

Checklist for compliance with  
*Section 508 §1194.22 - Web-based Intranet and Internet Information and Applications*

This checklist is intended to be used in conjunction with the Academic Senate approved *Guidelines for Designing and Online Course*, specifically section 8, which is available at <http://www.avc.edu/organizations/dec/documents.htm>.

<b>Accessibility Requirement</b>	<b>Yes</b>	<b>No</b>
<b><u>Text Tags</u></b> : Is there a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content)? Reference Section 508 §1194.22 (a).		
<b><u>Multimedia Presentations</u></b> : Are there equivalent alternatives for any multimedia presentation synchronized with the presentation? Reference Section 508 §1194.22 (b).		
<b><u>Color</u></b> : Are the Web pages designed so that all information conveyed with color is also available without color? Reference Section 508 §1194.22 (c).		
<b><u>Readability</u></b> : Are the documents organized so they are readable without requiring an associated style sheet? Reference Section 508 §1194.22 (d).		
<b><u>Server-Side Image Maps</u></b> : Are redundant text links provided for each active region of a server-side image map? Reference Section 508 §1194.22 (e).		
<b><u>Client-Side Image Maps</u></b> : Are client-side image maps provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape? Reference Section 508 §1194.22 (f).		
<b><u>Data Table</u></b> : Are row and column headers identified for data tables? Reference Section 508 §1194.22 (g).		
<b><u>Data Table</u></b> : Is markup used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers? Reference Section 508 §1194.22 (h).		
<b><u>Frames</u></b> : Are frames titled with text that facilitates frame identification and navigation? Reference Section 508 §1194.22 (i).		
<b><u>Flicker Rate</u></b> : Are pages designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz? Reference Section 508 §1194.22 (j).		
<b><u>Text-Only Alternative</u></b> : Is a text-only page, with equivalent information or functionality, provided to make a web site comply with Section 508 standards, when compliance cannot be accomplished in any other way? Reference Section 508 §1194.22 (k).		

Checklist for compliance with  
*Section 508 §1194.22 - Web-based Intranet and Internet Information and Applications*

<b>Accessibility Requirement</b>	<b>Yes</b>	<b>No</b>
<b><u>Scripts</u></b> : Do pages that utilize scripting languages to display content, or to create interface elements, provide functional text that can be read by assistive technology? Reference Section 508 §1194.22 (l).		
<b><u>Applets and Plug-Ins</u></b> : Do pages that require an applet, plug-in or other application be present on the client system to interpret page content, provide a link to a plug-in or applet that complies with 508 §1194.21(a) through (l)? Reference Section 508 §1194.22 (m).		
<b><u>Electronic Forms</u></b> : Do electronic forms that are designed to be completed on-line, allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues? Reference Section 508 §1194.22 (n).		
<b><u>Navigation Links</u></b> : Are there methods that permit users to skip repetitive navigation links? Reference Section 508 §1194.22 (o).		
<b><u>Time Delays</u></b> : When a timed response is required, will the user be alerted and given sufficient time to indicate more time is required? Reference Section 508 §1194.22 (p).		

# Web Content Accessibility Guidelines

## Guideline 1. Provide equivalent alternatives to auditory and visual content.



**Provide content that, when presented to the user, conveys essentially the same function or purpose as auditory or visual content.**

Although some people cannot use images, movies, sounds, applets, etc. directly, they may still use pages that include equivalent information to the visual or auditory content. The equivalent information must serve the same purpose as the visual or auditory content. Thus, a text equivalent for an image of an upward arrow that links to a table of contents could be "Go to table of contents". In some cases, an equivalent should also describe the appearance of visual content (e.g., for complex charts, billboards, or diagrams) or the sound of auditory content (e.g., for audio samples used in education).

This guideline emphasizes the importance of providing text equivalents of non-text content (images, pre-recorded audio, video). The power of text equivalents lies in their capacity to be rendered in ways that are accessible to people from various disability groups using a variety of technologies. Text can be readily output to speech synthesizers and braille displays, and can be presented visually (in a variety of sizes) on computer displays and paper. Synthesized speech is critical for individuals who are blind and for many people with the reading difficulties that often accompany cognitive disabilities, learning disabilities, and deafness. Braille is essential for individuals who are both deaf and blind, as well as many individuals whose only sensory disability is blindness. Text displayed visually benefits users who are deaf as well as the majority of Web users.

