W3C WD-WAI-PAGEAUTH-19990217

Web Content Accessibility Guidelines

W3C Working Draft *17-Feb-1999*

This version:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990217
Latest version:
http://www.w3.org/TR/WD-WAI-PAGEAUTH
Previous version:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990115
Related documents:
Techniques for Web Content Accessibility Guidelines
Checklist for Web Content Accessibility Guidelines

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 guidelines that Web content developers [p. 15] should follow in order to make their pages more accessible for people with disabilities as well as more useful to other users, new page viewing technologies (mobile and voice), and electronic agents such as indexing robots. Tools that generate documents in HTML (authoring tools, file conversion packages, or other products) should make it easy for authors to produce documents that follow these guidelines. This document is part of a series of accessibility documents published by the Web Accessibility Initiative.

Accessibility does not mean minimal page design, it means thoughtful page design. These guidelines outline procedures for authors, particularly those using multimedia content, to ensure that the content and functions provided by those elements are available to all users. In general, authors should not be discouraged from using multimedia, but rather should use it in a manner which ensures that the material they publish is accessible to the widest possible audience.

In this document references are made to some technologies that are not yet widely supported or not supported consistently in current user agents (browsers), such as new features introduced in HTML 4.0. Since this document is "frozen" we can not list which features are supported as of "today." Therefore, the Techniques for Web Content Accessibility Guidelines document links to information about feature support of various browsers.

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.

Please note that previous versions of this document were entitled "WAI Page Author Guidelines".

This document has been produced as part of the W3C WAI Activity, and is intended as a draft of a Proposed Recommendation for authoring accessible Web pages. 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-19990217/wai-pageauth.html
A plain text file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990217/wai-pageauth.txt,
HTML as a gzip'ed tar file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990217/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-19990217/wai-pageauth.zip,
A PostScript file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990217/wai-pageauth.ps,
A PDF file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990217/wai-pageauth.pdf.

In case of a discrepancy between the various formats of the specification, http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990217/wai-pageauth.html is considered the definitive version.

Comments

Please send detailed comments on this document to w3c-wai-gl@w3.org.

Table of Contents

A. Transform Gracefully 4
A.1 Provide alternative text for all images, applets, and image maps. 5
A.2 Provide descriptions for important graphics, scripts, or applets if they 5
A.3 Provide textual equivalents (captions) for all audio information. 6
A.4 Provide verbal descriptions of moving visual information in both auditory 6
A.5 Ensure that text and graphics are perceivable and understandable when ... 7
A.6 Indicate structure with structural elements, and control presentation with presentation 7
A.7 Provide supplemental information needed to pronounce or interpret abbreviated or 7
A.8 Ensure that tables have necessary markup to be properly restructured 8
A.9 Ensure that pages using newer technologies will transform gracefully into 8
A.10 Ensure that moving, blinking, scrolling, or auto-updating objects or pages 9
A.11 Ensure that embedded custom user interfaces are directly accessible. ... 10
A.12 Use features that enable activation of page elements via input 10
A.13 Use interim accessibility solutions so that assistive technologies and older 10
A.14 Use W3C technologies and do so according to specification. Where ... 11
B. Orientation, Navigation, and Comprehension 12
B.1 Provide context and orientation information for complex pages or elements. 12
B.2 Provide mechanisms that facilitate navigation within your site. 13
B.3 Use language and formats that facilitate comprehension of information. ...13
C. Appendix - Testing 14
C.1 Validate your pages and assess the accessibility with automated tools, ... 14
D. Appendix - Definitions 14
E. Acknowledgments 15
F. References 16

Priorities

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

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

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

A. Transform Gracefully

Make sure pages transform gracefully across users, technologies, and situations

To "transform gracefully" means that a page remains usable despite user, technological, or situational constraints. User constraints require that the information is presented so that they are able to perceive or interact with it due to a physical or sensory limitation on their part. However, this limitation might be caused by a situation. The user who is blind has the same limitations as someone whose eyes are busy while they drive a car. Technological constraints are defined by the device a person is using to access a document. A small screen on a PDA creates a similar usability constraint to a person with low-vision who has magnified their screen such that they are looking at the information as if through a very small screen.

