Techniques for Web Content Accessibility Guidelines 1.0

W3C Note 20 September 2000

This version:
http://www.w3.org/TR/2000/NOTE-WCAG10-TECHS-20000920/
(plain text, PostScript, PDF, gzip tar file of HTML, zip archive of HTML)
Latest version:
http://www.w3.org/TR/WCAG10-TECHS/
Previous version:
http://www.w3.org/TR/1999/WAI-WEBCONTENT-TECHS-19990505/
Editors:
Wendy Chisholm, W3C;
Gregg Vanderheiden, Trace R & D Center, University of Wisconsin -- Madison;
Ian Jacobs, W3C

Copyright ©1999 - 2000 W3C® (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

Abstract

This document is the gateway to a series of related documents that provide techniques for satisfying the requirements defined in "Web Content Accessibility Guidelines 1.0" *[WCAG10] [p. 22]* . This series includes:

1. *"Techniques for Web Content Accessibility Guidelines 1.0"*, the current document, which is the gateway to the other documents.
2. *"Core Techniques for Web Content Accessibility Guidelines 1.0"* (*[WCAG10-CORE-TECHNIQUES] [p. 22]*), which discusses the accessibility themes and general techniques that apply across technologies.
3. *"HTML Techniques for Web Content Accessibility Guidelines 1.0"* (*[WCAG10-HTML-TECHNIQUES] [p. 22]*), which provides examples and strategies for authoring accessible Hypertext Markup Language (HTML) content.
4. *"CSS Techniques for Web Content Accessibility Guidelines 1.0"* (*[WCAG10-CSS-TECHNIQUES] [p. 22]*), which provides examples and strategies to help authors write Cascading Style Sheets (CSS) as part of accessible content design.

Status of this document

The 20 September 2000 version of this document is a Note in a series of Notes produced and endorsed by the Web Content Accessibility Guidelines Working Group. This Note has not been reviewed or endorsed by W3C Members. The series of documents supersedes the 5 May 1999 W3C Note "Techniques for Web Content Accessibility Guidelines 1.0". That single document has been divided into technology-specific documents that may evolve independently. Smaller technology-specific documents also allow authors to focus on a particular technology.

While the "Web Content Accessibility Guidelines 1.0" Recommendation *[WCAG10] [p. 22]* is a stable document, this series of companion documents is expected to evolve as technologies change and content developers discover more effective techniques for designing accessible Web sites and pages. In the near future, the Working Group intends to incorporate techniques for the Synchronized Multimedia Integration Language (SMIL) *[SMIL] [p. 21]* described in "Accessibility Features of SMIL" (*[SMIL-ACCESS] [p. 21]*) and techniques for Scalable Vector Graphics (SVG) *[SVG] [p. 21]* described in "Accessibility Features of SVG" (*[SVG-ACCESS] [p. 21]*). The Working Group also intends to incorporate techniques for non-W3C technologies such as ECMAScript, PDF and Flash.

The history of changes to the series of documents as well as the list of open and closed issues are available. Readers are encouraged to comment on the document and propose resolutions to current issues. Please send detailed comments on this document to the Working Group at w3c-wai-gl@w3.org; public archives are available.

The English version of this document is the only normative version. However, for translations in other languages see "http://www.w3.org/WAI/GL/WAI-WEBCONTENT-TRANSLATIONS".

The list of known errors in this document is available at "Errata in Web Content Accessibility Guidelines." Please report errors in this document to wai-wcag-editor@w3.org.

The Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C) makes available a variety of resources on Web accessibility. WAI Accessibility Guidelines are produced as part of the WAI Technical Activity. The goals of the WCAG WG are described in the charter.

A list of current W3C Recommendations and other technical documents is available.

Table of Contents

Abstract 1
Status of this document 2
1 How this Document is Organized 4
1.1 Priorities 4
2 Techniques for Web Content Accessibility Guidelines 6
1. Provide equivalent alternatives to auditory and visual content. . . .6
2. Don’t rely on color alone. 7
3. Use markup and style sheets and do so properly. 7
4. Clarify natural language usage 8
5. Create tables that transform gracefully. 8
6. Ensure that pages featuring new technologies transform gracefully. .9
7. Ensure user control of time-sensitive content changes. . . . 10
8. Ensure direct accessibility of embedded user interfaces. . . . 11
9. Design for device-independence. 11
10. Use interim solutions. 12
11. Use W3C technologies and guidelines. 12
12. Provide context and orientation information. 13
13. Provide clear navigation mechanisms. 13
14. Ensure that documents are clear and simple. 14
3 Glossary 15
4 References 21
5 Resources 22
5.1 Other Guidelines 22
5.2 User agents and other tools 22
6 Acknowledgments 23

1 How this Document is Organized

Section 2 of this document reproduces the guidelines and checkpoints of the "Web Content Accessibility Guidelines 1.0" *[WCAG10] [p. 22]* . Each guideline includes:

- The guideline number.
- The statement of the guideline.
- A list of checkpoint definitions. Checkpoints are ordered according to their priority [p. 4] , e.g., Priority 1 before Priority 2.

