Abstract

This specification defines the HyperText Markup Language (HTML), version 4.0, the publishing language of the World Wide Web. In addition to the text, multimedia, and hyperlink features of the previous versions of HTML, HTML 4.0 supports more multimedia options, scripting languages, style sheets, better printing facilities, and documents that are more accessible to users with disabilities. HTML 4.0 also takes great strides towards the internationalization of documents, with the goal of making the Web truly World Wide.

HTML 4.0 is an SGML application conforming to International Standard ISO 8879 -- Standard Generalized Markup Language [ISO8879][p.327].

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.

W3C recommends that user agents and authors (and in particular, authoring tools) produce HTML 4.0 documents rather than HTML 3.2 documents (see [HTML32][p.329]). For reasons of backwards compatibility, W3C also recommends that tools interpreting HTML 4.0 continue to support HTML 3.2 and HTML 2.0 as well.
A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR.

Public discussion on HTML features takes place on www-html@w3.org.

This document is a revised version of the document first released on 18 December 1997. Changes from the original version are only editorial in nature.

Available formats

The HTML 4.0 W3C Recommendation is also available in the following formats:

A plain text file:
  http://www.w3.org/TR/1998/REC-html40-19980424/html40.txt (735Kb),
A gzip’ed tar file containing HTML documents:
  http://www.w3.org/TR/1998/REC-html40-19980424/html40.tgz (357Kb),
A zip file containing HTML documents (this is a ’.zip’ file not an ’.exe’):
  http://www.w3.org/TR/1998/REC-html40-19980424/html40.zip (389Kb),
A gzip’ed Postscript file:
  http://www.w3.org/TR/1998/REC-html40-19980424/html40.ps.gz (600Kb, 367 pages),
A PDF file:

In case of a discrepancy between electronic and printed forms of the specification, the electronic version is the definitive version.

Available languages

The English version of this specification is the only normative version. However, for translations of this document, see http://www.w3.org/MarkUp/html40-updates/translations.html.

Errata

The list of known errors in this specification is available at:
  http://www.w3.org/MarkUp/html40-updates/REC-html40-19980424-errata.html

Please report errors in this document to www-html-editor@w3.org.
### Table of Contents

1. **About the HTML 4.0 Specification**
   1. How the specification is organized ........................................... 13
   2. Document conventions
      1. Elements and attributes ............................................. 14
      2. Notes and examples ............................................. 15
   3. Acknowledgments ....................................................... 15
   4. Copyright Notice .................................................... 16

2. **Introduction to HTML 4.0**
   1. What is the World Wide Web? ........................................ 17
      1. Introduction to URIs ............................................. 17
      2. Fragment identifiers ........................................... 18
      3. Relative URIs .................................................... 18
   2. What is HTML?
      1. A brief history of HTML ........................................... 19
   3. HTML 4.0
      1. Internationalization ............................................. 20
      2. Accessibility ..................................................... 20
      3. Tables ............................................................ 21
      4. Compound documents ............................................. 21
      5. Style sheets ...................................................... 21
      6. Scripting ......................................................... 22
      7. Printing .......................................................... 22
   4. Authoring documents with HTML 4.0
      1. Separate structure and presentation ................................ 22
      2. Consider universal accessibility to the Web ....................... 22
      3. Help user agents with incremental rendering ..................... 22

3. **On SGML and HTML**
   1. Introduction to SGML ................................................ 23
   2. SGML constructs used in HTML
      1. Elements ............................................................ 24
      2. Attributes ........................................................ 24
      3. Character references ............................................. 25
      4. Comments .......................................................... 25
   3. How to read the HTML DTD
      1. DTD Comments .................................................... 26
      2. Parameter entity definitions .................................... 26
      3. Element declarations
         • Content model definitions ..................................... 27
      4. Attribute declarations
         • DTD entities in attribute definitions ......................... 28
         • Boolean attributes ............................................. 29
4. Conformance: requirements and recommendations
   1. Definitions
   2. SGML
   3. The text/html content type

5. HTML Document Representation
   1. The Document Character Set
   2. Character encodings
      1. Choosing an encoding
      2. Specifying the character encoding
   3. Character references
      1. Numeric character references
      2. Character entity references
   4. Undisplayable characters

6. Basic HTML data types
   1. Case information
   2. SGML basic types
   3. Text strings
   4. URIs
   5. Colors
      1. Notes on using colors
   6. Lengths
   7. Content types (MIME types)
   8. Language codes
   9. Character encodings
   10. Single characters
   11. Dates and times
   12. Link types
   13. Media descriptors
   14. Script data
   15. Style sheet data
   16. Frame target names

