Hypertext Markup Language
Internet Draft
Expires 13 January 1994

Tim Berners-Lee, CERN
Daniel Connolly, Atrium Technology Inc.
13 July 1993

Hypertext Markup Language

A Representation of Textual Information and Metainformation for Retrieval and Interchange

Status of this Document

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 working documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress".

Distribution of this document is unlimited. The document is a draft form of a standard for interchange of information on the network which is proposed to be registered as a MIME (RFC1341) content type. Please send comments to timbl@info.cern.ch or the discussion list www-talk@info.cern.ch.

This is version 1.2 of this draft. This document is available in hypertext on the World-Wide Web as http://info.cern.ch/hypertext/WWW/Ma

Abstract

HyperText Markup Language (HTML) can be used to represent

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

The World Wide Web (W3) initiative links related information throughout the globe. HTML provides one simple format for providing linked information, and all W3 compatible programs are required to be capable of handling HTML. W3 uses an Internet protocol (Hypertext Transfer Protocol, HTTP), which allows transfer representations to be negotiated between client and server, the result being returned in an extended MIME message. HTML is therefore just one, but an important one, of the representations used with W3.

HTML is proposed as a MIME content type.

HTML refers to the URL specification of RFCxxxx.

Implementations of HTML parsers and generators can be found in the various W3 servers and browsers, in the public domain W3 code, and may also be built using various public domain SGML parsers such as [SGMLS]. HTML is an SGML document type with fairly generic semantics appropriate for representing information from a wide range of applications. It is more generic than many specific SGML applications, but is still completely device-independent.

1. In this document

This document contains the following parts:

Vocabulary used in this document, degrees of imperative.

HTML and MIME with discussion of character sets.

HTML and SGML and the relationship between them, and Structured text : an introduction for beginners to SGML.

HTML Elements A list with description, example, and typical rendering.

HTML Entities Entities used to describe characters.

The HTML DTD The text of the SGML DTD for HTML

Link relationship values . A provisional list. Not part of the standard.

Registration Authority The authority for extending lists of valid vales.

References to related documents

Authors addresses Contact information.

table of contents

1.1 Vocabulary

This specification uses the words below with the precise meaning given.

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

Rendering The form of presentation to information to the human reader.

1.1.1 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 as "must"

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

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

1.1.2 Notes

Sections marked "Note:" are not mandatory parts of the specification but for guidance only.

1.1.3 Status of features

Mainstream All parsers must recognize these features. Features are mainstream unless otherwise mentioned.

Extra Standard HTML features which may safely be ignored by parsers. It is legal to ignore these, treat the contents as though the tags were not there. (e.g. EM, and any undefined elements)

Obsolete Not standard HTML. Parsers should implement these features as far as possible in order to preserve back-compatibility with previous versions of this specification.

2. HTML and MIME

The definition of the HTML content subtype is

MIME Type name	text
MIME subtype name:	html
Required parameters:	none
Optional parameters:	charset

2.1 Character sets

The base character set (the SGML BASESET) for HTML is ISO Latin-1. This is the set referred to by any numeric character references . The actual character set used in the representation of an HTML document may be ISO Latin 1, or its 7-bit subset which is ASCII. 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 W3 always allows 8 bit transfer.

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

3. Introduction

The HyperText Markup Language is defined in terms of the ISO Standard Generalized Markup Language []. SGML is a system for defining structured document types and markup languages to represent instances of those document types.

Every SGML document has three parts:

- An SGML declaration, which 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 a tag is </ and the maximum length of a name is 40 characters.
- A prologue including one or more document type declarations, which specifiy the element types, element relationships and attributes, and references that can be represented by markup. The HTML DTD specifies, for example, that the HEAD element contains at most one TITLE element.
- An instance, which contains the data and markup of the document.

We use the term HTML to mean both the document type and the markup language for representing instances of that document type.

All HTML documents share the same SGML declaration an prologue. Hence implementations of the WorldWide Web generally only transmit and store the instance part of an HTML document. To construct an SGML document entity for processing by an SGML parser, it is necessary to prefix the text from “HTML DTD” on page 10 to the HTML instance.

Conversely, to implement an HTML parser, one need only implement those parts of an SGML parser that are needed to parse an instance after parsing the HTML DTD.

3.1 Structured Text

An HTML instance is like a text file, except that some of the characters are interpreted as markup. The markup gives structure to the document.

The instance represents a hierarchy of elements. Each element has a name , some attributes , and some content. Most elements are represented in the document as a start tag, which gives the name and attributes, followed by the content, followed by the end tag. For example:

```
<HTML>
 <TITLE>
  A sample HTML instance
 </TITLE>
 <H1>
  An Example of Structure
 </H1>
 Here’s a typical paragraph.
 <P>
 <UL>
  <LI>
  Item one has an
  <A NAME="anchor">
   anchor
  </A>
  <LI>
  Here’s item two.
 </UL>
</HTML>
```

Some elements (e.g. P, LI) are empty. They have no content. They show up as just a start tag.

For the rest of the elements, the content is a sequence of data characters and nested elements. Note that the HTML DTD in fact severely limits the amount of nesting which is allowed: most things cannot be nested, in fact. No elements may be recursively nested. Anchors and character highlighting may be put inside other constructs.

3.1.1 Tags

Every element starts with a tag, and every non-empty element ends with a tag. Start tags are delimited by < and >, and end tags are delimited by </ and >.

Names

The element name immediately follows the tag open delimiter. Names consist of a letter followed by up to 33 letters, digits, periods, or hyphens. Names are not case sensitive.