Each checkpoint definition includes:

- The checkpoint number.
- The statement of the checkpoint.
- The priority of the checkpoint.
- A link back to the definition of the checkpoint in "Web Content Accessibility Guidelines 1.0" *[WCAG10] [p. 22]* . Definitions may also include informative notes, examples, cross references, and commentary to help readers understand the scope of the checkpoint.

Each checkpoint is followed by one or more links to techniques in the following documents:

- *"Core Techniques for Web Content Accessibility Guidelines 1.0"* (*[WCAG10-CORE-TECHNIQUES] [p. 22]*), which discusses the accessibility themes and general techniques that apply across technologies.
- *"HTML Techniques for Web Content Accessibility Guidelines 1.0"* (*[WCAG10-HTML-TECHNIQUES] [p. 22]*), which provides examples and strategies for authoring accessible Hypertext Markup Language (HTML) content.
- *"CSS Techniques for Web Content Accessibility Guidelines 1.0"* (*[WCAG10-CSS-TECHNIQUES] [p. 22]*), which provides examples and strategies to help authors write Cascading Style Sheets (CSS) as part of accessible content design.

1.1 Priorities

Each checkpoint has a priority level assigned by the Working Group based on the checkpoint's impact on accessibility.

[Priority 1]
A Web content developer **must** satisfy this checkpoint. Otherwise, one or more groups 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]
A Web content developer **should** satisfy this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing Web documents.
[Priority 3]
A Web content developer **may** address this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this checkpoint will improve access to Web documents.

Some checkpoints specify a priority level that may change under certain (indicated) conditions.

2 Techniques for Web Content Accessibility Guidelines

Guideline 1. Provide equivalent alternatives to auditory and visual content.

Checkpoints:

1.1 Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). *This includes*: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ASCII art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video. [Priority 1] (Checkpoint 1.1)

- Core Techniques: Text equivalents
- HTML Techniques: Images used as bullets
- HTML Techniques: Text for images used as links
- HTML Techniques: Short text equivalents for images ("alt-text")
- HTML Techniques: Long descriptions of images
- HTML Techniques: Text equivalents for client-side image maps
- HTML Techniques: Text and non-text equivalents for applets and programmatic objects
- HTML Techniques: Text equivalents for multimedia
- HTML Techniques: Describing frame relationships
- HTML Techniques: Writing for browsers that do not support FRAME
- HTML Techniques: Graphical buttons
- HTML Techniques: Alternative presentation of scripts

1.2 Provide redundant text links for each active region of a server-side image map. [Priority 1] (Checkpoint 1.2)

Refer also to checkpoint 1.5 and checkpoint 9.1.

- Core Techniques: Text equivalents
- HTML Techniques: Server-side image maps

1.3 Until user agents [p. 20] can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation. [Priority 1] (Checkpoint 1.3)

- Core Techniques: Visual information and motion

1.4 For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation. [Priority 1] (Checkpoint 1.4)

- Core Techniques: Audio information
- HTML Techniques: Directly accessible applets

1.5 Until user agents [p. 20] render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map. [Priority 3] (Checkpoint 1.5)

Refer also to checkpoint 1.2 and checkpoint 9.1.

- Core Techniques: Text equivalents

- HTML Techniques: Redundant text links for client-side image maps

Guideline 2. Don’t rely on color alone.

Checkpoints:

2.1 Ensure that all information conveyed with color is also available without color, for example from context or markup. [Priority 1] (Checkpoint 2.1)
- Core Techniques: Structure vs. Presentation
- CSS Techniques: Ensuring information is not in color alone

2.2 Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen. [Priority 2 for images, Priority 3 for text]. (Checkpoint 2.2)
- HTML Techniques: Color in images
- CSS Techniques: Color Contrast

Guideline 3. Use markup and style sheets and do so properly.

Checkpoints:

3.1 When an appropriate markup language exists, use markup rather than images to convey information. [Priority 2] (Checkpoint 3.1)
- Core Techniques: Structure vs. Presentation
- HTML Techniques: Markup and style sheets rather than images: The example of math
- CSS Techniques: Generated content

3.2 Create documents that validate to published formal grammars. [Priority 2] (Checkpoint 3.2)
- HTML Techniques: The !DOCTYPE statement

3.3 Use style sheets to control layout and presentation. [Priority 2] (Checkpoint 3.3)
- Core Techniques: Structure vs. Presentation
- HTML Techniques: Emphasis
- CSS Techniques: Text instead of images
- CSS Techniques: Text formatting and position
- CSS Techniques: Layout, positioning, layering, and alignment