7. The global structure of an HTML document
   1. Introduction to the structure of an HTML document
   2. HTML version information
   3. The HTML element
   4. The document head
      1. The HEAD element
      2. The TITLE element
      3. The title attribute
      4. Meta data
         • Specifying meta data
         • The META element
Table of Contents

5. The document body
   1. The BODY element
   2. Element identifiers: the id and class attributes
   3. Block-level and inline elements
   4. Grouping elements: the DIV and SPAN elements
   5. Headings: The H1, H2, H3, H4, H5, H6 elements
   6. The ADDRESS element

8. Language information and text direction
   1. Specifying the language of content: the lang attribute
   2. Specifying the direction of text and tables: the dir attribute

9. Text
   1. White space
   2. Structured text
      1. Phrase elements: EM, STRONG, DFN, CODE, SAMPL, KBD, VAR, CITE, ABBR, and ACRONYM
      2. Quotations: The BLOCKQUOTE and Q elements
         • Rendering quotations
      3. Subscripts and superscripts: the SUB and SUP elements
   3. Lines and Paragraphs
      1. Paragraphs: the P element
      2. Controlling line breaks
         • Forcing a line break: the BR element
         • Prohibiting a line break
      3. Hyphenation
   4. Preformatted text: The PRE element
   5. Visual rendering of paragraphs

10. Lists
    1. Introduction to lists
    2. Unordered lists (UL), ordered lists (OL), and list items (LI)
    3. Definition lists: the DL, DT, and DD elements
       1. Visual rendering of lists
    4. The DIR and MENU elements
11. Tables

1. Introduction to tables

2. Elements for constructing tables
   1. The TABLE element
   2. Table directionality
   3. Table Captions: The CAPTION element
   4. Row groups: the THEAD, TFOOT, and TBODY elements
   5. Column groups: the COLGROUP and COL elements
      1. The COLGROUP element
      2. The COL element
      3. Calculating the number of columns in a table
      4. Calculating the width of columns
   6. Table rows: The TR element
   7. Table cells: The TH and TD elements
      1. Cells that span several rows or columns

3. Table formatting by visual user agents
   1. Borders and rules
   2. Horizontal and vertical alignment
      1. Inheritance of alignment specifications
   3. Cell margins

4. Table rendering by non-visual user agents
   1. Associating header information with data cells
   2. Categorizing cells
   3. Algorithm to find heading information

5. Sample table

12. Links - Hypertext and Media-Independent Links

1. Introduction to links and anchors
   1. Visiting a linked resource
   2. Other link relationships
   3. Specifying anchors and links
   4. Link titles
   5. Internationalization and links

2. The A element
   1. Syntax of anchor names
   2. Nested links are illegal
   3. Anchors with the id attribute
   4. Unavailable and unidentifiable resources

3. Document relationships: the LINK element
   1. Forward and reverse links
   2. Links and external style sheets
   3. Links and search engines

4. Path information: the BASE element
   1. Resolving relative URIs
# Table of Contents

1. Font style elements: the TT, I, B, BIG, SMALL, STRIKE, S, and U elements ........................................... 187
2. Font modifier elements: FONT and BASEFONT ................................................................................. 188
3. Rules: the HR element ......................................................................................................................... 190

16. Frames - Multi-view presentation of documents ............................................................................... 193
   1. Introduction to frames ...................................................................................................................... 193
   2. Layout of frames ............................................................................................................................... 194
      1. The FRAMESET element ............................................................................................................ 194
         - Rows and columns .................................................................................................................... 195
         - Nested frame sets .................................................................................................................... 196
         - Sharing data among frames .................................................................................................... 196
      2. The FRAME element .................................................................................................................... 197
         - Setting the initial contents of a frame ...................................................................................... 198
         - Visual rendering of a frame ....................................................................................................... 199

3. Specifying target frame information ................................................................................................. 200
   1. Setting the default target for links ............................................................................................... 201
   2. Target semantics ......................................................................................................................... 202

4. Alternate content ............................................................................................................................... 202
   1. The NOFRAMES element ............................................................................................................ 202
   2. Long descriptions of frames ......................................................................................................... 203

5. Inline frames: the IFRAME element ................................................................................................. 204

