Setting Up an Accessible Web Page: Making Forms Accessible (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/)

We’ve talked about what needs to be done to make forms accessible multiple times before, but since the relationship between these elements and accessibility/inclusivity is an important conversation to have, we’ll go over them (maybe in a bit more detail) again.

**Labels and Inputs**

Labels are used to identify all types of inputs and other form controls (such as <select>s or <textarea>s). Browsers allow the labels to become part of the clickable area that can be used to allow the input to receive focus. Labels also insure that assistive technologies correctly associate the label with the corresponding input field.

To associate a label with an input field, just set the for attribute in the label element to the id of the input it corresponds with. For example:

<label for=”firstName”>First Name:</label>

<input type=”text” class=”form-control” name=”textbox” id=”firstName” />

<label for=”pizza”>Pizza Type:</label>

<select id=”pizza” name=”select”>

<optgroup label=”Veggie Friendly”>

<option value=”1”>Cheese</option>

<option value=”3”>Veggie</option>

</optgroup>

<optgroup label=”Meaty Pizzas”>

<option value=”2”>Pepperoni</option>

<option value=”4”>Sausage</option>

</optgroup>

</select>

Text inputs can have an attribute called placeholder that you can use to give your users an example of what they should type in the input field. Never use the placeholder attribute instead of a label for the field. Using the placeholder attribute as the text input label can cause the user confusion since the placeholder disappears as soon as the user starts typing or as soon as the field receives focus.

**Radio Buttons and Checkboxes**

The best way to make radio buttons and checkboxes accessible is to use the <fieldset> and <legend> tags. The fieldset tag surrounds the group of radio buttons or checkboxes to show that those are the options for this particular group. The legend tag gives the fieldset a name to show the user what they’re picking. Screen readers repeat the legend text for each option in the fieldset, so it is important to keep your legend text as concise as possible.

<fieldset>

<legend>Pizza Type</legend>

<input id=”cheese” type=”radio” name=”pizza” value=”cheese” /><label for=”cheese”>Cheese</label>

<input id=”veggie” type=”radio” name=”pizza” value=”veggie” /><label for=”veggie”>Veggie</label>

<input id=”pepperoni” type=”radio” name=”pizza” value=”pepperoni” /><label for=”pepperoni”>Pepperoni</label>

<input id=”sausage” type=”radio” name=”pizza” value=”sausage” /><label for=”sausage”>Sausage </label>

</fieldset>

There’s no point in using a fieldset without a legend, so always make sure the checkboxes or radio button groups warrant needing a legend. Sometimes the meaning of the group can be determined by the labels of the inputs alone, for example, radio buttons that say “male” or “female” or a single checkbox that says “I agree to these terms” or “is billable”. In this case, a fieldset and legend shouldn’t be used because the purpose is clear and addition of a legend could just confuse the user.

The American Foundation for the Blind also suggests that you could include the fieldset into the label of the first option. This may be accessible, but requires the developer to add breakpoints (<br>) between what should be the legend and the first option’s label. <br> tags are hard to standardize across devices and browsers and are not semantic. Therefore, this option should not be used.