3.4 Use relative rather than absolute units in markup language attribute values and style sheet property values. [Priority 2] (Checkpoint 3.4)
- HTML Techniques: Directly accessible applets
- HTML Techniques: Sizing frames with relative units
- CSS Techniques: Units of measure

3.5 Use header elements to convey document structure and use them according to specification. [Priority 2] (Checkpoint 3.5)
- Core Techniques: Structure vs. Presentation
- HTML Techniques: Section headings

3.6 Mark up lists and list items properly. [Priority 2] (Checkpoint 3.6)
- Core Techniques: Structure vs. Presentation
- HTML Techniques: Lists
- CSS Techniques: Providing contextual clues in HTML lists

3.7 Mark up quotations. Do not use quotation markup for formatting effects such as indentation. [Priority 2] (Checkpoint 3.7)
- HTML Techniques: Quotations

Guideline 4. Clarify natural language usage

Checkpoints:

4.1 Clearly identify changes in the natural language of a document's text and any text equivalents [p. 17] (e.g., captions). [Priority 1] (Checkpoint 4.1)
- HTML Techniques: Identifying changes in language

4.2 Specify the expansion of each abbreviation or acronym in a document where it first occurs. [Priority 3] (Checkpoint 4.2)
- HTML Techniques: Acronyms and abbreviations

4.3 Identify the primary natural language of a document. [Priority 3] (Checkpoint 4.3)
- HTML Techniques: Identifying the primary language

Guideline 5. Create tables that transform gracefully.

Checkpoints:

5.1 For data tables, identify row and column headers. [Priority 1] (Checkpoint 5.1)
- HTML Techniques: Identifying rows and column information

5.2 For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells. [Priority 1] (Checkpoint 5.2)
- HTML Techniques: Identifying rows and column information

5.3 Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version [p. 18]). [Priority 2] (Checkpoint 5.3)
- Core Techniques: Structure vs. Presentation
- HTML Techniques: Tables for layout
- CSS Techniques: Layout, positioning, layering, and alignment

5.4 If a table is used for layout, do not use any structural markup for the purpose of visual formatting. [Priority 2] (Checkpoint 5.4)
- Core Techniques: Structure vs. Presentation
- HTML Techniques: Tables for layout

5.5 Provide summaries for tables. [Priority 3] (Checkpoint 5.5)
- HTML Techniques: Providing summary information

5.6 Provide abbreviations for header labels. [Priority 3] (Checkpoint 5.6)
- HTML Techniques: Providing summary information

Refer also to checkpoint 10.3.

Guideline 6. Ensure that pages featuring new technologies transform gracefully.

Checkpoints:

6.1 Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document. [Priority 1] (Checkpoint 6.1)

- CSS Techniques: Generated content
- CSS Techniques: Rules and borders
- CSS Techniques: Using style sheet positioning and markup to transform gracefully

6.2 Ensure that equivalents for dynamic content are updated when the dynamic content changes. [Priority 1] (Checkpoint 6.2)

- HTML Techniques: Text and non-text equivalents for applets and programmatic objects
- HTML Techniques: Frame sources
- HTML Techniques: Alternative presentation of scripts

6.3 Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page. [Priority 1] (Checkpoint 6.3)

- HTML Techniques: Text and non-text equivalents for applets and programmatic objects
- HTML Techniques: Directly accessible scripts

6.4 For scripts and applets, ensure that event handlers are input device-independent. [Priority 2] (Checkpoint 6.4)

- Core Techniques: Structure vs. Presentation
- HTML Techniques: Directly accessible applets
- HTML Techniques: Directly accessible scripts

6.5 Ensure that dynamic content is accessible or provide an alternative presentation or page. [Priority 2] (Checkpoint 6.5)

- Core Techniques: Alternative pages
- Core Techniques: Audio information
- HTML Techniques: The LINK element and alternative documents
- HTML Techniques: Directly accessible applets
- HTML Techniques: Writing for browsers that do not support FRAME
- HTML Techniques: Graceful transformation of scripts

Refer also to checkpoint 11.4.

Guideline 7. Ensure user control of time-sensitive content changes.

