CSS Techniques for Web Content Accessibility Guidelines 1.0

W3C Working Draft 23 June 2000

This version:
http://www.w3.org/WAI/GL/WD-WCAG10-CSS-TECHS-20000623
(plain text, postscript, pdf, gzip tar file of HTML, zip archive of HTML)
Latest version:
http://www.w3.org/WAI/GL/WCAG10

Editors:
Wendy Chisholm, W3C,
Gregg Vanderheiden, Trace R & D Center, University of Wisconsin -- Madison
Ian Jacobs, W3C

Abstract

This document describes techniques for satisfying the checkpoints of "Web Content Accessibility Guidelines 1.0" [WCAG10] [p. 12] when using CSS ([CSS1] [p. 12] , [CSS2] [p. 12]).

CSS1 allows content developers to duplicate most HTML 4.0 presentation capabilities and offer more power with less cost. However, until most users have browsers that support style sheets, not every presentation idiom may be expressed satisfactorily with style sheets. We also provide examples of how to use HTML 4.0 features (e.g., tables, bitmap text) more accessibly when they must be used.

This document is part of a series of accessibility documents published by the Web Accessibility Initiative. "Techniques for Web Content Accessibility Guidelines 1.0" [WCAG10-TECHS] [p. 12] contains additional techniques and references for other formats and languages.

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.

While Web Content Accessibility Guidelines 1.0 strives to be a stable document (as a W3C Recommendation), the current document is expected to evolve as technologies change and content developers discover more effective techniques for designing accessible Web sites and pages.

This is a W3C Working Draft for review by W3C Members and other interested parties. It is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress". This is work in progress and does not imply endorsement by, or the consensus of, either W3C or participants in the Web Content Accessibility Guidelines (WCAG) Working Group.

This document has been produced as part of the W3C Web Accessibility Initiative. The goal of the Web Content Guidelines Working Group is discussed in the Working Group charter.

A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR.

Please send detailed comments on this document to wai-wcag-editor@w3.org.

To do

- Get rid of @@’s placed by editors.
- Incorporate content from the CSS accessibility note.

Table of Contents

Abstract 1
Status of this document 1
1 Guidelines for creating style sheets 4
2 Fonts 5
3 Text style 6
3.1 Text instead of images 6
4 Text formatting 6
5 Colors 7
6 Layout, positioning, layering, and alignment 8
6.1 If you must use images as spacers 9
7 Rules and borders 9
8 Creating style sheets that transform gracefully 10
Acknowledgments 11
9 References 12
10 Resources 12
10.1 Accessibility resources 12

1 Guidelines for creating style sheets

Checkpoints in this section: 14.3 and 3.4.

Here are guidelines for creating style sheets that promote accessibility:

- Use a minimal number of style sheet for your site
- If you have more than one, use the same "class" name for the same concept in all of the style sheets.
- Use linked style sheets rather than embedded styles, and avoid inline style sheets.
- Content developers should not write "!important" rules. Users should where necessary.
- Use the "em" unit to set font sizes.
- Use relative length units and percentages. CSS allows you to use relative units even in absolute positioning. Thus, you may position an image to be offset by "3em" from the top of its containing element. This is a fixed distance, but is relative to the current font size, so it scales nicely.
- Only use absolute length units when the physical characteristics of the output medium are known.
- Always specify a fallback generic font.
- Use numbers, not names, for colors.
- Provide a text equivalent for any important image or text generated by style sheets (e.g., via the ’background-image’, ’list-style’, or ’content’ properties). **Note.** Text generated by style sheets is part of the document source and will not be available to assistive technologies that access content through DOM, level 1 ([DOM1] [p. 12]).
- Be sure to validate that your pages are still readable without style sheets.

Some examples follow.

Example.

Use em to set font sizes, as in:

```
H1 { font-size: 2em }
```

rather than:

```
H1 { font-size: 12pt }
```

End example.

Example.

Use relative length units and percentages.

```
BODY { margin-left: 15%; margin-right: 10%}
```

End example.

Example.

Only use absolute length units when the physical characteristics of the output medium are known.

```
.businesscard { font-size: 8pt }
```

End example.

Example.

