



HTML Techniques for Web Content Accessibility Guidelines 1.0

W3C Note 6 November 2000

This version:

<http://www.w3.org/TR/2000/NOTE-WCAG10-HTML-TECHS-20001106/>
(plain text, PostScript, PDF, gzip tar file of HTML, zip archive of HTML)

Latest version:

<http://www.w3.org/TR/WCAG10-HTML-TECHS/>

Previous version:

<http://www.w3.org/TR/2000/NOTE-WCAG10-HTML-TECHS-20000920/>

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 describes techniques for authoring accessible Hypertext Markup Language (HTML) content (refer to HTML 4.01 [*HTML4*] [p. 54]). This document is intended to help authors of Web content who wish to claim conformance to "Web Content Accessibility Guidelines 1.0" (*WCAG10*] [p. 54]). While the techniques in this document should help people author HTML that conforms to "Web Content Accessibility Guidelines 1.0", these techniques are neither guarantees of conformance nor the only way an author might produce conforming content.

This document is part of a series of documents about techniques for authoring accessible Web content. For information about the other documents in the series, please refer to "Techniques for Web Content Accessibility Guidelines 1.0" [*WCAG10-TECHS*] [p. 54].

Note: This document contains a number of examples that illustrate accessible solutions in CSS but also deprecated examples that illustrate what content developers should not do. The deprecated examples are highlighted and readers should approach them with caution -- they are meant for illustrative purposes only.

Status of this document

This version has been published to correct some broken links in the previous version.

The 6 November 2000 version of this document is a Note in a series of Notes produced and endorsed by the Web Content Accessibility Guidelines Working Group (WCAG WG). This Note has not been reviewed or endorsed by W3C Members. The series of documents supersedes the single document 5 May 1999 W3C Note Techniques for Web Content Accessibility Guidelines 1.0. The topics from the earlier document have been separated into technology-specific documents that may evolve independently. Smaller technology-specific documents allow authors to focus on a particular technology.

While the "Web Content Accessibility Guidelines 1.0" Recommendation [WCAG10] [p. 54] 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 content.

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 specification is the only normative version. Translations of this document may be available.

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 Web Content Accessibility Guidelines Working Group 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 Document structure and metadata | .5 |
| 1.1 Metadata | .5 |
| 1.2 Structural grouping | .8 |
| 2 Language information | .9 |
| 2.1 Identifying changes in language | .9 |
| 2.2 Identifying the primary language | 10 |
| 3 Text markup | 10 |
| 3.1 Emphasis | 10 |
| 3.2 Acronyms and abbreviations | 11 |
| 3.3 Quotations | 11 |
| 3.4 Markup and style sheets rather than images: The example of math | 12 |
| 3.5 Eight other structural elements (to identify citations, code fragments, deleted text, and others) | 13 |
| 4 Lists | 13 |
| 4.1 Use style sheets to change list bullets | 14 |
| 5 Tables | 16 |
| 5.1 Tables of data | 16 |
| 5.2 Tables for layout | 21 |
| 5.3 Linearizing tables | 21 |
| 5.4 Backward compatibility issues for tables | 22 |
| 6 Links | 22 |
| 6.1 Link text | 22 |
| 6.2 Grouping and bypassing links | 24 |
| 6.3 Keyboard access | 25 |
| 6.4 Anchors and targets | 25 |
| 7 Images and image maps | 25 |
| 7.1 Short text equivalents for images ("alt-text") | 25 |
| 7.2 Long descriptions of images | 26 |
| 7.3 Ascii art | 28 |
| 7.4 Image maps | 29 |
| 7.5 Color in images | 32 |
| 7.6 Animated images | 32 |
| 8 Applets and other programmatic objects | 32 |
| 8.1 Text and non-text equivalents for applets and programmatic objects | 32 |
| 8.2 Directly accessible applets | 33 |
| 9 Audio and video | 34 |
| 9.1 Audio information | 34 |
| 9.2 Text equivalents for multimedia | 34 |
| 9.3 Embedding multimedia objects | 35 |

| | |
|---|----|
| 10 Frames | 35 |
| 10.1 Providing a frame title | 36 |
| 10.2 Describing frame relationships | 36 |
| 10.3 Writing for browsers that do not support FRAME | 38 |
| 10.4 Frame sources | 38 |
| 10.5 Using FRAME targets | 39 |
| 10.6 Alternatives to frames | 40 |
| 11 Forms | 41 |
| 11.1 Keyboard access to forms | 41 |
| 11.2 Grouping form controls | 41 |
| 11.3 Labeling form controls | 43 |
| 11.4 Graphical buttons | 43 |
| 11.5 Techniques for specific controls | 44 |
| 11.6 Backward compatibility issues for forms | 44 |
| 12 Scripts | 45 |
| 12.1 Graceful transformation of scripts | 45 |
| 12.2 Scripts that cause flickering | 45 |
| 12.3 Scripts that cause movement and blinking | 45 |
| 12.4 Directly accessible scripts | 46 |
| 12.5 Alternative presentation of scripts | 47 |
| 12.6 Page updates and new windows | 47 |
| 13 Index of HTML elements and attributes | 48 |
| Elements | 48 |
| Attributes | 51 |
| 14 References | 54 |
| 15 Resources | 54 |
| 15.1 Other guidelines | 55 |
| 15.2 User agents and other tools | 55 |
| 15.3 Accessibility resources | 55 |
| 16 Acknowledgments | 56 |

1 Document structure and metadata

Content developers should use structural markup and use it according to specification. Structural elements and attribute (refer to the index of HTML elements and attributes [p. 48] to identify them) promote consistency in documents and supply information to other tools (e.g., indexing tools, search engines, programs that extract tables to databases, navigation tools that use heading elements, and automatic translation software that translates text from one language into another).

1.1 Metadata

Checkpoints in this section:

- 13.2 Provide metadata to add semantic information to pages and sites. [Priority 2]

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

1.1.1 *TITLE: The document title.*

Note that the (mandatory) TITLE element, which only appears once in a document, is different from the "title [p. 52]" attribute, which applies to almost every HTML 4.01 element. Content developers should use the "title" attribute in accordance with the HTML 4.01 specification. For example, "title" should be used with links to provide information about the target of the link.

1.1.2 *The ADDRESS element*

This element can be used to provide information about the creator of the page.

1.1.3 *The META element*

Checkpoints in this section:

- 7.4 Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages. [Priority 2] ,
- 7.5 Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects. [Priority 2] .

This element can specify metadata for a document including keywords, and information about the author. Please refer to the section on automatic page refresh for information on why META should **not** be used to redirect or auto-refresh pages.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.