W3C WD-WAI-PAGEAUTH-TECH-19990115

Techniques for WAI Page Author Guidelines

W3C Working Draft *15-Jan-1999*

This version:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth-tech
Latest version:
http://www.w3.org/TR/WD-WAI-PAGEAUTH/wai-pageauth-tech
Previous version:
http://www.w3.org/WAI/GL/wai-gl-techniques-19980918
Latest "WAI Page Author Guidelines":
http://www.w3.org/TR/WD-WAI-PAGEAUTH

Editors:
Wendy Chisholm <chisholm@trace.wisc.edu>
Gregg Vanderheiden <gv@trace.wisc.edu>
Ian Jacobs <ij@w3.org>

Copyright © 1999 W3C (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

Abstract

This document is a list of techniques that implement the guidelines described in "WAI Guidelines: Page Authoring". This document includes primarily techniques that HTML authors may use to implement the guidelines, but also refers to some CSS techniques as well.

While "WAI Guidelines: Page Authoring" strives to be a stable document (as a W3C Recommendation), the current document will undoubtedly evolve as technologies change and page authors discover more effective techniques for designing accessible pages.

This document is part of a series of accessibility documents published by the Web Accessibility Initiative.

Status of this document

This is a W3C Working Draft for review by W3C members and other interested parties. It is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress". This is work in progress and does not imply endorsement by, or the consensus of, either W3C or members of the WAI GL Working Group.

This document has been produced as part of the W3C WAI Activity. The goal of the WAI-GL working group is discussed in our charter.

Available formats

This document is available in the following formats:

HTML:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth-tech.html
A plain text file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth-tech.txt,
HTML as a gzip'ed tar file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth.tgz,
HTML as a zip file (this is a '.zip' file not an '.exe'):
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth.zip,
A PostScript file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth-tech.ps,
A PDF file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth-tech.pdf.

In case of a discrepancy between the various formats of the specification, http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115/wai-pageauth-tech.html is considered the definitive version.

Comments

Please send detailed comments on this document to w3c-wai-gl@w3.org. Public comments about the WAI author guidelines can also be sent to this mailing list.

Table of Contents

A. Accessibility Themes for HTML 4
A.1 Structure vs. Presentation 4
A.2 Alternative text and descriptions 7
A.3 Alternative pages 9
A.4 Keyboard access 10
A.5 Navigation 11
A.6 Comprehension 12
A.7 Other topics 12
A.8 Testing 12
A.9 Browser Support 12
B. HTML topics 13
B.1 Document structure and metadata 13
B.2 Language information 14
B.3 Text markup 14
B.4 Lists 15
B.5 Tables 17
B.6 Links 21
B.7 Images and image maps 21
B.8 Applets and other objects 26
B.9 Audio and video 28
B.10 Style and style sheets 29
B.11 Frames 33
B.12 Forms 35
B.13 Scripts 37
C. Acknowledgments 38
D. References 39
D.1 Normative References 39
E. Index of elements and attributes 39
E.1 Elements 39
E.2 Attributes 42

Priorities

Each checkpoint in this document (introduced by the word "Checkpoint") is rated according to the following system:

[Priority 1]
: This checkpoint **must** be implemented by an author, or one or more groups of users will find it impossible to access information in the document. Implementing this checkpoint is a basic requirement for some groups to be able to use Web documents.

[Priority 2]
: This checkpoint **should** be implemented by an author, or one or more groups of users will find it difficult to access information in the document. Implementing this checkpoint will significantly improve access to Web documents.

[Priority 3]
: This checkpoint **may** be implemented by an author to make it easier for one or more groups of users to access information in the document. Implementing this checkpoint will improve access to Web documents.

Some checkpoints may have more than one rating. This occurs when the importance of the checkpoint varies with the importance of the information being presented. For instance, if an author considers a vital chart or graph more important than a small graphic, it may be more important to implement a given checkpoint for the former but not the latter.

The checkpoints in this document are numbered to match their numbering in "WAI Guidelines: Page Authoring".

A. Accessibility Themes for HTML

The following sections discuss some accessibility that authors should keep in mind as they design HTML documents.

A.1 Structure vs. Presentation

When designing a document or series of documents, page authors should strive to identify the desired structure for their documents without thinking about how the documents will be presented to the user. Distinguishing the structure of a document from how the content is presented offers a number of advantages, including improved accessibility, manageability, and scalability.

Identifying what is structure and what is presentation may be a challenging task at times that authors must develop a mindset for recognizing. For instance, many authors consider that a horizontal rule (the HR element) communicates a structural division. This may be true for sighted users, but to unsighted users or users without graphical browsers, a rule has next to no meaning (One might"guess" that an HR element implies a structural division, but without other information, there is no guarantee.) Authors should use the HTML 4.0 header elements (H1-H6) to identify new sections. These may be *complemented* by visual or other cues such as horizontal rules, but should not be replaced by them.

The inverse holds as well: authors should not use structural elements to achieve presentation effects. For instance, even though the BLOCKQUOTE element may cause indented text in some browsers, that is not its meaning, only a presentation side-effect. BLOCKQUOTE elements used for indentation confuse users and search robots alike, who expect the element to be used to mark up block quotations.

The next two sections identify (by theme) precisely those elements and attributes considered structural (and some of their uses) and those that are considered to control presentation. The section on style and style sheets [p. 29] discusses how to use CSS to accomplish the same tasks as the HTML presentation elements and attributes.

Elements and attributes that are deprecated in HTML 4.0 ([HTML40] [p. 39]) appear in red and followed by an asterisk (*) in this document. Most presentation elements have been deprecated in HTML 4.0.

[Checkpoint A.14.2] If W3C technologies are used, avoid deprecated language features whenever possible. [Priority 2]

Structural elements and attributes

Document head and body	HTML [p. 41] , HEAD [p. 40] , BODY [p. 40]
Data about the document	TITLE [p. 42] , META [p. 41] , ADDRESS [p. 39]
Attributes	alt, title, longdesc
Chapters, sections, etc.	H1 [p. 40] -H6
Author-defined structures	DIV [p. 40] , SPAN [p. 42]
Attributes	class, id
Language, writing direction	BDO [p. 40]
Attributes	lang, hreflang, dir
Creating paragraphs	EM [p. 40] , STRONG [p. 42]
Acronyms and abbreviations	ACRONYM [p. 39] , ABBR [p. 39]
Subscripts and superscripts	SUB [p. 42] , SUP [p. 42]
Inserted and deleted text	INS [p. 41] , DEL [p. 40]
Attributes	cite, datetime
Quotations	BLOCKQUOTE [p. 40] , Q [p. 41]
Attributes	cite

Identifying chunks of text	DFN [p. 40] , CODE [p. 40] , SAMP [p. 41] , KBD [p. 41] , VAR [p. 42] , CITE [p. 40]
Lists	UL [p. 42] , OL [p. 41] , LI [p. 41] , DL [p. 40] , DT [p. 40] , DD [p. 40] , DIR* [p. 40] ,
Attributes	start*, value*
Tables	TABLE [p. 42] , CAPTION [p. 40] , THEAD [p. 42] , TFOOT [p. 42] , TBODY [p. 42] , COLGROUP [p. 40] , COL [p. 40] , TR [p. 42] , TH [p. 42] , TD [p. 42]
Attributes	abbr, axis, headers, scope, rowspan, span, summary
Links	LINK [p. 41] , IMG [p. 41] , OBJECT [p. 41] , PARAM [p. 41] , APPLET* [p. 39] , MAP [p. 41] , AREA [p. 39]
Attributes	shape, ismap, coords
Forms and keyboard control	FORM [p. 40] , INPUT [p. 41] , BUTTON [p. 40] , SELECT [p. 42] , OPTGROUP [p. 41] , OPTION [p. 41] , TEXTAREA [p. 42] , LABEL [p. 41] , FIELDSET [p. 40] , LEGEND [p. 41] , ISINDEX* [p. 41]
Attributes	tabindex, accesskey, label, for
Scripts	SCRIPT [p. 42] , NOSCRIPT [p. 41]

Presentation elements and attributes

Note 1. Not all attributes are listed in this section. Only those deemed pertinent to accessibility have been listed.

Note 2. While the STYLE element is presentational, it may be used to *enhance* the presentation of structural elements. The other presentation elements are often used *instead of* structural elements.

Style sheets	PRE [p. 41] , CENTER* [p. 40] , BR [p. 40] , STYLE [p. 42]
Attributes	align*, valign*, clear*, nowrap*, char, charoff , style
Spacing	
Attributes	hspace*, vspace*, cellpadding, cellspacing, compact*, type*
Text style and fonts	TT [p. 42] , I [p. 41] , B [p. 40] , BIG [p. 40] , SMALL [p. 42] , STRIKE* [p. 42] , S* [p. 41] , U* [p. 42] , FONT* [p. 40] , BASEFONT* [p. 40]
Attributes	face*, size*
Colors	
Attributes	background*, bgcolor*, color*, text*, link*, alink*, vlink*
Rules and borders	HR [p. 41]
Attributes	border, noshade*, rules, size (deprecated according to element).
Frames	FRAMESET [p. 40] , FRAME [p. 40] , IFRAME [p. 41] , NOFRAMES [p. 41]
Attributes	target, marginheight, marginwidth, frame, frameborder, noresize, rows, cols, scrolling

A.2 Alternative text and descriptions

Text is considered accessible to almost all users since it may be handled by screen readers, non-visual browsers, braille readers, etc. It is good practice, as you design a document containing non-textual information (images, graphics, applets, sounds, etc.) to think about supplementing that information with textual equivalents wherever possible.

There are several types of textual supplements to consider:

Alternative text (Alt-text)

Alternative text represents the **function** of the content. Alternative text should not describe visual appearance or how something sounds. For example, if an image of a magnifying glass is used for a search button, the alt-text would be "Search" rather than "Magnifying glass".

Quicktest! A good test to determine if alternative text is useful is to imagine reading the document aloud over the telephone. What would you say upon encountering this image to make the page comprehensible to the listener?

