W3C

Web Content Accessibility Guidelines

W3C Working Draft 26-Feb-1999

This version:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226
Latest version:
http://www.w3.org/TR/WD-WAI-PAGEAUTH
Previous version:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990217
Related Documents:
Techniques for Web Content Accessibility Guidelines
List of Checkpoints for the 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. 21] should follow in order to make their pages more accessible to people with disabilities. Following these guidelines will also make pages more useful to people using a broad range of devices (desktop browsers, voice browsers, mobile phones, automobile-based PC's, etc.) and to search engines. Tools that create Web content (HTML editors, document conversion tools, tools that generate Web content from databases) should generate content that is consistent with these guidelines.

The guidelines in this document explain to authors, including those providing multimedia content, how to ensure that the content and functions they provide are available to all users. These guidelines do not discourage authors from using images, video, etc. Rather, they encourage authors to make them accessible to the widest possible audience.

This document is part of a series of accessibility guidelines published by the W3C Web Accessibility Initiative. The series also includes User Agent Accessibility Guidelines ([WAI-USERAGENT] [p. 23]) and Authoring Tool Accessibility Guidelines ([WAI-AUTOOLS] [p. 23]).

Status of this document

This is a W3C Working Draft for review by W3C members and other interested parties. With the publication of this draft, the Web Content Accessibility Guidelines enters "last call." The last call period will end on March 19, 1999. Please note that only the current document (not the accompanying Techniques Document.) is entering last call. Comments about that document are still encouraged.

This 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 Web Accessibility Initiative, and is intended as a draft of a Proposed Recommendation for authoring accessible Web pages. The goal of the WAI Page Author Guidelines Working Group is discussed in the Working Group charter.

Available formats

This document is available in the following formats:

HTML:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226/wai-pageauth.html
A plain text file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226/wai-pageauth.txt,
HTML (Guideline, Techniques, List of Checkpoints) as a gzip'ed tar file :
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226/wai-pageauth.tgz,
HTML (Guideline, Techniques, List of Checkpoints) as a zip file (this is a '.zip' file not an '.exe'):
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226/wai-pageauth.zip,
A PostScript file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226/wai-pageauth.ps,
A PDF file:
http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226/wai-pageauth.pdf.

In case of a discrepancy between the various formats of the specification, http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226/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

Abstract 1
Status of this document 2
Introduction 4
How the Guidelines are organized 5
Priorities 5
Conformance 6
A. Web Content Accessibility Guidelines 7
1. Provide text equivalents for visual information. 7
2. Provide descriptions of important visual information. 8
3. Provide text equivalents for audio information. 9
4. Don’t rely on color alone. 10
5. Use markup and style sheets properly. 10
6. Supplement markup to aid interpretation of text. 11
7. Create tables that transform gracefully. 12
8. Ensure that pages featuring new technologies transform gracefully. 12
9. Ensure user control of time-sensitive content changes. . . . 13
10. Ensure direct accessibility of embedded user interfaces. . . . 14
11. Design for device-independence. 15
12. Consider interim solutions. 15
13. Use W3C technologies and guidelines. 16
14. Supply context and orientation information. 17
15. Design clear navigation structures. 18
16. Design for consistency and simplicity. 19
B. Appendix - Testing 19
C. Appendix - Definitions 20
Acknowledgments 22
References 23

See also the List of Checkpoints for the Web Content Accessibility Guidelines.

Introduction

For those unfamiliar with accessibility issues pertaining to Web page design, consider that many users may be using documents in contexts very different from your own:

- They may not be able to see, hear, move, or understand easily or at all.
- They may be in a situation where their eyes, ears, or hands are busy or interfered with (e.g., driving to work, working in a loud environment, etc.)
- They may have difficulty reading or comprehending text.
- They may not have or be able to use a keyboard or mouse.
- They may have an early version of a browser, a different browser entirely, a voice browser, or a different operating system.
- They may have a slow connection, a small screen, a text-only screen, etc.
- They may not speak or understand fluently the language in which the document is written.

