Accessibility and Samvera

Michael Tribone, Tracy MacMath, Kate Lynch

Introductions

Michael Tribone
User Interface Developer, The Pennsylvania State University

Tracy MacMath
User Interface Programmer, Yale University Libraries

Kate Lynch
Senior Application Developer, University of Pennsylvania Libraries

Outline

- Intro to Web Accessibility
- Accessibility in Samvera Applications
  - Hyrax
  - Blacklight
- Incorporating accessibility every day
- Tools and Resources
- Next Steps

Intro to Web Accessibility

“Web accessibility means that people with disabilities can use the web.”
- Tim Berners-Lee

Intro to Web Accessibility

Perceivable

Operable

Understandable

Robust

Accessibility in Hyrax

- Primarily focused on screen readers and keyboard navigation
- Got some (a lot) work to do
- Initiatives to incorporate into development process

Accessibility in Hyrax

Basic HTML

- Forms
- Hierarchy of information
- Images, alt text, icons

Accessibility in Hyrax

Basic HTML

- Forms
    - Grouping, fieldsets, and legends
        - Everyone benefits from well-organized forms
    - Labels
        - Form controls need labels
- Hierarchy of Information
    - Document Outline
    - Properly nested H1 to H6

Accessibility in Hyrax

Color and Images

- Images, alternative text, and icons
  - Decorative versus informative
  - alt=”J098zb128?file=thumbnail”
  - Icons, icon fonts, and descriptive text
  - Content or function of image
- Color contrast
  - Foreground and background colors
  - Links, buttons, helper text, dropdowns, form controls, etc.
  - Deficient color vision and low vision
  - Text size is important

Accessibility in Hyrax

Keyboard Access

"A major blocker preventing access for screen reader users and many people with mobility issues in their hands are functions that can only be accessed with a mouse and have no keyboard alternatives."

- Tabbing through an interface
- Interacting with user interface elements
- Performing actions
- Again basic HTML and leveraging HTML5 and ARIA

Accessibility in Blacklight

- Accessible features in Blacklight
- Blacklight accessibility work funneling back into Hyrax
- Where else can Blacklight inform Hyrax’s accessibility, and vice versa?

- Bootstrap

Incorporating accessibility every day

- Remediation of existing features
- Testing
- Developing new features
- Must / Should / Can
  - Gates of compliance
  - “How do I know when I’m done?”

Tools and Resources

Validators

- Compliance Sheriff & Compliance Deputy ($)
- Funnelback Understand Accessibility Auditor ($)
- Asqatasun (OS)
- W3C code validator; text editor and IDE validators

- WAVE - http://wave.webaim.org/
- Accessibility Developer Tools (Chrome plugin)
- Accessibility Bookmarklets (HTML_CodeSniffer)

Tools and Resources

Automated Testing

- AccessLint CI
- Capybara-accessible -- anyone using for RSpec?
- Axe-matchers
  - Ruby integration for aXe-core

Tools and Resources

Self-Guided Testing

- Screen reader testing
- Ad hoc switch input testing
- Other methods
  - Motor & perceptual
  - Color contrast checking

Tools and Resources

Accessibility User Testing

- Use the same test scripts as self-guided accessibility tests

Next Steps

- Accessibility Testing for Hyrax
    - https://wiki.duraspace.org/display/samvera/Accessibility+Testing+for+Hyrax
- GitHub labeling to identify accessibility-related issues/PRs
    - https://github.com/samvera/hyrax/issues?q=is%3Aissue+is%3Aopen+label%3A%22user+interface%22
    - Review and add labels
- Incorporating accessibility into Hyrax release testing (planning)
- Other stuff?

Thank you!

samvera