Related checkpoints:

1. [Checkpoint A.1.1] Provide alternative text for all images [p. 22] (e.g., in HTML, via the "alt" attribute of the IMG and INPUT elements, or via "title" or within the content of OBJECT). **Note.** This includes images used as image maps [p. 24] , spacers [p. 32] , bullets in lists [p. 16] , graphical buttons [p. 37] and links [p. 21] . [Priority 1]
2. [Checkpoint A.1.2] Provide alternative text for all applets and other programmatic objects [p. 27] (e.g., in HTML, via the "alt" attribute or within the content of APPLET, or via the "title" attribute or within the content of

OBJECT). (See also A.11) [Priority 1]

3. For all image map links:
   - [Checkpoint A.1.3] For all image map links, provide alternative text for each link [p. 24] (e.g., via the "alt" attribute of HTML AREA element). [Priority 1]
   - [Checkpoint A.1.4] For all image map links, provide redundant textual links. [p. 25] [Priority 2] if client-side image maps are used, [Priority 1] for server-side [p. 25] .
4. [Checkpoint A.1.5] Do not use an image map to create a set of buttons in a form. Instead, use separate buttons or images [p. 26] (accompanied by alternative text [p. 22]). [Priority 2]
5. [Checkpoint A.1.6] Replace ASCII art [p. 23] with an image and alternative text. [Priority 1] or [Priority 2] depending on the importance of the information (e.g., an important chart). **Note.** If the description of (important) ASCII art is long, provide a description in addition to alternative text. (See also A.2)
6. [Checkpoint A.4.1] For short animations [p. 29] such as animated "gifs" images, provide alternative text [p. 22] (See also A.1) and a long description (See also A.2) if needed. [Priority 1]
7. [Checkpoint A.6.4] Use style sheets [p. 29] to control layout and presentation wherever possible as soon as a majority of browsers in use support them well (See also A.9). Until then, simple tables (to control layout) and bitmap text with alt-text [p. 30] (for special text effects) may be used, with alternative pages used as necessary to ensure that the information on the page is accessible (See also A.14). [Priority 2]
8. [Checkpoint A.12.1] For image maps, provide alternative text for links. [p. 24] (See also A.1) [Priority 1]

Brief descriptions

Sometimes, a brief inline description of content (of a sound, an abbreviation, an image) may complement alt-text. In general, a brief description should describe visual appearance or how something sounds.

1. [Checkpoint B.1.1] Name each frame so that users can keep track of frames by name (e.g., via the "title" attribute on HTML FRAME elements). [Priority 1]
2. [Checkpoint A.1.2] Provide alternative text for all applets and other programmatic objects [p. 27] (e.g., in HTML, via the "alt" attribute or within the content of APPLET, or via the "title" attribute or within the content of OBJECT). (See also A.11) [Priority 1]

Long descriptions

In general, a long description should describe visual appearance or how something sounds. Long descriptions are generally stored in external documents (but may be inline if necessary).

1. [Checkpoint A.2.1] Provide a long description [p. 26] of all graphics, scripts, or applets that convey important information (e.g., in HTML, via "longdesc" on IMG, with a d-link (or an invisible d-link), or as content of OBJECT). [Priority 1]
2. [Checkpoint B.1.2] Describe the purpose of frames and how frames relate to each other if it is not obvious by frame names alone. (e.g., in HTML, use "longdesc". Until "longdesc" is widely supported, also use a d-link or invisible d-link). [Priority 2]

HTML allows authors to specify these substitutes in several ways. Here is a summary of attributes used for alt-text and long descriptions and the elements they apply to (by version of HTML):

To implement alt-text, use the "alt" attribute.
In HTML 4.0, applies to: IMG, AREA, INPUT, APPLET. The attribute is mandatory for the IMG and AREA elements. For other elements, use the "title" attribute.
In HTML 3.2, applies to: IMG, AREA, APPLET.
In HTML 2.0, applies to: IMG.
To implement brief descriptions, use the "title" attribute
In HTML 4.0, applies to almost every element
In HTML 3.2, applies to: LINK, A
In HTML 2.0, applies to: LINK, A
To implement long descriptions, use the "longdesc" attribute
In HTML 4.0, applies to: FRAME, IFRAME, IMG
In HTML 3.2, applies to: N/A.
In HTML 2.0, applies to: N/A.

Not every user agent supports these attributes. When required to design documents for a version of HTML known not to support one of these attributes, authors should implement alt-text or descriptions in other ways. Methods include:

- Inline alternative text or descriptions. For example, include a description of the image immediately after the image.
- Links to descriptions (called description links or "d-links"). Description links should be ordinary links (on the same page or another designed for such links) that designate a document containing a long description. The link text (content of the A element) should be "[D]".

Each of the HTML topics [p. 13] below describes prioritized scenarios for alt-text and descriptions.

Some image formats allow internal text in the data file along with the image information. If an image format supports such text (e.g., PNG) authors may also supply brief descriptions there as well.

A.3 Alternative pages

Although authors should avoid as much as possible document content for which there is no accessible alternative or supplement, it may happen that all or part of a page remains inaccessible. There are several techniques for creating accessible alternatives:

- Allow users to navigate to a separate page that is accessible that is maintained with the same frequency as the inaccessible (original) page. There are several techniques for linking to an accessible alternative page:
  1. Insert links at the top of both the main and alternative pages to allow a user to move back and forth between them.
  2. Once a majority of users use browsers that support the LINK element, use it to designate alternative documents. Browsers should then load alternative pages automatically, based on the user’s browser type.
- Instead of static alternative pages, Webmasters may set up server-side scripts that

generate accessible versions of a page on demand.

- This technique may be taken even further, if Webmasters store information in databases and then generate accessible pages in a desired format on the fly.
- If it is not possible to provide alternative pages, authors should provide a phone number, fax number, email address, or postal address where information is available and accessible, preferably 24 hours a day.
- Otherwise provide a phone number, fax number, e-mail, or postal address where information is available and accessible, preferably 24 hours a day

Example.

User agents that support LINK will load the alternative page for those users whose browsers may be identified as supporting "aural","braille", or "tty" rendering.

```
<HEAD>
<TITLE>Welcome to the Virtual Mall!</TITLE>
<LINK title="Text-only version"
      rel="alternate"
      href="text_only.html"
      media="aural, braille, tty">
</HEAD>
<BODY><P>...</BODY>
```

End example.

A.4 Keyboard access

Not every user has a graphic environment with a mouse or other pointing device. Some users rely on keyboard or voice input to navigate links, activate form controls, etc. Authors should always ensure that users may interact with a page with devices other than a pointing device. A page designed for keyboard access (in addition to mouse access) will generally be accessible to users with other input devices. What's more, designing a page for keyboard access will usually improve its overall design as well.

Keyboard access to links and form controls may be specified in two ways:

Keyboard shortcuts
: Keyboard shortcuts allow users to combine keystrokes to navigate links or form controls on a page.

Tabbing order
: Tabbing order describes a (logical) order for navigating from link to link or form control to form control (usually by pressing the "tab" key, hence the name).

Here is a summary of attributes used for keyboard access and tabbing order and the elements they apply to (by version of HTML):

To implement shortcuts, use the "accesskey" attribute.
: In HTML, 4.0 applies to: A, AREA, BUTTON, INPUT, LABEL, LEGEND, TEXTAREA
: In HTML, 3.2 applies to: N/A.
: In HTML, 2.0 applies to: N/A.

To implement tabbing order, use the "tabindex" attribute
: In HTML, 4.0 applies to: A, AREA, BUTTON, INPUT, OBJECT, SELECT, TEXTAREA
: In HTML, 3.2 applies to: N/A

In HTML, 2.0 applies to: N/A

In the following example, if the accesskey "C" is activated,"doc.html" is retrieved by the browser:

Example.

```
<A accesskey="C" href="doc.html" hreflang="en"
   title="XYZ company home page">
      XYZ company home page</A>
```

End example.

The next example assigns "U" as the access key. Typing "U" gives focus to the label, which in turn gives focus to the input control, so that the user can input text.

Example.

```
<FORM action="submit" method="post">
<P>
      <LABEL for="user" accesskey="U">name</LABEL>
      <INPUT type="text" id="user">
</FORM>
```

End example.

In the next example, we specify a tabbing order among elements (in order, "field2", "field1", "submit"):

Example.

```
<FORM action="submit" method="post">
<P>
<INPUT tabindex="2" type="text" name="field1">
<INPUT tabindex="1" type="text" name="field2">
<INPUT tabindex="3" type="submit" name="submit">
</FORM>
```

End example.

[Checkpoint A.12.2] If possible, ensure that all elements that have their own interface are keyboard operable. (See also A.12) [Priority 2] Some elements import objects whose interfaces cannot be controlled through HTML (applets, images, etc.) In such cases, users should ensure that the imported objects themselves provide accessible interfaces.

A.5 Navigation

- [Checkpoint B.3.3] Create a consistent style of presentation between pages. [Priority 3]
- [Checkpoint B.2.2] Use a clear, consistent navigation structure. [Priority 3]
- [Checkpoint B.2.3] Offer navigation bars for easy access to the navigation structure. [Priority 3]
- [Checkpoint B.2.5] Provide a description of the general layout of the site, the access features used, and how to use them. [Priority 3]
- [Checkpoint B.2.4] Offer a site map. [Priority 3]
- [Checkpoint B.2.6] Offer different types of searches for different skill levels and preferences. [Priority 3]
- [Checkpoint B.2.7] Place distinguishing information at the beginning of headings, paragraphs, lists, etc. [Priority 3]
- [Checkpoint B.2.9] Group related links, such as links used to create a navigation bar,

and attach a meaningful title on the element creating the group (e.g., in HTML use "title" on FRAME, DIV, SPAN, etc. Use class="nav" on elements creating navigation groups). [Priority 3]

- [Checkpoint B.2.10] Provide a link at the beginning of a group of related links to bypass the group. [Priority 3]

A.6 Comprehension

1. [Checkpoint B.3.1] Use the simplest and most straightforward language that is possible for the content of your site. [Priority 2]
2. [Checkpoint B.3.2] Use icons or graphics (with alternative text) where they facilitate comprehension of the page. [Priority 3]

A.7 Other topics

[Checkpoint B.2.8] Facilitate off-line browsing by creating a single downloadable file for documents that exist as a series of separate pages (e.g., by using the HTML LINK element, or creating a "zip" archive). [Priority 3]

For example, Indicate which is the first page of the document and which page follows the current one. (e.g., by using LINK).

