blank. When submitting a form, the line terminators are implementation dependent.

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. HTML user agents should allow text to extend 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. SGML's default (Reference Quantity Set) limits the length of attribute literals to only 240 characters. The HTML 2.0 SGML declaration increases the limit to 1024 characters.

2.16 Character Data

Level 0

The characters between HTML tags represent text encoded according to ISO 8859/1 8-bit single-byte coded graphic character set known as Latin Alphabet No. 1, or simply Latin-1. There are 256 character positions in the Latin-1 encoding. Latin-1 includes characters from most Western European languages. It consists of the space character, 186 characters that form a subset of the graphic characters in ISO 6937/2 (1983), and four additional characters that are intended for inclusion in ISO 6937/2. Also see Section 2.4.

The lower 128 character positions include a space, 33 control characters, the 26 upper- and lowercase letters of the english alphabet, 10 numerals and 32 other printing characters This subset, functionally identical to ASCII, is defined by ISO 646 7-bit coded character set for information interchange, also known as the International Reference Version. ISO 646 is identical in most respect to the ANSI standard for ASCII (American Standard Code for Information Interchange). The only significant difference between ISO 646 and ASCII is the specific names assigned to the control characters in positions 00-31 and 127.

The upper 128 positions include a non-breaking space, a soft hyphen indicator, 93 graphical characters, 8 unassigned characters, and 25 control characters. *Because non-breaking space and soft hyphen indicator are not recognized and interpreted by all HTML user agents, their use is discouraged.*

There are 58 character positions occupied by control characters. See Section 2.16.2 for details on the interpretation of control characters.

Because certain special characters are subject to interpretation and special processing, information providers and HTML user agent implementors should follow the guidelines in Section 2.16.1.

Certain characters may not be accessible from your keyboard, or some part of your system (i.e. translation software) may not be equipped to deal with 8-bit character codes. HTML and many HTML user agents provide character entity references (see Section 2.17.2) and numerical character references (see Section 2.17.3) to facilitate the entry and interpretation of characters by name and by numerical position.

Because certain characters will be interpreted as markup, they must be represented by markup as described in Section 2.16.3 and Section 2.16.4.

2.16.1 Special Characters

Certain characters have special meaning in HTML documents. There are two printing characters which may be interpreted by an HTML application to have an effect of the format of the text:

Space

- Interpreted as a word space (place where a line can be broken) in all contexts except the Preformatted Text element.
- Interpreted as a nonbreaking space within the Preformatted Text element.

Hyphen

- Interpreted as a hyphen glyph in all contexts
- Interpreted as a potential word space by hyphenation engine

2.16.2 Control Characters

Control characters are non-printable characters that are typically used for communication and device control, as format effectors, and as information separators.

In SGML applications, the use of control characters is limited in order to maximize the chance of successful interchange over heterogenous networks and operating systems. In HTML, only three control characters are used. The valid control characters and their interpretation are:

Horizontal Tab (HT - 9 dec)

- Interpreted as a word space in all contexts except preformatted text.
- Within preformatted text, the tab should be interpreted to shift the horizontal column position to the next position which is a multiple of 8 on the same line; that is, col := (col+8) mod 8

Line Feed (LF - 10 dec)

- Interpreted as a word space in all contexts except preformatted text.
- Within the Preformatted Text element, the tab should be interpreted as a shift to the start of a new line; that is, col := 0; row := row+1

Carriage Return (CR - 13 dec)

- Interpreted as a word space in all contexts.

2.16.3 Numeric Character References

Any *printing* character within the 8-bit character encoding of ISO 8859/1 (256 character positions) or the 7-bit character encoding of ISO 646 (128 character positions) may be represented within the text of an HTML document by a numeric character reference. See Section 2.17.1 for a list of the characters, their names and input syntax.

Two reasons for using a numeric character reference:

- the keyboard does not provide a key for the character, such as on U.S. keyboards which do not provide European characters
- the character may be interpreted as SGML coding, such as the ampersand (&), double quotes ("), the lesser (<) and greater (>) characters

Numeric character references are represented in an HTML document as SGML entities whose name is number sign (#) followed by a numeral from 32-126 and 161-255. The HTML DTD includes a numeric character for each of the printing characters in Latin-1, so that one may reference them by number if it is inconvenient to enter them directly:

```
the ampersand (&#38;), double quotes (&#34;),
lesser (&#60;) and greater (&#62;) characters
```

2.16.4 Character Entities

Many of the Latin alphabet No. 1 set of printing characters may be represented within the text of an HTML document by a character entity. See 2.17.2 for a list of the characters, names, input syntax, and descriptions. See 5.2.1 for the SGML entity definitions of “Added Latin 1 for HTML”.

Two reasons for using a character entity:

- the keyboard does not provide a key for the character, such as on U.S. keyboards which do not provide European characters
- the character may be interpreted as SGML coding, such as the ampersand (&), double quotes ("), the lesser (<) and greater (>) characters

A character entity is represented in an HTML document as an SGML entity whose name is defined in the HTML DTD. The HTML DTD includes a character entity for each of the SGML markup characters and for each of the printing characters in the upper half of Latin-1, so that one may reference them by name if it is inconvenient to enter them directly:

```
the ampersand (&amp;), double quotes (&quot;),
lesser (&lt;) and greater (&gt;) characters
Kurt G&ouml;del was a famous logician and mathematician.
```

NOTE: To ensure that a string of characters is not interpreted as markup, represent all occurrences of <, >, and & by character or entity references.

NOTE: There are SGML features, CDATA and 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 used in this version of the HTML DTD. An earlier HTML specification included an Example element (<XMP>) whose syntax is not expressible in

SGML. No markup was recognized inside of the Example element except the </XMP> end tag. While HTML user agents are encouraged to support this idiom, its use is deprecated.

2.17 Character Entity Sets

The following entity names are used in HTML, always prefixed by ampersand (&) and 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.

2.17.1 Numeric and Special Graphic Entities

The following table lists each of the supported characters specified in the Numeric and Special Graphic entity set, along with its name, syntax for use, and description. This list is derived from ISO Standard 8879:1986//ENTITIES Numeric and Special Graphic//EN however HTML does *not* provide support for the entire entity set. Only the entities listed below are supported.

GLYPH	NAME	SYNTAX	DESCRIPTION
<	lt	<	Less than sign
>	gt	>	Greater than sign
&	amp	&	Ampersand
"	quot	"	Double quote sign

2.17.2 ISO Latin 1 Character Entities

The following table lists each of the characters specified in the Added Latin 1 entity set, along with its name, syntax for use, and description. This list is derived from ISO Standard 8879:1986//ENTITIES Added Latin 1//EN. HTML supports the entire entity set.

