Authoring Tool Accessibility Guidelines 1.0

W3C Proposed Recommendation 18 December 1999

This version:
http://www.w3.org/WAI/AU/PR-WAI-AUTOOLS-19991218
(plain text, HTML gzip tar archive, HTML zip archive, PostScript, PDF)
Latest version:
http://www.w3.org/WAI/AU/WAI-AUTOOLS
Previous version:
http://www.w3.org/WAI/AU/PR-WAI-AUTOOLS-19991210

Editors:
Jutta Treviranus - ATRC, University of Toronto
Jan Richards - University of Toronto
Ian Jacobs - W3C
Charles McCathieNevile - W3C

Abstract

This specification provides guidelines for Web authoring tool developers. Its purpose is two-fold: to assist developers in designing authoring tools that produce accessible Web content and to assist developers in creating an accessible authoring interface.

Authoring tool users ("authors") can be enabled, encouraged and assisted to create accessible Web content through prompts, alerts, checking and repair functions, help files and automated tools. It is equally important that all people can be the authors of Web content, rather than merely recipients. The tools used to create this information must therefore be accessible themselves. Adoption of these guidelines will contribute to the proliferation of Web content that can be read by a broader range of readers and in authoring tools that can be used by a broader range of authors.

This document is part of a series of accessibility documents published by the W3C Web Accessibility Initiative.

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.

This document is a revision of the Authoring Tool Guidelines Proposed Recommendation dated 26 October 1999, reviewed by W3C Members and other interested parties. The current document, for Working Group review, follows the Working Group meeting on 15 December 1999 and reflects the resolutions of Working Group meetings discussing issues raised during the Proposed Recommendation period. A log of changes between successive Working Drafts is available. For further information consult the minutes of Working Group Meetings.

The Working Group anticipates no changes to this specification that would affect implementation, and encourages active implementation to test the specification during the Proposed Recommendation review period.

Publication as a Proposed Recommendation does not imply endorsement by the W3C Membership. This is still a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite W3C Proposed Recommendations as other than "work in progress."

The goals of the WAI AU Working Group are discussed in the WAI AU charter.

Please send general comments about this document to the public mailing list: w3c-wai-au@w3.org, archived at http://lists.w3.org/Archives/Public/w3c-wai-au

A list of the current AU Working Group participants is available.

A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR.

Table of Contents

Abstract 1
Status of this document 2
1. Introduction 4
1.1 How the Guidelines are organized. 5
1.2 Checkpoint Priorities 5
1.3 Conformance to these Guidelines 7
2. Guidelines 8
1. Support accessible authoring practices 8
2. Generate standard markup 8
3. Support the creation of accessible content 9
4. Provide ways of checking and correcting inaccessible content . . 10
5. Integrate accessibility solutions into the overall "look and feel" . . 11
6. Promote accessibility in help and documentation 11
7. Ensure that the authoring tool is accessible to authors with disabilities 11
3. Glossary of Terms and Definitions 13
4. Acknowledgments 17
5. References 18

An appendix to this document [WAI-AUTOOLS-CHECKLIST] lists all checkpoints for convenient reference.

1. Introduction

In these guidelines, the term authoring tool refers to the wide range of software used for creating Web content, including:

- Editing tools specifically designed to produce Web content (e.g., WYSIWYG HTML and XML editors);
- Tools that offer the option of saving material in a Web format (e.g., word processors or desktop publishing packages);
- Tools that translate documents into Web formats (e.g., filters to translate desktop publishing formats to HTML);
- Tools that produce multimedia, especially where it is intended for use on the Web (e.g., video production and editing suites, SMIL authoring packages);
- Tools for site management or site publication, including tools that automatically generate Web sites dynamically from a database, on-the-fly conversion and Web site publishing tools;
- Tools for management of layout (e.g., CSS formatting tools).