Authoring accessible Web pages means being aware of all these different environments, and accounting for them in design choices. Fortunately, accessible design does not generally mean extra work (or page duplication) and it usually benefits the Web community at large. Following the guidelines will generally shorten page download times and make sites easier to manage (e.g., by sharing style sheets).

Consider what happens when an image is inserted into an HTML document with the IMG element. Some users may not be able to see images, others may have support for images turned off (e.g., due to a slow Internet connection) or their browsers may not support images at all. Accessible authoring does not mean avoiding images or video. Multimedia should be used where appropriate (which we explain later on) but also accompanied by accessible alternatives to the visual information. In the case of the IMG, this means providing a short description of the image's function via the "alt" attribute. There are additional ways to make images accessible described in this document and the accompanying Techniques Document. Making images more accessible will benefit a number of "content consumers":

- Users with blindness will understand the function of the image. This is particularly helpful when the image is part of a hyperlink since the link has no other text to suggest the nature of its target.
- User agents can render "alt" text as a tool tip, thus providing additional information to the general populace.
- Search engines will find more information about a page or site.

These guidelines describe how to use the features of HTML, SMIL, CSS, and other languages that promote accessibility. Reading these guidelines will help authors understand:

- What accessibility issues to be aware of.
- Why certain pages are inaccessible.
- How to improve the design of a page or site without heavy additional cost.

Once you've reviewed the guidelines, we encourage you to seek out and correct any accessibility barriers on your Web sites by borrowing solutions from the very detailed Techniques Document.

How the Guidelines are organized

This document refers to three concepts:

Guideline
: A guideline is a general principle of accessible design. A guideline addresses the question "What should I keep in mind while authoring?"

Checkpoint
: A checkpoint is a specific way of satisfying one or more guidelines. While checkpoints describe verifiable actions that may be carried out by the author, implementation details are described elsewhere. A checkpoint answers the question "What steps can I take to improve accessibility?"

Technique
: A technique is an implementation of one or more checkpoints in a given language (e.g., HTML, CSS, MathML, etc.). A technique answers the question "How do I do that in HTML or CSS or MathML?"

This document lists guidelines and the checkpoints that satisfy them. It is accompanied by a List of Checkpoints so that authors can evaluate their documents and Web sites.

The Techniques Document provides in-depth descriptions of the same checkpoints and examples of how to implement the checkpoints using HTML, CSS, and more. Please note that the Techniques Document will be updated more regularly than the current document in order to keep up with changes in technology.

Note. The current document and the Techniques Document refer to some technologies (e.g., features of HTML, CSS, or SMIL) that may not be supported yet in some browsers, multimedia tools, etc. Browser support for language features is discussed in the Techniques Document.

Priorities

Some checkpoints are more important to accessibility than others and so each one has been assigned a priority level. Some checkpoints specify a priority level that may change under some (indicated) conditions.

The priority levels used in this document are defined as follows:

[Priority 1]
This checkpoint **must** be satisfied 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 satisfied by an author, or one or more groups of users will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing Web documents.
[Priority 3]
This checkpoint **may** be satisfied 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.

The checkpoints have been written so that it will be possible to verify when they have been satisfied.

Conformance

To conform to these guidelines, a document or process must satisfy one of the following priority levels:

- **Level P1**: all priority one checkpoints, or
- **Level P12**: all priority one and two checkpoints, or
- **Level P123**: all priority one, two, and three checkpoints.

Every claim of conformance to this document must:

1. identify which level of conformance (P1, P12, or P123) is satisfied, and
2. identify this document with this URI <http://www.w3.org/TR/1999/WD-WAI-PAGEAUTH-19990226>, and
3. identify the document title "Web Content Accessibility Guidelines"

A. Web Content Accessibility Guidelines

By following these guidelines, authors can create pages that **transform gracefully.** Pages that transform gracefully remain accessible despite any of the constraints described in the introduction [p. 4] , including physical, sensory, and cognitive disabilities, work constraints, and technological barriers. Here are some keys to designing pages that transform gracefully:

- Separate content (what is said) from structure (how it is organized) from presentation (how it appears on the screen, when spoken, when rendered by a braille device, etc.).
- Use text or provide a text equivalent. Textual information is generally available to all users and browsing devices.
- Don’t create documents that only work if the user can see and/or hear. For content that is sensory-specific (e.g., audio, video, applets that present visual information), provide equivalent information in forms suited to other senses as well. This does not mean creating an entire auditory version of a site. Screen readers will be able to speak all information on a page as long as it is available in text.
- Don’t create documents that rely on one type of hardware. Pages should be usable by people without mice, with small screens, low resolution screens, black and white screens, no screens, with only voice or text output, etc.