GLYPH	NAME	SYNTAX	DESCRIPTION
Á	Aacute	Á	Capital A, acute accent
À	Agrave	À	Capital A, grave accent
Â	Acirc	Â	Capital A, circumflex accent
	Atilde	Ã	Capital A, tilde
Å	Aring	Å	Capital A, ring
Ä	Auml	Ä	Capital A, dieresis or umlaut mark
Æ	AElig	Æ	Capital AE dipthong (ligature)
Ç	Ccedil	Ç	Capital C, cedilla
É	Eacute	É	Capital E, acute accent
È	Egrave	È	Capital E, grave accent

Ê	Ecirc	Ê	Capital E, circumflex accent
Ë	Euml	Ë	Capital E, dieresis or umlaut mark
Í	Iacute	Í	Capital I, acute accent
Ì	Igrave	Ì	Capital I, grave accent
Î	Icirc	Î	Capital I, circumflex accent
Ï	Iuml	Ï	Capital I, dieresis or umlaut mark
	ETH	Ð	Capital Eth, Icelandic
Ñ	Ntilde	Ñ	Capital N, tilde
Ó	Oacute	Ó	Capital O, acute accent
Ò	Ograve	Ò	Capital O, grave accent
Ô	Ocirc	Ô	Capital O, circumflex accent
Õ	Otilde	Õ	Capital O, tilde
Ö	Ouml	Ö	Capital O, dieresis or umlaut mark
Ø	Oslash	Ø	Capital O, slash
Ú	Uacute	Ú	Capital U, acute accent
Ù	Ugrave	Ù	Capital U, grave accent
Û	Ucirc	Û	Capital U, circumflex accent
Ü	Uuml	Ü	Capital U, dieresis or umlaut mark
	Yacute	Ý	Capital Y, acute accent
	THORN	Þ	Capital THORN, Icelandic
ß	szlig	ß	Small sharp s, German (sz ligature)
á	aacute	á	Small a, acute accent
à	agrave	à	Small a, grave accent
â	acirc	â	Small a, circumflex accent
ã	atilde	ã	Small a, tilde
ä	auml	ä	Small a, dieresis or umlaut mark
æ	aelig	æ	Small ae dipthong (ligature)
ç	ccedil	ç	Small c, cedilla
é	eacute	é	Small e, acute accent
è	egrave	è	Small e, grave accent
ê	ecirc	ê	Small e, circumflex accent
ë	euml	ë	Small e, dieresis or umlaut mark
í	iacute	í	Small i, acute accent
ì	igrave	ì	Small i, grave accent
î	icirc	î	Small i, circumflex accent
ï	iuml	ï	Small i, dieresis or umlaut mark
	eth	ð	Small eth, Icelandic
ñ	ntilde	ñ	Small n, tilde
ó	oacute	ó	Small o, acute accent
ò	ograve	ò	Small o, grave accent
ô	ocirc	ô	Small o, circumflex accent
õ	otilde	õ	Small o, tilde
ö	ouml	ö	Small o, dieresis or umlaut mark
ø	oslash	ø	Small o, slash
ú	uacute	ú	Small u, acute accent
ù	ugrave	ù	Small u, grave accent
û	ucirc	û	Small u, circumflex accent
ü	uuml	ü	Small u, dieresis or umlaut mark
y	yacute	ý	Small y, acute accent
	thorn	þ	Small thorn, Icelandic
ÿ	yuml	ÿ	Small y, dieresis or umlaut mark

2.17.3 Numerical Character References

This list, sorted numerically, is derived from ISO 8859/1 8-bit single-byte coded graphic character set:

```
GLYPH   REFERENCE            DESCRIPTION

        &#00; - &#08;        Unused
        &#09;                Horizontal tab
        &#10;                Line feed
        &#11; - &#31;        Unused

        &#32;                 Space
!       &#33;                 Exclamation mark
"       &#34;                 Quotation mark
#       &#35;                 Number sign
$       &#36;                 Dollar sign
%       &#37;                 Percent sign
&       &#38;                 Ampersand
'       &#39;                 Apostrophe
(       &#40;                 Left parenthesis
)       &#41;                 Right parenthesis
*       &#42;                 Asterisk
+       &#43;                 Plus sign
,       &#44;                 Comma
-       &#45;                 Hyphen
.       &#46;                 Period (fullstop)
/       &#47;                 Solidus (slash)

0-9     &#48; - &#57;        Digits 0-9

:       &#58;                 Colon
;       &#59;                 Semi-colon
<       &#60;                 Less than
=       &#61;                 Equals aign
>       &#62;                 Greater than
?       &#63;                 Question mark
@       &#64;                 Commercial at

A-Z     &#65; - &#90;        Letters A-Z

[       &#91;                 Left square bracket
\       &#92;                 Reverse solidus (backslash)
]       &#93;                 Right square bracket
_       &#95;                 Horizontal bar
`       &#96;                 Acute accent

a-z    &#97; - &#122;        Letters a-z

{      &#123;                 Left curly brace
|      &#124;                 Vertical bar
```

}	}	Right curly brace
~	~	Tilde
	&#127; -	Unused
¡	¡	Inverted exclamation
¢	¢	Cent sign
£	£	Pound sterling
¤	¤	General currency sign
¥	¥	Yen sign
|	¦	Broken vertical bar
§	§	Section sign
¨	¨	Umlaut (dieresis)
©	©	Copyright
ª	ª	Feminine ordinal
«	«	Left angle quote, guillemotleft
¬	¬	Not sign
-		Soft hyphen
®	®	Registered trademark
	¯	Macron accent
°	°	Degree sign
±	±	Plus or minus
2	²	Superscript two
3	³	Superscript three
´	´	Acute accent
µ	µ	Micro sign
¶	¶	Paragraph sign
·	·	Middle dot
,	¸	Cedilla
1	¹	Superscript one
°	º	Masculine ordinal
»	»	Right angle quote, guillemotright
/	¼	Fraction one-fourth
/	½	Fraction one-half
/	¾	Fraction three-fourths
¿	¿	Inverted question mark
Á	À	Capital A, acute accent
À	Á	Capital A, grave accent
Â	Â	Capital A, circumflex accent
	Ã	Capital A, tilde
Ä	Ä	Capital A, ring
Å	Å	Capital A, dieresis or umlaut mark
Æ	Æ	Capital AE dipthong (ligature)
Ç	Ç	Capital C, cedilla
É	È	Capital E, acute accent
È	É	Capital E, grave accent
Ê	Ê	Capital E, circumflex accent
Ë	Ë	Capital E, dieresis or umlaut mark
Í	Ì	Capital I, acute accent
Ì	Í	Capital I, grave accent
Î	Î	Capital I, circumflex accent
Ï	Ï	Capital I, dieresis or umlaut mark
	Ð	Capital Eth, Icelandic

