National Science Digital Library

Web Development and Accessibility Best Practices

http://www.nsdl.org/

Revision Date: 10/20/2008

Table of Contents

Introduction

This document contains the general web development and accessibility best practices for the creation of new web sites and the revision of existing web sites for NSDL.org. Accessible design is important to provide equal access to users with disabilities, older web browsing devices and to promote forward compatibility with future technologies. Partners are not required to follow these guidelines but are encouraged to keep these concepts in mind when creating their websites in order to provide access to their digital content to the widest possible audience.

These guidelines are based on Section 508 of the Federal Rehabilitation Act, the W3C Web Content Accessibility Guidelines 1.0, and the XHTML 1.0 Specification.

Page Structure

Use proper semantic XHTML markup to define the web page structure and use Cascading Style Sheets (CSS) to control the visual layout and presentation. Make sure the web page passes and XHTML validator to help ensure accessibility and greater device compatibility before duplicating your page templates. Declare the intended Document Type Definition (DTD) to ensure that the web browser knows how to interpret the page markup. For NSDL.org web pages, use the XHTML transitional DTD.

Page Titles

Page titles should clearly identify the web document. Pages that are a part of the main NSDL.org web site should have titles that adhere to the following structure:

```
NSDL.org – <PAGE TITLE> – The National Science Digital Library
```

Since NSDL.org uses centrally located header and footer files, programmatic mechanisms have been included to enable distinct titles and descriptions for each page. A default page title and META description are included in the header templates. To define a custom page title and META description for pages on NSDL.org, assign values to the following PHP variables when calling the page templates.

To assign a page title: `$Page_Title = "";`
To assign a Meta description: `$Page_Desc = "";`

Frames

If using frames in your web site, include descriptive text for the title attribute of each frame and include identifying text within each frame page that briefly describes the frame's content. Provide a <noframes> equivalent for users who cannot view frames.

```
<frameset cols="30%,60%">
<frame src="navlinks.html" name="navlinks" title="Navigational Links
```

```
Frame">
<frame src="main.html" name="contents_page" title="Contents Frame">
<noframes>
     <p>Here is the <a href="main-noframes.html">non-frame based
version of the document.</a></p>
     </noframes>
</frame>
```

Stylesheets

Make sure your web content is well-organized and readable without requiring an associated stylesheet. Use external stylesheets to ensure that you do not override any user-defined stylesheets.

Example source code:

```
<link rel=stylesheet type="text/css" href="style.css" />
```

Links

Give your links meaningful labels that describe their destination and assure that they make sense when read out of context. If the page is printable, links should include the destination URL.

Poor link label: `<a href="homepage.html">click here</a>`
Good link label: `<a href="homepage.html">Return to the homepage</a>`

CSS code to enable printable links

```
a:link:after, a:visited:after {
   content: " (" attr(href) ") ";
 }
a[href^="/"]:after {
  content: " (http://www.nsdl.org" attr(href) ") ";
}
```

Multimedia and PDF files linked from the web site should explicitly include the file type and size directly next to the linked file.

```
<a href="annual_report_2004.pdf">2004 Annual Report</a> [1.4MB] [.pdf]
```

Tables

Try to avoid using tables for layout. When possible, reserve tables for actual tabular data and use tags and CSS stylesheets for visual layout. When using tables, be sure to properly label your cells as header cells or data cells with the <th> and <td> elements. Apply scope="col" or scope="row" to your table headers to associate the corresponding rows and columns as related. Include the <caption> tag to associate a title with the table and a SUMMARY attribute to describe the contents of the table. Most importantly,

ensure that the table content is ordered correctly when the table is linearized and that the table content makes sense when read by a screen reader.

Forms

Accessibility problems can exist when a form element is separated from its associated label. Avoid placing input fields and labels in separate table cells. While the label and input field may be visually adjacent, a screen reader may not interpret the layout accurately. Instead, markup forms with HTML form tags and style them with CSS stylesheets. Use <label> tags and associate form input fields to the corresponding label text using the label's FOR attribute and the input field's ID attribute. Group related form sections with the <fieldset> and <optgroup> tags.