Providing non-text equivalents (e.g., pictures, videos, and pre-recorded audio) of text is also beneficial to some users, especially nonreaders or people who have difficulty reading. In movies or visual presentations, visual action such as body language or other visual cues may not be accompanied by enough audio information to convey the same information. Unless verbal descriptions of this visual information are provided, people who cannot see (or look at) the visual content will not be able to perceive it.

### Checkpoints:

**1.1** Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). *This includes:* images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ascii art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video. **[Priority 1]**

For example, in HTML:

- Use "alt" for the IMG, INPUT, and APPLET elements, or provide a text equivalent in the content of the OBJECT and APPLET elements.

- For complex content (e.g., a chart) where the "alt" text does not provide a complete text equivalent, provide an additional description using, for example, "longdesc" with IMG or FRAME, a link inside an OBJECT element, or a [description link](#).
- For image maps, either use the "alt" attribute with AREA, or use the MAP element with A elements (and other text) as content.

Refer also to [checkpoint 9.1](#) and [checkpoint 13.10](#).

#### [Techniques for checkpoint 1.1](#)

**1.2** Provide redundant text links for each active region of a server-side image map.

[Priority 1]

Refer also to [checkpoint 1.5](#) and [checkpoint 9.1](#).

#### [Techniques for checkpoint 1.2](#)

**1.3** [Until user agents](#) can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation. [Priority 1]

Synchronize the [auditory description](#) with the audio track as per [checkpoint 1.4](#). Refer to [checkpoint 1.1](#) for information about textual equivalents for visual information.

#### [Techniques for checkpoint 1.3](#)

**1.4** For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation. [Priority 1]

#### [Techniques for checkpoint 1.4](#)

**1.5** [Until user agents](#) render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map. [Priority 3]

Refer also to [checkpoint 1.2](#) and [checkpoint 9.1](#).

#### [Techniques for checkpoint 1.5](#)

## Guideline 2. Don't rely on color alone.



### Ensure that text and graphics are understandable when viewed without color.

If color alone is used to convey information, people who cannot differentiate between certain colors and users with devices that have non-color or non-visual displays will not receive the information. When foreground and background colors are too close to the same hue, they may not provide sufficient contrast when viewed using monochrome displays or by people with different types of color deficits.

#### Checkpoints:

**2.1** Ensure that all information conveyed with color is also available without color, for example from context or markup. [Priority 1]

#### [Techniques for checkpoint 2.1](#)

**2.2** Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen. [Priority 2 for images, Priority 3 for text].

[Techniques for checkpoint 2.2](#)

## Guideline 3. Use markup and style sheets and do so properly.



### Mark up documents with the proper structural elements. Control presentation with style sheets rather than with presentation elements and attributes.

Using markup improperly -- not according to specification -- hinders accessibility. Misusing markup for a presentation effect (e.g., using a table for layout or a header to change the font size) makes it difficult for users with specialized software to understand the organization of the page or to navigate through it. Furthermore, using presentation markup rather than structural markup to convey structure (e.g., constructing what looks like a table of data with an HTML PRE element) makes it difficult to render a page intelligibly to other devices (refer to the description of [difference between content, structure, and presentation](#)).

Content developers may be tempted to use (or misuse) constructs that achieve a desired formatting effect on older browsers. They must be aware that these practices cause accessibility problems and must consider whether the formatting effect is so critical as to warrant making the document inaccessible to some users.

At the other extreme, content developers must not sacrifice appropriate markup because a certain browser or assistive technology does not process it correctly. For example, it is appropriate to use the TABLE element in HTML to mark up [tabular information](#) even though some older screen readers may not handle side-by-side text correctly (refer to [checkpoint 10.3](#)). Using TABLE correctly and creating tables that transform gracefully (refer to [guideline 5](#)) makes it possible for software to render tables other than as two-dimensional grids.

### Checkpoints:

**3.1** When an appropriate markup language exists, use markup rather than images to convey information. [Priority 2]

For example, use MathML to mark up mathematical equations, and [style sheets](#) to format text and control layout. Also, avoid using images to represent text -- use text and style sheets instead. Refer also to [guideline 6](#) and [guideline 11](#).

