Accessibility

- The four principles of web accessibility
  - Perceivable
  - Operable
  - Understandable
  - Robust
- Perceivable and understandable
  - Color and color contrast
  - Resizable text
  - Images
    - Non-decorative images
    - Decorative images
    - Featured images
- Operable
  - Controls
  - Headings
  - Links
    - Link text
    - Link underlining and styling link states
    - Skip Links
  - Forms
    - On Form Submission
- Robust
- Resources

A WordPress theme should generate pages everyone can use, including those who cannot see or use a mouse.

To create an accessible theme, you need to have knowledge of web standards for HTML, CSS, and JavaScript.
You also need to be aware of best practices for web accessibility and have basic knowledge of the

Web Content Accessibility Guidelines, WCAG.

To succeed in making your theme accessible, accessibility should be considered from the start of your project as part of your project specification. By making the right decisions from the beginning, you can avoid making last-minute adjustments that can be costly, time-consuming, and of low quality.

The Accessibility Team have documented many of the best practices and resources for development, design, and content in their handbook. In the handbook, you can also learn about testing for accessibility.

The Themes Team has two sets of accessibility requirements for themes submitted to the WordPress.org theme directory:

- Requirements for all themes
- Requirements for using the "'accessibility-ready" tag

The accessibility-ready requirements are based on WCAG but adapted for WordPress themes.

"Accessibility Ready" does **not** mean that the theme meets the WCAG guidelines AA-level. It means that the theme reaches the minimum standards that the theme review team has set.

The four principles of web accessibility

Although web accessibility can be a complex subject, it boils down to only four principles. The content must be:

Perceivable

Content must be available to all. It should not depend on a specific device or browser or require a specific physical sense, such as sight or sound.

Operable

Users must be able to move around and operate the final site effectively, whether they're using a keyboard, a mouse, or assistive technology.

Understandable

The content should be presented in a manner that supports understanding, including supporting the construction of a mental model of the site for screen reader users. Similarly, the site's operation (navigation menus, links, forms, etc.) should be easily understandable. Building a theme that incorporates known user behaviors (such as underlining links within the main content area) helps in this respect.

Robust

Content must be equally available across a wide range of user agents. Disabled users may employ a range of hardware and software solutions (commonly referred to as "assistive technology") to allow them to access the Web—including screen reading & voice recognition software, braille readers, and switches (single input devices).

A theme designed with these four principles in mind eases the creation of an accessible website.

Perceivable and understandable

Color and color contrast

Having a high enough color contrast between background and foreground colors makes content easier to read. Theme authors must ensure that all background/foreground color contrasts for plain content text are within the level AA contrast ratio (4.5:1) specified in the Web Content Accessibility Guidelines (WCAG) 2.0 for color luminosity.

- Color may not be the only way to identify controls, links in text content, or error messages.
- If there is no text decoration on linked text, there must be a 3:1 color contrast between the link text color and the surrounding non-link text color, in addition to the other color contrast requirements.

There are separate requirements for links in content (links surrounded by other text) and links that are grouped together in navigation menus. Groups of links do not need to be underlined if it is obvious from the context that they are links.

Best practices for the use of color

Accessibility-ready requirements for contrast

Resizable text

Users have many means to resize text, including browser settings. All content and functionality must remain available if the user resizes the text up to 200% of the original size.

- Resizing text must not trigger multi-dimensional scrolling
- Enlarged texts must not cause overflows or overlaps

To avoid these issues, it is recommended to:

- Use a relative unit for font sizes and line heights
- Test your theme in different browsers and screen widths

Best practices for font sizes and resized text

Images

Non-decorative images

Example:

- A header image that replaces header text
- Images used in place of text for navigation

Non-decorative images included with `img` elements should have an `alt` attribute.

Decorative images

Example:

- A header image used alongside header text
- Images and icons accompanying navigation text links

When possible, decorative images should be included using CSS.

- Decorative images included with `img` elements should have an empty `alt` attribute: `alt=""`.
- Decorative images that are displayed together with text should be hidden from screen readers using `aria-hidden`.

Featured images

The alt attribute for featured images is defined in the media library.

- If the featured image is unlinked, the alt attribute should describe the image
- If the featured image is linked to a post, the alt text should use the post title

Best practices for alt texts

Accessibility-ready requirement for images

Operable

Controls

All controls must be usable with the keyboard on all screen sizes, including but not limited to buttons for opening and closing menus, submenus, any type of dialog, modal, and popup.

Best practices for dialog modals

Accessibility-ready requirements for controls

Headings

Headings are an essential way of breaking content down into logical sub-sections. Heading levels indicate the importance of the content. Screen reader users can scan the contents of a page by reading the headings, and navigating to a section via its heading. That is why it is important that headings are used logically and not for presentational purposes.

Best practices for Heading structure in theme development

Accessibility-ready requirements for headings

Links

Link text

Link text should describe the resource that it links to, even when the text is read out of context. Some assistive software scans a page for links and presents them to the user as a simple list. In these situations, all the links will be read out of context. So it is important the text used in a link is descriptive. Bare URLs should never be used as links.

Good link texts

Accessibility-ready requirements: Avoiding repetitive Read More or Continue reading texts.

Link underlining and styling link states

Generally speaking, links should be underlined if they are outside navigation menus and lists. Using color alone to distinguish links is insufficient as not everyone can perceive color.

Users must be able to tell if a text is linked if they are hovering over a link, if they are focusing on a link, and if they have already visited a link. The default browser style for the focus state should not be removed unless replaced with a more visible focus style.

Accessibility-ready requirements for underlining links in the content

Skip Links

Skip links provide a mechanism that enables users to navigate directly to content or navigation on entering any given page. For example, a skip to content link might allow a user to skip a header area and go directly to the main content.

In designs with multiple menus and content areas, multiple skip links can be used:

- Skip to the main navigation
- Skip to content
- Skip to footer

These links may be positioned off-screen initially using an appropriate CSS technique but should remain available to screen reader users and be visible on focus for sighted keyboard navigators.

Best practices for Skip Links

Accessibility-ready requirements for Skip Links

Forms

- Provide enough information for the user to be able to complete the form.
- All input fields must have a label. A placeholder text is not a replacement for the label. Input fields must also have a visible focus style.
- Group controls that belong together, for example, a set of related checkboxes, with a `fieldset` and `legend`.
- Make sure that the tab order in the form matches the visual order of the input fields: Do not move focus unexpectedly or skip past input fields.

On Form Submission

Post-submission responses—including any error messages—should always be perceivable. If possible, error messages should be generated at the top of the post-submission page so the user is immediately aware of any issues. Error messages should also make sense when read out of context.

Best practices for forms

Accessibility-ready requirements for forms

Robust

Use the HTML element that is the best match for the content. Use buttons when performing an action, and links when navigating to a part of a page or a new page.

- The content of the website should be available even if the user disables both JavaScript and CSS
- Establish which browsers your theme supports and test your theme with these browsers on different screen sizes

Resources

Make WordPress Accessible is the official blog for the WordPress Accessibility Team, dedicated to making WordPress accessible to as many people as possible. Anyone can join in the discussions, bug scrubs, and meetings. You can also follow discussions via email or subscribe to feeds for posts and comments.

- Test for web accessibility
- Development tools
- WCAG 2 (external link)

Changelog:

- **Rewritten and published** 2023-02-16

□□□ #2
□ □□□ □□□ 2026-08-29 15:36:32 UTC
□ □□□ □□□ 2026-08-29 15:36:32 UTC