

# Section 508 Web Accessibility Checklist for HTML

Updated March 29, 2001

SEC. 508 STANDARD §1194.22	PASS	FAIL
<p>(a) A text equivalent for every non-text element shall be provided (e.g., via <i>alt</i> tags, <i>longdesc</i>, or in element content).</p> <p>[See Note 1]</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Every image, Java applet, Flash file, video file, audio file, plug-in, etc. has an <i>alt</i> description.</li> <li><input type="checkbox"/> Complex graphics (graphs, charts, etc.) are accompanied by detailed text descriptions.</li> <li><input type="checkbox"/> The alt descriptions succinctly describe the <u>purpose</u> of the objects, without being too verbose (for simple objects) or too vague (for complex objects).</li> <li><input type="checkbox"/> <i>Alt</i> descriptions for images used as links are descriptive of the link destination.</li> <li><input type="checkbox"/> Decorative graphics with no other function have <u>empty alt</u> descriptions (<i>alt= ""</i>), but they never have <u>missing alt</u> descriptions.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> A non-text element has no <i>alt</i> description.</li> <li><input type="checkbox"/> Complex graphics have no alternative text, or the alternative does not fully convey the meaning of the graphic.</li> <li><input type="checkbox"/> <i>Alt</i> descriptions are verbose, vague, misleading, inaccurate or redundant to the context (e.g. the <i>alt</i> text is the same as the text immediately preceding or following it in the document).</li> <li><input type="checkbox"/> <i>Alt</i> descriptions for images used as links are not descriptive of the link destination.</li> <li><input type="checkbox"/> Purely decorative graphics have <i>alt</i> descriptions that say "spacer," "decorative graphic," or other titles that only increase the time that it takes to listen to a page when using a screen reader.</li> </ul>
<p>(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Multimedia files have <u>synchronized</u> captions.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Multimedia files do not have captions, or captions which are not synchronized.</li> </ul>
<p>(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> If color is used to convey important information, an alternative indicator is used, such as an asterisk (*) or other symbol.</li> <li><input type="checkbox"/> Contrast is good.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> The use of a color monitor is required.</li> <li><input type="checkbox"/> Contrast is poor.</li> </ul>
<p>(d) Documents shall be organized so they are readable without requiring an associated style sheet.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Style sheets may be used for color, indentation and other presentation effects, but the document is still understandable (even if less visually appealing) when the style sheet is turned off.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> The document is confusing or information is missing when the style sheet is turned off.</li> </ul>
<p>(e) Redundant text links shall be provided for each active region of a server-side image map.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Separate text links are provided outside of the server-side image map to access the same content that the image map hot spots access.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> The only way to access the links of a server-side image map is through the image map hot spots, which usually means that a mouse is required and that the links are unavailable to assistive technologies.</li> </ul>

## Section 508 Web Accessibility Checklist for HTML (continued)

SEC. 508 STANDARD §1194.22	PASS	FAIL
(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	<input type="checkbox"/> Standard HTML client-side image maps are used, and appropriate <i>alt</i> descriptions are provided for the image as well as the hot spots.	<input type="checkbox"/> Server-side image maps are used when a client-side image map would suffice.
(g) Row and column headers shall be identified for data tables.	<input type="checkbox"/> Data tables have the column and row headers appropriately identified (using the <th> tag). <input type="checkbox"/> Tables used strictly for <u>layout purposes</u> do NOT have header rows or columns.	<input type="checkbox"/> Data tables have no header rows or columns. <input type="checkbox"/> Tables used for layout use the header attribute when there is no true header.
(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.	<input type="checkbox"/> Table cells are associated with the appropriate headers (e.g. with the <i>id</i> , <i>headers</i> , <i>scope</i> and/or <i>axis</i> HTML attributes).	<input type="checkbox"/> Columns and rows are not associated with column and row headers, or they are associated incorrectly.
(i) Frames shall be titled with text that facilitates frame identification and navigation.	<input type="checkbox"/> Each frame is given a title that helps the user understand the frame's purpose.	<input type="checkbox"/> Frames have no titles, or titles that are not descriptive of the frame's purpose.
(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.	<input type="checkbox"/> No elements on the page flicker at a rate of 2 to 55 cycles per second, thus reducing the risk of optically-induced seizures.	<input type="checkbox"/> An element on the page flickers at a rate of 2 to 55 cycles per second, increasing the risk of optically-induced seizures.
(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.  [See Note 2]	<input type="checkbox"/> A text-only version is created only when there is no other way to make the content accessible, or when it offers significant advantages over the "main" version for certain disability types. <input type="checkbox"/> The text-only version is up-to-date with the "main" version. <input type="checkbox"/> The text-only version provides the functionality equivalent to that of the "main" version. <input type="checkbox"/> An alternative is provided for components (e.g. plug-ins, scripts) that are not directly accessible.	<input type="checkbox"/> A text-only version is provided only as an excuse not to make the "main" version fully accessible. <input type="checkbox"/> The text-only version is not up-to-date with the "main" version. <input type="checkbox"/> The text-only version is an unequal, lesser version of the "main" version. <input type="checkbox"/> No alternative is provided for components that are not directly accessible.
(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.  [See Note 3]	<input type="checkbox"/> Information within the scripts is text-based, or a text alternative is provided within the script itself, in accordance with (a) in these standards. <input type="checkbox"/> All scripts (e.g. Javascript pop-up menus) are either directly accessible to assistive technologies (keyboard accessibility is a good measure of this), or an alternative method of accessing equivalent functionality is provided (e.g. a standard HTML link).	<input type="checkbox"/> Scripts include graphics-as-text with no true text alternative. <input type="checkbox"/> Scripts only work with a mouse, and there is no keyboard-accessible alternative either within or outside of the script.

