

IWAS Made Easy

A Condensed Version of The Illinois Web Accessibility Standards

In February 2002, the Illinois Technology Office established the Illinois Web Accessibility Standards to aid web designers in the creation of sites that can be easily interpreted by people with various disabilities. What follows is a concise edition of those helpful hints that aims to serve web designers at all levels of expertise. Notations to the right of each heading refer to the standard's placement in Mike Scott's original document, "Illinois Web Accessibility Standards, Version 1.2." For additional guidance, a version of IWAS can be found at <http://www.illinois.gov/iwas/>.

Create standard hypertext code which is validated and tested. (1.1, 1.2, 21.1)

- Indicate the programming language you are using by starting your code with a document type declaration such as: `<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">`.
- Use the W3C CSS Validation Service (<http://jigsaw.w3.org/css-validator>) to check your CSS code, if you use style sheets.
- Identify section heading, paragraphs, lists, quotes, etc. using the appropriate tags, instead of relying on formatting commands to distinguish these elements. For example, use `<h1>` tags to identify top-level headings, rather than simply applying font size or bold formatting commands.
- Using validators mentioned in other help documents, test for accessibility.

Create page layouts in an accessible manner. (1.3, 17.1, 17.2)

- Use style sheets for formatting whenever possible, see the W3C's Cascading Style Sheets (CSS) site (<http://www.w3.org/Style/CSS/>) for specifics.
- Check the reading order by following the order the table cells and elements appear in code.

Make text easy for both people and screen readers to understand. (2.1, 2.2, 2.3, 2.4, 18.1)

- Identify the primary language of text using the lang attribute on the `<html>` element.
- Identify foreign words by using the lang attribute on `` or other elements.
- Use written words or graphics instead of emoticons such as ":-)" or "->".
- Convey information with text, not images, as often as possible.
- Use relative measurements to set font sizes, or do not set them at all.
- Use clear, concise language that the average viewer of your site will understand.

Provide alternate text for all images, and each area in client-side image maps. (4.1, 4.2 5.1)

- Null images that do not contain important content by using the command `alt= ""`
- Include a full textual description that clearly identifies each image or link's purpose or destination.

Make images and text signifying links as large and unambiguous as possible. (9.1, 9.3)

- Avoid general text such as “More,” “Click here,” “Enter,” “Next,” and “Previous.”
- Use either standard or enlarged fonts for textual links.
- Avoid textual links shorter than four characters in length.
- Leave adequate space between individual links.

Do not convey information with color alone, and use contrasting foreground and background colors. (3.1, 3.2)

- When using color as an indicator, use a non-colored-based indicator (asterisks, italics, etc.) as well.
- Use light colored text with dark backgrounds, or dark colored text with light backgrounds.

Design simple data tables with headings for all columns and rows. (11.1, 11.2)

- Try to divide complex data tables into multiple tables.
- Use tools such as headers, axis, scope, <col>, and <colgroup> to make cells easily relatable to their headers.
- Identify cells using <th> (table header) or <td> (table data) elements.
- Use scope= “col” (for column headers) or scope= “row” (for row headers).

Label fill-in forms completely and consider keyboard-only users carefully. (10.1, 10.2, 10.3, 10.4)

- Label all form fields using the <label for= “”> tag.
- Position labels above or to the left of their corresponding text boxes, and to the right of checkboxes and radio buttons.
- Make sure that form fields appear in logical order in the HTML code, so the tab key can be used to fill in the form.
- Provide instructions before form fields and, if possible, within field labels.

Provide meaningful titles for all frames and only use them for non-design purposes. (12.1, 12.2)

- Use names for frames and pages that clearly identify their functions for users.
- Unless the use of a frame is necessary, do not use one.

Provide accessible alternatives for downloadable documents, applets, and plug-ins. (14.1, 14.2, 15.1, 15.2, 19.1, 20.1)

- When a link is present, always provide hypertext or text versions for downloadable documents.
- If a downloadable document cannot be provided in an accessible electronic format, provide contact information so that a person can request an alternate format.
- When links are provided to inaccessible applets, provide a link to an equivalent accessible version.
- When using applets and plug-ins, consult with manufacturers to determine their accessibility, and provide links to any necessary instructions or software.
- Use alternate accessible content only as a necessary substitute for other IWAS guidelines.

Make sure that significant interactions can be performed with both keyboard and mouse. (13.1)

- Mouse controls and keyboard controls should be the same, and trigger the same actions.

Allow users to access content whether their web browsers support scripts or not. (13.2)

- When using client-side scripting, ensure all information and functionality delivered with client-side scripts is accessible for browsers, such as screen readers, which do not support them.

Allow users to skip past repetitive navigation links. (9.2)

- Place navigation links at the end of a document, or provide a way to skip them.

Notify users of actions that will open a new window and time limits. (16.1, 16.3)

- Avoid actions that will cause new windows to open automatically.
- Identify links that will open new windows in the link text or title attributes.
- Inform people of time limitations, and make the amount of time reasonable.

Avoid using server-side image maps. (5.2)

- Substitute client-side image maps, they are easier to encode and to use.
- When server-side image maps are necessary, provide textual links that detail the functions and destinations in an image map.

Avoid an excessive need for horizontal scrolling. (17.3)

- Design pages with widths that can be resized by other browsers.
- Use relative widths for tables and frames.
- Position important content on the left area of the page.

Avoid flickering, blinking, and unnecessary animation. (8.1)

- Avoid using animations, and do not use flickering or blinking between 2 and 55 Hz.

Design pages that will not automatically refresh or reload themselves, or redirect users to a different document. (16.2)

- Do not use HTTP-EQUIV= “refresh”.
- When working with pages that will reload their content, provide a link that will allow users to refresh the pages at their discretion.

Provide all multimedia content in multiple formats. (6.1, 6.2, 7.1, 7.2)

- Provide transcripts of audio containing speech.
- Synchronize captions with sounds using appropriate technologies like Synchronized Multimedia Integration Language (SMIL).
- When important information is presented with sound, use visual indicators to explain the information as well.
- Consider using video content carefully.