Since HTML, XML, and other W3C technologies (such as MathML and SMIL) are designed with the flexibility to create documents that may be formatted in various ways on a variety of platforms, by virtue they support accessible design. Non-accessible pages are a result of giving up this flexibility. Creating pages that transform gracefully is not more costly, but requires a different design approach that also makes pages compatible with emerging mobile technologies. The following section A guidelines address the issue of creating pages that transform gracefully. They all stem from the following general guideline:

> ***Always separate the content on your site (what you say), and the way you choose to structure that content (how you organize it), from the way the content and structure are presented (how you want people to "see" it or perceive it).*** (If the content is sensory specific, such as audio or video, make it available in a form that allows presentation in other senses.)

Documents that transform gracefully are:

1. **Able to be perceived entirely visually and entirely through auditory means.** This does not mean creating an entire auditory version of your site. *Screen readers* will be able to speak all information on a page *as long as it is available in text*.
2. **Operable on various types of hardware including devices without mice, with small, low resolution, or black and white screens, with only voice or text output,**

without screens, etc. Due to the inherent flexibility of the design of W3C technologies your pages will be cross platform if you follow the basic principles outlined in this document.

Guidelines A.1 - A.12 address these issues.

A.1 Provide alternative text for all images, applets, and image maps.

This includes images used as submit buttons, bullets in lists, and all of the links within an image map as well as invisible images used to layout a page. Alternative text does not describe the visual appearance of an image, applet, or image map. Rather, it is used to represent the function that the image, applet, or image map performs whether it be decorative, informative, or for purposes of layout. If alternative text is not provided, users who are **blind**, have **low vision**, or **any user who cannot or has chosen not to view graphics** will not know the purpose of the visual components on the page. Since "bare" ASCII art [p. 14] (characters that form images) does not allow alt-text, it must be marked up specially for this purpose.

Checkpoints:

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, spacers, bullets in lists, graphical buttons, links, and to present math equations. [Priority 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) [p. 10] [Priority 1]
3. For all image map links, provide text for each link (e.g., via the "alt" attribute of HTML AREA element or by using the MAP element with anchors defined with the A element). [Priority 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.
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). [Priority 2]
6. For ASCII art , either replace it with an image and alternative text or provide a description and a means to skip over the art (e.g., a link). [Priority 1] or [Priority 2] depending on the importance of the information (e.g., an important [p. 15] chart). **Note.** If the description of (important) ASCII art is long, provide a description in addition to alternative text. (See also A.2) [p. 5]

A.2 Provide descriptions for important graphics, scripts, or applets if they are not fully described through alternative text or in the document’s content.

Otherwise, important [p. 15] information presented graphically (charts, billboards, diagrams) will not be perceivable to people with **blindness**, some people with **low vision**, and users who have **chosen not to view graphics, scripts, or applets** or whose **browser does not support scripts or applets.**

Note. Long descriptions serve a different purpose than alternative text: a long description provides a *description* of the visual information of the image whereas alternative text acts as a substitute for the *function* of the image.

Checkpoints:

1. Provide a long description of all graphics, scripts, or applets that convey important [p. 15] 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]

A.3 Provide textual equivalents (captions) for all audio information.

If the audio is associated with a visual presentation (movie or animation), **synchronize the textual equivalents with the visual presentation**. Otherwise, users who are **deaf**, or **hard of hearing**, or **any user who cannot or has chosen not to hear sound** cannot perceive the information presented through speech, sound effects, music, etc.

Checkpoints:

1. For stand-alone audio files, provide a textual transcript of all words spoken or sung as well as all significant sounds. [Priority 1]
2. For audio associated with video, provide a textual transcript (of dialog and sounds) synchronized with the video (e.g., captions). [Priority 1]
3. Where sounds are played automatically, provide visual notification and transcripts. [Priority 1] or [Priority 2] depending on the importance of the sound.

A.4 Provide verbal descriptions of moving visual information in both auditory and text form (for movies, animations, etc.).

If the visual presentation is associated with an auditory presentation (e.g., for a movie), **synchronize the audio version of the descriptions with the existing auditory presentation** and **collate [p. 15] the text version of the descriptions with the text transcript (captions) of the primary audio track.** Otherwise, if actions, body language, or other visual cues present information that is not expressed through auditory means as well (through dialogue, sound effects, etc.), **users who cannot see** (or look at) the page will not be able to perceive it. The collated text version allows access to the information by **devices that do not play movies and by people who are deaf-blind**.