17. Forms - User-input Forms: Text Fields, Buttons, Menus, and more .............................................. 207
   1. Introduction to forms ...................................................................................................................... 207
   2. Controls ....................................................................................................................................... 208
      1. Control types ............................................................................................................................... 208
   3. The FORM element ....................................................................................................................... 210
   4. The INPUT element ....................................................................................................................... 211
      1. Control types created with INPUT ............................................................................................ 213
      2. Examples of forms containing INPUT controls ......................................................................... 214
   5. The BUTTON element ................................................................................................................... 215
   6. The SELECT, OPTGROUP, and OPTION elements .......................................................................... 217
      1. Preselected options ...................................................................................................................... 218
   7. The TEXTAREA element ............................................................................................................... 221
   8. The ISINDEX element ................................................................................................................... 223
   9. Labels .......................................................................................................................................... 223
      1. The LABEL element ................................................................................................................... 224
   10. Adding structure to forms: the FIELDSET and LEGEND elements ........................................... 225
   11. Giving focus to an element ............................................................................................................ 227
      1. Tabbing navigation ......................................................................................................................... 228
      2. Access keys .................................................................................................................................. 229
   12. Disabled and read-only controls .................................................................................................... 230
      1. Disabled controls .......................................................................................................................... 230
      2. Read-only controls ...................................................................................................................... 231
   13. Form submission ............................................................................................................................ 231
## Table of Contents

- **Deprecated elements** ........................................... 301
- **Obsolete elements** ........................................... 302
- **Changes to attributes** ........................................ 302
- **Changes for accessibility** .................................... 302
- **Changes for meta data** ....................................... 302
- **Changes for text** .............................................. 302
- **Changes for links** ............................................. 302
- **Changes for tables** ........................................... 302
- **Changes for images, objects, and image maps** ............ 303
- **Changes for forms** ............................................ 303
- **Changes for style sheets** ..................................... 304
- **Changes for frames** ........................................... 304
- **Changes for scripting** ......................................... 304
- **Changes for internationalization** ........................... 304
- **Changes from the 18 December 1997 specification** ....... 304
  - **Errors that were corrected** ................................. 305
  - **Minor typographical errors that were corrected** ....... 307
- **B. Performance, Implementation, and Design Notes** .... 309
  - **Notes on invalid documents** ................................ 310
  - **Special characters in URI attribute values** ............... 310
    1. **Non-ASCII characters in URI attribute values** ....... 310
    2. **Ampersands in URI attribute values** ..................... 311
  - **SGML implementation notes** ................................ 311
    1. **Line breaks** .............................................. 311
    2. **Specifying non-HTML data** ............................... 312
      - **Element content** ........................................ 312
      - **Attribute values** ....................................... 313
    3. **SGML features with limited support** ..................... 313
  - **Boolean attributes** ......................................... 313
  - **Marked Sections** ........................................... 313
  - **Processing Instructions** ................................... 314
  - **Shorthand markup** .......................................... 314
  - **Notes on helping search engines index your Web site** .. 315
    1. **Search robots** .......................................... 316
      - **The robots.txt file** .................................... 316
      - **Robots and the META element** ......................... 317
  - **Notes on tables** ........................................... 317
    1. **Design rationale** ........................................ 317
      - **Dynamic reformatting** .................................. 318
      - **Incremental display** .................................... 318
      - **Structure and presentation** ............................. 318
      - **Row and column groups** ................................ 319
      - **Accessibility** ........................................... 319

10

Page 10 of 30
Table of Contents

2. **Recommended Layout Algorithms** ........................................... 319
   - **Fixed Layout Algorithm** .................................................. 320
   - **Autolayout Algorithm** ................................................... 320

6. **Notes on forms** ................................................................. 322
   1. **Incremental display** ....................................................... 322
   2. **Future projects** ............................................................. 322

7. **Notes on scripting** ............................................................ 323
   1. **Reserved syntax for future script macros** ............................. 323
   - **Current Practice for Script Macros** .................................... 323

8. **Notes on frames** ............................................................... 324

9. **Notes on accessibility** ....................................................... 325

10. **Notes on security** ............................................................. 325
    1. **Security issues for forms** .............................................. 325

- **References** ................................................................. 327
  1. **Normative references** .................................................... 327
  2. **Informative references** ................................................... 329
- **Index of Elements** ............................................................ 333
- **Index of Attributes** ........................................................... 337
- **Index** ............................................................................. 353

Copyright[p.16] © 1997 W3C (MIT INRIA Keio), All Rights Reserved.
1 About the HTML 4.0 Specification

Contents

1. How the specification is organized ........................................... 13
2. Document conventions ......................................................... 14
   1. Elements and attributes .................................................. 14
   2. Notes and examples ...................................................... 15
3. Acknowledgments ................................................................... 15
4. Copyright Notice .................................................................... 16

1.1 How the specification is organized

This specification is divided into the following sections:

Sections 2 and 3: Introduction to HTML 4.0

The introduction describes HTML’s place in the scheme of the World Wide Web, provides a brief history of the development of HTML, highlights what can be done with HTML 4.0, and provides some HTML authoring tips.

