

IT Accessibility Guidelines for Web Developers

Anyone who codes static Web sites or dynamic Web applications must ensure that the content is accessible to individuals with disabilities.

The Web has become such a sensory-rich medium that it's easy to forget about making the content accessible to people with visual, hearing, and/or motion impairments.

- Blind individuals often use screen readers, which is software that uses audio output to interpret and "read" the content out loud to the user.
- Hearing impaired individuals may depend on captions and transcripts to access audio.
- Individuals with motor impairments may prefer to use a keyboard rather than a mouse to navigate a Web page.

Guidelines

1. Provide alt attributes for meaningful images.

An alt attribute is a short description of an image that a screen reader can "read" to the user. Alt attributes are always required, but images that don't add any meaning to the Web page, such as divider lines, should have empty alt attributes:

```

```

2. Use headings appropriately to convey hierarchy.

Use <h1> through <h6> tags for page headings, and be sure to use them in the proper order (e.g., <h1> precedes <h2>). Use CSS to style heading tags:

```
<h1>The Future of UC Retirement Benefits</h1>
```

3. Style text with CSS rather than use images.

Do not create graphics that look like text. Use text and style it with CSS instead:

```
<h2 class="introBlurb">The 10 campuses of the University of California  
open their doors to all who work hard and dream big.</h2>
```

4. Make sure the site is readable with stylesheets disabled.

If you disable CSS, the site must be readable and logical. It may be less visually appealing, but the site should not be confusing or lose information.

5. Use skip navigation links.

"Skip nav" links allow users to jump to the content and skip the repetitive navigation links. You can display the skip nav link at the top of a page or use CSS to hide it until the link is active. **DO NOT** use `display: none;` to hide the skip nav link, or else it will also be hidden from screen reader users! Instead, use a negative value for margin-left to "hide" the link until it has focus.

```
<p id="skipNav"><a href="#mainContent">Skip to Main Content</a></p>
```

6. Use device-independent code.

Do not assume that a visitor is using a particular input device. The `onchange`, `onhover`, `onmouseover`, and `onmouseout` JavaScript event handlers require use of a mouse. Instead, use the `onselect`, `onfocus`, and `onblur` handlers, which work with either a

keyboard or mouse. The onclick handler must be used in conjunction with other link and form elements.

7. JavaScript should have an alternate presentation.

If you use JavaScript on a Web page, it's important to provide the equivalent value for users who do not have JavaScript on their browsers. A common method is to add <noscript> tags immediately after the closing </script> tag.

8. Label form elements.

Visually impaired people may need a screen reader to associate a form's input elements and labels. If the input elements aren't labeled appropriately, the user may complete the wrong field or be unable to access the form.

```
<label id="name"> Name (first and last name):</label>
<input type="text" id="name" value="name" size="20" />
```

9. Caption all video and audio.

Video and audio files should have synchronized captions. Videos without audio should have an accompanying audio track to describe the video for the blind or vision impaired.

10. Don't rely on color to convey meaning.

Color-blind users may have trouble with a Web site if color is used to convey important information. Avoid instances of "Click the green button to proceed."

11. Ensure appropriate color contrast.

Use tools, such as [JuicyStudio](#) or [Paciello Contrast Analyzer](#), to measure the contrast among graphics, fonts, and backgrounds on your page. Users can change the site's contrast to suit their needs if you used plain text instead of images of text (**see tip #3**).

12. Mark up data tables correctly.

Tables are reserved for tabular data, not page layouts. Designate row and column headers with <th>. Use the <thead>, <tbody> and <tfoot> tags. Use the scope or id/header attribute to associate data with the appropriate header(s).

13. Use descriptive text for links.

Screen reading software scans a page for links and provides the list to the user. Numerous "click here" links are useless when separated from their context. It's better to use descriptive text such as "Explore our academic programs."

14. Identify links to PDFs, Word documents, Excel spreadsheets, etc.

This is important for screen reader users, but it's also useful for sighted viewers who may not want to open another software program. Include the document designation in the link so that it will be read by screen readers:

```
<a href="documents/uc_mission.pdf">UC Mission Statement (pdf)</a>
```

15. Open off-site links in the same browser window.

Modern browsers offer tabbed browsing, so a user has control over whether to open a link in a new tab or new window. By forcing an off-site link to open in a new browser window, you are taking away that control from the user.

Additionally, some older versions of screen reading software do not notify users when they open a new browser window. Users won't know what has happened until they try to use the "Back" button and find they can't go back. Because screen reading software can be quite expensive (up to \$1,100 for a single JAWS license!), an upgrade to a new version may not be feasible.

16. Don't use frames. If you must use them, correctly mark up frames and iframes.

Frames are not inaccessible to modern screen readers, but they do introduce accessibility issues. Add title attributes to all <frame> and <iframe> tags. If you must use frames, provide equivalent content (or a link to it) within the <noframes> tags. Similarly, when you're using iframes, provide equivalent content (or a link to it) within the <iframe> tags. This allows a user to access the content directly if his/her browser doesn't support frames or iframes.

17. Give a user sufficient time to respond to timed content.

When a timed response is required, alert users and give them sufficient time to respond. People with mobility issues may have difficulty with limited time constraints.

18. Don't create flickering pages or multimedia.

To prevent seizures among optically sensitive users, don't create any pages or multimedia that flicker between 2 to 55 times per second.

19. Validate HTML and CSS.

IT accessibility begins with clean, valid code, so be sure to check your code with the [W3C X/HTML Validator](#) and [W3C CSS Validator](#).

20. Check for IT accessibility compliance.

Use tools such as [Total Validator](#) and the [WAVE Toolbar for Firefox](#) to verify that your Web page is accessible.

Sources

- WebAIM – [Creating Accessible JavaScript](#)
- WebAIM – [WebAIM WCAG 2.0 Checklist](#)
- University of California – [Resources for Developing Accessible Web Sites](#)