

National Center on Accessible Information Technology in Education

Publication #10

National Center on
Accessible Information
Technology in Education
(AccessIT)

Regional Disability
and Business Technical
Assistance Centers

National Institute on
Disability and
Rehabilitation Research

Further information about
AccessIT at the University of
Washington is available:
(866) 968-2223 (voice),
(866) 866-0162 (TTY),
email: accessit@u.washington.edu
or on the website:
www.washington.edu/accessit.

Information, technical
assistance, and training is also
available from the Regional Disability
and Business Technical Assistance
Centers,
(800) 949-4232 (V/TTY),
or on the Web: www.adata.org.

AccessIT

Webmaster Accessibility FAQ

AccessIT (<http://www.washington.edu/accessit>) and the ten regional ADA & IT Technical Assistance Centers (<http://www.adata.org/centers.htm>) provide trainings, workshops, presentations, and technical assistance to educational entities on how to make websites and other information technologies accessible to people with disabilities. Both organizations have worked extensively with K-12 and higher education webmasters, and have pooled their clients' most frequently asked questions into this publication. Short answers are provided along with information on how to access more detailed responses in the AccessIT Knowledge Base.

Web Accessibility Standards

Q: Which set of web accessibility standards or guidelines should I comply with?

A: Many states and educational entities have adopted Section 508 standards (a minimum standard for federal agencies) and/or the more comprehensive Web Content Accessibility Guidelines from the World Wide Consortium (W3C). In some cases, they have developed their own standards or guidelines. For more information, see *Which set of web accessibility standards or guidelines should I comply with?* (<http://www.washington.edu/accessit/articles?255>).

Alternate Text for Graphics

Q: What is appropriate alternate text for purely decorative images?

A: If an image is purely decorative and contains no informative content, use an empty string as the "alt" attribute, i.e., alt="". This technique is widely supported by screen readers, which respond by ignoring the image.

Q: How long can an "alt" attribute be?

A: The HTML specification does not define a maximum length for "alt" attributes. However, their intended function is to provide short, efficient alternate text so that users who cannot see an image can access the content presented in the image. Browsers' and certain assistive technologies' current method of rendering "alt" attributes reinforces the need to keep them short. More information is available in *How long can an "alt" attribute be?* (<http://www.washington.edu/accessit/articles?257>).

Q: What is the current recommendation for providing long descriptions for complex graphics?

A: In HTML, the "longdesc" attribute was specifically designed to describe complex graphics. Although historically this attribute has not been well supported by assistive technologies, current and recent versions of most screen reader applications do support it. For additional information, see *What is the current recommendation for providing long descriptions for complex graphics?* (<http://www.washington.edu/accessit/articles?258>).

Navigation

Q: What is a “skip navigation” link?

A: A "skip navigation" link is a same-page link that allows screen reader users and users navigating by keyboard to skip past redundant navigation systems and jump efficiently to the main content of the page, just as a sighted user can do at a glance. For more information, see *What is a “skip navigation” link?*

(<http://www.washington.edu/accessit/articles?259>).

Q: A same-page “skip navigation” link messes up my page layout and is potentially confusing to sighted users. Is it OK if the link is invisible?

A: "Skip navigation" links, although very useful for nonsighted users, are also useful for sighted users who navigate by keyboard rather than mouse, including people with upper-body mobility impairments. For these users, the "skip navigation" link reduces the number of keystrokes required to access another link in the main content of the document. However, the "skip navigation" link must be visible in order for these users to know that it is available. One solution is to make the link invisible (using Cascading Style Sheets) until a keyboard user tabs to it, at which point it becomes visible. For more information see *Is it OK to make “skip navigation” links invisible?*

(<http://www.washington.edu/accessit/articles?260>).

Q: In addition to including a skip navigation link, how else can I facilitate page navigation?

A: All web documents should be divided into short sections for readability, and should be organized using a clear hierarchy of HTML headings and subheadings (i.e., <h1>, <h2>, etc.) This helps all users to understand the document contents, but it especially helps users of screen readers. Most screen readers include functionality that allows users to jump between headings with a single keystroke, so blind users can effectively scan the document looking for sections that particularly interest them, just as sighted users typically do. For more information see *How can I make web page navigation accessible?*

(<http://www.washington.edu/accessit/articles?269>).