Any online form that collects personal information must be linked to a privacy policy to describe how the information collected will be used. Ensure that each form checks any uploaded content for acceptable file types and validates input fields for reasonable and appropriate submissions to guard against potential exploits. Form pages should include manual contact alternatives such as an email address, telephone number or mailing address, in case the form cannot be successfully submitted. A link to a contact page that contains this information is sufficient.

Navigation

Use skip-links to allow screen readers to easily skip over common navigational elements and quickly access the page content. Skip-navigation links can either be visible to regular users or only viewable by text-based browsers or screen readers by attaching the skip-link to a small, transparent image or hidden with CSS. Assure that your site is navigable using only a keyboard and that any page elements that use mouse events provide another method of accessing the content.

Fonts

Define font sizes and line spacing using relative units, such as ems or percentages rather than pixels and points, so users can resize font size as needed for legibility. For reference, one em is equal to the point size of the font. For 12 point text, one em equals 12 points. Use sans-serif fonts that are widely available and designed for screen legibility, such as Verdana, Georgia, Trebuchet, and Tahoma. Do not use Times New Roman or other serif fonts for web sites. Define font attributes using CSS and not with the deprecated <font> tag.

Alternative Text

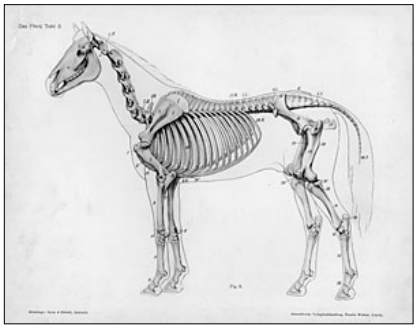
Be sure to give all non-text elements an equivalent and descriptive alternate text. Non-text elements include images, graphical representations of text, image map regions, animations, applets, frames, scripts, images used as list bullets, spacers, graphical

buttons, sounds, and stand-alone audio and video files that are used to convey information. Non-text elements requiring alternate text descriptions are elements that provide information required to understand page content or those used for navigation. Any image or applet that contains necessary information must include descriptive information in the form of ALT attribute, LONGDESC attribute, d-link, caption, or transcript.

ALT and Title Attributes

ALT attributes are read by screen-readers and appear when a non-text element is unavailable from the server or when the user chooses to "turn-off" or "hide" images in his/her browser. TITLE attributes populate the tooltip when a user hovers over a non-text element with the mouse and provides additional, non-essential information about the image.

ALT and TITLE attributes should communicate the same information as the image. When choosing alternate texts, make sure that the text actually provides useful information to anyone not seeing the image, and that it makes sense in context. ALT attributes should describe the meaning of the image rather than what it looks like.



Example of a useless ALT attribute:

```
<img src="skeleton.gif" alt="skeleton image"
width="61" height="34"/>
```

Example of a meaningful ALT attribute:

```
<img src="skeleton.gif" alt="left lateral
view of a horse skeleton" width="61"
height="34"/>
```

Simple images which provide content to the page should have a succinct ALT attribute to describe the graphic. Use as little text as necessary to accurately describe the image and use no more than 100 characters. Images with descriptive text should use the same text shown in the graphic for the ALT attributes.


```
<img src="nsf_funded.gif" alt="Funded by the National Science
Foundation" height="55" width="176" />
```

What is the image's purpose on the webpage? A navigational image should have an ALT attribute that describes the image's navigational intent. ALT descriptions of linked images should describe the link's destination.

Poor ALT attribute for a navigational item:

```
<img src="nextArrow.gif" alt="arrow" width=”24” height=”10”/>
```

Good ALT attribute for a navigational item:

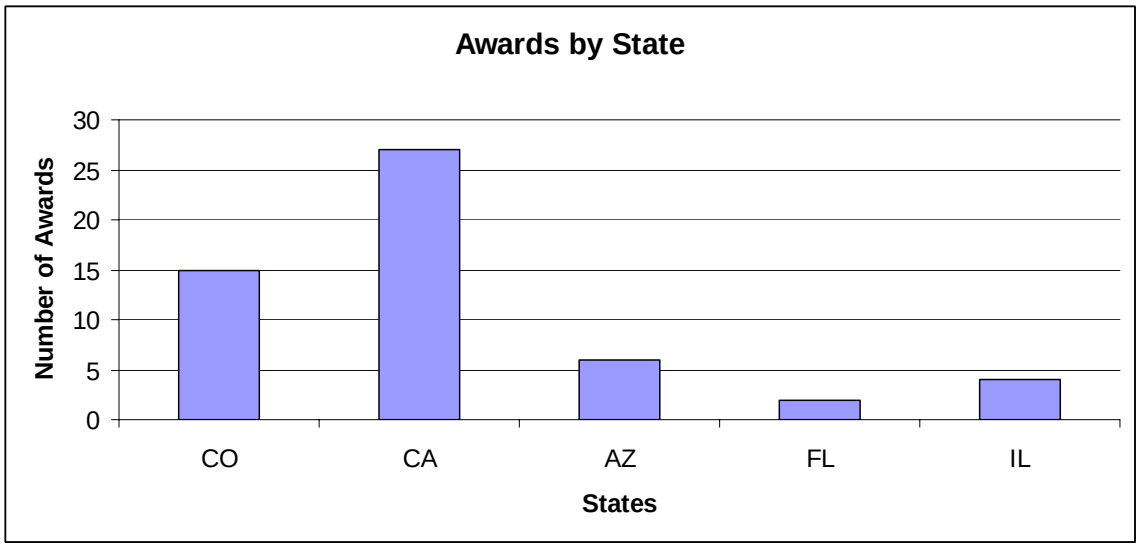
```
<img src="nextArrow.gif" alt="Next search results page" width=”24”
height=”10”/>
```

Images that do not provide content, such as spacers, dividers and borders, should use an empty string for the alternative description. Do not omit the ALT attribute because some screen readers will read the image's file name if an ALT attribute is not provided.

Example ALT attribute for a spacer:

```
<IMG src="transparent.gif" alt=""/>
```

If an image conveys complex information, such as a graph, and cannot be sufficiently described in an ALT or title attribute, use a brief ALT attribute and link the image to a longer description with an extended explanation using a LONGDESC tag or provide a caption immediately below the image. Since the LONGDESC attribute is not fully supported at this time, include a d-link when using a LONGDESC attribute. To make a d-link, place a single "d" near the image and link it to a separate page that contains a full description of the image. If using a d-link, be certain to provide a return link to the original page.



Caption: Graph 1.4 shows the number of awards per state: Colorado=15; California=27; Arizona=6; Florid =2; Illinois =4.

Code example for caption usage

```
<p class="caption">
<img src="graph.gif" alt="Graph 1.4, refer to caption for description."
width="300" height="154"><br>
Caption: Graph 1.4 shows the number of awards per state: Colorado=15;
California=27; Arizona=6; Florida=2; Illinois=4.</p>
```

Code example for longdesc and d-link usage

```
<img src="graph.gif" alt="graph of awards by state"
longdesc=”http://www.nsdl.org/awardsbystate.html” title="graph of
awards by state"><a href=”
http://www.nsdl.org/awardsbystate.html”>d</a>
```

Note: Images used to convey complex information should be delivered to the web developer with ALT attribute and long description text. Developers should not be responsible for creating this text.

Image Maps

If using an image map, use client-side image maps except where regions cannot be defined with an available geometric shape. For client-side image maps, identify each hotspot with a corresponding ALT attribute for each <area> tag in order to identify the link’s destination and purpose to the user. Provide redundant links for each active region of a server-side image map.

Multimedia Content

Images

Any images used on NSDL.org should be optimized for quality and filesize. Include the WIDTH and HEIGHT attribute in all <img> tags to help page load time. Do not use the WIDTH and HEIGHT attribute to change the image dimensions. If a different size of the same image is needed, create additional images with the necessary dimensions. Save line art or simple images as GIFs and photos or complex images as JPEGs.

Video

Video should be captioned for the hearing impaired and descriptive audio should be included for the visually impaired. The audio portion of a multimedia presentation should be captioned and synchronized with the video portion so the user reading the captions can watch the speaker and associate relevant body language with the speech. Live video and webcasts should be captioned in real-time. Ensure that the user can start and stop the movie.

Audio

If an audio file does not have an accompanying video, a transcript of the audio should be available either within the same page or via a d-link. For best results, script a presentation prior to recording it.