Guidelines 1 to 13 list checkpoints related to the above topics.

Authors should also consider in their design how users will navigate and understand each page and the relation between pages. Providing navigation tools and orientation information in pages will maximize usability. Remember, not all users can make use of visual clues such as image maps, proportional scroll bars, side-by-side frames, or icons that guide sighted users of graphical desktop browsers. Users also lose contextual information when they can only view a portion of a page, either because they are accessing the page one word at a time (speech synthesis or braille display [p. 20]), or one section at a time (small display, or a magnified display). Very large tables, lists, menus, etc. without orientation information may be very disorienting to users.

Guidelines 14 to 16 discuss ways to add context and orientation to documents, for example by grouping form controls, providing navigation bars, offering a site map, and numbering lists and sections.

Guideline 1. Provide text equivalents for visual information.

Provide text equivalents for all images, applets, and image maps.

A text equivalent describes the purpose or function of an image, applet, image map, or other visual information. For example, the text equivalent for a company logo image in a link might be "Return to home page".

If text equivalents are not provided for visual information, people 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.

Checkpoints:

1.1 Provide text equivalents for all images [Priority 1]
For example, in HTML, use the "alt" attribute of the IMG and INPUT elements, or for OBJECT, use "title" or the element's content.
Note. The Techniques Document explains how to write text equivalents specifically for images used as image maps, as spacers, as bullets in lists, as graphical buttons, as links, or to present mathematical equations.
1.2 Provide text equivalents for all applets and other programmatic objects. [Priority 1]
For example, in HTML, for OBJECT use the "title" attribute or the element's content. Also in HTML, for APPLET, use the "alt" attribute or within the content of the element. See also guideline 10.
1.3 Provide a text equivalent for each active region of an image map. [Priority 1]
For example, in HTML, use the "alt" attribute of the AREA element or the MAP element with A elements as content.
1.4 Provide redundant textual links for each active region of an image map. [Priority 2 - if client-side image maps [p. 21] are used, Priority 1 - if server-side image maps are used.]
1.5 Use individual button controls in a form rather than simulating a set of buttons with an image map. [Priority 2]
Note. When a button uses an image, specify a text equivalent for the image.
1.6 Replace ASCII art with an image or describe the ASCII art and offer a means (e.g., a link) to skip over it. [Priority 1 or Priority 2 depending on the importance of the information (e.g., an important [p. 21] chart).]
Note. When replacing ASCII art with an image, specify a text equivalent for the image
Note. If the description of (important) ASCII art is long, provide a description in addition to a text equivalent. See also guideline 2.

Guideline 2. Provide descriptions of important visual information.

Provide descriptions of important information in graphics, scripts, applets, videos, or animations if it is not fully described through text equivalents or in the document's content.

A long description, unlike a text equivalent [p. 7] , describes the visual information presented by an image, script, or applet. Long descriptions make information presented graphically (charts, billboards, diagrams) perceivable to people with **blindness**, some people with **low vision**, and to people who have **chosen not to view graphics, scripts, or applets** or whose **browser does not support scripts**

or applets.

In movies or visual presentations, some visual action, body language, or other visual cues may not be accompanied by enough audio information to understanding what's going on without the video. Unless verbal descriptions of this visual information are provided, **people who cannot see** (or look at) the visual content will not be able to perceive it.

If a visual presentation has an associated auditory presentation (e.g., a movie), the audio version of the descriptions should be **synchronized with the existing auditory presentation**, and a text version of the descriptions should be **collated [p. 20] with the text transcripts (captions)** of the primary audio track. The collated information will make the presentation available to **people who are deaf-blind** and to **people who cannot play or choose not to play movies, animations, etc.**.