There are tools (such as text editors) that allow motivated users to create accessible Web content, or are accessible themselves to users with certain disabilities, but that may not conform to this specification. The purpose of this specification is to address the needs of the largest population possible, so that there is more accessible content on the Web. The goals of this document can be stated as follows: that the authoring tool be accessible to authors regardless of disability, that it produce accessible content by default, and that it support and encourage the author in creating accessible content. Because most of the content of the Web is created using authoring tools, they play a critical role in ensuring the accessibility of the Web. Since the Web is both a means of receiving information and communicating information, it is important that both the Web content produced and the authoring tool itself be accessible.

To achieve these goals, authoring tool developers must take steps such as ensuring conformance to accessible standards (e.g., HTML 4.0), checking and correcting accessibility problems, prompting, and providing appropriate documentation and help. For detailed information about what constitutes accessible content, this specification relies on the Web Content Accessibility Guidelines [WAI-WEBCONTENT] . Similarly, rather than directly reproducing existing specifications that address general accessible software design the present specification relies on other sources. It does address accessible design considerations specific to Web authoring tools such as providing flexible editing views, navigation aids and access to display properties for authors.

In addition, accessible design will benefit many people who do not have a physical disability but with similar needs. For example they may be working in a noisy environment and unable to hear, or need to use their eyes for another task, and be unable to view a screen. They may be using a small mobile device, with a small screen, no keyboard and no mouse.

A separate document, entitled "Techniques for Authoring Tool Accessibility" [WAI-AUTOOLS-TECHS] , provides suggestions and examples of how each checkpoint might be satisfied. It also includes references to other accessibility resources (such as platform-specific software accessibility guidelines) that provide additional information on how a tool may satisfy each checkpoint. Readers are strongly encouraged to become familiar with the Techniques Document. Note that the Techniques provided are informative examples only, and other strategies may be used to meet the checkpoint as well as, or in place of, those discussed.

1.1 How the Guidelines are organized.

This document includes guidelines, which are general principles of accessible design. Each guideline includes:

- The guideline number;
- The statement of the guideline;
- The rationale behind the guideline;
- A list of checkpoint definitions.

The checkpoint definitions in each guideline specify requirements for authoring tools to follow the guideline. Each checkpoint definition includes:

- The checkpoint number;
- The statement of the checkpoint;
- The priority of the checkpoint;
- In some cases informative notes, clarifying examples, or cross references to related guidelines or checkpoints;
- A link to a section of "Techniques for Authoring Tool Accessibility" [WAI-AUTOOLS-TECHS] where implementations and examples of the checkpoint are discussed;

Each checkpoint is intended to be specific enough that it can be verified, while being sufficiently general to allow developers the freedom to use the most appropriate strategies to meet the checkpoint.

An appendix to this specification [WAI-AUTOOLS-CHECKLIST] lists all checkpoints for convenient reference.

1.2 Checkpoint Priorities

Each checkpoint has a priority level. The priority level reflects the impact of the checkpoint in meeting the goals of this specification. These goals are:

- That the authoring tool be accessible
- That the authoring tool produce accessible content by default
- That the authoring tool encourage the creation of accessible content

The three priority levels are assigned as follows:

[Priority 1]
If the checkpoint is essential to meeting the goals
[Priority 2]
If the checkpoint is important to meeting the goals
[Priority 3]
If the checkpoint is beneficial to meeting the goals
[Relative Priority]

Some checkpoints that refer to generating, authoring, or checking Web content have multiple priorities. The priority is dependent on the priority in the Web Content Accessibility Guidelines (WCAG) [WAI-WEBCONTENT] .

- It is priority 1 to implement the checkpoint for content features that are a priority 1 requirement in WCAG.
- It is priority 2 to implement the checkpoint for content features that are a priority 2 requirement in WCAG.
- It is priority 3 to implement the checkpoint for content features that are a priority 3 requirement in WCAG.

