ACC=SS university

Top 10 Takeaways

ARIA for Developers II

1 **What are Grids?**

Grids allow users to navigate tabular cells of information, and understand the relationship between each data cell and the header under which it falls. Grids are keyboard focusable and should be navigated using the arrow keys

2 **Grids: Structure of Controls within Grids**

The first container should always have a role equal to grid. The rows should have a role equal to row and column headers should have a role equal to columnheader. Each element has a unique ID associated with it. Inside each column header contains the text within a given data cell in the grid with role equal to gridcell. In order to associate an individual data cell with its corresponding header, use aria-describedby on the role=”gridcell” node to point to any relevant role=columnheader or role=rowheader cell nodes.

3 **Forms: Grouping Controls**

When groups of form controls exist, such as Billing Address and Shipping Address, it is important to provide accessible labels to distinguish between the form fields. Use aria-labelledby to make the association between form field and label. Aria-label can be used to give the group a name (i.e. aria-label=”Billing” or aria-label=”Shipping”). Note that these groups of form fields can also be made accessible using the fieldset and legend elements.

4 **Forms: Required Fields**

To identify required fields using ARIA, use aria-required=”true” on the required form fields. It is not necessary to set aria-required=”false”, because not setting this attribute at all is the same thing as setting it to false.

5 Error Messages: Alert Boxes

Alert boxes are used to get the user's attention and are detected by assistive technology without additional code. A disadvantage is they can be disruptive and may require the user to then locate the error after the alert is closed. To implement error messages in an accessible manner, set the screen reader's focus to the error status text that appears at the beginning of the form after submit. This element has an id attribute value of "text-status" and has a tabindex="-1". Use the setFocus method to move focus to the newly appeared status message.

6 Error Messages: Simulated Dialogs

The ARIA role alertdialog can be used to create a live region that causes the alert's text to be announced by screen readers.

7 Sliders: Keyboard Accessibility

There are expected keyboard behaviors that must be provided for custom widgets. For instance, set focus to the slider using the tab key. For horizontal sliders, users can move the indicator using the left and right arrow keys. For vertical sliders, the up and down arrow keys should move the indicator. Additionally, press the End key to move to the end of the slider and the Home key to move to the beginning of the slider. To jump by small increments, such as 10%, use the PageUp or PageDown keys. Mouse users can still drag the slider as usual.

8 Sliders: ARIA Attributes

Set role="slider" on the slider. Provide a label using aria-label. Set the highest and lowest possible values using aria-valuemin and aria-valuemax attributes. aria-valuenow attribute defines the slider's current value in that range.

9 Accordions: Navigating Accordion Headers

Users should be able to use the Tab key or the Up and Down arrow keys to navigate between headers on an accordion. The Enter key should expand the section to see the options or content inside. Set the role attribute to region, as there is no role="accordion". Use aria-labelledby to reference the accordion headers.

10 Accordions: Expand and Collapse

Screen reader users must be able to determine the state of the accordion (i.e. expanded or collapsed). Set aria-expanded=”true” when the accordion is expanded and aria-expanded=”false” when the accordion is collapsed. Use the aria-controls attribute on the element controlling the accordion’s display of content.