Ñ	Ñ	Capital N, tilde
Ó	Ò	Capital O, acute accent
Ò	Ó	Capital O, grave accent
Ô	Ô	Capital O, circumflex accent
Õ	Õ	Capital O, tilde
Ö	Ö	Capital O, dieresis or umlaut mark
X	×	Multiply sign
Ø	Ø	Capital O, slash
Ú	Ù	Capital U, acute accent
Ù	Ú	Capital U, grave accent
Û	Û	Capital U, circumflex accent
Ü	Ü	Capital U, dieresis or umlaut mark
Y	Ý	Capital Y, acute accent
	Þ	Capital THORN, Icelandic
ß	ß	Small sharp s, German (sz ligature)
á	à	Small a, acute accent
à	á	Small a, grave accent
â	â	Small a, circumflex accent
ã	ã	Small a, tilde
ä	ä	Small a, tilde
å	å	Small a, dieresis or umlaut mark
æ	æ	Small ae dipthong (ligature)
ç	ç	Small c, cedilla
é	è	Small e, acute accent
è	é	Small e, grave accent
ê	ê	Small e, circumflex accent
ë	ë	Small e, dieresis or umlaut mark
í	ì	Small i, acute accent
ì	í	Small i, grave accent
î	î	Small i, circumflex accent
ï	ï	Small i, dieresis or umlaut mark
	ð	Small eth, Icelandic
ñ	ñ	Small n, tilde
ó	ò	Small o, acute accent
ò	ó	Small o, grave accent
ô	ô	Small o, circumflex accent
õ	õ	Small o, tilde
ö	ö	Small o, dieresis or umlaut mark
÷	÷	Division sign
ø	ø	Small o, slash
ú	ù	Small u, acute accent
ù	ú	Small u, grave accent
û	û	Small u, circumflex accent
ü	ü	Small u, dieresis or umlaut mark
	ý	Small y, acute accent
	þ	Small thorn, Icelandic

```
ÿ       &#255;                  Small y, dieresis or umlaut mark
```

3. Security Considerations

Anchors, embedded images, and all other elements which contain URIs as parameters may cause the URI to be dereferenced in response to user input. 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.

4. Obsolete and Proposed Features

4.1 Obsolete Features

This section describes elements that are no longer part of HTML. Client implementors should implement these obsolete elements for compatibility with previous versions of the HTML specification.

4.1.1 Comment

The Comment element is used to delimit unneeded text and comments. The Comment element has been introduced in some HTML applications but should be replaced by the SGML comment feature in new HTML user agents (see Section 2.6.5).

4.1.2 Highlighted Phrase

The Highlighted Phrase element (<HP>) should be ignored if not implemented. This element has been replaced by more meaningful elements (see Section 2.9).

Example of use:

```
<HP1>first highlighted phrase</HP1>non
highlighted text<HP2>second highlighted phrase</HP2> etc.
```

4.1.3 Plain Text

<PLAINTEXT>

The Plain Text element is used to terminates the HTML entity and to indicate that what follows is not SGML which does not require parsing. Instead, an old HTTP convention specified that what followed was an ASCII (MIME “text/plain”) body. Its presence is an optimization. There is no closing tag.

Example of use:

```
<PLAINTEXT>
0001 This is line one of a long listing
0002 file from <ANY@HOST.INC.COM> which is sent
```

4.1.4 Example and Listing

<XMP> … </XMP> and <LISTING> … </LISTING>

The Example element and Listing element have been replaced by the Preformatted Text element.

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 typically rendered in a monospaced 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-1 printable characters, expect for the end tag opener. The Example 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. HTML user agents may vary on how they interpret other tags within Example and Listing elements.
- Line boundaries within the text are rendered as a move to the beginning of the next line, except for one immediately following a start tag or immediately preceding an end tag.
- The ASCII horizontal tab 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 rendered so that at least 132 characters fit on a line. The Example element is rendered to that at least 80 characters fit on a line but is otherwise identical to the Listing element.

4.2 Proposed Features

This section describes proposed HTML elements and entities that are not currently supported under HTML Levels 0, 1, or 2, but may be supported in the future.

4.2.1 Defining Instance

<DFN> … </DFN>

The Defining Instance element indicates the defining instance of a term. The typical rendering is bold or bold italic. This element is not widely supported.

4.2.2 Special Characters

To indicate special characters, HTML uses entity or numeric representations. Two additional character presentations are proposed:

```
CHARACTER                   REPRESENTATION
Non-breaking space           
Soft-hyphen                 &shy;
Registered                  &reg;
Copyright                   &copy;
```

4.2.3 Strike

<STRIKE> … </STRIKE>

The Strike element is proposed to indicate strikethrough, a font style in which a horizontal line appears through characters. This element is not widely supported.

4.2.4 Underline

<U> … </U>

The Underline element is proposed to indicate that the text should be rendered as underlined. This proposed tag is not supported by all HTML user agents.

Example of use:

```
The text <U>shown here</U> is rendered in the document as underlined.
```

5. HTML DOCUMENT TYPE DEFINITIONS

5.1 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).

--

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 surprising 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
>
<!--
        $Id: html.decl,v 1.9 1994/11/15 19:54:44 connolly 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
-->
```

5.1.1 Sample SGML Open Style Entity Catalog for HTML

The SGML standard describes an "entity manager" as the portion or component of an SGML system that maps SGML entities into the actual storage model (e.g., the file system). The standard itself does not define a particular mapping methodology or notation.

To assist the interoperability among various SGML tools and systems, the SGML Open consortium has passed a technical resolution that defines a format for an application-independent entity catalog that maps external identifiers and/or entity names to file names.

Each entry in the catalog associates a storage object identifier (such as a file name) with information about the external entity that appears in the SGML document. In addition to entries that associate public identifiers, a catalog entry can associate an entity name with a storage object indentifier. For example, the following are possible catalog entries:

```
PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN" "iso-lat1.gml"
PUBLIC "-//ACME DTD Writers//DTD General Report//EN" report.dtd
ENTITY "graph1" "graphics\graph1.cgm"
```

In particular, the following shows entries relevant to HTML.