A.8 Testing

1. Use an automated accessibility, and browser validation tool such as Bobby
2. Use the W3C HTML Validation Service, available at http://validator.w3.org/
3. Use the W3C CSS Validation Service, available at http://jigsaw.w3.org/css-validator/
4. Test your pages with a text-only browser such as Lynx or a Lynx emulator such as Lynx Viewer or Lynx-me.
5. Use multiple graphic browsers, with:
   - sounds and graphics loaded,
   - graphics not loaded,
   - sounds not loaded,
   - no mouse,
   - frames, scripts, style sheets, and applets not loaded
6. Use a several browsers, old and new.
7. Self-voicing browsers include "pwwebspeak".
8. Use a spell checker. A person reading a page with a speech synthesizer may not be able to decipher the synthesizer’s best guess for a word with a spelling error.

If, after completing these tests and adjusting your design accordingly, you find that your page is still not accessible, you must create an alternative page [p. 9] that is accessible.

A.9 Browser Support

1. [Checkpoint A.14.1] If W3C technologies are used, use the latest W3C specification whenever possible. [Priority 2] . The latest W3C specifications are available online.

B. HTML topics

The following sections list some techniques for using HTML and CSS to design accessible documents and some techniques for avoiding HTML accessibility traps. The sections are organized by topic (and mirror the organization of the HTML 4.0 specification, [HTML40] [p. 39]).

B.1 Document structure and metadata

As discussed above, authors should use structural markup wherever possible (and use it as intended by the authors of W3C specifications). Structural elements promote consistency in documents and supply information to other tools (e.g., indexing tools, search engines, programs that extract tables to databases, navigation tools that use header elements, and automatic translation software that translates text from one language into another.

Metadata

Some structural elements provide information about the document itself. This is called "metadata" about the document (Metadata is information about data). Well-crafted metadata can provide important orientation information to users. HTML elements that provide useful information about a document include:

- TITLE: The document title. Note that the TITLE element (one time only in a document) is different from the "title" attribute, which applies to almost every HTML 4.0 element. This document makes use of the "title" attribute for many occasions that require brief descriptions of element content (e.g., "a sailboat" when an image depicts a sailboat).
- ADDRESS: Can be used to provide information about the author of the page.
- LINK: Can be used to indicate alternative documents (different structure, different language, different target device, etc.).
- The META element can be used to describe metadata about a document. Many page authors use the META element to send users from one page to another automatically (usually after a time delay). This can be disorienting to some users.

  [Checkpoint A.10.1] For auto-refreshing or timed response pages, provide a second copy of the page where refresh only happens after a link has been selected (until user agents provide this ability themselves). [Priority 1]

  Note. Automatic refresh is valid behavior, supported by some user agents, but this behavior is not required by the HTML 4.0 specification.

Section headers

Sections should be introduced with the HTML header elements (H1-H6). Other markup may complement these elements to improve presentation (e.g., the HR element to create a horizontal dividing line), but visual presentation is not sufficient to identify document sections.

[Checkpoint A.6.1] Nest headings properly (e.g., in HTML, H1 - H6). [Priority 2]
Since some users skim through a document by navigating its headings, it is important to nest headings correctly, i.e., to follow an H1 with an H2, not and H3, and so on. Headings should not be used for other purposes (such as formatting text in a larger font size) since

this may disorient users; use style sheets [p. 29] for text formatting.

B.2 Language information

[Checkpoint A.7.1] Clearly identify changes in the language of text (e.g., the HTML "lang" attribute). [Priority 2] If you use a number of different languages on a page, make sure that any changes in language are clearly identified by using the "lang" attribute:

Example.

```
<P>And with a certain <SPAN lang="fr">je ne sais quoi</SPAN>,
she entered both the room, and his life, forever. <Q>My name
is Natasha,</Q> she said. <Q lang="it">Piacere,</Q>
he replied in impeccable Italian, locking the door.
```

End example.

B.3 Text markup

As mentioned above, structural elements add information to a page that may be used by browsers, search engines, and other software. Authors are encouraged to use structural elements and attributes [p. 5] whenever possible. Below we discuss how to further improve accessibility by careful use of attributes with these elements.

Emphasis

The proper HTML elements should be used to mark up emphasis: EM and STRONG. The B and I elements should not be used; they are used to create a visual presentation effect. The EM and STRONG elements were designed to indicate structural emphasis that may be rendered in a variety of ways (font style changes, speech inflection changes, etc.)

Acronyms and abbreviations

[Checkpoint A.7.2] Specify the expansion of abbreviations and acronyms (e.g., with the "title" attribute of the HTML ABBR or ACRONYM elements). [Priority 2] Mark up abbreviations and acronyms with ABBR and ACRONYM and use "title" to indicate the expansion:

Example.

```
<P>Welcome to the <ACRONYM title="World Wide Web">WWW</ACRONYM>!
```

End example.

Quotations

[Checkpoint A.6.3] Mark up quotations (e.g., with the Q and BLOCKQUOTE elements in HTML). Do not use quotation markup for formatting effects such as indentation. [Priority 2] See the language example [p. 14] above for an illustration of the Q element.

Example.

```
<BLOCKQUOTE cite="http://www.shakespeare.com/loveslabourlost">
  <P>Remuneration! O! that’s the Latin word for three farthings.
     --- William Shakespeare (Love’s Labor Lost).
  </P>
</BLOCKQUOTE>
```

End example.

B.4 Lists

[Checkpoint A.6.2] Encode list structure and list items properly (e.g., in HTML: UL, OL, DL, LI). [Priority 2] The HTML list elements DL, UL, and OL (available in HTML 3.2 and HTML 4.0) should only be used to create lists, not for formatting effects such as indentation.

When possible, use ordered (numbered) lists to help navigation. Non-visual users often "get lost" in lists, especially those with several layers of embedding and those that do not indicate the specific level of indentation for each item. In addition, it is often difficult for non-visual users to know where the list itself begins and ends and where each list item starts. Finally, if a list entry wraps to the next line, it may appear to be two separate items in the list.

Example.

Instead of nesting bulleted lists like this:

- writing tools
  - pens
    - highlighters
      - red
      - green
      - blue
    - ball-point
      - green
      - purple
      - mauve
  - pencils
    - soft lead
    - #2 lead
- erasers

which might be read by a screen reader (stripping bullets and indentation) as:

```
writing tools
pens
highlighters
red
green
blue
ball-point
green
purple
mauve
pencils
#2 lead
soft lead
erasers
```

Use ordered lists (or make sure to put distinguishing information at the beginning of each list item):

1. writing tools
    1. pens
        1. highlighters
            1. red
            2. green
            3. blue
        2. ball-point
            1. green
            2. purple
            3. mauve
    1. pencils
        1. soft lead
        2. #2 lead
2. erasers

End example.

Use style sheets to change list bullets

To change the "bullet" style of unordered list items, use style sheets. This way, if images are not loaded, the browser will draw a default bullet.

Example.

```
<HEAD>
<TITLE>Using style sheets to change bullets</TITLE>
<STYLE type="text/css">
   UL { list-style: url(star.gif) }
</STYLE>
</HEAD>
<BODY>
<UL>
   <LI>Audrey
   <LI>Laurie
   <LI>Alice
</UL>
```

End example.

Avoid using images as bullets in definition lists. However, if this method is used, be sure to provide alt-text [p. 7] for the images.

Deprecated example.

```
<DL>
   <DD><IMG src="star.gif" alt="Item">Audrey
   <DD><IMG src="star.gif" alt="Item">Laurie
   <DD><IMG src="star.gif" alt="Item">Alice
</DL>
```

Authors should avoid list styles where bullets provide additional (visual) information. However, if this is done, be sure to provide alt-text describing meaning of the bullet:

Deprecated example.

```
<DL>
<DD><IMG src="red.gif" alt="New:">Roth IRA</DD>
<DD><IMG src="yellow.gif" alt="Old:">401(k)</DD>
</DL>
```

Here is a better way to change list bullet styles (using style sheets). To further ensure that users understand differences between list items indicated visually, authors should provide a label before or after the list item phrase:

Example.

```
<HEAD>
<TITLE>Bullet styles example</TITLE>
<STYLE type="text/css">
   .newtxt { font-weight: bold;
             color: red;
             background-color: yellow }
   .newbullet { list-style : url(yellow.gif) }
</STYLE>
</HEAD>
<BODY>
<UL>
   <LI class="newbullet">Roth IRA <SPAN class="newtext">New</SPAN></LI>
   <LI> 401(k)</LI>
</UL>
</BODY>
```

End example.

B.5 Tables

Page authors may make tables more accessible in a number of ways:

- Provide a caption via the CAPTION element.
- [Checkpoint A.8.4] Provide summaries for tables (e.g., via the "summary" attribute on HTML TABLE elements). [Priority 3] Summaries are especially useful for non-visual readers.
- [Checkpoint A.8.2] Identify headers for rows and columns (e.g., the HTML TD and TH elements). [Priority 2] Future browsers and assistive technologies will be able to automatically translate tables into linear sequences if data is labeled appropriately.
- [Checkpoint A.8.3] Where tables have structural divisions beyond those implicit in the rows and columns, use markup to identify those divisions (e.g., in HTML, THEAD, TFOOT, TBODY, COLGROUP, the "axis", "scope", and "headers" attributes, etc.). [Priority 2] Identify structural groups of rows (THEAD for repeated table headers, TFOOT for repeated table footers, and TBODY for other groups of rows) and groups of columns (COLGROUP and COL). Label table elements with the"scope", "headers", and "axis" attributes so that future browsers and assistive technologies will be able to select data from a table by filtering on categories. This markup will also help browsers translate tables into linear sequences automatically (also called table "serialization"). A linear sequence is usually generated by reading a row left to right and proceeding each cell with the label of its column.
- [Checkpoint A.8.5] Provide abbreviations for header labels (e.g., in HTML, the "abbr" attribute on TH). [Priority 3] Provide terse substitutes for header labels with the "abbr" attribute on TH. These will be particularly useful for future speaking technologies that can read row and column labels for each cell. Abbreviations cut down on repetition and reading time.
- [Checkpoint A.8.1] If a table is used for layout, do not use any structural markup for the purpose of visual formatting. For example, in HTML do not use the table header (TH) element to cause the contents of a cell to be displayed centered and in bold. Other attributes of a table, such as a caption describing the layout purpose and

content of columns is valuable, particularly if some cells become navbars, frames, images, imagemaps, or lists of links. [Priority 1]

Most of the above elements and attributes are only available in HTML 4.0.