Always specify a fallback generic font:

```
BODY { font-family: "Gill Sans", sans-serif }
```

End example.

Example.

Use numbers, not names, for colors:

```
H1 {color: #808000}
H1 {color: rgb(50%,50%,0%)}
```

End example.

2 Fonts

Checkpoints in this section: 11.2.

Instead of using deprecated presentation elements and attributes , use the many CSS properties to control font characteristics: ’font-family’, ’font-size’, ’font-size-adjust’, ’font-stretch’, ’font-style’, ’font-variant’, and ’font-weight’.

The following CSS2 properties can be used to control font information: ’font’, ’font-family’, ’font-size’, ’font-size-adjust’, ’font-stretch’, ’font-style’, ’font-variant’, and ’font-weight’.

Use them instead of the following deprecated font elements and attributes in HTML: FONT, BASEFONT, "face", and "size".

If you must use HTML elements to control font information, use BIG and SMALL, which are not deprecated.

Example.

```
<STYLE type="text/css">
   P.important { font-weight: bold }
   P.less-important { font-weight: lighter; font-size: smaller }
   H2.subsection { font-family: Helvetica, sans-serif }
</STYLE>
```

End example.

3 Text style

Checkpoints in this section: 7.2.

The following CSS2 properties can be used to style text:

- Case: ’text-transform’ (for uppercase, lowercase, and capitalization).
- Shadow effects: ’text-shadow’
- Underlines, overlinks, blinking: ’text-decoration’. **Note.** If blinking content (e.g., a headline that appears and disappears at regular intervals) is used, provide a mechanism for stopping the blinking. In CSS, ’text-decoration: blink’ will cause content to blink and will allow users to stop the effect by turning off style sheets or overriding the rule in a user style sheet. Do not use the BLINK and MARQUEE elements. These elements are not part of any W3C specification for HTML (i.e., they are non-standard elements).

3.1 Text instead of images

Checkpoints in this section: 3.3.

Content developers should use style sheets to style text rather than representing text in images. Using text instead of images means that the information will be available to a greater number of users (with speech synthesizers, braille displays, graphical displays, etc.). Using style sheets will also allow users to override author styles and change colors or fonts sizes more easily.

If it is necessary to use a bitmap to create a text effect (special font, transformation, shadows, etc.) the bitmap must be accessible (see the sections on text equivalents and alternative pages).

Example.

In this example, the inserted image shows the large red characters "Example", and is captured by the value of the "alt" attribute.

```
<P>This is an
   <IMG src="BigRedExample.gif" alt="example">
   of what we mean.
</P>
```

End example.

4 Text formatting

Checkpoints in this section: 3.3.

The following CSS2 properties can be used to control the formatting and position of text:

- Indentation: ’text-indent’. Do not use the BLOCKQUOTE or any other structural element to indent text.
- Letter/word spacing: ’letter-spacing’, ’word-spacing’. For example instead of writing "H E L L O" (which users generally recognize as the word "hello" but would hear as individual letters), authors may create the same visual effect with the ’word-spacing’ property applied to "HELLO". Text without spaces will be transformed more effectively to speech.
- White space: ’white-space’. This property controls the white space processing of an element’s content.
- Text direction: ’direction’, ’unicode-bidi’.
- The :first-letter and :first-line pseudo-elements allow authors to refer to the first letter or line of a paragraph of text.

The following example shows how to use style sheets to create a drop-cap effect.

Example.

```
<HEAD>
<TITLE>Drop caps</TITLE>
<STYLE type="text/css">
      .dropcap { font-size : 120%; font-family : Helvetica }
</STYLE>
</HEAD>
<BODY>
<P><SPAN class="dropcap">O</SPAN>nce upon a time...
</BODY>
```

Note. As of the writing of this document, the CSS pseudo-element ’:first-letter’, which allows content developers to refer to the first letter of a chunk of text, is not widely supported.

5 Colors

Checkpoints in this section: 2.1 and 2.2.

Use these CSS properties to specify colors:

- ’color’, for foreground text color.
- ’background-color’, for background colors.
- ’border-color’, ’outline-color’ for border colors.
- For link colors, refer to the :link, :visited, and :active pseudo-classes.

