



DIGITAL ACCESSIBILITY BY WECO INTRODUCTION TO WCAG 2.1 AA

PART THREE: PAGE CONSISTENCY

WeCo Accessibility Learning Takeaway

What is page consistency?

Page consistency focuses on using consistent layout and design so that users have a comfortable and satisfying experience when navigating websites, software, etc. For a user living with a disability, this will mean not having to “figure out/relearn” how to navigate and find information as they move between different pages/screens.

Maintaining consistent layout

SUCCESS CRITERION	DESCRIPTION
WCAG 3.2.3 (Level AA)	Consistent presentation and layout should be used for navigation links and menus within web sites, software, etc. This allows users with sight-related disabilities and users who rely on spatial memory to easily navigate between pages/screens.

RECOMMENDATION:

- Use a consistent presentation and layout for navigation links and menus within web pages, software, etc.

SUCCESS CRITERION	DESCRIPTION
WCAG 3.2.4 (Level AA)	Ensure that interactive elements with the same functionality are identified consistently. This will prevent confusion for users with cognitive disabilities and users of assistive technology.

RECOMMENDATIONS:

- Ensure elements that appear repeatedly are labeled and named consistently.
- Ensure all link text is consistent for links leading to the same destination or remove the duplicate links.

WeCo tips and recommendations

Accessibility policy and contact information

Having an accessibility policy that includes contact information for accessibility issues is also recommended by the WCAG. Accessibility policies should include information about the testing, and possibly training, your organization has completed in an effort to improve accessibility.

RECOMMENDATION:

- Provide a policy outlining a commitment to website accessibility for disabled users. Include contact information for a person/department to assist with issues. For more information, visit [WeCo's Accessibility Policy Development page](#).

Consistent code and content labels

SUCCESS CRITERION	DESCRIPTION
WCAG 2.5.3 (Level A)	Controls that have visible text labels must have the same text within their programmatic name. Speech recognition software users interact with elements by speaking their visible text labels, which isn't possible when the programmatic name does not include the visible text.

RECOMMENDATION:

- For each interactive component that includes a visible text label, make sure the accessible name includes or matches the visible label text.

SUCCESS CRITERION	DESCRIPTION
WCAG 3.1.1 (Level A)	Having the language of the page identified ensures that screen readers use correct pronunciation rules, allows browsers to display characters correctly, and allows media players to display captions in the correct language.

RECOMMENDATION:

- Indicate the language of web pages using the HTML "lang" attribute (e.g., `<html lang="en">`).

Following programming language rules

SUCCESS CRITERION	DESCRIPTION
WCAG 4.1.1 (Level A)	Ensure that code conforms to the appropriate programming language specifications. Code that does not conform can cause assistive technology to process the content incorrectly.

RECOMMENDATIONS:

- Ensure that HTML/XHTML validation/parsing errors are avoided. Check code at the [W3C Validator](#).
- Ensure that HTML elements have complete start (< >) and end (</ >) tags.
- Check that HTML elements don't contain duplicate attributes.
- Provide an "alt" attribute on all images/graphics. According to HTML specifications, elements must have an "alt" attribute, even if they are considered decorative.

SUCCESS CRITERION	DESCRIPTION
WCAG 4.1.2 (Level A)	Assistive technology should be able to gather information about and interact with user interface components. For assistive technology users, when using custom controls or interface elements that are programmed with a different role and/or function than usual, controls may not convey important information and/or users may be unable to interact with them.

RECOMMENDATIONS:

- Code elements based on their intended function. For example, code buttons using <button> when the element is used to submit data or perform an on-screen action.
- Ensure that all inputs in forms are associated with corresponding label elements.
- Provide an expand/collapse state for elements that need it. Provide an "aria-expanded" attribute on each expand/collapse button. Use JavaScript to toggle its value between true/false, depending on whether the section is expanded. View the [Inclusive Components on collapsible sections](#) article for explanations and examples. Begin at "The adapted markup" section.
- For elements which convey a selection state to users, ensure that any change in, or presence of, the state is programmatically exposed to assistive technology. Use JavaScript to dynamically add an "aria-current" attribute on the element conveying the state. Set the value of the "aria-current" attribute to one of the following, based on the situation: "page", "step", "location", "date", "time", or "true".

- Tabs must be implemented with the correct keyboard support, roles, and selection state. Refer to [section 3.22, Tabs](#) of the WAI-ARIA Authoring Practices Document at the W3C. Implement the keyboard support as described under the “Keyboard Interaction” heading, and ARIA attributes as described under the “WAI-ARIA Roles, States, and Properties” heading.
- Code modals using appropriate dialog semantics. Refer to the [WAI-ARIA Authoring Practices 1.1 document](#) for details.
- Use the <a> element for all links. Links should always have an href attribute, even when used for same page links. Without an href attribute, the link will not be properly exposed to assistive technology (e.g., a link that uses an onclick event, in place of a href attribute).
- The title attribute has numerous issues and should not be used if the information being provided is important for all users to access.

Communicating with the user

SUCCESS CRITERION	DESCRIPTION
WCAG 4.1.3 (Level AA)	Status messages make users aware of changes in content that are not given focus. When appropriate roles or properties are assigned to status messages, the changes are conveyed by screen readers to users.

RECOMMENDATIONS:

- Use the ARIA status role on the element serving as the status message.
- Ensure that status messages meet the following criteria:
 - The message "provides information to the user on the success or results of an action, on the waiting state of an application, on the progress of a process, or on the existence of errors".
 - The message is not delivered via a change in context.

SUCCESS CRITERION	DESCRIPTION
WCAG 2.2.1 (Level A)	Provide users with the ability to turn off, adjust, or extend the time limit if web functions are time-dependent (e.g., completing a purchase within a specific timeframe), as it will be difficult for some users to perform the required action before a time limit occurs.

RECOMMENDATIONS:

- Use the ARIA status role on the element serving as the status message so that assistive technology can automatically convey status message to users.
- This is not a requirement for real-time events (e.g., an online auction), where the time limit is absolutely required, or if the time limit is longer than 20 hours.

Checking contrast ratio

SUCCESS CRITERION	DESCRIPTION
WCAG 1.4.3 (Level AA)	Not having enough contrast between text color and the surrounding background color causes users with low vision to have difficulty when reading text on web pages.

RECOMMENDATIONS:

- Use a light background color with dark text or a dark background color with light text. You can use “WebAIM’s Color Contrast Checker” found in [WeCo’s Free Accessibility Library](#) to verify your color choices.
- Large text (at least 18 point or 14 point bold) contrast ratio must be at least 3:1.
- Text within decorative elements/logos does not have contrast ratio requirements.

SUCCESS CRITERION	DESCRIPTION
WCAG 1.4.11 (Level AA)	Visual details needed to identify graphics and active user interface controls and their states must have a contrast ratio of at least 3:1 contrast against adjacent colors. (This includes, but is not limited to buttons, form fields, focus indicators, and selected state indicators.) Users with low vision often have difficulty identifying graphics and interactive components that have insufficient contrast.

RECOMMENDATIONS:

- The following must have a contrast ratio of at least 3:1 against surrounding colors:
 - User interface components
 - Visual information required to identify user interface components and states, except for inactive components or where the appearance of the component is determined by the user agent and not modified by the author.
 - Graphical objects: parts of graphics required to understand the content, except when a particular presentation of graphics is essential to the information being conveyed.