Attributes

In a start tag, whitespace and attributes are allowed between the element name and the closing delimiter. An attribute consists of a name, an equal sign, and a value. Whitespace is allowed around the equal sign.

The value is specified in a string surrounded by single quotes or a string surrounded by double quotes. (See: other tolerated forms @@)

The string is parsed like RCDATA (see below) to determine the attribute value. This allows, for example, quote characters in attribute values to be represented by character references.

The length of an attribute value (after parsing) is limited to 1024 characters.

3.1.2 Element Types

The name of a tag refers to an element type declaration in the HTML DTD. An element type declaration associates an element name with

- A list of attributes and their types and statuses
- A content type (one of EMPTY, CDATA, RCDATA, ELEMENT, or MIXED) which determines the syntax of the element's content
- A content model, which specifies the pattern of nested elements and data

Empty Elements

Empty elements have the keyword EMPTY in their declaration. For example:

```
<!ELEMENT NEXTID - O EMPTY>
<!ATTLIST NEXTID N NUMBER #REQUIRED>
```

This means that the following:

```
<nextid n=’’27’’>
```

is legal, but these others are not:

```
<nextid>
<nextid n=’’abc’’>
```

Character Data

The keyword CDATA indicates that the content of an element is character data. Character data is all the text up to the next end tag open delimiter-in-context. For example:

```
<!ELEMENT XMP - - CDATA>
```

specifies that the following text is a legal XMP element:

```
<xmp>Here's an example. It looks like it has
<tags> and <!--comments-->
in it, but it does not. Even this
</ is data.</xmp>
```

The string </ is only recognized as the opening delimiter of an end tag when it is "in context," that is, when it is followed by a letter. However, as soon as the end tag open delimiter is recognized, it terminates the CDATA content. The following is an error:

```
<xmp>There is no way to represent </end> tags
in CDATA </xmp>
```

Replaceable Character Data

Elements with RCDATA content behave much like those with CDATA, except for character references and entity references. Elements declared like:

```
<!ELEMENT TITLE - - RCDATA>
```

can have any sequence of characters in their content.

Character References To represent a character that would otherwise be recognized as markup, use a character reference. The string &# signals a character reference when it is followed by a letter or a digit. The delimiter is followed by the decimal character number and a semicolon. For example:

```
<title>You can even represent &#60;/end> tags in RCDATA </title>
```

Entity References The HTML DTD declares entities for the less than, greater than, and ampersand characters and each of the ISO Latin 1 characters so that you can reference them by name rather than by number.

The string & signals an entity reference when it is followed by a letter or a digit. The delimiter is followed by the entity name and a semicolon. For example:

```
Kurt G&ouml;del was a famous logician and mathematician.
```

Note: To be sure that a string of characters has no markup, HTML writers should represent all occurrences of <, >, and & by character or entity references.

Element Content

Some elements have, in stead of a keyword that states the type of content, a content model, which tells what patterns of data and nested elements are allowed. If the content model of an element does not include the symbol #PCDATA , the content is element content.

Whitespace in element content is considered markup and ignored. Any characters that are not markup, that is, data characters, are illegal.

For example:

```
<!ELEMENT HEAD - - (TITLE? & ISINDEX? & NEXTID? & LINK*)>
```

declares an element that may be used as follows:

```
<head>
 <isindex>
 <title>Head Example</title>
</head>
```

But the following are illegal:

```
<head> no data allowed! </head>
<head><isindex><title>Two isindex tags</title><isindex></head>
```

Mixed Content

If the content model includes the symbol #PCDATA, the content of the element is parsed as mixed content. For example:

```
<!ELEMENT PRE - - (#PCDATA | A | B | I | U | P)+>
<!ATTLIST PRE
WIDTH NUMBER #implied
>
```

This says that the PRE element contains one or more A, B, I, U, or P elements or data characters. Here's an example of a PRE element:

```
<pre>
<b>NAME</b>
    cat -- concatenate<a href=''terms.html#file''>files</a>
<b>EXAMPLE</b>
    cat <xyz
</pre>
```

The content of the above PRE element is:

- A B element
- The string " cat –– concatenate"
- An A element
- The string "\n"
- Another B element
- The string "\n cat <xyz"

3.1.3 Comments and Other Markup

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

For example:

```
<HEAD>
<TITLE>HTML Guide: Recommended Usage</TITLE>
<!-- $Id: recommended.html,v 1.3 93/01/06 18:38:11 connolly Exp $ -->
</HEAD>
```

There are a few other SGML markup constructs that are deprecated or illegal.

Delimiter	Signals...
<**?**	Processing instruction. Terminated by >.
<**![**	Marked section. Marked sections are deprecated. See the SGML standard for complete information.
<**!**	Markup declaration. HTML defines no short reference maps, so these are errors. Terminated by >.

3.1.4 Line Breaks

A line break character is considered markup (and ignored) if it is the first or last piece of content in an element. This allows you to write either

```
<PRE>some example text</pre>
```

or

```
<pre>
some example text
</pre>
```

and these will be processed identically.

Also, a line that's not empty but contains no content will be ignored altogether. For example, the element

```
<pre>
<!-- this line is ignored, including the linebreak character -->
first line

third line<!-- the following linebreak is content: -->
fourth line<!-- this one's ignored because it's the last piece of content: -->
</pre>
```

contains only the strings

```
 first line

 third line
 fourth line.
```

3.1.5 Spaces and Tabs

Space characters must be rendered as horizontal white space. In HTML, multiple spaces should be rendered as proportionally larger spaces.