Checkpoints:

7.1 Until user agents [p. 20] allow users to control flickering, avoid causing the screen to flicker. [Priority 1] (Checkpoint 7.1)
- Core Techniques: Screen flicker
- Core Techniques: Visual information and motion
- HTML Techniques: Directly accessible applets
- HTML Techniques: Scripts that cause flickering

7.2 Until user agents [p. 20] allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off). [Priority 2] (Checkpoint 7.2)
- HTML Techniques: Directly accessible applets
- HTML Techniques: Scripts that cause movement and blinking
- CSS Techniques: Text style effects

7.3 Until user agents [p. 20] allow users to freeze moving content, avoid movement in pages. [Priority 2] (Checkpoint 7.3)
- Core Techniques: Visual information and motion
- HTML Techniques: Animated images
- HTML Techniques: Directly accessible applets
- HTML Techniques: Scripts that cause movement and blinking
- CSS Techniques: Creating movement with style sheets and scripts

7.4 Until user agents [p. 20] provide the ability to stop the refresh, do not create periodically auto-refreshing pages. [Priority 2] (Checkpoint 7.4)
- Core Techniques: Automatic page refresh
- HTML Techniques: The META element
- HTML Techniques: Directly accessible applets
- HTML Techniques: Page updates and new windows

7.5 Until user agents [p. 20] provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects. [Priority 2] (Checkpoint 7.5)
- Core Techniques: Automatic page refresh
- HTML Techniques: The META element
- HTML Techniques: Page updates and new windows

Note. The BLINK and MARQUEE elements are not defined in any W3C HTML specification and should not be used. Refer also to guideline 11.

Guideline 8. Ensure direct accessibility of embedded user interfaces.

Checkpoint:

8.1 Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies [Priority 1 if functionality is important [p. 18] and not presented elsewhere, otherwise Priority 2.] (Checkpoint 8.1)

Refer also to guideline 6.

- HTML Techniques: Directly accessible applets
- HTML Techniques: Directly accessible scripts

Guideline 9. Design for device-independence.

Checkpoints:

9.1 Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape. [Priority 1] (Checkpoint 9.1)

Refer also to checkpoint 1.1, checkpoint 1.2, and checkpoint 1.5.

- HTML Techniques: Client-side versus server-side image maps

9.2 Ensure that any element that has its own interface can be operated in a device-independent manner. [Priority 2] (Checkpoint 9.2)

Refer to the definition of device independence [p. 16] .

Refer also to guideline 8.

- Core Techniques: Alternative pages
- HTML Techniques: Directly accessible applets

9.3 For scripts, specify logical event handlers rather than device-dependent event handlers. [Priority 2] (Checkpoint 9.3)

- Core Techniques: Alternative pages
- HTML Techniques: Directly accessible scripts

9.4 Create a logical tab order through links, form controls, and objects. [Priority 3] (Checkpoint 9.4)

- Core Techniques: Alternative pages
- HTML Techniques: Keyboard access
- HTML Techniques: Keyboard access to forms

9.5 Provide keyboard shortcuts to important links (including those in client-side image maps [p. 18]), form controls, and groups of form controls. [Priority 3] (Checkpoint 9.5)

- Core Techniques: Alternative pages
- HTML Techniques: Keyboard access
- HTML Techniques: Keyboard access to forms

Guideline 10. Use interim solutions.

Checkpoints:

10.1 Until user agents [p. 20] allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user. [Priority 2] (Checkpoint 10.1)

- HTML Techniques: Anchors and targets
- HTML Techniques: Directly accessible applets
- HTML Techniques: Using FRAME targets
- HTML Techniques: Page updates and new windows

10.2 Until user agents [p. 20] support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned. [Priority 2] (Checkpoint 10.2)

- HTML Techniques: Labeling form controls

10.3 Until user agents [p. 20] (including assistive technologies) render side-by-side text correctly, 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 3] (Checkpoint 10.3)

- HTML Techniques: Linearizing tables

10.4 Until user agents [p. 20] handle empty controls correctly, include default, place-holding characters in edit boxes and text areas. [Priority 3] (Checkpoint 10.4)

- HTML Techniques: Techniques for specific controls

10.5 Until user agents [p. 20] (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links. [Priority 3] (Checkpoint 10.5)

- HTML Techniques: Grouping and bypassing links

Guideline 11. Use W3C technologies and guidelines.

Checkpoints:

11.1 Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported. [Priority 2] (Checkpoint 11.1)

- Core Techniques: Technologies Reviewed for Accessibility

11.2 Avoid deprecated features of W3C technologies. [Priority 2] (Checkpoint 11.2)

- HTML Techniques: Index of HTML elements and attributes
- CSS Techniques: User override of styles
- CSS Techniques: Fonts

11.3 Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.) [Priority 3] (Checkpoint 11.3)

Note. Use content negotiation where possible.

- Core Techniques: Content negotiation
- CSS Techniques: Aural Cascading Style Sheets
- CSS Techniques: Access to alternative representations of content
- CSS Techniques: Media types

11.4 If, after best efforts [p. 13] , you cannot create an accessible [p. 15] page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent [p. 17] information (or functionality), and is updated as often as the inaccessible (original) page. [Priority 1] (Checkpoint 11.4)

- Core Techniques: Alternative pages

Note. Content developers should only resort to alternative pages when other solutions fail because alternative pages are generally updated less often than "primary" pages. 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 unavailable. Automatically generating alternative pages may lead to more frequent updates, but content developers 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; making it accessible is likely to improve it for all users.