Checkpoints:

1. For short animations such as animated "gifs" images, provide alternative text (See also A.1) [p. 5] and a long description (See also A.2) [p. 5] if needed. [Priority 1]
2. For movies, provide auditory descriptions that are synchronized with the original audio. [Priority 1]
3. Provide text version of the auditory description that is collated with the text transcript (captions) of the primary audio track. [Priority 2]

A.5 Ensure that text and graphics are perceivable and understandable when viewed without color.

Otherwise, if color is used to convey information, **users who cannot differentiate between certain colors** and **users with devices that have non-color or non-visual displays** will not receive the information.
When foreground and background colors are too close to the same hue, they may not provide sufficient contrast when viewed using **monochrome displays** or by **people with different types of color deficiencies**.

Checkpoints:

1. Don't use color to convey information unless the information is also clear from the markup and/or text. [Priority 1]
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]

A.6 Indicate structure with structural elements, and control presentation with presentation elements and style sheets.

When structural elements and attributes are used to create presentation effects, **user agents that allow users to navigate through the structure** will be unable to do so properly. Such practices also make it difficult to render the page on other media and devices. For instance, in HTML, don't use H1 to create large, bold face text unless that text is actually a top-level heading.

Checkpoints:

1. Nest headings properly (e.g., in HTML, H1 - H6). [Priority 2]
2. Encode list structure and list items properly (e.g., in HTML: UL, OL, DL, LI). [Priority 2]
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]
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) [p. 8] . Until then, simple tables (to control layout) and bitmap text with alt-text (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) [p. 11] . [Priority 2]
5. Where it's possible to mark up content (for example mathematical equations) instead of using images, use a markup language (such as MathML). [Priority 2]
6. Use relative sizing and positioning (e.g., percent values) rather than absolute (e.g., pixel or point values). [Priority 2]

A.7 Provide supplemental information needed to pronounce or interpret abbreviated or foreign text.

Unless changes between multiple languages on the same page are identified, and expansions for abbreviations and acronyms are provided, they may be indecipherable when **spoken** or **brailled**. This also means that the predominant language on the page must be identified. Supplemental information also helps search engines find key words and identify documents in a desired language as well as improving readability for

everyone, in particular people with reading and language disabilities.

Checkpoints:

1. Clearly identify changes in the language of text (e.g., the HTML "lang" attribute, configure your server to take advantage of content negotiation mechanisms to allow browsers to retrieve files of the preferred language automatically). [Priority 2]
2. Specify the expansion of abbreviations and acronyms (e.g., with the "title" attribute of the HTML ABBR or ACRONYM elements). [Priority 2]

A.8 Ensure that tables have necessary markup to be properly restructured or presented by accessible browsers and other user agents.

Many user agents restructure tables to present them. Without appropriate markup, the tables will not make sense when restructured. Tables also present special problems to **users of screen readers [p. 15]** .

These guidelines benefit users that are **accessing the table through auditory means (e.g., an Automobile PC which operates by speech input and output)** or **viewing only a portion of the page at a time (e.g., users with blindness or low vision using speech or a braille display [p. 15] , or other users of devices with small displays, etc.).**

Checkpoints:

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]
2. For data tables, identify headers for rows and columns (e.g., the HTML TD and TH elements). [Priority 1]
3. For data tables that have more than one row and/or more than one column of header cells, use markup to associate data cells and header cells (e.g., in HTML, THEAD, TFOOT, TBODY, COLGROUP, the "axis", "scope", and "headers" attributes, etc.). [Priority 1]
4. Provide summaries for tables (e.g., via the "summary" attribute on HTML TABLE elements). [Priority 3]
5. Provide abbreviations for header labels (e.g., in HTML, the "abbr" attribute on TH). [Priority 3]

Note. See also A.13.5 [p. 11] and A.6.4 [p. 7] .

A.9 Ensure that pages using newer technologies will transform gracefully into an accessible form if the technology is not supported or is turned off.

The ever-changing landscape of technologies available on the Internet is influenced by W3C Recommendations (such as HTML) and proprietary formats (such as ShockWave, and PDF). However, with each new release of Recommendations, browsers, or plug-ins, new features are added that may not be completely backwards compatible [p. 14] . For

example, HTML 4.0 added the ability to attach style sheets to a page and to embed scripts and applets into a page. **Older browsers** ignore new features and some **users configure their browser not to make use of new features**. These users often see nothing more than a blank page or an unusable page when new features do not transform gracefully.