The rendering of a horizontal tab (HT) character is not defined, and HT should therefore not be used, except within a PRE (or obsolete XMP, LISTING or PLAINTEXT) element.

Neither spaces nor tabs should be used to make SGML source layout more attractive or easier to read.

3.1.6 Summary of Markup Signals

The following delimiters may signal markup, depending on context.

Delimiter	Signals
<!−−	Comment
&#	Character reference
&	Entity reference
</	End tag
<!	Markup declaration
]]>	Marked section close (an error)
<	Start tag

4. HTML Elements

This is a list of elements used in the HTML language. Documents should (but need not absolutely) contain an initial HEAD element followed by a BODY element.
Old style documents may contain a just the contents of the normal HEAD and BODY elements, in any order. This is deprecated but must be supported by parsers.
See also: Status of elements

4.1 Properties of the whole document

Properties of the whole document are defined by the following elements. They should appear within the HEAD element. Their order is not significant.

TITLE	The title of the document
ISINDEX	Sent by a server in a searchable document
NEXTID	A parameter used by editors to generate unique identifiers
LINK	Relationship between this document and another. See also the Anchor element , Relationships . A document may have many LINK elements.
BASE	A record of the URL of the document when saved

4.2 Text formatting

These are elements which occur within the BODY element of a document. Their order is the logical order in which the elements should be rendered on the output device.

Headings	Several levels of heading are supported.
Anchors	Sections of text which form the beginning and/or end of hypertext links are called "anchors" and defined by the A tag.
Paragraph marks	The P element marks the break between two paragraphs.
Address style	An ADDRESS element is displayed in a particular style.
Blockquote style	A block of text quoted from another source.
Lists	Bulleted lists, glossaries, etc.
Preformatted text	Sections in fixed-width font for preformatted text.
Character highlighting	Formatting elements which do not cause paragraph breaks.

4.3 Graphics

IMG	The IMG tag allows inline graphics.

4.4 Obsolete elements

The other elements are obsolete but should be recognised by parsers for back-compatibility.

4.5 HEAD

The HEAD element contains all information about the document in general. It does not contain any text which is part of the document: this is in the BODY. Within the head element, only certain elements are allowed.

4.6 BODY

The BODY element contains all the information which is part of the document, as opposed information about the document which is in the HEAD .
The elements within the BODY element are in the order in which they should be presented to the reader.
See the list of things which are allowed within a BODY element .

4.7 Anchors

An anchor is a piece of text which marks the beginning and/or the end of a hypertext link.

The text between the opening tag and the closing tag is either the start or destination (or both) of a link. Attributes of the anchor tag are as follows.

HREF OPTIONAL. If the HREF attribute is present, the anchor is sensitive text: the start of a link. If the reader selects this text, (s)he should be presented with another document whose network address is defined by the value of the HREF attribute . The format of the network address is specified elsewhere . This allows for the form HREF="#identifier" to refer to another anchor in the same document. If the anchor is in another document, the attribute is a relative name , relative to the documents address (or specified base address if any).

NAME OPTIONAL. If present, the attribute NAME allows the anchor to be the destination of a link. The value of the attribute is an identifier for the anchor. Identifiers are arbitrary strings but must be unique within the HTML document. Another document can then make a reference explicitly to this anchor by putting the identifier after the address, separated by a hash sign .

REL OPTIONAL. An attribute REL may give the relationship (s) described by the hypertext link. The value is a comma-separated list of relationship values. Values and their semantics will be registered by the HTML registration authority . The default relationship if none other is given is void. REL should not be present unless HREF is present. See Relationship values , REV .

REV OPTIONAL. The same as REL , but the semantics of the link type are in the reverse direction. A link from A to B with REL="X" expresses the same relationship as a link from B to A with REV="X". An anchor may have both REL and REV attributes.

URN OPTIONAL. If present, this specifies a uniform resource number for the document. See note .

TITLE OPTIONAL. This is informational only. If present the value of this field should equal the value of the TITLE of the document whose address is given by the HREF attribute. See note .

METHODS OPTIONAL. The value of this field is a string which if present must be a comma separated list of HTTP METHODS supported by the object for public use. See note .

All attributes are optional, although one of NAME and HREF is necessary for the anchor to be useful. See also: LINK .

4.7.1 Example of use:

```
See <A HREF="http://info.cern.ch/">CERN</A>'s information for
more details.

A <A NAME=serious>serious</A> crime is one which is associated
with imprisonment.
...
The Organization may refuse employment to anyone convicted
of a <a href="#serious">serious</A> crime.
```

4.7.2 Note : Universal Resource Numbers

URNs are provided to allow a document to be recognized if duplicate copies are found. This should save a client implementation from picking up a copy of something it already has.

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

4.7.3 Note: TITLE attribute of links

The link may carry a TITLE attribute which should if present give the title of the document whose address is given by the HREF attribute.
This is useful for at least two reasons

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

4.7.4 Note: METHODS attribute of Links

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

4.8 Address

This element is for address information, signatures, authorship, etc, often at the top or bottom of a document.

4.8.1 Typical rendering

Typically, an address element is italic and/or right justified or indented. The address element implies a paragraph break. Paragraph marks within the address element do not cause extra white space to be inserted.

4.8.2 Examples of use:

```
<ADDRESS><A HREF="Author.html">A.N.Other</A></ADDRESS>
```

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

4.9 BASE

This 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 reader will use the URL it used to access the document to resolve any relative URLs.
The one attribute is:

HREF the URL

4.10 BLOCKQUOTE

The BLOCKQUOTE element allows text quoted from another source to be rendered specially.

