

Developing Accessible Rich Web Applications

Course Summary

Description

This course will explain how screen readers use an off screen model, which effectively takes a snapshot of the web page, and provides controller keys to allow the user to interact with the snapshot. This course also talks about the different types of cursors screen readers provide to help the user interact with a Web page (or the snapshot), and how a screen reader user might update their snapshot in certain modes.

It will also observe how the W3C's WAI Accessible Rich Internet Applications (WAI-ARIA) roadmap can solve these problems by allowing the author to provide role, state, and other important accessibility information.

This course will demonstrate examples of creating rich interface elements that are not native to HTML, and adding WAI-ARIA properties so that they are voiced correctly by screen readers. It will also demonstrate WAI-ARIA's live regions properties, which are a set of properties that allow the author to determine what needs voicing, when it should be voiced, and the politeness of the voicing (rude, assertive, polite, or off) for regions without the user losing their current place in the application.

Objectives

At the end of this course, students will be able to:

- Describe issues including references to their corresponding guidelines
- Demonstrate the issue using assistive technology, a particular browser type or an evaluation tool
- Describe the step by step solution of the issue including simple code examples and demonstrations
- Apply interactive sessions to the issue which attendees can participate in identifying and solving issues

Topics

- Background and Issues
- Keyboard Accessibility
- Dynamic Content
- Integrating CSS
- Describing Custom Controls
- Implementing WAI- ARIA and Exercises
- Accessibility Evaluation and Recap

Prerequisites

There are no prerequisites for this course.

Duration

One day

Developing Accessible Rich Web Applications

Course Outline

- I. Background and Issues**
 - A. The What
 - i. Major differences between web applications and traditional web sites
 - ii. Different types of disabilities, related issues and solutions
 - B. The Why
 - i. Moral obligations
 - ii. Legal obligations
 - iii. Business reasons
 - C. The How
 - i. Assistive technology
 - ii. WCAG/ARIA and TEITAC Requirements
- II. Keyboard Accessibility**
 - A. Programmatic Focus
 - i. Using natively focusable controls
 - ii. Using tab index attribute and scripting
 - B. Event Handling
 - C. Preventing Change of Use Context (cursor focus, page refresh, etc.)
- III. Dynamic Content: Traditional and ARIA Methods**
 - A. Ensuring perceivable content (visual & DOM)
 - B. Notifying user of changed content
 - C. Providing user with clear path to updated content including:
 - i. Cursor focus change
 - ii. Skip links
 - iii. Filtered views
 - D. Using ARIA Live Regions
- IV. Integrating CSS**
 - A. Using CSS to hide and show content
 - B. Implementing user-perceivable content
 - C. Conveying structure and/or meaning
- V. Describing Custom Controls**
 - A. Providing dynamic labeling for interactive controls
 - i. Prompts
 - ii. Hints
 - iii. Validation results
 - B. Understanding custom control interfaces and behavior
 - C. Implementing custom control descriptors
- VI. Implementing WAI –AIRA and Exercises**
 - A. What it is and why it will change everything
 - B. Making Web 2.0 widgets compatible with assistive technologies
 - C. Applying ARIA roles and states in XHTML or HTML
 - i. W3C roles for ARIA and
 - ii. States and Properties module for ARIA specifications
- VII. Accessibility Evaluation and Recap**
 - A. Demonstrations of different types of evaluation tools
 - B. Final course summation covering all topics discussed