[Techniques for checkpoint 3.1](#)

**3.2** Create documents that validate to published formal grammars. [Priority 2]

For example, include a document type declaration at the beginning of a document that refers to a published DTD (e.g., the strict HTML 4.0 DTD).

[Techniques for checkpoint 3.2](#)

**3.3** Use style sheets to control layout and presentation. [Priority 2]

For example, use the CSS 'font' property instead of the HTML FONT element to control font styles.

[Techniques for checkpoint 3.3](#)

**3.4** Use relative rather than absolute units in markup language attribute values and style sheet property values. [Priority 2]

For example, in CSS, use 'em' or percentage lengths rather than 'pt' or 'cm', which are absolute units. If absolute units are used, validate that the rendered content is usable (refer to the [section on validation](#)).

[Techniques for checkpoint 3.4](#)

**3.5** Use header elements to convey document structure and use them according to specification. [Priority 2]

For example, in HTML, use H2 to indicate a subsection of H1. Do not use headers for font effects.

[Techniques for checkpoint 3.5](#)

**3.6** Mark up lists and list items properly. [Priority 2]

For example, in HTML, nest OL, UL, and DL lists properly.

[Techniques for checkpoint 3.6](#)

**3.7** Mark up quotations. Do not use quotation markup for formatting effects such as indentation. [Priority 2]

For example, in HTML, use the Q and BLOCKQUOTE elements to markup short and longer quotations, respectively.

[Techniques for checkpoint 3.7](#)

## Guideline 4. Clarify natural language usage



### Use markup that facilitates pronunciation or interpretation of abbreviated or foreign text.

When content developers mark up natural language changes in a document, speech synthesizers and braille devices can automatically switch to the new language, making the document more accessible to multilingual users. Content developers should identify the predominant *natural language* of a document's content (through markup or HTTP headers). Content developers should also provide expansions of abbreviations and acronyms.

In addition to helping assistive technologies, natural language markup allows search engines to find key words and identify documents in a desired language. Natural language markup also improves readability of the Web for all people, including those with learning disabilities, cognitive disabilities, or people who are deaf.

When abbreviations and natural language changes are not identified, they may be indecipherable when machine-spoken or brailled.

### Checkpoints:

**4.1** Clearly identify changes in the natural language of a document's text and any [text equivalents](#) (e.g., captions). **[Priority 1]**

For example, in HTML use the "lang" attribute. In XML, use "xml:lang".

[Techniques for checkpoint 4.1](#)

**4.2** Specify the expansion of each abbreviation or acronym in a document where it first occurs. [Priority 3]

For example, in HTML, use the "title" attribute of the ABBR and ACRONYM elements.

Providing the expansion in the main body of the document also helps document usability.

[Techniques for checkpoint 4.2](#)

**4.3** Identify the primary natural language of a document. [Priority 3]

For example, in HTML set the "lang" attribute on the HTML element. In XML, use "xml:lang". Server operators should configure servers to take advantage of HTTP content negotiation mechanisms ([RFC2068](#), section 14.13) so that clients can automatically retrieve documents of the preferred language.

[Techniques for checkpoint 4.3](#)

## Guideline 5. Create tables that transform gracefully.



**Ensure that tables have necessary markup to be transformed by accessible browsers and other user agents.**

Tables should be used to mark up truly [tabular information](#) ("data tables"). Content developers should avoid using them to lay out pages ("layout tables"). Tables for any use also present special problems to users of [screen readers](#) (refer to [checkpoint 10.3](#)).

Some [user agents](#) allow users to navigate among table cells and access header and other table cell information. Unless marked-up properly, these tables will not provide user agents with the appropriate information. ([Refer also to guideline 3.](#))

The following checkpoints will directly benefit people who access a table through auditory means (e.g., a screen reader or an automobile-based personal computer) or who view only a portion of the page at a time (e.g., users with blindness or low vision using speech output or a [braille display](#), or other users of devices with small displays, etc.).

### Checkpoints:

**5.1** For data tables, identify row and column headers. **[Priority 1]**

For example, in HTML, use TD to identify data cells and TH to identify headers.

[Techniques for checkpoint 5.1](#)

**5.2** For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells. **[Priority 1]**

For example, in HTML, use THEAD, TFOOT, and TBODY to group rows, COL and COLGROUP to group columns, and the "axis", "scope", and "headers" attributes, to describe more complex relationships among data.