Guideline 12. Provide context and orientation information.

Checkpoints:

12.1 Title each frame to facilitate frame identification and navigation. [Priority 1] (Checkpoint 12.1)

- HTML Techniques: Providing a frame title

12.2 Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone. [Priority 2] (Checkpoint 12.2)

- Core Techniques: Text equivalents
- HTML Techniques: Describing frame relationships

12.3 Divide large blocks of information into more manageable groups where natural and appropriate. [Priority 2] (Checkpoint 12.3)

- HTML Techniques: Structural grouping
- HTML Techniques: Grouping form controls

12.4 Associate labels explicitly with their controls. [Priority 2] (Checkpoint 12.4)

- HTML Techniques: Labeling form controls

Guideline 13. Provide clear navigation mechanisms.

Checkpoints:

13.1 Clearly identify the target of each link. [Priority 2] (Checkpoint 13.1)

- HTML Techniques: Link text

13.2 Provide metadata to add semantic information to pages and sites. [Priority 2] (Checkpoint 13.2)

- Core Techniques: Navigation
- HTML Techniques: Metadata
- CSS Techniques: Providing contextual clues in HTML lists

13.3 Provide information about the general layout of a site (e.g., a site map or table of contents). [Priority 2] (Checkpoint 13.3)

- Core Techniques: Navigation

13.4 Use navigation mechanisms in a consistent manner. [Priority 2] (Checkpoint 13.4)

- Core Techniques: Navigation

13.5 Provide navigation bars to highlight and give access to the navigation mechanism. [Priority 3] (Checkpoint 13.5)

- Core Techniques: Navigation

13.6 Group related links, identify the group (for user agents), and, until user agents [p. 20] do so, provide a way to bypass the group. [Priority 3] (Checkpoint 13.6)

- HTML Techniques: Grouping and bypassing links

13.7 If search functions are provided, enable different types of searches for different skill levels and preferences. [Priority 3] (Checkpoint 13.7)

- Core Techniques: Navigation

13.8 Place distinguishing information at the beginning of headings, paragraphs, lists, etc. [Priority 3] (Checkpoint 13.8)

- Core Techniques: Comprehension

13.9 Provide information about document collections (i.e., documents comprising multiple pages.). [Priority 3] (Checkpoint 13.9)

For example, in HTML specify document collections with the LINK element and the "rel" and "rev" attributes. Another way to create a collection is by building an archive (e.g., with zip, tar and gzip, stuffit, etc.) of the multiple pages.

- Core Techniques: Bundled documents
- HTML Techniques: The LINK element and navigation tools

13.10 Provide a means to skip over multi-line ASCII art. [Priority 3] (Checkpoint 13.10)

- HTML Techniques: Ascii art

Guideline 14. Ensure that documents are clear and simple.

Checkpoints:

14.1 Use the clearest and simplest language appropriate for a site’s content. [Priority 1] (Checkpoint 14.1)

- Core Techniques: Comprehension

14.2 Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page. [Priority 3] (Checkpoint 14.2)

- Core Techniques: Comprehension

14.3 Create a style of presentation that is consistent across pages. [Priority 3] (Checkpoint 14.3)

- Core Techniques: Navigation
- CSS Techniques: Decrease maintenance and increase consistency

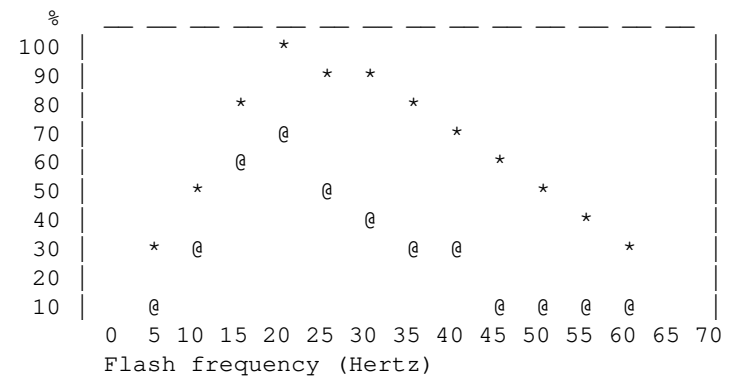
3 Glossary

Accessible
Content is accessible when it may be used by someone with a disability.

Applet
A program inserted into a Web page.

Assistive technology
Software or hardware that has been specifically designed to assist people with disabilities in carrying out daily activities. Assistive technology includes wheelchairs, reading machines, devices for grasping, etc. In the area of Web Accessibility, common software-based assistive technologies include screen readers, screen magnifiers, speech synthesizers, and voice input software that operate in conjunction with graphical desktop browsers (among other user agents [p. 20]). Hardware assistive technologies include alternative keyboards and pointing devices.