Q: With Cascading Style Sheets (CSS), navigational menus can appear visually at the top of a document, despite their being at the bottom of the document in the HTML source code. Does coding in this way eliminate the need for a “skip navigation” link?

A: This technique eliminates the need for a "skip navigation" link, but it creates the opposite problem: the need for an efficient means of jumping quickly to the navigation system. It also is likely to cause the tab order of the document to be different from the logical intuitive order that a sighted

keyboard user would expect. For more information, including possible solutions to these problems, see *Does CSS positioning eliminate the need for a “skip navigation” link?* (<http://www.washington.edu/accessit/articles?261>).

Javascript / Dynamic Menus

Q: Is it possible to develop an accessible dynamic menu?

A: A variety of experiments have focused on developing accessible dynamic menu systems. None of the models is fully accessible to all users; most require that the user have Javascript enabled, and none can be activated by handheld computer users who select menu items by clicking with a stylus. Therefore, the only way to ensure that all users have access to secondary menus is to populate the primary menu items with links that lead users to a separate page or pages where they can access the secondary menu items. For more information, see *Is it possible to develop an accessible dynamic menu?*

(<http://www.washington.edu/accessit/articles?204>).

Q: Is it reasonable to expect that all users today are using Javascript-enabled browsers?

A: Javascript is well supported but not ubiquitous. Javascript is not accessible to users with certain assistive technologies, older browsers, pocket versions of browsers, text-based browsers such as Lynx or Emacs/W3, or newer browsers that have scripts disabled to eliminate possible security risks. For additional information, see *Are all web users today using Javascript-enabled browsers?*

(<http://www.washington.edu/accessit/articles?262>).

Colors and Color Blindness

Q: There are many different types of color blindness. How can I ensure that color blind users can access my site?

A: A variety of tools are available to help with evaluating the accessibility of color combinations, including a tool that allows users to sample their websites as they might be viewed by someone with any of three types of color blindness, and tools that measure whether particular color combinations have sufficient color and brightness contrast. A general rule is that any information that is communicated with color (e.g., "fields with red labels are required") should also be communicated with some mechanism other than color (e.g., bold text, marked with an asterisk). This way, a user who is unable to perceive color can still access the information. More information, including links to assessment tools, is available in *How do my choice and use of color affect the accessibility of my website?*

(<http://www.washington.edu/accessit/articles?222>).

Accessible Tables

Q: What is wrong with using HTML tables for layout?

A: HTML tables were originally intended to be used for presenting tabular data, not for layout. The W3C discourages the use of tables for layout because they are striving for a web in which content and structure are completely separate from presentation. In the W3C's world view, CSS is the vehicle by which presentation and layout are defined. Browser support for CSS has improved significantly in recent versions, and growing numbers of sites are migrating over to CSS entirely for positioning. For more information, see *What is wrong with using HTML tables for layout?* (<http://www.washington.edu/accessit/articles?263>).

Q: How do I make data tables accessible?

A: Data tables are tables used to represent actual tabular data, with rows and columns of related information. The technique for making data tables accessible depends on the complexity of the table but may involve any of a combination of TH and CAPTION elements and/or "summary", "scope", "id", and "headers" attributes. For specific coding techniques and links to additional resources, see *How do I make data tables accessible?* (<http://www.washington.edu/accessit/articles?22>).

Accessible Forms

Q: To make a form accessible, is it enough just to be sure that form labels appear immediately to the right of the fields they represent?

A: The key accessibility problem with HTML forms is that screen readers typically have to guess which label goes with which field. If a form consists solely of text fields and labels always appear immediately to the right of the fields they represent, a web author can be reasonably confident that screen readers will read this correctly. However, confidence goes down as forms become more complex, and accessible markup must be used in order to prevent screen readers from having to guess. Additional information, including specific techniques, is available in *How can I develop accessible web-based forms?* (<http://www.washington.edu/accessit/articles?159>).

Q: Are "QuickLinks" dropdown lists accessible?