This markup will allow accessible browsers and other user agents to restructure tables for non-visual media.

For information about table headers, see the table header algorithm and discussion in the HTML 4.0 Recommendation.

The following example shows how to associate data cells with their corresponding headers by means of the "headers" attribute. The "headers" attribute specifies a list of header cells (row and column labels) associated with the current data cell. This requires each header cell to have an "id" attribute.

Example.

```
<TABLE border="1"
       summary="This table charts the number of
                cups of coffee consumed by each senator,
                the type of coffee (decaf or regular),
                and whether taken with sugar.">
  <CAPTION>Cups of coffee consumed by each senator</CAPTION>
  <TR>
      <TH id="t1">Name</TH>
      <TH id="t2">Cups</TH>
      <TH id="t3" abbr="Type">Type of  Coffee</TH>
      <TH id="t4">Sugar?</TH>
  <TR>
      <TD headers="t1">T. Sexton</TD>
      <TD headers="t2">10</TD>
      <TD headers="t3">Espresso</TD>
      <TD headers="t4">No</TD>
  <TR>
      <TD headers="t1">J. Dinnen</TD>
      <TD headers="t2">5</TD>
      <TD headers="t3">Decaf</TD>
     <TD headers="t4">Yes</TD>
</TABLE>
```

End example.

A speech synthesizer might render this tables as follows:

```
Caption: Cups of coffee consumed by each senator
Summary: This table charts the number of cups of coffee
         consumed by each senator, the type of coffee
        (decaf or regular), and whether taken with sugar.
Name: T. Sexton, Cups: 10, Type: Espresso, Sugar: No
Name: J. Dinnen, Cups: 5, Type: Decaf, Sugar: Yes
```

A visual user agent might render this table as follows:

Cups of coffee consumed by each senator

Name	Cups	Type of Coffee	Sugar?
T. Sexton	10	Espresso	No
J. Dinnen	5	Decaf	Yes

The next example associates the same header and data cells as before, but this time uses the "scope" attribute rather than"headers." "Scope" must have one of the following values: row, col, rowgroup or colgroup. Scope specifies the set of data cells to be

associated with the current header cell. This method is particularly useful for simple tables. It should be noted that the spoken rendering of this table would be identical to that of the previous example. A choice between the "headers" and "scope" attributes is dependent on the complexity of the table. It does not affect the output so long as the relationships between header and data cells are made clear in the markup.

Example.

```
<TABLE border="1"
       summary="This table charts ...">
   <CAPTION>Cups of coffee consumed by each senator</CAPTION>
   <TR>
        <TH scope="col">Name</TH>
        <TH scope="col">Cups</TH>
        <TH scope="col" abbr="Type">Type of Coffee</TH>
        <TH scope="col">Sugar?</TH>
  <TR>
        <TD>T. Sexton</TD>  <TD>10</TD>
        <TD>Espresso</TD>   <TD>No</TD>
  <TR>
        <TD>J. Dinnen</TD>  <TD>5</TD>
        <TD>Decaf</TD>       <TD>Yes</TD>
</TABLE>
```

End example.

The following example shows how to create categories within a table using the "axis" attribute.

Example.

```
<TABLE border="1">
  <CAPTION>Travel Expense Report</CAPTION>
  <TR>
       <TH></TH>
       <TH id="a2" axis="expenses">Meals
       <TH id="a3" axis="expenses">Hotels
       <TH id="a4" axis="expenses">Transport
       <TD>subtotals</TD>
  <TR>
       <TH id="a6" axis="location">San Jose
       <TH> <TH> <TH> <TD>
  <TR>
      <TD id="a7" axis="date">25-Aug-97
      <TD headers="a6 a7 a2">37.74
      <TD headers="a6 a7 a3">112.00
      <TD headers="a6 a7 a4">45.00
      <TD>
  <TR>
      <TD id="a8" axis="date">26-Aug-97
      <TD headers="a6 a8 a2">27.28
      <TD headers="a6 a8 a3">112.00
      <TD headers="a6 a8 a4">45.00
      <TD>
  <TR>
      <TD>subtotals
      <TD>65.02
      <TD>224.00
      <TD>90.00
      <TD>379.02
  <TR>
      <TH id="a10" axis="location">Seattle
      <TH> <TH> <TH> <TD>
```

```
    <TR>
        <TD id="a11" axis="date">27-Aug-97
        <TD headers="a10 a11 a2">96.25
        <TD headers="a10 a11 a3">109.00
        <TD headers="a10 a11 a4">36.00
        <TD>
    <TR>
        <TD id="a12" axis="date">28-Aug-97
        <TD headers="a10 a12 a2">35.00
        <TD headers="a10 a12 a3">109.00
        <TD headers="a10 a12 a4">36.00
        <TD>
    <TR>
        <TD>subtotals
        <TD>131.25
        <TD>218.00
        <TD>72.00
        <TD>421.25
    <TR>
        <TH>Totals
        <TD>196.27
        <TD>442.00
        <TD>162.00
        <TD>800.27
</TABLE>
```

End example.

This table lists travel expenses at two locations: San Jose and Seattle, by date, and category (meals, hotels, and transport). The following image shows how a visual user agent might render it.

Cups of coffee consumed by each senator

Name	Cups	Type of Coffee	Sugar?
T. Sexton	10	Espresso	No
J. Dinnen	5	Decaf	Yes

Wrapped text in tables

One source of problems for screen readers that do not interpret the source HTML is wrapped text in table cells. They read across the page, reading sentences on the same row from different columns as one sentence.

For example, if a table is rendered like this on the screen:

```
There is a 30% chance of              Classes at the University of Wisconsin
rain showers this morning, but they   will resume on September 3rd.
should stop before the weekend.
```

This might be read by a screen reader as:

```
There is a 30% chance of Classes at the University of Wisconsin
rain showers this morning, but they will resume on September 3rd.
should stop before the weekend.
```

Screen readers that read the source HTML will recognize the structure of each cell, but for older screen readers, authors should minimize the risk of word wrapping by limiting the amount of text in each cell. Also, the longest chunks of text should all be in the last column (rightmost for left-to-right tables). This way, if they wrap, they will still be read coherently.

Authors should test tables for wrapping with a browser window dimension of "640x480".

Quicktest! To get a better understanding of how a screen reader would read a table, run a piece of paper down the page and read your table line by line.

B.6 Links

Users who are blind often jump from link to link when skimming a page or looking for information. When they do this, only the text of the link (the "link text") is read.

"Auditory users," people who are **blind**, **have difficulty seeing**, or who are **using devices with small or no displays** are unable to scan the page quickly with their eyes and often use a list of links to get an overview of a page or to quickly find a link. When links are not descriptive enough, do not make sense when read out of context, or are not unique, the auditory user must stop to read the text surrounding each link to identify it.

[Checkpoint B.2.1] Wherever possible, make link phrases as terse as possible yet as meaningful as possible when read on their own or in succession . Avoid non-meaningful phrases, such as "click here." [Priority 2] Therefore, it is important that link text make sense when read without surrounding text. For example, authors should not use "click here" as link text several times on the same page; this requires a user browsing the page with a screen reader to step through each link and read the surrounding text to determine the purpose of the link. Instead, link text should carry sufficient information, as in"download this document in ASCII text," "view the full version in HTML," or "for the text version select this link."

When an image is used as the content of a link, specify alt-text [p. 7] for the image that makes sense in context.

Quicktest! To choose alt-text in this case, think of what you would say in words rather than an image in this context.

Example.

```
<A href="routes.html">
   <IMG src="topo.html"
        alt="Current routes at Boulders Climbing Gym">
</A>
```

End example.

If more than one link on a page shares the same link text, all those links should point to the same resource. Such consistency will help page design as well as accessibility. Link text should be terse but complete. [p. 21]

[Checkpoint A.14.4] When linking to resources that are not W3C technologies, indicate what type of resource you are linking to. For example, to link to a PDF file from an HTML document, set the "type" attribute to "application/pdf" on the A element. [Priority 3]

B.7 Images and image maps

Images may be inserted by two elements in HTML:

IMG
: Available in HTML 4.0, 3.2, 2.0

OBJECT
: Available in HTML 4.0.

Images include those that carry out simple animations (e.g., a"gif" image).

Alternative text for images

[Checkpoint A.1.1] Provide alternative text for all images (e.g., in HTML, via the "alt" attribute of the IMG and INPUT elements, or via "title" or within the content of OBJECT). **Note.** This includes images used as image maps [p. 24] , spacers [p. 32] , bullets in lists [p. 16] , graphical buttons [p. 37] and links [p. 21] . [Priority 1] When using IMG, specify alt-text with the "alt" attribute.

Example.

```
<IMG src="magnifyingglass.gif" alt="Search">
```

End example.

When using OBJECT, specify alt-text [p. 7] either with the "title" attribute:

Example.

```
<OBJECT classid="Duke.class" title="Java applet: Duke waving."
        width="50" height="50">
</OBJECT>
```

End example.

or in the body of the OBJECT element:

Example.

```
<OBJECT data="magnifyingglass.gif" type="image/gif">
   Search
</OBJECT>
```

End example.

Brief descriptions for images

When using IMG, specify a brief description [p. 8] of the image with the "title" attribute:

Example.

```
<IMG src="bell.gif" alt="Return to home page"
     title="Cow logo shaped like a bell">
```

End example.