[Techniques for checkpoint 5.2](#)

**5.3** Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a *linearized version*). [Priority 2]

**Note.** *Once user agents* support style sheet positioning, tables should not be used for layout. [Refer also to checkpoint 3.3.](#)

[Techniques for checkpoint 5.3](#)

**5.4** If a table is used for layout, do not use any structural markup for the purpose of visual formatting. [Priority 2]

For example, in HTML do not use the TH element to cause the content of a (non-table header) cell to be displayed centered and in bold.

[Techniques for checkpoint 5.4](#)

**5.5** Provide summaries for tables. [Priority 3]

For example, in HTML, use the "summary" attribute of the TABLE element.

[Techniques for checkpoint 5.5](#)

**5.6** Provide abbreviations for header labels. [Priority 3]

For example, in HTML, use the "abbr" attribute on the TH element.

[Techniques for checkpoint 5.6](#)

[Refer also to checkpoint 10.3.](#)

**Guideline 6. Ensure that pages featuring new technologies transform gracefully.**



**Ensure that pages are accessible even when newer technologies are not supported or are turned off.**

Although content developers are encouraged to use new technologies that solve problems raised by existing technologies, they should know how to make their pages still work with older browsers and people who choose to turn off features.

**Checkpoints:**

**6.1** Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document. **[Priority 1]**

When content is organized logically, it will be rendered in a meaningful order when style sheets are turned off or not supported.

[Techniques for checkpoint 6.1](#)

**6.2** Ensure that equivalents for dynamic content are updated when the dynamic content changes. **[Priority 1]**

[Techniques for checkpoint 6.2](#)

**6.3** Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page. **[Priority 1]**

For example, ensure that links that trigger scripts work when scripts are turned off or not supported (e.g., do not use "javascript:" as the link target). If it is not possible to make the page usable without scripts, provide a text equivalent with the NOSCRIPT element, or use a server-side script instead of a client-side script, or provide an alternative accessible page as per [checkpoint 11.4](#). [Refer also to guideline 1](#).

[Techniques for checkpoint 6.3](#)

**6.4** For scripts and applets, ensure that event handlers are input device-independent. **[Priority 2]**

Refer to the definition of [device independence](#).

[Techniques for checkpoint 6.4](#)

**6.5** Ensure that dynamic content is accessible or provide an alternative presentation or page. **[Priority 2]**

For example, in HTML, use NOFRAMES at the end of each frameset. For some applications, server-side scripts may be more accessible than client-side scripts.

[Techniques for checkpoint 6.5](#)

[Refer also to checkpoint 11.4](#).

## Guideline 7. Ensure user control of time-sensitive content changes.



**Ensure that moving, blinking, scrolling, or auto-updating objects or pages may be paused or stopped.**

Some people with cognitive or visual disabilities are unable to read moving text quickly enough or at all. Movement can also cause such a distraction that the rest of the page becomes unreadable for people with cognitive disabilities. [Screen readers](#) are unable to read moving text. People with physical disabilities might not be able to move quickly or accurately enough to interact with moving objects.

**Note.** All of the following checkpoints involve some content developer responsibility [until user agents](#) provide adequate feature control mechanisms.

### Checkpoints:

**7.1** [Until user agents](#) allow users to control flickering, avoid causing the screen to flicker. **[Priority 1]**

**Note.** People with photosensitive epilepsy can have seizures triggered by flickering or flashing in the 4 to 59 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).

[Techniques for checkpoint 7.1](#)

**7.2** [Until user agents](#) allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off). [Priority 2]

[Techniques for checkpoint 7.2](#)

**7.3** [Until user agents](#) allow users to freeze moving content, avoid movement in pages. [Priority 2]

When a page includes moving content, provide a mechanism within a script or applet to allow users to freeze motion or updates. Using style sheets with scripting to create movement allows users to turn off or override the effect more easily. [Refer also to guideline 8.](#)

[Techniques for checkpoint 7.3](#)

**7.4** [Until user agents](#) provide the ability to stop the refresh, do not create periodically auto-refreshing pages. [Priority 2]

For example, in HTML, don't cause pages to auto-refresh with "HTTP-EQUIV=refresh" until user agents allow users to turn off the feature.