Simulations

Simulations should be accompanied by alternate text descriptions or other meaningful equivalent. Alternate, low-bandwidth versions should be provided for any interactive content and Flash movies. When creating Flash movies include captions and use the built-in Flash accessibility features. Ensure that the user can start and stop the simulation.

Plug-ins and Scripts

If a page requires that an applet or plug-in be installed before the user can access the page content, the page should include a link to the required applet or plug-in.

Pages that require scripting languages to expose content or change page elements should include descriptive text to inform the user of the script's intent. When possible, ensure that page content is still accessible without scripts if scripting is disabled on the user's browser by including a <noscript> equivalent.

XHTML treats the < and & characters as markup. Since these characters are used in JavaScript code, place any inline JavaScript code within a CDATA section to ensure

compatibility with XHTML. Any characters within a CDATA section will not be processed as XHTML markup. The CDATA section should be commented out with a JavaScript single-line comment so the JavaScript interpreter does not interpret the CDATA markup as JavaScript.

```
<script language="JavaScript" type="text/JavaScript">
//<![CDATA[

alert("<Text & more text>");

//]]>
</script>
```

If possible, use external JavaScript code rather than inline JavaScript code. External JavaScript code is compatible with XHTML and does not need the CDATA markup.

Timed Responses

If a page requires a timed response from the user, provide sufficient time for the user to respond. If a response has not been received prior to the specified amount of time, alert the user so they can request additional time before timing out the page or changing page content.

Screen Flicker

Flickering and flashing page elements can be distracting to normal users and dangerous to users with epilepsy and light sensitivity. Seizures can be triggered by flickering or flashing elements in the 4 to 59 flashes per second range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light, such as with strobe lights. Preference is to avoid the use of flickering or flashing elements. If flickering or flashing elements must be used be sure the animations are slow enough to not injure visitors and that a time-out has been implemented so the animation will stop flickering after a certain number of seconds or animation loops. Pages must be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz

Color Usage

Make sure that color is not the only way that information is conveyed on your site. Users with vision deficiencies may be unable to distinguish between certain color-combinations, and some printers and hand-held devices use single-color or black-and-white-only displays. When using color to draw attention to important page elements, use a symbol or an additional style effect that makes sense without the use of color. For form fields, indicate required fields with an asterisk and not just the color red to aid users in locating the required fields.

Assure that there is enough contrast between foreground and background colors for images and within the page itself.

Third Party Software Formats

For any content using third-party formats, follow the same guidelines for web pages to aid accessibility. This section will only cover the three most common formats used on NSDL.org: Adobe PDFs, Word documents, and PowerPoint presentations.

Adobe PDFs

Ensure that your PDF file contains searchable text and is not just an image. Searchable text is required for a screen reader to be able to process and convey the information in the PDF file. Be certain that all images, links, and form elements contain appropriate alternate descriptions and that security settings do not interfere with screen reader functionality. The easiest way to create an accessible PDF file is to convert from another file format that has already been properly formatted which will apply the necessary structure tags to the PDF file. Use Adobe Acrobat to apply structure tags to already converted PDF files.

Word Documents

Use Word styles and headings to organize your document as opposed to resizing and bolding text to define section headings. Using official Word styles will help screen readers to create an outline of the document. Provide an alternate description of all images by using the format picture dialog box. Clearly label all table headers and assure that link targets are understood out of context. Ensure that all information is understandable without the use of color. To enhance accessibility, do not post Word files on the web; provide an HTML equivalent or an accessible PDF. If you organized your Word document with Word styles and coherent links, your PDF should be appropriately tagged and accessible after conversion.

PowerPoint Slideshows

Provide an alternate description of all images by using the format picture dialog box. Ensure that all information is understandable without the use of color and ensure that link targets are understood out of context. Ensure that a transcript is available for any audio track and that audio is synchronized with video for any multimedia movies. HTML is the only reliably accessible format for PowerPoint presentations. PowerPoint presentations will not be accessible to anyone who does not have the full version of PowerPoint available and not all screen readers support the PowerPoint plugin. For best results, maintain an HTML equivalent.