Checkpoints:

2.1 Provide a long description of each graphic, script, or applet that conveys important [p. 21] information. [Priority 1]
    For example, in HTML, for IMG use "longdesc", for OBJECT use the element's content, or specify a description link (d-link)
2.2 For short animations such as "animated gifs," provide a text equivalent and a long description if needed. [Priority 1]
    See also guidelines 1 and 2.
2.3 For movies, provide auditory descriptions that are synchronized with the original audio. [Priority 1]
2.4 Provide a text version of the auditory description that is collated with the text transcript (captions) of the primary audio track. [Priority 2]

Guideline 3. Provide text equivalents for audio information.

Provide text transcripts, text descriptions, or captions of auditory events that occur in audio and video.

When a transcript is synchronized with a video presentation it is called a "caption". Captions are used by people who cannot hear the audio track of the video material. Without transcripts and captions, people 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:

3.1 For stand-alone audio files, provide a text transcript of all words (spoken or sung) and all significant sounds. [Priority 1]
3.2 For audio associated with video, synchronize the text transcript with the video. [Priority 1]
    The transcript (of dialog and sounds) should be synchronized (i.e., made into a caption).

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

Guideline 4. Don't rely on color alone.

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

If color alone is used to convey information, **people 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 deficits**.

Checkpoints:

4.1 Ensure that all information conveyed with color is also available without color, for example from context or markup. [Priority 1]
4.2 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 2]

Guideline 5. Use markup and style sheets properly.

Mark up documents with the proper structural elements. Control presentation with style sheets rather than with presentation elements and attributes.

Using markup improperly -- not according to specification -- hinders accessibility. Misusing markup for a presentation effect (e.g., using a table for layout or a header to change the font size) makes it difficult for **users with specialized software** to understand the structure of the page or to navigate through it. Furthermore, presentation effects used alone to convey structure (e.g., constructing what looks like a table of data with an HTML PRE element) make it difficult to render a page intelligibly to other devices. For more information, see the Techniques Document discussion of structure versus presentation.

Checkpoints:

5.1 Nest headings properly. [Priority 2]
For example, in HTML, nest H2 elements within H1 elements, H3 elements within H2 elements, etc.
5.2 Encode list structure and list items properly. [Priority 2]
For example, in HTML, nest OL, UL, and DL lists properly.

5.3 Mark up quotations. Do not use quotation markup for formatting effects such as indentation. [Priority 2]

For example, in HTML, use the Q and BLOCKQUOTE elements to markup short and longer quotations, respectively.

5.4 Use style sheets to control layout and presentation. [Priority 2]

5.5 Where possible, use a markup language to mark up content rather than using images. [Priority 2]

For example, use MathML to mark up mathematical equations, and style sheets to format text and control layout.

Note. If important information is conveyed in many images on the page, provide an alternative accessible page.

See also guidelines 8 and 13.

5.6 Use relative rather than absolute units in markup language attribute values and style sheet property values. [Priority 2]

For example, in CSS, use 'em' or percentages lengths rather than 'pt' or 'cm', which are absolute units.

Guideline 6. Supplement markup to aid interpretation of text.

Provide supplemental information to facilitate pronunciation or interpretation of abbreviated or foreign text.

Changes between multiple languages on the same page and abbreviations can both be indecipherable when **spoken** or **brailled** unless they are identified. Authors should identify the predominant (natural) language of a document's text and indicate when language changes occur. They should also provide expansions of abbreviations and acronyms. This supplemental information also helps search engines find key words and identify documents in a desired language. The information also improves readability for the general populace, including those with reading and language disabilities.

Checkpoints:

6.1 Clearly identify changes in the (natural) language of a document's text. [Priority 1]

For example, in HTML use the "lang" attribute. Server operators should configure their server to take advantage of HTTP content negotiation mechanisms so that clients can automatically retrieve documents of the preferred language.

6.2 Identify the primary (natural) language of a document. [Priority 3]

For example, in HTML, set the "lang" attribute on the HTML element.

6.3 Specify the expansion of abbreviations and acronyms. [Priority 2]

For example, in HTML, use the "title" attribute of the ABBR and ACRONYM elements.

Guideline 7. Create tables that transform gracefully.