The brief SGML tutorial gives readers some understanding of HTML’s relationship to SGML and gives summary information on how to read the HTML Document Type Definition (DTD).

Sections 4 - 24: HTML 4.0 reference manual

The bulk of the reference manual consists of the HTML language reference, which defines all elements and attributes of the language.

This document has been organized by topic rather than by the grammar of HTML. Topics are grouped into three categories: structure, presentation, and interactivity. Although it is not easy to divide HTML constructs perfectly into these three categories, the model reflects the HTML Working Group’s experience that separating a document’s structure from its presentation produces more effective and maintainable documents.

The language reference consists of the following information:

- Basic data types [p.43] of an HTML document.
- Elements that govern the structure of an HTML document, including text [p.81], lists [p.93], tables [p.101], links [p.135], and included objects, images, and applets [p.149].
- Elements that govern the presentation of an HTML document, including style sheets [p.171], fonts, colors, rules, and other visual presentation [p.183], and frames for multi-windowed presentations [p.193].
1.2 Document conventions

This document has been written with two types of readers in mind: authors and implementors. We hope the specification will provide authors with the tools they need to write efficient, attractive, and accessible documents, without over-exposing them to HTML’s implementation details. Implementors, however, should find all they need to build conforming user agents.

The specification may be approached in several ways:

- **Read from beginning to end.** The specification begins with a general presentation of HTML and becomes more and more technical and specific towards the end.
- **Quick access to information.** In order to get information about syntax and semantics as quickly as possible, the online version of the specification includes the following features:
  1. Every reference to an element or attribute is linked to its definition in the specification. Each element or attribute is defined in only one location.
  2. Every page includes links to the indexes, so you never are more than two links away from finding the definition of an element or attribute.
  3. The front pages of the three sections of the language reference manual extend the initial table of contents with more detail about each section.

1.2.1 Elements and attributes

Element names are written in uppercase letters (e.g., BODY). Attribute names are written in lowercase letters (e.g., lang, onsubmit). Recall that in HTML, element and attribute names are case-insensitive; the convention is meant to encourage readability.
Element and attribute names in this document have been marked up and may be rendered specially by some user agents.

Each attribute definition specifies the type of its value. If the type allows a small set of possible values, the definition lists the set of values, separated by a bar (|).

After the type information, each attribute definition indicates the case-sensitivity of its values, between square brackets ("[]"). See the section on case information [p.43] for details.

1.2.2 Notes and examples

Informative notes are emphasized to stand out from surrounding text and may be rendered specially by some user agents.

All examples illustrating [deprecated] [p.34] usage are marked as "DEPRECATED EXAMPLE". Deprecated examples also include recommended alternate solutions. All examples that illustrate illegal usage are clearly marked "ILLEGAL EXAMPLE".

Examples and notes have been marked up and may be rendered specially by some user agents.

1.3 Acknowledgments

Thanks to everyone who has helped to author the working drafts that went into the HTML 4.0 specification, and to all those who have sent suggestions and corrections.

Many thanks to the Web Accessibility Initiative task force (WAI HC group) for their work on improving the accessibility of HTML and to T.V. Raman (Adobe) for his early work on developing accessible forms.

The authors of this specification, the members of the W3C HTML Working Group, deserve much applause for their diligent review of this document, their constructive comments, and their hard work: John D. Burger (MITRE), Steve Byrne (JavaSoft), Martin J. Dürst (University of Zurich), Daniel Glazman (Electricité de France), Scott Isaacs (Microsoft), Murray Maloney (GRIF), Steven Pemberton (CWI), Robert Pernett (Lotus), Jared Sorensen (Novell), Powell Smith (IBM), Robert Stevahn (HP), Ed Tecot (Microsoft), Jeffrey Veen (HotWired), Mike Wexler (Adobe), Misha Wolf (Reuters), and Lauren Wood (SoftQuad).

Thank you Dan Connolly (W3C) for rigorous and bountiful input as part-time editor and thoughtful guidance as chairman of the HTML Working Group. Thank you Sally Khudairi (W3C) for your indispensable work on press releases.

