

How to Implement WCAG 2.2 AA Accessibility Standards in eLearning Content

A step-by-step guide for instructional designers and content creators

Overview

Accessibility compliance isn't just a legal requirement; it's about creating inclusive learning experiences that effectively serve all users. This tutorial provides a practical framework for implementing WCAG 2.2 AA standards in your eLearning content, drawing from enterprise implementation experience across multiple learning management systems.

Prerequisites

- Access to authoring tools (Articulate Storyline 360, Rise 360, or similar)
- Basic understanding of HTML structure
- Content ready for accessibility review

Step 1: Establish Your Accessibility Foundation

1.1 Create an Accessibility Checklist

Before beginning development, establish a standardized checklist that covers the four WCAG principles:

Perceivable Content:

- All images include meaningful alt text
- Videos include captions and transcripts
- Color contrast ratios meet 4.5:1 minimum for normal text
- Content is readable when zoomed to 200%

Operable Interface:

- All functionality is keyboard accessible
- No content flashes more than 3 times per second
- Users can pause, stop, or hide moving content
- Sufficient time limits are provided for timed content

Understandable Information:

Sample Writing

Skyler Scott

- Content language is programmatically identified
- Navigation is consistent across modules
- Error messages are clear and actionable
- Instructions are provided for complex interactions

Robust Technology:

- Content works with assistive technologies
- Proper HTML semantic structure is used
- Form elements include appropriate labels

1.2 Set Up Your Development Environment

Configure your authoring tool settings for accessibility:

1. In Articulate Storyline 360:

- Go to Player > Other > Accessibility
- Enable "Include in tab order" for interactive elements
- Set focus order manually for complex slides

2. In Rise 360:

- Review the built-in accessibility features
- Customize color themes to meet contrast requirements
- Use heading structures appropriately

Step 2: Design for Screen Reader Compatibility

2.1 Structure Content Hierarchically

Organize your content using proper heading levels:

H1: Course Title

H2: Module Title

H3: Lesson Title

H4: Section Title

H5: Subsection (if needed)

Sample Writing

Skyler Scott

2.2 Write Effective Alt Text

For Informative Images:

- Describe the essential information the image conveys
- Keep descriptions concise but complete
- Example: "Bar chart showing 45% increase in employee engagement after accessibility training implementation"

For Decorative Images:

- Use empty alt text (alt="")
- Let screen readers skip purely decorative elements

For Complex Graphics:

- Provide brief alt text plus detailed description
- Consider providing data tables for chart information

Step 3: Implement Keyboard Navigation

3.1 Test Tab Order

Ensure logical tab progression through your content:

1. Use Tab key to navigate through interactive elements
2. Verify skip links work properly
3. Confirm focus indicators are clearly visible
4. Test that trapped focus works in modal dialogs

3.2 Provide Keyboard Alternatives

For mouse-dependent interactions:

- Drag-and-drop: Provide keyboard-accessible alternatives
- Hover states: Ensure information is available via keyboard focus
- Custom controls: Include ARIA labels and roles

Step 4: Optimize Color and Contrast

4.1 Check Color Contrast

Sample Writing

Skyler Scott

Use tools like WebAIM's Color Contrast Checker:

1. Test all text/background combinations
2. Verify interactive elements meet contrast requirements
3. Ensure focus indicators have sufficient contrast
4. Check that error states are distinguishable

4.2 Design Beyond Color

Don't rely solely on color to convey information:

- Use icons alongside color coding
- Employ patterns, textures, or shapes
- Provide text labels for status indicators

Step 5: Create Accessible Media

5.1 Video Accessibility

Captions:

- Include speaker identification
- Describe relevant sound effects
- Ensure timing accuracy
- Format consistently

Audio Descriptions:

- Describe visual information not conveyed through dialogue
- Insert during natural pauses
- Keep descriptions concise and objective

5.2 Interactive Elements

Form Controls:

```
<label for="email">Email Address (required)</label>
```

```
<input type="email" id="email" name="email" required aria-describedby="email-error">
```

```
<div id="email-error" role="alert">Please enter a valid email address</div>
```

Sample Writing

Skyler Scott

Step 6: Test and Validate

6.1 Automated Testing

Use accessibility testing tools:

- WAVE Web Accessibility Evaluator
- axe DevTools
- Lighthouse accessibility audit

6.2 Manual Testing

Conduct thorough manual reviews:

1. Keyboard-Only Navigation:

- Unplug your mouse
- Navigate using only Tab, Enter, Space, and arrow keys
- Verify all functionality remains accessible

2. Screen Reader Testing:

- Test with NVDA (free) or JAWS
- Listen to content flow and comprehension
- Verify proper announcement of interactive elements

3. Zoom Testing:

- Test at 200% zoom level
- Ensure content remains readable and functional
- Check for horizontal scrolling issues

Step 7: Document and Maintain

7.1 Create Accessibility Statements

Document your accessibility features:

- Compliance level achieved (WCAG 2.2 AA)
- Known limitations
- Contact information for accessibility feedback

Sample Writing

Skyler Scott

- Date of last review

7.2 Establish Review Cycles

- Schedule quarterly accessibility audits
- Update content as WCAG guidelines evolve
- Train team members on accessibility best practices
- Maintain testing documentation

Troubleshooting Common Issues

Issue: Low Color Contrast

- Solution: Adjust color palette or add text shadows/outlines

Issue: Images Without Alt Text

- Solution: Audit all visual content and add descriptive alt text

Issue: Keyboard Trap

- Solution: Implement proper focus management and escape mechanisms

Issue: Missing Form Labels

- Solution: Associate all form elements with descriptive labels

Measuring Success

Track accessibility improvements through:

- User feedback and support requests
- Compliance audit scores
- Task completion rates across user groups
- Time-to-completion metrics for keyboard users

Conclusion

Implementing WCAG 2.2 AA standards requires systematic planning and consistent execution, resulting in learning content that effectively serves all users. By following this framework and establishing regular review processes, you can create truly inclusive learning experiences that meet both compliance requirements and user needs.

Sample Writing

Skyler Scott

Remember: Accessibility is not a one-time task but an ongoing commitment to inclusive design. Start with these fundamentals, then continuously refine your approach based on user feedback and evolving best practices.