Long descriptions for images

[Checkpoint A.2.1] Provide a long description of all graphics, scripts, or applets that convey important information (e.g., in HTML, via "longdesc" on IMG, with a d-link (or an invisible d-link), or as content of OBJECT). [Priority 1] When using IMG, specify a long description [p. 8] of the image with the "longdesc" attribute:

Example.

```
<IMG src="97sales.gif" alt="Sales for 1997"
     title="Sales pie chart"
     longdesc="sales97.html">
```

In sales97.html:

A chart showing how sales in 1997 progressed. The chart
is a bar-chart showing percentage increases in sales

by month. Sales in January were up 10% from December 1996,
sales in February dropped 3%, ..

End example.

For browsers that don't support "longdesc", provide a description link as well next to the graphic:

Example.

```
<IMG src="97sales.gif" alt="Sales for 1997">
<A href="sales.html" title="Description of 1997 sales figures">[D]</A>
```

End example.

When using OBJECT, provide a long description in the body of the element:

Example.

```
<OBJECT data="97sales.gif" type="image/gif">
       Sales in 1997 were down subsequent to our
       anticipated purchase ...
</OBJECT>
```

End example.

Or, provide a link to a long description within the body of the element:

Example.

```
<OBJECT data="97sales.gif" type="image/gif">
       Chart of our Sales in 1997.
       A <A href="desc.html">textual description</A> is available.
</OBJECT>
```

End example.

Ascii art

[Checkpoint A.1.6] Replace ASCII art with an image and alternative text. [Priority 1] or [Priority 2] depending on the importance of the information (e.g., an important chart).
Note. If the description of (important) ASCII art is long, provide a description in addition to alternative text. (See also A.2)

Avoid ascii art (character illustrations) and use real images instead since it is easier to supply alt-text [p. 7] and long descriptions [p. 8] for images. The priority of this checkpoint depends on the importance of the information (e.g., an important chart).

However, if ascii art must be used, mark it up as such:

Example.

```
<P>
<OBJECT data="cow.txt" type="text/x-ascii-art" title="drawing of a cow">
include cow ascii art from guidelines
</OBJECT>
```

End example.

Another option is to use a SPAN element with "title".

Example.

```
<P><SPAN class="smile" title="smiley in ascii art">:-)</SPAN>
```

End example.

If the description of (important) ASCII art is long, provide a description [p. 8] in addition to alt-text.

Another way to replace ascii art is to use human language substitutes. For example, <wink> might substitute for the emoticon <SPAN title="wink smiley">;-)</SPAN>, the word"therefore" could replace arrows consisting of dashes and greater than signs (e.g., -->), and the word "great" for the uncommon abbreviation "gr8".

Image maps

An image map is an image that has "active regions". When the user selects one of the regions, some action takes place -- a link may be followed, information sent to a server, etc. To make an image map accessible, authors must ensure that each action associated with a visual region may be activated without a pointing device.

Image maps are created with the MAP element (available in HTML 4.0 and 3.2). HTML allows two types of image maps: client-side (the user's browser processes a URI) and server-side (the server processes click coordinates). For all image maps, authors must supply alt-text [p. 7] (described below), and should provide brief descriptions [p. 8] where needed (see the section on brief descriptions for images [p. 22]), and long descriptions [p. 8] where needed (see the section on long descriptions for images [p. 22]).

Authors should either avoid server-side image maps (because they require a specific input device - a mouse) or provide the same functionality or information in an alternative accessible format. One way to achieve this is to provide a textual link for each active region so that each link is navigable with the keyboard [p. 10] . If you must use a server-side image map, please consult the section on server-side image maps [p. 25]

Client-side image maps

The active regions of a client-side image map are defined within the MAP element and may be created with two elements:

AREA
: Available in HTML 4.0, 3.2

A
: Available in HTML 4.0.

[Checkpoint A.12.1] For image maps, provide alternative text for links. (See also A.1) [Priority 1] Provide alternative text [p. 7] for image maps since they convey visual information.

[Checkpoint A.1.3] For all image map links, provide alternative text for each link (e.g., via the "alt" attribute of HTML AREA element). [Priority 1] If AREA is used, use the"alt" attribute:

Example.

```
<IMG src="welcome.gif" alt="Image map of areas in the library"
     usemap="#map1">
<MAP name="map1">
  <AREA shape="rect" coords="0,0,30,30"
        href="reference.html" alt="Reference">
  <AREA shape="rect" coords="34,34,100,100"
        href="media.html" alt="Audio visual lab">
</MAP>
```

End example.

The same idea, but use OBJECT instead of IMG to insert the image to provide more information about the image:

Example.

```
<OBJECT data="welcome.gif" type="image/gif" usemap="#map1">
    There are several areas in the library including
    the <A href="reference.html">Reference</A> section and the
    <A href="media.html">Audio Visual Lab</A>.
</OBJECT>
<MAP name="map1">
  <AREA shape="rect" coords="0,0,30,30"
        href="reference.html" alt="Reference">
  <AREA shape="rect" coords="34,34,100,100"
        href="media.html" alt="Audio visual lab">
</MAP>
```

End example.

[Checkpoint A.1.4] For all image map links, provide redundant textual links. [Priority 2] if client-side image maps are used, [Priority 1] for server-side [p. 25] . In addition to providing alt-text, provide redundant textual links. If the A element is used instead of AREA, the author may describe the active regions and provide redundant links at the same time:

Example.

```
<OBJECT data="navbar1.gif" type="image/gif" usemap="#map1">
<MAP name="map1">
  <P>Navigate the site.
  <A href="guide.html" shape="rect"
     coords="0,0,118,28">[Access Guide]</A>
  <A href="shortcut.html" shape="rect"
     coords="118,0,184,28">[Go]</A>
  <A href="search.html" shape="circle"
     coords="184.200,60">[Search]</A>
  <A href="top10.html" shape="poly"
     coords="276,0,373,28,50,50">[Top Ten]</A>
</MAP>
</OBJECT>
```

End example.

Note that in the previous example the MAP element is the content of the OBJECT element so that the alternative links will only be displayed if the image map (navbar1.gif) is not.

Note also that links have been separated by brackets ([]). This is to prevent screen readers from reading several adjacent links as a single link.

[Checkpoint A.13.3] Include non-link, printable characters (surrounded by spaces) between links that occur consecutively. [Priority 3] Authors should make sure they include printable characters (such as brackets or a vertical bar (|)) surrounded by spaces between adjacent links.

Server-side image maps

When a server-side image map must be used, authors should provide an alternative list of image map choices. There are three techniques:

- If an alternative list of links follows the image map, authors should indicate the existence and location of the alternative list. If IMG is used to insert the image, provide this information in the"alt" attribute. If OBJECT is used, provide it in the "title" attribute.

  Example.

```
<IMG src="welcome.gif" alt="Links to this image map follow immediately"
     usemap="#map1" title="Welcome insignia" >
<MAP name="map1">
  <AREA shape="rect" coords="0,0,30,30"
        href="reference.html" alt="Reference">
  <AREA shape="rect" coords="34,34,100,100"
        href="media.html" alt="Audio visual lab">
</MAP>

<P><A href="reference.html">[Reference]</A>
   <A href="media.html">[Audio Visual Lab]</A>
```

  End example.
- A more straightforward solution, although newer and less backwards compatible, is to include the alternative links within the body of an OBJECT element (see the previous example illustrating links in the OBJECT element [p. 25]).
- One final possibility is to create an alternative page [p. 9] that is accessible.

[Checkpoint A.1.5] Do not use an image map to create a set of buttons in a form. Instead, use separate buttons or images (accompanied by alternative text [p. 22]). [Priority 2]

B.8 Applets and other objects

Applets may be inserted by two elements in HTML:

APPLET
: Available in HTML 4.0 (deprecated), 3.2.

OBJECT
: Available in HTML 4.0.

There are several checkpoints that authors should use to ensure that audio or visual information presented by an applet is accessible:

1. [Checkpoint A.1.2] Provide alternative text for all applets and other programmatic objects [p. 27] (e.g., in HTML, via the "alt" attribute or within the content of APPLET, or via the "title" attribute or within the content of OBJECT). (See also A.11) [Priority 1]
2. Provide a long description if necessary. See the section on long descriptions for images [p. 22] .
3. [Checkpoint A.2.1] Provide a long description of all graphics, scripts, or applets that convey important information (e.g., in HTML, via "longdesc" on IMG, with a d-link (or an invisible d-link), or as content of OBJECT). [Priority 1]
4. [Checkpoint A.11.1] Where possible, make programmatic elements, [p. 27] such as scripts and applets, directly accessible. (See also A.9). [Priority 1] if information or functionality is important, and not presented elsewhere, otherwise [Priority 2].
5. [Checkpoint A.9.5] For applets and programmatic objects, when possible provide an alternative function or presentation in a format other than an applet. For example, a

canned "mpeg" movie of a physics simulation (written in Java) or a single frame of the animation saved as a "gif" image. [Priority 2]

Alternative text and descriptions for applets and programmatic objects

[Checkpoint A.1.2] Provide alternative text for all applets and other programmatic objects (e.g., in HTML, via the "alt" attribute or within the content of APPLET, or via the "title" attribute or within the content of OBJECT). (See also A.11) [Priority 1] If OBJECT is used, provide alt-text [p. 7] as the content of the element:

Example.

```
<OBJECT classid="java:Press.class" width="500" height="500"
        title="Java applet: how temperature affects pressure">
      As temperature increases, the molecules in the balloon...
</OBJECT>
```

End example.

A more complex example takes advantage of the fact the OBJECT elements may be embedded to provide for alternative representations of information:

Example.

```
<OBJECT title="How temperature affects pressure"
        classid="java:Press.class" width="500" height="500">
   <OBJECT data="Pressure.mpeg" type="video/mpeg">
      <OBJECT data="Pressure.gif" type="image/gif">
         As temperature increases, the molecules in the balloon...
      </OBJECT>
   </OBJECT>
</OBJECT>
```

End example.

[Checkpoint A.1.2] Provide alternative text for all applets and other programmatic objects [p. 27] (e.g., in HTML, via the "alt" attribute or within the content of APPLET, or via the "title" attribute or within the content of OBJECT). (See also A.11) [Priority 1] If APPLET is used, provide alt-text [p. 7] with the "alt" attribute and content in the APPLET element. This enables them to transform gracefully for those user agents that only support one of the two mechanisms ("alt" or content).