Thanks to David M. Abrahamson and Roger Price for their careful reading of the specification and constructive comments.

Thanks to Jan Kärrman, author of html2ps for helping so much in creating the Postscript version of the specification.
Of particular help from the W3C at Sophia-Antipolis were Janet Bertot, Bert Bos, Stephane Boyera, Daniel Dardailler, Yves Lafon, Håkon Lie, Chris Lilley, and Colas Nahaboo (Bull).

Lastly, thanks to Tim Berners-Lee without whom none of this would have been possible.

1.4 Copyright Notice


Documents on the W3C site are provided by the copyright holders under the following license. By obtaining, using and/or copying this document, or the W3C document from which this statement is linked, you agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URI to the original W3C document.
2. The pre-existing copyright notice of the original author, if it doesn’t exist, a notice of the form: "Copyright © World Wide Web Consortium (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved."
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. In addition, credit shall be attributed to the copyright holders for any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives is granted pursuant to this license.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.
16 Frames

16.1 Introduction to frames

HTML frames allow authors to present documents in multiple views, which may be independent windows or subwindows. Multiple views offer designers a way to keep certain information visible, while other views are scrolled or replaced. For example, within the same window, one frame might display a static banner, a second a navigation menu, and a third the main document that can be scrolled though or replaced by navigating in the second frame.

Here is a simple frame document:

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
"http://www.w3.org/TR/REC-html40/frameset.dtd">
<HTML>
<HEAD>
<TITLE>A simple frameset document</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
  <FRAMESET rows="100, 200">
    <FRAME src="contents_of_frame1.html">
    <FRAME src="contents_of_frame2.gif">
  </FRAMESET>
  <FRAME src="contents_of_frame3.html">
  <NOFRAMES>
    <P>This frameset document contains:
    <UL>
      <LI><A href="contents_of_frame1.html">Some neat contents</A>
      <LI><IMG src="contents_of_frame2.gif" alt="A neat image">
  </NOFRAMES>
</FRAMESET>
</HTML>
```
that might create a frame layout something like this:

```
|         |                             |
|         |                             |
| Frame 1 |                             |
|         |          Frame 3            |
|         |                             |
| Frame 2 |                             |
```

If the user agent can’t display frames or is configured not to, it will render the contents of the \texttt{NOFRAMES} element.

### 16.2 Layout of frames

An HTML document that describes frame layout (called a \textit{frameset document}) has a different makeup than an HTML document without frames. A standard document has one \texttt{HEAD} section and one \texttt{BODY}. A frameset document has a \texttt{HEAD} and a \texttt{FRAMESET} in place of the \texttt{BODY}.

The \texttt{FRAMESET} section of a document specifies the layout of views in the main user agent window. In addition, the \texttt{FRAMESET} section can contain a \texttt{NOFRAMES} element to provide alternate content \cite{202} for user agents that do not support frames or are configured not to display frames.

Elements that might normally be placed in the \texttt{BODY} element must not appear before the first \texttt{FRAMESET} element or the \texttt{FRAMESET} will be ignored.

#### 16.2.1 The \texttt{FRAMESET} element

```xml
<! [ %HTML.Frameset; ]>
<!ELEMENT FRAMESET ( (FRAMESET | FRAME)* & NOFRAMES?) -- window subdivision-->
<!ATTLIST FRAMESET %coreattrs;                          -- id, class, style, title --
rows %MultiLengths; #IMPLIED -- list of lengths,
default: 100% (1 row) --
cols %MultiLengths; #IMPLIED -- list of lengths,
default: 100% (1 col) --
```
Attribute definitions

**rows** = multi-length-list

This attribute specifies the layout of horizontal frames. It is a comma-separated list of pixels, percentages, and relative lengths. The default value is 100%, meaning one row.

**cols** = multi-length-list

This attribute specifies the layout of vertical frames. It is a comma-separated list of pixels, percentages, and relative lengths. The default value is 100%, meaning one column.

Attributes defined elsewhere

- **id**, **class** (document-wide identifiers)
- **title** (element title)
- **style** (inline style information)
- **onload**, **onunload** (intrinsic events)

The **FRAMESET** element specifies the layout of the main user window in terms of rectangular subspaces.

**Rows and columns**

Setting the **rows** attribute defines the number of horizontal subspaces in a frameset. Setting the **cols** attribute defines the number of vertical subspaces. Both attributes may be set simultaneously to create a grid.

If the **rows** attribute is not set, each column extends the entire length of the page. If the **cols** attribute is not set, each row extends the entire width of the page. If neither attribute is set, the frame takes up exactly the size of the page.

Frames are created left-to-right for columns and top-to-bottom for rows. When both attributes are specified, views are created left-to-right in the top row, left-to-right in the second row, etc.

The first example divides the screen vertically in two (i.e., creates a top half and a bottom half).