4.10.1 Typical rendering

A typical rendering might be a slight extra left and right indent, and/or italic font. BLOCKQUOTE causes a paragraph break, and typically a line or so of white space will be allowed between it and any text before or after it.

Single-font rendition may for example put a vertical line of ">" characters down the left margin to indicate quotation in the Internet mail style.

4.10.2 Example

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

4.11 Headings

Six levels of heading are supported. (Note that a hypertext node within a hypertext work tends to need less levels of heading than a work whose only structure is given by the nesting of headings.)

A heading element implies all the font changes, paragraph breaks before and after, and white space (for example) necessary to render the heading. Further character emphasis or paragraph marks are not required in HTML.

H1 is the highest level of heading, and is recommended for the start of a hypertext node. It is suggested that the the text of the first heading be suitable for a reader who is already browsing in related information, in contrast to the title tag which should identify the node in a wider context.

The heading elements are

```
<H1>, <H2>, <H3>, <H4>, <H5>, <H6>
```

It is not normal practice to jump from one header to a header level more than one below, for example for follow an H1 with an H3. Although this is legal, it is discouraged, as it may produce strange results for example when generating other representations from the HTML.

4.11.1 Example:

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

4.11.2 Parser Note:

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

4.11.3 Typical Rendering

H1 Bold very large font, centered. One or two lines clear space between this and anything following. If printed on paper, start new page.

H2 Bold, large font,, flush left against left margin, no indent. One or two clear lines above and below.

H3 Italic, large font, slightly indented from the left margin. One or two clear lines above and below.

H4 Bold, normal font, indented more than H3. One clear line above and below.

H5 Italic, normal font, indented as H4. One clear line above.

H6 Bold, indented same as normal text, more than H5. One clear line above.

These typical values are just an indication, and it is up to the designer of the presentation software to define the styles. The reader may have options to customize these. When writing documents, you should assume that whatever is done it is designed to have the same sort of effect as the styles above.

The rendering software is responsible for generating suitable vertical white space between elements, so it is NOT normal or required to follow a heading element with a paragraph mark.

4.12 IMG: Embedded Images

Status: Extra

The IMG element allows another document to be inserted inline. The document is normally an icon or small graphic, etc. This element is NOT intended for embedding other HTML text.

Browsers which are not able to display inline images ignore IMG elements. Authors should note that some browsers will be able to display (or print) linked graphics but not inline graphics. If the graphic is essential, it may be wiser to make a link to it rather than to put it inline. If the graphic is essentially decorative, then IMG is appropriate.

The IMG element is empty: it has no closing tag. It has two attributes:

SRC The value of this attribute is the URL of the document to be embedded. Its syntax is the same as that of the HREF attribute of the A tag. SRC is mandatory.

ALIGN Take values TOP or MIDDLE or BOTTOM, defining whether the tops or middles of bottoms of the graphics and text should be aligned vertically.

Note that IMG elements are allowed within anchors.

4.12.1 Example

```
Warning: < IMG SRC ="triangle.gif"> This must be done by a
qualified technician.

< A HREF="Go">< IMG SRC ="Button"> Press to start</A>
```

4.13 ISINDEX

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

The node may be queried with a keyword search by suffixing the node address with a question mark, followed by a list of keywords separated by plus signs. See the network address format .

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

Status: standard.

4.13.1 Example of use:

```
<ISINDEX>
```

4.14 LINK

The LINK element occurs within the HEAD element of an HTML document. It is used to indicate a relationship between the document and some other object. A document may have any number of LINK elements.

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

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

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

4.15 Forms of list in HTML

4.15.1 Glossaries

A glossary (or definition list) is a list of paragraphs each of which has a short title alongside it. Apart from glossaries, this element is useful for presenting a set of named elements to the reader. The elements within a glossary follow are

DT The "term", typically placed in a wide left indent

DD The "definition", which may wrap onto many lines

These elements must appear in pairs. Single occurrences of DT without a following DD are illegal. The one attribute which DL can take is

COMPACT suggests that a compact rendering be used, because the enclosed elements are individually small, or the whole glossary is rather large, or both.

Typical rendering

The definition list DT, DD pairs are arranged vertically. For each pair, the DT element is on the left, in a column of about a third of the display area, and the DD element is in the right hand two thirds of the display area. The DT term is normally small enough to fit on one line within the left-hand column. If it is longer, it will either extend across the page, in which case the DD section is moved down to separate them, or it is wrapped onto successive lines of the left hand column.

White space is typically left between successive DT,DD pairs unless the COMPACT attribute is given. The COMPACT attribute is appropriate for lists which are long and/or have DT,DD pairs which each take only a line or two. It is of course possible for the rendering software to discover these cases itself and make its own decisions, and this is to be encouraged.

The COMPACT attribute may also reduce the width of the left-hand (DT) column.

Examples of use

```
<DL>
<DT>Term the first<DD>definition paragraph is reasonably
long but is still displayed clearly
<DT>Term2 follows<DD>Definition of term2
</DL>

<DL COMPACT>
<DT>Term<DD>definition paragraph
<DT>Term2<DD>Definition of term2
</DL>
```

4.15.2 Lists

A list is a sequence of paragraphs, each of which may be preceded by a special mark or sequence number. The syntax is:

```
<UL>
<LI> list element
<LI> another list element ...
</UL>
```

The opening list tag may be any of UL, OL, MENU or DIR. It must be immediately followed by the first list element.

Typical rendering

The representation of the list is not defined here, but a bulleted list for unordered lists, and a sequence of numbered paragraphs for an ordered list would be quite appropriate. Other possibilities for interactive display include embedded scrollable browse panels.