Deprecated example.

```
<APPLET code="Press.class" width="500" height="500"
        alt="Java applet: how temperature affects pressure">
      As temperature increases, the molecules in the balloon...
</APPLET>
```

Directly accessible applets

[Checkpoint A.11.1] Where possible, make programmatic elements, such as scripts and applets, directly accessible. (See also A.9). [Priority 1] if information or functionality is important, and not presented elsewhere, otherwise [Priority 2]. If an applet requires user interaction (e.g., the ability to manipulate a physics experiment) that cannot be duplicated in an alternative format, make the applet directly accessible.

For more information about accessible applets, please see:

- Java Accessibility -- Trace R&D Center
- IBM Guidelines for Writing Accessible Applications Using 100% Pure Java -- IBM Special Needs Systems

B.9 Audio and video

Audio and video should be accompanied by *text transcripts*, textual descriptions or equivalents of auditory or visual events. When these transcripts are presented synchronously with a video presentation they are called "captions" and are used by people who cannot hear the audio track of the video material. Full audio transcripts include spoken dialogue as well as any other significant sounds including on-screen and off-screen sounds, music, laughter, applause, etc. The following two examples show captions, a text transcript, and auditory descriptions.

Example.

Captions for a scene from "E.T." The phone rings three times, then is answered.

[phone rings]
[ring]
[ring]
Hello?"

End example.

Example.

Here's an example of a transcript of a clip from "The Lion King" (available at the WGBH/NCAM Descriptive Video Service site).

Simba: Yeah!
Describer: Simba races outside, followed by his parents. Sarabi smiles and nudges Simba gently toward his father. The two sit side-by-side, watching the golden sunrise.
Mufasa: Look Simba, everything the light touches is our kingdom.
Simba: Wow.

End example.

Audio information

[Checkpoint A.3.1] For stand-alone audio files, provide a textual transcript of all words spoken or sung as well as all significant sounds. [Priority 1]

[Checkpoint A.3.2] For audio associated with video, provide a textual transcript (of dialog and sounds) synchronized with the video (e.g., captions). [Priority 1]

Some media formats, such as QuickTime 3.0, allow captions and video descriptions to be added to the multimedia clip.

Until the format you are using supports alternative tracks, two versions of the movie could be made available, one with captions and descriptive video, and one without. Future technologies, on the other hand, will allow separate audio/visual files to be combined with text files via a synchronization file to create captioned audio and movies. It will also allow the user to choose from multiple sets of captions to match their reading skills. For more information see the SMIL 1.0 ([SMIL] [p. 39]) specification.

[Checkpoint A.3.3] Where sounds are played automatically, provide visual notification and transcripts. [Priority 1] or [Priority 2] depending on the importance of the sound.

This can be provided in the form of a text phrase on the page that links to a text transcript or description of the sound file. The link to the transcript should appear in a highly visible location such as at the top of the page. However, if a script is automatically loading a sound, it should also be able to automatically load a visual indication that the sound is currently being played and provide a description or transcript of the sound.

Note. Some controversy surrounds this technique because the browser should load the visual form of the information instead of the auditory form if the user preferences are set to do so. However, strategies must also work with today's browsers.

For more information, please consult NCAM's discussions of captioning and audio description on the Web

Visual information and motion

Video descriptions are used primarily by people who are blind to follow the action and other non-auditory information in video material. The description provides narration of the key visual elements without interfering with the audio or dialogue of a movie. Key visual elements include actions, settings, body language, graphics, and displayed text.

[Checkpoint A.4.2] For movies, provide auditory descriptions that are synchronized with the original audio. [Priority 1] For movies, provide auditory descriptions that are synchronized with the original audio. See the section on audio information [p. 28] for more information about multimedia formats.

[Checkpoint A.4.3] Provide text version of the auditory description that is collated with the text transcript (captions) of the primary audio track. [Priority 2] This text transcript, in conjunction with the full audio transcript described above, allows access by people with both visual and hearing disabilities. This also provides everyone with the ability to index and search for information contained in audio/visual materials.

[Checkpoint A.10.3] Movement should be avoided when possible, but if it must be used, provide a mechanism to allow users to freeze motion or updates in applets and scripts or use style sheets and scripting to create movement. (See also A.11) [Priority 2]

However, if necessary to include an applets that involves motion or updates, authors should provide a mechanism for freezing this motion. In an example of motion control created by Mark Novak, if the user presses the escape key while the Java marquee has focus, the text will be displayed statically. Authors should use animated gifs to create motion that may be suspended by the browser for people that have trouble with it.

[Checkpoint A.4.1] For short animations such as animated "gifs" images, provide alternative text [p. 22] (See also A.1) and a long description (See also A.2) if needed. [Priority 1] Depending on the context, either a brief description [p. 8] or a long description should be provided for visual information that is necessary to understanding the page. For example for an ad that displays text like a marquee, the text should be provided in the alternative text, unless there is a lot of text. For a looping image of cloud cover over the United States, if the image is in the context of a weather status report, where the information is presented in text, a less verbose description of the image is necessary. However, if the image appears on in a pedagogical setting, elaborating on cloud formations in relation to land mass, ought to be described.

See also the section on text style [p. 31] for controlling blinking.

B.10 Style and style sheets

[Checkpoint A.6.4] Use style sheets to control layout and presentation wherever possible as soon as a majority of browsers in use support them well (See also A.9). Until then, simple tables (to control layout) and bitmap text with alt-text [p. 30] (for special text effects) may be used, with alternative pages used as necessary to ensure that the information on the page is accessible (See also A.14). [Priority 2] CSS1 ([[CSS1] [p. 39]) and CSS2 ([[CSS2] [p. 39]) allow authors to duplicate almost every HTML 4.0 presentation feature and offer more power with less cost. However, until most users have browsers that support style sheets, not every presentation idiom may be expressed

satisfactorily with style sheets. In the following sections, we show how style sheets may be used to create accessible pages. We also provide examples of how to use HTML 4.0 features (e.g., tables, bitmap text) more accessibly when they must be used.

General style sheet techniques

- [Checkpoint A.9.4] For pages that use style sheets, ensure that the contents of each page are ordered and structured so that they may read in the intended order even when the style sheet is not used. [Priority 1]
- [Checkpoint A.6.5] Use relative sizing and positioning (e.g., percent values) rather than absolute (e.g., pixel or point values). [Priority 2]

Text formatting

Authors should use style sheets for text formatting rather than converting text to images. For example, stylized text on a colored background can be created with style sheets instead of as an image. This provides flexibility for people to view the text in a form that is most readable to them including magnified, in a particular color combination such as white on black, or in a particular font.

However, if you must use a bitmap to create a text effect (special font, transformation, shadows, etc.) it must be accessible.

When bitmapped text is used in a way that makes a page inaccessible, authors must supply alternative pages [p. 9] .

To make a bitmap representing text accessible, it must have alt-text [p. 7] that is the same text represented by the image.

Example.

In this example, the inserted image shows the large red characters "Example", reflected by the alt-text.

```
<P>This is an
   <IMG src="BigRedExample.gif" alt="Example"> of what we mean.
</P>
```

End example.

This is true of Drop Caps (large first letter of a paragraph) as well. However, we recommend using style sheets to create the effect, as the following example illustrates.

Example.

```
<HEAD>
<TITLE>Drop caps</TITLE>
<STYLE type="text/css">
      .dropcap { font-size : 120%; font-family : Helvetica }
</STYLE>
</HEAD>
<BODY>
<P><SPAN class="dropcap">O</SPAN>nce upon a time...
</BODY>
```

Note. As of the writing of this document, the CSS pseudo-element ':first-letter', which allows authors to refer to the first letter of a chunk of text, is not widely supported.

Text style

Authors should use style sheets instead of deprecated presentation elements and attributes [p. 6] that control visual presentation.

- [Checkpoint A.10.2] Avoid any blinking or updating of the screen that causes flicker. [Priority 1] Blinking may cause seizures in some users and is annoying to many other users. They should also provide a mechanism for freezing motion. If style sheets are used to create an effect (e.g., 'text-decoration: blink'), users may cancel the effect through style sheets as well.

  Note. Do not use the BLINK and MARQUEE elements. These elements are not part of any W3C specification for HTML (i.e., they are non-standard elements).
- Indentation: Use the CSS 'text-indent' property to indent text. Do not use the BLOCKQUOTE or any other structural element to indent text.
- Spacing: Authors should achieve spacing effects with the CSS 'word-spacing' and 'white-space' properties rather than by putting actual spaces between letters and using the PRE element.

Fonts

Instead of using deprecated presentation elements and attributes [p. 6] , use the many CSS properties to control font characteristics: 'font-family', 'font-size', 'font-size-adjust', 'font-stretch', 'font-style', 'font-variant', and 'font-weight'.

Colors

Use these CSS properties to specify colors:

- 'color', for foreground text color.
- 'background-color', for background colors.
- 'border-color', for border colors.
- For link colors, see the :link, :visited, and :active pseudo-classes.

[Checkpoint A.5.1] Don't use color to convey information unless the information is also clear from the markup and/or text. [Priority 1] For example, in this document, examples appear in a different color than the rest of the text. However, that is not enough to identify them as examples, so we precede each one with the word "Example." or "Deprecated example."

Quicktest! To text whether your page passes the text, examine it with a monochrome monitor or colors turned off.

[Checkpoint A.5.2] Use foreground and background color combinations that provide sufficient contrast when viewed by someone with color deficits or when viewed on a black and white screen. [Priority 1]

Quicktest! To test whether color contrast is sufficient to be read by people with color deficiencies or by those with low resolution monitors, print pages on a black and white printer (with backgrounds and colors appearing in grayscale).

For more information about colors and contrasts, please consult The Lighthouse and

Layout, positioning, and alignment

Layout, positioning, and alignment should be done through style sheets (notably by using CSS floats and absolute positioning).

When laying out tabular information, authors should use tables that are designed for accessibility (see the section on tables [p. 17] . Do not use PRE to create a tabular layout of text.

