



OSU Minimum Web Accessibility Standards (rev. 06/30/2004)

<http://www.wac.ohio-state.edu/standards/>



Best Practices.

Included in this standard: [Page Headers](#) || [Relative Sizing](#) || [Code](#).

Although not required by the OSU Minimum Web Accessibility Standards, following these guidelines will add additional functionality and usability for all users, including users of assistive technology.

Best Practices -- Page Header Information.

Do	Don't
Give every page a descriptive title that indicates the page contents.	Use cryptic or general page titles (e.g. "cr751s1Ap4" or "home page")

Why: Titles are used in window title bars, bookmark lists, and results from searches. The title appears in the window of the browser and is also used in tracking the HISTORY (BACK button) of where the user has been. It can be particularly helpful when a user has multiple windows open, if it uniquely identifies each page.

How: The title should ideally be less than 64 characters in length. That is, many applications will display document titles in window titles, menus, etc where there is only limited room. Whilst there is no limit on the length of a title (as it may be automatically generated from other data), it may be truncated if long.

Do	Don't
Identify the language of the document and any changes in that language.	Assume your users' language is English.

Why: The language declaration may: assist search engines, assist speech synthesizers, help a browser choose special characters, help a browser make decisions about hyphenation and spacing, and assist spelling and grammar checkers.

How: The LANG attribute is part of the HTML element and uses the two-letter primary language-code. Example: `<html lang="en">` Some other codes include: FR (French), de (German), it (Italian), es (Spanish), zh (Chinese), and ja (Japanese). You can specify a change in language using the DIV element: `<div`

lang="fr">Bon Jour!</div>

Do	Don't
Include meta data (keywords and summary) for better searching.	Use the same keywords and summary for every page in your site.

Why: Meta-data helps search engines find related pages faster and insures better, more accurate searching. The OSU search engine uses meta keywords and descriptions to prioritize hits in search results.

For more on the uses of META tags, visit the [META Tag Tutorial from webdeveloper.com](#).

How: Place META tags between the HEAD tags in your HTML documents.

For page summary/descriptions: <meta name="description" content="description of page goes here.">

For keywords: <meta name="keywords" content="searchterm1, searchterm2, searchterm3">

Get more help with page header information, visit WAI's ["The Global Structure of an HTML Document"](#).

[Return to index.](#)

Best Practices -- Relative Sizing.

Do	Don't
Use relatively size font definitions (e.g., font-size="medium" or font-size="1em").	Use definite font sizes (e.g., font-size = "14 point").

Why: Relatively sized fonts transform gracefully regardless of the default font size the user sets the browser to display. Thus, a user who sets the default size to 24 points still sees your headers as larger than the body of the page. If you set a specific font size, your selections may override the user definitions making your page unreadable to those with visual impairments.

How: Create a style sheet that defines font sizes using a relative definition. Recommended: use the "M" (written as "em"); a typesetter value based on the largest letter in the alphabet (along with "W"). The WAC site sets our paragraph text to 1em and our largest header to 2ems with a range of sizes in between to denote "levels" of organization.

Do	Don't
Use relatively sized tables (e.g. width="85%").	Format layout tables to a fit a specific resolution (e.g., width="800").

Why: Setting tables to fit lower resolutions means wasted space on user screens. Setting tables to fit higher resolutions causes users to scroll left-right as well as up-down. Using relatively sized tables allows the table to transform gracefully regardless of screen resolution or window size.

How: Express all table widths as a percentage rather than a pixel size. Avoid specifying a table height -- allow content to determine height.

Get more help with relative sizing, visit ["CSS Design: Size Matters,"](#) by Todd Fahrner.

[Return to index.](#)

Best Practices -- Code.

Do	Don't
Include a DOCTYPE statement on every page and validate your code to the specified code grammar.	Include the wrong DOCTYPE or use elements and attributes not supported by the grammar you specify.

Why: Writing the "cleanest" code possible should ensure your pages will look best across the widest range of web display technology. Adhering to a specific grammar helps insure browsers will interpret your code correctly.

How: Choose a code standard (HTML 4.01 strict, HTML 4.01 transitional, XHTML 1.0 transitional, etc.) and stick to it. Define your preferences in your web editor to follow the rules of that standard. Identify that standard on each page using the correct DOCTYPE statement. Validate your code to insure you are following the rules of standard you have chosen.

Do	Don't
Use ABBR and ACRONYM to identify abbreviated text.	Rely on ABBR and ACRONYM to describe abbreviations (also include the description the first time you use the abbreviation or acronym).

Why: The ABBR and ACRONYM elements help screen-readers interpret abbreviations and acronyms. They may also provide browser users with additional information. However, they should not be relied upon to convey the explanation of the shortened term.

How: ABBR is used for standard abbreviations that are intended to be spoken as abbreviations (e.g., I.R.S.) or as the full term (e.g., Univ., spoken as "university"). ACRONYM is used when the abbreviation is spoken as a word (e.g. PETA, spoken as "Peetah" rather than as individual letters: P, E, T, A.).



Do	Don't
Use style sheets for formatting.	Use deprecated code elements, including FONT, CENTER, B, I, and U.

Why: Using style sheets for formatting helps insure that users can override your style choices when necessary and customize colors, font size, and resolution as desired. Deprecated code elements, like the FONT tag, are more likely to block users from overriding your formatting selections.

How: Rather than using the FONT tag, redefine how HTML elements like P, H1, UL and others appear on your pages using Cascading Style Sheets. Then, turn-off style sheets in your browser and test your pages. Can you still understand all the content on the page? If so, you are using style sheets correctly. You might also consider offering your users a selection of style sheets to render your pages in larger text, higher contrast, or with reduced graphics.



Do	Don't
Use BLOCKQUOTE to identify quoted text.	Use BLOCKQUOTE for indenting.

Why: The BLOCKQUOTE element has special functionality in screen readers and text surrounded by BLOCKQUOTE is often read in a different voice or set-off with "quote" and "end quote." Using BLOCKQUOTE incorrectly can be confusing.

How: Identify quoted text by enclosing it in the BLOCKQUOTE tag. Do not use the tag for indenting. Instead, create an "indent" style in your style sheet.



Get more help with using code correctly, review the [Web Design Group's "HTML 4.0 Deprecated Features"](#).

OSU Web Accessibility Center -- 2004. All documents in this guide may be reproduced and distributed in print or electronic format only if offered at no cost to recipients. Visit the WAC online: www.wac.ohio-state.edu. Email the WAC: webaccess@osu.edu.