```
        -- catalog: SGML Open style entity catalog for HTML --
        -- $Id: catalog,v 1.1 1994/10/07 21:35:07 connolly Exp $ --

        -- Ways to refer to Level 2: most general to most specific --
PUBLIC  "-//IETF//DTD HTML//EN"                 html.dtd
PUBLIC  "-//IETF//DTD HTML//EN//2.0"            html.dtd
PUBLIC  "-//IETF//DTD HTML Level 2//EN"         html.dtd
PUBLIC  "-//IETF//DTD HTML Level 2//EN//2.0"    html.dtd

        -- Ways to refer to Level 1: most general to most specific --
PUBLIC  "-//IETF//DTD HTML Level 1//EN"         html-1.dtd
PUBLIC  "-//IETF//DTD HTML Level 1//EN//2.0"    html-1.dtd

        -- Ways to refer to Level 0: most general to most specific --
PUBLIC  "-//IETF//DTD HTML Level 0//EN"         html-0.dtd
PUBLIC  "-//IETF//DTD HTML Level 0//EN//2.0"    html-0.dtd

        -- ISO latin 1 entity set for HTML --
PUBLIC  "-//IETF//ENTITIES Added Latin 1//EN"   ISOlat1.sgml
```

5.2 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.dtd,v 1.21 1994/11/15 19:54:38 connolly 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
        "-//IETF//DTD HTML//EN//2.0"

        -- Typical usage:

            <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
            <html>
            …
            </html>
        --
        >


<!--================== Feature Test Entities ===============================-->

<!ENTITY % HTML.Recommended "IGNORE"
        -- Certain features of the language are necessary for compatibility
           with widespread usage, but they may compromise the structural
           integrity of a document. This feature test entity enables
           a more prescriptive document type definition that eliminates
           those features.
        -->

<![ %HTML.Recommended [
        <!ENTITY % HTML.Deprecated "IGNORE">
]]>

<!ENTITY % HTML.Deprecated "INCLUDE"
        -- Certain features of the language are necessary for compatibility
           with earlier versions of the specification, but they tend
           to be used an implemented inconsistently, and their use is
           deprecated. This feature test entity enables a document type
           definition that eliminates these features.
        -->
```

```
<!ENTITY % HTML.Highlighting "INCLUDE"
        -- Use this feature test entity to validate that a document
           uses no highlighting tags, which may be ignored on minimal
           implementations.
        -->

<!ENTITY % HTML.Forms "INCLUDE"
        -- Use this feature test entity to validate that a document
           contains no forms, which may not be supported in minimal
           implementations
        -->

<!--================== Imported Names ====================================-->

<!ENTITY % Content-Type "CDATA"
        -- meaning an internet media type
           (aka MIME content type, as per RFC1521)
        -->

<!ENTITY % HTTP-Method "GET | POST"
        -- as per HTTP specification, in progress
        -->

<!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
        aka RFC 1630

        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.

        -->


<!--================== DTD "Macros" ==========================================-->

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

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


<!--================ Character mnemonic entities ===========================-->

<!ENTITY % ISOlat1 PUBLIC
  "-//IETF//ENTITIES Added Latin 1 for HTML//EN">
%ISOlat1;

<!ENTITY amp CDATA "&#38;"     -- ampersand          -->
<!ENTITY gt CDATA "&#62;"      -- greater than       -->
```

```
<!ENTITY lt CDATA "&#60;"      -- less than          -->
<!ENTITY quot CDATA "&#34;"    -- double quote       -->


<!--=================== Text Markup =======================================-->

<![ %HTML.Highlighting [

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

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

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

<!ELEMENT (%font;|%phrase) - - (%text)+>
<!-- <TT>       Typewriter text                         -->
<!-- <B>        Bold text                               -->
<!-- <I>        Italic text                             -->

<!-- <EM>       Emphasized phrase                       -->
<!-- <STRONG>   Strong emphais                          -->
<!-- <CODE>     Source code phrase                      -->
<!-- <SAMP>     Sample text or characters               -->
<!-- <KBD>      Keyboard phrase, e.g. user input        -->
<!-- <VAR>      Variable phrase or substituable         -->
<!-- <CITE>     Name or title of cited work             -->

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

]]>

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

<!ELEMENT BR    - O EMPTY>
<!-- <BR>       Line break      -->


<!--================== Link Markup ========================================-->

<![ %HTML.Recommended [
        <!ENTITY % linkName "ID">
]]>

<!ENTITY % linkName "CDATA">

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

<!ENTITY % linkExtraAttributes
        "REL %linkType #IMPLIED
        REV %linkType #IMPLIED
        URN CDATA #IMPLIED
```

```
        TITLE CDATA #IMPLIED
        METHODS NAMES #IMPLIED
        ">

<![ %HTML.Recommended [
        <!ENTITY % A.content   "(%text)+"
        -- <H1><a name="xxx">Heading</a></H1>
                is preferred to
           <a name="xxx"><H1>Heading</H1></a>
        -->
]]>

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

<!ELEMENT A     - - %A.content -(A)>
<!ATTLIST A
        HREF %URI #IMPLIED
        NAME %linkName #IMPLIED
        %linkExtraAttributes;
        >
<!-- <A>                Anchor; source and/or destination of a link     -->
<!-- <A NAME="...">     Name of this anchor                             -->
<!-- <A HREF="...">     Address of link destination                     -->
<!-- <A URN="...">      Permanent address of destination                -->
<!-- <A REL=...>        Relationship of this anchor to destination      -->
<!-- <A REV=...>        Relationship of destination to this anchor      -->
<!-- <A TITLE="...">    Title of destination (advisory)
                -->
<!-- <A METHODS="...">  Operations allowed on destination
(advisory)      -->


<!--=================== Images ==========================================-->

<!ELEMENT IMG    - O EMPTY>
<!ATTLIST IMG
        SRC %URI;  #REQUIRED
        ALT CDATA #IMPLIED
        ALIGN (top|middle|bottom) #IMPLIED
        ISMAP (ISMAP) #IMPLIED
        >

<!-- <IMG>              Image; icon, glyph or illustration      -->
<!-- <IMG SRC="...">    Address of image object                 -->
<!-- <IMG ALT="...">    Textual alternative                     -->
<!-- <IMG ALIGN=...>    Position relative to text               -->
<!-- <IMG ISMAP>        Each pixel can be a link                -->

<!--=================== Paragraphs=======================================-->

<!ELEMENT P     - O (%text)+>
<!-- <P>        Paragraph       -->
```

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

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

<!ELEMENT ( %heading )  - -  (%text;)+>
<!-- <H1>       Heading, level 1 -->
<!-- <H2>       Heading, level 2 -->
<!-- <H3>       Heading, level 3 -->
<!-- <H4>       Heading, level 4 -->
<!-- <H5>       Heading, level 5 -->
<!-- <H6>       Heading, level 6 -->


<!--=================== Text Flows ============================================>

<![ %HTML.Forms [
        <!ENTITY % block.forms "| FORM | ISINDEX">
]]>

<!ENTITY % block.forms "">

<![ %HTML.Deprecated [
        <!ENTITY % preformatted "PRE | XMP | LISTING">
]]>

<!ENTITY % preformatted "PRE">

<!ENTITY % block "P | %list | DL
        | %preformatted
        | BLOCKQUOTE %block.forms">