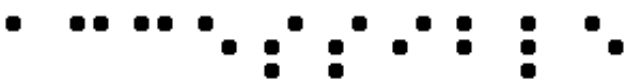
ASCII art
ASCII art refers to text characters and symbols that are combined to create an image. For example ";-)" is the smiley emoticon. The following is an ASCII figure showing the relationship between flash frequency and photoconvulsive response in patients with eyes open and closed [skip over ASCII figure [p. 15] or consult a description of chart]:



Authoring tool
HTML editors, document conversion tools, tools that generate Web content from databases are all authoring tools. Refer to the "Authoring Tool Accessibility Guidelines 1.0" (*[ATAG10] [p. 21]*) for information about developing accessible tools.

Backward compatible
Design that continues 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. Current dynamic braille displays range in size from one cell (six or eight dots) to an eighty-cell line, most having between twelve and twenty cells per line.

Content developer

Someone who authors Web pages or designs Web sites.

Deprecated

A deprecated element or attribute is one that has been outdated by newer constructs. Deprecated elements may become obsolete in future versions of HTML. The index of HTML elements and attributes in the Techniques Document indicates which elements and attributes are deprecated in HTML 4.01.

Authors should avoid using deprecated elements and attributes. User agents should continue to support them for reasons of backward compatibility.

Device independent

Users must be able to interact with a user agent (and the document it renders) using the supported input and output devices of their choice and according to their needs. Input devices may include pointing devices, keyboards, braille devices, head wands, microphones, and others. Output devices may include monitors, speech synthesizers, and braille devices.

Please note that "device-independent support" does not mean that user agents must support every input or output device. User agents should offer redundant input and output mechanisms for those devices that are supported. For example, if a user agent supports keyboard and mouse input, users should be able to interact with all features using either the keyboard or the mouse.

Document Content, Structure, and Presentation

The content of a document refers to what it says to the user through natural language, images, sounds, movies, animations, etc. The structure of a document is how it is organized logically (e.g., by chapter, with an introduction and table of contents, etc.). An element [p. 17] (e.g., P, STRONG, BLOCKQUOTE in HTML) that specifies document structure is called a ***structural element***. The presentation of a document is how the document is rendered (e.g., as print, as a two-dimensional graphical presentation, as an text-only presentation, as synthesized speech, as braille, etc.) An element [p. 17] that specifies document presentation (e.g., B, FONT, CENTER) is called a ***presentation element***.

Consider a document heading, for example. The content of the heading is what the heading says (e.g., "Sailboats"). In HTML, the heading is a structural element marked up with, for example, an H2 element. Finally, the presentation of the heading might be a bold block text in the margin, a centered line of text, a title spoken with a certain voice style (like an aural font), etc.

Dynamic HTML (DHTML)

DHTML is the marketing term applied to a mixture of standards including HTML, style sheets [p. 19] , the Document Object Model *[DOM1] [p. 21]* and scripting. However, there is no W3C specification that formally defines DHTML. Most guidelines may be applicable to applications using DHTML, however the following guidelines focus on issues related to scripting and style sheets: guideline 1, guideline 3, guideline 6, guideline 7, and guideline 9.

Element

This document uses the term "element" both in the strict SGML sense (an element is a syntactic construct) and more generally to mean a type of content (such as video or sound) or a logical construct (such as a heading or list). The second sense emphasizes that a guideline inspired by HTML could easily apply to another markup language.

Note that some (SGML) elements have content that is rendered (e.g., the P, LI, or TABLE elements in HTML), some are replaced by external content (e.g., IMG), and some affect processing (e.g., STYLE and SCRIPT cause information to be processed by a style sheet or script engine). An element that causes text characters to be part of the document is called a ***text element***.

Equivalent

Content is "equivalent" to other content when both fulfill essentially the same function or purpose upon presentation to the user. In the context of this document, the equivalent must fulfill essentially the same function for the person with a disability (at least insofar as is feasible, given the nature of the disability and the state of technology), as the primary content does for the person without any disability. For example, the text "The Full Moon" might convey the same information as an image of a full moon when presented to users. Note that equivalent information focuses on **fulfilling the same function**. If the image is part of a link and understanding the image is crucial to guessing the link target, an equivalent must also give users an idea of the link target. Providing equivalent information for inaccessible content is one of the primary ways authors can make their documents accessible to people with disabilities.

As part of fulfilling the same function of content an equivalent may involve a description of that content (i.e., what the content looks like or sounds like). For example, in order for users to understand the information conveyed by a complex chart, authors should describe the visual information in the chart.

Since text content can be presented to the user as synthesized speech, braille, and visually-displayed text, these guidelines require ***text equivalents*** for graphic and audio information. Text equivalents must be written so that they convey all essential content. ***Non-text equivalents*** (e.g., an auditory description of a visual presentation, a video of a person telling a story using sign language as an equivalent for a written story, etc.) also improve accessibility for people who cannot access visual information or written text, including many individuals with blindness, cognitive disabilities, learning disabilities, and deafness.