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

Tables should be used to structure truly tabular data ("data tables"). They should not be used to lay out pages ("layout tables"). Tables for any use also present special problems to **users of screen readers [p. 21]** .

Many user agents restructure tables to present them and if not marked up properly, the tables will not make sense when rendered. See also guideline 5.

The following checkpoints will directly benefit people who **access a table through auditory means** (e.g., through an Automobile PC that operates by speech input and output) or who **view only a portion of the page at a time** (e.g., users with blindness or low vision using speech or a braille display [p. 20] , or other users of devices with small displays, etc.).

Checkpoints:

7.1 Avoid using tables for layout. [Priority 2]
7.2 If a table is used for layout, do not use any structural markup for the purpose of visual formatting. [Priority 2]
For example, in HTML do not use the TH element to cause the contents of a (non-table header) cell to be displayed centered and in bold. Authors should use attributes or table captions to explain that the table is a layout table.
7.3 For data tables, identify headers for rows and columns. [Priority 1]
For example, in HTML, use TD to identify data cells and TH to identify headers.
7.4 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. [Priority 1]
For example, in HTML, use THEAD, TFOOT, and TBODY to group rows, COL and COLGROUP to group columns, and the "axis", "scope", and "headers" attributes, to describe more complex relationships among data.
7.5 Provide summaries for tables. [Priority 3]
For example, in HTML, use the "summary" attribute of the TABLE element.
7.6 Provide abbreviations for header labels. [Priority 3]
For example, in HTML, use the "abbr" attribute on the TH element.

Note. See also Checkpoints 12.5 and 5.5.

Guideline 8. Ensure that pages featuring new technologies transform gracefully.

Ensure that pages are accessible even when newer technologies are not supported or are turned off.

Although authors should be encouraged to use new technologies that solve problems raised by existing technologies, they should know how to make their pages still work with **older browsers** and **people who choose to turn off features.**

Checkpoints:

8.1 Provide an alternative presentation or page when the primary content is dynamic. [Priority 2]
For example: In HTML, use NOFRAMES at the end of each frameset, NOSCRIPT for every script, and server-side instead of client-side scripts.
8.2 Ensure that descriptions of dynamic content are updated when the dynamic content changes. [Priority 1]
For example, in HTML ensure that the source of a frame is a document.
8.3 For scripts that present important [p. 21] information or functionality, provide an alternative, equivalent presentation or mechanism. [Priority 1]
For example, in HTML use NOSCRIPT or a server-side script.
8.4 For pages that use style sheets or presentation markup [p. 10] , ensure that the contents of each page are ordered and structured. [Priority 1]
This makes it more likely that the document will be understood even when styles are turned off or overridden by the user. For applets and programmatic objects, at a minimum, follow the techniques for text equivalents [p. 8] and long descriptions, [p. 9] where needed.
8.5 For applets and programmatic objects, when possible provide an alternative function or presentation in a format other than an applet. [Priority 2]
For example, for an applet, provide a video equivalent. For an animation, provide a snapshot of the animation.

See also checkpoint 13.5.

Guideline 9. Ensure user control of time-sensitive content changes.

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

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. 21]** 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).

This guideline *does not apply when the server performs the instant redirection. [p. 21]*

Checkpoints:

9.1 Until user agents provide the ability to stop the refresh, do not use periodically auto-refreshing pages [Priority 1]
For example, in HTTP, auto-refreshing is done with "HTTP-EQUIV=refresh".
9.2 Until user agents provide the ability to stop the refresh, do not use one-time refresh in place of server redirect. [Priority 2]
9.3 Avoid any blinking or updating of the screen that causes flicker. [Priority 1]
9.4 Avoid movement in pages, 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. [Priority 2]
Note. Style sheets and scripts can be turned off or overridden.
See also guideline 10.

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 guideline 13.

Guideline 10. Ensure direct accessibility of embedded user interfaces.

Ensure that the user interface follows principles of accessible design: device-independent access to functionality, keyboard operability, self-voicing, etc.

When an embedded object has its "own interface", the interface -- like the interface to the browser itself -- must be accessible. If the interface of the embedded object cannot be made accessible, an alternative accessible solution must be provided.