[Techniques for checkpoint 7.4](#)

**7.5** [Until user agents](#) provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects. [Priority 2]

[Techniques for checkpoint 7.5](#)

**Note.** The BLINK and MARQUEE elements are not defined in any W3C HTML specification and should not be used. [Refer also to guideline 11.](#)

## Guideline 8. Ensure direct accessibility of embedded user interfaces.



**Ensure that the user interface follows principles of accessible design: device-independent access to functionality, keyboard operability, self-voicing, etc.**

When an embedded object has its "own interface", the interface -- like the interface to the browser itself -- must be accessible. If the interface of the embedded object cannot be made accessible, an alternative accessible solution must be provided.

**Note.** For information about accessible interfaces, please consult the User Agent Accessibility Guidelines ([\[WAI-USERAGENT\]](#)) and the Authoring Tool Accessibility Guidelines ([\[WAI-AUTOOL\]](#)).

**Checkpoint:**

**8.1** Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies [Priority 1 if functionality is *important* and not presented elsewhere, otherwise Priority 2.]

[Refer also to guideline 6.](#)

[Techniques for checkpoint 8.1](#)

## Guideline 9. Design for device-independence.



## Use features that enable activation of page elements via a variety of input devices.

[Device-independent](#) access means that the user may interact with the user agent or document with a preferred input (or output) device -- mouse, keyboard, voice, head wand, or other. If, for example, a form control can only be activated with a mouse or other pointing device, someone who is using the page without sight, with voice input, or with a keyboard or who is using some other non-pointing input device will not be able to use the form.

**Note.** Providing text equivalents for image maps or images used as links makes it possible for users to interact with them without a pointing device. [Refer also to guideline 1.](#)

Generally, pages that allow keyboard interaction are also accessible through speech input or a command line interface.

### Checkpoints:

**9.1** Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape. **[Priority 1]**

Refer also to [checkpoint 1.1](#), [checkpoint 1.2](#), and [checkpoint 1.5](#).

[Techniques for checkpoint 9.1](#)

**9.2** Ensure that any element that has its own interface can be operated in a device-independent manner. [Priority 2]

Refer to the definition of [device independence](#).

[Refer also to guideline 8.](#)

[Techniques for checkpoint 9.2](#)

**9.3** For scripts, specify logical event handlers rather than device-dependent event handlers. [Priority 2]

[Techniques for checkpoint 9.3](#)

**9.4** Create a logical tab order through links, form controls, and objects. [Priority 3]

For example, in HTML, specify tab order via the "tabindex" attribute or ensure a logical page design.

[Techniques for checkpoint 9.4](#)

**9.5** Provide keyboard shortcuts to important links (including those in [client-side image maps](#)), form controls, and groups of form controls. [Priority 3]

For example, in HTML, specify shortcuts via the "accesskey" attribute.

[Techniques for checkpoint 9.5](#)

## Guideline 10. Use interim solutions.



Use interim accessibility solutions so that assistive technologies and older browsers will operate correctly.

For example, older browsers do not allow users to navigate to empty edit boxes. Older screen readers read lists of consecutive links as one link. These active elements are therefore difficult or impossible to access. Also, changing the current window or popping up new windows can be very disorienting to users who cannot see that this has happened.

**Note.** The following checkpoints apply *until user agents* (including *assistive technologies*) address these issues. These checkpoints are classified as "interim", meaning that the Web Content Guidelines Working Group considers them to be valid and necessary to Web accessibility *as of the publication of this document*. However, the Working Group does not expect these checkpoints to be necessary in the future, once Web technologies have incorporated anticipated features or capabilities.

### Checkpoints:

**10.1** *Until user agents* allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user. [Priority 2]

For example, in HTML, avoid using a frame whose target is a new window.

[Techniques for checkpoint 10.1](#)

**10.2** *Until user agents* support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned. [Priority 2]

The label must immediately precede its control on the same line (allowing more than one control/label per line) or be in the line preceding the control (with only one label and one control per line). [Refer also to checkpoint 12.4](#).

[Techniques for checkpoint 10.2](#)

**10.3** *Until user agents* (including assistive technologies) render side-by-side text correctly, provide a linear text alternative (on the current page or some other) for *all* tables that lay out text in parallel, word-wrapped columns. [Priority 3]

**Note.** Please consult the definition of [linearized table](#). This checkpoint benefits people with *user agents* (such as some [screen readers](#)) that are unable to handle blocks of text presented side-by-side; the checkpoint should not discourage content developers from using tables to represent [tabular information](#).