Equivalent information may be provided in a number of ways, including through attributes (e.g., a text value for the "alt" attribute in HTML and SMIL), as part of element content (e.g., the OBJECT in HTML), as part of the document's prose,

or via a linked document (e.g., designated by the "longdesc" attribute in HTML or a ***description link***). Depending on the complexity of the equivalent, it may be necessary to combine techniques (e.g., use "alt" for an abbreviated equivalent, useful to familiar readers, in addition to "longdesc" for a link to more complete information, useful to first-time readers).

A ***text transcript*** is a text equivalent of audio information that includes spoken words and non-spoken sounds such as sound effects. A ***caption*** is a text transcript for the audio track of a video presentation that is synchronized with the video and audio tracks. Captions are generally rendered visually by being superimposed over the video, which benefits people who are deaf and hard-of-hearing, and anyone who cannot hear the audio (e.g., when in a crowded room). A ***collated text transcript*** combines (collates) captions with text descriptions of video information (descriptions of the actions, body language, graphics, and scene changes of the video track). These text equivalents make presentations accessible to people who are deaf-blind and to people who cannot play movies, animations, etc. It also makes the information available to search engines.

One example of a non-text equivalent is an ***auditory description*** of the key visual elements of a presentation. The description is either a prerecorded human voice or a synthesized voice (recorded or generated on the fly). The auditory description is synchronized with the audio track of the presentation, usually during natural pauses in the audio track. Auditory descriptions include information about actions, body language, graphics, and scene changes.

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.

Content developers 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

Information in a document is important if understanding that information is crucial to understanding the document.

Linearized table

A table rendering process where the contents of the cells become a series of paragraphs (e.g., down the page) one after another. The paragraphs will occur in the same order as the cells are defined in the document source. Cells should make sense when read in order and should include structural elements [p. 16] (that create paragraphs, headings, lists, etc.) so the page makes sense after

linearization.

Link text

The rendered text content of a link.

Natural Language

Spoken, written, or signed human languages such as French, Japanese, American Sign Language, and braille. The natural language of content may be indicated with the "lang" attribute in HTML (*[HTML4] [p. 21]* , section 8.1) and the "xml:lang" attribute in XML (*[XML] [p. 22]* , section 2.12).

Navigation Mechanism

A navigation mechanism is any means by which a user can navigate a page or site. Some typical mechanisms include:

navigation bars

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

site maps

A site map provides a global view of the organization of a page or site.

tables of contents

A table of contents generally lists (and links to) the most important sections of a document.

Personal Digital Assistant (PDA)

A PDA is a small, portable computing device. Most PDAs are used to track personal data such as calendars, contacts, and electronic mail. A PDA 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. Screen magnifiers are used primarily by individuals with low vision.

Screen reader

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

Style sheets

A style sheet is a set of statements that specify presentation of a document. Style sheets may have three different origins: they may be written by content providers, created by users, or built into user agents. In CSS (*[CSS2] [p. 21]*), the interaction of content provider, user, and user agent style sheets is called the *cascade*.

Presentation markup is markup that achieves a stylistic (rather than structuring) effect such as the B or I elements in HTML. Note that the STRONG and EM elements are not considered presentation markup since they convey information that is independent of a particular font style.

Tabular information

When tables are used to represent logical relationships among data -- text, numbers, images, etc., that information is called "tabular information" and the tables are called "data tables". The relationships expressed by a table may be

rendered visually (usually on a two-dimensional grid), aurally (often preceding cells with header information), or in other formats.

Until user agents ...

In most of the checkpoints, content developers are asked to ensure the accessibility of their pages and sites. However, there are accessibility needs that would be more appropriately met by user agents [p. 20] (including assistive technologies [p. 15]). As of the publication of this document, not all user agents or assistive technologies provide the accessibility control users require (e.g., some user agents may not allow users to turn off blinking content, or some screen readers may not handle tables well). Checkpoints that contain the phrase "until user agents ..." require content developers to provide additional support for accessibility until most user agents readily available to their audience include the necessary accessibility features.

Note. The WAI Web site (refer to *[WAI-UA-SUPPORT] [p. 22]*) provides information about user agent support for accessibility features. Content developers are encouraged to consult this page regularly for updated information.

User agent

Software to access Web content, including desktop graphical browsers, text browsers, voice browsers, mobile phones, multimedia players, plug-ins, and some software assistive technologies used in conjunction with browsers such as screen readers, screen magnifiers, and voice recognition software. Refer to the "User Agent Accessibility Guidelines 1.0" (*[UAAG10] [p. 21]*) for information about developing accessible tools.

---

4 References

For the latest version of any W3C specification please consult the list of W3C Technical Reports.