For example:

- providing text equivalents for images and audio is a priority 1 requirement in WCAG since without it one or more groups will find it impossible to access the information. Therefore, it is a priority 1 requirement for the authoring tool to check for (4.1) or ask the author for (3.1) equivalent alternatives for these types of content.
- Grouping links in navigation bars is a priority 3 in WCAG. Therefore, it is only priority 3 for the authoring tool to check for (4.1) or ask the author for (3.2) groups of links that are not grouped in the markup as a navigation mechanism.

The implementation of the checkpoints will vary from tool to tool. When a checkpoint within this document refers to the WCAG [WAI-WEBCONTENT] , only the WCAG, checkpoints that refer to content supported or automatically generated by the authoring tool apply, as noted in the relevant checkpoints. In some cases support can be provided automatically, without the need for explicit author participation, in other cases human judgment is required and support is provided by the tool in the form of prompts and documentation.

In choosing priority levels for checkpoints, the Working Group has assumed that "the author" is a competent, but not necessarily expert, user of the authoring tool, and that the author has no prior knowledge of accessibility. For example, the author is not expected to have read all of the documentation but is expected to know how to turn to the documentation for assistance.

1.3 Conformance to these Guidelines

This section defines three levels of conformance to this document:

- Conformance Level "A": all Priority 1 checkpoints are satisfied;
- Conformance Level "Double-A": all Priority 1 and 2 checkpoints are satisfied;
- Conformance Level "Triple-A": all Priority 1, 2, and 3 checkpoints are satisfied;

Note. Some example conformance evaluations are available. It should be noted that conformance claims are not necessarily validated or endorsed by W3C.

Claims of conformance to this document must use one of the following two forms.

Form 1: Specify:

- The guidelines' title: "Authoring Tool Accessibility Guidelines 1.0"
- The guidelines' URI: http://www.w3.org/WAI/AU/PR-WAI-AUTOOLS-19991218
- The conformance level satisfied: "A", "Double-A", or "Triple-A".
- The product covered by the claim (e.g., tool name and version number, upgrades or plug-ins required).
- The date of the claim.
- The checkpoints that are satisfied, and those that are considered not applicable

Example of Form 1: "MyAuthoringTool version 2.3 conforms to W3C's "Authoring Tool Accessibility Guidelines 1.0", available at http://www.w3.org/WAI/AU/PR-WAI-AUTOOLS-19991218, level Double-A."

Form 2: Include, on each statement of conformance, one of three icons provided by W3C and link the icon to the appropriate W3C explanation of the claim.

[Editors' note: In the event this document becomes a Recommendation, by that date WAI will provide a set of three icons, for "A", "Double-A", or "Triple-A" conformance levels of "Authoring Tool Accessibility Guidelines 1.0", together with a stable URI to the W3C Web site for linking the icons to the W3C explanation of conformance claims.]

2. Guidelines

Guideline 1. Support accessible authoring practices

If the tool automatically generates markup, many authors will be unaware of the accessibility status of the final content unless they expend extra effort to make appropriate corrections by hand. Since many authors are unfamiliar with accessibility, the onus is on the authoring tool to automatically generate accessible markup, and where appropriate, to guide the author in producing accessible content.

Many applications feature the ability to convert documents from other formats (e.g., Rich Text Format) into a markup format specifically intended for the Web such as HTML. Markup changes may also be made to facilitate efficient editing and manipulation. It is essential that these processes do not introduce inaccessible markup , or remove accessibility content, particularly since the markup changes are hidden from the author’s view in many tools.