Checkpoints:

1. Provide a fallback page for pages with dynamic content (HTML examples: NOFRAMES at the end of each frameset, NOSCRIPT for every script, server-side scripts instead of client-side). [Priority 2]
2. Ensure that descriptions of dynamic content are updated with changes in content (e.g., in HTML, [Priority 1]
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]
4. For pages that use style sheets or presentational markup, ensure that the contents of each page are ordered and structured so that they may read properly even when the style sheet or presentational markup is overridden by the user. [Priority 1]
5. For applets and programmatic objects, at a minimum, follow the techniques for alternative text [p. 5] and long descriptions, [p. 6] where needed.
6. 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]

(See also A.14.3) [p. 11]

A.10 Ensure that moving, blinking, scrolling, or auto-updating objects or pages may be paused or frozen.

This is particularly important for objects that contain text but *does not apply to instant redirection [p. 15]* . Some people with **cognitive limitations** or **visual disabilities** are unable to read moving text quickly enough or at all. Movement can also cause such a distraction that the rest of the page becomes unreadable for people with **cognitive disabilities. Screen readers [p. 15]** are unable to read moving text. People with **physical disabilities** might not be able to move quickly or accurately enough to interact with moving objects. People with **photosensitive epilepsy** can have seizures triggered by flickering or flashing in the 4 to 59 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).

Checkpoints:

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]
2. Avoid any blinking or updating of the screen that causes flicker. [Priority 1]
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) [p. 10] [Priority 2]

Note 1. The BLINK and MARQUEE elements, commonly used in HTML pages, are not defined in any W3C HTML specification and should not be used. (See also A.14) [p. 11]

Note 2. (See also A.4) [p. 6]

A.11 Ensure that embedded custom user interfaces are directly accessible.

The accessibility of objects with their own interface is independent of the accessibility of the user agent. Accessibility must therefore be built into the objects or an alternative must be provided (See also A.12) [p. 10] See the User Agent Accessibility Guidelines for details.

Checkpoint:

1. Where possible, make programmatic elements, such as scripts and applets, directly accessible. (See also A.9) [p. 8] . [Priority 1] if information or functionality is important [p. 15] , and not presented elsewhere, otherwise [Priority 2].

A.12 Use features that enable activation of page elements via input devices other than a pointing device (e.g., via keyboard, voice, etc.).

Someone who is using the page **without sight, with voice input, or with a keyboard** (or input device other than a pointing device, e.g., a mouse or braille display [p. 15]) will have a difficult time navigating a page if operation requires a pointing device. If a page is usable via a keyboard, it is more likely that it should also be operable via speech input, or a command line interface. ***Access to image maps is impossible for these users if alternatives are not provided.***

Checkpoints:

1. For image maps, provide alternative text for links. (See also A.1) [p. 5] [Priority 1]
2. If possible, ensure that all elements that have their own interface are keyboard operable. (See also A.12) [p. 10] [Priority 2]
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]
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]

A.13 Use interim accessibility solutions so that assistive technologies and older browsers will operate correctly.

Older browsers are unable to "Tab" to edit boxes, text areas and lists of consecutive links, making it difficult to impossible for users to access them. **Users not operating in a graphical environment** are disoriented by being transferred to a new window without warning.

Checkpoints until most users are able to secure newer technologies that address these issues:

1. Do not use pop-up windows, new windows, or change active window unless the user is aware that this is happening. [Priority 2]
2. Include default, place-holding characters in edit boxes and text areas (e.g., TEXTAREA and INPUT in HTML). [Priority 3]
3. Include non-link, printable characters (surrounded by spaces) between links that occur consecutively. [Priority 3]
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]
5. Until user agents and screen readers [p. 15] 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]

A.14 Use W3C technologies and do so according to specification. Where it is not possible to use a W3C technology, or doing so results in material that does not transform gracefully, you must provide an alternative version of the content that is accessible.

Many non-HTML technologies (e.g., PDF, Shockwave, and other non-W3C data formats) used to encode information require either plug-ins or stand-alone applications that often create pages that cannot be viewed or navigated using standard Web access tools. Also, W3C technologies may be used in ways that do not transform gracefully (e.g., because the visual components are too complex, or because assistive technologies or user agents (browsers) are lacking a specific feature). By avoiding non-standard features (elements, attributes, properties, etc. only supported by a specific browser type) and ensuring that all technologies transform gracefully, your pages will be accessible to more people using a wider variety of hardware and software.