List elements with typical rendering are:

UL A list of multi-line paragraphs, typically separated by some white space and/or marked by bullets, etc.

OL As UL, but the paragraphs are typically numbered in some way to indicate the order as significant.

MENU A list of smaller paragraphs. Typically one line per item, with a style more compact than UL.

DIR A list of short elements, typically less than 20 characters. These may be arranged in columns across the page, typically 24 character in width. If the rendering software is able to optimize the column width as function of the widths of individual elements, so much the better.

Example of use

```
<OL>
<LI> When you get to the station, leave
by the southern exit, on platform one.
```

```
<LI>Turn left to face toward the mountain
<LI>Walk for a mile or so until you reach the
"Asquith Arms" then
<LI>Wait and see...
</OL>

< MENU >
<LI>The oranges should be pressed fresh
<LI>The nuts may come from a packet
<LI>The gin must be good quality
</MENU>

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

4.16 Next ID

This tag takes a single attribute which is the number of the next document-wide numeric identifier to be allocated of the form z123.

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

4.16.1 Example of use:

```
<NEXTID N=27>
```

4.17 P: Paragraph mark

The empty P element indicates a paragraph break. The exact rendering of this (indentation, leading, etc) is not defined here, and may be a function of other tags, style sheets etc.

<P> is used between two pieces of text which otherwise would be flowed together.

You do NOT need to use <P> to put white space around heading, list, address or blockquote elements which imply a paragraph break. It is the responsibility of the rendering software to generate that white space. A paragraph mark which is preceded or followed by such elements which imply a paragraph break is has undefined effect and should be avoided.

4.17.1 Typical rendering

Typically, <P> will generate a small vertical space (of a line or half a line) between the paragraphs. This is not the case (typically) within ADDRESS or (ever) within PRE elements. With some implementations, in normal text, <P> may generate a small extra left indent on the first line.

4.17.2 Examples of use

```
<h1>What to do</h1>
This is a one paragraph.< p >This is a second.
< P >
This is a third.
```

4.17.3 Bad example

```
        <h1><P>What not to do</h1>
<p>I found that on my XYZ browser it looked prettier to
me if I put some paragraph marks
<p>
<ul><p><li>Around lists, and
<li>After headings.
</ul>
<p>
None of the paragraph marks in this example should
be there.
```

4.18 PRE: Preformatted text

Preformatted elements in HTML are displayed with text in a fixed width font, and so are suitable for text which has been formatted for a teletype by some existing formatting system.

The optional attribute is:

WIDTH This attribute gives the maximum number of characters which will occur on a line. It allows the presentation system to select a suitable font and indentation. Where the WIDTH attribute is not recognized, it is recommended that a width of 80 be assumed. Where WIDTH is supported, it is recommended that at least widths of 40, 80 and 132 characters be presented optimally, with other widths being rounded up.

Within a PRE element,

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

Example of use

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

Note: Highlighting

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

Note: Margins

The above references to the "beginning of a new line" must not be taken as implying that the renderer is forbidden from using a (constant) left indent for rendering preformatted text. The left indent may of course be constrained by the width required.

4.19 TITLE

The title of a document is specified by the TITLE element. The TITLE element should occur in the HEAD of the document.

There may only be one title in any document. It should identify the content of the document in a fairly wide context.

The title is not part of the text of the document, but is a property of the whole document. It may not contain anchors, paragraph marks, or highlighting. The title may be used to identify the node in a history list, to label the window displaying the node, etc. It is not normally displayed in the text of a document itself. Contrast titles with headings . The title should ideally be less than 64 characters in length. That is, many applications will display document titles in window titles, menus, etc where there is only limited room. Whilst there is no limit on the length of a title (as it may be automatically generated from other data), information providers are warned that it may be truncated if long.

Examples of use

Appropriate titles might be

```
<TITLE>Rivest and Neuman. 1989(b)</TITLE>
```

or

```
<TITLE>A Recipe for Maple Syrup Flap-Jack</TITLE>
```

or

```
<TITLE>Introduction -- AFS user's Guide</TITLE>
```

Examples of inappropriate titles are those which are only meaningful within context,

```
<TITLE>Introduction</TITLE>
```

or too long,

```
<TITLE>Remarks on the Quantum-Gravity effects of "Bean
Pole" diversification in Mononucleosis patients in Developing
Countries under Economic Conditions Prevalent during
the Second half of the Twentieth Century, and Related Papers:
a Summary</TITLE>
```

4.20 Character highlighting

Status: Extra

These elements allow sections of text to be formatted in a particular way, to provide emphasis, etc. The tags do NOT cause a paragraph break, and may be used on sections of text within paragraphs.

Where not supported by implementations, like all tags, these tags should be ignored but the content rendered.

All these tags have related closing tags, as in

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

Some of these styles are more explicit than others about how they should be physically represented. The logical styles should be used wherever possible, unless for example it is necessary to refer to the formatting in the text. (Eg, "The italic parts are mandatory".)

Note:

Browsers unable to display a specified style may render it in some alternative, or the default, style, with some loss of quality for the reader. Some implementations may ignore these tags altogether, so information providers should attempt not to rely on them as essential to the information content.

These element names are derived from TeXInfo macro names.

4.20.1 Physical styles

TT	Fixed-width typewriter font.
B	Boldface, where available, otherwise alternative mapping allowed.
I	Italic font (or slanted if italic unavailable).
U	Underline.

4.20.2 Logical styles

EM	Emphasis, typically italic.
STRONG	Stronger emphasis, typically bold.
CODE	Example of code. typically monospaced font. (Do not confuse with PRE)
SAMP	A sequence of literal characters.
KBD	in an instruction manual, Text typed by a user.
VAR	A variable name.
DFN	The defining instance of a term. Typically bold or bold italic.
CITE	A citation. Typically italic.

4.20.3 Examples of use

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

4.21 Obsolete elements

The following elements of HTML are obsolete. It is recommended that client implementors implement the obsolete forms for compatibility with old servers.

Plaintext

Status: Obsolete .

The empty PLAINTEXT tag terminates the HTML entity. What follows is not SGML. In stead, there's an old HTTP convention that what follows is an ASCII (MIME "text/plain") body.

An example if its use is:

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

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

XMP and LISTING: Example sections

Status: Obsolete . This are in use and should be recognized by browsers. New servers should use <PRE> instead.

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