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

<!ENTITY % pre.content "#PCDATA | A | HR | BR">
<!ELEMENT PRE - - (%pre.content)+>
<!ATTLIST PRE
        WIDTH NUMBER #implied
        >

<!-- <PRE>              Preformatted text               -->
<!-- <PRE WIDTH=...>    Maximum characters per line     -->

<![ %HTML.Deprecated [

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

<!ELEMENT (XMP|LISTING) - -  %literal>
```

```
<!-- <XMP>              Example section         -->
<!-- <LISTING>          Computer listing        -->

<!ELEMENT PLAINTEXT - O %literal>
<!-- <PLAINTEXT>        Plain text passage      -->

]]>


<!--=================== Lists ================================================-->

<!ELEMENT DL    - -  (DT | DD)+>
<!ATTLIST DL
        COMPACT (COMPACT) #IMPLIED>

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

<!-- <DL>               Definition list, or glossary    -->
<!-- <DL COMPACT>       Compact style list              -->
<!-- <DT>               Term in definition list         -->
<!-- <DD>               Definition of term              -->

<!ELEMENT (OL|UL) - -  (LI)+>
<!ELEMENT (DIR|MENU) - -  (LI)+ -(%block)>
<!ATTLIST (%list)
        COMPACT (COMPACT) #IMPLIED>
<!-- <UL>               Unordered list                  -->
<!-- <UL COMPACT>       Compact list style              -->
<!-- <OL>               Ordered, or numbered list       -->
<!-- <OL COMPACT>       Compact list style              -->
<!-- <DIR>              Directory list                  -->
<!-- <DIR COMPACT>      Compact list style              -->
<!-- <MENU>             Menu list                       -->
<!-- <MENU COMPACT>     Compact list style              -->

<!ELEMENT LI    - O %flow>

<!-- <LI>               List item                       -->

<!--=================== Document Body ========================================-->

<![ %HTML.Recommended [
        <!ENTITY % body.content "(%heading|%block|HR|ADDRESS)*"
        -- <h1>Heading</h1>
           <p>Text ...
                is preferred to
           <h1>Heading</h1>
           Text ...
        -->
]]>

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

```
<!ELEMENT BODY O O  %body.content>
<!-- <BODY>     Document body   -->

<!ELEMENT BLOCKQUOTE - - %body.content>
<!-- <BLOCKQUOTE>       Quoted passage  -->

<!ELEMENT ADDRESS - - (%text|P)*>
<!-- <ADDRESS>  Address, signature, or byline for document or
passage -->


<!--================ Forms ===============================================-->

<![ %HTML.Forms [

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

<!-- <FORM>                     Fill-out or data-entry form     -->
<!-- <FORM ACTION="...">        Address for completed form      -->
<!-- <FORM METHOD=...>          Method of submitting form       -->
<!-- <FORM ENCTYPE="...">       Representation of form data     -->

<!ENTITY % InputType "(TEXT | PASSWORD | CHECKBOX |
                        RADIO | SUBMIT | RESET |
                        IMAGE | HIDDEN )">
<!ELEMENT INPUT - O EMPTY>
<!ATTLIST INPUT
        TYPE %InputType TEXT
        NAME CDATA #IMPLIED
        VALUE CDATA #IMPLIED
        SRC %URI #IMPLIED
        CHECKED (CHECKED) #IMPLIED
        SIZE CDATA #IMPLIED
        MAXLENGTH NUMBER #IMPLIED
        ALIGN (top|middle|bottom) #IMPLIED
        >

<!-- <INPUT>                    Form input datum                -->
<!-- <INPUT TYPE=...>           Type of input interaction       -->
<!-- <INPUT TYPE=...>           Name of form datum              -->
<!-- <INPUT VALUE="...">        Default/initial/selected value  -->
<!-- <INPUT SRC="...">          Address of image                -->
<!-- <INPUT CHECKED>            Initial state is "on"           -->
<!-- <INPUT SIZE=...>           Field size hint                 -->
<!-- <INPUT MAXLENGTH=...>      Data length maximum             -->
<!-- <INPUT ALIGN=...>          Image
alignment                       -->
```

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

<!-- <SELECT>                  Selection of option(s)                 -->
<!-- <SELECT NAME=...>         Name of form datum                     -->
<!-- <SELECT SIZE=...>         Number of options displayed at a time  -->
<!-- <SELECT MULTIPLE>         Multiple selections allowed            -->

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

<!-- <OPTION>                  A selection option                     -->
<!-- <OPTION SELECTED>         Initial state                          -->
<!-- <OPTION VALUE="">         Form datum value for this option       -->

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

<!-- <TEXTAREA>                An area for text input                 -->
<!-- <TEXTAREA NAME=...>       Name of form datum                     -->
<!-- <TEXTAREA ROWS=...>       Height of area                         -->
<!-- <TEXTAREA COLS=...>       Width of area                          -->

]]>


<!--================ Document Head =======================================-->

<!ENTITY % head.link "& LINK*">

<![ %HTML.Recommended [
        <!ENTITY % head.nextid "">
]]>
<!ENTITY % head.nextid "& NEXTID?">

<!ENTITY % head.content "TITLE & ISINDEX? & BASE? & META*
                         %head.nextid
                         %head.link">

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

```
<!ELEMENT TITLE - -  (#PCDATA)>
<!-- <TITLE>    Title of document -->

<!ELEMENT LINK - O EMPTY>
<!ATTLIST LINK
        HREF %URI #REQUIRED
        %linkExtraAttributes; >
<!-- <LINK>             Link from this document                          -->
<!-- <LINK HREF="...">  Address of link destination                      -->
<!-- <LINK URN="...">   Lasting name of destination                      -->
<!-- <LINK REL=...>     Relationship of this document to destination      -->
<!-- <LINK REV=...>     Relationship of destination to this document
        -->
<!-- <LINK TITLE="..."> Title of destination (advisory)                  -->
<!-- <LINK METHODS="..."> Operations allowed on destination (advisory)      -->

<!ELEMENT ISINDEX - O EMPTY>
<!-- <ISINDEX>          Document is a searchable index          -->

<!ELEMENT BASE - O EMPTY>
<!ATTLIST BASE
        HREF %URI; #REQUIRED
        >
<!-- <BASE>             Base context document                   -->
<!-- <BASE HREF="...">  Address for this document               -->

<!ELEMENT NEXTID - O EMPTY>
<!ATTLIST NEXTID N %linkName #REQUIRED>
<!-- <NEXTID>           Next ID to use for link name                     -->
<!--
<NEXTID N=...>  Next ID to use for link name                    -->

<!ELEMENT META - O EMPTY>
<!ATTLIST META
        HTTP-EQUIV  NAME    #IMPLIED
        NAME        NAME    #IMPLIED
        CONTENT     CDATA   #REQUIRED
        >
<!-- <META>                     Generic Metainformation         -->
<!-- <META HTTP-EQUIV=...>      HTTP response header name       -->
<!-- <META HTTP-EQUIV=...>      Metainformation name            -->
<!-- <META CONTENT="...">       Associated information          -->

<!--================ Document Structure ===================================-->

<![ %HTML.Deprecated [
        <!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;
        >

<!-- <HTML>                     HyperText Markup Language Document      -->
<!-- <HTML
VERSION="...">  Version of HTML specification           -->
```