Checkpoints:

1.1 Ensure that the author can produce accessible content in the markup language(s) supported by the tool. [Priority 1]
1.2 Ensure that the tool preserves all accessibility information during authoring, transformations and conversions . [Priority 1]
1.3 Ensure that when the tool automatically generates markup it conforms to the W3C’s Web Content Accessibility Guidelines [WAI-WEBCONTENT] .
[Relative Priority]
1.4 Ensure that templates provided by the tool conform to the Web Content Accessibility Guidelines [WAI-WEBCONTENT] . [Relative Priority]

Guideline 2. Generate standard markup

Conformance with standards promotes interoperability and accessibility, by making it easier to create specialized user agents that address the needs of users with disabilities. In particular many assistive technologies used with browsers and multimedia players are only able to provide access to Web documents that use valid markup. Therefore, valid markup is an essential aspect of authoring tool accessibility.

Where applicable use W3C Recommendations, which have been reviewed to ensure accessibility and interoperability. If there are no applicable W3C Recommendations, use a published standard that enables accessibility.

Checkpoints:

2.1 Use the latest versions of W3C Recommendations when they are available and appropriate for a task. [Priority 2]

> W3C specifications have undergone review specifically to ensure that they do not compromise accessibility, and where possible, they enhance it.

2.2 Ensure that the tool automatically generates valid markup. [Priority 1]

This is necessary for user agents to be able to render Web content in a manner appropriate to a particular user's needs.

2.3 If markup produced by the tool does not conform to W3C specifications, inform the author. [Priority 3]

Guideline 3. Support the creation of accessible content

Well-structured information and equivalent alternative information are cornerstones of accessible design, allowing information to be presented in a way most appropriate for the needs of the user without constraining the creativity of the author. Yet generating equivalent information, such as textual alternatives for images and auditory descriptions of video, can be one of the most challenging aspects of Web design, and authoring tool developers should attempt to facilitate and automate the mechanics of this process. For example, prompting authors to include equivalent alternative information such as text equivalents , captions , and auditory descriptions at appropriate times can greatly ease the burden for authors. Where such information can be mechanically determined and offered as a choice for the author (e.g., the function of icons in an automatically-generated navigation bar, or expansion of acronyms from a dictionary) the tool can assist the author. At the same time it can reinforce the need for such information and the author's role in ensuring that it is used appropriately in each instance.

Checkpoints:

3.1 Prompt the author to provide equivalent alternative information (e.g., captions , auditory descriptions and collated text transcripts for video). [Relative Priority]

Note. Some Checkpoints in Web Content Accessibility Guidelines [WAI-WEBCONTENT] may not be applicable.

3.2 Help the author create structured content and separate information from its presentation. [Relative Priority]

Note: Some Checkpoints in Web Content Accessibility Guidelines [WAI-WEBCONTENT] may not be applicable.

3.3 Ensure that prepackaged content conforms to Web Content Accessibility Guidelines [WAI-WEBCONTENT] . [Relative Priority]

For example include captions , an auditory description , and a collated text transcript with prepackaged movies. Refer also to checkpoint 3.4.

3.4 Do not automatically generate equivalent alternatives . Do not reuse previously authored alternatives without author confirmation, except when the function is known with certainty. [Priority 1]

For example, prompt the user for a text equivalent of an image. If the author has already provided a text equivalent for the same image used in another document, offer to reuse that text and prompt the author for confirmation. If the tool automatically generates a "Search" icon, it would be appropriate to automatically reuse the previously authored text equivalent for that icon. Refer also to checkpoints checkpoint 3.3 and checkpoint 3.5.

Note. Human-authored equivalent alternatives may be available for an object (for example through checkpoint 3.5 and/or checkpoint 3.3). It is appropriate for the tool to offer these to the author as defaults.

3.5 Provide functionality for managing, editing, and reusing alternative equivalents for multimedia objects. [Priority 3]

Note: These alternative equivalents may be packaged with the tool, written by the author, retrieved from the Web, etc.

Guideline 4. Provide ways of checking and correcting inaccessible content

Many authoring tools allow authors to create documents with little or no knowledge about the underlying markup. To ensure accessibility, authoring tools must be designed so that they can (where possible, automatically) identify inaccessible markup , and enable its correction even when the markup itself is hidden from the author.