SEC. 508 STANDARD §1194.22	PASS	FAIL
<p>(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).</p> <p>[See Notes 4-6]</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> A link is provided to a disability-accessible page where the plug-in can be downloaded.</li> <li><input type="checkbox"/> All Java applets, scripts and plug-ins (including Acrobat PDF files and PowerPoint files, etc.) and the content within them are accessible to assistive technologies, or else an alternative means of accessing equivalent content is provided.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> No link is provided to a page where the plug-in can be downloaded and/or the download page is not disability-accessible.</li> <li><input type="checkbox"/> Plugins, scripts and other elements are used indiscriminately, without alternatives for those who cannot access them.</li> </ul>
<p>(n) When electronic forms are designed to be completed online, the form shall 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.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> All form controls have text labels adjacent to them.</li> <li><input type="checkbox"/> Form elements have labels associated with them in the markup (i.e. the <i>id</i> and <i>for</i> elements).</li> <li><input type="checkbox"/> Dynamic HTML scripting of the form does not interfere with assistive technologies.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Form controls have no labels, or the labels are not adjacent to the controls.</li> <li><input type="checkbox"/> There is no linking of the form element and its label in the HTML.</li> <li><input type="checkbox"/> Dynamic HTML scripting makes parts of the form unavailable to assistive technologies.</li> </ul>
<p>(o) A method shall be provided that permits users to skip repetitive navigation links.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> A link is provided to skip over lists of navigational menus or other lengthy lists of links.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> There is no way to skip over lists of links.</li> </ul>
<p>(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> The user has control over the timing of content changes.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> The user is required to react quickly, within limited time restraints.</li> </ul>

**Note 1:** Until the *longdesc* tag is better supported by browsers, it is impractical to use.

**Note 2:** “Text-only” and “accessible” are NOT synonymous. Text-only sites may help people with certain types of visual disabilities, but are not always helpful to those with cognitive, motor, or hearing disabilities.

**Note 3:** At this time, many elements of Dynamic HTML (client-side scripted HTML, which is usually accomplished with Javascript) cannot be made directly accessible to assistive technologies and keyboards, especially when the *onMouseover* command is used. If an *onMouseover* (or similar) element does not contain any important information (e.g. the script causes a button to “glow”), then there is no consequence for accessibility. If this scripted event reveals important information, then a keyboard-accessible alternative is required.

**Note 4:** When embedded into web pages, few plug-ins are currently directly accessible. Some of them e.g. RealPlayer) are more accessible as standalone products. It may be better to invoke the whole program rather than embed movies into pages at this point, although this may change in the future.

**Note 5:** Acrobat Reader 5.0 allows screen readers to access PDF documents. However, not all users have this version installed, and not all PDF documents are text-based (some are scanned in as graphics), which renders them useless to many assistive technologies. It is recommended that an accessible HTML version be made available as an alternative to PDF.

**Note 6:** PowerPoint files are currently not directly accessible unless the user has a full version of the PowerPoint program on the client computer (and not just the PowerPoint viewer). It is recommended that an accessible HTML version be provided as well.

# Section 508 Web Accessibility Checklist for Scripts, Plug-ins, Java, etc.

## SEC. 508 STANDARD §1194.21

- (a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.
- (b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.
- (c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.
- (d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text.
- (e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance.
- (f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.
- (g) Applications shall not override user selected contrast and color selections and other individual display attributes.
- (h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.
- (i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.
- (j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.
- (k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.
- (l) When electronic forms are used, the form shall 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.