For information about accessible interfaces, please consult the User Agent Accessibility Guidelines ([WAI-USERAGENT] [p. 23]), the Authoring Tool Accessibility Guidelines ([WAI-AUTOOL] [p. 23]), and the discussion of applets and other objects in the Techniques Document.

Checkpoint:

10.1 Where possible, make programmatic elements, such as scripts and applets, directly accessible or compatible with assistive technologies [Priority 1 if information or functionality is important [p. 21] and not presented elsewhere, otherwise Priority 2.]
See also guideline 8.

Guideline 11. Design for device-independence.

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

Not everyone uses a mouse. Interaction with a document should not depend on a particular input device. If, for example, a form control can only be activated with a mouse or other pointing device, someone who is using the page **without sight, with voice input, or with a keyboard** or who is using an input device other than a mouse (e.g., a braille display [p. 20]) will not be able to use the form.

Note. Providing text equivalents for image maps or images used as links makes it possible for users to interact with them without a pointing device. See also guideline 1.

Generally, pages that allow keyboard interaction are also accessible through speech input or a command line interface.

Checkpoints:

11.1 If possible, ensure that all elements that have their own interface are keyboard operable. [Priority 2]
See also guideline 11.
11.2 Create a logical tab order through links, form controls, and objects. [Priority 3]
For example, in HTML, use the "tabindex" attribute or ensure a logical page design.
11.3 Provide keyboard shortcuts to links, including those in client-side image maps, [p. 21] form controls, and groups of form controls. [Priority 3]
For example, in HTML, use the "accesskey" attribute.
11.4 For scripts, specify logical event handlers rather than device-dependent event handlers. [Priority 2]
For example, in HTML use "onfocus", "onblur", and "onselect".

Guideline 12. Consider interim solutions.

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

For example, **older browsers** do not allow users to navigate to empty edit boxes. Older screen readers read lists of consecutive links as one link. These activate elements are therefore difficult or impossible to access. Also, changing the current window or popping up new windows can be very disorienting to **users who have available, but aren't using, the graphical features of the desktop environment**.

Note. The following checkpoints apply until most users are able to secure newer technologies that address these issues.

Checkpoints:

12.1 Do not cause pop-ups or other windows to appear and do not change the current window without informing the user. [Priority 2]

For example, in HTML, avoid using a frame whose target is a new window.

12.2 Include default, place-holding characters in edit boxes and text areas. [Priority 3]

For example, in HTML, do this for TEXTAREA and INPUT.

12.3 Include non-link, printable characters (surrounded by spaces) between links that occur consecutively. [Priority 3]

12.4 For all form controls with implicitly associated labels, ensure that the label is properly positioned. [Priority 2]

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).

12.5 Provide a linear text alternative (on the current page or some other) for *all* tables that lay out text in parallel, word-wrapped columns. [Priority 2]

Note. This checkpoint applies to user agents and screen readers [p. 21] that are unable to handle blocks of text presented side-by-side.

Guideline 13. Use W3C technologies and guidelines.

Use W3C technologies (according to specification) and follow accessibility guidelines. Where it is not possible to use a W3C technology, or doing so results in material that does not transform gracefully, 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 which often create pages that cannot be viewed or navigated using standard Web access or screen reading tools. Also, W3C technologies may be used in ways that do not transform gracefully (e.g., the visual components are too complex, assistive technologies or browsers lack a specific feature, etc.). Avoiding non-standard features (elements, attributes, properties, etc. only supported by a specific browser type) and ensuring that all technologies transform gracefully will make pages more 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. Therefore, test each page for readability after the translation process. If a page does not readily translate, either revise the page until its original representation converts appropriately or provide an HTML or plain text equivalent.