Validation and Testing Checklist

Page Structure

1. Does each page have a unique and descriptive title?
2. Are all frames titled with frame identifications and navigation using the "title" and "name" attributes?
3. Are skip-navigation links provided to bypass large navigational link areas?
4. Are targets of each link clearly identified?
5. Do data tables clearly markup data cells and header cells?
6. Do tables used for layout transform gracefully when linearized?
7. Are web pages readable when stylesheets are turned off?

Alternative Descriptions

8. Do all important images have meaningful ALT attributes?
9. Do all client-side image maps have meaningful ALT attributes associated with the appropriate links?
10. Do all server-side image maps have redundant text links for each active region of the image map?

Web-based Forms

11. Are web-based forms labeled appropriately with correct HTML markup?
12. Is a link to contact information included on all web-based form pages?
13. Is a link to the privacy policy available on all web-based forms pages that collect personally identifiable information?

Color

14. Are pages designed so all information required for navigation or meaning is not dependent on the ability to identify specific colors?
15. Is there enough contrast between foreground and background colors for images and within the page itself?

Multimedia Content

16. Has all multimedia content been captioned?
17. Has captions for any multimedia presentation been synchronized with the presentation?
18. Has a transcript been provided for all audio tracks?
19. Are pages usable when scripts, applets, or other programmatic objects are turned off?
20. Is there an equivalent alternative accessible page for pages that are not usable when scripts, applets, or other programmatic objects are turned off?
21. Are links provided for applets, plug-ins or other applications required to interpret pages?
22. Are any flickering or flashing elements used on the page?

23. Is the page designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz?
24. Do all animations have a time-out implemented so the animation will stop flickering after a certain number of seconds or animation loops?
25. Do timed responses alert the user and give sufficient time to indicate more time is required?

General Validation

26. Has the page been validated for correct markup using the W3C validator?
27. Has the page been checked with at least one accessibility tool for potential accessibility issues (i.e. WAVE or WebXact)?
28. Have you stepped through the page using only a keyboard to ensure logical tab-order for links and forms?
29. Have you viewed the page with a screen reader?
30. Has the page been tested by a disabled user?
31. Does the page function reasonably in all browsers used by greater than 1% of NSDL.org visitors in the past calendar year?*

*Web pages must be validated for proper XHTML prior to publishing on the NSDL.org production server. Web pages should be tested for functionality on all browsers used by greater than 1% of NSDL.org site visitors in the previous calendar year and degrade gracefully on all older browsers and text browsers. According to Omniture tracking statistics, browsers used by greater than 1% of NSDL.org site visitors between January 1, 2006 and December 31, 2006 are Internet Explorer 6+, Firefox 1.5+ and Safari 2.0+.

Resources

Standards

WCAG 1.0: http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/
Section 508: http://www.access-board.gov/sec508/standards.htm
XHTML Specification: http://www.w3.org/TR/xhtml1/

Validators

W3C Markup Validation: http://validator.w3.org/
Cynthia Says Accessibility Validator: http://www.contentquality.com/
TotalValidator: http://www.totalvalidator.com/validator/ValidatorForm
WAVE: http://www.wave.webaim.org/index.jsp

Tutorials and Resources

Page Structure

Jim Thatcher's Section 508 tutorial
Webaim tutorial on skip-navigation links
Webaim tutorial on creating accessible images with the ALT attribute
Webaim tutorial on creating accessible images with long descriptions

Multimedia Content

Flash Accessibility Features
Webaim tutorial on captions and transcripts
Create captions and audio descriptions with the WGBH Magpie captioning tool
Create synchronized media with SMIL
Create real-time captioning for live webcasts with Caption Colorado

Color Usage

Webaim tutorial on color-blindness
Check webpages for color-blindness issues with Vischeck

Third Party Software Formats

Webaim tutorial on Word document accessibility
Webaim tutorial on PowerPoint slideshow accessibility
Adobe information on creating accessible PDF documents

Other Resources

AccessNSDL: http://www.accessnsdl.org
NCAM: http://ncam.wgbh.org/
Lynx text-based browser: http://lynx.browser.org/
Vision Australia accessibility toolbar (IE only):
http://www.visionaustralia.org.au/info.aspx?page=614