In supporting the creation of accessible Web content, authoring tools should take into account differing authoring styles. In general, authors will prefer to be able to configure their tools to support their working style. Tools that allow such configuration can help authors feel that accessible authoring is a natural practice (refer to guideline 5) rather than an intrusion on their normal work pattern. For example some users may prefer to be alerted to accessibility problems when they occur, whereas others may prefer to perform a check at the end of an editing session. This is analogous to programming environments that allow users to decide whether to check for correct code during editing or at compile time.

Note. Validation of markup is an essential aspect of checking the accessibility of content.

Checkpoints:

4.1 Check for and inform the author of accessibility problems . [Relative Priority]

Note: Accessibility problems should be detected automatically where possible. Where this is not possible, the tool may need to prompt the user to make decisions, or to manually check for certain types of problem.

4.2 Assist authors in correcting accessibility problems . [Relative Priority]

At a minimum, provide context-sensitive help with the accessibility checking required by 4.1

4.3 Allow the author to preserve markup not recognized by the tool. [Priority 2]

Note. The author may have included or imported markup that enhances accessibility but is not recognized by the tool.

4.4 Provide the author with a summary of the document’s accessibility status. [Priority 3]

4.5 Allow the author to transform presentation markup that is misused to convey structure into structural markup , and to transform presentation markup used for style into style sheets. [Priority 3]

Guideline 5. Integrate accessibility solutions into the overall "look and feel"

When a new feature is added to an existing software tool without proper integration, the result is often an obvious discontinuity. Differing color schemes, fonts, interaction styles and even application stability can be factors affecting user acceptance of the new feature. In addition, the relative prominence of different ways to accomplish the same task can be an important factor in which method an author chooses. Therefore, it is important that creating accessible content is a natural process when using an authoring tool.

Checkpoints:

5.1 Ensure that functionality related to accessible authoring practices is naturally integrated into the overall look and feel of the tool. [Priority 2]
5.2 Ensure that Web Content Accessibility Guidelines [WAI-WEBCONTENT] Priority 1 accessible authoring practices are among the most obvious and easily initiated by the author. [Priority 2]

Guideline 6. Promote accessibility in help and documentation

The issues surrounding the creation of accessible Web content are often unknown to Web authors. Help and documentation must include explanations of accessibility problems , and should demonstrate solutions with examples.

Checkpoints:

6.1 Document all features that promote the production of accessible content. [Priority 1]
6.2 Ensure that creating accessible content is a naturally integrated part of the documentation, including examples. [Priority 2]
6.3 In a dedicated section, document all features of the tool that promote the production of accessible content. [Priority 3]

Guideline 7. Ensure that the authoring tool is accessible to authors with disabilities

The authoring tool is a software program with standard user interface elements and as such must be designed according to relevant user interface accessibility guidelines. When custom interface components are created it is essential that they are accessible through standard access mechanisms.

Some additional user interface design considerations apply specifically to Web authoring tools . For instance, authoring tools must ensure that the author can edit (in the editing view) using one set of stylistic preferences and publish using different styles. For instance, authors with low vision may need large text when editing but want to publish with a smaller default text size. The style preferences of the editing view must not affect the markup of the published document.

Authoring tools must also ensure that the author can navigate a document efficiently while editing, regardless of disability. Authors who use screen readers, refreshable braille displays, or screen magnifiers can make limited use (if at all) of graphical artifacts that communicate the structure of the document and act as signposts when traversing it. For authors with blindness or motor disabilities, fatigue and other problems that arise when serial access is the only navigation technique are major usability issues. Authoring tools should therefore provide an editing view that conveys a sense of the overall structure and allows structured navigation.

Note. Documentation, help files, and installation are part of the software and need to be available in an accessible form.