Checkpoints:

13.1 If W3C technologies are used (e.g. HTML, XML, SMIL, MathML, etc.), use the latest W3C specification whenever possible. [Priority 2]
13.2 If W3C technologies are used, avoid deprecated language features whenever possible. [Priority 2]
13.3 Indicate the content type of the link target, especially when linking to resources that are not W3C technologies. [Priority 3]
For example, to link to a PDF file from an HTML document, set the "type" attribute to "application/pdf" on the A element.
13.4 *For servers.* If a resource is served in various formats or languages, use content negotiation to determine the format or language preferred by the user. [Priority 3]
Also allow the user to switch from one version of a document to another.
13.5 If, after best efforts [p. 17] , you can not avoid using a non-W3C technology or any W3C technology in an accessible way, 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]

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 may lead to more frequent updates, but authors must still be careful to ensure that generated pages always make sense and that users are able to navigate a site by following links on primary pages, alternative pages, or both. Before resorting to an alternative page, reconsider the design of the original page; simplifying it is likely to make it more effective for all users.

Guideline 14. Supply context and orientation information.

Supply context and orientation information to help users understand complex pages or elements

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

Checkpoints:

14.1 Title each frame so that users can keep track of frames by title. [Priority 1]
For example, in HTML use the "title" attribute on FRAME elements.
14.2 Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone. [Priority 2]
For example, in HTML, use "longdesc," or a description link (d-link).

14.3 Group form controls. [Priority 2 - for radio buttons and checkboxes, Priority 3 - for other controls.]
For example, in HTML use the FIELDSET and LEGEND elements.
14.4 Associate labels explicitly with their controls. [Priority 2]
For example, in HTML use LABEL and its "for" attribute.
14.5 Divide long lists of choices into manageable groups. [Priority 2]
For example, in HTML use the OPTGROUP element.

Guideline 15. Design clear navigation structures.

Use clear navigation structures, navigation bars, etc. to increase the likelihood that a person will find what they are looking for at a site.

Navigation structures benefit people with **cognitive disabilities**. Furthermore, **everyone** who visits the site will appreciate them.

To make it easier for people to find important [p. 21] 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 with serial presentation techniques such as speech**.

Checkpoints:

15.1 Wherever possible, make link phrases as terse as possible yet as meaningful as possible when read on their own or in succession. [Priority 2]
Avoid general phrases, such as "click here" (which is device-dependent in addition to saying nothing about what is to be found at the end of the link).
15.2 Use metadata to add semantic information to pages and sites. [Priority 2]
For example, use RDF ([RDF] [p. 23]) to indicate the author, the type of content, etc.
15.3 Use a clear, consistent navigation structure. [Priority 3]
15.4 Offer navigation bars to highlight and give access to the navigation structure. [Priority 3]
15.5 Offer a site map or table of contents that makes the structure of a Web site apparent and facilitates navigation. [Priority 3]
15.6 Provide a description of the general layout of the site, the access features used, and how to use them. [Priority 3]
15.7 Offer different types of searches for different skill levels and preferences. [Priority 3]
15.8 Place distinguishing information at the beginning of headings, paragraphs, lists, etc. [Priority 3]
15.9 Facilitate off-line browsing by creating a single downloadable file for documents that exist as a series of separate pages. [Priority 3]
For example, in HTML use the LINK element. Or create an archive of the different pages (e.g., with zip, gzip, stuffit, etc.).

15.10 Group related links, identify the group (for user agents), and provide a mechanism to bypass the group (until user agents provide this mechanism). [Priority 3]

For example, when creating a navigation bar composed of links in HTML use "title" on FRAME, DIV, SPAN, etc. Use class="nav" to identify the group. Use "tabindex=1" on an anchor after the group so users may quickly navigate to it.

Guideline 16. Design for consistency and simplicity.

Use consistency and simplicity to promote comprehension.

Consistent page layout, recognizable icons, and easy to understand language benefit all who visit a site. In particular, they help people with **cognitive disabilities** or **who have difficulty reading**. However, ensure that images have text equivalents for people who are **blind**, have **low vision**, or for **any user who cannot or has chosen not to view graphics**. See also guideline 1.

Using direct, simple language is an integral part of effective communication. Conditions ranging from dyslexia to deafness can make access to written information difficult to impossible for some users. This consideration also applies to the many people **whose first language differs from your own**.

Checkpoints:

16.1 Use language that is as simple as possible, while appropriate for the site's content. [Priority 1]
16.2 Use icons or graphics (with a text equivalent) where they facilitate comprehension of the page. [Priority 3]
16.3 Create a consistent style of presentation between pages. [Priority 3]

B. Appendix - Testing

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

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

1. Use an automated accessibility tool and browser validation tool.
2. Validate all HTML.
3. Validate all 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.