[Checkpoint A.13.5] Until user agents and screen readers are able to handle text presented side-by-side, *all* tables that lay out text in parallel, word-wrapped columns require a linear text alternative (on the current page or some other). [Priority 2] However, until user agents and screen readers are able to handle text presented side-by-side, all tables that lay out text in parallel, word-wrapped columns require a linear text alternative (on the current page or some other). See also the section on tables [p. 17] for additional information on table accessibility.

Provide alternative text for all images, including invisible or transparent images.

If authors cannot use style sheets and must use invisible or transparent images to lay out images on the page, they must supply"null" (alt=) or "white space" (alt=" ") alt-text, whichever the context requires. Note that the HTML 4.0 specification recommends that attribute values not contain leading or trailing spaces. It states, "User agents may ignore leading and trailing white space in CDATA attributes values (e.g., "myval " may be interpreted as "myval")." Therefore, empty alt-text (alt=" ") might be ignored.

Deprecated example.

In this example, an image is used to create a carefully defined space between words or graphics. "White space" alt-text is used to prevent the words from running together when the image is not loaded:

```
my poem requires a big space<IMG src="10pttab.gif" alt="&amp;nbsp;&amp;nbsp;&amp;nbsp;">here
```

In this next example, an image is used to force a graphic to appear in a certain position:

```
<IMG src="spacer.gif" alt="spacer">
<IMG src="colorfulwheel.gif" alt="The wheel of fortune">
```

Rules and borders

Rules and borders may convey the notion of "separation" to visually enabled users but that meaning cannot be inferred out of a visual context.

While authors may use HR to create a horizontal rule, they should do so in a way that also conveys the structure in a non-visual way (e.g., by using DIV in conjunction with the "class" attribute).

Example.

```
<DIV class="navigation-bar">
   <HR title="navigation-bar">
   <A rel="Next" href="next.html">[Next page]</A>
   <A rel="Previous" href="previous.html">[Prevous page]</A>
   <A rel="First" href="first.html">[First page]</A>
</DIV>
```

End example.

When using graphics (e.g., horizontal rules) as section separators, authors should provide a brief description [p. 8] of what the graphic represents to the visually enabled user via the "title" attribute. Hence, in the previous example, we specified title="navigation-bar".

Example.
In this example, a red line is used to separate Chapter 7 from Chapter 8:

```
<IMG src="redline.gif" alt="redline graphic"
     title="End of Chapter 7 - Visual Displays">
<H1>Chapter 8 - Auditory and Tactile Displays</H1>
```

We recommend using style sheets to accomplish such styling of the line:

```
<HEAD>
<TITLE>Redline with style sheets</TITLE>
<STYLE type="text/css">
     HR.redline { color : red }
</STYLE>
</HEAD>
<BODY>
<HR class="redline" title="End of Chapter 7 - Visual Displays">
<H1>Chapter 8 - Auditory and Tactile Displays</H1>
</BODY>
```

End example.

B.11 Frames

For visually enabled users, frames may organize a page into different zones. For non-visual users, relationships between the content in frames (e.g., one frame has a table of contents, another the contents themselves). must be conveyed through other means.

Name frames for easy orientation

[Checkpoint B.1.1] Name each frame so that users can keep track of frames by name (e.g., via the "title" attribute on HTML FRAME elements). [Priority 1]
Example.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
<HEAD>
<TITLE>A simple frameset document</TITLE>
</HEAD>
<FRAMESET cols="10%, 90%"
          title="Our library of electronic documents">
    <FRAME src="nav.html" title="Navigation bar">
    <FRAME src="doc.html" title="Documents">
    <NOFRAMES>
       <A href="lib.html" title="Library link">
            Select to go to the electronic library</A>
    </NOFRAMES>
</FRAMESET>
```

End example.

Long descriptions of frames

[Checkpoint B.1.2] Describe the purpose of frames and how frames relate to each other if it is not obvious by frame names alone. (e.g., in HTML, use "longdesc". Until "longdesc" is widely supported, also use a d-link or invisible d-link). [Priority 2]

Example.

In this example, the file "chart.html" inserts a complex chart. The "longdesc" attribute designates the file "chart-desc.html", which contains a long description of the chart.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
<HEAD>
<TITLE>A simple frameset document</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
     <FRAME src="table_of_contents.html">
     <FRAME src="chart.html" longdesc="chart-desc.html">
</FRAMESET>
```

Note that if the second frame's contents change (from"chart.html"), the long description will no longer apply. Long descriptions of frames should only be used for frames with static contents.

End example.

Invisible d-links

Ensure documents are readable without frames

[Checkpoint A.9.1] For frames, provide a fallback page for pages that contain frames (e.g., by using NOFRAMES in HTML at the end of each frameset). [Priority 1]

Example.

In this example, if the user reads "top.html":

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
<HEAD>
<TITLE>This is top.html</TITLE>
</HEAD>
<FRAMESET cols="50%, 50%" title="Our big document">
    <FRAME src="main.html" title="Where the content is displayed">
    <FRAME src="table_of_contents.html" title="Table of Contents">
    <NOFRAMES>
        <A href="table_of_contents.html">Table of Contents.</A>
     <!-- other navigational links that are available in main.html
          are available here also. -->
    </NOFRAMES>
</FRAMESET>
</HTML>
```

and the user agent is not displaying frames, the user will have access (via a link) to a non-frames version of the same information.

End example.

Always make the source of a frame an HTML document

Authors must provide descriptions of frames so that their contents and the relationships between frames make sense. Note that as the contents of a frame change, so must change any description. This is not possible if an IMG is inserted directly into a frame, as in this deprecated example:

Deprecated example.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
<HEAD>
<TITLE>A bad frameset document</TITLE>
</HEAD>
<FRAMESET cols="100%" title="Do not do this">
   <FRAME name="badframe"
          src="apples.gif" title="Apples">
</FRAMESET>
</HTML>
```

Note that if, for example, a link causes a new image to be inserted into the frame:

```
  <P>Visit a beautiful grove of
  <A target="badframe" href="oranges.gif" title="Oranges">oranges</A>
```

the initial title of the frame ("Apples") will no longer match the current content of the frame ("Oranges").

[Checkpoint A.9.2] For frames, ensure that the source of each frame is a markup file, such as HTML. [Priority 1] To solve this problem, authors should always make the source ("src") of a frame an HTML file. Images may be inserted into the HTML file and their descriptions will evolve correctly.

Example.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
<HEAD>
<TITLE>A bad frameset document</TITLE>
</HEAD>
<FRAMESET cols="100%" title="Do not do this">
<FRAME name="goodframe" src="apples.html" title="Apples">
</FRAMESET>
</HTML>

   <!-- In apples.html -->
   <P><IMG src="apples.gif" alt="Apples">
```

End example.

Avoid opening a new window as the target of a frame

[Checkpoint A.13.1] Do not use pop-up windows, new windows, or change active window unless the user is aware that this is happening. [Priority 2] For example, authors should avoid specifying a new window as the target of a frame with target="_blank".

B.12 Forms

Make controls keyboard accessible

[Checkpoint A.12.3] Create a logical tab order through links, form controls, and objects (e.g., in HTML, via the "tabindex" attribute or through logical page design). [Priority 3]

[Checkpoint A.12.4] Provide keyboard shortcuts to links, including those in client-side image maps, form controls, and groups of form controls (e.g., in HTML, via the "accesskey" attribute). [Priority 3] See the section on keyboard access [p. 10] for more information.

Group form controls

[Checkpoint B.1.3] Group form controls (e.g., in HTML use the FIELDSET and LEGEND elements). [Priority 2] for radio buttons and checkboxes, [Priority 3] for other controls. Authors should group form controls into semantic units with the FIELDSET element and label those units with the LEGEND element (both available in HTML 4.0):

Example.

```
<FORM action="http://somesite.com/adduser" method="post">
   <FIELDSET>
   <LEGEND>Personal information</LEGEND>
   <LABEL for="firstname">First name: </LABEL>
   <INPUT type="text" id="firstname" tabindex="1">
   <LABEL for="lastname">Last name: </LABEL>
   <INPUT type="text" id="lastname" tabindex="2">
   ...more personal information...
   </FIELDSET>
   <FIELDSET>
   <LEGEND>Medical History</LEGEND>
   ...medical history information...
   </FIELDSET>
</FORM>
```

End example.

Label form controls explicitly

[Checkpoint B.1.4] Associate labels explicitly with their controls (e.g., in HTML use LABEL and its "for" attribute). [Priority 2] An example of LABEL used with "for" in HTML 4.0 is given in the previous section.

[Checkpoint A.13.4] For all form controls with implicitly associated labels, ensure that the label is properly positioned. The label must immediately precede its control on the same line (allowing more than one control/label per line) or be on the line before the control (with only one label and one control per line). [Priority 2]

Group menu options

[Checkpoint B.1.5] Create a hierarchy of long lists of choices (e.g., with the HTML OPTGROUP element). [Priority 2] For long lists of selections (which are hard to remember), authors should group items into a hierarchy using the OPTGROUP element (available in HTML 4.0).

Example.

```
<FORM action="http://somesite.com/prog/someprog" method="post">
 <P>
 <SELECT name="ComOS">
     <OPTGROUP label="PortMaster 3">
       <OPTION label="3.7.1" value="pm3_3.7.1">PortMaster 3 with ComOS 3.7.1
       <OPTION label="3.7" value="pm3_3.7">PortMaster 3 with ComOS 3.7
       <OPTION label="3.5" value="pm3_3.5">PortMaster 3 with ComOS 3.5
     </OPTGROUP>
     <OPTGROUP label="PortMaster 2">
       <OPTION label="3.7" value="pm2_3.7">PortMaster 2 with ComOS 3.7
       <OPTION label="3.5" value="pm2_3.5">PortMaster 2 with ComOS 3.5
     </OPTGROUP>
     <OPTGROUP label="IRX">
       <OPTION label="3.7R" value="IRX_3.7R">IRX with ComOS 3.7R
```

```
        <OPTION label="3.5R" value="IRX_3.5R">IRX with ComOS 3.5R
      </OPTGROUP>
  </SELECT>
</FORM>
```

End example.

Techniques for specific controls

[Checkpoint A.13.2] Include default, place-holding characters in edit boxes and text areas (e.g., TEXTAREA and INPUT in HTML). [Priority 3]

Example.

