

# Web Accessibility Best Practices

For Drupal Content  
Editors



# About the Author



**Michelle Williamson**

Accessibility Lead

Michelle Williamson, Mediacurrent's resident expert on accessibility, has completed specialized training in website accessibility, multimedia accessibility, and accessibility in the Word -> InDesign -> PDF publishing workflow. She is knowledgeable in Section 508 and WCAG 2.0 compliance and the use of the Accessibility API, WAI-ARIA.

Michelle holds a Certificate of Accessibility in Information Technology from the University of Southern Maine. Her final project for the program focused on accessibility in Drupal.

# Websites for Everyone

You've just been handed a website and it's your responsibility to add content. Oh, by the way, one of the requirements of the site when it was being built was that it be made accessible to the widest audience possible—and it's your job to maintain that accessibility.

You may think that your site's accessibility depends only on whether or not accessibility principles were used in the build process by the developers.

In fact, quite the opposite is true. As a content editor, you can have a big impact on the accessibility of your Drupal site.



# Understanding Web Accessibility

*Simply put, web accessibility means making your site available to all users, regardless of their (dis)abilities.*

It affects a wide range of users:

- Blind users who rely on a screen reader
- Those with low vision who need to greatly increase text size
- Users with mobility impairments who are unable to use a mouse
- A person who broke his wrist and needs to use the mouse in his non-dominant hand

...and millions of others.



# Why is Accessibility So Important?

Following web accessibility best practices improves usability for everyone who visits your site, and provides the following benefits:

1. Opens your website up to a wider audience and customer base.
2. Boosts search engine rankings. Google rankings favor accessible, descriptive websites.
3. Increases usability on mobile devices.
4. Earns respect and often, repeat business. When a person finds a site that has the information they are looking for and the information is accessible to them where others are not, they may bookmark the website and promote it to others.

# By the Numbers



Nearly **20%** of **AMERICANS** live with a disability.



**1.3 BILLION** disabled worldwide



plus their **2.3 BILLION** family, friends, caregivers and colleagues.

**\$8**  
trillion



in **DISPOSABLE**  
income

Source: Forbes Magazine. Business's Next Frontier: People With Disabilities



# Best Practices

Here are five basic and easy-to-implement best practices that every content editor should know.

# Best Practices

## Use Alternative Text for Images

Adding alternative text to images may be the single biggest thing you do to enhance the accessibility of your site. Alt text informs users with vision impairments whether an image is important, and describes what the image is about. Consider the following:

- Every image should have an alt attribute, even if it's null (alt="").
- Alt text should be succinct (no more than 140-characters or the length of one tweet).
- Avoid phrases like “image of...” or “picture of...” A screen reader will automatically identify images.
- Don't use CSS to display images unless they are purely decorative. Alt attributes cannot be added to CSS images.
- Read [Describing Images for Improved Accessibility](#) for examples of appropriate alt text in action.

# Best Practices

## Assign Unique and Descriptive Page Titles

Bear in mind that a good page title helps the user to surmise whether the content they seek can be found on a particular page. For example, “Describing Images for Improved Web Accessibility” is a better title than just “Describing Images.”

## Keep it in HTML

When possible, avoid uploading other document types (PDF, Powerpoint, Word doc, etc.) in place of page content. There are some instances when this is useful (for example, a downloadable PDF flyer meant to be printed). However, many times, the PDF or Word doc content should be converted to HTML format. Non-HTML documents (particularly PDFs) have their own methods of being made accessible. So, unless you’re also building accessibility into your non-HTML documents, convert the contents to HTML.

# Best Practices

## Identify non-HTML Content and Large Downloads

For those times when you absolutely do need to provide non-HTML content, be sure to put the extension in parenthesis after the link text and include a file size for large downloads. Be sure to keep this info inside the linked text so that a screen reader will read the file type/size along with the link.

For example: [\*Printable map of the Pine Grove Campground \(PDF, 22 Megs\)\*](#).

## No Yelling!

Avoid entering titles and other text fields in all caps for emphasis; it's poor online etiquette and it creates a pattern that is difficult to skim. Also, some screen readers will read off the individual letters of words typed in all caps instead of the words themselves.



# WYSIWYG Editor

Our top tips for formatting web content in the Drupal WYSIWYG (What You See Is What You Get) editor.

# Working in the WYSIWYG

## Impact

The biggest impact a content editor has on accessibility is in the free-form text area fields, or WYSIWYG fields as they're better known. Essentially, it's like working in a Microsoft Word document; you style and structure the text using buttons, dropdowns, and dialog boxes. The way you style and structure your content in the WYSIWYG can make or break the accessibility of your site.

## Learn Basic HTML

WYSIWYGs can be finicky. Knowing how to edit directly in the HTML source code instead of deleting all the formatting you've already done and starting over can save your sanity. Spend a day or two reviewing an online HTML primer, pick up a "learn HTML in 24 hours" book, or take your favorite computer geek out to lunch. Once you master the basic concepts of opening and closing tags and attributes, your life as a content editor will suddenly become a whole lot easier.

# WYSIWYG Editing Tips – Formatting

## Do Your Formatting in the WYSIWYG

Imagine that you've spent hours writing and formatting content in a Word document. Finally, it's ready to post on your website. Not so fast! Copying and pasting content from other sources such as Word can bring along a lot of undesired cruft – behind the scenes formatting code that is proprietary to the program it's being copied from.

The best practice is to paste content from another application into the WYSIWYG using either the “Paste from Word” or “Paste as Text” button to strip out the proprietary formatting. This way, you can format within the WYSIWYG and get proper HTML markup that won't interfere with the accessibility of your content.