```xml
<FRAMESET rows="50%, 50%">
  ...the rest of the definition...
</FRAMESET>
```

The next example creates three columns: the second has a fixed width of 250 pixels (useful, for example, to hold an image with a known size). The first receives 25% of the remaining space and the third 75% of the remaining space.

```xml
<FRAMESET cols="1*,250,3*">
  ...the rest of the definition...
</FRAMESET>
```
The next example creates a 2x3 grid of subspaces.

<FRAMESET rows="30%,70%" cols="33%,34%,33%">
  ...the rest of the definition...
</FRAMESET>

For the next example, suppose the browser window is currently 1000 pixels high. The first view is allotted 30% of the total height (300 pixels). The second view is specified to be exactly 400 pixels high. This leaves 300 pixels to be divided between the other two frames. The fourth frame’s height is specified as "2*", so it is twice as high as the third frame, whose height is only ":" (equivalent to 1*). Therefore the third frame will be 100 pixels high and the fourth will be 200 pixels high.

<FRAMESET rows="30%,400,*,2*">
  ...the rest of the definition...
</FRAMESET>

Absolute lengths that do not sum to 100% of the real available space should be adjusted by the user agent. When underspecified, remaining space should be allotted proportionally to each view. When overspecified, each view should be reduced according to its specified proportion of the total space.

**Nested frame sets**

Framesets may be nested to any level.

In the following example, the outer FRAMESET divides the available space into three equal columns. The inner FRAMESET then divides the second area into two rows of unequal height.

<FRAMESET cols="33%, 33%, 34%">
  ...contents of first frame...
  <FRAMESET rows="40%, 50%">
    ...contents of second frame, first row...
    ...contents of second frame, second row...
  </FRAMESET>
  ...contents of third frame...
</FRAMESET>

**Sharing data among frames**

Authors may share data among several frames by including this data via an OBJECT element. Authors should include the OBJECT element in the HEAD element of a frameset document and name it with the id attribute. Any document that is the contents of a frame in the frameset may refer to this identifier.

The following example illustrates how a script might refer to an OBJECT element defined for an entire frameset:

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN" "http://www.w3.org/TR/REC-html40/frameset.dtd">
<html>
<head>
<title>This is a frameset with OBJECT in the HEAD</title>
<!-- This OBJECT is not rendered! -->
</head>
<body>
<object id="myobject" data="data.bar"></object>
</body>
</html>
```
16.2.2 The FRAME element

Attribute definitions

name = [CDATA] [p.44] [CI] [p.43]

This attribute assigns a name to the current frame. This name may be used as the target of subsequent links.

longdesc = [UR] [p.44] [CT] [p.43]

This attribute specifies a link to a long description of the frame. This description should supplement the short description provided using the title attribute, and may be particularly useful for non-visual user agents.

src = [UR] [p.44] [CT] [p.43]

This attribute specifies the location of the initial contents to be contained in the frame.
noresize

When present, this boolean attribute tells the user agent that the frame window must not be resizeable.

scrolling = auto | yes | no

This attribute specifies scroll information for the frame window. Possible values:
- auto: This value tells the user agent to provide scrolling devices for the frame window when necessary. This is the default value.
- yes: This value tells the user agent to always provide scrolling devices for the frame window.
- no: This value tells the user agent not to provide scrolling devices for the frame window.

frameborder = 1 | 0

This attribute provides the user agent with information about the frame border. Possible values:
- 1: This value tells the user agent to draw a separator between this frame and every adjoining frame. This is the default value.
- 0: This value tells the user agent not to draw a separator between this frame and every adjoining frame. Note that separators may be drawn next to this frame nonetheless if specified by other frames.

marginwidth = pixels

This attribute specifies the amount of space to be left between the frame’s contents in its left and right margins. The value must be greater than one pixel. The default value depends on the user agent.

marginheight = pixels

This attribute specifies the amount of space to be left between the frame’s contents in its top and bottom margins. The value must be greater than one pixel. The default value depends on the user agent.

Attributes defined elsewhere

- id, class (document-wide identifiers)
- title (element title)
- style (inline style information)
- target (target frame information)

The FRAME element defines the contents and appearance of a single frame.