Ensure that foreground and background colors contrast well. If specifying a foreground color, always specify a background color as well (and vice versa).

Ensure that information is not conveyed through color alone. For example, when asking for input from users, do not write "Please select an item from those listed in green." Instead, ensure that information is available through other style effects (e.g., a font effect) and through context (e.g,. comprehensive text links).

For instance, in this document, examples are styled by default (through style sheets) as follows:

- They are surrounded by a border.
- They use a different background color.
- They begin with the word "Example" (or "Deprecated Example".
- They also end with the phrase "End example", but that phrase is hidden by default with ’display: none’. For user agents that don’t support style sheets or when style sheets are turned off, this text helps delineate the end of an example for readers who may not be able to see the border around the example.

Quicktest! To test whether your document still works without colors, examine it with a monochrome monitor or browser colors turned off. Also, try setting up a color scheme in your browser that only uses black, white, and the four browser-safe greys and see how your page holds up.

Quicktest! To test whether color contrast is sufficient to be read by people with color deficiencies or by those with low resolution monitors, print pages on a black and white printer (with backgrounds and colors appearing in grayscale). Also try taking the printout and copying it for two or three generations to see how it degrades. This will show you where you need to add redundant cues (example: hyperlinks are usually underlined on Web pages), or whether the cues are two small or indistinct to hold up well.

For more information about colors and contrasts, refer to [LIGHTHOUSE] [p. 12] .

6 Layout, positioning, layering, and alignment

Checkpoints in this section: 3.3 and 5.3.

Layout, positioning, layering, and alignment should be done through style sheets (notably by using CSS floats and absolute positioning):

- ’text-indent’, ’text-align’, ’word-spacing’, ’font-stretch’. Each of these properties allows users to control spacing without adding additional spaces. Use ’text-align: center’ instead of the deprecated CENTER element.
- ’margin’, ’margin-top’, ’margin-right’, ’margin-bottom’, ’margin-left’. With these properties, authors can create space on four sides of an element’s content instead of adding non-breaking spaces (&nbsp;), which are non-standard mark-up, to create space around an element.
- ’float’, ’position’, ’top’, ’right’, ’bottom’, ’left’. With these properties, the user can control the visual position of almost any element in a manner independent of where the element appears in the document. Authors should always design documents that make sense without style sheets (i.e., the document should be written in a "logical" order) and then apply style sheets to achieve visual effects. The positioning properties may be used to create margin notes (which may be automatically numbered), side bars, frame-like effects, simple headers and footers, and more.

- The ’empty-cells’ property allows users to leave table cells empty and still give them proper borders on the screen or on paper. A data cell that is meant to be empty should not be filled with white space or a non-breaking space just to achieve a visual effect.

6.1 If you must use images as spacers

Provide text equivalents for all images, including invisible or transparent images.

If content developers cannot use style sheets and must use invisible or transparent images (e.g., with IMG) to lay out images on the page, they should specify `alt=""` for them.

Deprecated example.

Authors should **not** use spaces for the value of "alt" to prevent the words from running together when the image is not loaded:

```
my poem requires a big space<IMG src="10pttab.gif" alt="&nbsp;&nbsp;&nbsp;">here
```

In this next example, an image is used to force a graphic to appear in a certain position:

```
<IMG src="spacer.gif" alt="spacer">
<IMG src="colorfulwheel.gif" alt="The wheel of fortune">
```

End example.

7 Rules and borders

Rules and borders may convey the notion of "separation" to visually enabled users but that meaning cannot be inferred out of a visual context.

Use these CSS properties to specify border styles:

- ’border’, ’border-width’, ’border-style’, ’border-color’.
- ’border-spacing’ and ’border-collapse’ for tables.
- ’outline, ’outline-color’, ’outline-style’, and ’outline-width’ for dynamic outlines.

Authors should use style sheets to create rules and borders.

Example.

In this example, the H1 element will have a top border that is 2px thick, red, and separated from the content by 1em:

```
<HEAD>
<TITLE>Redline with style sheets</TITLE>
<STYLE type="text/css">
     H1 { padding-top: 1em; border-top: 2px red }
</STYLE>
</HEAD>
<BODY>
<H1>Chapter 8 - Auditory and Tactile Displays</H1>
</BODY>
```

End example.