C. 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:

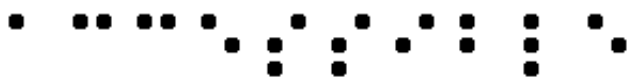
```
          (__)
          (oo)
    /-------\/
   / |     ||
  *  ||----||
```

Backward 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 follows:



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 the most common.

Collate

When referring to transcripts, collating means combining the text version of the descriptions and the text transcript (captions) of the primary audio track into a single document to read like a script of the movie. In other words, the two

documents are not combined but flow as a single document.

Content developer

Page authors and site designers.

Description link (D-link)

A description link is a link from a document to a long description of an image, applet, or other visual information.

DHTML (Dynamic HTML)

The interaction of HTML, style sheets, the Document Object Model [DOM1] [p. 23] and scripting is commonly referred to as "Dynamic HTML" or DHTML. However, there is no W3C specification that formally defines DHTML. Most guidelines may be applicable to DHTML-type of applications, however the following guidelines focus on issues related to scripting, and style sheets: 1, 2, 8, and 9.

Image

A graphical presentation.

Image map

An image that has been divided into regions with associated actions. Clicking on an active region causes an action to occur.

When a user clicks on an active region of a client-side image map, the user agent calculates in which region the click occurred and follows the link associated with that region. Clicking on an active region of a server-side image map causes the coordinates of the click to be sent to a server, which then performs some action.

Authors can make client-side image maps accessible by providing device-independent access to the same links associated with the image map's regions. Client-side image maps allow the user agent to provide immediate feedback as to whether or not the user's pointer is over an active region.

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.

Navigation Bar

A navigation bar is a collection of links to the most important parts of a document or site.

Personal Digital Assistant (PDA)

A PDA is a class of small, portable computing devices. usually used for tracking personal data such as calendars, contacts, and electronic mail. It is generally a handheld device with a small screen that allows input from various sources.

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.

Site map

A site map is similar to a navigation bar [p. 21] but provides more detail about the organization of a site and links to important pages.

Acknowledgments

WAI Markup Guidelines Working Group Co-Chairs:

Chuck Letourneau, Starling Access Services

Gregg Vanderheiden, Trace Research and Development

W3C Team contacts:

Judy Brewer and Daniel Dardailler

We wish to thank the following people who have contributed 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. 24]) compiled by the Trace R & D Center at the University of Wisconsin. That document includes a list of additional contributors.

References

[CSS1]
"CSS, level 1 Recommendation", B. Bos, H. Wium Lie, eds. The CSS1 Recommendation is available at:
http://www.w3.org/TR/1999/REC-CSS1-19990111.

[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/1998/REC-CSS2-19980512.

[DOM1]
"Document Object Model (DOM) Level 1 Specification", V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, A. Le Hors, G. Nicol, J. Robie, R. Sutor, C. Wilson, and L. Wood, eds. The DOM Level 1 Recommendation is available at:
http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001.

[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/1998/REC-html40-19980424.

[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/1998/REC-MathML-19980407.

[RDF]
"Resource Description Framework (RDF) Model and Syntax Specification", O. Lassila, R. Swick, eds. The RDF Recommendation is available at:
http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.

[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/1998/REC-smil-19980615

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

[WAI-USERAGENT]
"User Agent Accessibility Guidelines", J. Gunderson and I. Jacobs, eds. The latest Working Draft of these guidelines for designing accessible user agents is available at:
http://www.w3.org/TR/WD-WAI-USERAGENT/

[UWSAG]
"The Unified Web Site Accessibility Guidelines", G. Vanderheiden, W. Chisholm, eds. 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. This document is available at: http://www.tracecenter.org/docs/html_guidelines/version8.htm

[XML]
"Extensible Markup Language (XML) 1.0.", T. Bray, J. Paoli, C.M. Sperberg-McQueen, eds. The XML 1.0 Recommendation is available at: http://www.w3.org/TR/1998/REC-xml-19980210