Setting the initial contents of a frame

The src attribute specifies the initial document the frame will contain.

The following example HTML document:

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
   "http://www.w3.org/TR/REC-html40/frameset.dtd">
<html>
<head>
<title>A frameset document</title>
</head>
<frameset cols="33%,33%,33%">
  <frameset rows="*,200">
    <frame src="contents_of_frame1.html">
    <frame src="contents_of_frame2.gif">
  </frameset>
</frameset>
</html>
```
should create a frame layout something like this:

```
<table>
<thead>
<tr>
<th>Frame 1</th>
<th>Frame 3</th>
<th>Frame 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

and cause the user agent to load each file into a separate view.

The contents of a frame must not be in the same document as the frame’s definition.

**ILLEGAL EXAMPLE:**
The following frameset definition is not legal HTML since the contents of the second frame are in the same document as the frameset.

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
 "http://www.w3.org/TR/REC-html40/frameset.dtd">
<html>
<head>
<title>A frameset document</title>
</head>
<frameset cols="50%,50%">
  <frame src="contents_of_frame1.html">
  <frame src="#anchor_in_same_document">
  <noframes>
    ...some text...
    <h2><a name="anchor_in_same_document">Important section</a></h2>
    ...some text...
  </noframes>
</frameset>
</html>
```

**Visual rendering of a frame**

The following example illustrates the usage of the decorative attributes. We specify that frame 1 will allow no scroll bars. Frame 2 will leave white space around its contents (initially, an image file) and the frame will not be resizeable. No border will be drawn between frames 3 and 4. Borders will be drawn (by default) between frames 1, 2, and 3.
16.3 Specifying target frame information

**Note.** For information about current practice in determining the target of a frame, please consult the section on frames [p.324] in the appendix.

**Attribute definitions**

```
target = [frame-target][p.51][[CI][p.43]
```

This attribute specifies the name of a frame where a document is to be opened.

By assigning a name to a frame via the `name` attribute, authors can refer to it as the "target" of links defined by other elements. The `target` attribute may be set for elements that create links (A, LINK), image maps (AREA), and forms (FORM).

Please consult the section on target frame names [p.51] for information about recognized frame names.

This example illustrates how targets allow the dynamic modification of a frame’s contents. First we define a frameset in the document `frameset.html`, shown here:

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN" 
     "http://www.w3.org/TR/REC-html40/frameset.dtd">
<html>
  <head>
    <title>A frameset document</title>
  </head>
  <frameset rows="50%,50%">
    <frame name="fixed" src="init_fixed.html">
    <frame name="dynamic" src="init_dynamic.html">
  </frameset>
</html>
```

Then, in `init_dynamic.html`, we link to the frame named "dynamic".
Activating either link opens a new document in the frame named "dynamic" while the other frame, "fixed", maintains its initial contents.

**Note.** A frameset definition never changes, but the contents of one of its frames can. Once the initial contents of a frame change, the frameset definition no longer reflects the current state of its frames.

There is currently no way to encode the entire state of a frameset in a URI. Therefore, many user agents do not allow users to assign a bookmark to a frameset.

Framesets may make navigation forward and backward through your user agent’s history more difficult for users.

### 16.3.1 Setting the default target for links

When many links in the same document designate the same target, it is possible to specify the target once and dispense with the `target` attribute of each element. This is done by setting the `target` attribute of the `BASE` element.

We return to the previous example, this time factorizing the target information by defining it in the `BASE` element and removing it from the `A` elements.
16.3.2 Target semantics

User agents should determine the target frame in which to load a linked resource according to the following precedences (highest priority to lowest):

1. If an element has its target attribute set to a known frame, when the element is activated (i.e., a link is followed or a form is processed), the resource designated by the element should be loaded into the target frame.
2. If an element does not have the target attribute set but the BASE element does, the BASE element’s target attribute determines the frame.
3. If neither the element nor the BASE element refers to a target, the resource designated by the element should be loaded into the frame containing the element.
4. If any target attribute refers to an unknown frame F, the user agent should create a new window and frame, assign the name F to the frame, and load the resource designated by the element in the new frame.

User agents may provide users with a mechanism to override the target attribute.

16.4 Alternate content

Authors should supply alternate content for those user agents that do not support frames or are configured not to display frames.