```
<LISTING>
...
</LISTING>
```

or

```
<XMP>
...
</XMP>
```

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

- The text may contain any ISO Latin printable characters, but not the end tag opener. (See Historical note)

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

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

Highlighted Phrase HP1 etc

Status: Obsolete . These tags like all others should be ignored if not implemented. Replaced will more meaningful elements –– see character highlighting .

Examples of use:

```
<HP1>...</HP1>   <HP2>... </HP2> etc.
```

Comment element

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

4.21.1 Historical Note: XMP and LISTING

The XMP and LISTING elements used historically to have non SGML conforming specifications, in that the text could contain any ISO Latin printable characters, including the tag opener, so long as it does not contain the closing tag in full.

This form is not supported by SGML and so is not the specified HTML interpretation. Providers should be warned that implementations may vary on how they interpret end tags apparently within these elements

5. Entities

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.

<	The less than sign <
>	The "greater than" sign >
&	The ampersand sign & itself.
"	The double quote sign "

Also allowed are references to any of the ISO Latin-1 alphabet, using the entity names in the following table.

5.1 ISO Latin 1 character entities

This list is derived from "ISO 8879:1986//ENTITIES Added Latin 1//EN".

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

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

6. The HTML DTD

The HTML DTD follows . Its relationship to the content of an SGML document is explained in the section "HTML and SGML" .

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

NOTE: This is a definition of HTML with respect to
SGML, and assumes an understanding of SGML terms.
--

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 95 32
               255  1 UNUSED


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 31 127 255
         BASESET  "ISO 646:1983//CHARSET
                   International Reference Version (IRV)//ESC 2/5 4/0"
         DESCSET  0 128 0
         FUNCTION 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  34
                  TAGLVL   100
                  LITLEN   1024
                  GRPGTCNT 150
                  GRPCNT   64

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

<!DOCTYPE HTML [
<!-- Jul 1 93 -->
<!-- Regarding clause 6.1, SGML Document:

[1] SGML document = SGML document entity,
    (SGML subdocument entity |
    SGML text entity | non-SGML data entity)*

The role of SGML document entity is filled by this DTD,
followed by the conventional HTML data stream.
-->

<!-- DTD definitions -->

<!ENTITY % heading "H1|H2|H3|H4|H5|H6" >
<!ENTITY % list "UL|OL|DIR|<A
NAME=z29 HREF="Lists.html#z36">MENU">
<!ENTITY % literal "XMP|LISTING">

<!ENTITY % headelement
 "TITLE|NEXTID|ISINDEX" >

<!ENTITY % bodyelement
 "P | %heading |
 %list | DL | HEADERS | ADDRESS | PRE | BLOCKQUOTE
```

```
| %literal">

<!ENTITY % oldstyle "%headelement | %bodyelement | #PCDATA">

<!ENTITY % URL "CDATA"
-- The term URL means a CDATA attribute
   whose value is a Uniform Resource Locator,
   as defined. (A URN may also be usable here when defined.)
-->

<!ENTITY % linkattributes
        "NAME NMTOKEN #IMPLIED
HREF %URL; #IMPLIED
        REL CDATA #IMPLIED -- forward relationship type --
REV CDATA #IMPLIED -- reversed relationship type
                              to referent data:

PARENT CHILD, SIBLING, NEXT, TOP,
DEFINITION, UPDATE, ORIGINAL etc. --

URN CDATA #IMPLIED -- universal resource number --

TITLE CDATA #IMPLIED -- advisory only --

METHODS NAMES #IMPLIED -- supported public methods of the object:
TEXTSEARCH, GET, HEAD, ... --

">


<!-- Document Element -->

<!ELEMENT HTML O O  (( HEAD | BODY | %oldstyle)*, PLAINTEXT?)>

<!ELEMENT HEAD - -  (TITLE? & ISINDEX? & NEXTID? & LINK*
                              & BASE?)>

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

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

<!ELEMENT NEXTID - O EMPTY>
<!ATTLIST NEXTID N NAME #REQUIRED
  -- The number should be a name suitable for use
     for the ID of a new element. When used, the value
```

```
     has its numeric part incremented. EG Z67 becomes Z68
  -->
<!ELEMENT LINK - O EMPTY>
<!ATTLIST LINK
        %linkattributes>

<!ELEMENT BASE - O EMPTY    -- Reference context for URLS -->
<!ATTLIST BASE

HREF %URL; #IMPLIED

        >
<!ENTITY % inline "EM | TT | STRONG | B | I | U |
CODE | SAMP | KBD | KEY | VAR | DFN | CITE "
>

<!ELEMENT (%inline;) - - (#PCDATA)>

<!ENTITY % text "#PCDATA | IMG | %inline;">

<!ENTITY % htext "A | %text">

