Setting Up an Accessible Web Page: Filtering Data (A UX Accessibility Blog Post)

This UX Accessibility Blog Series, “Setting Up An Accessible Web Page,” will walk you through the basic elements of an accessible web page in hopes that you can take what you’ve learned an apply it to more complex pages. This blog series is loosely based on Heydon Pickering’s book [“Inclusive Design Patterns: Coding Accessibility Into Web Design.”](https://www.smashingmagazine.com/inclusive-design-patterns/)

Filtering data is a great way to give users control of the information they want to see. This helps many different types of users all at once. Sighted users have the ability to narrow down a page of data results from, for instance, 2000 results to 200 results, making it much easier to find the specific content they are looking for. Visually impaired users use filters for the same reason. Having a screen reader read out 2000 results with no way to filter is no doubt cumbersome.

It is also important to give users multiple ways to filter through. Giving users this extra mechanism over what they’re able to see helps them have a sense of control over the information they are consuming.

**Waiting Spinner**

Anytime the user is waiting on data to load, there needs to be a waiting spinner. Waiting spinners are a great, simple way of showing sighted users that there will be information displayed momentarily. However, screen reader users are at a disadvantage. They’re typically left in the dark on the status of their action or if it went through at all. Make sure that you use WAI-ARIA’s live regions to tell the screen reader user that everything is fine and the data will be available soon.

To make this truly inclusive, make sure the screen reader announces both when the content is loading and when it is done. For example:

<div class=”sr-only” aria-live=”assertive”>Please wait. Populating table.</div>

<div class=”sr-only” aria-live=”assertive”>Data population complete.</div>

Another helpful addition could be to add a statement with the number of entries loaded at the end of your “Data population complete.” statement.

**aria-live**

The aria-live attribute is used to set the priority of the live region for screen readers. aria-live can be set to off, polite, or assertive. The polite value tells the screen reader to wait until it has finished reading the rest of the page before coming back to the section that has updated. Usually, aria-live=”polite” is the best option since it reduces the chance that you’ll startle or confuse the user by interrupting the screen reader. The assertive value interrupts the screen reader’s set script of the page to alert the user to a change. When using aria-live with a waiting spinner, it is best to use the assertive value since it will immediately alert the user to the change.

To maximize browser compatibility, some accessibility experts recommend using both aria-live=”assertive” and role=”alert”. This can be problematic, according to the Mozilla Developer Network’s “ARIA Live Regions” article, as it can cause double speaking issues for VoiceOver in iOS.