



HTML 3.2 Reference Specification

W3C Recommendation 14-Jan-1997

Author: *Dave Raggett* <dsr@w3.org>

Status of this document

This document has been reviewed by W3C members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

A list of current W3C Recommendations and other technical documents can be found at <http://www.w3.org/pub/WWW/TR/>.

Abstract

The 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. This specification defines HTML version 3.2. HTML 3.2 aims to capture recommended practice as of early '96 and as such to be used as a replacement for HTML 2.0 ([RFC 1866](#)).

Contents

- [Introduction to HTML 3.2](#)
 - [HTML as an SGML application](#)
 - [The Structure of HTML documents](#)
 - [The HEAD element and its children](#)
 - [The BODY element and its children](#)
 - [Sample SGML Open Catalog for HTML 3.2](#)
 - [SGML Declaration for HTML 3.2](#)
 - [HTML 3.2 Document Type Definition](#)
 - [Character Entities for ISO Latin-1](#)
 - [Table of printable Latin-1 Character codes](#)
 - [Acknowledgements](#)
 - [Further Reading ...](#)
-

Introduction to HTML 3.2

HTML 3.2 is W3C's specification for HTML, developed in early '96 together with vendors including IBM, Microsoft, Netscape Communications Corporation, Novell, SoftQuad, Spyglass, and Sun Microsystems. HTML 3.2 adds widely deployed features such as tables, applets and text flow around images, while providing full backwards compatibility with the existing standard HTML 2.0.

W3C is continuing to work with vendors on extensions for accessibility features, multimedia objects, scripting, style sheets, layout, forms, math and internationalization. W3C plans on incorporating this work in further versions of HTML.

HTML as an SGML Application

HTML 3.2 is an SGML application conforming to International Standard ISO 8879 -- Standard Generalized Markup Language. As an SGML application, the syntax of conforming HTML 3.2 documents is defined by the combination of the [SGML declaration](#) and the [document type definition](#) (DTD). This specification defines the intended interpretation of HTML 3.2 elements, and places further constraints on the permitted syntax which are otherwise inexpressible in the DTD.

The SGML rules for record boundaries are tricky. In particular, a record end immediately following a start tag should be discarded. For example:

```
<P>  
Text
```

is equivalent to:

```
<P>Text
```

Similarly, a record end immediately preceding an end tag should be discarded. For example:

```
Text  
</P>
```

is equivalent to:

```
Text</P>
```

Except within literal text (e.g. the PRE element), HTML treats contiguous sequences of white space characters as being equivalent to a single space character (ASCII decimal 32). These rules allow authors considerable flexibility when editing the marked-up text directly. Note that future revisions to HTML may allow for the interpretation of the horizontal tab character (ASCII decimal 9) with respect to a tab rule defined by an associated style sheet.

SGML entities in PCDATA content or in CDATA attributes are expanded by the parser, e.g. é is expanded to the ISO Latin-1 character decimal 233 (a lower case letter e with an acute accent). This could also have been written as a named character entity, e.g. é. The & character can be included in its own right using the named character entity &.

HTML allows CDATA attributes to be unquoted provided the attribute value contains only letters (a to z and A to Z), digits (0 to 9), hyphens (ASCII decimal 45) or, periods (ASCII decimal 46). Attribute values can be quoted using double or single quote marks (ASCII decimal 34 and 39 respectively). Single quote marks can be included within the attribute value when the value is delimited by double quote marks, and vice versa.

Note that some user agents require attribute minimisation for the following attributes: COMPACT, ISMAP, CHECKED, NOWRAP, NOSHADE and NOHREF. These user agents don't accept syntax such as COMPACT=COMPACT or ISMAP=ISMAP although this is legitimate according to the HTML 3.2 DTD.

The SGML declaration and the DTD for use with HTML 3.2 are given in appendices. Further guidelines for parsing HTML are given in [WD-html-lex](#).