5.2.1 ISO Latin 1 Definitions for HTML

```
<!-- (C) International Organization for Standardization 1986
     Permission to copy in any form is granted for use with
     conforming SGML systems and applications as defined in
     ISO 8879:1986, provided this notice is included in all copies.
-->
<!-- Character entity set. Typical invocation:
     <!ENTITY % ISOlat1 PUBLIC
       "-//IETF//ENTITIES Added Latin 1 for HTML//EN">
     %ISOlat1;
-->
<!-- Modified for use in HTML
   $Id: ISOlat1.sgml,v 1.1 1994/09/24 14:06:34 connolly Exp $ -->
<!ENTITY AElig  CDATA "&#198;" -- capital AE diphthong (ligature) -->
<!ENTITY Aacute CDATA "&#193;" -- capital A, acute accent -->
<!ENTITY Acirc  CDATA "&#194;" -- capital A, circumflex accent -->
<!ENTITY Agrave CDATA "&#192;" -- capital A, grave accent -->
<!ENTITY Aring  CDATA "&#197;" -- capital A, ring -->
<!ENTITY Atilde CDATA "&#195;" -- capital A, tilde -->
<!ENTITY Auml   CDATA "&#196;" -- capital A, dieresis or umlaut mark -->
<!ENTITY Ccedil CDATA "&#199;" -- capital C, cedilla -->
<!ENTITY ETH    CDATA "&#208;" -- capital Eth, Icelandic -->
<!ENTITY Eacute CDATA "&#201;" -- capital E, acute accent -->
<!ENTITY Ecirc  CDATA "&#202;" -- capital E, circumflex accent -->
<!ENTITY Egrave CDATA "&#200;" -- capital E, grave accent -->
<!ENTITY Euml   CDATA "&#203;" -- capital E, dieresis or umlaut mark -->
<!ENTITY Iacute CDATA "&#205;" -- capital I, acute accent -->
<!ENTITY Icirc  CDATA "&#206;" -- capital I, circumflex accent -->
<!ENTITY Igrave CDATA "&#204;" -- capital I, grave accent -->
<!ENTITY Iuml   CDATA "&#207;" -- capital I, dieresis or umlaut mark -->
<!ENTITY Ntilde CDATA "&#209;" -- capital N, tilde -->
<!ENTITY Oacute CDATA "&#211;" -- capital O, acute accent -->
<!ENTITY Ocirc  CDATA "&#212;" -- capital O, circumflex accent -->
<!ENTITY Ograve CDATA "&#210;" -- capital O, grave accent -->
<!ENTITY Oslash CDATA "&#216;" -- capital O, slash -->
<!ENTITY Otilde CDATA "&#213;" -- capital O, tilde -->
<!ENTITY Ouml   CDATA "&#214;" -- capital O, dieresis or umlaut mark -->
<!ENTITY THORN  CDATA "&#222;" -- capital THORN, Icelandic -->
<!ENTITY Uacute CDATA "&#218;" -- capital U, acute accent -->
<!ENTITY Ucirc  CDATA "&#219;" -- capital U, circumflex accent -->
<!ENTITY Ugrave CDATA "&#217;" -- capital U, grave accent -->
<!ENTITY Uuml   CDATA "&#220;" -- capital U, dieresis or umlaut mark -->
<!ENTITY Yacute CDATA "&#221;" -- capital Y, acute accent -->
```

```
<!ENTITY aacute CDATA "&#225;" -- small a, acute accent -->
<!ENTITY acirc  CDATA "&#226;" -- small a, circumflex accent -->
<!ENTITY aelig  CDATA "&#230;" -- small ae diphthong (ligature) -->
<!ENTITY agrave CDATA "&#224;" -- small a, grave accent -->
<!ENTITY aring  CDATA "&#229;" -- small a, ring -->
<!ENTITY atilde CDATA "&#227;" -- small a, tilde -->
<!ENTITY auml   CDATA "&#228;" -- small a, dieresis or umlaut mark -->
<!ENTITY ccedil CDATA "&#231;" -- small c, cedilla -->
<!ENTITY eacute CDATA "&#233;" -- small e, acute accent -->
<!ENTITY ecirc  CDATA "&#234;" -- small e, circumflex accent -->
<!ENTITY egrave CDATA "&#232;" -- small e, grave accent -->
<!ENTITY eth    CDATA "&#240;" -- small eth, Icelandic -->
<!ENTITY euml   CDATA "&#235;" -- small e, dieresis or umlaut mark -->
<!ENTITY iacute CDATA "&#237;" -- small i, acute accent -->
<!ENTITY icirc  CDATA "&#238;" -- small i, circumflex accent -->
<!ENTITY igrave CDATA "&#236;" -- small i, grave accent -->
<!ENTITY iuml   CDATA "&#239;" -- small i, dieresis or umlaut mark -->
<!ENTITY ntilde CDATA "&#241;" -- small n, tilde -->
<!ENTITY oacute CDATA "&#243;" -- small o, acute accent -->
<!ENTITY ocirc  CDATA "&#244;" -- small o, circumflex accent -->
<!ENTITY ograve CDATA "&#242;" -- small o, grave accent -->
<!ENTITY oslash CDATA "&#248;" -- small o, slash -->
<!ENTITY otilde CDATA "&#245;" -- small o, tilde -->
<!ENTITY ouml   CDATA "&#246;" -- small o, dieresis or umlaut mark -->
<!ENTITY szlig  CDATA "&#223;" -- small sharp s, German (sz ligature) -->
<!ENTITY thorn  CDATA "&#254;" -- small thorn, Icelandic -->
<!ENTITY uacute CDATA "&#250;" -- small u, acute accent -->
<!ENTITY ucirc  CDATA "&#251;" -- small u, circumflex accent -->
<!ENTITY ugrave CDATA "&#249;" -- small u, grave accent -->
<!ENTITY uuml   CDATA "&#252;" -- small u, dieresis or umlaut mark -->
<!ENTITY yacute CDATA "&#253;" -- small y, acute accent -->
<!ENTITY yuml   CDATA "&#255;" -- small y, dieresis or umlaut mark -->
```