[ATAG10]
"Authoring Tool Accessibility Guidelines 1.0", J. Treviranus, C. McCathieNevile, I. Jacobs, and J. Richards, eds., 3 February 2000. This ATAG 1.0 Recommendation is http://www.w3.org/TR/2000/REC-ATAG10-20000203/.

[CSS2]
"CSS, level 2 Recommendation", B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, eds., 12 May 1998. This CSS2 Recommendation is http://www.w3.org/TR/1998/REC-CSS2-19980512/. The latest version of CSS2 is available at http://www.w3.org/TR/REC-CSS2.

[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., 1 October 1998. This DOM Level 1 Recommendation is http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001. The latest version of DOM Level 1 is available at http://www.w3.org/TR/REC-DOM-Level-1.

[HTML4]
"HTML 4.01 Recommendation", D. Raggett, A. Le Hors, and I. Jacobs, eds., 24 December 1999. This HTML 4.01 Recommendation is http://www.w3.org/TR/1999/REC-html401-19991224/.

[SMIL]
"Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", P. Hoschka, ed., 15 June 1998. This SMIL 1.0 Recommendation is http://www.w3.org/TR/1998/REC-smil-19980615/. The latest version of SMIL 1.0 is available at http://www.w3.org/TR/REC-smil.

[SMIL-ACCESS]
"Accessibility Features of SMIL", M. Koivunen and I. Jacobs, eds., 21 September 1999. This W3C Note is http://www.w3.org/TR/1999/NOTE-SMIL-access-19990921/.

[SVG]
"Scalable Vector Graphics (SVG) 1.0 Specification", J. Ferraiolo, ed., 2 August 2000. This W3C Candidate Recommendation is http://www.w3.org/TR/2000/CR-SVG-20000802/.

[SVG-ACCESS]
"Accessibility Features of SVG", C. McCathieNevile and M. Koivunen, eds., 7 August 2000. This W3C Note is http://www.w3.org/TR/2000/NOTE-SVG-access-20000807.

[UAAG10]
"User Agent Accessibility Guidelines", J. Gunderson and I. Jacobs, eds. The latest version of the User Agent Accessibility Guidelines is available at http://www.w3.org/TR/UAAG10/.

[WCAG10]
"Web Content Accessibility Guidelines 1.0", W. Chisholm, G. Vanderheiden, and I. Jacobs, eds., 5 May 1999. This WCAG 1.0 Recommendation is http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/.

[WCAG10-CORE-TECHNIQUES]
"Core Techniques for Web Content Accessibility Guidelines 1.0", W. Chisholm, G. Vanderheiden, and I. Jacobs, eds. The latest version of this document is available at http://www.w3.org/TR/WCAG10-CORE-TECHS/.

[WCAG10-CSS-TECHNIQUES]
"CSS Techniques for Web Content Accessibility Guidelines 1.0", W. Chisholm, G. Vanderheiden, and I. Jacobs, eds. The latest version of this document is available at http://www.w3.org/TR/WCAG10-CSS-TECHS/.

[WCAG10-HTML-TECHNIQUES]
"HTML Techniques for Web Content Accessibility Guidelines 1.0", W. Chisholm, G. Vanderheiden, and I. Jacobs, eds. The latest version of this document is available at http://www.w3.org/TR/WCAG10-HTML-TECHS/.

[XML]
"Extensible Markup Language (XML) 1.0.", T. Bray, J. Paoli, C.M. Sperberg-McQueen, eds., 10 February 1998. This XML 1.0 Recommendation is: http://www.w3.org/TR/1998/REC-xml-19980210. The latest version of XML 1.0 is available at http://www.w3.org/TR/REC-xml.

5 Resources

Note: *W3C does not guarantee the stability of any of the following references outside of its control. These references are included for convenience. References to products are not endorsements of those products.*

5.1 Other Guidelines

[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.

5.2 User agents and other tools

A list of alternative Web browsers (assistive technologies and other user agents designed for accessibility) is maintained at the WAI Web site.

[WAI-UA-SUPPORT]
User Agent Support for Accessibility

6 Acknowledgments

Web Content Guidelines Working Group Co-Chairs:

Jason White, University of Melbourne

Gregg Vanderheiden, Trace Research and Development

W3C Team contact:

Wendy Chisholm

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, Eric Hansen, Phill Jenkins, Leonard Kasday, George Kerscher, Marja-Riitta Koivunen, Josh Krieger, Chuck Letourneau, Scott Luebking, William Loughborough, Murray Maloney, Charles McCathieNevile, MegaZone (Livingston Enterprises), Masafumi Nakane, Mark Novak, Charles Oppermann, Mike Paciello, David Pawson, Michael Pieper, Greg Rosmaita, Liam Quinn, Dave Raggett, T.V. Raman, Robert Savellis, Jutta Treviranus, Steve Tyler, and Jaap van Lelieveld