The Structure of HTML documents

HTML 3.2 Documents start with a `<!DOCTYPE>` declaration followed by an HTML element containing a [HEAD](#) and then a [BODY](#) element:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
<TITLE>A study of population dynamics</TITLE>
... other head elements
</HEAD>
<BODY>
... document body
</BODY>
</HTML>
```

In practice, the HTML, [HEAD](#) and [BODY](#) start and end tags can be omitted from the markup as these can be inferred in all cases by parsers conforming to the [HTML 3.2 DTD](#).

Every conforming HTML 3.2 document **must** start with the `<!DOCTYPE>` declaration that is needed to distinguish HTML 3.2 documents from other versions of HTML. The HTML specification is not concerned with storage entities. As a result, it is not required that the document type declaration reside in the same storage entity (i.e. file). A Web site may choose to dynamically prepend HTML files with the document type declaration if it is known that all such HTML files conform to the HTML 3.2 specification.

Every HTML 3.2 document must also include the descriptive title element. A minimal HTML 3.2 document thus looks like:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<TITLE>A study of population dynamics</TITLE>
```

Note: the word "Final" replaces "Draft" now that the HTML 3.2 specification has been ratified by the W3C member organizations.

The HEAD element

This contains the document head, but you can always omit both the start and end tags for HEAD. The contents of the document head is an unordered collection of the following elements:

- [The TITLE element](#)
- [The STYLE element](#)
- [The SCRIPT element](#)
- [The ISINDEX element](#)
- [The BASE element](#)
- [The META element](#)
- [The LINK element](#)

```
<!ENTITY % head.content "TITLE & ISINDEX? & BASE?">
<!ENTITY % head.misc "SCRIPT|STYLE|META|LINK">

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

rel

The forward relationship also known as the "link type". It can be used to determine to how to deal with the linked resource when printing out a collection of linked resources.

rev

This defines a reverse relationship. A link from document A to document B with REV=relation expresses the same relationship as a link from B to A with REL=relation. REV=made is sometimes used to identify the document author, either the author's email address with a mailto URL, or a link to the author's home page.

title

An advisory title for the linked resource.

IMG - inline images

```
<!ENTITY % IAlign "(top|middle|bottom|left|right)">
```

```
<!ELEMENT IMG      - O EMPTY --  Embedded image -->
```

```
<!ATTLIST IMG
```

```
    src      %URL      #REQUIRED  -- URL of image to embed --
    alt      CDATA     #IMPLIED   -- for display in place of image --
    align    %IAlign   #IMPLIED   -- vertical or horizontal alignment --
    height   %Pixels   #IMPLIED   -- suggested height in pixels --
    width    %Pixels   #IMPLIED   -- suggested width in pixels --
    border   %Pixels   #IMPLIED   -- suggested link border width --
    hspace   %Pixels   #IMPLIED   -- suggested horizontal gutter --
    vspace   %Pixels   #IMPLIED   -- suggested vertical gutter --
    usemap   %URL      #IMPLIED   -- use client-side image map --
    ismap    (ismap)   #IMPLIED   -- use server image map --
  >
```

Used to insert images. IMG is an empty element and so the end tag is forbidden. Images can be positioned vertically relative to the current textline or floated to the left or right. See BR with the CLEAR attribute for control over textflow.

e.g. `<IMG SRC="canyon.gif" ALT="Grand Canyon">`

IMG elements support the following attributes:

src

This attribute is required for every IMG element. It specifies a URL for the image resource, for instance a GIF, JPEG or PNG image file.

alt

This is used to provide a text description of the image and is vital for interoperability with speech-based and text only user agents.

align

This specifies how the image is positioned relative to the current textline in which it occurs:

`align=top`

positions the top of the image with the top of the current text line. User agents vary in how they interpret this. Some only take into account what has occurred on the text line prior to the IMG element and ignore what happens after it.

`align=middle`

aligns the middle of the image with the baseline for the current textline.

`align=bottom`

is the default and aligns the bottom of the image with the baseline.

`align=left`

floats the image to the current left margin, temporarily changing this margin, so that subsequent text is flowed along the image's righthand side. The rendering depends on whether there is any left aligned text or images that appear earlier than the current image in the markup. Such text (but not

DataDome Ex. 1018, p. 4

DataDome v. Arkose Labs

IPR2025-00694

images) generally forces left aligned images to wrap to a new line, with the subsequent text continuing on the former line.

align=right

floats the image to the current right margin, temporarily changing this margin, so that subsequent text is flowed along the image's lefthand side. The rendering depends on whether there is any right aligned text or images that appear earlier than the current image in the markup. Such text (but not images) generally forces right aligned images to wrap to a new line, with the subsequent text continuing on the former line.

Note that some browsers introduce spurious spacing with multiple left or right aligned images. As a result authors can't depend on this being the same for browsers from different vendors. See [BR](#) for ways to control text flow.

width

Specifies the intended width of the image in pixels. When given together with the height, this allows user agents to reserve screen space for the image before the image data has arrived over the network.

height

Specifies the intended height of the image in pixels. When given together with the width, this allows user agents to reserve screen space for the image before the image data has arrived over the network.

border

When the IMG element appears as part of a hypertext link, the user agent will generally indicate this by drawing a colored border (typically blue) around the image. This attribute can be used to set the width of this border in pixels. Use `border=0` to suppress the border altogether. User agents are recommended to provide additional cues that the image is clickable, e.g. by changing the mouse pointer.

hspace

This can be used to provide white space to the immediate left and right of the image. The HSPACE attribute sets the width of this white space in pixels. By default HSPACE is a small non-zero number.

vspace

This can be used to provide white space above and below the image. The VSPACE attribute sets the height of this white space in pixels. By default VSPACE is a small non-zero number.

usemap

This can be used to give a URL fragment identifier for a client-side image map defined with the [MAP](#) element.

ismap

When the IMG element is part of a hypertext link, and the user clicks on the image, the ISMAP attribute causes the location to be passed to the server. This mechanism causes problems for text-only and speech-based user agents. Whenever its possible to do so use the MAP element instead.

Here is an example of how you use ISMAP:

```
<a href="/cgibin/navbar.map"><img src=navbar.gif ismap border=0></a>
```

The location clicked is passed to the server as follows. The user agent derives a new URL from the URL specified by the HREF attribute by appending '?' the x coordinate ',' and the y coordinate of the location in pixels. The link is then followed using the new URL. For instance, if the user clicked at the location x=10, y=27 then the derived URL will be: `"/cgibin/navbar.map?10,27"`. It is generally a good idea to suppress the border and use graphical idioms to indicate that the image is clickable.

Note that pixel values refer to screen pixels, and should be multiplied by an appropriate factor when rendering to very high resolution devices such as laser printers. For instance if a user agent has a display with 75 pixels per inch and is rendering to a laser printer with 600 dots per inch, then the pixel values given in HTML attributes should be multiplied by a factor of 8.

APPLET (*Java Applets*)

```

<!-- CENTER is a shorthand for DIV with ALIGN=CENTER -->
<!ELEMENT center - - %body.content>

<!--===== The Anchor Element =====>

<!ELEMENT A - - (%text)* -(A)>
<!ATTLIST A
    name      CDATA      #IMPLIED      -- named link end --
    href      %URL       #IMPLIED      -- URL for linked resource --
    rel       CDATA      #IMPLIED      -- forward link types --
    rev       CDATA      #IMPLIED      -- reverse link types --
    title     CDATA      #IMPLIED      -- advisory title string --
>

<!--===== Client-side image maps =====>

<!-- These can be placed in the same document or grouped in a
    separate document although this isn't yet widely supported -->

<ENTITY % SHAPE "(rect|circle|poly)">
<ENTITY % COORDS "CDATA" -- comma separated list of numbers -->

<!ELEMENT MAP - - (AREA)*>
<!ATTLIST MAP
    name      CDATA      #IMPLIED
>

<!ELEMENT AREA - 0 EMPTY>
<!ATTLIST AREA
    shape     %SHAPE     rect
    coords    %COORDS    #IMPLIED      -- defines coordinates for shape --
    href      %URL       #IMPLIED      -- this region acts as hypertext link --
    nohref    (nohref)   #IMPLIED      -- this region has no action --
    alt       CDATA      #REQUIRED     -- needed for non-graphical user agents --
>

<!--===== The LINK Element =====>

<ENTITY % Types "CDATA"
    -- See Internet Draft: draft-ietf-html-relrev-00.txt
    LINK has been part of HTML since the early days
    although few browsers as yet take advantage of it.

