

Surviving Dev Frameworks: Lessons Learned with Drupal, Wordpress, jQuery, and More

Kate Walser

Wednesday, May 15 @ 1:45 – 3:15

Thursday, May 16 @ 8:45 – 10:15

Moody 204

Please print page 2 of this doc for the handout. Cover sheet just for handout delivery to room.

Surviving Dev Frameworks: Lessons Learned...

In general...

1. Content and static images had great support. Examples: page headings, alternative text, table headings
2. Accessible interactions fell short, especially when items appeared (modals) or changed with interactions
3. Establishing standards for the most needed widgets (label + field + field help) avoids a11y issues later.

1 Changes to Consider

1. Enable keyboard access (via keyup, keydown, keypress listeners or href on links) to items that change or appear
 - Carousels, to let users move thru the carousel items and access information
 - Megamenu, especially parent link that serves as trigger
 - Dialogs – set focus to first item in modal, enable Esc key to close, make Close button keyboard reachable
 - Hover help and information, via tabindex or keyboard equivalent
 - Tabs
 - Table pager links
2. Ensure that forms and form validation are as accessible as possible
 - For required elements, change * to required and include in <label></label>. Add required attribute to field.
 - If items refresh other widgets, maintain keyboard focus at widget where change occurred
 - If helper text or field descriptions exist, show them onscreen and use aria-describedby to associate descriptions (or error messages) with input / select (alternative: include keyboard reachable tooltip)
3. Adjust search results options
 - o Use Filter as label to filter results
 - o Set focus back to right spot after Previous, Next, page # (e.g., pagination / pager) links used

2 Strategy

	Task	Consider...
1.	List major features	Navigation, carousels & galleries, search / faceted drill-down, “Share” options / plugins, maps and data charts, expand / collapse accordions
2.	Describe interactions, test criteria, exemplar sites	Clicks, hovers, messages displayed, validation results Exemplars: Knowbility.org, WCAG, Assets.CMS.gov, Standards.USA.gov
3.	Pick tools with large communities	Examples, APIs, forums, resources
4.	Use forums, API, examples	Framework API sites Developer FAQs / test areas: StackOverflow, Quora, JSFiddle.net, CodePen.io, Github
5.	Apply accessibility frameworks	Assets.CMS.gov, WordPress accessibility plug-in
6.	Establish standards for your team	Widgets most needed, accessible patterns for your team

3 Tools & Resources

1. Accessible Component Framework created by Centers for Medicare & Medicaid Services: Assets.cms.gov
2. Chrome getEventListeners, monitorEvents APIs (via View > Developer > JavaScript Console)
3. Vue.js Accessibility audit: <https://github.com/vue-a11y/vue-axe>
4. Angular.js A11y: <https://docs.angularjs.org/guide/accessibility>, <https://material.angular.io/cdk/a11y/overview>