Setting Up an Accessible Web Page: Keyboard Focus (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/)

It’s important for your users to know where they are on your page at all times. This is especially true for your keyboard users, who often find it difficult to determine their location on a webpage.

Some elements, such as those that use divs, do not automatically receive keyboard focus or allow the user to navigate to them via the keyboard. Elements that are clickable but wouldn’t receive focus normally, such as infographic boxes in Ensemble, need to have the tabindex=”0” attribute added to them. *Interactive elements should always be accessible.* This way, keyboard users can navigate to them, see that they have received focus, and hit enter to navigate to the same place a mouse user would be able to navigate to.

[pictures]

For elements that are mostly text, such as paragraphs, keyboard focus isn’t necessary, although some users may find it helpful. To make sure blocks of text receive focus, add the tabindex=”0” attribute to your paragraph tag. This will addition will only enhance the experience of your keyboard users without adding any unnecessary visual noise for your mouse users.

Interactive elements that are written using semantic HTML have keyboard focus features baked in. Some examples of these would be <input type=”[text,email,tel,etc/checkbox/radio]”>, <button>, <select>, and <textarea>. Even though these elements are typically accessible, it’s important to always test your page for keyboard accessibility by tabbing through and making sure the tab order is logical.

Browsers typically implement their own focus styles, which are usually fuzzy blue halos or dotted lines. I like to use the browser default, but there are other ways to add focus styles to make focus more clear. It is important, however, to NEVER remove or add styles that contradict the browser’s default implementation of focus styles. These are very important and help a variety of users.

An example of an improved focus style would be gov.uk’s implementation. They give the focused element a different background color from the rest of the text on the page.

[gif]