Checkpoints:

7.1 Use all applicable operating system and accessibility standards and conventions (Priority 1 for standards and conventions that are essential to accessibility, Priority 2 for those that are important to accessibility, Priority 3 for those that are beneficial to accessibility).

> The techniques for this checkpoint include references to checklists and guidelines for a number of platforms and to general guidelines for accessible applications.

7.2 Allow the author to change the presentation within editing views without affecting the document markup. [Priority 1]

> This allows the author to edit the document according to personal requirements, without changing the way the document is rendered when published.

7.3 Allow the author to edit all properties of each element and object in an accessible fashion. [Priority 1]

7.4 Ensure the editing view allows navigation via the structure of the document in an accessible fashion. [Priority 1]

7.5 Enable editing of the structure of the document in an accessible fashion. [Priority 2]

7.6 Allow the author to search within editing views . [Priority 2]

3. Glossary of Terms and Definitions

Accessibility (Also: ***Accessible***)

Within these guidelines, "accessible Web content" and "accessible authoring tool" mean that the content and tool can be used by people regardless of disability.

To understand the accessibility issues relevant to authoring tool design, consider that many users may be creating content in contexts very different from your own:

- They may not be able to see, hear, move, or may not be able to process some types of information easily or at all;
- They may have difficulty reading or comprehending text;
- They may not have or be able to use a keyboard or mouse;
- They may have a text-only display, or a small screen.

Accessible design will benefit people in these different authoring scenarios and also many people who do not have a physical disability but who have similar needs. For example, someone may be working in a noisy environment and thus require an alternative representation of audio information. Similarly, someone may be working in an eyes-busy environment and thus require an audio equivalent to information they cannot view. Users of small mobile devices (with small screens, no keyboard, and no mouse) have similar functional needs as some users with disabilities.

Accessibility Information

Accessibility information is content, including information and markup, that is used to improve the accessibility of a document. Accessibility information includes, but is not limited to, equivalent alternative information .

Accessibility Problem (Also: ***Inaccessible Markup***)

Inaccessible Web content or authoring tools cannot be used by some people with disabilities. The Web Content Accessibility Guidelines [WAI-WEBCONTENT] describes how to create accessible Web content.

Accessible Authoring Practice

Practices that improve the accessibility of Web content. Both authors and tools engage in accessible authoring practices. For example, authors write clearly, structure their content, and provide navigation aids. Tools automatically generate valid markup and assist authors in providing and managing appropriate equivalent alternatives.

Alert

An alert draws the author’s attention to an event or situation. It may require a response from the author.

Alternative Information (Also: ***Equivalent Alternative***)

Content is "equivalent" to other content when both fulfill essentially the same function or purpose upon presentation to the user. Equivalent alternatives play an important role in accessible authoring practices since certain types of content may not be accessible to all users (e.g., video, images, audio, etc.). Authors are encouraged to provide text equivalents for non-text content since text may be rendered as synthesized speech for individuals who have visual or learning disabilities, as braille for individuals who are blind, or as graphical text for individuals who are deaf or do not have a disability. For more information about equivalent alternatives, please refer to the Web Content Accessibility Guidelines [WAI-WEBCONTENT] .

Attribute

This document uses the term "attribute" as used in SGML and XML ([XML]): Element types may be defined as having any number of attributes. Some attributes are integral to document accessibility (e.g., the `"alt"`, `"title"`, and `"longdesc"` attributes in HTML).

Auditory Description

An auditory description provides information about actions, body language, graphics, and scene changes in a video. Auditory descriptions are commonly used by people who are blind or have low vision, although they may also be used as a low-bandwidth equivalent on the Web. An auditory description is either a pre-recorded human voice or a synthesized voice (recorded or automatically generated in real time). The auditory description must be synchronized with the audio track of a video presentation, usually during natural pauses in the audio track.

