

Make WordPress Accessible

The [WP Accessibility plugin](#) — developed by [Joe Dolson](#) — can help non-technical site owners correct a variety of technical accessibility issues within many WordPress themes.

The current version of the plugin can:

- [Remove redundant title attributes](#) from page lists, category lists, and archive menus.
- [Enable skip links](#) with WebKit support by enqueueing JavaScript support for moving keyboard focus.
- Add skip links with user-defined targets.
- [Add language and text direction](#) attributes to your HTML attribute
- [Remove the target attribute](#) from links.
- [Force a search page error](#) when a search is made with an empty text string.
- [Remove tabindex](#) from elements that are focusable.
- [Strip title attributes](#) from images inserted into content.
- [Add post titles to standard “read more” links](#).
- [Add a :focus outline](#) to focusable elements.
- [Toggle for high-contrast and large font stylesheets](#)
- [Custom admin stylesheet](#)
- [Provide color contrast testing tool](#) to compare colors against WCAG contrast standards

However, whilst the plugin can correct some issues within themes, it **cannot** correct every problem. So it is **not** a substitute for [good web accessibility practice](#). There is still a great deal you can do to make your site more accessible when creating your content. But it may help those who lack the HTML skills necessary by allowing them to focus on creating accessible content.

And do remember that creating an accessible site will immediately increase your potential audience as well as facilitating search engine indexing. All in all, a win-in situation!

Removing title attributes

By default, WordPress adds title attributes to all system-generated links — navigation menus, lists of posts, etc. These title attributes have exactly the same value as the text of the link — if the text of the link is “Joe’s Blue Widgets”, then the title attribute is also “Joe’s Blue Widgets”. In the best case scenario, where a screen reader is set up to ignore title attributes, having this set makes little difference. In the worst case scenario, where a screen reader reads all attributes, it’s awful: the screen reader reads the title attribute and the link text, meaning that every link is, effectively, read twice. This is an enormous cognitive overload that benefits nobody. The purpose of the title attribute is to add context to a link that otherwise does not have sufficient context: but in almost all cases, the WordPress use does not meet that criteria. [Further reading about title attributes and accessibility](#)

Skiplinks.

Skiplinks are a means to help users of screenreaders jump from the beginning of a page to another section of the page. Visual users can scan a page quickly to identify the larger sections and skip over large blocks of text or links, but users of screen readers can’t easily do that. Skiplinks provide a means to leap from the very top of a page to an alternate section of that page — one principle use being to leap over the main navigation to the main page content. [Further reading about skip links](#).

Add site language and text direction to HTML element.

Screen readers use the site’s declared language to figure out what pronunciation rules to use when reading the content. If no language is declared, then the screen reader will generally use the screen-reader user’s installation’s language or try to guess: e.g., if a French user of a screen reader visits your page, and no HTML language is declared, it will either read as if the text was written in French, or attempt to guess the actual language and pronunciation. Declaring the actual default language greatly improves comprehension of the text.

As problems go, this is relatively rare as a problem in a theme: what is actually a bigger problem is sections of text that are not in the primary language of the site. This plug-in doesn’t currently handle changing languages in the middle of content. [Further reading about document language](#).

Remove target attribute from links

Links opened in new windows can break web site flow (creating a change of context) and disorient users who don’t realize they’ve moved to a new window. Also, it takes away the option of opening in the same window/tab: all browsers have an option to open links in a new tab, few have an option to open in the same tab. [Further reading about changes of context](#)

Force search error on empty search submission

The default WordPress behavior when a search is submitted with an empty search field is to simply return the main index with no notice at all. This is disorienting and confusing. The plug-in will change this so that the submission of an empty search field will return the same as a search with no results, which makes a great deal more sense in terms of user feedback. [Further reading on form feedback.](#)

Remove tabindex from focusable elements

Many (primarily older) themes had tabindex defined for comment input fields, in particular. Tabindex was an early concept for an accessibility feature for HTML, but was quickly demonstrated to cause more problems than it solved. What tabindex does is force keyboard navigation to follow a particular sequence according to the values defined in tabindex attributes. Removing those attributes usually makes keyboard navigation much more linear and predictable. The intention behind tabindex was to create a logical tab sequence when it would otherwise have been unnatural; but in practice, that required absolutely insane degrees of maintenance to use in a real-life web site. This option removes the attribute from standard focusable elements: links and form fields. It does *not* remove it from other fields, because a common technique to grant keyboard focusability to normally non-focusable elements is to define a negative tabindex: this places the element into the focus sequence but does not force it into a particular place in that sequence. [Further reading about focus order.](#) [Further reading about tabindex.](#)

Remove title attribute from inserted post images/featured images.

Similar to the title attribute above: WordPress automatically adds the title element to inserted images. This automatically strips it out; which is primarily a labor saver for people who would otherwise be doing this manually.

Add post title to “more” links

A common pattern for WordPress in many views is to show an excerpt of your post with a link to read the entire post. By default, this link is just plain text “more” or “continue reading” or an equivalent. This is problematic for accessibility because it means that there are many links on the same page with the same link text: for accessibility, all links should make sense without requiring context, to grant screen readers the ability to scan the page and have information. This plug-in adds the title of the post into this pattern, so that instead of hearing “Continue Reading. Continue Reading. Continue Reading.”, the user will hear something like “Continue Reading About The Storm. Continue Reading The Storm Aftermath. Continue Reading Cleaning Up”, and be able to pick the article they actually want to read from that set of links. [Further reading about identifying link purpose.](#)

Add outline to elements on keyboard focus.

Many designs do not provide a design for the view of links or other focusable fields when a keyboard user brings focus to that element. This makes it very difficult for visual users who are keyboard dependent to keep track of where they are. A common pattern for this is to change the appearance of the element on keyboard focus; and this item adds an outline. Outline is rarely defined in most designs, so that ensures that the definition by this plug-in will almost always work and be visible; the same can’t be said for text decorations like underline, since that may be the default state for links, and therefore would not be a change indicator. [Further reading about keyboard focus.](#)

Toggle for high-contrast and large font stylesheets

While not a solution in itself, if your theme has low contrast issues or small fonts, using this custom control to toggle high-contrast or large font modes can be useful. All styles can be customized or extended within your theme directory, since default states for these tools may not be suitable for your site. The accessibility toolbar is based on the original written by [Chris Rodriguez.](#)

Custom admin stylesheet

The WordPress admin has some miscellaneous accessibility issues, which this stylesheet will correct. It can also be modified to customize according to your specific needs.

Color Contrast Tester

This tool doesn’t make any active changes to your web site, but allows you to easily test whether a set of colors you’re considering for customization of your site meet the standards set by WCAG 2 for color contrast. [Further reading about color contrast.](#)



Joe Dolson 11:45 am on October 26, 2012

Thanks for highlighting my plug-in, Mel! I appreciate that. I hope that it will be helpful in adding an additional level of accessibility to existing WordPress sites. I’d like to note, as well, that all features can be disabled, so you don’t end up duplicating issues that your existing theme is already taking care of.

About
Blog
Hosting
Jobs

Support
Get Involved
Learn

Showcase
Plugins
Themes
Ideas

WordCamp
WordPress.TV
BuddyPress
bbPress

WordPress.com
Matt
Privacy
License / GPLv2

Follow @WordPress

Like 752k