<!ELEMENT BODY - -  (%bodyelement|%htext;)*>


<!ELEMENT A - -  (%text)>
<!ATTLIST A
%linkattributes;
>

<!ELEMENT IMG    - O EMPTY --  Embedded image -->
<!ATTLIST IMG
        SRC %URL; #IMPLIED -- URL of document to embed --
>


<!ELEMENT P - O EMPTY -- separates paragraphs -->

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

<!ELEMENT DL - -  (DT | DD | P | %htext;)*>
<!-- Content should match ((DT,(%htext;)+)+,(DD,(%htext;)+))
But mixed content is messy.
  -->

<!ELEMENT DT - O EMPTY>
<!ELEMENT DD - O EMPTY>

<!ELEMENT (UL|OL) - -  (%htext;|LI|P)+>
<!ELEMENT (DIR|MENU) - -  (%htext;|LI)+>
<!--    Content should match ((LI,(%htext;)+)+)
        But mixed content is messy.
```

```
  -->
<!ATTLIST (%list)
COMPACT NAME #IMPLIED -- COMPACT, etc.--
>

<!ELEMENT LI - O EMPTY>

<!ELEMENT BLOCKQUOTE - - (%htext;|P)+
-- for quoting some other source -->

<!ELEMENT ADDRESS - - (%htext;|P)+>

<!ELEMENT PRE - - (#PCDATA|%inline|A|P)+>
<!ATTLIST PRE
WIDTH NUMBER #implied
        >

<!-- Mnemonic character entities. -->
<!ENTITY AElig "&#198;" -- capital AE diphthong (ligature) -->
<!ENTITY Aacute "&#193;" -- capital A, acute accent -->
<!ENTITY Acirc "&#194;" -- capital A, circumflex accent -->
<!ENTITY Agrave "&#192;" -- capital A, grave accent -->
<!ENTITY Aring "&#197;" -- capital A, ring -->
<!ENTITY Atilde "&#195;" -- capital A, tilde -->
<!ENTITY Auml "&#196;" -- capital A, dieresis or umlaut mark -->
<!ENTITY Ccedil "&#199;" -- capital C, cedilla -->
<!ENTITY ETH "&#208;" -- capital Eth, Icelandic -->
<!ENTITY Eacute "&#201;" -- capital E, acute accent -->
<!ENTITY Ecirc "&#202;" -- capital E, circumflex accent -->
<!ENTITY Egrave "&#200;" -- capital E, grave accent -->
<!ENTITY Euml "&#203;" -- capital E, dieresis or umlaut mark -->
<!ENTITY Iacute "&#205;" -- capital I, acute accent -->
<!ENTITY Icirc "&#206;" -- capital I, circumflex accent -->
<!ENTITY Igrave "&#204;" -- capital I, grave accent -->
<!ENTITY Iuml "&#207;" -- capital I, dieresis or umlaut mark -->
<!ENTITY Ntilde "&#209;" -- capital N, tilde -->
<!ENTITY Oacute "&#211;" -- capital O, acute accent -->
<!ENTITY Ocirc "&#212;" -- capital O, circumflex accent -->
<!ENTITY Ograve "&#210;" -- capital O, grave accent -->
<!ENTITY Oslash "&#216;" -- capital O, slash -->
<!ENTITY Otilde "&#213;" -- capital O, tilde -->
<!ENTITY Ouml "&#214;" -- capital O, dieresis or umlaut mark -->
<!ENTITY THORN "&#222;" -- capital THORN, Icelandic -->
<!ENTITY Uacute "&#218;" -- capital U, acute accent -->
<!ENTITY Ucirc "&#219;" -- capital U, circumflex accent -->
<!ENTITY Ugrave "&#217;" -- capital U, grave accent -->
<!ENTITY Uuml "&#220;" -- capital U, dieresis or umlaut mark -->
<!ENTITY Yacute "&#221;" -- capital Y, acute accent -->
<!ENTITY aacute "&#225;" -- small a, acute accent -->
<!ENTITY acirc "&#226;" -- small a, circumflex accent -->
<!ENTITY aelig "&#230;" -- small ae diphthong (ligature) -->
<!ENTITY agrave "&#224;" -- small a, grave accent -->
```

```
<!ENTITY amp "&amp;" -- ampersand -->
<!ENTITY aring "&#229;" -- small a, ring -->
<!ENTITY atilde "&#227;" -- small a, tilde -->
<!ENTITY auml "&#228;" -- small a, dieresis or umlaut mark -->
<!ENTITY ccedil "&#231;" -- small c, cedilla -->
<!ENTITY eacute "&#233;" -- small e, acute accent -->
<!ENTITY ecirc "&#234;" -- small e, circumflex accent -->
<!ENTITY egrave "&#232;" -- small e, grave accent -->
<!ENTITY eth "&#240;" -- small eth, Icelandic -->
<!ENTITY euml "&#235;" -- small e, dieresis or umlaut mark -->
<!ENTITY gt "&#62;" -- greater than -->
<!ENTITY iacute "&#237;" -- small i, acute accent -->
<!ENTITY icirc "&#238;" -- small i, circumflex accent -->
<!ENTITY igrave "&#236;" -- small i, grave accent -->
<!ENTITY iuml "&#239;" -- small i, dieresis or umlaut mark -->
<!ENTITY lt "&lt;" -- less than -->
<!ENTITY ntilde "&#241;" -- small n, tilde -->
<!ENTITY oacute "&#243;" -- small o, acute accent -->
<!ENTITY ocirc "&#244;" -- small o, circumflex accent -->
<!ENTITY ograve "&#242;" -- small o, grave accent -->
<!ENTITY oslash "&#248;" -- small o, slash -->
<!ENTITY otilde "&#245;" -- small o, tilde -->
<!ENTITY ouml "&#246;" -- small o, dieresis or umlaut mark -->
<!ENTITY szlig "&#223;" -- small sharp s, German (sz ligature) -->
<!ENTITY thorn "&#254;" -- small thorn, Icelandic -->
<!ENTITY uacute "&#250;" -- small u, acute accent -->
<!ENTITY ucirc "&#251;" -- small u, circumflex accent -->
<!ENTITY ugrave "&#249;" -- small u, grave accent -->
<!ENTITY uuml "&#252;" -- small u, dieresis or umlaut mark -->
<!ENTITY yacute "&#253;" -- small y, acute accent -->
<!ENTITY yuml "&#255;" -- small y, dieresis or umlaut mark -->

<!-- deprecated elements -->

<!ELEMENT (%literal) - -  CDATA>

<!ELEMENT PLAINTEXT - O EMPTY>

<!-- Local Variables: -->
<!-- mode: sgml -->
<!-- compile-command: "sgmls -s -p " -->
<!-- end: -->
]>
```