Note. Converting documents (from PDF, PostScript, RTF, etc.) to W3C markup languages (HTML, XML) does not always create an accessible document. Individually test each page for readability after the translation process. If a page does not automatically translate, revise the page until its original representation converts appropriately or prepare and post an HTML or plain text equivalent.

Checkpoints:

1. If W3C technologies are used (e.g. MathML, SMIL, HTML, XML, etc.), use the latest W3C specification whenever possible. [Priority 2]
2. If W3C technologies are used, avoid deprecated language features whenever possible. [Priority 2]
3. If, after all of your best efforts [p. 12] , you can not avoid using a non-W3C technology or any W3C technology in an accessible way then you **must** provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information, and is updated as often as the inaccessible (original) page. [Priority 1]
4. Indicate what type of resource you are linking to , especially when linking to resources that are not W3C technologies, 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]
5. If a resource is served in various formats or languages, use content negotiation to

determine the format or language preferred by the user. Also allow the user to switch from one version to another. [Priority 3]

Note. Alternative pages should be used sparingly since authors tend to neglect updating the alternative page as often as the original page. An out-of-date page may be as frustrating as one that is inaccessible since, in both cases, the information presented on the original page is not available. Automatically generating alternative pages can make it easier to keep up with updates. However, authors must be careful to ensure that generated pages always make sense and that users be able to navigate a site by following links on primary pages, alternative pages, or both.. Before resorting to an alternative page you may want to reconsider the design of the original page. A page that is too elaborate may deter a larger audience than you might expect.

B. Orientation, Navigation, and Comprehension

Maximize usability by providing context and orientation information and simplifying presentations of information.

To provide context and orientation information means that additional information is provided to help users gain an understanding of the "big picture" presented by a page, table, frame, or form. Oftentimes users are limited to viewing only a portion of a page, either because they are accessing the page one word at a time (speech synthesis or braille display [p. 15]), or one section at a time (small display, or a magnified display).

B.1 Provide context and orientation information for complex pages or elements.

Complex relationships between elements on a page may also be difficult for people with **cognitive disabilities** and people with **visual disabilities** to interpret. Grouping and providing contextual information about the relationships between elements can be useful for **all users**

Checkpoints:

1. Title each frame so that users can keep track of frames by title (e.g., via the "title" attribute on HTML FRAME elements). [Priority 1]
2. Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone. (e.g., in HTML, use "longdesc," or a d-link). [Priority 2]
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.
4. Associate labels explicitly with their controls (e.g., in HTML use LABEL and its "for" attribute). [Priority 2]
5. Divide long lists of choices into groups (e.g., with the HTML OPTGROUP element). [Priority 2]

B.2 Provide mechanisms that facilitate navigation within your site.

Through good design, increase the chance that a person can easily find what they are looking for and can easily navigate throughout the site. A clear navigation structure will not only benefit people with **cognitive disabilities**, but **everyone** that visits your site.

To decrease the amount of sifting readers perform to find important [p. 15] information, place distinguishing information at the beginning of headings, paragraphs, lists, etc. This is commonly referred to as "front-loading" and is especially helpful for **people accessing information serially**.

Checkpoints:

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]
2. Use a clear, consistent navigation structure. [Priority 3]
3. Offer navigation bars for easy access to the navigation structure. [Priority 3]
4. Offer a site map. [Priority 3]
5. Provide a description of the general layout of the site, the access features used, and how to use them. [Priority 3]
6. Offer different types of searches for different skill levels and preferences. [Priority 3]
7. Place distinguishing information at the beginning of headings, paragraphs, lists, etc. [Priority 3]
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]
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]
10. Provide a link at the beginning of a group of related links to bypass the group. [Priority 3]

B.3 Use language and formats that facilitate comprehension of information.

Consistent page layouts, common icons, and easy to understand language will benefit everyone in general, but especially people with **cognitive disabilities**, **people whose native language is not the same as yours**, or **people who have difficulty reading**. However, ensure that images have alternative text for people who are **blind**, have **low vision**, or **any user who cannot or has chosen not to view graphics** (See also A.1) [p. 5]

Checkpoints:

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

3. Create a consistent style of presentation between pages. [Priority 3]

C. Appendix - Testing

C.1 Validate your pages and assess the accessibility with automated tools, manual tests, and other services.