    Relationship values can be used in principle:

        a) for document specific toolbars/menus when used
           with the LINK element in the document head:
        b) to link to a separate style sheet
        c) to make a link to a script
        d) by stylesheets to control how collections of
           html nodes are rendered into printed documents
        e) to make a link to a printable version of this document
           e.g. a postscript or pdf version
-->

<!ELEMENT LINK - 0 EMPTY>
<!ATTLIST LINK
    href      %URL       #IMPLIED      -- URL for linked resource --
    rel       %Types     #IMPLIED      -- forward link types --
    rev       %Types     #IMPLIED      -- reverse link types --
    title     CDATA      #IMPLIED      -- advisory title string --
>

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

```

```
<!ENTITY % Length "CDATA" -- nn for pixels or nn% for percentage length -->
<!ENTITY % Pixels "NUMBER" -- integer representing length in pixels -->
```

```
<!-- Suggested widths are used for negotiating image size
      with the module responsible for painting the image.
      align=left or right cause image to float to margin
      and for subsequent text to wrap around image -->
```

```
<!ENTITY % IAlign "(top|middle|bottom|left|right)">
```

```
<!ELEMENT IMG - O EMPTY -- Embedded image -->
```

```
<!ATTLIST IMG
```

```
  src      %URL      #REQUIRED -- URL of image to embed --
  alt      CDATA      #IMPLIED  -- for display in place of image --
  align    %IAlign    #IMPLIED  -- vertical or horizontal alignment --
  height   %Pixels    #IMPLIED  -- suggested height in pixels --
  width    %Pixels    #IMPLIED  -- suggested width in pixels --
  border   %Pixels    #IMPLIED  -- suggested link border width --
  hspace   %Pixels    #IMPLIED  -- suggested horizontal gutter --
  vspace   %Pixels    #IMPLIED  -- suggested vertical gutter --
  usemap   %URL       #IMPLIED  -- use client-side image map --
  ismap    (ismap)    #IMPLIED  -- use server image map --
>
```

```
<!-- USEMAP points to a MAP element which may be in this document
      or an external document, although the latter is not widely supported -->
```

```
<!--===== Java APPLET tag =====>
```

```
<!--
```

```
  This tag is supported by all Java enabled browsers. Applet resources
  (including their classes) are normally loaded relative to the document
  URL (or <BASE> element if it is defined). The CODEBASE attribute is used
  to change this default behavior. If the CODEBASE attribute is defined then
  it specifies a different location to find applet resources. The value
  can be an absolute URL or a relative URL. The absolute URL is used as is
  without modification and is not effected by the documents <BASE> element.
  When the codebase attribute is relative, then it is relative to the
  document URL (or <BASE> tag if defined).
```

```
-->
```

```
<!ELEMENT APPLET - - (PARAM | %text)*>
```

```
<!ATTLIST APPLET
```

```
  codebase %URL      #IMPLIED  -- code base --
  code     CDATA      #REQUIRED -- class file --
  alt      CDATA      #IMPLIED  -- for display in place of applet --
  name     CDATA      #IMPLIED  -- applet name --
  width    %Pixels    #REQUIRED -- suggested width in pixels --
  height   %Pixels    #REQUIRED -- suggested height in pixels --
  align    %IAlign    #IMPLIED  -- vertical or horizontal alignment --
  hspace   %Pixels    #IMPLIED  -- suggested horizontal gutter --
  vspace   %Pixels    #IMPLIED  -- suggested vertical gutter --
>
```

```
<!ELEMENT PARAM - O EMPTY>
```

```
<!ATTLIST PARAM
```

```
  name     NMTOKEN    #REQUIRED -- The name of the parameter --
  value    CDATA      #IMPLIED  -- The value of the parameter --
>
```

```
<!--
```

```
Here is an example:
```

```
<applet codebase="applets/NervousText"
  code=NervousText.class
  width=300
  height=50>
```