7. Link Relationship values

Status: This list is not part of the standard. It is intended to illustrate the use of link relationships and to provide a framework for further development.

Additions to this list will be controlled by the HTML registration authority . Experimental values may be used on the condition that they begin with "X-".

These values of the REL attribute of hypertext links have a significance defined here, and may be treated in special ways by HTML applications.

These relationships relate whole documents (objects), rather than particular anchors within them. If the relationship value is used with a link between anchors rather than whole documents, the semantics are considered to apply to the documents.

In the explanations which follows, A is the source document of the link and B is the destination document specified by the HREF attribute.

A relationship marked "Acyclic" has the property that no sequence of links with that relationship may be followed from any document back to itself. These types of links may therefore be used to define trees.

7.1 Relationships between documents

These relationships are between the documents themselves rather than the subjects of the documents.

7.1.1 UseIndex

B is a related index for a search by a user reading this document who asks for an index search function.

A document may have any number of index links, causing several indexes top be searched in a client-defined manner.

B must support SEARCH operations under its access protocol.

7.1.2 UseGlossary

B is an index which should be used to resolve glossary queries in the document. (Typically, a double-click on a word which is not within an anchor).

A document may have any number of glossary links.

7.1.3 Annotation

The information in B is additional to and subsidiary to that in A.

Annotation is used by one person to write the equivalent of "margin notes" or other criticism on another's document, for example.

Example: The relationship between a newsgroup and its articles.

Acyclic.

7.1.4 Reply

Similar to Annotation, but there is no suggestion that B is subsidiary to A: A and B are on equal footings.

Example: The relationship between a mail message and its reply, a news article and its reply.

Acyclic.

7.1.5 Embed

If this link is followed, the node at the end of it is embedded into the display of the source document.

Acyclic.

7.1.6 Precedes

In an ordered structure defined by the author, A precedes B, B is followed by A.
Acyclic.
Any document may only have one link of this relationship, and/or one link of the reverse relationship.
Note: May be used to control navigational aids, generate printed material, etc. In conjunction with "subdocument", may be used to define a tree such as a printed book made of hypertext document. The document can only have one such tree.

7.1.7 Subdocument

B is a lower part in the author's hierarchy to A. Acyclic. See also Precedes.

7.1.8 Present

Whenever A is presented, B must also be presented. This implies that whenever A is retrieved, B must also be retrieved.

7.1.9 Search

When the link is followed, the node B should be searched rather than presented. That is, where the client software allows it, the user should immediately be presented with a search panel and prompted for text. The search is then performed without an intermediate retrieval or presentation of the node B

7.1.10 Supersedes

B is a previous version of A.
Acyclic.

7.1.11 History

B is a list of versions of A
A link reverse link must exist from B to A and to all other known versions of A.

7.2 Relationships about subjects of documents

These relationships convey semantics about objects described by documents, rather than the documents themselves.

7.2.1 Includes

A includes B, B is part of A. For example, a person described by document A is a part of the group described by document B.
Acyclic.

7.2.2 Made

Person (etc) described by node A is author of, or is responsible for B
This information can be used for protection, and informing authors of interest, for sending mail to authors, etc.

7.2.3 Interested

Person (etc) described by A is interested in node B
This information can be used for informing readers of changes.

8. Registration Authority

The HTTP Registration Authority is responsible for maintaining lists of:

- Relationship names for link and anchor elements

It is proposed that the Internet Assigned Numbers Authority or their successors take this role. Unregistered values may be used for experimental purposes if they are start with "X-".

9. References

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

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

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

URL Universal Resource Locators. RFCxxx. Currently available by anonymous FTP from info.cern.ch in /pub/ietf.

10. Author's addresses

This document was prepared with the help and advice of many people across the net. Dan Connolly prepared the DTD and the section on HTML and SGML whilst with Convex Computer Corporation of 3000 Waterview Parkway Richardson, TX 75083. He is now with Atrium Technology Inc., and is not a current editor of the document.


```
Tim Berners-Lee
Address CERN
1211 Geneva 23
Switzerland
Telephone:  +41(22)767 3755
Fax:        +41(22)767 7155
email:     timbl@info.cern.ch

Daniel Connolly
Address: Atrium Technologies, Inc.
5000 Plaza on the Lake, Suite 275
Austin, TX 78746
USA
email: connolly@atrium.com
```