It is important to test your site with various types of browsers, older versions of current browsers, and services that emulate browsers. Testing your site with a variety of browsers and other services will allow you to gain firsthand experience of some of the issues people deal with. Adjustments to your design based on the results of tests will increase the likelihood that your site will be usable by a wide range of people and technologies.

1. Use an automated accessibility and browser validation tool.
2. Validate your HTML.
3. Validate your CSS.
4. Use a text-only browser or emulator.
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. It may also be helpful to test a site with a self-voicing browser, a screen reader, magnification software, a small display, etc.
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.

D. Appendix - Definitions

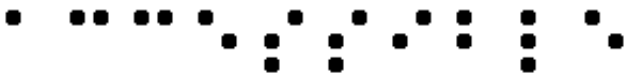
Applet
: A program inserted into a Web page.

ASCII art
: ASCII art refers to text characters and symbols that are combined to create an image. For example ";-)" is the smiley emoticon and the following drawing represents a cow.

```
           (__)
           (oo)
     /-------\/
    / |     ||
   *  ||----||
      ^^    ^^
```

Backwards compatible
: Something that has been designed to work with earlier versions of a language, program, etc.

Braille
: Braille uses six raised dots in different patterns to represent letters and numbers to be read by people who are blind with their fingertips.


The word "Accessible" in braille.

Braille display
: A braille display, commonly referred to as a "dynamic braille display," raises or lowers dot patterns on command from an electronic device, usually a computer. The result is a line of braille that can change from moment to moment. Dynamic braille displays range in size from one cell (six or eight dots) to an eighty cell line. Displays with twelve to twenty cells per line are most common, due to cost.

Collate
: Webster's dictionary defines collate as, "To gather and place in order, as the sheets of a book for binding." When referring to transcripts, the text version of the descriptions and the text transcript (captions) of the primary audio track are combined into a single document to read like a script of the movie. In other words, the two documents are not combined but flows as a single document.

Content developers
: Page authors and site designers.

Image
: A graphical presentation.

Image map
: An image that has been divided into regions; clicking in a region causes an action to occur.

Important
: Something is important if understanding it in detail is necessary for the overall understanding of a document.

Instant redirection
: A page is loaded but immediately replaced by another due to meta information in the transient document.

Screen magnifier
: A software program that magnifies a portion of the screen, so that it can be more easily viewed. Used primarily by individuals with low vision.

Screen reader
: A software program that reads the contents of the screen aloud to a user. Used primarily by individuals who are blind, screen readers can usually only read text that is printed, not painted, to the screen.

E. 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

We wish to thank the following people who have dedicated their time and valuable comments to shaping these guidelines:

Harvey Bingham, Kevin Carey, Chetz Colwell, Neal Ewers, Geoff Freed, Al Gilman, 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, and Jason White

The original draft of this document is based on "The Unified Web Site Accessibility Guidelines" ([UWSAG] [p. 17]) compiled by the Trace R & D Center at the University of Wisconsin. That document includes a list of additional contributors.

F. 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.

[MATHML]
"Mathematical Markup Language", P. Ion and R. Miner, eds. The MathML 1.0 Recommendation is available at:
http://www.w3.org/TR/REC-MathML/.

[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/

[WAI-AUTOOLS]
"Authoring Tool Accessibility Guidelines", J. Treviranus, J. Richards, I. Jacobs, C. McCathieNevile, eds. These guidelines for designing accessible authoring tools are available at:
http://www.w3.org/TR/WD-WAI-AUTOOLS/

[WAI-USERAGENT]
"User Agent Accessibility Guidelines", J. Gunderson and I. Jacobs, eds. These guidelines for designing accessible user agents are available at:
http://www.w3.org/TR/WD-WAI-USERAGENT/

[UWSAG]
"The Unified Web Site Accessibility Guidelines", G. Vanderheiden, W. Chisholm, eds. This document is available at http://www.tracecenter.org/docs/html_guidelines/version8.htm The Unified Web Site Guidelines were compiled by the Trace R & D Center at the University of Wisconsin under funding from the National Institute on Disability and Rehabilitation Research (NIDRR), U.S. Dept. of Education.