Authoring Tool

An authoring tool is any software that is used to produce content for publishing on the Web. Authoring tools include:

- Editing tools specifically designed to produce Web content (e.g., WYSIWYG HTML and XML editors);
- Tools that offer the option of saving material in a Web format (e.g., word processors or desktop publishing packages);
- Tools that translate documents into Web formats (e.g., filters to translate desktop publishing formats to HTML);
- Tools that produce multimedia, especially where it is intended for use on the Web (e.g., video production and editing suites, SMIL authoring packages);
- Tools for site management or site publication, including tools that automatically generate Web sites dynamically from a database, on-the-fly conversion and Web site publishing tools;
- Tools for management of layout (e.g., CSS formatting tools).

Captions

Captions are essential text equivalents for movie audio. Captions consist of a text transcript of the audio track of the movie (or other video presentation) that is synchronized with the video and audio tracks. Captions are generally rendered graphically and benefit people who can see but are deaf, hard-of-hearing, or cannot hear the audio.

Conversion Tool

A conversion tool is any application or application feature (e.g., "Save as HTML") that transforms convent in one format to another format (such as a markup language).

Check for

As used in checkpoint 4.1, check for can refer to three types of checking:

1. In some instances an authoring tool will be able to check automatically. For example checking for validity (checkpoint 2.2) or testing whether an image is the only content of a link.
2. In some cases the tool will be able to "suspect" or "guess" that there is a problem, but will need to confirm with the author. For example, in making sure that a sensible reading order is preserved a tool can present a linearized version of a page to the author.
3. In some cases a tool must rely mostly on the author, and can only ask the author to check. For example, prompting the author to check whether equivalent alternatives for multimedia are appropriate. This is the minimal standard to be satisfied. Subtle, rather than extensive, prompting is more likely to be effective in encouraging the user to verify accessibility where it cannot be done automatically.

Document

A document is a series of elements that are defined by a markup language (e.g., HTML 4.0 or an XML application).

Editing View

A view provided by the authoring tool that allows editing.

Element

An element is any identifiable object within a document, for example a character, word, image, paragraph or spreadsheet cell. In [HTML40] and [XML] , an element refers to a pair of tags and their content, or an "empty" tag - one that requires no closing tag or content.

Inform

Make the author aware of an event or situation through alert , prompt , sound, flash, or other means.

Markup Language

Authors encode information using a markup language such as HTML ([HTML40]), SVG ([SVG]), or MathML ([MATHML]).

Presentation Markup

Markup language such as Cascading Style Sheets [future reference to CSS] used to encode information about the desired presentation or layout of the content. CSS, for example, can be used to control fonts and positioning. Presentation markup should not be used incorrectly to present or layout information to resemble structural content. For example, CSS or HTML should not be used incorrectly to visually layout information to resemble a list with out also using the structural markup for a list.

Prompt

A prompt is a request for user input, either information or a decision. A prompt requires author response. For example, a text equivalent entry field prominently displayed in an image insertion dialog would constitute a prompt. Prompts can be used to encourage authors to provide information needed to make content accessible (such as alternative text equivalents).

Property

A property is a piece of information about an element, for example structural information (e.g., it is item number 7 in a list, or plain text) or presentation information (e.g., that it is marked as bold, its font size is 14). In XML and HTML, properties of an element include the type of the element (e.g., `IMG` or `DL`), the values of its attributes , and information associated by means of a style sheet. In a database, properties of a particular element may include values of the entry, and acceptable data types for that entry.

Structural Markup

Markup language such as HTML used to encode information about the structural role of elements of the content. For example, headings, sections, members of a list, and components of a complex diagram can be identified using structural markup. Structural markup should not be used incorrectly to control presentation or layout. For example, HTML structural markup should not be used incorrectly to achieve an indentation visual layout effect by using the `blockquote` element. Structural markup should be used correctly to communicate the roles of the elements of the content and presentation markup should be used separately to control the presentation and layout.