# WYSIWYG Editing Tips - Structure

## Use Proper Structure

Part of formatting your content in the WYSIWYG is chunking your content. To achieve this, you must create a hierarchy within the content:

- Use heading tags (h2-h6) to label each section **in proper order** instead of changing the font color and size to make it appear as a heading.
- Use the list button to add bullets or numbering to a list of items rather than placing a bullet image in front of each line.
- Create paragraphs by hitting enter or return at the end of a paragraph.

Why is this important? Because a person who is blind often browses a page by using this hierarchy. It's a sighted person's equivalent of scanning a page with their eyes. A screen reader allows the blind person to bring up and easily access a list of all headings or a list of all lists or a list of all links, and so on.

# WYSIWYG Editing Tips – Links

## Do Not Underline Non-linked Text

If an underline button is available in your WYSIWYG configuration, forget it exists. Links are the only element that should ever be underlined and that should be done via CSS, not with an underline button. Text that is not linked but is underlined for emphasis leads users to believe it is a link and when they move their mouse to the text or try to focus on it with a keyboard, and it's not clickable, it causes confusion and frustration.

## Use Descriptive Link Text

As mentioned above, one of the ways screen reader users browse a page is by bringing up a list of all the links on the page. Always choose link text that is meaningful out of context.

For instance, instead of “[Click here](#) to learn more about accessibility,” use “Learn more about [why accessibility matters](#).” Avoid using generic link text such as “Read more” or “Click here”, as it fails to provide helpful context. Using the website URL as the link text is often insufficient as well because screen readers read URLs letter by letter.

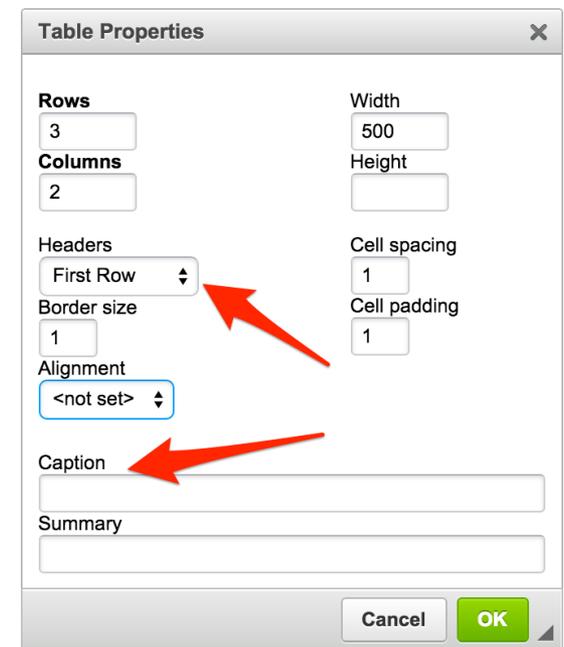
# WYSIWYG Editing Tips – Tables

## Use Tables Only for Tabular Data

It used to be all the rage to format a page using tables and led to tables nested in tables nested in tables to get the proper placement. That was back in like 1998. We've come a long way, thankfully. Use tables only for what they're designed for – tabular data. And be sure to set the header column/row along with a descriptive caption for the table.

## Alert User of New Windows

When opening a link in a new window/tab, alert the user by adding “(new window)” to the link text. Taking a user to a brand new window or tab can be disorienting for screen reader users and people with low vision.



The image shows a 'Table Properties' dialog box with the following fields and controls:

- Rows:** Input field with value 3.
- Columns:** Input field with value 2.
- Width:** Input field with value 500.
- Height:** Input field.
- Headers:** Dropdown menu with 'First Row' selected.
- Border size:** Input field with value 1.
- Alignment:** Dropdown menu with '<not set>' selected.
- Cell spacing:** Input field with value 1.
- Cell padding:** Input field with value 1.
- Caption:** Text input field.
- Summary:** Text input field.
- Buttons:** 'Cancel' and 'OK' buttons.

Red arrows point to the 'First Row' dropdown, the 'Caption' text field, and the 'OK' button.



# Testing

Ready to put your website's accessibility to the test?  
The right tools will help.

# Test Your Content

Finally, after you've followed all the rules above, run your page through an accessibility checker. Here are 3 ways to test your content as you're creating it in the WYSIWYG:

1. For Drupal sites, the [Drupal Accessibility Module](#) provides a new button in the CKEditor WYSIWYG that enables you to check accessibility as you're working. One drawback is that it uses an old version of the Quail library.
2. A more up-to-date option is [CKSource's Accessibility Checker](#). It also puts a button in the WYSIWYG editor and allows you to check accessibility as you work, but is rather pricey.
3. Another good way to test your content is to run it through Webaim's Wave Toolbar. This will check the entire page, however, and not just individual fields. So you may see a lot of errors that you can't do anything about. Webaim offers their toolbar in two formats:
  - The [Online WAVE Tool](#)
  - The [WAVE Toolbar Browser Plugin](#)

# Accessibility Checklist for Content Editors

- Set descriptive page titles.
- Avoid using all caps except for acronyms.
- Avoid using non-html document types unless absolutely necessary.
- Identify non-html document type and document size inside links.
- Set alt text on images.
- Learn basic HTML.
- Use Paste as Text and Paste from Word buttons and format in the WYSIWYG.
- Chunk content using proper markup and structure.
- Avoid underlining non-link text.
- Alert the user to links opening in a new tab/window.
- Set descriptive link text.
- Use tables only for tabular data.
- Test your content with an accessibility checker.

# Thank You

[Mediacurrent  
Resources](#)

[Contact Us](#)