[Techniques for checkpoint 10.3](#)

**10.4** *Until user agents* handle empty controls correctly, include default, place-holding characters in edit boxes and text areas. [Priority 3]

For example, in HTML, do this for TEXTAREA and INPUT.

[Techniques for checkpoint 10.4](#)

**10.5** *Until user agents* (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links. [Priority 3]

[Techniques for checkpoint 10.5](#)

## Guideline 11. Use W3C technologies and guidelines.



**Use W3C technologies (according to specification) and follow accessibility guidelines. Where it is not possible to use a W3C technology, or doing so results in material that does not transform gracefully, provide an alternative version of the content that is accessible.**

The current guidelines recommend W3C technologies (e.g., HTML, CSS, etc.) for several reasons:

- W3C technologies include "built-in" accessibility features.
- W3C specifications undergo early review to ensure that accessibility issues are considered during the design phase.
- W3C specifications are developed in an open, industry consensus process.

Many non-W3C formats (e.g., PDF, Shockwave, etc.) require viewing with either plug-ins or stand-alone applications. Often, these formats cannot be viewed or navigated with standard [user agents](#) (including [assistive technologies](#)). Avoiding non-W3C and non-standard features (proprietary elements, attributes, properties, and extensions) will tend to make pages more accessible to more people using a wider variety of hardware and software. When inaccessible technologies (proprietary or not) must be used, equivalent accessible pages must be provided.

Even when W3C technologies are used, they must be used in accordance with accessibility guidelines. When using new technologies, ensure that they transform gracefully ([Refer also to guideline 6.](#)).

**Note.** Converting documents (from PDF, PostScript, RTF, etc.) to W3C markup languages (HTML, XML) does not always create an accessible document. Therefore, validate each page for accessibility and usability after the conversion process (refer to the [section on validation](#)). If a page does not readily convert, either revise the page until its original representation converts appropriately or provide an HTML or plain text version.

## Checkpoints:

**11.1** Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported. [Priority 2]

Refer to the [list of references](#) for information about where to find the latest W3C specifications and [\[WAI-UA-SUPPORT\]](#) for information about user agent support for W3C technologies.

[Techniques for checkpoint 11.1](#)

**11.2** Avoid deprecated features of W3C technologies. [Priority 2]

For example, in HTML, don't use the [deprecated](#) FONT element; use style sheets instead (e.g., the 'font' property in CSS).

[Techniques for checkpoint 11.2](#)

**11.3** Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.) [Priority 3]

**Note.** Use content negotiation where possible.

[Techniques for checkpoint 11.3](#)

11.4 If, [after best efforts](#), you cannot create an [accessible](#) page, provide a link to an alternative page that uses W3C technologies, is accessible, has [equivalent](#) information (or functionality), and is updated as often as the inaccessible (original) page. [Priority 1]  
[Techniques for checkpoint 11.4](#)

**Note.** Content developers should only resort to alternative pages when other solutions fail because alternative pages are generally updated less often than "primary" pages. An out-of-date page may be as frustrating as one that is inaccessible since, in both cases, the information presented on the original page is unavailable. Automatically generating alternative pages may lead to more frequent updates, but content developers must still be careful to ensure that generated pages always make sense, and that users are able to navigate a site by following links on primary pages, alternative pages, or both. Before resorting to an alternative page, reconsider the design of the original page; making it accessible is likely to improve it for all users.

## Guideline 12. Provide context and orientation information.



**Provide context and orientation information to help users understand complex pages or elements.**

Grouping elements and providing contextual information about the relationships between elements can be useful for all users. Complex relationships between parts of a page may be difficult for people with cognitive disabilities and people with visual disabilities to interpret.

### Checkpoints:

**12.1** Title each frame to facilitate frame identification and navigation. [Priority 1]

For example, in HTML use the "title" attribute on FRAME elements.

[Techniques for checkpoint 12.1](#)

**12.2** Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone. [Priority 2]

For example, in HTML, use "longdesc," or a [description link](#).

[Techniques for checkpoint 12.2](#)

**12.3** Divide large blocks of information into more manageable groups where natural and appropriate. [Priority 2]

For example, in HTML, use OPTGROUP to group OPTION elements inside a SELECT; group form controls with FIELDSET and LEGEND; use nested lists where appropriate; use headings to structure documents, etc. [Refer also to guideline 3.](#)

[Techniques for checkpoint 12.3](#)

**12.4** Associate labels explicitly with their controls. [Priority 2]

For example, in HTML use LABEL and its "for" attribute.