16.4.1 The NOFRAMES element

```
<!DOCTYPE HTML.Frameset; [
<!ENTITY % noframes.content "(BODY) -(NOFRAMES)>]
]]>

<!ENTITY % noframes.content "%flow;">  

<!ELEMENT NOFRAMES %noframes.content;  
-- alternate content container for non frame-based rendering -->  
<!ATTLIST NOFRAMES %attrs;  
-- %coreattrs, %i18n, %events -->  >
```

The NOFRAMES element specifies content that should be displayed only when frames are not being displayed. User agents that support frames must only display the contents of a NOFRAMES declaration when configured not to display frames. User agents that do not support frames must display the contents of NOFRAMES in any case.

NOFRAMES can be used in the FRAMESET section of a frameset document.

For example:
16.4.2 Long descriptions of frames

The `longdesc` attribute allows authors to make frame documents more accessible to people using non-visual user agents. This attribute designates a resource that provides a long description of the frame. Authors should note that long descriptions associated with frames are attached to the frame, not the frame’s contents. Since the contents may vary over time, the initial long description is likely to become inappropriate for the frame’s later contents. In particular, authors should not include an image as the sole content of a frame.

The following frameset document describes two frames. The left frame contains a table of contents and the right frame initially contains an image of an ostrich:

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN" "http://www.w3.org/TR/REC-html40">
<html>
<head>
<title>A poorly-designed frameset document</title>
</head>
<frameset cols="20%, 80%">
  <frame src="table_of_contents.html">
  <frame src="ostrich.gif" longdesc="ostrich-desc.html">
</frameset>
</html>
```

Note that the image has been included in the frame independently of any HTML element, so the author has no means of specifying alternate text other than via the `longdesc` attribute. If the contents of the right frame change (e.g., the user selects a rattlesnake from the table of contents), users will have no textual access to the frame’s new content.

Thus, authors should not put an image directly in a frame. Instead, the image should be specified in a separate HTML document, and therein annotated with the appropriate alternate text:
16.5 Inline frames: the IFRAME element

<!ELEMENT IFRAME -- (%flow)* -- inline subwindow -->
<!ATTLIST IFRAME
%coreattrs; -- id, class, style, title --
longdesc %URI; #IMPLIED -- link to long description
(name CDATA #IMPLIED -- name of frame for targetting --
src %URI; #IMPLIED -- source of frame content --
frameborder (1|0) 1 -- request frame borders? --
marginwidth %Pixels; #IMPLIED -- margin widths in pixels --
marginheight %Pixels; #IMPLIED -- margin height in pixels --
scrolling (yes|no|auto) auto -- scrollbar or none --
align %IAAlign; #IMPLIED -- vertical or horizontal alignment --
height %Length; #IMPLIED -- frame height --
width %Length; #IMPLIED -- frame width --
>

Attribute definitions

longdesc = [uri][p.44][CT][p.43]
This attribute specifies a link to a long description of the frame. This description should supplement
the short description provided using the title attribute, and is particularly useful for non-visual
user agents.

name = [cdata][p.44][CT][p.43]
This attribute assigns a name to the current frame. This name may be used as the target of subsequent
links.

width = [length][p.46][CN][p.43]
The width of the inline frame.
The height of the inline frame.

Attributes defined elsewhere

- `id` (document-wide identifiers [p.65])
- `class` (document-wide identifiers [p.65])
- `title` (element title [p.57])
- `style` (inline style information [p.174])
- `name`, `src`, `frameborder`, `marginwidth`, `marginheight`, `scrolling` (frame controls and decoration [p.197])
- `target` (target frame information [p.200])
- `align` (alignment [p.183])

The `<IFRAME>` element allows authors to insert a frame within a block of text. Inserting an inline frame within a section of text is much like inserting an object via the `<OBJECT>` element: they both allow you to insert an HTML document in the middle of another, they may both be aligned with surrounding text, etc.

The information to be inserted inline is designated by the `src` attribute of this element. The contents of the `<IFRAME>` element, on the other hand, should only be displayed by user agents that do not support frames or are configured not to display frames.

For user agents that support frames, the following example will place an inline frame surrounded by a border in the middle of the text.

```html
<IFRAME src="foo.html" width="400" height="500"
    scrolling="auto" frameborder="1">
[Your user agent does not support frames or is currently configured not to display frames. However, you may visit <A href="foo.html">the related document.</A>]
</IFRAME>
```

Inline frames may not be resized (and thus, they do not take the `noresize` attribute).

**Note.** HTML documents may also be embedded in other HTML documents with the `<OBJECT>` element. See the section on embedded documents [p.162] for details.
16.5 Inline frames: the IFRAME element