Transcript

A transcript is a line by line record of sounds within an audio clip, or an audio track from a video clip. A collated text transcript for a video combines (collates) caption text with text descriptions of video information (descriptions of the actions, body language, graphics, and scene changes of the video track). Collated text transcripts are essential for individuals who are deaf-blind and rely on braille for access to movies and other content.

Transformation

A process that changes a document or object into another, equivalent, object according to a discrete set of rules. This includes conversion tools , software that allows the author to change the DTD defined for the original document to another DTD, and the ability to change the markup of lists and convert them into tables.

User Agent

An application that retrieves and renders Web content. User agents include browsers, plug-ins for a particular media type, and some assistive technologies.

View

Authoring tools may render the same content in a variety of ways; each rendering is called a view. Some authoring tools will have several different types of view, and some allow views of several documents at once. For instance, one view may show raw markup, a second may show a structured tree, a third may show markup with rendered objects while a final view shows an example of how the document may appear if it were to be rendered by a particular browser. A typical way to distinguish views in a graphic environment is to place each in a separate window.

4. Acknowledgments

Many thanks to the following people who have contributed through review and comment: Jim Allan, Denis Anson, Kitch Barnicle, Kynn Bartlett, Harvey Bingham, Judy Brewer, Carl Brown, Dick Brown, Wendy Chisholm, Rob Cumming, Daniel Dardailler, Mark Day, BK Delong, Martin Dürst, Kelly Ford, Jamie Fox, Edna French, Sylvain Galineau, Al Gilman, Eric Hansen, Phill Jenkins, Len Kasday, Brian Kelly, Marja-Riitta Koivunen, Sho Kuwamoto, Jaap van Lelieveld, William Loughborough, Karen McCall, Charles Oppermann, Dave Pawson, Dave Poehlman, Bruce Roberts, Chris Ridpath, Gregory Rosmaita, Janina Sajka, John Slatin, Jim Thatcher, Irène Vatton, Gregg Vanderheiden, Pawan Vora, Jason White, and Lauren Wood.

5. References

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

[HTML40]
"HTML 4.0 Recommendation," D. Raggett, A. Le Hors, and I. Jacobs, eds., 17 December 1997, revised 24 April 1998. This HTML 4.0 Recommendation is http://www.w3.org/TR/1998/REC-html40-19980424. The latest version of HTML 4.0 is available at http://www.w3.org/TR/REC-html40.

[MATHML]
"Mathematical Markup Language," P. Ion and R. Miner, eds., 7 April 1998, revised 7 July 1999. This MathML 1.0 Recommendation is http://www.w3.org/TR/1998/REC-MathML-19990707. The latest version of MathML 1.0 is available at http://www.w3.org/TR/REC-MathML.

[SVG]
"Scalable Vector Graphics (SVG) 1.0 Specification" (Working Draft), J. Ferraiolo, ed. The latest version of the SVG specification is available at http://www.w3.org/TR/SVG

[WAI-AUTOOLS-CHECKLIST]
An appendix to this document lists all of the checkpoints, sorted by priority. The checklist is available in either tabular form (at http://www.w3.org/WAI/AU/PR-WAI-AUTOOLS-19991218/checkpoint-table) or list form (at http://www.w3.org/WAI/AU/PR-WAI-AUTOOLS-19991218/checkpoint-list).

[WAI-AUTOOLS-TECHS]
"Techniques for Authoring Tool Accessibility," J. Treviranus, J. Richards, I. Jacobs, and C. McCathieNevile eds. The latest version is available at http://www.w3.org/WAI/AU/WAI-AUTOOLS-TECHS.

[WAI-WEBCONTENT]
"Web Content Accessibility Guidelines 1.0," W. Chisholm, G. Vanderheiden, and I. Jacobs, eds., 5 May 1999. This Recommendation is http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505. The latest version of the Web Content Accessibility Guidelines 1.0" is available at http://www.w3.org/TR/WAI-WEBCONTENT/.

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