[Techniques for checkpoint 12.4](#)

## Guideline 13. Provide clear navigation mechanisms.



**Provide clear and consistent navigation mechanisms -- orientation information, navigation bars, a site map, etc. -- to increase the likelihood that a person will find what they are looking for at a site.**

Clear and consistent [navigation mechanisms](#) are important to people with cognitive disabilities or blindness, and benefit all users.

### Checkpoints:

**13.1** Clearly identify the target of each link. [Priority 2]

[Link text](#) should be meaningful enough to make sense when read out of context -- either on its own or as part of a sequence of links. Link text should also be terse.

For example, in HTML, write "Information about version 4.3" instead of "click here". In addition to clear link text, content developers may further clarify the target of a link with an informative link title (e.g., in HTML, the "title" attribute).

[Techniques for checkpoint 13.1](#)

**13.2** Provide metadata to add semantic information to pages and sites. [Priority 2]

For example, use RDF ([IRDFI](#)) to indicate the document's author, the type of content, etc.

**Note.** Some HTML [user agents](#) can build navigation tools from document relations described by the HTML LINK element and "rel" or "rev" attributes (e.g., rel="next", rel="previous", rel="index", etc.). [Refer also to checkpoint 13.5.](#)

[Techniques for checkpoint 13.2](#)

**13.3** Provide information about the general layout of a site (e.g., a site map or table of contents). [Priority 2]

In describing site layout, highlight and explain available accessibility features.

[Techniques for checkpoint 13.3](#)

**13.4** Use navigation mechanisms in a consistent manner. [Priority 2]

[Techniques for checkpoint 13.4](#)

**13.5** Provide navigation bars to highlight and give access to the navigation mechanism. [Priority 3]

[Techniques for checkpoint 13.5](#)

**13.6** Group related links, identify the group (for user agents), and, [until user agents](#) do so, provide a way to bypass the group. [Priority 3]

[Techniques for checkpoint 13.6](#)

**13.7** If search functions are provided, enable different types of searches for different skill levels and preferences. [Priority 3]

[Techniques for checkpoint 13.7](#)

**13.8** Place distinguishing information at the beginning of headings, paragraphs, lists, etc. [Priority 3]

**Note.** This is commonly referred to as "front-loading" and is especially helpful for people accessing information with serial devices such as speech synthesizers.

[Techniques for checkpoint 13.8](#)

**13.9** Provide information about document collections (i.e., documents comprising multiple pages.). [Priority 3]

For example, in HTML specify document collections with the LINK element and the "rel" and "rev" attributes. Another way to create a collection is by building an archive (e.g., with zip, tar and gzip, stuffit, etc.) of the multiple pages.

**Note.** The performance improvement gained by offline processing can make browsing much less expensive for people with disabilities who may be browsing slowly.

[Techniques for checkpoint 13.9](#)

**13.10** Provide a means to skip over multi-line ASCII art. [Priority 3]

Refer to [checkpoint 1.1](#) and [the example of ascii art in the glossary](#).

[Techniques for checkpoint 13.10](#)

## Guideline 14. Ensure that documents are clear and simple.



### Ensure that documents are clear and simple so they may be more easily understood.

Consistent page layout, recognizable graphics, and easy to understand language benefit all users. In particular, they help people with cognitive disabilities or who have difficulty reading. (However, ensure that images have text equivalents for people who are blind, have low vision, or for any user who cannot or has chosen not to view graphics. [Refer also to guideline 1.](#))

Using clear and simple language promotes effective communication. Access to written information can be difficult for people who have cognitive or learning disabilities. Using clear and simple language also benefits people whose first language differs from your own, including those people who communicate primarily in sign language.

#### Checkpoints:

**14.1** Use the clearest and simplest language appropriate for a site's content. [Priority 1]

[Techniques for checkpoint 14.1](#)

**14.2** Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page. [Priority 3]

[Refer also to guideline 1.](#)

[Techniques for checkpoint 14.2](#)

**14.3** Create a style of presentation that is consistent across pages. [Priority 3] [Techniques for checkpoint 14.3](#)