A: A growing number of websites use a single drop-down list of links that takes the user immediately to a selected link, without requiring that the user press a Submit button. For people who can't use a mouse because of a visual impairment or physical disability, these form fields present problems since they automatically select the first field that a user arrows to and activate that link. The user is therefore unable

to scroll down any farther in the list. The solution is to always provide a Submit button and to avoid using onSelect or onChange events in dropdown lists. For more information, see *Are "QuickLinks" dropdown lists accessible?* (<http://www.washington.edu/accessit/articles?264>).

Accessibility of Adobe PDF

Q: Is PDF Accessible?

A: With the release of Acrobat 5.0, Adobe unveiled a new type of Portable Document Format (PDF) called "tagged PDF." It is the only type of PDF that is optimized for accessibility, including support for alternate text for graphics and an underlying document structure that resembles HTML. Currently, only a few PDF creation tools allow authors to create tagged PDF. Among those that do, few authors are taking advantage of its accessibility features in practice. A growing number of assistive technologies are supporting tagged PDF, but none do so as fully as they support HTML. For more information, see *Is PDF accessible?* (<http://www.washington.edu/accessit/articles?2>).

Accessibility of Macromedia Flash

Q: Can I build an accessible online application using Macromedia Flash?

A: With the release of Flash MX, Macromedia had made significant improvements to its product's ability to produce accessible content. Using Flash's Accessibility Panel, developers can assign text to interface elements so that screen reader users can access them. In some Flash applications, developers can explicitly define a tab order for keyboard users. Flash accessibility depends on authors to create accessible content and on assistive technologies to support the accessibility features. Only a few currently do. See also *Is Flash content accessible?* (<http://www.washington.edu/accessit/articles?23>).

Accessibility of Frames

Q: Are frames accessible?

A: Frames are not in and of themselves inaccessible. Keyboard users and assistive technology users are able to navigate among the various frames that comprise a web page. In order to facilitate navigation, each frame must be assigned a title (using the "title" attribute) that clearly communicates the function of the frame (e.g., "Navigation" or "Content"). Also, screen reader users should be provided with clear instructions on how and where to locate content within the frameset. See also *Are frames accessible?* (<http://www.washington.edu/accessit/articles?21>).



This publication is available in alternate formats upon request and is one of a series on accessible electronic and information technology in education published by the

National Center on Accessible Information Technology in Education (AccessIT),
in collaboration with the **Regional Disability and Business Technical Assistance Centers,**
projects funded by the **National Institute on Disability and Rehabilitation Research**

Accessibility Evaluation and Repair Tools

Q: Which web accessibility software tool should I use, if any?

A: A growing number of software products allow web developers to evaluate the accessibility of their web pages and sites. These tools can be useful in identifying web accessibility problems much more efficiently than could be done manually. However, many aspects of web accessibility are inherently subjective, and can not be assessed automatically. All of these products prompt users regarding accessibility issues that require human judgment, which can help designers to learn about web accessibility. A key in identifying which product is best is determining how clearly a product explains and coaches users on accessibility issues. Many other variables must be considered as well, as documented in *How can I select a web accessibility software tool?* (<http://www.washington.edu/accessit/articles?244>)

Resources

Q: Where can I learn more about web accessibility?

A: The AccessIT Knowledge Base is a continually growing and searchable online resource that includes answers to common questions regarding accessible information technology in education. Also, the federally funded network of regional ADA & IT Centers provides resources, training, and technical assistance regarding information technology accessibility in education, including web accessibility. The AccessIT Knowledge Base is at <http://www.washington.edu/accessit>. To locate your regional center, see the *ADA & IT Technical Assistance Centers web site* (<http://www.adata.org/centers.htm>) or call (800) 949-4232.



Copyright © 2006 by University of Washington. Permission is granted to copy these materials for educational, noncommercial purposes provided the source is acknowledged. This product was funded by the National Institute on Disability and Rehabilitation Research of the U.S. Department of Education (grant #H133D010306). However, the contents do not necessarily represent the policy of the Department of Education, and you should not assume their endorsement.

National Center on Accessible Electronic
and Information Technology in Education
University of Washington
Box 357920
Seattle, WA 98195