5.3 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):

```
<!--    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-0.dtd,v 1.9 1994/11/15 19:54:42 connolly 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
        "-//IETF//DTD HTML Level 0//EN//2.0"
        -- public identifier for "minimal conformance" version             --

        -- Typical usage:

            <!DOCTYPE HTML PUBLIC
                "-//IETF//DTD HTML Level 0//EN">
            <html>
            ...
            </html>
        --
        >

<!-- Feature Test Entities -->

<!ENTITY % HTML.Highlighting "IGNORE">
<!ENTITY % HTML.Forms "IGNORE">

<!ENTITY % head.link " " -- no link in head at level 0 -->
<!ENTITY % linkExtraAttributes " ">

<!ENTITY % html PUBLIC "-//IETF//DTD HTML//EN//2.0">
%html;
```

5.4 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-1.dtd,v 1.5 1994/09/23 22:46:54 connolly 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
        "-//IETF//DTD HTML Level 1//EN//2.0"

        -- Typical usage:

            <!DOCTYPE HTML PUBLIC
                "-//IETF//DTD HTML Level 1//EN">
            <html>
            ...
```

```
            </html>
        --
        >

<!-- Feature Test Entities -->
<!ENTITY % HTML.Forms "IGNORE">

<!ENTITY % html PUBLIC "-//IETF//DTD HTML//EN//2.0">
%html;
```

6. DTD Element References

Document type definition (DTD) element references are aids to reading and understanding the DTDs.

6.1 Recommended Level 2 Element Reference

This listing eliminates deprecated idioms. Consult this reference when generating new documents.

This reference is available as hypertext at http://www.hal.com/%7Econnolly/html-spec/L2Pindex.html

```
HTML DTD Reference

Generated from
-//IETF//DTD HTML Recommended//EN//2.0

Alphabetical Index


A, ADDRESS, B, BASE, BLOCKQUOTE, BODY, BR, CITE, CODE, DD, DIR, DL, DT,
EM, FORM, H1, H2, H3, H4, H5, H6, HEAD, HR, HTML, I, IMG, INPUT, ISINDEX
, KBD, LI, LINK, MENU, META, NEXTID, OL, OPTION, P, PRE, SAMP, SELECT,
STRONG, TEXTAREA, TITLE, TT, UL, VAR,
------------------------------------------------------------------------

A

Required Parts
<A>characters... </A>
All Parts
<A HREF="..." NAME="..." REL="..." REV="..." URN="..." TITLE="..."
METHODS="..." >characters... <IMG> <BR> <EM> <STRONG> <CODE> <SAMP>
<KBD> <VAR> <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>
<VAR> <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> <VAR>
<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> <VAR>

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
<BR>
All Parts
<BR>
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> <VAR>

--------------------------------------------------------------------------

CITE

Required Parts
<CITE>characters... </CITE>
All Parts
<CITE>characters... <A> <IMG> <BR> <EM> <STRONG> <CODE> <SAMP> <KBD>
<VAR> <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> <VAR>

--------------------------------------------------------------------------

CODE

Required Parts
<CODE>characters... </CODE>
All Parts
<CODE>characters... <A> <IMG> <BR> <EM> <STRONG> <CODE> <SAMP> <KBD>
<VAR> <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> <VAR>

--------------------------------------------------------------------------

DD

Required Parts
<DD>characters...
All Parts
<DD>characters... <A> <IMG> <BR> <EM> <STRONG> <CODE> <SAMP> <KBD> <VAR>
 <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> <VAR>
<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> <VAR>
<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> <VAR>

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> <ISINDEX> <HR> <ADDRESS> <INPUT> <SELECT> <TEXTAREA> </FORM>

Allowed In Content Of...

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

H1

Required Parts

<H1>characters... </H1>

All Parts

<H1>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <VAR> <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> <VAR> <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> <VAR> <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> <VAR> <CITE> <TT> <B> <I> </H4>

Allowed In Content Of...

<BLOCKQUOTE> <BODY> <FORM>

```
--------------------------------------------------------------------------

H5

Required Parts
<H5>characters... </H5>
All Parts
<H5>characters... <A> <IMG> <BR> <EM> <STRONG> <CODE> <SAMP> <KBD> <VAR>
 <CITE> <TT> <B> <I> </H5>
Allowed In Content Of...
<BLOCKQUOTE> <BODY> <FORM>

--------------------------------------------------------------------------

H6

Required Parts
<H6>characters... </H6>
All Parts
<H6>characters... <A> <IMG> <BR> <EM> <STRONG> <CODE> <SAMP> <KBD> <VAR>
 <CITE> <TT> <B> <I> </H6>
Allowed In Content Of...
<BLOCKQUOTE> <BODY> <FORM>

--------------------------------------------------------------------------

HEAD

Required Parts
All Parts
<HEAD><TITLE> <ISINDEX> <BASE> <META> <NEXTID> <LINK> </HEAD>
Allowed In Content Of...
<HTML>

--------------------------------------------------------------------------

HR

Required Parts
<HR>
All Parts
<HR>
Allowed In Content Of...
<BLOCKQUOTE> <BODY> <FORM> <PRE>

--------------------------------------------------------------------------

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> <VAR>
<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> <VAR>

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...
<FORM>

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>
<VAR> <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> <VAR>

LI

Required Parts
<LI>characters...
All Parts
<LI>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <VAR>
<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>

META

Required Parts
<META CONTENT="..." >
All Parts

```
<META HTTP-EQUIV="..." NAME="..." CONTENT="..." >
Allowed In Content Of...
<HEAD>

------------------------------------------------------------------------

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> <BR> <EM> <STRONG> <CODE> <SAMP> <KBD> <VAR>
<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> <HR>
 <TT> <B> <I> <EM> <STRONG> <CODE> <SAMP> <KBD> <VAR> <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> <VAR> <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> <VAR>

SELECT

Required Parts
<SELECT NAME="..." ></SELECT>
All Parts
<SELECT NAME="..." SIZE="..." MULTIPLE><OPTION> </SELECT>
Allowed In Content Of...
<FORM>

STRONG

Required Parts
<STRONG>characters... </STRONG>
All Parts
<STRONG>characters... <A> <IMG>
 <EM> <STRONG> <CODE> <SAMP> <KBD> <VAR> <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> <VAR>

TEXTAREA

Required Parts
<TEXTAREA NAME="..." ROWS="..." COLS="..." >characters... </TEXTAREA>
All Parts
<TEXTAREA NAME="..." ROWS="..." COLS="..." >characters... </TEXTAREA>

Allowed In Content Of...
<FORM>

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> <VAR> <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> <VAR>

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> <VAR> <CITE> <TT> <B> <I> </VAR>
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> <VAR>

6.2 Recommended Level 0 Element Reference

This listing eliminates deprecated idioms. Consult this reference when generating new documents aimed at minimally conforming implementations.