If a rule (e.g., the HR element) is used to indicate structure, be sure to indicate the structure in a non-visual way as well. (e.g., by using structural markup).

Example.

In this example, the DIV element is used to create a navigation bar, which includes a horizontal separator.

```
<DIV class="navigation-bar">
   <HR>
   <A rel="Next" href="next.html">[Next page]</A>
   <A rel="Previous" href="previous.html">[Prevous page]</A>
   <A rel="First" href="first.html">[First page]</A>
</DIV>
```

End example.

8 Creating style sheets that transform gracefully

Checkpoints in this section: 6.1, 7.3, and 11.3.

@@To discuss in this section:

- Reading order of a document. testing when css not loaded.
- Use of style sheets (along with scripting and dom) to hide/show content, or change presentation of (movement, colors).
- Using the @media to select appropriate style sheet.
- Aural cascading style sheets?

Acknowledgments

Web Content Guidelines Working Group Co-Chairs:

Jason White, University of Melbourne

Gregg Vanderheiden, Trace Research and Development

W3C Team contact:

Wendy Chisholm

We wish to thank the following people who have contributed their time and valuable comments to shaping these guidelines:

Harvey Bingham, Kevin Carey, Chetz Colwell, Neal Ewers, Geoff Freed, Al Gilman, Larry Goldberg, Jon Gunderson, Eric Hansen, Phill Jenkins, Leonard Kasday, George Kerscher, Marja-Riitta Koivunen, Josh Krieger, Chuck Letourneau, Scott Luebking, William Loughborough, Murray Maloney, Charles McCathieNevile, MegaZone (Livingston Enterprises), Masafumi Nakane, Mark Novak, Charles Oppermann, Mike Paciello, David Pawson, Michael Pieper, Greg Rosmaita, Liam Quinn, Dave Raggett, T.V. Raman, Robert Savellis, Jutta Treviranus, Steve Tyler, and Jaap van Lelieveld

The original draft of this document is based on "The Unified Web Site Accessibility Guidelines" [[UWSAG]]] compiled by the Trace R & D Center at the University of Wisconsin. That document includes a list of additional contributors.

9 References

For the latest version of any W3C specification please consult the list of W3C Technical Reports at http://www.w3.org/TR.

[CSS1]
"CSS, level 1 Recommendation", B. Bos, H. Wium Lie, eds., 17 December 1996, revised 11 January 1999. This CSS1 Recommendation is http://www.w3.org/TR/1999/REC-CSS1-19990111. The latest version of CSS1 is available at http://www.w3.org/TR/REC-CSS1.

[CSS2]
"CSS, level 2 Recommendation", B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, eds., 12 May 1998. This CSS2 Recommendation is http://www.w3.org/TR/1998/REC-CSS2-19980512. The latest version of CSS2 is available at http://www.w3.org/TR/REC-CSS2.

[DOM1]
"Document Object Model (DOM) Level 1 Specification", V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, A. Le Hors, G. Nicol, J. Robie, R. Sutor, C. Wilson, and L. Wood, eds., 1 October 1998. This DOM Level 1 Recommendation is http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001. The latest version of DOM Level 1 is available at http://www.w3.org/TR/REC-DOM-Level-1.

[WCAG10]
"Web Content Accessibility Guidelines 1.0", W. Chisholm, G. Vanderheiden, and I. Jacobs, eds., 5 May 1999. This WCAG 1.0 Recommendation is http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505.

[WCAG10-TECHS]
"Techniques for Web Content Accessibility Guidelines 1.0", W. Chisholm, G. Vanderheiden, I. Jacobs, eds. This document explains how to implement the checkpoints defined in "Web Content Accessibility Guidelines 1.0". The latest draft of the techniques is available at http://www.w3.org/WAI/GL/WCAG10-TECHS/.

10 Resources

Note: *W3C does not guarantee the stability of any of the following references outside of its control. These references are included for convenience. References to products are not endorsements of those products.*

10.1 Accessibility resources

[LIGHTHOUSE]
The Lighthouse provides information about accessible colors and contrasts.

W3C WAI-AAA WCAG 1.0