```
<FORM action="http://somesite.com/prog/text-read" method="post">
     <P>
     <TEXTAREA name=yourname rows="20" cols="80">
     Please enter your name here.
     </TEXTAREA>
     <INPUT type="submit" value="Send"><INPUT type="reset">
     </P>
</FORM>
```

End example.

Provide alt-text [p. 7] for images used as "submit" buttons:

Example.

```
<FORM action="http://somesite.com/prog/text-read" method="post">
<P>
<INPUT type="image" name=submit src="button.gif" alt="Submit">
</FORM>
```

End example.

[Checkpoint A.1.5] Do not use an image map to create a set of buttons in a form. Instead, use separate buttons or images [p. 26] (accompanied by alternative text [p. 22]). [Priority 2]

Also see the section on keyboard access [p. 10] since this applies to form controls.

B.13 Scripts

Authors must ensure that pages are accessible with scripts turned off or in browsers that don’t support scripts.

Alternative presentation of scripts

[Checkpoint A.9.3] For scripts that present critical information or functions, provide an alternative, equivalent presentation or mechanism (e.g., by using NOSCRIPT in HTML, or a server-side script). [Priority 1] One way to accomplish this is with the NOSCRIPT element (available in HTML 4.0). The content of this element is rendered with scripts are not enabled.

Example.

```
<SCRIPT type="text/tcl">
 ...some Tcl script to show a billboard of sports scores...
</SCRIPT>
<NOSCRIPT>
   <P>Results from yesterday's games:</P>
   <DL>
      <DT>Bulls 91,  Sonics 80.
      <DD><A href="bullsonic.html">Bulls vs. Sonics game highlights</A>
      ...more scores...
   </DL>
</NOSCRIPT>
```

End example.

C. Acknowledgments

WAI Markup Guidelines Working Group Co-Chairs:
Chuck Letourneau, Starling Access Services
Gregg Vanderheiden, Trace Research and Development
Staff contacts:
Judy Brewer and Daniel Dardailler

Special thanks to the following people who have made major contributions to shaping the content of this Working Draft:
Harvey Bingham, Daniel Dardailler, Al Gilman, Jason White

In addition we would like to thank the following people who have contributed through review and comment:
Judy Brewer, Kevin Carey, Chetz Colwell, Neal Ewers, Geoff Freed, Larry Goldberg, Jon Gunderson, Phill Jenkins, Leonard Kasday, George Kerscher, Marja-Riitta Koivunen, Josh Krieger, Scott Luebking, William Loughborough, Murray Maloney, Charles McCathieNevile, MegaZone (Livingston Enterprises), Masafumi Nakane, Charles Oppermann, Mike Paciello, David Pawson, Michael Pieper, Greg Rosmaita, Liam Quinn, Dave Raggett, T.V. Raman, Robert Savellis, Jutta Treviranus, Steve Tyler, Jaap van Lelieveld

The original draft of this W3C working document is based on The Unified Web Site Accessibility Guidelines compiled by the Trace R & D Center, University of Wisconsin under funding from the National Institute on Disability and Rehabilitation Research (NIDRR), U.S. Dept. of Education.

For a full list of Contributors to the Unified Guidelines is available at http://trace.wisc.edu/docs/html_guidelines/version7.htm .

We would also like to thank the people listed in the References section below whose works were used in the compilation of the unified guidelines on which this document is based.

D. References

D.1 Normative References

[CSS1]
"CSS, level 1 Recommendation", B. Bos, H. Wium Lie, eds. The CSS1 Recommendation is available at:
http://www.w3.org/TR/REC-CSS1.

[CSS2]
"CSS, level 2 Recommendation", B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, eds. The CSS2 Recommendation is available at:
http://www.w3.org/TR/REC-CSS2/.

[HTML40]
"HTML 4.0 Recommendation", D. Raggett, A. Le Hors, and I. Jacobs, eds. The HTML 4.0 Recommendation is available at:
http://www.w3.org/TR/REC-html40/.

[HTML32]
"HTML 3.2 Recommendation", D. Raggett, ed. The HTML 3.2 Recommendation is available at:
http://www.w3.org/TR/REC-html32.

[SMIL]
"Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", P. Hoschka, editor. The SMIL 1.0 Recommendation is available at:
http://www.w3.org/TR/REC-smil/

E. Index of elements and attributes

E.1 Elements

This index lists all elements in HTML 4.0 and whether they are defined in earlier versions of HTML. Elements and attributes that are deprecated in HTML 4.0 ([HTML40] [p. 39]) are followed by an asterisk (*). Elements that are obsolete in HTML 4.0 or don't exist in any version of HTML (2.0, 3.2, 4.0) do not appear in this table. An entry of "N/A" means that an element is not discussed in this document.

Name	Also defined in	Related HTML topics
A	2.0, 3.2	Keyboard access [p. 10] , Links [p. 21]
ABBR		Acronyms and abbreviations [p. 14]
ACRONYM		Acronyms and abbreviations [p. 14]
ADDRESS	2.0, 3.2	Metadata [p. 13]
APPLET*	3.2	Applets and other objects [p. 26]
AREA	3.2	Keyboard access [p. 10] , Images and image maps [p. 21] ,

B	2.0, 3.2	Text style [p. 31] , Emphasis [p. 14]
BASE	2.0, 3.2	Links [p. 21]
BASEFONT*	3.2	Fonts [p. 31]
BDO		N/A
BIG	3.2	Text style [p. 31]
BLOCKQUOTE	2.0, 3.2	Quotations [p. 14] , Text style [p. 31]
BODY	2.0, 3.2	N/A
BR	2.0, 3.2	Layout, positioning, and alignment [p. 32]
BUTTON		Keyboard access [p. 10] , Server-side image maps [p. 25] , Forms [p. 35]
CAPTION		Tables [p. 17]
CENTER*	3.2	Layout, positioning, and alignment [p. 32]
CITE	2.0, 3.2	N/A
CODE	2.0, 3.2	N/A
COL		Tables [p. 17]
COLGROUP		Tables [p. 17]
DD	2.0, 3.2	Lists [p. 15]
DEL		N/A
DFN	3.2	N/A
DIR*	2.0, 3.2	N/A
DIV	3.2	N/A
DL	2.0, 3.2	Lists [p. 15]
DT	2.0, 3.2	Lists [p. 15]
EM	2.0, 3.2	Emphasis [p. 14] , Text style [p. 31]
FIELDSET		Grouping form controls [p. 35]
FONT*	3.2	Fonts [p. 31]
FORM	2.0, 3.2	Forms [p. 35]
FRAME		Frames [p. 33]
FRAMESET		Frames [p. 33]
H1-H6	2.0, 3.2	Section headers [p. 13]
HEAD	2.0, 3.2	N/A

HR	2.0, 3.2	Rules and border [p. 32] , Section headers [p. 13]
HTML	2.0, 3.2	N/A
I	2.0, 3.2	Text style [p. 31] , Emphasis [p. 14]
IFRAME		Frames [p. 33]
IMG	2.0, 3.2	List bullets [p. 16] ,
INPUT	2.0, 3.2	Images and image maps [p. 21] , Keyboard access [p. 10] , Layout, positioning, and alignment [p. 32] , Rules and borders [p. 32]
INS		N/A
ISINDEX*	2.0, 3.2	N/A
KBD	2.0, 3.2	N/A
LABEL		Keyboard access [p. 10] , Form labels [p. 36]
LEGEND		Keyboard access [p. 10] ,
LI	2.0, 3.2	Lists [p. 15]
LINK	2.0, 3.2	Links [p. 21] , Metadata [p. 13]
MAP	3.2	Images and image maps [p. 21]
MENU*	2.0, 3.2	N/A
META	2.0, 3.2	Metadata [p. 13]
NOFRAMES		Frames [p. 33]
NOSCRIPT		Scripts [p. 37]
OBJECT		Keyboard access [p. 10] , Images and image maps [p. 21] , Applets and other objects [p. 26]
OL	2.0, 3.2	Lists [p. 15]
OPTGROUP	2.0, 3.2	Forms [p. 35]
OPTION	2.0, 3.2	Forms [p. 35]
P	2.0, 3.2	N/A
PARAM		Forms [p. 35]
PRE	2.0, 3.2	Layout, positioning, and alignment [p. 32] , Text style [p. 31]
Q		Quotations [p. 14]
S		Text style [p. 31]
SAMP	2.0, 3.2	N/A

SCRIPT		Scripts [p. 37]
SELECT	2.0, 3.2	Keyboard access [p. 10] , Forms [p. 35]
SMALL	3.2	Text style [p. 31]
SPAN		N/A
STRIKE*	3.2	Text style [p. 31]
STRONG	2.0, 3.2	Emphasis [p. 14] , Text style [p. 31]
STYLE		Keyboard access [p. 29] ,
SUB	3.2	N/A
SUP	3.2	N/A
TABLE	3.2	Tables [p. 17]
TBODY		Tables [p. 17]
TD	3.2	Tables [p. 17]
TEXTAREA	2.0, 3.2	Keyboard access [p. 10] , Forms [p. 35]
TFOOT		Tables [p. 17]
TD	3.2	Tables [p. 17]
TD	3.2	Tables [p. 17]
THEAD		Tables [p. 17]
TITLE	2.0, 3.2	Metadata [p. 13]
TR	3.2	Tables [p. 17]
TT	2.0, 3.2	Text style [p. 31]
U	3.2	Text style [p. 31]
UL	2.0, 3.2	Lists [p. 15]
VAR	2.0, 3.2	N/A

E.2 Attributes

Attributes that are deprecated in HTML 4.0 ([HTML40] [p. 39]) are marked up in this document and followed by an asterisk (*). Attributes that are obsolete in HTML 4.0 or don’t exist in any version of HTML (2.0, 3.2) do not appear in this table. An entry of "N/A" means this attribute is not discussed in this document.

Under construction