This reference is available as hypertext at
http://www.hal.com/%7Econnolly/html-spec/L0Pindex.html

```
HTML DTD Reference

Generated from
-//IETF//DTD HTML Level 0 Recommended//EN//2.0

Alphabetical Index


A, ADDRESS, BASE, BLOCKQUOTE, BODY, BR, DD, DIR, DL, DT, H1, H2, H3, H4,
H5, H6, HEAD, HR, HTML, IMG, ISINDEX, LI, LINK, MENU, META, NEXTID, OL,
P, PRE, TITLE, UL,
------------------------------------------------------------------------

A

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

All Parts

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

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> <BR> </H3>
Allowed In Content Of...
<BLOCKQUOTE> <BODY>

------------------------------------------------------------------------

H4

Required Parts
<H4>characters... </H4>
All Parts
<H4>characters... <A> <IMG> <BR> </H4>
Allowed In Content Of...
<BLOCKQUOTE> <BODY>

------------------------------------------------------------------------

H5

Required Parts
<H5>characters... </H5>
All Parts
<H5>characters... <A> <IMG> <BR> </H5>
Allowed In Content Of...
<BLOCKQUOTE> <BODY>

------------------------------------------------------------------------

H6

Required Parts
<H6>characters... </H6>
All Parts
<H6>characters... <A> <IMG> <BR> </H6>
Allowed In Content Of...
<BLOCKQUOTE> <BODY>

------------------------------------------------------------------------

HEAD

Required Parts
All Parts
<HEAD><TITLE> <ISINDEX> <BASE> <META> <NEXTID> </HEAD>
Allowed In Content Of...
<HTML>

------------------------------------------------------------------------

HR
```

Required Parts
<HR>
All Parts
<HR>
Allowed In Content Of...
<BLOCKQUOTE> <BODY> <PRE>

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>

```
LINK

Required Parts
<LINK HREF="..." >
All Parts
<LINK HREF="..." >
Allowed In Content Of...

------------------------------------------------------------------------

MENU

Required Parts
<MENU></MENU>
All Parts
<MENU><LI> </MENU>
Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <LI>

------------------------------------------------------------------------

META

Required Parts
<META CONTENT="..." >
All Parts
<META HTTP-EQUIV="..." NAME="..." CONTENT="..." >
Allowed In Content Of...
<HEAD>

------------------------------------------------------------------------

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> <BR> </P>
Allowed In Content Of...
<BLOCKQUOTE> <BODY> <DD> <LI>

------------------------------------------------------------------------

PRE

Required Parts
<PRE>characters... </PRE>
All Parts
<PRE WIDTH="..." >characters... <A> <HR> <BR> </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>
```

7. Glossary

The HTML specification uses these words with precise meanings:

attribute

A syntactical component of an HTML element which is often used to specify a characteristic quality of an element, other than type or content.

document type definition (DTD)

A DTD is a collection of declarations (entity, element, attribute, link, map, etc.) in SGML syntax that defines the components and structures available for a class (type) of documents.

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 an end-tag.

HTML

HyperText Markup Language.

HTML user agent

Any tool used with HTML documents.

HTML document

A collection of information represented as a sequence of characters. An HTML document consists of data characters and markup. In particular, the markup describes a structure conforming to the HTML document type definition.

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

(document) instance

The document itself including the actual content with the actual markup. Can be a single document or part of a document instance set that follows the DTD.

markup

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

Multipurpose Internet Mail Extensions (MIME)

An extension to Internet email which provides the ability to transfer non-textual data, such as graphics, audio and fax. It is defined in RFC 1341.

representation

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

rendering

Formatting and presenting information.

SGML

Standard Generalized Markup Language is a data encoding that allows the information in documents to be shared — either by other document publishing systems or by applications for electronic delivery, configuration management, database management, inventory control, etc. Defined in 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 at ftp.ifi.uio.no/pub/SGML/SGMLS.

tag

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

URI

Universal Resource Identifiers (URIs) is the name for a generic WWW identifier. The URI specification simply defines the syntax for encoding arbitrary naming or addressing schemes, and has a list of such schemes. See also: http://info.cern.ch/hypertext/WWW/Addressing/Addressing.html

WWW

A hypertext-based, distributed information system created by researchers at CERN in Switzerland. Users may create, edit or browse hypertext documents. The clients and servers are freely available.See also: http://info.cern.ch/hypertext/WWW/TheProject.html

7.1 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 specification, undesirable results may occur in practice.

typical

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

8. 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 Standard 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 at ftp.ifi.uio.no/pub/SGML/SGMLS.

URI

Universal Resource Identifiers. Available by anonymous FTP as Postscript (info.cern.ch/pub/www/doc/url.ps) or text (info.cern.ch/pub/www/doc/url.txt)

WWW

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.

9. 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 Andreessen; Netscape Communications Corp; marca@mcom.com
- Paul Burchard; The Geometry Center, University of Minnesota; burchard@geom.umn.edu
- James Clark; jjc@jclark.com
- Daniel W. Connolly; HaL Computer Systems; connolly@hal.com
- Roy Fielding; University of California, Irvine; fielding@ics.uci.edu
- Peter Flynn; University College Cork, Ireland; pflynn@www.ucc.ie
- Jay Glicksman; Enterprise Integration Technology; jay@eit.com
- Paul Grosso; ArborText, Inc.; paul@arbortext.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
- Mike Knezovich; Spyglass, Inc.; mike@spyglass.com
- Tim Berners-Lee; CERN; timbl@info.cern.ch
- Tom Magliery; NCSA; mag@ncsa.uiuc.edu

- Murray Maloney; Toronto Development Centre, The Santa Cruz Operation (SCO); murray@sco.com
- Larry Masinter; Xerox Palo Alto Research Center; masinter@parc.xerox.com
- Karen Olson Muldrow; HaL Computer Systems; karen@hal.com
- Bill Perry, Spry, Inc., wmperry@spry.com
- Dave Raggett, Hewlett Packard, dsr@hplb.hpl.hp.com
- E. Corprew Reed; Cold Spring Harbor Laboratory; corp@cshl.org
- Yuri Rubinsky; SoftQuad, Inc.; yuri@sq.com
- Eric Schieler; Spyglass, Inc.; eschieler@spyglass.com
- Eric W. Sink; Spyglass, Inc.; eric@spyglass.com
- Stuart Weibel; OCLC Office of Research; weibel@oclc.org
- Chris Wilson; Spry, Inc.; cwilson@spry.com

10. AUTHOR'S ADDRESSES

Tim Berners-Lee
timbl@quag.lcs.mit.edu

Daniel W. Connolly
Hal Software Systems
3006A Longhorn Blvd.
Austin, TX 78758

phone: (512) 834-9962 extension 5010
fax: (512) 823-9963
URL: http://www.hal.com/~connolly
email: connolly@hal.com